Web 服务组合中行为兼容性分析与优化控制策略

毕 敬[1],朱志良[1,2],范玉顺[3]
(1.东北大学信息科学与工程学院,辽宁沈阳 110004; 2.北京仿真中心航天系统仿真重点实验室,北京 100854;
3.清华大学自动化系,北京 100084)

摘 要: 针对 Web 服务组合中交互行为不兼容问题,本文提出了基于 Petri 网的优化控制策略来规避.首先,给出了多个服务交互行为不兼容的实例.其次,给出了受控服务组合的形式化定义,并根据约减规则生成了约减的服务组合网状态可达图,从而识别出死锁状态和无死锁状态.在导致死锁状态的关键变迁上添加相应的控制库所和弧,并结合最大允许反馈控制策略,从而推导出最优控制器,并证明了该方法的正确性.此外,通过服务组合中交互行为不兼容的实例证实了基于最优控制器策略的有效性.最后,本文将最优控制器模型转换成 BPEL.

关键词: Web 服务组合; 兼容性; 最优控制器; 死锁避免
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2011) 12-2842-08

Behavioral Compatibility Analysis and Optimal Control Policy in Web Services Composition

BI Jing[1], ZHU Zhi-liang[1,2], FAN Yu-shun[3]
(1. *School of Information Science and Engineering*, *Northeastern University*, *Shenyang*, *Liaoning* 110004, *China*;
2. *Beijing Simulation center*, *Science and Technology on Space Simulation Laboratory*, *Beijing* 100854, *China*;
3. *Department of Automation*, *Tsinghua University*, *Beijing* 100084, *China*)

Abstract: A Petri net based optimal control policy is proposed for the check of behavioral incompatibility in web services composition. According to a real case of multiple services interaction, this paper presents a formalized definition of controlled service composition, and the reduced state reachability graph of service composition net is given according to the reduce rules, thus the deadlock states and the deadlock-free states are identified. With the maximally permissive feedback control strategy developed, the appropriate control place and arc are appended in the key transition which can lead to deadlock states. Thus the appropriate optimal controller is developed, the proposed approach is verified. In addition, for the behavioral incompatibility case, a policy of appending optimal controller is presented. It is proved that our policy can be a good solution. Finally, the proposed controller is transformed as the activity of BPEL.

Key words: web services composition; compatibility; optimal controller; deadlock prevention

1 引言

在过去的几年,面向服务体系架构(SOA)和 Web 服务作为 SOA 最广泛使用的执行技术已经变得十分流行.特别在 Web 服务组合方面,因其能实现 Web 服务的重用和增值而成为学术界和工业界共同关注的焦点,在面向服务模式下的应用设计具有较高的灵活性,容易重用并且减少开发成本.因此,一种分布式应用的新方式已经出现.应用可以被现存的组件(如 Web 服务)组合,其现存的组件可以通过互联网上的第三方来提供(采用软件即服务(SaaS)业务模型)或者应用服务开发者来提供(在本地的网络内或者在互联网上).当使用 Web 服务作为执行技术,这些组合的 Web 服务常常被称为业务流程或者更一般的称为工作流.

Web 服务组合语言 BPEL(Business Process Execution Language for Web Services)[1]是业务流程模型事实上的工

收稿日期:2010-05-01;修回日期:2010-10-20
基金项目:国家自然科学基金(No.60872040)

业标准,并且是一种整合商业应用或科学应用的工具,其运行在网格或者云环境中.它能够构建复杂的 Web 服务组合,即流程服务,并采用执行引擎通过 Web 服务接口(Web 服务描述语言,WSDL[2])来访问流程服务,同时该流程服务可以作为一个基本的活动在其它的流程服务中被使用.这种软件开发不同于传统的方法,流程服务是松耦合的通过消息交互控制在执行引擎中.进一步,流程服务不必安装在相同的机器上或者相同的网络上.通常,各个流程服务是单独开发的,且在开发的时候往往无法预见流程服务组合的所有情况;甚至被使用的流程服务不属于开发者的管理域,即开发者没有控制权力在该流程服务上.由于这些方面的事实和组合流程服务的分布本质,使得各流程服务组合成员之间的交互行为存在部分兼容现象是十分普遍的.部分兼容是指两个或多个流程服务提供互补的功能,但是它们的接口或交互模式并不完全匹配.在部分兼容的情况下,两个流程服务是无法直接组合的,从而影响流程服务组合功能的可用性.

对 Web 服务组合的可用性分析,需要综合考虑各 Web 服务的行为.目前,描述 Web 服务组合行为的形式化语义有基于图的方法[3]、基于时序逻辑的方法[4]、基于并发事务逻辑的方法[5]、基于有限状态机的方法[6]、基于进程代数的方法[7]和基于 Petri 网的方法[8,9]等.由于 Petri 网适合于对 Web 服务这种松散耦合的分布式系统进行建模,因此本文采用 Petri 网作为服务流程分析的理论基础,特别在 Web 服务组合语言 BPEL 的验证方面.然而目前研究缺乏对 BPEL 服务流程进行深入讨论,没有明确指出如何进行两个或多个服务流程的可用性分析.进一步地,目前研究工作也都没有涉及到基于控制器的 BPEL 服务流程组合方法的研究.

本文采用 Petri 网实现 Web 服务交互行为的建模,在此基础上,借助 Petri 网的形式化语义、可达图分析和结构分析等基础理论,生成最优控制器,解决服务组合过程中存在的行为部分兼容现象.本文将通过一个实例来实现给出的概念和算法.

2　一个例子

下面的例子说明了服务组合中的行为不兼容问题.描述了客户、航班预定系统服务和第三方结算服务(Third Party Checkout,简称 TPC)的组合.航班预定系统为客户提供网上购票的服务,并允许将购票流程中的结算环节外包给第三方.也就是说,在客户确定预定并准备结账的时候,他们可以被引导到一个第三方结算服务提供商的站点进行结算.

