第 29 卷 第 7 期
2006 年 7 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 29 No. 7
July 2006

Web 服务组合的有色网模型及运算性质

郭玉彬[1)] 杜玉越[2)] 奚建清[1)]

[1)](华南理工大学计算机科学与工程学院 广州 510640)

[2)](山东科技大学信息科学与工程学院 青岛 266510)

摘 要 定义了一种基于有色 Petri 网的 Web 服务组合模型,这一模型能清晰表达服务组合中各组成部分之间的逻辑关系,并可模拟组合服务的执行,分析其动态性质.给出了服务、服务组合运算的定义及性质、服务组合的建立与执行算法.

关键词 Web 服务;服务组合;运算性质;CP-net

中图法分类号 TP311

A CP-Net Model and Operation Properties for Web Service Composition

GUO Yu-Bin[1)] DU Yu-Yue[2)] XI Jian-Qing[1)]

[1)](*College of Computer Science and Engineering, South China University of Technology, Guangzhou* 510640)

[2)](*College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao* 266510)

Abstract A CP-net Model for Web service composition is proposed. To each service, a CP-net model is constructed to describe the logical relation of components graphically. Furthermore dynamic behaviors of services can be simulated, analyzed by executing of the CP-net model. Operators to construct new complex services from known ones as blocks are defined formally. Some algebra and dynamic properties of this model are studied and proved. Algorithms to construct and execute a composite service are delivered also.

Keywords Web service; service composition; operation property; CP-net

1 引 言

目前,Web 服务技术正从基础设施构建与概念推广阶段向大规模商业应用阶段快速发展.通过 Web 服务组合来动态生成新的应用系统,以满足企业的动态需求,已成为 Web 服务技术不断向前发展的技术动力和研究热点[1].Web 服务组合[2]是将已存在的 Web 服务,按照一定的规则动态地发现,并组装成为一个增值的、更大粒度的服务或系统,以满足用户的复杂需求,并提高软件生产率的过程.

当前,Web 服务组合方法主要有三种:基于工作流的动态组合[3,4]、基于语义网络的方法[5,6]及基于 PI-演算[7]或 Petri 网[8,9]等工具的形式化方法.基于代数进程或 Petri 网等工具的形式化方法是 Web 服务组合的一种常用方法.如文献[8]提出了一个基于 Petri 网的服务组合模型,并通过对 10 种组合元模型的代数与 Petri 网形式化语义的分析,建立起了从代数结构模型向 Petri 网的关系映射,从而为服务组合过程中的属性验证以及一致性的检

收稿日期:2006-02-17;修改稿收到日期:2006-05-18.本课题得到山东省"泰山学者"建设工程专项经费、国家自然科学基金(60573018,60125205,90412013,60473094)、国家"九七三"重点基础研究发展规划项目基金(2003CB316902,2004CB318001-03)、山东省自然科学基金(2004ZX17)、广东省科技攻关计划项目基金(G03B2040770)和广东省自然科学基金(B6480598)资助.郭玉彬,女,1973 年生,博士研究生,讲师,研究方向为数据库与网络计算、Petri 网理论与应用.E-mail: Guoyubin@people.com.cn.杜玉越,男,1960 年生,博士,教授,研究领域为 CSCW、服务计算、形式化技术、Petri 网理论与应用.奚建清,男,1962 年生,博士,教授,博士生导师,研究领域为数据库与网络计算.

测和优化提供了一种形式化工具. 文献[9,10]基于一种高级 Petri 网——OOPN(Object Oriented Petri Net),给出了一种 Web 服务组合方法. 基于该方法,一个 Web 服务可以映射为一个 OOPN 系统. 不同的网系统可以通过消息传递机制整合到一起,形成组合服务及控制流.

本文利用有色网对这一问题进行分析建模,给出了基于 CP-net 的 Web 服务及服务运算定义,分析了运算的代数性质及运行性质,并给出了服务组合的建立与执行算法. 这一模型能够清晰地表达各组成服务之间的逻辑关系,并模拟组合服务的执行. 可利用可达图、不变量等有色网工具分析系统的动态性质.

本文第 2 节给出基于 CP-net 的 Web 服务组合模型,提出 Web 服务组合中基本运算的定义及组合服务的构造与执行算法,并分析组合服务运算与模型的性质;第 3 节给出一个实例及其性质分析;第 4 节总结全文,并指出下一步工作.

2 基于CP-net的Web服务组合模型

有色 Petri 网[11]结合了 Petri 网和高级程序语言的优点,可对复杂系统进行图形化的概念模拟,对系统行为既具有动态模拟能力,又可进行形式化推理. 本文首先利用 CP-net 对服务进行建模,然后分析模型性质,并给出服务组合执行算法. 限于幅篇,此处只给出有色网的概念,关于有色网的详细内容读者可参看文献[11].

2.1 基本概念

定义 1[11]. 有色网 CP-net 是一个八元组 $CPN=(\Sigma,P,T,F,C,G,E,I)$,其中

Σ 是类型(types)的非空有限集,也称颜色集;

P 是有限库所集;

T 是有限变迁集;

F 是有限弧集且 $F\subseteq P\times T\cup T\times P$;

C 是颜色函数,$C:P\rightarrow\Sigma$;

G 为哨函数,$G:T\rightarrow Boolexpression$; $\forall t\in T$: $Type(G(t))=Boolean\wedge type(var(G(t)))\subseteq\Sigma$;

E 为弧函数,$E:F\rightarrow Boolexpression$; $\forall f\in F$: $Type(E(f))=C(p)_{M_S}\wedge type(var(E(f)))\subseteq\Sigma$;其中 p 为 f 连接的库所;

I 为初始化函数. $I:P\rightarrow\Sigma$ 为每一个库所赋颜色值生成初始标识 M_S;即

$$\forall p\in P:Type(I(p))=C(p)_{M_S}.$$

从组成框架及实现目标的角度讲[12],Web 服务是一种网络操作,它能够利用标准的 Web 协议及接口进行应用间的交互. 因此,本文把服务定义为一个带输入输出的 CP-net,即通过输入输出库所描述其接口信息,通过变迁描述其内部逻辑. 在给出服务定义前先定义颜色集.

