

## 基于 Petri 网的语义 Web 服务自动组合方法\*

汤宪飞<sup>1,2+</sup>, 蒋昌俊<sup>1,2</sup>, 丁志军<sup>1,2</sup>, 王成<sup>1,2</sup>

<sup>1</sup>(同济大学 计算机科学与技术系, 上海 201804)

<sup>2</sup>(国家高性能计算机工程技术研究中心 同济分中心, 上海 201804)

### A Petri Net-Based Semantic Web Service Automatic Composition Method

TANG Xian-Fei<sup>1,2+</sup>, JIANG Chang-Jun<sup>1,2</sup>, DING Zhi-Jun<sup>1,2</sup>, WANG Cheng<sup>1,2</sup>

<sup>1</sup>(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

<sup>2</sup>(Tongji Branch, National Engineering and Technology Center of High Performance Computer, Shanghai 201804, China)

+ Corresponding author: Phn: +86-21-69589864, E-mail: xianfei.tang@gmail.com

**Tang XF, Jiang CJ, Ding ZJ, Wang C. A Petri net-based semantic Web service automatic composition method. *Journal of Software*, 2007,18(12):2991-3000.** <http://www.jos.org.cn/1000-9825/18/2991.htm>

**Abstract:** Web service composition allows developers to create applications rapidly. But due to the tremendous growth in the number of Web services available, the Web service composition problem is still a challenging research issue. This paper introduces an automatic Web service composition method which considers both services' input/output type compatibility and behavioral constraint compatibility. The services available are translated into a set of Horn clause-like rules. User's input and output requirements are modeled as a set of facts and a goal statement in the Horn clauses respectively. Then Petri net is chosen to model the Horn clause set and T-invariant technique is used to determine the existence of composite services fulfilling the user's input/output requirements. Two algorithms are presented for obtaining the Petri net models of the composite Web services which satisfy not only the user's input/output requirements but also the user's behavioral constraints.

**Key words:** Web service; Web service composition; Horn clause; Petri net; T-invariant

**摘要:** Web 服务组合使得开发人员可以快速地创建自己的应用程序.但是,随着 Internet 上可用的 Web 服务数目的增加,Web 服务组合是一项高度复杂的任务.针对语义 Web 服务的自动组合问题,提出了一种既考虑服务输入/输出又考虑服务行为约束的自动组合方法.首先,注册服务被转化为一组 Horn 子句形规则,用户的输入和输出请求分别被转化为 Horn 子句中的事实和目标,从而将寻找满足用户输入/输出请求的合成服务问题转化为 Horn 子句的逻辑推理问题;然后,用 Petri 网来为该 Horn 子句集建模,T-不变量技术被用来判定是否存在满足用户输入/输出请求的合成服务;最后给出了两种算法来获取既满足用户输入/输出请求又满足用户行为约束的合成服务的 Petri 网模型.

**关键词:** Web 服务;Web 服务组合;Horn 子句;Petri 网;T-不变量

中图法分类号: TP393 文献标识码: A

\* Supported by the National Natural Science Foundation of China under Grant Nos.60534060, 60473094 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z136 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2003CB317002 (国家重点基础研究发展计划(973))

Received 2007-06-10; Accepted 2007-10-16

随着一系列开放、标准技术的提出,Web 服务以其松散耦合、语言中立、平台无关性和开放性有效地解决了平台异构性和互操作等问题,正在成为一个崭新的分布式计算模型和一种新兴的互联网应用模式.作为一种新的分布式计算模式,Web 服务技术允许异构软件和应用系统的互联互通,并使得组织可以通过 Internet 共享数据、软件和硬件资源<sup>[1]</sup>.异构软件和应用系统的功能可作为 Web 服务统一发布并在服务注册处注册.Internet 用户可以使用标准协议来发现并调用注册的服务.

但是,为了提高服务的可重用性、分散和简化应用逻辑,单个 Web 服务一般都不会做得太复杂<sup>[2]</sup>.每个服务所提供的功能一般都比较简单.要实现具体的复杂应用,就必须要实现多个服务的交互协作.可以说,Web 服务的真正潜力在于服务组合.如果注册的 Web 服务不能结合起来形成合成服务以满足用户更加复杂的要求,其用途就很有有限.此外,随着支持电子商务解决方案的 Web 服务的迅速增加,对于整合这些基于 Web 的自治、异构服务的需求进一步变大.

从软件复用的角度来说,当把 Web 服务视为可复用的软件构件实体时,Web 服务组合也可看作是一种在 Internet 上的基于构件的软件开发<sup>[3]</sup>.

由于 Internet 上有大量的 Web 服务可用,手工分析这些服务并生成合成服务规划已经超出了人的能力范围<sup>[4]</sup>.因此,实现 Web 服务的自动或者半自动组合是十分必要的.此外,语义 Web 服务通过引进语义来帮助消除服务发现和组合等过程中的二义性和模糊性,从而为服务自动发现、组合、执行等提供了良好的基础.目前,主要的语义 Web 服务描述语言主要有 OWL-S、WSMO(Web service modeling ontology)和 SAWSDL(semantic annotations for WSDL and XML schema)<sup>[5,6]</sup>等.