客户、航班预定和 TPC 服务的 BPEL 流程,如图 1 所示.其中客户端服务流程(即,WS1)是:

(1)客户调用航班预定服务,并传递航班预定消息给航班预定服务.

(2)接收来自航班预定服务的机票消息.

(3)客户调用 Approve 或 Cancel 操作给航班预定服务,并返回 Approve 或 Cancel 消息给航班预定服务.

(4)当调用 Approve 操作时,则进入付款操作,并等待 TPC 服务返回付款消息;当调用 Cancel 操作时,则客户结束预定流程.

(5)调用 Visa 或 Master 操作,并返回 Visa 或 Master 消息给 TPC 服务.

航班预定服务的流程(即,WS2)是:

(1)接收到客户的航班预定之后进入提供机票状态.

(2)调用预定机票操作,并返回机票消息给客户端服务.

(3)启动 Approve 或 Cancel 接收活动异步等待客户返回 Approve 或 Cancel 消息.

TPC 服务的流程(即,WS3)是:

(1)接收来自航班预定服务的付款通知,并进行结算活动.

(2)调用付款操作,并返回付款消息给客户端服务.

(3)启动 Visa 或 Master 接收活动异步等待客户返回 Visa 或 Master 消息.

为完成这项业务,需要对客户、航班预定和 TPC 服务进行组合.从上面描述可以看到,三个服务提供互补的功能,但是由于是独立开发的,它们之间的行为不完全兼容.也就是说,它们之间提供互补的功能并且在各自的流程中能够正确地执行.然而,它们之间的操作交互模式不完全匹配.

因此,现有的直接组合方法难以解决服务组合行

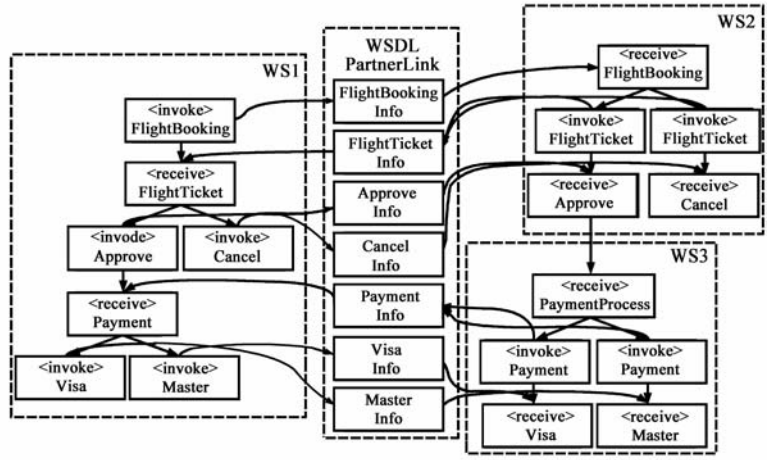

图1　客户、航班预定和TPC服务的BPEL流程

为部分兼容的情况.在此例中,假设三个服务的操作之间交互的消息接口是完全匹配的,只是它们之间服务操作的顺序不一致.如客户服务等待用 Master 来接收结算业务,而 TPC 服务却在调用 Visa 来支付结算业务,难以执行正确的交互活动.为解决服务组合过程中服务交互行为不一致问题,本文提出基于控制器的方法,在不改变已有服务流程内部业务逻辑的情况下,通过添加控制器来避免服务组合过程中部分兼容现象的发生.

为解决上述问题,我们接下来给出基于控制 Petri 网的服务组合定义,以方便后续分析.

3 受控服务组合的形式化定义

针对服务组合过程中行为部分兼容的现象,本文建立控制器来解决此问题.控制器可看作服务间的一个中间模块,它能够通过控制消息的执行来弥合服务间的行为不一致,使得部分兼容的服务能够协同工作.从形式化模型上来说,控制器是一个 Petri 网[10],其定义如下:

定义 1 (控制器,Controller)一个控制器是一个三元组 $C=(P_C,T_C,B,m_{c0})$,当且仅当:

(1)P_C 是控制库所的集合;

(2)T_C 是可控变迁的集合;

(3)$B\subseteq(P_C\times T_C)\cup(T_C\times P_C)$是控制库所 P_C 与可控变迁 T_C 之间有向弧的集合,它的元素称为控制弧;

(4)m_{c0}是控制器的初始标识.

其中,一个控制 $u:P_C=\{0,1\}$分配一个双重的托肯给每一个控制库所.所有控制的集合记为 U.对于 $\forall p_c\in P_C$,u_{zero}被定义为 $u(p_c)\equiv 0$;u_{one}被定义为 $u(p_c)\equiv 1$.给定两个控制 $u',u\in U$,如果对于 $\forall p_c\in P_C$ 使得 $u(p_c)\leqslant u'(p_c)$并且对于 $\exists p_c\in P_C$ 使得 $u(p_c)<u'(p_c)$,那么控制 u'被认为是允许的.

结合服务流网[11]和控制器,给出如下的基于控制的服务组合网定义,用于描述受控的服务组合.

定义 2 (基于控制的服务组合网,Controller based Composition of Web Services Net)给定 n 元服务流网 $\text{WSN}_i=(P_{Ii}\cup P_{Mi},T_i,F_i,m_{0i}),(i=1,2,\cdots,n)$,通过添加控制器 C,得到一个控制服务组合网 $\text{CWSN}^C=(P,T,F,C,A,m_0)$,当且仅当:

(1)P 是服务组合网库所的有限集合.$P=P_I\cup P_M\cup P_C\cup\{p_\alpha,p_\beta\}$,且

(a)$P_I=P_{I1}\cup P_{I2}\cup\cdots\cup P_{In}$是服务组合网内部库所的有限集合,其中,$P_{Ii}$是第 i 个服务流网内部库所的集合,且 $p_{ai}\in P_{Ii}$和$p_{\beta i}\in P_{Ii}$分别是第 i 个服务流网的起始库所和终止库所;