定义 2. Σ(颜色集)定义为

颜色 $ARGU=string$;

颜色 $RES=\{success,failed\}$;

颜色 $OUTPUT=record\ re:RES*outp:ARGU$;

颜色 $CONDS=Boolexpression$;

颜色 $BOOLVAL=\{\text{true},\text{false}\}$.

一般颜色 $ARGU$ 表示服务输入,简化为字符串. 颜色 $OUTPUT$ 表示服务,每个输出由两部分构成,re 表示服务是否正常执行,$outp$ 部分则是输出结果,用字符串表示. $CONDS$ 是布尔表达式类型,用来表示条件. $BOOLVAL$ 则表示逻辑常量 true 和 false. 如未作特别说明,本文规定以下服务的颜色集都取 Σ.

定义 3. 服务 S 是一个 CP-net $S=(\Sigma,P,T,F,C,G,E,In,Out,I)$(见图 1),其中

变量:$x:ARGU$; $o:OUTPUT$;

库所集 $P=\{In,Out\}$;

变迁集 $T=\{service\}$;

弧集 $F=\{\langle In,service\rangle,\langle service,Out\rangle\}$;

弧函数集 $E=\{E(\langle In,service\rangle)=x,$
$E(\langle service,Out\rangle)=o\}$;

颜色函数 $C=\{C(In)=ARGU,$
$C(Out)=OUTPUT\}$;

$Guard$ 函数依据具体服务的语义来确定,针对不同输入决定服务调用能否执行.

In 和 Out 分别是服务 S 的输入、输出,即变迁 $service$ 的输入和输出库所. $In,Out\in P$ 且 $\neg\exists t\in T:\langle t,In\rangle\in F$, $\neg\exists t\in T:\langle Out,t\rangle\in F$,即库所 In 在 S 中没有前集元素,库所 Out 没有后集元素;

初始化函数 $I:I(In)=1'x$; $\forall p\in P:p\neq In\rightarrow I(p)=\text{NULL}$ 用以确定网系统的初始条件.

为描述方便,可将服务划分成原子服务和组合服务两类.

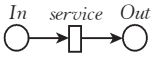

图 1 服务的 CP-net 模型

定义 4. 不依赖其它服务,且可单独被调用的服务称为原子服务;通过调用其它服务,完成自身提

供功能的服务称为组合服务.

由定义 4 知，原子服务是服务组合的基础，如图 1，它只包含一个服务变迁，且 $P=\{In,Out\}$.

组合服务则是由原子服务或其它组合服务，按一定规则组合而成的服务. 其 CPN 模型相应的是由多个变迁、库所按流关系组合而成的. 如图 2 可看作一个由服务 S_1，S_2 组合而成的组合服务的 CPN 模型. 但当一个组合服务 S_1 作为组成部分出现其它组合服务 S_2 中时，可用图 1 表示它. 依据有色网规则，服务变迁（图 1 中的 *service*）是一个替代变迁（substitution transition[11]）. 可用专门页描述替代变迁，即服务 S_1 的结构. 此时组合服务 S_2 是由自身 CPN、各组成部分 CPN 共同组成的一个层次有色网. 但文献[11]同时指出，层次有色网与一般有色网是等价的. 因此，本文仅在有色网上讨论服务组合问题.

现对组合服务中的变迁进行分类，将用于表示被调用服务的变迁称服务变迁，如图 1 中的 *service*. 将用于把服务组织成组合服务的变迁称为控制变迁. 一般地，控制变迁在发起服务的结点执行，其执行后状态可通过相应变迁的引发计算得出. 而各服务变迁则要通过服务调用执行，其执行后状态由服务返回结果决定. 在图中，用灰色表示控制变迁，用白色表示服务变迁.

原子服务可构造成组合服务，组合服务又可作为服务构造成更复杂的服务，如此嵌套可构造出任意复杂的服务. 将多个服务组合为一个服务的组合方式称为服务的运算.

定义 5（服务运算）.　设 S_1，S_2 为两个服务，且

$S_1=(\Sigma,P_1,T_1,F_1,C_1,G_1,E_1,In_1,Out_1,I_1)$，

$S_2=(\Sigma,P_2,T_2,F_2,C_2,G_2,E_2,In_2,Out_2,I_2)$.

（1）顺序运算（见图 2）

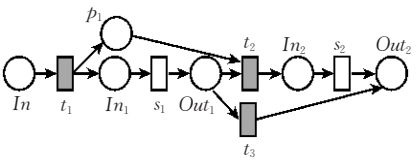

图 2　服务组合的顺序运算

$S=S_1\cdot S_2=(\Sigma,P,T,F,C,G,E,In,Out,I)$，

其中，$\cdot$ 为服务组合的顺序运算符；

变量定义：$x,x_1,x_2:ARGU$；$o:OUTPUT$；

$P=P_1\cup P_2$；$T=T_1\cup T_2\cup\{t_1,t_2,t_3\}$；

$F=F_1\cup F_2\cup\{\langle In,t_1\rangle,\langle t_1,p_1\rangle,\langle t_1,In_1\rangle,$
$\langle p_1,t_2\rangle,\langle t_2,In_2\rangle,\langle Out_1,t_2\rangle,$
$\langle Out_1,t_3\rangle,\langle t_3,Out_2\rangle\}$；

$C=C_1\cup C_2\cup\{C(In)=ARGU,$
$C(p_1)=ARGU\}$；

$G(s_1)$，$G(s_2)$ 依据服务的语义设定①.

