# 一种基于逻辑 Petri 网的 Web 服务簇组合方法

吴洪越 杜玉越
（山东科技大学信息科学与工程学院 山东 青岛 266590）

**摘 要** Web 服务组合是实现快速服务增值和软件重用的重要方式，但现有的静态服务组合和动态服务组合方式都有待于进一步完善. 文中集成静态服务组合和动态服务组合的优点，提出了一种在虚拟层面上基于服务簇进行服务组合的思想；研究并提出了一种基于服务簇的服务组合方法，并应用逻辑 Petri 网对其进行形式化建模描述；给出了服务簇网的基本组合模型，并分别对其完备性继承作了分析；研究了服务簇网组合的代数运算性质. 最后，通过实验证明了服务簇网组合运算的可行性和有效性.

**关键词** Web 服务；服务簇；服务组合；逻辑 Petri 网；服务簇网
**中图法分类号** TP311 **DOI 号** 10.3724/SP.J.1016.2015.00204

# A Logical Petri Net-Based Approach for Web Service Cluster Composition

WU Hong-Yue DU Yu-Yue
(*College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong* 266590)

**Abstract** Web service composition is an important way to achieve value-added services and software reuse, but the existing static service composition and dynamic service composition methods are all needed to be further improved. This paper integrates the advantages of static composition and dynamic composition methods and proposes a method that composes service clusters in a virtual layer to realize service composition. A service composition approach based on service clusters is developed and logical Petri nets are used for its formal modeling. Basic composition models of service cluster nets (SCNs) are presented and the inheritance conditions of their completeness are analyzed respectively. The algebraic properties of SCN composition are also presented. Finally, experimental results are presented to illustrate the applicability and effectiveness of SCN composition.

**Keywords** Web service; service cluster; service composition; logical Petri nets; service cluster nets

## 1 引 言

Web 服务技术具有高度的互操作性、跨平台性和松散耦合等优点. 近年来，云计算、物联网等新型计算模式的涌现和发展，极大地推动了 Web 服务技术的发展. Web 服务技术引起了世界范围内学术界和产业界的广泛关注. 随着大数据时代的到来，Web 服务不仅在数量上迅猛增长，并且在越来越多的领域得到广泛应用.

然而，单个 Web 服务提供的功能较为单一，难以满足复杂多样的用户请求，只有通过对已有服务进行组合，才能发挥其巨大潜力. Web 服务组合技术将多个 Web 服务进行组合，构造粒度更大的流程服务，以此提供更多功能复杂的服务来满足不同用户的复杂需求，与此同时也极大地提高软件的生产效率.

收稿日期：2011-11-30；最终修改稿收到日期：2014-03-21. 本课题得到国家自然科学基金(61170078)、国家“九七三”重点基础研究发展规划项目基金(2010CB328101)、教育部高等学校博士学科点专项科研基金博导类资助课题(20113718110004)、山东科技大学科研创新团队支持计划项目(2011KYTD102)和青岛市科技计划基础研究项目(13-1-4-116-jch)资助. **吴洪越**，男，1988 年生，硕士研究生，主要研究方向为面向服务的计算、Petri 网理论与应用. E-mail：hongyue_wu@163.com. **杜玉越**，男，1960 年生，博士，教授，博士生导师，主要研究领域为 CSCW、软件工程、形式化技术、Petri 网理论与应用.

近年来，随着大数据时代的到来，服务数量迅速增加，为服务请求者提供了更多的选择，同时也给服务的查找和组合带来了更大的压力，服务组合过程变得缓慢而复杂. 组合服务的动态自适应性较差，当服务失效或用户请求改变时，很难找到替代服务. 为此，一些学者提出了服务聚类[1-3]的思想，将功能相同或相似的服务映射为一个服务簇，利用服务簇作为服务请求和响应的基本单位. 服务粒度的增大，可以有效地缩减服务查找空间，提高搜索效率，减小服务查找和绑定难度. 服务失效时可以在簇内查找相似服务进行替换，因此，大大提高了服务的自适应性.

本文以服务簇为基本单位，在抽象层面上基于服务簇进行服务组合，不仅可以大大提高服务的发现和组合效率，而且可以充分利用 BPEL 流程，压缩服务搜索空间，极大地提高服务的自适应性. 由于服务簇模式下存在服务参数不确定、用户需求多样等不确定性因素，因此，传统的 Petri 网的建模方法已经不再适用于以服务簇为基本单位的服务组合. 本文采用逻辑 Petri 网(Logic Petri Net，LPN)[4-8]对服务簇模式的服务组合进行建模描述，并给出了其完备性分析. 逻辑 Petri 网是抑制弧 Petri 网的一种高级抽象，已成功应用于电子商务[5-6]、实时协同系统[7-8]等领域. 逻辑 Petri 网适用于描述传值不确定性，因此易于描述与分析基于服务簇的服务组合模型.

本文第 2 节给出基于服务簇网组合的动机和服务簇网的层次结构；第 3 节介绍逻辑 Petri 网的基本知识；第 4 节对 Web 服务和服务簇进行形式化建模描述；第 5 节给出服务簇网的组合模型，并对其完备性继承和代数运算性质作分析；第 6 节通过实验分析证明基于服务簇网的服务组合方法的可行性和有效性；第 7 节对全文进行总结，并简要介绍下一步工作.

## 2　服务簇模式

传统的 Web 服务组合方法分为动态服务组合和静态服务组合. 动态服务组合是一种在线的、动态的和实时的组合方式. 当用户提出服务请求时，系统根据用户请求动态地选择 Web 服务并进行组合，提供个性化服务来满足不同用户的需求. 静态服务组合是一种离线的或预编译的稳定的组合方式，它将简单的具有不同功能的 Web 服务组合成一种具有特殊功能的服务进行封装，并作为一种新的服务在服务库中发布，供用户调用，以此达到服务增值的目的.

然而，无论是动态服务组合还是静态服务组合都存在缺陷. 动态服务组合需要根据用户的具体需求在茫茫服务库中寻找单个服务组件，效率极其低下，而且不能重复利用服务组合的 BPEL 流程. 静态服务组合虽然效率高，且能重复利用服务组合流程，但是组合服务中的单个服务组件并不一定适合某个具体的用户需求. 因此，封装好的组合服务并不能满足不同用户的个性化需求，并且自适应性较低. 如果将所有可能的服务组合都进行封装，又会造成极大的浪费，并导致服务空间爆炸问题.

因此，本文集成动态服务组合和静态服务组合的优点，给出一种基于服务簇模式的服务组合方法：(1) 在 Web 服务的物理资源层之上建立虚拟资源层，将功能相同或相似的服务聚集为一个服务簇；(2) 在虚拟资源层之上建立业务模型层，对内在关系比较稳定的抽象服务簇进行静态组合，得到服务簇网；(3) 将功能独立的服务簇映射到业务模型层，作为原子服务簇网；(4) 对服务簇网进行存储，将服务簇网作为用户请求和响应的基本单位. 服务簇模式的层次结构模型如图 1 所示.