(b)$P_M=P_{M1}\cup P_{M2}\cup\cdots\cup P_{Mn}$是服务组合网消息库所的有限集合,其中,$P_{Mi}$是第 i 个服务流网消息库所的集合;

(c)p_α,p_β 分别是服务组合网的起始库所和终止库所,${}^\bullet p_\alpha=\varnothing,p_\beta^\bullet=\varnothing$;

(d)$P_{Ii}\cap P_{Mi}\cap P_C\cap\{p_\alpha,p_\beta\}=\varnothing$;

(2)$T=(T_1\cup T_2\cup\cdots\cup T_n)\cup T_C\cup\{t_\alpha,t_\beta\}$是服务组合网变迁的有限集合,

(a)t_α,t_β 分别是服务组合网的起始变迁和终止变迁;

(b)如果在 WSN_i 中存在连接库所 p_{ai}和 $p_{\beta i}$的变迁 t_i^*(即 ${t_i^*}^\bullet=p_{ai},{}^\bullet t_i^*=p_{\beta i}$),则 WSN_i 是强连通的;

(c)$(T_1\cup T_2\cup\cdots\cup T_n)\cap T_C\neq\varnothing$,$(T_1\cup T_2\cup\cdots\cup T_n)\cap\{t_\alpha,t_\beta\}=\varnothing$,$T_C\cap\{t_\alpha,t_\beta\}=\varnothing$,$T\cap P=\varnothing$;

(3)$F=(F_1\cup F_2\cup\cdots\cup F_n)\cup\{(p_\alpha,t_\alpha),(t_\alpha,p_{\alpha 1}),\cdots,(t_\alpha,p_{\alpha n}),(p_{\beta 1},t_\beta),\cdots,(p_{\beta n},t_\beta),(t_\beta,p_\beta)\}$,且 $F\subseteq(P\times T)\cup(T\times P)$是服务组合网 P 和 T 之间有向弧的集合,$F\cap B\neq\varnothing$;

(4)C 是控制器,由定义 1 给出;

(5)A 是服务组合网公共消息库所的集合,$A=\{A_1,A_2,\cdots,A_{n-1},\cdots\}=\{P_{M1}\cap P_{M2},P_{M2}\cap P_{M3},\cdots,P_{M(n-1)}\cap P_{Mn},\cdots\}$;

(6)m_0 是服务组合网的初始标识.

基于定义 1 和定义 2 可以给出如下的服务组合兼容性的定义.

定义 3 (服务组合的兼容性,Compatibility of Composition Service)给定服务流网 $\text{WSN}_i=(P_{Ii}\cup P_{Mi},T_i,F_i,m_{0i}),(i=1\cdots n)$,及其直接组合 $\text{CWSN}=\text{WSN}_1\otimes_{A_1}\text{WSN}_2\otimes_{A_2}\cdots\otimes_{A_{n-1}}\text{WSN}_n$.$m_0=m_{01}\times m_{02}\times\cdots\times m_{0n}$,$m_e=m_{e1}\times m_{e2}\times\cdots\times m_{en}$,其中 m_{0i}和 m_{ei}是 WSN_i 的初始和终止标识.$\text{WSN}_1,\text{WSN}_2,\cdots,\text{WSN}_n$ 关于 A 是兼容的,当且仅当 $\text{CWSN}=\otimes_{i=1}^n\text{WSN}_i$ 的可达图 R 是良构的,即:

(1)$\forall m\in R(\text{CWSN},m_0)$,存在一个变迁序列 $\sigma=<t_1,t_2,\cdots,t_n>\in T_1\cup T_2\cup\cdots\cup T_n$ 和一个标识 m_σ 使得 $m[\sigma\rangle m_\sigma$ 且 $m_\sigma\geqslant m_e$;

(2)对于标识 $m\in R(\text{CWSN},m_0)$且 $m\geqslant m_e$,如果 $\exists p\in P$ 使得 $m(p)>m_e(p)$,那么 $p\in P_{M1}\cup P_{M2}\cup\cdots\cup P_{Mn}$.

定义 4 (基于控制器的服务组合兼容性,Controller based Compatibility of Service Com-position)服务组合网 $\text{CWSN}=\otimes_{i=1}^n\text{WSN}_i$ 关于控制器 C 是兼容的,当且仅当 $\otimes_{i=1}^n\text{WSN}_i\otimes_{P_C}C$ 的可达图是良构的.即存在控制器 C 使得服务组合网 CWSN 关于控制器 C 是兼容的.

本节给出控制器和基于控制的服务组合网的形式化描述,并给出了服务组合的兼容性定义.下一节将给

出服务组合兼容性的判定方法.

4　服务组合网的兼容性分析和策略

为了分析 Web 服务组合的兼容性,本文将第 2 节中的客户预定航班交互流程映射为服务组合网,如图 2 所示.