$G(t_1)=\text{null}$，$G(t_2)=(o.re==success)$，

$G(t_3)=(o.re==failed)$；

$E=E_1\cup E_2\cup\{E(\langle In,t_1\rangle)=1'x,$
$E(\langle t_1,In_1\rangle)=1'x_1,$
$E(\langle t_2,In_2\rangle)=E(\langle p_1,t_2\rangle)=$
$E(\langle t_1,p_1\rangle)=1'x_2,$
$E(\langle Out_1,t_2\rangle)=E(\langle Out_1,t_3\rangle)=$
$E(\langle t_3,Out_2\rangle)=1'o\}$.

颜色集 Σ、初始函数 I 同定义 3.

此处给出的弧函数是一种示例，表示当服务 S_1 执行成功（S_1 输出 $o.re=success$）时，继续调用服务 S_2. 而当 S_1 不能执行成功（S_1 输出中 $o.re=failed$）时，不调用服务 S_2，直接返回. 也可依据现实语义设定相反逻辑.

（2）并发运算（见图 3）

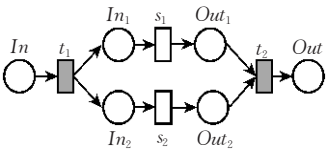

图 3　服务组合的并发操作

$S=S_1\parallel S_2=(\Sigma,P,T,F,C,G,E,In,Out,I)$，

其中，$\parallel$ 为服务组合的并发运算符；

变量：$x,x_1,x_2:ARGU$；$o,o_1,o_2:OUTPUT$；

$P=P_1\cup P_2\cup\{In,Out\}$；$T=T_1\cup T_2\cup\{t_1,t_2\}$；

$F=F_1\cup F_2\cup\{\langle In,t_1\rangle,\langle t_1,In_1\rangle,\langle t_1,In_2\rangle,$
$\langle t_2,Out\rangle,\langle Out_1,t_2\rangle,\langle Out_2,t_2\rangle\}$；

$C=C_1\cup C_2\cup\{C(In)=ARGU,$
$C(Out)=OUTPUT\}$；

$G=G_1\cup G_2\cup\{G(t_1)=G(t_2)=\text{null}\}$；

$E=E_1\cup E_2\cup\{E(\langle Out_1,t_2\rangle)=1'o_1,$
$E(\langle Out_2,t_2\rangle)=1'o_2,E(\langle t_2,Out\rangle)=1'o,$
$E(\langle In,t_1\rangle)=1'x,E(\langle t_1,In_1\rangle)=1'x_1,$
$E(\langle t_1,In_2\rangle)=1'x_2\}$.

颜色集 Σ、初始函数 I 同定义 3.

变迁 t_1 的功能是，依据输入生成服务 S_1，S_2 的初始条件. 变迁 t_2 的功能则是汇总 S_1，S_2 的输出生成总输出结果.

---

① 图 2 中服务变迁的 G 函数约束服务的执行，且应依据服务的语义设定. 控制变迁的 G 函数约束运算逻辑，由运算进行定义. 因此，定义中只给出控制变迁的 G 函数定义.

(3) **选择运算(如图 4)**

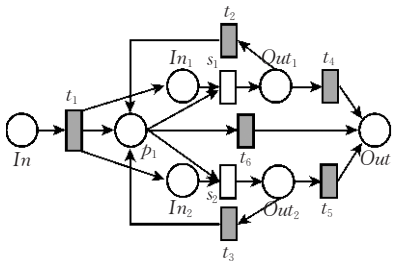

图 4 服务组合的选择操作

$S=S_1\oplus S_2=(\Sigma,P,T,F,C,G,E,In,Out,I)$,

其中,$\oplus$为服务组合的选择运算符;

变量定义:$x,x_1,x_2:ARGU$;$o,o_1,o_2:OUTPUT$;$con,con_1,con_2:CONDS$;

$P=P_1\cup P_2\cup\{In,Out,p_1\}$;

$T=T_1\cup T_2\cup\{t_1,t_2,t_3,t_4,t_5,t_6\}$;

$F=F_1\cup F_2\cup\{\langle In,t_1\rangle,\langle t_1,In_1\rangle,\langle t_1,In_2\rangle,$
$\langle t_1,p_1\rangle,\langle t_2,p_1\rangle,\langle t_3,p_1\rangle,\langle Out_1,t_2\rangle,$
$\langle Out_1,t_4\rangle,\langle Ou_2,t_3\rangle,\langle Out_2,t_5\rangle,\langle p_1,S_1\rangle,$
$\langle p_1,S_2\rangle,\langle p_1,t_6\rangle,\langle t_4,Out\rangle,\langle t_5,Out\rangle,$
$\langle t_6,Out\rangle\}$;

$C=C_1\cup C_2\cup\{C(In)=ARGU,$
$C(Out)=OUTPUT,C(p_1)=CONDS\}$;

$G=G_1\cup G_2\cup\{G(t_1)=\text{null},$
$G(t_2)=(o.re==failed),$
$G(t_3)=(o.re==failed),$
$G(t_4)=(o.re==success),$
$G(t_5)=(o.re==success),$
$G(t_6)=(M(In_1)==\text{null})\wedge$
$(M(In_2)==\text{null})\}$;

$E=E_1\cup E_2\cup\{E(\langle In,t_1\rangle)=1'x,$
$E(\langle t_1,In_1\rangle)=1'x_1,E(\langle t_1,In_2\rangle)=1'x_2,$
$E(\langle t_3,p_1\rangle)=1'con_2,E(\langle t_1,p_1\rangle)=$
$E(\langle p_1,S_1\rangle)=E(\langle p_1,S_2\rangle)=$
$E(\langle p_1,t_6\rangle)=1'con,E(\langle Out_1,t_2\rangle)=$
$E(\langle Out_1,t_4\rangle)=E(\langle t_4,Out\rangle)=1'o_1,$
$E(\langle Out_2,t_5\rangle)=E(\langle t_5,Out\rangle)=$
$E(\langle Out_2,t_3\rangle)=1'o_2,E(\langle t_6,Out\rangle)=1'o,$
$E(\langle t_2,p_1\rangle)=1'con_1\}$;

颜色集 Σ、初始函数 I 同定义 3.