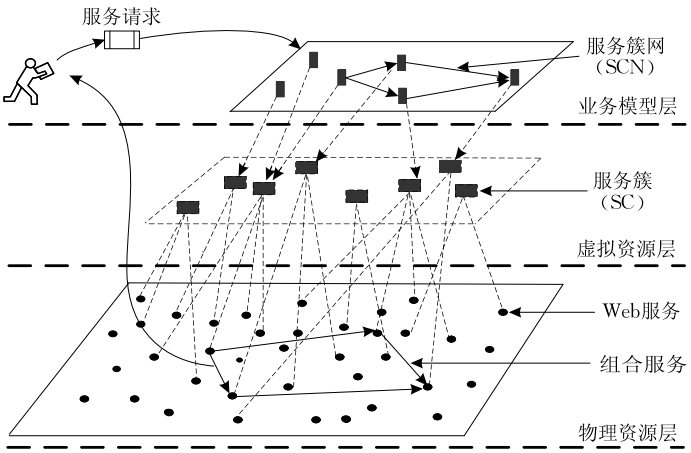



图 1　服务簇网层次结构模型

当用户发出服务请求时，首先在业务模型层对服务请求作出响应，查找满足用户请求的服务簇网. 如果业务模型层的服务簇网不能满足用户需求，则基于服务簇网进行动态组合，得到满足用户需求的服务簇网. 然后，针对用户具体请求，借助抽象服务簇到单个服务的绑定算法，为服务簇网中的每个服务簇选择最符合用户需求的单个服务，得到最能满足用户需求的组合服务. 因此，基于服务簇的服务组合使得用户不必在服务库中逐一地查找特定服务，解决了动态服务组合服务发现和组合效率低的问题. 同时，又可以重复利用 BPEL 流程，实现流程的复用，根据具体的用户需求选择绑定适合用户的最佳服务. 因此，这种服务簇组合模式，又解决了静态服务组合不能满足个性化服务需求以及服务空间爆炸问题.

本文只给出了业务模型层上服务簇网的组合方法及其完备性分析，服务请求处理、服务簇发现、服务簇到具体服务的绑定等问题的研究将在以后的工作中完成.

## 3 逻辑 Petri 网

传统的基于 Petri 网的描述方法只能对单个的 Web 服务组合进行描述，不适用于以服务簇为基本单位的服务组合. 因此，本文采用基于逻辑 Petri 网的方法描述和分析基于服务簇的服务组合.

下面简单介绍 Petri 网的基本概念，更深入的了解参阅文献[9-10].

**定义 1.** 网[10]. 三元组 $N=(P,T,F)$ 称为一个网，当且仅当

(1) $P$ 是一个有限库所集，$T$ 是一个有限变迁集；

(2) $P\cup T\neq\varnothing$，$P\cap T=\varnothing$；

(3) $F\subseteq(P\times T)\cup(T\times P)$ 是一个弧集.

**定义 2.** 输入集和输出集[10]. $\forall x\in P\cup T$，${}^{\bullet}x=\{y\mid(y,x)\in F\}$ 称为 $x$ 的输入集；$x^{\bullet}=\{y\mid(x,y)\in F\}$ 称为 $x$ 的输出集.

**定义 3.** Petri 网[10]. $PN=(N,M)$ 称为一个 Petri 网，当且仅当

(1) $N=(P,T,F)$ 是一个网；

(2) $M:P\rightarrow\{0,1,2,\cdots\}$ 是 $PN$ 的一个标识函数；

(3) 引发规则：

① $\forall p\in{}^{\bullet}t:M(p)>0$，则称变迁 $t$ 在标识 $M$ 使能，记为 $M[t\rangle$；

② 若 $t$ 在标识 $M$ 使能，则 $t$ 可以引发且引发后产生一个后继标识 $M'$ 记作 $M[t\rangle M'$，其中

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

**定义 4.** 可达标识集[10]. 设 $PN=(P,T,F,M_0)$ 为一个 Petri 网，其中 $M_0$ 是 $PN$ 的初始标识. $PN$ 的可达标识集 $R(M_0)$ 定义为满足下面两个条件的最小集合：

(1) $M_0\in R(M_0)$；

(2) 若 $M\in R(M_0)$，且存在 $t\in T$ 使得 $M[t\rangle M'$，则 $M'\in R(M_0)$.

在本文 Petri 网的图表示中，圆圈表示库所，矩形框表示变迁，圆点表示标识(token).

逻辑 Petri 网(Logical Petri Net，LPN)是抑制弧 Petri 网的一种高级抽象[11]. 在 LPN 中，在变迁上添加逻辑输入表达式 $f_I$ 和逻辑输出表达式 $f_O$，以此来描述变迁使能条件的不确定性和变迁引发时输入、输出库所状态改变的不确定性.

通常将逻辑输入表达式 $f_I$ 和逻辑输出表达式 $f_O$ 化为标准析取范式，使其中的每一个析取子句都是相关库所的合取式，称为逻辑表达式的子合取式. 如果一个变迁的输入库所集满足其逻辑输入表达式 $f_I$，则意味着它满足 $f_I$ 的一个子合取式 $f$，此时变迁使能. 变迁引发时按照 $f$ 对输入库所集合中的库所作相应变化，同时保证输出库所集满足 $f_O$ 的一个子合取式.

为了更加详细地描述库所与表达式的关系，用符号 $p|_f$ 表示库所 $p$ 在合取式 $f$ 中的出现形式，$p|_f=.T.$ 表示 $p$ 在 $f$ 中以 $p$ 的形式出现，$p|_f=.F.$ 表示 $p$ 在 $f$ 中以 $\neg p$ 的形式出现，$.T./.F.$ 表示逻辑真值为真/假. 例如，在合取式 $f=p_1\wedge p_2\wedge\neg p_3$ 中，$p_1|_f=.T.$，$p_2|_f=.T.$，$p_3|_f=.F.$.

**定义 5.** 必要库所. 设表达式 $f$ 为库所集 $P$ 上的逻辑表达式，$p\in P$，称 $p$ 为表达式 $f$ 的必要库所，当且仅当对 $f$ 的任意子合取式 $f_i$ 都有 $p|_{f_i}=.T.$ 成立，记为 $p|_f=.T.$.

**定义 6.** 逻辑表达式. 设 $f$ 为关于库所集 $P$ 的逻辑表达式，对 $p\in P$，$p|_M$ 表示 $M$ 下 $p$ 的逻辑真值，记

$$p|_M=\begin{cases}.T., & \text{当 } M(p)=1\\ .F., & \text{当 } M(p)=0\end{cases}.$$

将所有库所的逻辑真值代入逻辑表达式，得到逻辑表达式 $f$ 的真值，$f|_M=.T./.F.$ 表示标识 $M$ 满足/不满足表达式 $f$.

**定义 7.** 逻辑 Petri 网. 设 $LN=(P,T,F,I,O)$，

$LPN=(LN,M)$称为一个逻辑 Petri 网当且仅当：

(1) $P$ 是一个有限库所集合；

(2) $T$ 是一个有限变迁集合，满足 $T\cup P\neq\varnothing$，$\forall t\in T$：${}^{\cdot}t\cap t^{\cdot}=\varnothing$，且 $\forall t\in T$，$t$ 的所有输入库所受一个逻辑输入表达式 $f_I$的限制，所有输出库所受一个逻辑输出表达式 $f_O$的限制；

(3) $F\subseteq(P\times T)\cup(T\times P)$是一个有向弧集合；

(4) $I$ 是一个逻辑输入函数，$\forall t\in T$，$I(t)=f_I$是变迁 $t$ 上的逻辑输入表达式；

(5) $O$ 是一个逻辑输出函数，$\forall t\in T$，$O(t)=f_O$是变迁 $t$ 上的逻辑输出表达式；

(6) $M:P\rightarrow\{0,1\}$是 $LPN$ 的标识函数，$\forall p\in P$，$M(p)$表示 $p$ 中的标识数量；

(7) 变迁引发规则：$\forall t\in T$，设 $I(t)=f_I$，$O(t)=f_O$，如果 $f_I|_M=.T.$，即 ${}^{\cdot}t$ 在 $M$ 满足逻辑表达式 $f_I$ 的一个子合取式 $f_i$，则称 $t$ 在 $M$ 使能；若 $t$ 在 $M$ 使能，则它可以在 $M$ 引发，引发时选择 $f_O$的一个子合取式 $f_o$，引发后演变到新的标识 $M'$：

$$M'(p)=\begin{cases}0, & \text{当 } p\in{}^{\cdot}t \text{ 且 } p|_{f_i}=.T.\\ 1, & \text{当 } p\in t^{\cdot} \text{ 且 } p|_{f_o}=.T..\\ M(p), & \text{其他}\end{cases}$$