本文通过语义 Web 服务组合建模,将可用的 Web 服务用一组 Horn 子句形规则表示,将用户提供的输入表示为 Horn 子句中的一组事实,用户期待的输出表示为 Horn 子句中的一个目标,从而将寻找一个满足用户输入/输出要求的合成服务问题转化为一个 Horn 子句推理问题;然后,我们利用 Petri 网作为该 Horn 子句集的形式化模型,使用 T-不变量技术来确定是否存在一个满足用户输入/输出要求的合成服务;最后,我们给出了两种算法以获得一个满足用户诸项要求和约束(输入/输出要求、行为约束兼容要求以及服务质量要求)的合成服务的 Petri 网模型.本文形式上采用 SAWSDL 作为语义 Web 服务的外部规范,但是,该方法也可以与其他语义 Web 服务描述语言相结合.

## 1 基本概念

### 1.1 语义Web服务

我们首先介绍一些在语义 Web 服务组合中用到的相关概念.

**定义 1(原子服务).** 一个原子 Web 服务,即 SAWSDL 中的一个操作,可以用一个四元组  $WS=(I,O;BC,QoS)$  表示,其中:

- (1)  $I$  表示该服务的输入参数所引用的语义概念集合;
- (2)  $O$  表示该服务的输出参数所引用的语义概念集合;
- (3)  $BC$  表示该服务的行为约束集合,行为约束是服务提供者为保证服务正确执行而施加于服务上的一些条件和策略;
- (4)  $QoS$  表示该服务的服务质量参数集合, $QoS$  属性可以考虑价格、响应时间、可用性、可靠性等.

在 SAWSDL 中,Web 服务的输入/输出参数可以通过 *modelReference* 引用外部语义模型(如本体)中的概念.服务的行为约束也可以通过 *modelReference* 加入到 SAWSDL 中;有多种类型的行为约束,如服务的可用时间、服务的覆盖范围以及其他领域相关的约束.以一个投递服务为例,该服务可能只在 7:00~20:00 这个时间段内接受服务请求(服务可用时间约束),其投递区域可能局限于上海(服务覆盖范围约束),接受的包裹重量不大于 50kg(领域相关的约束)等等.在 SAWSDL 中,这些约束是用 SWRL(semantic web rule language)来描述的.目前,SAWSDL 尚没有提供对于服务的 QoS 描述的支持,但用户服务提供者可以利用其他协议如 SLA(service level agreement)对服务的 QoS 进行描述.

**定义 2(用户请求).** 一个用户请求可以用一个四元组  $WS=(I_{prov}, O_{req}; BC_{users}, QoS_{user})$  来表示,其中:

- (1)  $I_{prov}$  表示服务请求者提供的输入所引用的语义概念集合;
- (2)  $O_{req}$  表示服务请求者渴望得到的输出所引用的语义概念集合;
- (3)  $BC_{user}$  表示服务请求者定义的行为约束集合;
- (4)  $QoS_{user}$  表示服务请求者定义的服务质量参数标准.

**定义 3(参数类型兼容).** 设  $C_i$  和  $C_j$  分别是参数  $Par_i$  和  $Par_j$  所引用的语义概念,如果  $C_i$  和  $C_j$  是等价概念(在 OWL 中由 *owl:equivalentClass* 或 *owl:equivalentProperty* 定义),或者  $C_i$  是  $C_j$  的子概念(在 OWL 中由 *rdfs:subClassof* 或 *rdfs:subPropertyof* 定义),那么参数  $Par_i$  和  $Par_j$  类型兼容,参数  $Par_i$  的值可以安全地传递给参数  $Par_j$ .

**定义 4(行为约束兼容).** 两个 Web 服务  $WS_i$  和  $WS_j$  是行为约束兼容的,如果对于这两个服务所有的任何一个同种类型的约束  $bc_k$ ,其值的交集不为空.

例如,有两个服务  $WS_1$  和  $WS_2$ .  $WS_1$  的可用时间是 7:00~20:00,服务覆盖区域是上海;  $WS_2$  的可用时间是 8:00~21:00,服务覆盖区域是北京.那么,这两个服务的可用时间约束是兼容的,但其覆盖区域约束不兼容,因此,其行为约束是不兼容的.行为约束不兼容的服务组合在一起时可能会导致结果的不正确.

**定义 5(服务组合问题).** 一个服务组合问题就是从可用的服务集合中选出一组服务,该组服务能够按照一定的构造方式形成一个新的可以执行的增值服务,此增值服务能够接受用户提供的输入值(或其子集),生成用户期望的输出,并且要考虑到构成该增值服务的各原子服务之间及其与用户请求规范之间的行为约束兼容,以及原子服务和增值服务的服务质量要达到用户指定的标准.

## 1.2 Petri网<sup>[7]</sup>和Horn子句

本文将基于 Horn 子句推理和 Petri 网来解决服务组合问题,因此,我们首先了解一下 Petri 网和 Horn 子句的相关概念.

**定义 6(Petri 网).** 五元式  $PN=(P,T;F,M_0,W)$  称作 Petri 网,当且仅当

- (1)  $P$  是库所的有限集合,  $T$  是变迁的有限集合,且  $P \cup T \neq \emptyset, P \cap T = \emptyset$ ;
- (2)  $F \subseteq (P \times T) \cup (T \times P)$  是弧的集合;
- (3)  $\text{dom}(F) \cup \text{cod}(F) = P \cup T$ ;
- (4)  $M: P \rightarrow \{0, 1, 2, \dots\}$  是标识函数.  $M_0$  是初始标识;
- (5)  $W: F \rightarrow \{1\}$  是每条弧所关联的权函数.

$\forall x \in P \cup T$ , 称  $x^* = \{y | (y \in P \cup T) \wedge ((x, y) \in F)\}$  为  $x$  的前置集,  $x^* = \{y | (y \in P \cup T) \wedge ((x, y) \in F)\}$  为  $x$  的后置集.