变迁 t_1 的功能是,依据输入生成服务 S_1,S_2 的引发条件和选择条件. 由 p_1 中选择条件确定执行哪一个服务. 若服务 S_1 执行,执行结果输出到库所 Out_1 中. 若输出为$\langle success,outp\rangle$,变迁 t_4 可引发,并将 S_1 执行结果输出到库所 Out. 若 S_1 输出为$\langle failed,outp\rangle$,变迁 t_2 引发将选择条件回送到 p_1,且重新选择另一个服务执行. 变迁 t_5 和 t_3 分别与 t_4 和 t_2 的功能对应. 当库所 In_1,In_2 中都没有 Token 时,表示两个选择都已执行且结果均为$\langle failed,outp\rangle$. 此时,变迁 t_6 可以引发,输出$\langle failed,outp\rangle$到库所 Out 中,表示整个服务执行失败.

(4) **迭代运算(见图 5)**

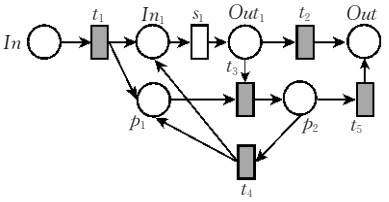

图 5 服务组合的迭代运算

$S=nS_1=(\Sigma,P,T,F,C,G,E,In,Out,I)$,

其中,nS_1 表示服务 S_1 重复执行 n 次;

变量定义:q,x,x_1,x_2:$ARGU$;o,o_1,o_2:$OUTPUT$;con:$CONDS$;bo:$BOOLVAL$;

$P=P_1\cup\{p_1,p_2,In,Out\}$;

$T=T_1\cup\{t_1,t_2,t_3,t_4,t_5\}$;

$F=F_1\cup\{\langle In,t_1\rangle,\langle t_1,In_1\rangle,\langle t_1,p_1\rangle,\langle t_4,p_1\rangle,$
$\langle p_1,t_3\rangle,\langle Out_1,t_3\rangle,\langle t_3,p_2\rangle,\langle p_2,t_4\rangle,\langle p_2,t_5\rangle,$
$\langle t_4,In_1\rangle,\langle Out_1,t_2\rangle,\langle t_2,Out\rangle,\langle t_5,Out\rangle\}$;

$C=C_1\cup\{C(p_1)=CONDS,C(In)=ARGU,$
$C(Out)=OUTPUT,C(p_2)=BOOLVAL\}$;

$G=G_1\cup\{G(t_1)=\text{NULL},$
$G(t_2)=(o.re==failed),$
$G(t_3)=(o.re==success),$
$G(t_4)=(bo==\text{true}),$
$G(t_5)=(bo==\text{false})\}$;

$E=E_1\cup\{E(\langle In,t_1\rangle)=x,E(\langle t_1,In_1\rangle)=x_1,$
$E(\langle t_4,In_1\rangle)=1'x_2,$
$E(\langle Out_1,t_2\rangle)=E(\langle Out_1,t_3\rangle)=o,$
$E(\langle p_2,t_5\rangle)=E(\langle p_2,t_4\rangle)=bo,$
$E(\langle t_2,Out\rangle)=\langle failed,q\rangle,$
$E(\langle t_5,Out\rangle)=1'\langle success,q\rangle,$
$E(\langle t_1,p_1\rangle)=E(\langle p_1,t_3\rangle)=$
$E(\langle t_4,p_1\rangle)=1'con,E(\langle t_3,p_2\rangle)=1'bo\}$;

颜色集 Σ、初始函数 I 同定义 2.

变迁 t_1 的功能是:依据输入生成服务 S_1 的引发条件和迭代条件. 变迁 t_2 在服务 S_1 非成功执行时引发,并输出结果到库所 Out. 变迁 t_5 在完成迭代时,输出结果到库所 Out. 变迁 t_4 在符合迭代条件时执行,并将服务 S_1 的输入送到库所 In_1. 变迁 t_3 取迭代条件与输出结果进行判断,若满足迭代条件,则输出

true；否则，输出 false.

以上针对两个服务的情况，定义了并发、选择、顺序、迭代运算. 很容易将它们推广到多个服务运算的情况.

2.2　模型性质

服务运算是对已知服务进行组合，生成新服务的代数运算. 所有服务组合的运算对服务集合都应该是封闭的，即对服务进行组合运算，所得结果仍是服务.

性质 1.　设 ξ 为所有服务的集合，$O=\{\cdot, \|, \oplus, n\}$. 则 O 对 ξ 封闭.

由定义 5，运算 $\cdot$，$\|$，n 和 $\oplus$ 对 ξ 都是封闭的. 下面证明，这几种运算的复合也是封闭的. 四种运算的优先级迭代最高，其余相同，即三种运算 $\cdot$，$\|$ 和 $\oplus$，先出现先计算. 运算的复合是指在一个服务组合包含多个运算. 对运算次数用数学归纳法，可证明 ξ 对运算的复合也封闭.

下面给出功能等价的定义. 在功能等价前提下，所定义的服务运算集 $O=\{\cdot, \|, \oplus, n\}$ 具有一些重要性质.