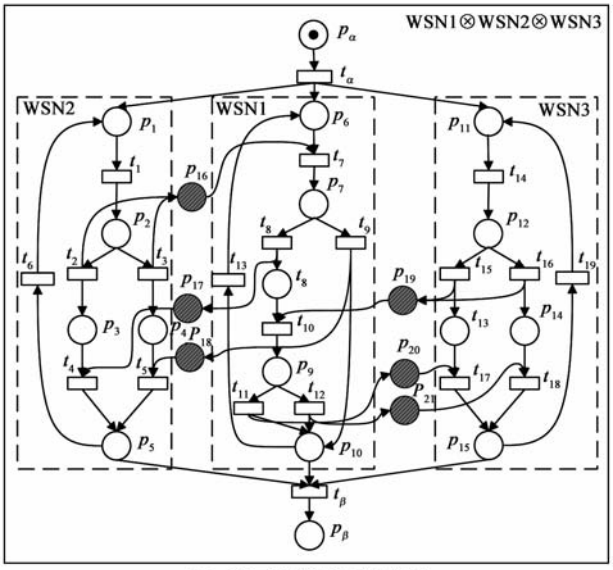

图2　不可用的服务组合网

从图 2 中可以看出,客户端服务 WSN_1 的内部库所集为 $P_{I1}=\{p_7,p_8,p_9\}$,起始和终止库所分别为 $p_{\alpha 1}=\{p_6\}$和 $p_{\beta 1}=\{p_{10}\}$,操作变迁集为 $T_1=\{t_7,t_8,t_9,t_{10},t_{11},t_{12}\}$,$t_1^*=\{t_{13}\}$;航班预定服务 WSN_2 的内部库所集为 $P_{I2}=\{p_2,p_3,p_4\}$,起始和终止库所分别为 $p_{\alpha 2}=\{p_1\}$和 $p_{\beta 2}=\{p_5\}$,操作变迁集为 $T_2=\{t_1,t_2,t_3,t_4,t_5\}$,$t_2^*=\{t_6\}$;TPC 服务 WSN_3 的内部库所集为 $P_{I3}=\{p_{12},p_{13},p_{14}\}$,起始和终止库所分别为 $p_{\alpha 3}=\{p_{11}\}$和 $p_{\beta 3}=\{p_{15}\}$,操作变迁集为 $T_3=\{t_{14},t_{15},t_{16},t_{17},t_{18}\}$,$t_3^*=\{t_{19}\}$;公共消息库所集为 $A=\{p_{16},p_{17},p_{18},p_{19},p_{20},p_{21}\}$;服务组合网的起始和终止库所分别为 p_α 和 p_β,起始和终止变迁分别为 t_α,t_β.为了简化起见,本文将图 1 中的客户调用航班预定操作和航班预定消息接口省略掉.

4.1　服务组合网的兼容性分析

为了检测服务的兼容性,即是否存在基于控制器的组合.这里首先将图 2 服务组合网的初始行为标识为一系列的可达状态,并通过稳固集的方法[12]来约减状态空间,这个约减的状态空间消去了部分与兼容性检测无关的状态,使得服务组合网的状态空间的产生得到了优化.简单来说,通过选择合适的稳固集,得到的服务组合网的状态空间是一个约减的状态空间.下面给出稳固集的概念.

定义 5　(稳固集,Stubborn Sets)[12]在标识 M 下,一个 Petri 网的稳固集定义为变迁 T_S 的集合:

(1)$(\exists t\in T,|M[t\rangle)\Rightarrow \exists t\in T_S|M[t\rangle$;

(2)如果 $t\in T_S$ 并且 $M[t\rangle$,那么$({}^\bullet t)^\bullet\subseteq T_S$;

(3)如果 $t\in T_S$ 并且 $\neg M[t\rangle$,那么 $\exists p\in{}^\bullet t, M(p)<I^-(p,t)$并且${}^\bullet p\subseteq T_S$.

稳固集是定义在每个状态上的可实施变迁集.在每个状态中,应该能够选择与其他可实施变迁子集相互独立的变迁子集予以实施.当稳固集外的变迁发射时,这个集合仍然保持稳固.例如,设 T_S 是状态 M 的稳固集,如果有 $t\in T_S,t_1,t_2,\cdots,t_n\notin T_S$,且 $M[t_1,t_2,\cdots,t_n\rangle M_n[t\rangle M_n'$,那么存在状态 M' 使得 $M[t\rangle M'[t_1,t_2,\cdots,t_n\rangle M_n'$.

结合稳固集的方法,生成了如图 3 所示约减后的服务组合网状态可达图(Reduced Reach-ability Graph, RRG),可以看出约减后的状态空间分为无死锁状态和死锁状态,即空心圆表示无死锁的状态,实心圆表示死锁的状态.其中,禁止(死锁)状态集为 $M_F=\{m_6,m_7,m_{18},m_{19}\}$;允许(无死锁)状态集为 $\overline{M_F}=\{m_0,m_1,m_2,m_3,m_4,m_5,m_8,m_9,m_{10},m_{11},m_{12},m_{13},m_{14},m_{15},m_{16},m_{17},m_{20},m_{21},m_{22}\}$;死锁变迁集(Deadlock Transition Domain,DTD)用虚线箭头表示,即 $DTD=\{t_8,t_9,t_{11},t_{12}\}$和无死锁变迁集(Deadlock-Free Transition Domain,DFTD)用实线箭头表示,即 $DFTD=\{t_2,t_3,\cdots,t_{19}\}$.在死锁变迁集和无死锁变迁集中都出现了$\{t_8,t_9,t_{11},t_{12}\}$,因此要对该变迁集加以控制,避免死锁现象的发生.

结合图 3 的服务组合网状态可达图,其中,死锁状态标识为 $m_6(p_3,[0,0,1],p_{10},[0,0,0],p_{12})$,$m_7(p_4,[0,1,0],p_8,[0,0,0],p_{12})$,$m_{18}(p_2,[0,0,0],p_{10},[0,0,1],p_{13})$,$m_{19}(p_2,[0,0,0],p_{10},[0,1,0],p_{14})$,也就是 RRG

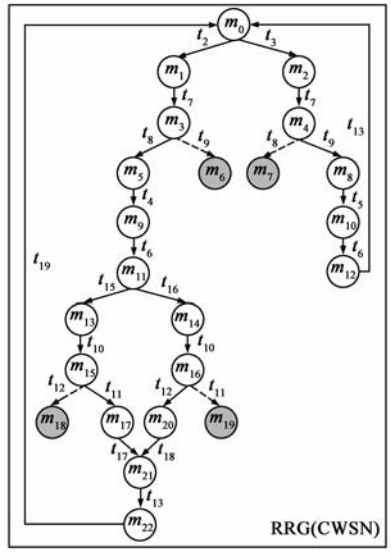

图3　服务组合网约减的状态可达图

(CWSN)中存在死锁状态$\{p_3,p_{18},p_{10},p_{12}\}$,$\{p_4,p_{17},p_8,p_{12}\}$,$\{p_2,p_{10},p_{21},p_{13}\}$,$\{p_2,p_{10},p_{20},p_{14}\}$.从保持服务组合网的良性运行的角度出发,把它们列为服务组合网运行中禁止出现的状态.