**定义 7(关联矩阵).** 一个 Petri 网的结构可以用一个关联矩阵  $C=[c_{ij}]_{n \times m}$  表示,其中,  $m=|P|, n=|T|$ , 并且

$$c_{ij} = W(t_i, p_j) - W(p_j, t_i).$$

**定义 8(T-不变量、支柱、极小支柱 T-不变量).** 设 Petri 网  $PN$  的关联矩阵为  $C=[c_{ij}]_{n \times m}$ , 称  $n$  维非零非负整数向量  $X$  是  $PN$  的 T-不变量, 当且仅当  $C^T X = 0$ . 令  $\|X\| = \{t_i | t_i \in T \text{ 且 } X(i) \neq 0\}$ , 则称  $\|X\|$  是  $PN$  的 T-不变量  $X$  的支柱. 称  $X$  是 Petri 网  $PN$  的极小支柱 T-不变量, 当且仅当不存在 T-不变量  $X', X' \neq X$ , 使得  $\|X'\| \subset \|X\|$ .

**定义 9(Horn 子句).** Horn 子句是最多有 1 个肯定文字的子句. 一个命题逻辑的 Horn 子句常具有如下形式:

$$P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q,$$

其中,  $P_i$  和  $Q$  是命题,  $\wedge$  和  $\rightarrow$  是逻辑连接符, 分别表示合取和蕴含. 该式表明如果条件  $P_1 \sim P_n$  都得到满足, 则可得出结论  $Q$ . 根据前件(前提)和后件(结论)是否为空, Horn 子句可以分为如下 3 种类型:

- (1) 前提和结论均不为空, 形如  $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$ , 这类子句常被称为一个“规则”.
- (2) 前提为空, 形如  $\rightarrow Q$ , 这类子句常被称为一个“事实”.
- (3) 结论为空, 形如  $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow$ , 这类子句常被称为一个“目标”.

## 2 语义 Web 服务自动组合方法

### 2.1 语义Web服务组合建模

通过如下规则,我们将语义 Web 服务以及用户请求等表示为一组 Horn 子句形规则.

**规则 1.** 一个原子服务  $WS=(I,O;BC,QoS)$  表示为一组命题 Horn 子句和附属于每条子句的一组参数.其中,服务的输入/输出  $(I,O)$  表示为一条命题规则:

$$I_1 \wedge I_2 \wedge \dots \wedge I_m \rightarrow O_1 \wedge O_2 \wedge \dots \wedge O_n, m=|I|, n=|O|.$$

该命题规则可以标准化为一组命题 Horn 子句形规则:

$$I_1 \wedge I_2 \wedge \dots \wedge I_m \rightarrow O_i, m=|I|, 1 \leq i \leq |O|.$$

Web 服务的名字、行为约束和服务质量  $(ServiceName, BC, QoS)$  作为一组约束性参数附属于每条规则.我们称该种类型的规则为可用服务规则.

**规则 2.** Web 服务的参数类型兼容可用 Horn 子句表示.设  $C_i$  和  $C_j$  是被 Web 服务(或用户请求)的输入/输出参数引用的两个语义概念(语义数据类型),如果  $C_i$  和  $C_j$  是等价的,则表示为 Horn 子句规则  $C_i \rightarrow C_j$  和  $C_j \rightarrow C_i$ ;如果  $C_i$  是  $C_j$  的子概念,则表示为 Horn 子句规则  $C_i \rightarrow C_j$ .

我们称此种类型的 Horn 子句规则为参数可组合性规则.

**规则 3.** 用户的输入/输出请求  $(I_{prov}, O_{req})$  可表示为一组 Horn 子句.其中,  $I_{prov}$  用 Horn 子句中的一组事实子句表示,即用户提供的每个输入参数所引用的语义数据类型都用一条事实型子句表示(代表了用户已知的事实):

$$\rightarrow I_{prov}^j, 1 \leq j \leq |I_{prov}|.$$

用户期待的所有输出参数所应引用的语义数据类型  $O_{req}$  用 Horn 子句中的一个目标子句表示(代表了用户想要证明的目标):

$$O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k \rightarrow, k=|O_{req}|.$$

我们称该种类型的 Horn 子句规则为用户请求规则.

根据可满足性原理:若  $S$  是一组一阶公式,  $G$  是一个一阶公式,  $G$  是  $S$  的逻辑推论当且仅当  $S \cup \{\neg G\}$  是不满足的.因此,要证明  $O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k$  为真,我们可以将  $O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k \rightarrow$ , 即  $\neg(O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k)$  加入到由 Web 服务、Web 服务的参数类型兼容以及用户输入所转化的 Horn 子句集中,证明该 Horn 子句集包含有矛盾(即不满足).因为该 Horn 子句集不满足就意味着  $O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k \rightarrow$  不成立,即  $\neg(O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k)$  不成立,从而得出  $O_{req}^1 \wedge O_{req}^2 \wedge \dots \wedge O_{req}^k$  成立,即至少存在着 1 个合成服务,该合成服务可以接受用户提供的输入,生成用户要求的输出.因此,通过上面 3 条规则,寻找一个满足用户输入/输出的合成服务的问题可以转化为 Horn 子句的逻辑推理问题.

在以上 3 种类型的 Horn 子句形规则中,可用服务规则和参数可组合性规则可以通过扫描服务注册中心以及本体推理得到,可以一次建立并存储在规则库中以供反复使用;用户请求规则则是在用户有服务请求到达的时候建立,可以临时存储在工作区中.