定义 6.　设服务 $S=(\Sigma, P, T, F, C, G, E, In, Out, I)$，若标识 M 满足 $(M\in R(M_0))\wedge(M(Out)\neq \mathrm{NULL})$，则称 M 为终止标识. 称 S 所有终止标识的集合为终止标识集，记作 M_e.

终止标识即从初始标识可达，且输出库所 Out 中包含 Token 的标识.

定义 7.　两个服务：

$S_1=(\Sigma, P_1, T_1, F_1, C_1, G_1, E_1, In_1, Out_1, I_1)$,

$S_2=(\Sigma, P_2, T_2, F_2, C_2, G_2, E_2, In_2, Out_2, I_2)$.

对任意 $I_1=I_2$，服务 S_1 引发变迁序列 σ 到达终止标识 M_e. 服务 S_2 引发变迁序列 σ' 到达终止标识 M'_e. 若 $M_e(Out)=M'_e(Out)$，则服务 S_1，S_2 功能等价，记为 $S_1\cong S_2$.

定义 7 说明，若两事务对相同输入，所得结果相同，则二者功能是等价的. 在实际应用中，功能等价的服务可互相替代. 这可以增加服务调用的灵活性. 如图 6 中的两个组合服务 S，S'，对相同输入，若子服务 S_1 能够正常执行，则服务 S，S' 都会继续调用服务 S_2. 同样，若 S_2 执行成功，服务 S，S' 也都会继续调用 S_3，最终返回执行结果 S_3. 否则，在 S_1，S_2 不能正常执行的情况下，两个服务得到的结果也是相同的. 因此，服务 S，S' 功能等价.

性质 2.　设所有服务的集合为 ξ，$S_1, S_2, S_3\in\xi$，则运算 $\cdot$，$\|$，n 和 $\oplus$ 具有下列代数性质：

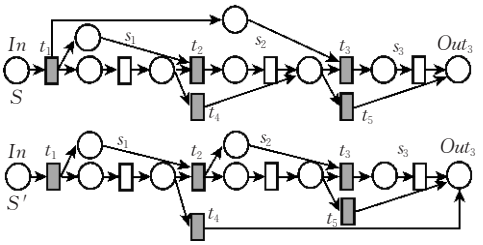

图 6　两个等价的服务组合

$$(S_1\cdot S_2)\cdot S_3\cong S_1\cdot(S_2\cdot S_3) \tag{1}$$

$$(S_1\oplus S_2)\oplus S_3\cong S_1\oplus(S_2\oplus S_3) \tag{2}$$

$$(S_1\| S_2)\| S_3\cong S_1\|(S_2\| S_3) \tag{3}$$

$$S_1\oplus S_2\cong S_2\oplus S_1 \tag{4}$$

$$S_1\| S_2\cong S_2\| S_1 \tag{5}$$

$$S_1\cdot(S_2\| S_3)\cong(S_1\| S_2)\cdot(S_1\| S_3) \tag{6}$$

$$S_1\cdot(S_2\oplus S_3)\cong(S_1\cdot S_2)\oplus(S_1\cdot S_3) \tag{7}$$

$$S_1\|(S_2\oplus S_3)\cong(S_1\| S_2)\oplus(S_1\| S_3) \tag{8}$$

$$n(S_1\| S_2)\cong nS_1\| nS_2 \tag{9}$$

不难看出，在图 6 中，$S=(S_1\cdot S_2)\cdot S_3$，$S'=S_1\cdot(S_2\cdot S_3)$. 由定义 7，式(1)成立. 其它公式可同样证明. 利用这些代数性质，可对组合服务进行化简. 由此可简化组合服务 CP-net 模型的规模，从而降低服务组合的复杂度.

推论 1.　设 ξ 为所有服务的集合，则 $\langle\xi, \cdot\rangle$ 是一个半群，$\langle\xi, \|\rangle$ 和 $\langle\xi, \oplus\rangle$ 是可交换半群.

证明.　由性质 1 式(1)运算 $\cdot$ 满足结合律. 而由运算 $\cdot$ 定义知，它将两个服务 S_1，S_2 组合为一个服务 $S=S_1\cdot S_2$. 因此，它对集合 ξ 封闭. 可见 $\langle S, \cdot\rangle$ 是一个半群. 同理，由性质 2 中式(2)～(5)，运算 $\|$，$\oplus$ 满足结合律、交换律. 再由 $\|$ 和 $\oplus$ 的定义知，$\langle\xi, \|\rangle$ 和 $\langle\xi, \oplus\rangle$ 是可交换半群.　　证毕.

实际上，若规定空服务为 Δ 元，且每个服务都有逆服务(即将其结果清除，使返回到服务调用前状态的服务)，则 $\langle\xi, \cdot\rangle$ 是一个群. 而 $\langle\xi, \|\rangle$ 和 $\langle S, \oplus\rangle$ 是可交换群. 因此，本文所定义的运算具有较好的代数性质，可以用近世代数的方法对其性质进行深入研究.

性质 3.　设服务 $S=(\Sigma, P, T, F, C, G, E, In, Out, I)$，初始标识为 $M_s(\forall p\in P: M_s(p)=I(p))$，则 S 有界.

证明.　由定义 3 知，服务的初始化函数 I 为 $I(In)=1'x$；$\forall p\in P$：$p\neq In\rightarrow I(p)=\mathrm{NULL}$. 而由定义 4，原子服务只有 In，Out 两个库所，则 M_s 是有界标识. 若变迁 s 可引发，$M_s[s>M$：$M(In)=\mathrm{NULL}$，$M(Out=1'o)$，则 M 也是有界标识. 而原子服务最多只有这两个运行状态，故在原子服务中，每

一个库所都是有界的，即原子服务有界.