为了构建服务组合网行为不可用情况下的应对策略,即避免死锁的优化控制策略.目的是获得活的(比如无死锁的)服务组合网模型.下面提出的可控变迁集的作用是,如果存在行为不兼容的服务组合网,使用控制变迁来避免死锁状态的发生.

定义 6 (可控变迁集,Controlled Transition Set).对于控制服务组合网,$m\in R(\mathrm{CWSN}^C,m_0)$,$T_C\in T$,如果$m[T_C>m_f$,并且 $m_f\in M_F$,其中,m_f 为服务组合网的禁止状态,M_F 为禁止状态的集合,则称 T_C 为 m 的可控变迁集.

服务组合网的禁止状态 m_f 就是需要控制的状态,通过可控变迁来避免禁止状态的发生.对于控制服务组合网,$m\in R(\mathrm{CWSN}^C,m_0)$,如果控制 $u\in U=\{u_{\mathrm{zero}},u_{\mathrm{one}}\}$满足$\forall t_c\in T_C$,且$\exists p_c\in P_C$ 使得 $u(p_c)=0$,则控制 u 是 m 的允许反馈控制.对于控制服务组合网,$m\in R(\mathrm{CWSN}^C,m_0)$,如果 $T_C\neq\varnothing$,则控制 $u\in U$ 为 m 的最大允许反馈控制的充要条件是:(1)$\forall t_c\in T_C$,$\exists p_c\in P_C$ 使得 $u(p_c)=0$;(2)$\forall p_c\in P_C$,如果 $u(p_c)=0$,则 $\exists t_c\in T_C$ 且 $p_c\in P_C$,对于 $\forall p_c'\in P_C$且 $p_c'\neq p_c$都有 $u(p_c')=1$.如果 $T_C=\varnothing$,显然 u_{one}就是最大允许反馈控制.最大允许反馈控制的作用在于它既实现了服务组合网的禁止状态避免,又最大限度地保留了服务组合网原有的可达状态.

接下来,本文给出了服务组合网的控制约减状态可达图(Controlled Reduced Reachability Graph,CRRG)的构造算法,如图 4 所示.

算法 1 CRRG 的构造方法

输入:流程服务网 $\mathrm{WSN}_i=(P_{Ii}\cup P_{Mi},T_i,F_i,M_{0i})$,$i=1,2,\cdots,n$;稳固集 T_S;最大允许反馈控制策略 U

输出:$\mathrm{CRRG}(\mathrm{CWSN},U)=(V^C,E^C)$,其中 V^C 是带有控制器的系统状态的集合,E^C 是带有控制器的引起系统状态变化变迁的集合.

步骤 1 构造 CWSN 约减的状态可达图 $\mathrm{RRG}(\mathrm{CWSN})=(V,E)$,其中 V 是系统状态的集合;E 是引起系统状态变化变迁的集合.

(1)初始化$(V,E)=(\{M_0=M_{01}\times M_{02}\times\cdots\times M_{0_n}\},\varnothing)$;$M_0$ 未被标记,并将标记值设为 false;

(2)若 V 中无未被标记的节点时,RRG 的构造过程结束转向步骤 2,否则继续下面的步骤;

(3)当 V 中还有未被标记的节点时

(a)选择一个未被标记的节点 $M\in V$,将其标记值设为 true(记 $M=M_1\times M_2\times\cdots\times M_n$);

(b)对标识 M 下的每一个可实施的变迁 $t\in T_S$

①计算 $M':M\xrightarrow{t}M'$;

②如果 $\exists M'\in V$,则 $E=E\cup\{(M,t,M')\}$,转向步骤 1 中的(4),否则继续下面的步骤;

③如果 $\exists M''\in V$,使得 $\forall p\in P$,都有 $M''\xrightarrow{\sigma}M'$,$M''(p)\leqslant M'(p)\wedge M''(p)\neq M'(p)$,并且 $\exists p\in P\wedge p\notin P_M$,都有 $M''(p)<M(p)$,算法退出并报错(不存在有界的可达图);

④如果 $\exists M''\notin V$,使得 $M''(p)=M'(p)$,则 $V=V\cup\{M'\}$,$E=E\cup\{(M,t,M')\}$,M' 未被标记,并将 M' 的标记值设为 false;

(4)将 M 的未被标记划去,并根据定义 5 中的 3 个前提条件,生成 RRG,然后转向步骤 1 中的(2);

步骤 2 如果 RRG 中存在死锁状态,继续下面的步骤,否则不存在控制器 C,使得 $M_{c0}(p_c)=0$,即 $\mathrm{RRG}(\mathrm{CWSN})=(V,E)=(V^C,E^C)$,转向步骤 4;

(1)根据 CWSN 禁止状态相关的库所集 M_F,计算可控变迁集 T_C,令 p_c 为 t_c 的控制库所,由此得到控制库所集 P_C;

(2)将任何节点 $M\in V$ 的标识向量长度由 $|P_I|+|P_M|$ 增长到 $|P_I|+|P_M|+|P_C|$;

(3)将相应的 $|P_C|$ 那部分子向量的值填入 $\mathrm{RRG}(\mathrm{CWSN})=(V,E)$中,对于 $\forall M',M''\in V$,则 $V^C=V\cup\Gamma_{p_c\in P_C}(M)$,其中 $\Gamma_{p_c\in P_C}(M)$为 M 在 P_C 上的投影子向量;

(4)如果$(M',M'')\in E$,并且对于标识 M 下的每个可实施的可控变迁 $t_c\in T_C$,$M'\xrightarrow{t_c}M''$,则 $E^C=E\cup\{(\Gamma_{p_c\in P_C}(M'),T_C,\Gamma_{p_c\in P_C}(M''))\}$;

步骤 3 根据最大允许反馈控制策略 U

(1)计算标识 M 的所有可控变迁集 T_C,满足定义 6,则

(2)设 $T_C=\varnothing$,计算变迁集 T;

(a)如果对于 $\forall T_C\in T$,则有 $M[T_C>M_f$;

(b)如果 $M_f\in M_F$,即 M_f 为禁止状态,则有 $t_c\in T_C$;

(3)如果 $T_C=\varnothing$,则 $u(p_c)=u_{\mathrm{one}}$;否则由 U 充要条件求得控制 $u(p_c)$;

步骤 4 算法结束,返回 CRRG(CWSN,U).