### 2.2 自动组合方法

本文采用 Petri 网作为 Horn 子句推理的形式化模型.Petri 网是描述具有异步、并发和不确定特征的系统的有力模型.此外,在 Petri 网理论方面已经建立了很多分析技术.因此, Petri 网常被用来作为逻辑编程、工作流和 Web 服务等一些涉及不确定、并发等活动形式化模型.图 1 给出了 3 种类型的 Horn 子句的 Petri 网模型,其中,库所表示前提/结论,变迁表示规则.

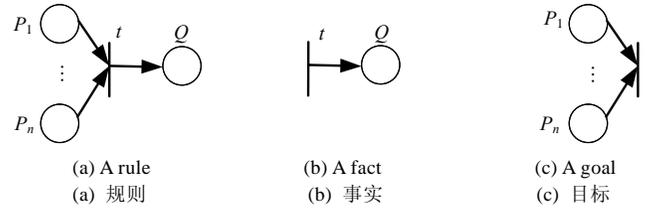


Fig.1 The Petri net representation of the Horn clauses

图 1 Horn 子句的 Petri 网模型

给定一个由  $n$  条子句和  $m$  个文字构成的 Horn 子句集,通过如下转换程序<sup>[8]</sup>可以得到该 Horn 子句集对应的 Petri 网模型的关联矩阵  $C=[c_{ij}]_{n \times m}$ :

- (1) 每一个子句对应着关联矩阵的一行.若用  $t_1, \dots, t_n$  表示  $n$  条子句,则子句  $t_i$  对应着  $C$  的第  $i$  行;
- (2) 每一个文字对应着关联矩阵的一列.若用  $p_1, \dots, p_m$  表示  $m$  个文字,文字  $p_j$  代表  $C$  的第  $j$  列;
- (3) 若文字  $p_j$  在子句  $t_i$  的前提中出现,则元素  $c_{ij}=1$ ;若文字  $p_j$  在子句  $t_i$  的结论中出现,则元素  $c_{ij}=-1$ ;否则,  $c_{ij}=0$ .

在本文针对服务组合问题所得到的 Horn 子句集中,文字代表语义概念(语义数据类型);因为有 3 种类型的规则,所以子句或代表(分解后的)Web 服务(有多于 1 个输出的 Web 服务被分解为多个单输出的服务),或代表参数类型的兼容性,或代表用户请求(提供的输入以及期待的输出).

因此,对于本文中得到的 Petri 网的关联矩阵,就其物理含义而言,矩阵的列表示语义概念.如果行表示(经过分解后的)Web 服务,则矩阵元素表示该语义概念是否在该服务中出现(-1 表示该元素所在列对应的语义概念在其所在行对应的服务的输入中出现,1 表示此语义概念在该服务的输出中出现,0 则表示此语义概念未在该服务中出现);如果行表示参数类型的兼容,则该行只有两列非零元素:-1 所在列对应的语义概念(语义数据类型)的值可以安全地传递给 1 所在列对应的语义概念;如果行表示用户请求,该行或者只有 1 个 1(代表了用户提供了该列的语义数据类型的输入),或者有若干个-1(每个-1 元素所在列的语义数据类型都是用户期待的输出类型).

针对基于 Petri 网的 Horn 子句的逻辑推理问题,Murata<sup>[9]</sup>和 Lautenbach<sup>[10]</sup>等人都曾研究过一组 Horn 子句包含有矛盾的充分和必要条件:

**定理 1<sup>[10]</sup>.** 若  $\alpha$  是一组 Horn 子句,并且  $N_\alpha=(S_\alpha, T_\alpha, F_\alpha)$  是其 Petri 网模型,则下列语句等价:

- (1)  $\alpha$  是矛盾的;
- (2)  $N_\alpha$  是 0-reproducing 的\*;
- (3)  $N_\alpha$  有一个非负 T-不变量  $X$ ,并且目标变迁(目标子句对应的变迁)  $t_g \in \|X\|$ ;
- (4) 在  $N_\alpha$  中存在一组从目标变迁(汇变迁)到事实变迁(源变迁)的逆向路径  $Y$ ,且对于  $Y$  中一条路径上的任何变迁  $t, t$  的邻接库所  $p \in {}^*t \cup t^*$  也是该路径上的节点.

因此,在得到 Horn 子句集的 Petri 网模型的关联矩阵后,我们就可以通过求解 Petri 网的 T-不变量来确定该 Horn 子句集是否是矛盾的.就本文而言,即确定是否存在功能上满足用户请求的合成服务.

关于 T-不变量的求解,Martine 等人给出了一种简单、快速的算法<sup>[11]</sup>.后来,有很多人在此算法的基础上进行优化改进.下面我们对本文中的关联矩阵的特征进行分析,以确定宜采用的 T-不变量求解算法.

一个原子 Web 服务一般都不会太复杂,其所引用的语义概念仅仅是所有出现在服务库中的语义概念的极小一部分.也就是说,矩阵的每一行中,大部分元素都是 0.因此,本文中 Petri 网的关联矩阵是一种大维数的 0-1 稀疏矩阵.针对具有这种特征的关联矩阵的 T-不变量求解,Borujij 等人<sup>[12]</sup>给出了一种分治求解的方法:首先将一个大的关联矩阵分解成若干小的子系统;然后针对每个子系统单独求解 T-不变量;最后汇聚这些局部 T-不变量

\* 所谓 0-reproducing,是指在  $N_\alpha$  的初始标识  $M_0=0$  的情况下,存在一个变迁引发序列  $\gamma$ ,满足  $0[\gamma > 0$ .

得到整个系统的 T-不变量.该方法宜于并行,对于求解 Web 服务组合这样大规模的问题非常合适.