由定义 4 及变迁引发规则，对原子服务进行 $\cdot$，$\|$，$\oplus$，n 四种操作，所得到的服务仍然是有界的. 再对原子服务和这些组合服务进行同样四种操作，所得服务仍然是有界的. 如此递推下去，复合操作所得到服务也是有界的. 所以，服务 S 有界. 证毕.

另外，服务 S 从初始状态 M_s 开始，若 M_s 不是一个死标识，则总存在一个可引发的变迁序列，使系统运行到达某一终止状态. 因此，可得到下面结论.

性质 4. 设服务 $S=(\Sigma,P,T,F,C,G,E,In,Out,I)$，初始标识为 $M_s(\forall p\in P:M_s())=I(p)$. 若 $\exists t\in T:M_s[t>$，则

$$\exists\sigma\in T^*:M_s[\sigma>M\wedge M\in M_e \text{ 且}$$

$$\forall M\in R(M_s)-M_e:\exists t\in T:M[t>.$$

证明.

(1) 对原子服务 S，只要 $M_s(In)\neq$NULL，就可以引发服务变迁 s. 当然，若参数不正确或其它原因可能导致服务调用不能正常执行时，服务变迁 s 引发，并在库所 Out 中得到 Token$\langle failed,outp\rangle$. 若参数正确，且服务能够正常执行，则在 Out 中得到 $\langle success,outp\rangle$. 显然结论成立.

(2) 设 S_1，S_2 是两个原子服务，现证对 $S=S_1\cdot S_2$ 以上结论成立. 由顺序操作定义，对若 M_s 不是死标识，则 $M_s[s_1>M,M(Out_1)=1'o$ 且 $o=\langle success,q\rangle$ 或 $\langle failed,q\rangle$. 此时，在变迁 t_1，t_2 中，必有且仅有一个变迁可引发. 若变迁 t_2 引发，则到达终止状态. 若变迁 t_1 引发，则生成服务变迁 s_2 的调用条件，变迁 s_2 引发后到达终止状态. 结论成立.

同理可证明，对运算 $S=S_1\oplus S_2$，$S=S_1\|S_2$，$S=nS_1$ 结论成立.

(3) 由性质 1，运算集 $O=\{\cdot,\|,\oplus,n\}$ 对服务集合 ξ 封闭，则对原子服务进行一步组合服务运算（一次 O 中运算符定义的运算）的情况，上述结论成立.

假设对 n 步复合计算所得到的服务结论成立，则 $n+1$ 步复合计算所得服务可看作两个复合服务 S_1，S_2 进行的一次计算. 其中复合事务 S_1，S_2 的复合计算次数为 n，每一个服务的复合计算次数一定都不超过 n. 由假设，结论对 S_1，S_2 成立，由步(2)，结论对 S_1，S_2 进行 O 中给定计算结论也成立. 所以，结论对 $n+1$ 步计算所得到的服务成立. 证毕.

根据上述性质，本文所定义的组合运算，能够保证每个组合服务的结构正确，且可正常执行到终止状态.

2.3 组合服务的构造与执行

下面由定义给出组合服务的构造方法.

定义 8. 组合服务是按以下步骤生成 CP-net 模型：

(1) 确定需要包含的服务的详细信息，如在何处调用、接口参数情况等；

(2) 对每一个构成组合服务的服务 S_i，生成 CP-net 模型，并严格定义输入输出字符串格式；

(3) 依据语义，将原子服务在组合服务中的关系，用顺序、并发、选择、迭代四种运算描述组合服务的执行流程；

(4) 利用服务运算的功能等价性，对服务组合表达式进行功能等价化简，从而依据定义 4 给出的运算结构，构造生成新服务的 CP-net 模型.

由定义 8 得到的组合服务，是一个具有 2.2 节性质的 CP-net 模型. 它可模拟原系统运行，并发现可能的错误. CP-net 从初始状态开始执行，变迁依据有色网的变迁引发规则依次引发，直至到达某一终止状态为止. 基于服务组合的 CP-net 模型，可对组合服务的执行步骤作如下描述.

算法 1. 组合服务的执行算法.

```
输入：组合服务的 CP-net 模型 SM(Service Model)
输出：组合服务执行结果(⟨success,q⟩或⟨failed,q⟩)
步骤：
{
 MR={M_s}；   //非死标识集合
 While MR<>NULL
   {
     取 M∈MR，MR=MR−{M}；
     ET={∀t∈T:M[t>}；//可执行变迁集
     While ET<>NULL do
     {
       取一个变迁 t∈ET，ET=ET−{t}；
       If(t 是控制变迁)
           { 引发变迁 t；
             M′=M[t>；
             If M′∈M_e exit；//达到某终止标识
             If ∃t∈T:M′[t> MR=MR∪{M′}；
             }
         Else
           {
             在相应网络结点执行服务调用；
             等待返回结果；
             //依据返回结果设置标识
             M′(·t)=M(·t)−1′x；//x∈ARGU
             If 成功执行 o=⟨success,q⟩；
             Else
```

```
            o=⟨failed,q⟩; //q∈ARGU
            M′(t˙)=M(t˙)+1′o; //o∈OUTPUT
            If M′∈Me exit; //达到某终止标识
            If ∃t∈T: M′[t>MR=MR∪{M′};
            } //End of If(t 是控制变迁)
        } //End of While ET<>NULL
    If M′∈Me exit;
    } //End of While MR<>NULL
    If M′∈Me   return(o);    //o 为 M 中 Token
    Else error('exit with error')
} //算法结束
```

性质 3 和性质 4 可保证算法是正确、可终止的.