在 $LPN$ 的图表示中，将变迁的逻辑输入表达式和逻辑输出表达式标注在变迁上，如图 2(a). 当逻辑输出表达式表示为所有输出库所的合取时，可以省去逻辑输出表达式，此时的变迁称为逻辑输入变迁，如图 2(b)，逻辑输入变迁的引发等同于普通 Petri 网变迁. 同样，当逻辑输入表达式表示为所有输入库所的合取时，可以省去逻辑输入表达式，此时的变迁称为逻辑输出变迁，如图 2(c)，逻辑输出变迁的使能条件等同于普通 Petri 网变迁. 当逻辑变迁的逻辑输入表达式和逻辑输出表达式都可以省去时，逻辑变迁变为普通的 Petri 网变迁，如图 2(d).

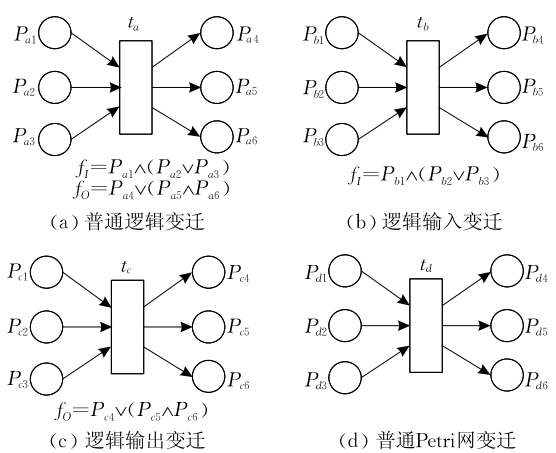



图 2　逻辑变迁模型图

# 4　Web 服务和服务簇的形式化描述

对服务进行建模描述的方法有许多种，其中，Petri 网采用可视化图形描述且有坚实的数学基础，易于表达离散事件动态系统的静态结构和动态变化，非常适合于描述服务组合，比如文献[12-15]都采用 Petri 网对服务进行建模描述.

## 4.1　Web 服务

一个单个的 Web 服务的 Petri 网描述称为服务元. Web 服务定义如下.

**定义 8.**　Web 服务. 一个服务可以描述为一个五元组$WS=(SID,BI,SF,QoS,SE)$，其中：

(1) $SID$ 表示 Web 服务的唯一标识；

(2) $BI$ 表示服务的基本信息，包括服务的名称、服务提供者的名称等；

(3) $SF$ 为服务的功能语义描述；

(4) $QoS$ 表示对服务质量的描述，$QoS$ 属性主要包括价格、响应时间、可靠性、可用性、声誉等；

(5) $SE=(P,t,F)$，表示描述该 Web 服务的服务元，其中：

① $P$ 是一个有限库所集合，表示服务元的输入、输出参数集合；

② $t$ 是服务元的服务变迁，表示服务元的服务操作；

③ $F$ 是一个有向弧的集合，表示库所集 $P$ 和服务变迁 $t$ 之间的流关系.

例如，图 3 给出了一个机票预订服务的服务元模型，表示一个机票预订服务需要输入出发日期、出发城市、目的城市等参数，服务执行完以后，可以得到飞机的起飞时间、机票价格、航班号等信息.

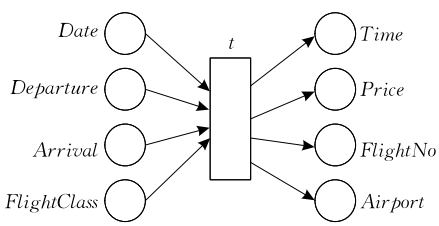



图 3　机票预订服务元模型

## 4.2　服务簇

服务聚类的思想已经被广泛认同. 服务聚类通过对多个服务的功能语义描述和输入输出参数进行相似度匹配，将功能相同或相似的服务聚在一起. 通过服务聚类处理以后，服务注册库中的服务按照功能被划分为多个服务簇，同一个服务簇中的服务功能具有高度的相似性，而不同服务簇的服务之

间功能高度相异.服务聚类的具体方法可参见文献[1-3],本文不再赘述.

服务簇的逻辑 Petri 网描述称为簇元,是将多个服务元组合到一起得到的.服务簇的形式化定义如下.

**定义 9.** 服务簇.一个服务簇可以描述为一个五元组 $SC=(CID,BI,CF,CS,CE)$,其中:

(1) $CID$ 表示 Web 服务簇的唯一标识;

(2) $BI$ 表示服务簇的基本信息,包括服务簇的名称等;

(3) $CF$ 为服务簇的功能语义描述;

(4) $CS$ 是一个标识的集合,描述了该服务簇所包含的服务的标识;

(5) $CE=(P,t,F,f_I,f_O)$,表示描述该服务簇的簇元,其中:

① $P$ 是服务簇的输入、输出库所的集合;

② $t$ 是服务簇的服务变迁,表示服务簇的服务操作;

③ $F$ 是一个有向弧的集合,表示库所集 $P$ 和服务变迁 $t$ 之间的流关系;

④ $f_I$ 是服务变迁的逻辑输入表达式,用来描述服务簇的输入参数限制;

⑤ $f_O$ 是服务变迁的逻辑输出表达式,用来描述服务簇的输出参数限制.

定义 9 指出,服务簇由服务簇唯一标识 $CID$ 进行标识.查找时,按照服务簇的功能语义描述、基本信息及其输入、输出参数进行查找.

每个簇元由一个变迁和若干个库所组成.变迁与各库所之间的逻辑关系通过逻辑表达式 $f_I$ 和 $f_O$ 来描述.例如,图 4 中给出了将两个机票预订服务聚类得到的机票预订服务簇的簇元模型.

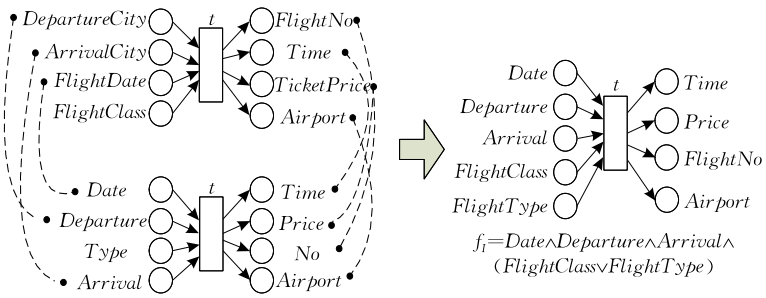



图 4 机票预订服务元模型

## 5 服务簇网及其组合

服务簇网是业务模型层上服务的存在形式,是用户提出服务请求后,服务请求和响应的基本单位.服务簇网通过虚拟层面上的组合来达到服务增值的目的.本节主要讨论业务模型层上服务簇网的组合.组合形成的服务簇网在发布之前,其完备性必须得到验证,因此,对每种组合运算的完备性继承作了分析.