在本文中,我们只对极小支柱 T-不变量感兴趣,任何 T-不变量都可以通过极小支柱 T-不变量的非负线性叠加得到.如果存在支柱中包含目标变迁的极小支柱 T-不变量,则说明存在满足用户输入/输出请求的合成服务,且此极小支柱 T-不变量对应着一个潜在的合成服务.

但是,这并不是说满足用户请求的合成服务存在.还要考虑用户限定的行为约束以及服务质量标准( $BC_{user}$ ,  $QoS_{user}$ ),只有满足用户输入/输出要求和用户限定的行为约束兼容并且满足用户的服务质量标准的合成服务才是真正满足用户请求的合成服务.另外,尽管 T-不变量及其支柱表明了要调用哪些服务才可以满足用户的请求,但它只是一个无序的变迁集合,必须确定变迁的引发次序,即 Web 服务的调用顺序,才算完成了服务组合.

下面我们给出了几种算法,根据求得的极小支柱 T-不变量来最终生成满足用户请求的合成服务.

#### 算法 1.

输入:服务组合对应的 Petri 网模型  $PN$  的支柱中包含有目标变迁  $t_g$  的 T-不变量集合  $XS=\{X_i|t_g\in\|X_i\|\}$ ;

输出:与用户给定的行为约束兼容的 T-不变量集合  $XS_c$ .

- (1)  $XS_c\leftarrow\emptyset$ ;
- (2) 如果  $XS=\emptyset$ ,则执行步骤(4),否则执行步骤(3);
- (3) 从  $XS$  中取一个元素  $X_i$ ,置  $XS\leftarrow XS-\{X_i\}$ ,判断是否存在一个变迁  $t_j\in\|X_i\|$ , $t_j$  对应着一条表示 Web 服务的规则  $r_j$ ,并且该服务的行为约束(附属于规则  $r_j$ )与用户指定的行为约束不兼容.若不存在这样一个变迁,则  $XS_c=XS_c\cup\{X_i\}$ .转步骤(2);
- (4) 返回  $XS_c$ .

通过算法 1,我们可以获得一个新的 T-不变量集合  $XS_c$ ,该集合中的每一个 T-不变量中的代表服务的那些元素所对应的服务与用户指定的行为约束兼容.如果  $XS_c$  是空集,则说明没有合成服务与用户指定的行为约束条件兼容,即没有满足用户请求的合成服务;否则,存在与用户的行为约束条件兼容的合成服务.如果用户没有指定明确的行为约束条件,那么,可能还需要考虑构成合成服务的各原子服务之间的行为约束兼容(在算法 1 的基础上稍作修改,判断参与服务组合的两个原子服务之间的行为约束是否兼容),以确保得到正确的合成服务.

下面的算法用来求出所有与用户定义的行为约束兼容的合成服务的 Petri 网.

#### 算法 2.

输入:服务组合对应的 Petri 网模型  $PN$  的关联矩阵  $C$ ,T-不变量  $X_i\in XS_c$ ;

输出: $X_i$  对应的合成服务的 Petri 网模型  $PN_{cs}^i$ .

- (1) 删除关联矩阵  $C$  中除  $X_i$  中不为 0 的变迁所对应的行以外的其他行,得到一个新的矩阵,记为  $C'$ ;
- (2) 删除矩阵  $C'$  中元素全是 0 的那些列,得到矩阵  $C''$ ;
- (3) 根据矩阵  $C''$ ,我们可以得到一个 Petri 网模型  $PN''$ ,并用每个变迁对应的规则的部分附属参数  $\langle ServiceName, QoS \rangle$  标记该变迁;
- (4) 合并网  $PN''$  中那些标记一样的变迁(表示这些变迁代表同一个 Web 服务),即将  $PN''$  中所有标记一样的变迁用一个新变迁表示.该新变迁的前集和后集分别是所有原变迁的前集和后集的并集,新变迁的标记不变.将新的 Petri 网模型标记为  $PN_{cs}^i$ .

我们可以求出所有潜在合成服务的 Petri 网模型.这里,我们仍然说是潜在的合成服务,是因为虽然它们满足用户的输入/输出请求,并且和用户指定的行为约束兼容,但是还有待于正确性验证(如是否存在死锁)和 QoS 性能评估.

在这些潜在的合成服务对应的 Petri 网中,源变迁代表了用户的输入动作,汇变迁代表了产生用户期待的输出的动作、内部变迁或表示服务的调用,或表示数据的传递,库所则代表了服务组合所涉及的语义概念.利用 Petri 网的可达图/可覆盖树、虹吸、陷阱等分析方法,可以分析该合成服务中是否有死锁存在等性质.

针对验证正确的合成服务,我们可以求出变迁的引发序列,类似于  $((t_1\|t_2)\cdot t_3)$ ,表示可并行引发  $t_1$  和  $t_2$ ,然后顺序引发  $t_3$ ,该变迁引发序列对应着一个服务的调用序列.根据该序列以及每个服务的 QoS 参数,可以计算合成服

务的 QoS 值,从而可以选出一个满足用户 QoS 要求并且性能最优的合成服务.合成服务的 QoS 计算公式可以参考文献[13],这里我们不再赘述.

### 2.3 示 例

下面给出一个简单的例子来具体说明 Web 服务自动组合方法.

假定在下面的情景中有 8 个可用的服务和一个用户请求(见表 1).每个 Web 服务接受输入,产生一定的输出,并有一些行为约束(QoS 值在该例中没有给出).服务请求者的请求也由 3 部分组成(不考虑 QoS):用户提供的输入、渴望得到的输出以及合成服务应满足的行为约束.可以看出,没有任何单一可用的服务可以满足该请求,要想满足用户的请求,我们只有进行服务组合.