图 4 CRRG 的构造方法

CRRG 的基本思路是,首先,根据服务组合网构建约减的状态可达图 RRG.其次,判断 RRG 是否存在死锁状态,如果存在死锁状态,则生成相应的控制状态可达图.再次,根据最大允许反馈控制策略 U,生成最优的控制状态.最后,得到可用的控制服务组合网的状态可达图.可以看出,如果一个完整的 CWSN 由 n 个 WSN 组成,每一个 WSN 有 m 个执行步骤,那么一个完整的 CWSN 就有 m^n 个取值.如果不同的 CWSN 的取值产生不同的子网,那么将得到 m^n 个子网.此时为了得到所有的子网,算法复杂度为 $O((|V^C|+|E^C|)\times m^n)$.

给定客户服务 WSN_1、航班预定服务 WSN_2 和 TPC

服务 WSN_3(图 2),以及最大允许反馈控制策略 U,根据算法 1 可以得到控制约减的状态可达图 $CRRG(WSN_1, WSN_2, WSN_3, U)$.接下来给出定理,如果控制器 C 存在,那么服务流网 WSN_1、WSN_2 和 WSN_3 的组合就是可用的.

定理 1　给定服务流网 $WSN_i, i = 1,2,\cdots,n$,以及最大允许反馈控制策略 U,存在一个符合 U 的最优控制器 C 使得服务组合网 CWSN 关于控制器 C 是可用的,当且仅当控制约减的状态可达图 $CRRG(CWSN, U)$ 是良构的,即:

(1) $\forall m \in R(CWSN, m_0)$,存在一个变迁序列 σ 和一个标识 m_σ 使得 $m[\sigma\rangle m_\sigma$ 且 $m_\sigma \geqslant m_e$(m_e 是终止标识);

(2)对于标识 $m \in R(CWSN, m_0)$ 且 $m \geqslant m_e$,如果 $\exists p \in P$ 使得 $m(p) > m_e(p)$,那么 $p \in P_{M1} \cup P_{M2} \cup \cdots \cup P_{Mn}$.

根据定理 1,可以断定服务流网 WSN_1、WSN_2 和 WSN_3 的组合 CWSN 关于控制器 C 是可用的,也就是说控制器 C 是存在的.容易看出,通过添加额外控制器的方法使得不相兼容的两个或多个服务可用.该方法不仅能解决服务组合的可用性问题,而且避免了当前常用的基于替换网络服务环境方法带来的求解复杂问题.

基于替换网络服务环境方法的理论基础是探测网络服务组合的所有可达状态空间.然而,基于可达状态空间的分析方法在判断每个完整的 CWSN 时,存在状态空间爆炸的问题.与该方法相比,基于添加额外控制器的可用性策略可以带来极大的方便.算法可以通过修改相应的 WSN 中导致部分不兼容的情况而不是替换全部的 WSN 从而使得可用性得到满足.另外,如果一个完整的 CWSN 由 n 个 WSN 组成,每一个 WSN 有 m 个执行步骤,那么其执行的状态空间为 $(m+1)^n$.此时通过稳固集的方法来约减状态空间可以使得状态空间约减为 $nm+1$,即约减掉了与可用性检测无关的部分.因此本文提出的基于添加优化控制器的方法也可以在线性时间内完成.

4.2　控制器的生成

在上一节中,我们使用状态可达图来检测客户服务 WSN_1、航班预定服务 WSN_2 和 TPC 服务 WSN_3 之间的部分兼容性,也就是控制器的存在性.如果 CRRG 是良构的,那么接下来需要产生相应的控制器.这一节讨论最优控制器的生成算法,如图 5 所示.

根据这一节提出的方法,可以构建客户服务 WSN_1、航班预定服务 WSN_2 和 TPC 服务 WSN_3 之间的控制器 C,如图 6 所示.从而得到相应的控制库所 $P_C = \{p_{c1}, p_{c2}, p_{c3}, p_{c4}\}$,其标记为浅色圆;控制变迁 $T_C = \{t_8, t_9, t_{11}, t_{12}\}$,其标记为黑色矩形.控制库所和控制变迁用控制弧来连接,用虚线表示,记为 $B = \{(t_2, p_{c1}), (p_{c1}, t_8), (t_3, p_{c2}), (p_{c2}, t_9), (t_{15}, p_{c3}), (p_{c3}, t_{11}), (t_{16}, p_{c4}), (p_{c4}, t_{12})\}$.这样,$WSN_1 \otimes_{A_1} WSN_2 \otimes_{A_2} WSN_3 \otimes_{P_C} C$ 的可达图是良构的,也就是说客户端服务 WSN_1、航班预定服务 WSN_2 和 TPC 服务 WSN_3 关于控制器 C 是可用的.