### 5.1 服务簇网的定义

**定义 10.** 服务簇网.称 $SCN=(NID,BI,NF,NS,LSN)$ 为一个服务簇网,其中:

(1) $NID$ 表示服务簇网的标识;

(2) $BI$ 表示服务簇网的基本信息,包括服务簇网的名称等;

(3) $NF$ 为服务的功能语义描述;

(4) $NS$ 是一个标识的集合,描述了该服务簇网所包含的服务簇的标识;

(5) $LSN=(P,T,F,I,O,f_S,f_E)$,表示描述该服务簇的逻辑服务网,其中:

① $P$ 表示 $LSN$ 中库所的集合,$LSN$ 有两个库所子集 $P_C$ 和 $P_D$,$P_D$ 为数据库所子集,$P_C$ 为控制库所子集,满足 $P=P_C\cup P_D$,且 $P_C\cap P_D=\varnothing$. $P_C$ 包含两个特殊库所:源控制库所 $i$(开始状态)和终控制库所 $o$(结束状态),满足 ${}^{\bullet}i=\varnothing$, $o^{\bullet}=\varnothing$. 输入库所集 $P_{\mathrm{In}}=\{p\in P\mid {}^{\bullet}p=\varnothing\}$,输出库所集合 $P_{\mathrm{Out}}=\{p\in P\mid p^{\bullet}=\varnothing\}$;

② $T$ 表示 $LSN$ 中变迁的集合,$\forall t\in T$, ${}^{\bullet}t$ 和 $t^{\bullet}$ 都至少有一个控制库所;

③ $F$ 是一个有向弧的集合,表示库所集 $P$ 和变迁集 $T$ 之间的流关系;

④ $I$ 是一个逻辑限制输入函数,使对 $\forall t\in T$, $I(t)=f_I$ 是一个逻辑输入表达式,$\forall p\in {}^{\bullet}t\cap P_C$,有 $p|_{f_I}=.T.$;

⑤ $O$ 是一个逻辑限制输出函数,使对 $\forall t\in T$, $O(t)=f_O$ 是一个逻辑输出表达式,$\forall p\in t^{\bullet}\cap P_C$, $p|_{f_O}=.T.$;

⑥ $f_S$ 是集合 $P_{\mathrm{In}}$ 上的 $LSN$ 的网逻辑输入表达式,表示要使 $LSN$ 使能,其输入库所应满足的表达式;

⑦ $f_E$ 是集合 $P_{\mathrm{Out}}$ 上的 $LSN$ 的网逻辑输出表达

式，表示 $LSN$ 运行完以后，其输出库所应满足的表达式.

逻辑服务网中有两个库所子集，$P_D$ 描述服务簇网中各个服务之间以及服务网与外部环境的参数传递，$P_C$ 控制各服务簇之间的逻辑关系. 服务组合的完整执行，对应服务网中的控制标识从原库所 $i$ 移动到终库所 $o$. $P_{\mathrm{In}}$ 和 $P_{\mathrm{Out}}$ 分别表示该 $LSN$ 的输入参数和输出参数的集合. $LSN$ 中的每个变迁表示一个服务簇操作. 服务网的逻辑输入表达式 $f_S$ 和逻辑输出表达式 $f_E$ 描述了服务网的逻辑输入和输出限制，分别表示用户调用服务时所需要输入的参数的限制条件和服务组合执行完成后得到的结果，由各变迁的逻辑输入表达式和逻辑输出表达式在组合过程中不断运算得到，对组合网的性质继承问题的分析起着关键作用.

逻辑服务网 $LSN$ 是服务网组合的主要部分，本文讨论的重点在于 $LSN$ 的组合及其性质的分析. 对于一个服务簇网，其逻辑 Petri 网描述即逻辑服务网决定了服务的组合流程. 因此，在服务簇网启用之前，逻辑服务网的正确性即健壮性必须得到形式验证. 为了分析逻辑服务网的健壮性，作以下定义.

**定义 11.**　初始标识和终止标识. 设 $LSN=(P,T,F,I,O,f_S,f_E)$，称 $M_0$ 为 $LSN$ 的初始标识当且仅当：

(1) $\forall p\notin P_{\mathrm{In}}\rightarrow M_0(p)=0$；

(2) 对集合 $P_{\mathrm{In}}$ 满足 $f_S|_{M_0}=.T..$

称初始标识的集合为初始标识集，记为 $M_S$.

称 $M_e$ 为 $LSN$ 的终止标识当且仅当：

(1) $\forall p\in P_C-\{o\}, M_e(p)=0, M_e(o)=1$；

(2) $f_E|_{M_e}=.T..$

称终止标识的集合为终止标识集，记为 $M_E$.

初始标识集表示所有满足该逻辑服务网的逻辑输入表达式的标识的集合，终止标识集表示所有从初始标识可达的，满足该逻辑服务网的网逻辑输出表达式的标识的集合. 对于 $f_S$ 的标准析取范式的任意一个合取式 $f_{Si}$，都有一个唯一的初始标识 $M_{0i}$ 与之对应. 因此，一个 $LSN$ 的初始标识的数量与 $f_S$ 的标准析取范式中的合取式的数量是相等的，即 $|R(f_S)|=|M_S|$，其中 $R(f_S)$ 表示 $f_S$ 的标准析取范式的子合取式的集合.

**定义 12.**　健壮性. 设逻辑服务网 $LSN=(P,T,F,I,O,f_S,f_E)$，$M_0\in Ms$，如果 $\forall M\in R(M_0)$，$\exists M'\in R(M)$ 且 $M'\in M_E$，则称 $LSN$ 在 $M_0$ 是健壮的. 如果 $\forall M_0\in Ms$，都有 $LSN$ 在 $M_0$ 是健壮的，则称逻辑服务网 $LSN$ 是健壮的.

定义 12 说明，$LSN$ 是健壮的，即任意一个初始标识 $M_0$，无论网如何运行，都能到达一个终止标识. 因此，$LSN$ 的健壮性可以保证服务簇网针对用户的所有可能的需求按其描述的流程进行执行并得到正确的输出结果，称此时的服务簇网是完备的.

**定义 13.**　完备性. 称服务簇网 $SCN$ 是完备的，当且仅当其对应的逻辑服务网 $LSN$ 是健壮的.

根据定义 13 可以将服务簇网的完备性转化到其 $LSN$ 的健壮性来讨论.

### 5.2　服务簇网代数

同 Web 服务的静态组合方式一样，服务簇网组合以后，仍然为一个服务簇网，供用户查找调用. 服务簇网组合可通过类 BNF 的符号进行定义. 设 $SCN_1$ 和 $SCN_2$ 为两个服务簇网，代数运算符的语法如下：

$$SCN::=CAN\,|\,SCN_1\cdot SCN_2\,|\,SCN_1\oplus SCN_2\,|\,SCN_1\diamond SCN_2\,|\,SCN_1\,\|\,SCN_2\,|\,\mu SCN_1,$$

其中：

(1) $CAN$ 表示原子服务簇网，即由一个服务簇所构成的服务簇网；

(2) $SCN_1\cdot SCN_2$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 顺序组合，即先执行服务簇网 $SCN_1$ 中的服务，再执行服务簇网 $SCN_2$ 中的服务，$\cdot$ 表示顺序运算符；

(3) $SCN_1\oplus SCN_2$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行选择运算（不可兼或），即执行服务簇网 $SCN_1$ 中的服务或服务簇网 $SCN_2$ 中的服务，$\oplus$ 为选择运算符；

(4) $SCN_1\diamond SCN_2$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行任意顺序运算，即先执行 $SCN_1$ 中的服务后执行 $SCN_2$ 中的服务，或先执行 $SCN_2$ 中的服务后执行 $SCN_1$ 中的服务，$\diamond$ 表示任意顺序运算符；

(5) $SCN_1\,\|\,SCN_2$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行并发运算，即服务簇网 $SCN_1$ 和 $SCN_2$ 中的服务是并发运行的，其中不含交互，$\|$ 表示并发运算符；

(6) $\mu SCN_1$ 表示服务簇网的迭代运算，即重复执行服务簇网 $SCN_1$ 中的某一项服务，$\mu$ 表示迭代运算符.