为方便计,这里我们假设所有的服务提供者和服务请求者使用同一个共享的本体,但在实际应用中,由于存在词汇的差异,即使同一个领域,服务提供者和请求者也可能使用不同的本体.在这种情况下,可以创建一个映射本体来捕获不同的本体所使用的概念之间的关系.

**Table 1** An illustrating example

表 1 示例

Service name (shortened form)	Input parameter (semantic concept)	Output parameter (semantic concept)	Constraint
Available Web services	SHPhoneLookup (SHP) name (PersonName) address (Address)	phone (PhoneNumber)	Area: Shanghai
	BJPhoneLookup (BJP) pName (PersonName) pAddress (Address)	phoneNum (PhoneNumber)	Area: Beijing
	City2ZipCode (C2Z) city (CityName)	zipCode (ZipCode)	Area: China
	CityState2ZipCode (CS2Z) city (CityName) state(StateName)	zipCode (ZipCode)	Area: USA
	OnLineBookShop (OLBS) recipients (PersonName) city (CityName) phone (PhoneNumber) zipCode (ZipCode) book (Book)	order (ProductOrder) packet (ParcelPost) amount (Amount)	Area: China
	VisaCard (VC) creditCard (CreditCard) amount (Amount)	approved (CardAuthorization)	CardType: Visa
	MasterCard (MC) creditCard (CreditCard) amount (Amount)	approved (CardAuthorization)	CardType: Master
	Shipping (Shipping) orderInfo (ProductOrder) parcel (ParcelPost) ccApproved(CardAuthorization)	shipped (ShippedConfirmation)	Area: China
User's query	consignee (PersonName) city (CityName) address (Address) book (Book) creditCard (CreditCard)	BookShipped (ShippedConfirmation)	Area: Shanghai CardType: Visa

通过上面的建模方法,我们建立一个包括 16 条 Horn 子句的子句集:

- $r_1: \text{PersonName} \wedge \text{Address} \rightarrow \text{PhoneNumber} | (\text{SHP}, \{ \text{Area}(\text{Shanghai}) \});$
- $r_2: \text{PersonName} \wedge \text{Address} \rightarrow \text{PhoneNumber} | (\text{BJP}, \{ \text{Area}(\text{Beijing}) \});$
- $r_3: \text{CityName} \rightarrow \text{ZipCode} | (\text{C2Z}, \{ \text{Area}(\text{China}) \});$
- $r_4: \text{CityName} \wedge \text{StateName} \rightarrow \text{ZipCode} | (\text{CS2Z}, \{ \text{Area}(\text{USA}) \});$
- $r_5: \text{PersonName} \wedge \text{CityName} \wedge \text{PhoneNumber} \wedge \text{ZipCode} \wedge \text{Book} \rightarrow \text{ProductOrder} | (\text{OLBS}, \{ \text{Area}(\text{China}) \});$
- $r_6: \text{PersonName} \wedge \text{CityName} \wedge \text{PhoneNumber} \wedge \text{ZipCode} \wedge \text{Book} \rightarrow \text{ParcelPost} | (\text{OLBS}, \{ \text{Area}(\text{China}) \});$
- $r_7: \text{PersonName} \wedge \text{CityName} \wedge \text{PhoneNumber} \wedge \text{ZipCode} \wedge \text{Book} \rightarrow \text{Amount} | (\text{OLBS}, \{ \text{Area}(\text{China}) \});$
- $r_8: \text{CreditCard} \wedge \text{Amount} \rightarrow \text{CardAuthorization} | (\text{VC}, \{ \text{CardType}(\text{Visa}) \});$
- $r_9: \text{CreditCard} \wedge \text{Amount} \rightarrow \text{CardAuthorization} | (\text{MC}, \{ \text{CardType}(\text{Master}) \});$
- $r_{10}: \text{ProductOrder} \wedge \text{ParcelPost} \wedge \text{CardAuthorization} \rightarrow \text{ShippedConfirmation} | (\text{Shipping}, \{ \text{Area}(\text{China}) \});$

$r_{11}: \rightarrow \text{PersonName}; r_{12}: \rightarrow \text{CityName}; r_{13}: \rightarrow \text{Address}; r_{14}: \rightarrow \text{Book}; r_{15}: \rightarrow \text{CreditCard};$   
 $r_{16}: \text{ShippedConfirmation} \rightarrow$ .

其中,“ $\rightarrow$ ”左边的 Horn 子句规则描述了服务的输入/输出功能信息(即接受哪些输入,可以产生什么输出);“ $\rightarrow$ ”右边的部分描述了服务的一些非功能信息(如服务名、行为约束等),这些信息作为一组描述以及约束性参数附属于每条 Horn 子句规则.

利用上面提到的 Horn 子句集到 Petri 网的转换程序,我们可以得到该子句集的 Petri 网模型的关联矩阵.图 2 给出了上述子句集对应的 Petri 网模型(根据其关联矩阵画出).