算法 2　最优控制器的构造方法

输入:流程服务网 $WSN_i, i = 1,2,\cdots,n$;最大允许反馈控制策略 U

输出:符合 U 的最优控制器 $C = (P_C, T_C, B, M_{c0})$

步骤 1　如果流程服务网 $WSN_i, i = 1,2,\cdots,n$,组合后发生死锁,即存在禁止状态 M_f,那么 $P_C = \{p_{cr} \mid r = 1,2,\cdots,v\} \in C$, $T_C = \{t_{cs} \mid s = 1,2,\cdots,w\} = B = \varnothing$,标记 $M_{c0} \in V^C$,并且使得 $M_{c0} \rightarrow Q$,其中 Q 表示队列;

步骤 2　如果 $Q \neq \varnothing$,则继续下面的步骤,否则构造过程结束转向步骤 6;

步骤 3　如果存在一个邻接于队列头 Q_h,而未被标记的节点 $M_c \in V^C$,则继续下面的步骤,否则转向步骤 5;

步骤 4　定义 $\Delta M_c = M_c - Q_h$,有 $Q_h \xrightarrow{t_{cs}} M_c$,则继续下面的步骤;

(1)存在 $\forall p_{cr} \in P_C$,定义 $k = \Delta(M_c(p_{cr})) > 0$,如果 $k > 0$

(a)在控制器 C 中加入控制变迁 $\{t_{c1}, t_{c2}, \cdots, t_{cw}\}$,即 $T_C = T_C \cup \{t_{c1}, t_{c2}, \cdots, t_{cw}\}$;加入控制库所 $\{p_{c1}, p_{c2}, \cdots, p_{cv}\}$,即 $P_C = P_C \cup \{p_{c1}, p_{c2}, \cdots, p_{cv}\}$;

(b)则 $B = B \cup \{(t_{cs}, p_{cr})\} = B \cup \{(t_{c1}, p_{c1}), (t_{c2}, p_{c2}), \cdots, (t_{cw}, p_{cv})\}$;

(2)存在 $\forall p_{cr} \in P_C$,定义 $k = \Delta(M_c(p_{cr})) < 0$,如果 $k < 0$

(a)在控制器 C 中加入控制库所 $\{p_{c1}, p_{c2}, \cdots, p_{cv}\}$,即 $P_C = P_C \cup \{p_{c1}, p_{c2}, \cdots, p_{cv}\}$;加入控制变迁 $\{t_{c1}, t_{c2}, \cdots, t_{cw}\}$,即 $T_C = T_C \cup \{t_{c1}, t_{c2}, \cdots, t_{cw}\}$;

(b)则 $B = B \cup \{(p_{cr}, t_{cs})\} = B \cup \{(p_{c1}, t_{c1}), (p_{c2}, t_{c2}), \cdots, (p_{cv}, t_{cw})\}$;

(3)标记 M_c 且 $M_c \rightarrow Q$,则转向步骤 3;

步骤 5　退出 Q,则转到步骤 2;

步骤 6　根据算法 1 中的最大允许反馈控制策略 U 的计算步骤,生成最优的控制器 C;

步骤 7　返回 $C = (P_C, T_C, B, M_{c0})$.

图 5　最优控制器的构造方法

对于禁止状态标识集 $M_F = \{m_6, m_7, m_{18}, m_{19}\}$ 的最大允许反馈控制策略,记为 $U = \{u_i, i = 1,2,3,4,5\}$,使得

$u_1: u_1(p_{c1}) = 1, u_1(p_{c2}) = 0, u_1(p_{c3}) = 0, u_1(p_{c4}) = 0$

$u_2: u_2(p_{c1}) = 1, u_2(p_{c2}) = 0, u_2(p_{c3}) = 1, u_2(p_{c4}) = 0$

$u_3: u_3(p_{c1}) = 1, u_3(p_{c2}) = 0, u_3(p_{c3}) = 0, u_3(p_{c4}) = 1$

$u_4: u_4(p_{c1}) = 0, u_4(p_{c2}) = 1, u_4(p_{c3}) = 0, u_4(p_{c4}) = 0$

$u_5: u_5(p_{c1}) = 0, u_5(p_{c2}) = 0, u_5(p_{c3}) = 0, u_5(p_{c4}) = 0.$

通过最大允许反馈控制策略 U,构建一个最优的控制器 $C = \{p_{c1}, p_{c2}, p_{c3}, p_{c4}, T_C, B\}$,并获得可用的控制服务组合网.按照 BPEL 语义,将这 4 条额外的控制通道转换回 BPEL 代码,获得新的 BPEL 过程.按照定理 1,

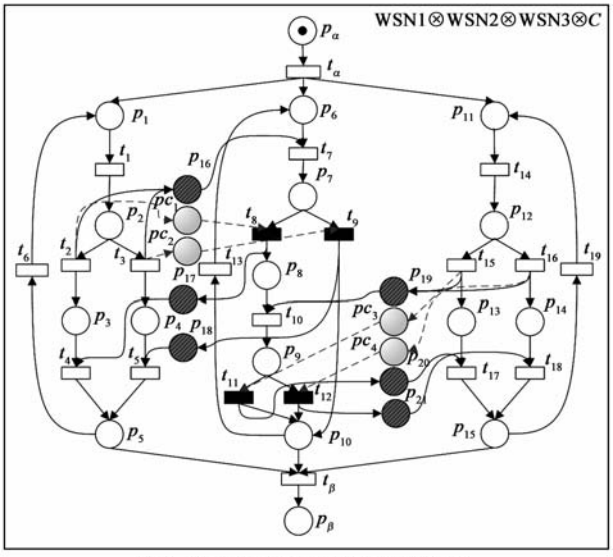

图6 基于控制器的可用服务组合网

新的 BPEL 过程是可用的.

5 相关工作

下面给出了现有的 Web 服务组合和可用性分析方面的研究,并与本文的工作进行比较.

文献[8]提出了一个死锁预防控制策略,即通过添加额外消息通道的方法来获得活的 Petri 网(siphons).随着新的网元素的添加,为了不出现死锁状态,部分变迁的发生将被停止.然而,结构分析技术的问题是为了获得一个活的(如:无死锁的)Petri 网模型,它将损坏系统部分好的状态.因此,系统的控制模型不是最大允许的.文献[13]提出了一种 BPEL 标注 Petri 网(BPN)并提出了一种基于通信图控制 BPN 网的决策算法.然而,只要新的网络服务组合中某一个网络服务有某一次错误的行为,整个网络服务组合就不可用.于是,如何在成千上万的备选网络服务中找到能和其他所有网络服务兼容的网络服务就成了一个难题.文献[14]使用 Martin 类型论(Martin Type Theory,MTT)建模 Web 服务行为,研究了 Web 服务行为一致性与相容性的判定方法.然而基于进程代数或 MTT 的方法并不能对服务的内部选择操作进行精细的刻画,也无法建模和分析 Web 服务的结构性质(如:良构性等).文献[15]采用基于离散控制的方法给 BPEL 工作流来避免其死锁的发生.然而,该方法对于一个大范围的 BPEL 工作流系统来说容易导致状态空间的爆炸.相比他们的研究,本文工作表达了服务组合环境下的死锁预防问题,结合 Petri 网技术提出了最优的控制器给不可用的服务组合网,并将其转换成为 BPEL 活动,同时采用约减状态空间的方法来避免状态空间爆炸问题的发生.