以上提出的服务簇网的代数运算都是封闭的运算. 它保证了服务簇网进行代数组合运算之后，仍然是一个服务簇网，同静态服务组合一样，仍然可以继续通过代数运算进行组合以构建更多功能更复杂的服务簇网.

### 5.3　服务簇网组合模型

此部分给出服务簇网组合运算的各个模型，服

务簇网组合以后即形成一个新的服务簇网添加到业务模型层中，对应的 $LSN$ 的组合通过合并库所、添加控制变迁和控制库所来完成. 假设 $SCN_i=(NID_i, BI_i, NF_i, NS_i, LSN_i)$，其中 $LSN_i=(P_i, T_i, F_i, I_i, O_i, f_{Si}, f_{Ei})$.

5.3.1 原子服务簇网

原子服务簇网用于当用户所需的服务为单个的原子服务存在于服务簇中时. 此时，服务簇网由一个单个的服务簇组成. 比如，用户要预订机票，只需查找机票预订服务簇，从中选择适合用户的服务，返回给用户.

**定义 14.** 原子服务簇网. 设服务簇 $SC=(CID, BI, CF, CS, CE)$，其中，$CE=(P, t, F, f_I, f_O)$，则 $CAN=(NID, BI, NF, NS, LSN)$ 表示 $SC$ 对应的原子服务簇网，其中：

(1) $NID$ 表示该原子服务簇网的标识；

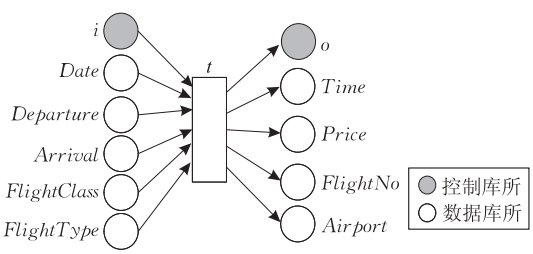
(2) $BI$ 表示原子服务簇网的基本信息，包括服务簇网的名称等；

(3) $NF=CF$ 为服务的功能语义描述；

(4) $NS=\{CID\}$，表示原子服务簇网中仅包含一个服务簇 $SC$；

(5) $LSN=(P', T, F', I, O, f_S, f_E)$，表示描述该服务簇的逻辑服务网. 其中，$P'=P\cup\{i, o\}$，$T=\{t\}$，$F'=F\cup\{\langle i, t\rangle, \langle t, o\rangle\}$，$f_S=I(t)=f_I\wedge i$，$f_E=O(t)=f_O\wedge o$.

原子服务簇网是在服务簇元的基础上，通过增加控制库所 $i$、$o$、网输入表达式 $f_S$ 和网输出表达式 $f_E$ 所得到的. 例如，图 3 所示的机票预订服务元的原子服务簇网如图 5 所示. 当用户需要预订机票时调用该原子服务簇网.




$f_I=i\wedge Date\wedge Departure\wedge Arrival\wedge(FlightClass\vee FlightType)$

图 5 机票预订原子服务簇网模型图

原子服务簇网能够根据用户的需求得到正确的输出结果，显然，原子服务簇网都是完备的.

5.3.2 顺序运算

顺序运算应用于两个服务顺序执行的情形. 使用顺序运算进行组合的情况有两种：(1) 要组合的两个服务簇网之间存在着参数依赖关系，前序服务簇网的输出参数是后序服务簇网的输入参数. 例如，在旅游订票系统中，用户先要预订到达目标城市飞机票，然后预订从机场到景点的汽车票，汽车票预订服务需要机票预订服务所提供的目标机场、到达时间等输出参数作为输入参数，如图 6 所示；(2) 要组合的两个服务簇网之间并无参数依赖关系，但存在着逻辑依赖关系，后序服务要在前序服务完成以后才可以开始. 比如，电子商务系统中，客户要先付款之后商家才发送商品，付款服务与发货服务之间并没有参数依赖的关系，但必须要按顺序执行，如图 7 所示.

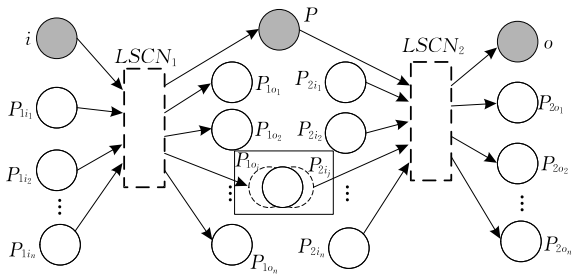



图 6 顺序运算模型-1

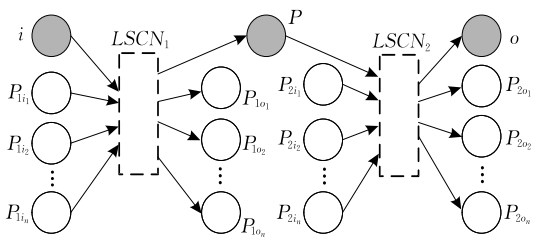



图 7 顺序运算模型-2

组合时应用参数相似度匹配的方法将前序服务簇网的输出参数和后序服务簇网的输入参数进行匹配，合并相匹配的库所，如图 6 中边框表示.

**定义 15.** 顺序运算. $SCN=SCN_1\cdot SCN_2=(NID, BI, NF, NS, LSN)$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行顺序运算，其中：

(1) $NID$ 表示组合得到的新服务簇网的标识；

(2) $BI$ 表示组合得到的新服务簇网的基本信息描述；

(3) $NF$ 表示组合得到的新服务簇网的功能语义描述；

(4) $NS=NS_1\cup NS_2$；

(5) $LSN=(P, T, F, I, O, f_S, f_E)$，其中：

① $P=P_1\cup P_2$，其中，$i=i_1$，$o=o_2$，$p=o_1\Theta i_2$（符号 $\Theta$ 表示库所合并），$P_E=P_{out1}\cap P_{In2}$，$P_{In}=P_{In1}\cup P_{In2}-P_E$，$P_{out}=P_{out1}\cup P_{out2}-P_E$；

② $T=T_1\cup T_2$；

③ $F=F_1\cup F_2$；

④ $I$ 是一个逻辑限制输入函数，使对 $\forall t\in T$：$I(t)=f_I$是一个逻辑输入表达式；

⑤ $O$ 是一个逻辑限制输出函数，使对 $\forall t\in T$：$O(t)=f_O$是一个逻辑输出表达式；

⑥ $f_S$和 $f_E$表示组合网$LSN$ 的逻辑输入表达式和逻辑输出表达式，由算法 1 生成.

下面给出一种顺序运算 $f_S$和 $f_E$的生成算法.