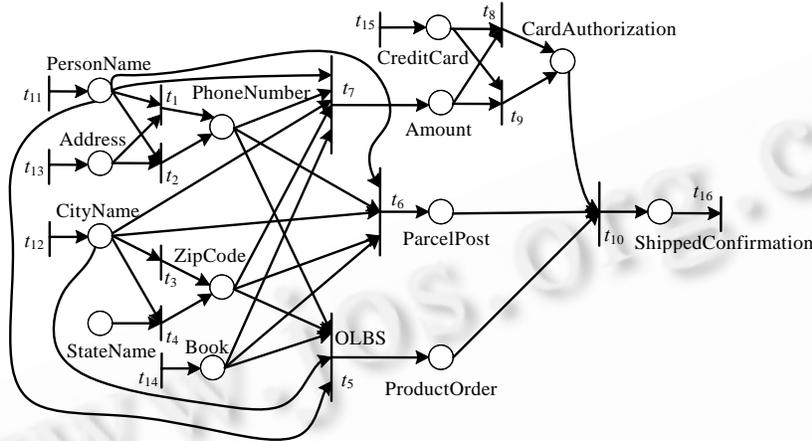


Fig.2 The Petri net representation of the Horn clause sets in this example

图 2 例中 Horn 子句集的 Petri 网模型

求解该 Petri 网模型的支柱中包含有目标变迁  $t_{16}$  的 T-不变量,有  $X_1=[3\ 0\ 3\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 6\ 6\ 3\ 3\ 1\ 1]$ ,  $X_2=[3\ 0\ 3\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 6\ 6\ 3\ 3\ 1\ 1]$ ,  $X_3=[0\ 3\ 3\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 6\ 6\ 3\ 3\ 1\ 1]$  和  $X_4=[0\ 3\ 3\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 6\ 6\ 3\ 3\ 1\ 1]$  共 4 个.

运行算法 1,可得到  $X_{S_c}=\{X_1\}$ ,这是因为  $X_2$  的支柱  $\|X_2\|$  和  $X_4$  的支柱  $\|X_4\|$  都包含有变迁  $t_9$ ,其所对应的 Web 服务 MasterCard 的约束条件和用户定义的 CardType 不兼容; $X_3$  的支柱  $\|X_3\|$  和  $X_4$  的支柱  $\|X_4\|$  都包含有变迁  $t_2$ ,其所对应的 Web 服务 BJPhoneLookup 的约束条件和用户定义的 Area 不兼容,则  $X_1$  对应着一个潜在的合成服务.

运行算法 2,我们可以得到  $X_1$  所对应的合成服务的 Petri 网模型,如图 3 所示.

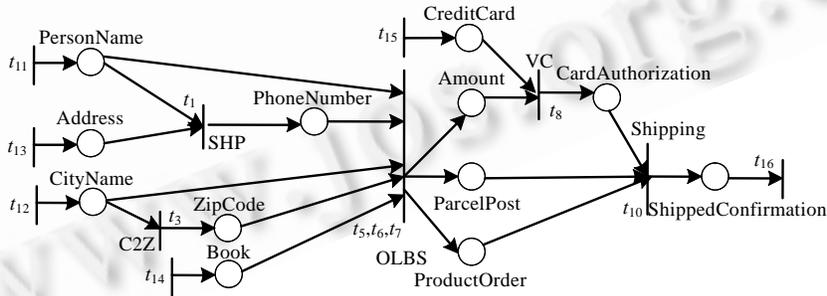


Fig.3 The Petri net representation of the composite Web service

图 3 合成 Web 服务的 Petri 网模型

根据已有的一些 Petri 网的分析方法(如虹吸、陷阱等),可以验证该合成服务中不存在死锁,这说明我们找到了一个正确的可以满足用户输入/输出功能要求和行为约束的一个合成服务.根据此模型可知,通过调用服务 SHPhoneLookup(变迁  $t_1$ )、City2Zipcode(变迁  $t_3$ )、OnLineBookShop(变迁  $t_5, t_6, t_7$ )、VisaCard(变迁  $t_8$ )和服务 Shipping(变迁  $t_{10}$ )就可以满足用户请求.进一步地,利用 Petri 网的可达图/可覆盖树的分析方法,可以得到服务调

用的代数规范((SHPhoneLookup||City2Zipcode)-OnLineBookShop·VisaCard·Shipping),其中,||表示并行执行,·表示顺序执行.根据该调用规范,结合每个服务的 QoS 值可以计算合成服务的 QoS 值<sup>[13]</sup>.

### 3 相关工作

文献[14]使用一组情景演算公理来定义 DAML-S 的语义,使用 Petri 网来提供情景演算公式的操作语义,合成服务的存在是通过 Petri 网的可达性分析来确定的.但是,Petri 网的可达性分析常常遭遇状态空间爆炸问题,特别是对于一个规模较大的网络.

文献[15]将用 DAML-S 描述的 Web 服务转化为一组线性逻辑公理,将用户的查询请求转化为一个线性逻辑公式,使用线性逻辑定理证明的方式来进行服务自动组合,其复杂性是 PSAPCE hard 的.该方法可以找到一个满足用户输入/输出要求的合成服务,但没有考虑用户以及服务的行为约束.

文献[16]通过分层任务网络规划来实现服务的自动组合,利用 SHOP2 规划器进行求解.

文献[1]给出了一种半自动的服务组合方法.它将 Web 服务组合问题形式化为一个 AND/OR 图中的搜索问题,给出了一种搜索算法以用来确定满足服务请求的合成服务.该文也仅仅考虑了输入/输出参数之间的依赖性,没有考虑到服务的行为约束条件.

文献[17]给出了一种基于领域本体的服务动态组合方法,该方法利用领域本体及其推理能力,生成一个优化的服务组合图,然后基于该图给出算法找出满足要求的路径,即一个满足服务请求的服务组合.但是,该文只考虑了单个服务之间的依赖性关系,没有考虑多个服务的输出联合起来为一个服务提供输入的情况.

文献[3,13]主要关注 Web 服务的选择优化问题.