3　一个实例

下面给出一个旅行订票系统实例. 假设客户要预约出行，则旅行社要做的工作是：预定从本地飞往目标城市的机票，预定从机场到目的地的汽车票及所在城市的宾馆房间. 设有两家航空公司的机票可以预定，且客户选择先考虑 A 公司. 若 A 公司预定不成功，再预定 B 公司机票. 宾馆有两家 C，D 可以预定，客户没有要求哪家优先.

分析可知，预定 A，B 公司机票为顺序关系，预定机票与预定汽车票两个"组合服务"可以并发，而这两个并发服务又同预定宾馆服务成顺序关系. 预定宾馆的两个服务是选择关系. 假设 S_1，S_2，S_3，S_4，S_5 是原子服务，分别代表预定 A 和 B 公司机票、预定汽车票、预定 C 和 D 宾馆房间. 目标服务 $S=((S_1 \cdot S_2) \parallel S_3) \cdot (S_4 \oplus S_5)$，如图 7 所示. 图 7 是由有色网工具 CPNTools[13] 生成的旅行预定系统的有色网模型，其中各控制变迁功能分别是

t_1（即图中 t1，后同）：组合服务开始，生成预定机票和预定车票服务输入条件；

t_2：预定机票服务开始，生成 S_1，S_2 的输入条件；

t_3：当 S_1 执行失败时生成服务 S_2 输入条件；

t_4：当 S_1 执行成功时传送执行结果；

t_5：当预定机票成功（S_1，S_2 有一个执行成功），且 S_3 成功时开始预定房间服务的执行；

t_{11}：当预定机票（S_1 及 S_2）失败，或预定车票失败时，将失败的消息送到库所 Out；

t_{12} 生成 S_4，S_5 的输入条件及选择条件；

t_6，t_7 功能相似，分别在 S_4 和 S_5 执行成功后将成功消息送到库所 Out；

t_8，t_9 功能相似，分别在 S_4 和 S_5 执行失败后，回送选择条件，并准备再次选择；

t_{10} 当服务 S_4 和 S_5 均失败时，将执行失败的结果输出到库所 Out.

变迁 restart 表示一个服务的结束与下一个服务的开始. 系统运行到达终止状态，在库所 Out 中输出结果后引发. 引发结果是进入下一次服务调用.

为了图形显示简洁，我们略去了其中一些弧上的弧函数和变迁的 G 函数及功能函数. 读者可根据前面各运算的定义自行补充完整.

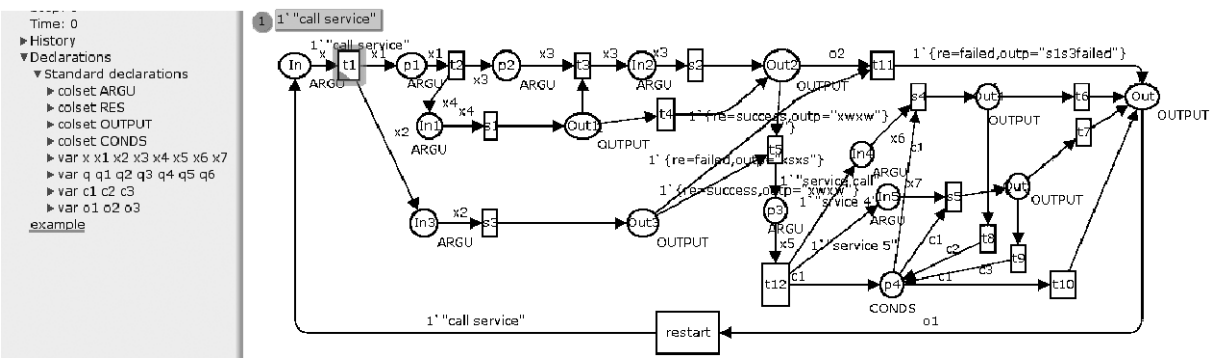

图 7　旅行预定系统 $S=((S_1 \cdot S_2) \parallel S_3) \cdot (S_4 \oplus S_5)$

依据算法 1，在初始标识下，$ET=\{\forall t \in T: M_0[t>\}=\{t_1\}$. 在变迁 t_1 引发后，生成预定机票和预定车票服务的输入条件. 此时，$ET=\{t_2, s_3\}$，可以执行控制变迁 t_2 及对 S_3 进行服务调用. 设 t_2 执行，则生成服务 S_1 和 S_2 的输入条件. 当 $ET=\{s_1, s_3\}$ 时，若服务调用 S_1 执行，则依据其执行是否成功，可确定 $ET=\{t_3, s_3\}$（执行不成功）或 $ET=\{t_4, s_3\}$（执行成功）. 如此运行，最终可到达终止状态. 变迁 restart 的功能是消耗输出库所中的托肯并生下一次服务调用的请求条件. 在系统运行到达终止状态后通过引发变迁 restart 可以进入下一次服务调用.

由此可知，此组合服务的运行是可终止且有效的. 由于服务的语义信息比较复杂，图 7 只对较简单的情况进行了模拟. 系统没有详细设定服务变迁的

功能函数,只通过弧函数限定了其输入输出的格式.可采取文件读入的办法,从事先准备好的文件读入调用服务的执行结果由服务变迁的功能函数生成服务变迁的输出.系统通过弧函数和变迁功能函数对控制变迁进行了较严格的定义.此系统是可以运行的.在 CPNTools 环境下,可以模拟组合服务的调用执行过程,并可生成 CP-net 模型的可达图和状态空间分析报告.