**算法 1.**　顺序运算 $f_S$、$f_E$生成算法.

输入：前序服务簇网输入表达式 $f_{S1}$、输出表达式 $f_{E1}$，
　　后序服务簇网输入表达式 $f_{S2}$、输出表达式 $f_{E2}$

输出：组合服务簇网输入表达式 $f_S$、输出表达式 $f_E$

```
将 f_E1 化为标准析取范式 f_E1 = f_1 ∨ f_2 ∨ … ∨ f_n；
P_E = {P_x1, P_x2, …, P_xm}；
FOR i := 1 TO n
{
  V_Si = f_S2；V_Ei = f_i；
  FOR j := 1 TO m
  {
    IF P_xj | f_i = .T.
          //库所 P_xj 在合取式 f_i 中的出现形式为 .T.
    {
      V_Si = V_Si(P_xj → .T.)；
    //将表达式V_Si 中的库所P_xj 置换为 .T. 后赋值给V_Si
      V_Ei = V_Ei(P_xj → .T.)；
    }
    ELSE
    {
      V_Si = V_Si(P_xj → .F.)；
      V_Ei = V_Ei(P_xj → .F.)；
    }
  }
}
f_S = f_S1 ∧ (V_S1 ∨ V_S2 ∨ … ∨ V_Sn)；
f_E = f_E2 ∧ (V_E1 ∨ V_E2 ∨ … ∨ V_En).
```

由于服务簇网的输出参数是不确定的，因此在算法 1 中，按照前序服务簇网 $SCN_1$ 的每一种输出方式，对输出表达式中的关联库所进行替换，得到 $P_{\text{out1}}-P_E$上的逻辑表达式 $V_{Ei}$，然后将每种输出方式替换得到的 $V_{Ei}$ 进行析取，再合取 $f_{E2}$，得到 $P_{\text{out}}$ 上的逻辑表达式 $f_E$. 同时，按照 $SCN_1$ 的每一种输出方式对 $f_{S2}$ 中的关联库所进行替换，得到 $P_{\text{In2}}-P_E$ 上的逻辑表达式 $V_{Si}$，然后将每种输出方式替换得到的 $V_{Si}$ 进行析取，再合取 $f_{S1}$，得到 $P_{\text{In}}$上的逻辑表达式 $f_S$.

服务簇网的顺序运算中涉及到关联库所的交互，而且参数传递的不确定性受到逻辑表达式的约束. 因此，不能保证通过定义 15 组合得到的服务簇网，能顺利执行并得到正确的输出结果. 下面对顺序组合服务簇网的健壮性保持进行讨论.

**引理 1.**　设逻辑服务网 $LSN_1$，$LSN_2$，$LSN=LSN_1\cdot LSN_2$，$p=o_1\Theta i_2$，在 $LSN$ 中，如果有 $M_0[t_1,t_2,\cdots,t_k\rangle M_e$，$M_e\in M_E$，则必有 $M_0[t_1,t_2,\cdots,t_i\rangle M_p[t_{i+1},\cdots,t_k\rangle M_e$，$M_p$ 定义为标识 $M$：$M(p)=1$，$\forall p'\in P_C-\{p\}$，有 $M(p')=0$.

证明.　$M_p$为顺序组合前 $LSN_1$ 的终止状态和 $LSN_2$ 的初始状态. 在顺序组合中，仅仅是 $LSN_1$ 的输出库所和 $LSN_2$ 的输入库所之间的关联，并没有破坏两个逻辑服务网的内部结构. 因此，$LSN$ 的 $LSN_1$ 部分运行完以后可以到达状态 $M_p$，$LSN_2$ 部分只有在 $M_p$状态下才可以引发到达终止标识 $M_e$.

证毕.

根据引理 1，可以将组合网的健壮性分成两个部分分别讨论. 组合以后，前序部分和后序部分仍然是两个独立的部分，在运行时互不影响. 因此，得到以下结论.

**定理 1.**　设逻辑服务网 $LSN_1$，$LSN_2$，$LSN=LSN_1\cdot LSN_2$，$p=o_1\Theta i_2$，组合网 $LSN$ 是健壮的当且仅当

(1) 逻辑服务网 $LSN_1$是健壮的；

(2) $\forall M_0\in Ms$，$\forall M_p\in R(M_0)$都有 $LSN_2$在 $M_p$ 是健壮的.

证明.　必要性. 若 $LSN_1$ 不是健壮的，则必然存在 $M_{10}\in M_{S1}$，使得 $LSN_1$ 对 $M_{10}$ 不是健壮的，而组合以后，$LSN_2$ 部分对 $LSN_1$ 部分的运行不产生任何影响，因此，必然存在 $M_0\in M_S$，使得 $M_p\notin R(M_0)$，根据引理 1，不存在 $M_e\in M_E$ 使得 $M_e\in R(M_0)$，即 $LSN$ 不是健壮的. 若 $\exists M_p\in R(M_0)$使得 $LSN_2$ 在 $M_p$下不是健壮的，显然 $LSN$ 不是健壮的.

充分性.　$\forall M_0\in Ms$，由于逻辑服务网 $LSN_1$ 是健壮的，因此，$\forall M'\in R(M_0)$，有 $M_p\in R(M')$. 由于 $LSN_2$ 在 $M_p$是健壮的，因此，$\forall M''\in R(M_p)$，$\exists M_e\in M_E$使得 $M_e\in R(M'')$，即 $LSN$ 是健壮的.　　证毕.

定理 1 中的第(1)个条件容易得到. 第(2)个条件需要作深入分析，$LSN_1$ 的输出库所要受到输出表达式的限制，$LSN_2$ 的输入库所要受到输入表达式的限制，因此，需要对表达式进行分析，首先作以下定义.

**定义 16.**　投影映射. 设 $X$、$Y$ 是两个字母表，$Y\subseteq X$. 称映射 $\Gamma_{X\to Y}:X^*\to Y^*$ 为 $X\to Y$ 的一个投影

映射，$\forall\sigma\in X^*$，$\Gamma_{X\to Y}(\sigma)$表示从 $\sigma$ 中删除 $X-Y$ 中的所有字符后的剩余子串.

设逻辑服务网 $LSN$ 的标识 $M$，$\Gamma_{P\to P1}(M)$表示服务网 $LSN$ 的 $LSN_1$部分的对应标识 $\overline{M}$.

**定义 17.** 公平性. 设 $X$、$Y$ 是两个字母表，$X\subseteq Y$，$f_X$、$f_Y$ 分别是 $X$、$Y$ 上的逻辑表达式，$R(f_X)$、$R(f_Y)$表示 $f_X$、$f_Y$ 的标准析取范式的合取式的集合. $\forall f_{X1},f_{X2}\in R(f_X)$，如果存在 $X-Y$ 上的合取式 $f_{(X-Y)1}$，使得 $f_{X1}\wedge f_{(X-Y)1}\in R(f_Y)$，则必有 $f_{X2}\wedge f_{(X-Y)1}\in R(f_Y)$，则称合取式 $f_{X1}$ 和 $f_{X2}$ 在 $R(f_Y)$上是公平的. 如果 $\forall f_{Xi},f_{Xj}\in R(f_X)$，都有 $f_{Xi},f_{Xj}$ 在 $R(f_Y)$上是公平的，则称 $R(f_X)$在 $R(f_Y)$上是公平的.

例如，$f_1=P_1\vee P_2$，$f_2=P_1\vee(P_2\wedge P_3)$，$f_3=P_1\wedge(P_2\vee P_3)$，则有 $R(f_1)=\{P_1\wedge P_2,P_1\wedge\neg P_2,\neg P_1\wedge P_2\}$，$R(f_2)=\{P_1\wedge P_2\wedge P_3,P_1\wedge P_2\wedge\neg P_3,P_1\wedge\neg P_2\wedge P_3,P_1\wedge\neg P_2\wedge\neg P_3,\neg P_1\wedge P_2\wedge P_3\}$，$R(f_3)=\{P_1\wedge P_2\wedge P_3,P_1\wedge P_2\wedge\neg P_3,P_1\wedge\neg P_2\wedge P_3\}$，容易验证，$R(f_1)$在 $R(f_2)$上是公平的. 对合取式 $f_{11}=P_1\wedge P_2$，$f_{12}=\neg P_1\wedge P_2$，$f_{11}\wedge P_3\in R(f_3)$且 $f_{12}\wedge P_3\notin R(f_3)$，因此 $f_{11}$ 和 $f_{12}$ 在 $R(f_3)$上不满足公平关系，$R(f_1)$在 $R(f_3)$上不是公平的.