6 结论

Web 服务组合的行为兼容性分析是保证服务组合正确性和健壮性的重要方面.在面向服务环境下,当进行客户需求的 Web 服务组合时,即使服务的接口互相匹配,它们的动态交互行为也很可能出现不完全匹配的情况,使得组合的 Web 服务流程不可用.传统采用基于替换网络服务环境方法来解决 Web 服务流程的不可用性,其存在状态空间爆炸的问题.本文在约减状态空间的基础上,通过分析组合服务的行为,提出了添加最优控制器的方法来有效解决 Web 服务组合中行为的不可用性.本文提出的方法不仅降低了分析的复杂度,而且对于服务的自动化组合及其兼容性分析都具有重要意义.

参考文献

[1] OASIS. Web Services Business Process Execution Language Version 2.0[EB/OL]. http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html,2007-01-31.

[2] W3C. Web Services Description Language[EB/OL]. http://www.w3.org/TR/wsdl,2001-03-15.

[3] Heckel R, Mariani L. Automatic conformance testing of Web services[A]. Proceedings of the 8th Interna-tional Conference on Fundamental Approaches to Software Engineering[C]. Berlin, Heidelberg: Springer-Verlag, 2005. 34 – 48.

[4] Singh M P. Distributed enactment of multiagent workflows: temporal logic for web service composi-tion[A]. Proceedings of the 2th International Con-ference on Autonomous Agent and Multiagent Sys-tems[C]. New York: ACM Press, 2003. 907 – 914.

[5] 王勇,代桂平,侯亚荣,方娟,任兴田.基于并发事务逻辑的 Web 服务编制验证[J].电子学报,2009,37(10):2228 – 2233.
Wang Yong, Dai Gui-ping, Hou Ya-rong, Fang Juan, Ren Xing-tian. Verification of Web service orches-tration based on con-current transaction logic[J]. Acta Electronica Sinica, 2009, 37(10): 2228 – 2233. (in Chinese)

[6] Bultan T, Su J, Fu X. Analyzing conversations of Web services[J]. IEEE Internet Computing, 2006, 10(1): 18 – 25.

[7] 龚洪泉,赵文耘,徐如志,钱乐秋.基于 Pi 演算的构件演化研究[J].电子学报,2004,32(S1):242 – 246.
Gong Hong-quan, Zhao Wen-yun, Xu Ru-zhi, Qian Le-qiu. A research on Pi-calculus based component evolution[J]. Acta Electronica Sinica, 2004, 32(S1): 242 – 246. (in Chinese)

[8] Pengcheng Xiong, Yushun Fan, Mengchu Zhou. A Petri net approach to analysis and composition of web services[J]. IEEE Trans on System, Man and Cybernetics, Part A: Systems and Human, 2009, 40(2): 376 – 387.

[9] Van der Aalst W M P. The application of Petri nets to workflow management[J]. The Journal of Circuits Systems and

Computers,1998,8(1):21 – 66.
[10] Murata T. Petri nets: Properties, analysis and appli-cations[J]. Proceedings of the IEEE,1989,77(4):541 – 580.
[11] Ouyang C, Verbeek E, van der Aalst W M P, Breutel S, Dumas M, ter Hofstede AHM. Formal semantics and analysis of control flow in WS-BPEL[J]. Science of Computer Programming,2007,67(2 – 3):125 – 332.
[12] Valmari A. Stubborn sets for reduced state space generation [A]. Proceedings of the 10th International Conference on Application and Theory of Petri Nets[C]. Bonn, West Germany, 1989. 1 – 22.
[13] Martens A. Analyzing Web service based business processes [A]. Proceedings of the 8th International Conference on Fundamental Approaches to Soft-ware Engineering[C]. Berlin, Heidelberg: Springer-Verlag,2005. 19 – 33.
[14] 殷昱煜,李莹,邓水光,尹建伟. Web 服务行为一致性与相容性判定[J]. 电子学报,2009,37(3):433 – 438.
Yin Yu-yu, Li Ying, Deng Shui-guang, Yin Jian-wei. Determining on consistency and compatibility of Web services behavior[J]. Acta Electronica Sinica,2009,37(3):433 – 438. (in Chinese)
[15] Wang Y, Kelly T, Lafortune S. Discrete control for safe execution of IT automation workflows[A]. Proceedings of the 2nd ACM SIGOPS European Conference on Computer Systems [C]. Lisbon, Portugal: ACM,2007. 305 – 314.

作者简介

毕　敬　女,1979 年生于辽宁沈阳,东北大学信息科学与工程学院博士研究生,主要研究领域为服务计算,Petri 网应用,云计算.
E-mail: neubijing@gmail.com

朱志良　男,1962 年生于吉林桦甸,东北大学信息科学与工程学院,教授,博士生导师,主要研究领域为服务计算与信息集成,非线性系统,混沌分形与复杂性理论,计算机网络与通信安全技术,数字信号与数字图像处理.
E-mail: zzl@mail.neu.edu.cn

范玉顺　男,1962 年生于江苏扬州,清华大学自动化系,教授,博士生导师,主要研究领域为面向服务的企业体系架构与集成技术,企业信息化战略管理与规划,企业建模与业务优化分析,企业经营过程重组与工作流管理,系统集成与软件互操作技术,Petri 网建模与分析.
E-mail: fanyus@tsinghua.edu.cn