由实例可知,此有色网模型可清晰地表达组合服务的组成逻辑.利用 CPNTools 也可完成系统行为的模拟以及对系统有界性、活性、可达性等重要性质进行分析.通常,组合服务的性质与其基础 CP-net 模型性质是相对应的.因此,研究组合服务特性与其基础 CP-net 模型性质的对应关系,并基于此开发一套专用服务组合验证与分析工具,是一项非常迫切的任务.我们基于算法 1 和有色网工具 CPNTools,对服务组合的有色网自动建模分析工具研制作了初步探索,正在进行和已完成的实验内容包括:

(1)建立服务计算语言,用户通过这种语言可以描述组合服务中服务之间的关系及所涉及服务的功能.由系统按语法自动生成组合服务的 CP-net 模型.

(2)建立一种组合服务性质与 CP-net 模型性质之间的对应关系.逐步实现模型对用户组合服务语义的透明化.也就是说用户可在服务层次上提出服务组合的要求及性质分析要求.系统建立组合服务的 CP-net 模型,运行并分析模型的性质.同时系统依据性质对应关系回答用户给出的服务性质分析要求.

(3)目前 CP-net 模型可用 CPN XML 语言描述.CPN XML 是 XML 语言的一个特例,CPNTools 利用它存储 CP-net 模型.我们正考虑通过对 CPN XML 语言的分析建立模型到 WSCI 或 BPEL4WS 语言(或扩展语言)的映射,加速向应用的转化.

4 结 论

本文提出了一种基于有色 Petri 网的 Web 服务组合模型,定义了几种服务组合运算,并分析了它们的运算性质.根据服务组合运算性质,给出了服务组合的构造与执行算法.服务组合的 CP-net 模型能够清晰表达服务组合的组成逻辑及明确描述服务之间的相互关系.一方面,用户可全面或部分地分析组成服务之间的逻辑关系;另一方面,用户还可以利用有色网的可达图、不变量等工具,对服务组合的动态行为进行模拟,分析其动态性质.本文给出的服务组合运算的代数性质及功能等价性,可以简化服务组合系统的规模,有效降低系统的分析复杂度,从而为分析大型复杂服务组合系统奠定了理论基础.但本文对服务组合 CP-net 模型的研究,仅仅处于初始阶段,还需要不断完善.因此,我们下一步要建立一套完整的组合服务模型自动生成、性质分析与验证工具.同时,将研究如何在保持服务语义的条件下,利用代数性质对服务组合进行化简.另外,结合事务模型,深入研究存在事务及安全条件约束的复杂服务的有效组合与调度算法,也是我们近期的工作.

参 考 文 献

1 Tsalgatidou A., Pilioura T.. An overview of standards and related technology in Web services. Distributed and Parallel Databases, 2002, 12(2/3): 135～162

2 Rao Yuan, Feng Bo-Qin, Li Zun-Chao. Survey of services composition technique research on Web Services. Systems Engineering and Electronics, 2005, 27(8): 1481～1489(in Chinese)
(饶 元,冯博琴,李尊朝.基于 Web Services 的服务合成技术研究综述.系统工程与电子技术, 2005, 27(8): 1481～1489)

3 Pustjarvi J., Tirri H., Veijalainen J.. Reusability and modularity in transactional workflows. Information Systems, 1997, 22(2/3): 101～120

4 Korhonen J., Pajunen L., Puustjarvi J.. Automatic composition of Web service workflows using a semantic agent. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI 2003), Halifax, Canada, 2003, 566～569

5 Peer J.. Bringing together semantic Web and Web services. In: Proceedings of the International Semantic Web Conference, Sardinia, Italia, 2002, 279～291

6 Ankolekar A., Burstein B., Hobbs J. *et al*. DAML-S: Semantic markup for Web services. In: Proceedings of the 1st International Semantic Web Working Symposium, Standford, CA, 2001, 411～430

7 Liao Jun, Tan Hao, Liu Jin-De. Describing and verifying Web service using PI-calculus. Chinese Journal of Computers, 2005, 28(4): 635～642(in Chinese)
(廖 军,谭 浩,刘锦德.基于 PI-演算的 Web 服务组合的描述和验证.计算机学报, 2005, 28(4): 635～644)

8 Rachid H., Boualem B.. A Petri net based model for Web service composition. In: Proceedings of the 14th Australian Database Conference on Database Technologies, Adelaide, South Australia, 2003, 191～200

9 Tao X. F., Jiang C. J.. Formalizing Web service and modeling

Web service based system based on object oriented Petri net. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer-Verlag, 2004

10 Tao Xiao-Feng, Sun Jian. Web service composition based on object oriented Petri net. Computer Applications, 2005, 25(6): 1424～1426(in Chinese)

(陶晓峰,孙 健.基于面向对象 Petri 网的 Web 服务合成.计算机应用,2005,25(6):1424～1426)

11 Jensen K.. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science. Berlin, Heidelberg, New York: Springer-Verlag, 1997

12 W3C Web Services Activity. http://www.w3.org/2002/ws/

13 CPN Tools homepage. http://www.daimi.au.dk/CPNTools/

GUO Yu-Bin, born in 1973, Ph. D. candidate. Her research interests include service computing, P2P transaction management, Petri nets etc.

DU Yu-Yue, born in 1960, Ph. D., professor. His research interests include CSCW, service computing, formal technology, and Petri nets etc.

XI Jian-Qing, born in 1962, Ph. D., professor and Ph. D. supervisor. His research interests mainly include service computing, P2P data management, database and network computing etc.

Background

Web services propose a service-oriented paradigm for computing in which loosely coupled services can be combined into business processes collaboratively. But it is a challenge to integrate them to create new value-added Web services in transparent ways. Therefore, technologies for Web service composition are hot to researchers and business users.

The project is aiming at developing more efficient methods for modeling and verifying Web service composition.

Authors propose a CP-net model for Web service composition. Some algebra and dynamic properties of the model are analyzed. And algorithms to construct and execute a composite service are delivered also. The results would be helpful to researchers who interest in Web service composition.