**定理 2.** 设 $X$、$Y$ 是两个字母表，$X\subseteq Y$，$f_X$、$f_Y$ 分别是 $X$、$Y$ 上的逻辑表达式，$R(f_X)$、$R(f_Y)$表示 $f_X$、$f_Y$ 的标准析取范式的合取式的集合，如果 $\exists Y-X$ 上的表达式 $f_{Y-X}$，使得 $f_Y=f_X\wedge f_{Y-X}$，则 $R(f_X)$在 $R(f_Y)$上是公平的.

证明. 记$f_X$的标准析取范式为 $f_{X1}\vee f_{X2}\vee\cdots\vee f_{Xm}$，$f_{Y-X}$的标准析取范式为 $f_{(Y-X)1}\vee f_{(Y-X)2}\vee\cdots\vee f_{(Y-X)m}$，则 $f_Y=f_X\wedge f_{Y-X}=(f_{X1}\vee f_{X2}\vee\cdots\vee f_{Xm})\wedge(f_{(Y-X)1}\vee f_{(Y-X)2}\vee\cdots\vee f_{(Y-X)m})=(f_{X1}\wedge f_{(Y-X)1})\vee(f_{X1}\wedge f_{(Y-X)2})\vee\cdots\vee(f_{X1}\wedge f_{(Y-X)m})\vee(f_{X2}\wedge f_{(Y-X)1})\vee(f_{X2}\wedge f_{(Y-X)2})\vee\cdots\vee(f_{X2}\wedge f_{(Y-X)m})\vee\cdots\vee(f_{Xm}\wedge f_{(Y-X)1})\vee(f_{Xm}\wedge f_{(Y-X)2})\vee\cdots\vee(f_{Xm}\wedge f_{(Y-X)m})$. 因此，对于任意的 $f_{Xi}\in R(f_X)$，$f_{(Y-X)j}\in R(f_{Y-X})$，都有$f_{Xi}\wedge f_{(Y-X)j}\in R(f_Y)$. 因此可以得到，$\forall f_{Xi},f_{Xj}\in R(f_X)$，如果 $\exists f_{Yi}\in R(f_Y)$使得 $f_{Yi}=f_{Xi}\wedge f$，则必有 $f\in R(f_{Y-X})$，而对任意的 $f_{(Y-X)i}\in R(f_{Y-X})$，都有 $f_{Xj}\wedge f_{(Y-X)i}\in R(f_Y)$，于是，$f_{Xj}\wedge f\in R(f_Y)$. 因此，对于 $\forall f_{Xi},f_{Xj}\in R(f_X)$，都有 $f_{Xi},f_{Xj}$ 在 $R(f_Y)$上是公平的，即 $R(f_X)$在 $R(f_Y)$上是公平的. 证毕.

有了投影和公平性的概念，可以对定理 1 做进一步研究，得到健壮性继承的充分必要条件.

**定理 3.** 设$LSN_1$，$LSN_2$，$LSN=LSN_1\cdot LSN_2$，$p=o_1\Theta i_2$，$P_E=P_{\text{out1}}\cap P_{\text{In2}}$，$f'_{E1}=\Gamma_{P\text{out1}\to PE}(f_{E1})$，$f'_{S2}=\Gamma_{P\text{In2}\to PE}(f_{S2})$，则 $LSN$ 是健壮的，当且仅当

(1) $LSN_1$是健壮的；

(2) $R(f'_{E1})\subseteq R(f'_{S2})$；

(3) $R(f'_{E1})$在 $R(f_{S2})$上是公平的；

(4) $\forall f_i\in R(f_{S2})$，如果 $\exists f_j\in R(f'_{E1})$，使得 $f_i=f_j\wedge f_k$，$f_k$是 $P_{\text{In2}}-P_{\text{Out1}}$上的合取式，则 $LSN_2$ 关于 $f_i$对应的初始标识是健壮的.

证明. 必要性. 条件(1)的证明已在定理 1 中给出，下面反证(2)、(3)、(4)：(2) 若$R(f'_{E1})\not\subseteq R(f'_{S2})$，则存在 $f\in R(f_{E1})$，$\Gamma_{P\text{out1}\to PE}(f)\notin R(f'_{S2})$，因此，若 $LSN_1$引发后达到 $f$ 对应的终止标识 $M_e$，无论 $P_{\text{In2}}$中的库所的标识如何都不能满足$f_{S2}$，$LSN_2$ 不能引发，因此，在此种情况下，$LSN$ 不是健壮的；(3) 若 $R(f'_{E1})$在 $R(f_{S2})$上不是公平的，则存在$f_1$，$f_2\in R(f'_{E1})$，$f_3\in R(f_{S2})$，使得 $f_3=f_1\wedge f_4$，而$f_2\wedge f_4\notin R(f_{S2})$，其中 $f_4$ 是库所集 $P_{\text{In2}}$ 上的表达式. $LSN_1$中存在初始标识 $M_{10}$ 和路径 $\sigma$，使得 $M_{10}[\sigma\rangle M_{1e}$，$\Gamma_{P1\to PE}(M_{1e})=M_{f_2}$. 根据算法 1，存在 $M_0\in M_S$，$\Gamma_{P\to P1}(M_0)=M_{10}$，$\Gamma_{P\to P\text{In2}}(M_0)=M_{f_4}$，$M_0[\sigma\rangle M_p$，$\Gamma_{P\to P\text{In2}}(M_p)=M_{f_2\wedge f_4}$，由于 $f_2\wedge f_4\notin R(f_{S2})$，$LSN_2$不能引发，$LSN$ 在标识 $M_0$ 下不是健壮的；(4) 设$f_i\in R(f_{S2})$，$f_j\in R(f'_{E1})$，且 $f_i=f_j\wedge f_k$，$f_k$是 $P_{\text{In2}}-P_{\text{Out1}}$上的合取式，结合算法 1，存在 $M_0\in M_S$和路径 $\sigma$，$M_0[\sigma\rangle M_p$，$\Gamma_{P\to P\text{In2}}(M_p)=M_{f_i}$，假设 $LSN_2$关于 $M_p$不是健壮的，则 $LSN$ 关于标识 $M_0$ 也不是健壮的.

充分性. $\forall M_0\in M_S$，由于 $LSN_1$是健壮的，因此，对于任意的路径 $\sigma$，$M_p$ 都是可达的. 由于 $R(f'_{E1})\subseteq R(f'_{S2})$且 $R(f'_{E1})$在 $R(f_{S2})$上是公平的，对于任意可达的 $M_p$ 都有 $M_p$ 满足 $f_{S2}$，因此 $LSN_2$ 部分一定可以引发，根据条件(4)可以得到，对 $M_0$ 可达的 $M_p$，都有 $LSN_2$关于 $M_p$ 是健壮的，因此，对 $\forall M_0\in M_S$，$\forall M\in R(M_0)$，都存在 $M'\in R(M)$且 $M'\in M_E$，因此 $LSN$ 是健壮的. 证毕.

定理 3 给出了顺序组合之后的逻辑服务网健壮的充分必要条件，通过该定理可以对服务簇网组合之后的完备性进行验证，得到下面的推论.

**推论 1.** 设服务簇网$SCN_1$，$SCN_2$，$SCN=SCN_1\cdot SCN_2$，$f'_{E1}=\Gamma_{P\text{out1}\to PE}(f_{E1})$，$f'_{S2}=\Gamma_{P\text{In2}\to PE}(f_{S2})$，如果 $SCN_1$、$SCN_2$ 都是完备的，$R(f'_{E1})\subseteq R(f'_{S2})$，且 $R(f'_{E1})$在 $R(f_{S2})$上是公平的，则服务簇网 $SCN$ 是完备的.