### 4 结论

本文将 Web 服务自动组合问题转化为 Horn 子句的逻辑推理问题,然后使用 Petri 网作为 Horn 子句逻辑推理的形式化模型,使用 Petri 网的 T-不变量技术来确定是否存在满足用户输入/输出请求的合成服务,然后考虑到用户给定的合成服务的行为约束,给出了两种算法用来生成满足用户输入/输出请求并且行为约束和用户所指定的兼容的合成服务的 Petri 网模型.基于此模型,利用 Petri 网的分析技术可以验证合成服务的正确性,以及生成验证正确的合成服务的调用序列,根据此序列可以进行合成服务的 QoS 评估,最终找到一个满足用户诸项要求的最优的合成服务.

Petri 网具有良好的操作语义,常被用来作为工作流、Web 服务等一些涉及不确定、并发等活动的形式化模型.T-不变量是 Petri 网论中比较有效的结构化分析方法,并且对于大规模网络有并行的优化算法可以用来求解.因此,与利用可达图等方法来做服务组合<sup>[14]</sup>相比,本方法还是比较有效的.

本文从操作级考虑服务合成,即将原子服务视作一个接受一定输入产生一定输出的黑盒(在 Petri 网模型中用变迁表示).虽然可以分析合成服务的死锁、可达等动态特性,但是,对于在运行过程中有动态交互行为(有同步/异步消息传输)的服务,不能简单地将它们用命题逻辑公式来表示.因此在未来的工作中,我们将考虑从 Petri 网语言的角度来考虑这类服务的组合.

### References:

- [1] Lang QHA, Su SYW. AND/OR graph and search algorithm for discovering composite Web services. *Int'l Journal of Web Services Research*, 2005,2(4):46-64.
- [2] Fan J, Kambhampati S. A snapshot of public Web services. *SIGMOD Record*, 2005,34(1):24-32.
- [3] Zhao JF, Xie B, Zhang L, Yang FQ. A Web services composition method supporting domain feature. *Chinese Journal of Computers*, 2005,28(4):731-738 (in Chinese with English abstract).
- [4] Rao JH, Su XM. A survey of automated Web service composition methods. In: Cardoso J, Sheth A, eds. *Proc. of the SWSWPC 2004*. LNCS 3387, Berlin, Heidelberg: Springer-Verlag, 2005. 43-54.
- [5] Farrell J, Lausen H. Semantic annotations for WSDL and XML schema. 2007. <http://www.w3.org/TR/sawSDL/>

- [6] Akkiraju R, Sapkota B. Semantic annotations for WSDL and XML schema—Usage guide. 2007. <http://www.w3.org/TR/sawSDL-guide/>
- [7] Murata T. Petri Nets: Properties, analysis and applications. Proc. of the IEEE, 1989,77(4):541–580.
- [8] Murata T, Zhang D. A predicate-transition net model for parallel interpretation of logic programs. IEEE Trans. on. Software Engineering, 1988,14(4):481–498.
- [9] Peterka G, Murata T. Proof procedure and answer extraction in Petri net model of logic programs. IEEE Trans. on. Software Engineering, 1989,15(2):209–217.
- [10] Lautenbach K. Logical reasoning and Petri nets. In: van der Aalst WMP, Best E, eds. Proc. of the ICATPN 2003. LNCS 2679, Berlin, Heidelberg: Springer-Verlag, 2003. 276–295.
- [11] Lin C, Chaudhury A, Whinston A, Marinescu DC. Logical inference of horn clauses in Petri net models. IEEE Trans. on Knowledge and Data Engineering, 1993,5(4):416–425.
- [12] Borujij A, Boutayeb M, Cecchin T. A decentralized approach for computing invariants in large scale and interconnected Petri nets. In: Proc of the 1997 IEEE Int'l Conf. on System Man and Cybernetics. Orlando: IEEE Computer Society, 1997. 1741–1746.
- [13] Zeng LZ, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. IEEE Trans. on Software Engineering, 2004,30(5):311–327.
- [14] Narayanan S, McIlraith S. Simulation, verification and automated composition of Web services. In: Proc. of the 11th Int'l World Wide Web Conf. (WWW 2002). New York: ACM Press, 2002. 77–88.
- [15] Rao J, Küngas P, Matskin M. Logic-Based Web services composition: From service description to process model. In: Zhang LJ, ed. Proc. of the IEEE Int'l Conf. on Web Services. IEEE Computer Society, 2004. 446–453.
- [16] Sirin E, Parsia B, Wu D, Hendler J, Nau D. HTN planning for Web service composition using SHOP2. Journal of Web Semantics, 2004,1(4):377–396.
- [17] Li M, Wang DZ, Du XY, Wang S. Dynamic composition of Web services based on domain ontology. Chinese Journal of Computers, 2005,28(4):644–650 (in Chinese with English abstract).

#### 附中文参考文献:

- [3] 赵俊峰,谢冰,张路,杨芙清.一种支持领域特性的 Web 服务组装方法.计算机学报,2005,28(4):731–738.
- [17] 李曼,王大治,杜小勇,王珊.基于领域本体的 Web 服务动态组合.计算机学报,2005,28(4):644–650.



汤宪飞(1983—),男,河南兰考人,博士生,主要研究领域为 Web 服务组合与验证,Petri 网理论及应用.



丁志军(1974—),男,讲师,主要研究领域为 Web 服务,语义 Web,Petri 网,并发处理.



蒋昌俊(1962—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网格计算,语义网格,Petri 网.



王成(1980—),男,硕士,主要研究领域为无线网络,Web 服务.