证明. 定理 3 中的条件(1)、(2)、(3)显然都满

足，如果 $LSN_2$ 都是健壮的，则对任意的 $f_i \in R(f_{S2})$，$LSN_2$ 关于 $f_i$ 对应的初始标识都是健壮的. 因此条件(4)也满足，$LSN$ 是健壮的，因此，$SCN$ 是完备的.　　证毕.

推论 1 给出了服务簇网顺序组合之后，完备性继承的一个充分条件. 如果两个服务簇网都是完备的，且其 $LSN$ 的逻辑输入、输出表达式满足定理 3 中的条件(2)、(3)，则顺序组合之后的服务簇网仍然是完备的.

**推论 2.**　设服务簇网 $SCN_1, SCN_2, SCN = SCN_1 \cdot SCN_2$，$f'_{E1} = \Gamma_{P\mathrm{out1} \to PE}(f_{E1})$，如果 $SCN_1$ 和 $SCN_2$ 都是完备的，且存在 $P_{\mathrm{In2}} - P_{\mathrm{Out1}}$ 上的表达式 $f$，使得 $f_{S2} = f'_{E1} \wedge f$，则 $SCN$ 是完备的.

证明.　如果存在 $P_{\mathrm{In2}} - P_{\mathrm{Out1}}$ 上的表达式 $f$，使得 $f_{S2} = f'_{E1} \wedge f$，则 $f'_{S2} = \Gamma_{P\mathrm{In2} \to PE}(f_{S2}) = \Gamma_{P\mathrm{In2} \to PE}(f'_{E1} \wedge f) = \Gamma_{P\mathrm{In2} \to PE}(f'_{E1}) \wedge \Gamma_{P\mathrm{In2} \to PE}(f)$，由于 $f'_{E1}$ 是 $P_E$ 上的表达式，因此再做投影时不变，即 $\Gamma_{P\mathrm{In2} \to PE}(f'_{E1}) = f'_{E1}$，$f$ 是 $P_E$ 上的表达式，因此往集合 $P_E$ 上做投影时为空表达式，因此 $f'_{E1} = f'_{S2}$，显然 $R(f'_{E1}) \subseteq R(f'_{S2})$. 由定理 2 知，$R(f'_{E1})$ 在 $R(f_{S2})$ 上是公平的，结合推论 1 可以得到，$SCN$ 是完备的.　　证毕.

由推论 2 知，如果两个服务簇网都是完备的，且前序 $LSN$ 的逻辑输出表达式在关联库所集上做投影后，是后序 $LSN$ 的逻辑输入表达式的前序表达式，则顺序组合之后的服务簇网仍然是完备的.

5.3.3　选择运算

当两种功能的服务都可以满足用户需求，用户需要选择其中的一种服务时，需要进行选择运算. 例如，在旅游订票系统中，用户既可以乘坐飞机到达目的地，也可以乘坐火车到达目的地. 此时，需要对机票预订和火车票预订的服务簇网进行选择运算组合，选择运算模型如图 8 所示.

**定义 18.**　选择运算. $SCN = SCN_1 \oplus SCN_2 = (NID, BI, NF, NS, LSN)$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行选择运算，其中：

(1) $NID$ 表示组合得到的新服务簇网的标识；

(2) $BI$ 表示组合得到的新服务簇网的基本信息描述；

(3) $NF$ 表示组合得到的新服务簇网的功能语义描述；

(4) $NS = NS_1 \cup NS_2$；

(5) $LSN = (P, T, F, I, O, f_S, f_E)$，其中：

① $P = P_1 \cup P_2$，其中，$i = i_1 \Theta i_2$，$o = o_1 \Theta o_2$，$P_{\mathrm{In}} = P_{\mathrm{In1}} \cup P_{\mathrm{In2}}$，$P_{\mathrm{out}} = P_{\mathrm{out1}} \cup P_{\mathrm{out2}}$；

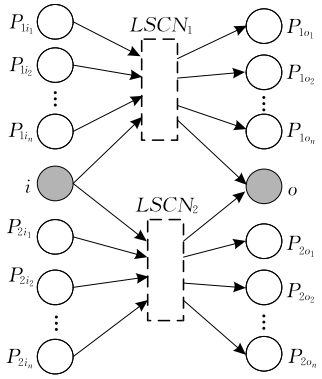



图 8　选择运算模型

② $T = T_1 \cup T_2$；

③ $F = F_1 \cup F_2$；

④ $I$ 是一个逻辑限制输入函数，使对 $\forall t \in T$：$I(t) = f_I$ 是一个逻辑输入表达式；

⑤ $O$ 是一个逻辑限制输出函数，使对 $\forall t \in T$：$O(t) = f_O$ 是一个逻辑输出表达式；

⑥ $f_S = f_{S1} \vee f_{S2}$；

⑦ $f_E = f_{E1} \vee^0 f_{E2}$，符号 $\vee^0$ 表示一种特殊的或运算，当且仅当 $f_{E1}$ 为真且 $f_{E2}$ 中未在 $f_{E1}$ 中出现的库所的真值都为假或 $f_{E2}$ 为真且 $f_{E1}$ 中未在 $f_{E2}$ 中出现的库所的真值都为假时，$f_E$ 为真.

通过选择运算的定义描述，可以看出，选择运算中不涉及关联库所的交互，容易得到以下结论.

**定理 4.**　设服务簇网 $SCN_1, SCN_2, SCN = SCN_1 \oplus SCN_2$，则 $SCN$ 是完备的，当且仅当 $SCN_1$ 和 $SCN_2$ 都是完备的.

证明.　充分性. 如果 $SCN_1$ 和 $SCN_2$ 都是完备的，则 $LSN_1$ 和 $LSN_2$ 都是健壮的. 由于 $f_S = f_{S1} \vee f_{S2}$，因此，$\forall M_0 \in Ms$ 必然有 $LSN_1$ 或 $LSN_2$ 可以执行，不妨设 $LSN_1$ 执行，由于 $LSN_1$ 是健壮的，因此一定能够到达终止状态 $M_e$，使 $f_{E1}$ 为真. 由于 $LSN_2$ 不能引发，因此，$\forall p \in P_{\mathrm{Out2}} - P_{\mathrm{Out1}}$，$M_e(p) = 0$，所以有 $f_E | M_e = .T.$，且 $\forall p \in P_C - \{o\}$，$M(p) = 0$，$M(o) = 1$，因此 $LSN$ 是健壮的，即 $SCN$ 是完备的.

必要性反证. 如果其中任意一个子服务簇网不是完备的，不妨设 $SCN_1$ 不完备，则在 $LSN_1$ 中，$\exists M_{01} \in Ms_1$，使得 $LSN_1$ 在 $M_{01}$ 不是健壮的. 由于 $f_S = f_{S1} \vee f_{S2}$，因此，存在 $M_0 \in Ms$，$\forall p \in P_1$，$M_0(p) = M_{01}(p)$ 且 $f_{S2} | M_0 = .F.$，在 $M_0$ 只有 $LSN_1$ 时可以执行，由于 $LSN_1$ 不是健壮的，因此终止状态不可达，即 $LSN$ 不是健壮的，$SCN$ 不是完备的.　　证毕.

5.3.4　任意顺序运算

任意顺序运算表示两个服务进行顺序组合运

算，但运算的顺序是任意的. 比如，在网上购物系统中，付款服务和发货服务进行顺序组合，既可以先付款后发货，也可以先发货后付款. 值得注意的是，参与任意顺序组合的两个服务不能有参数依赖关系.

**定义 19.** 任意顺序运算. $SCN = SCN_1 \diamond SCN_2 = (NID, BI, NF, NS, LSN)$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行任意顺序运算，其中：

(1) $NID$ 表示组合得到的新服务簇网的标识；

(2) $BI$ 表示组合得到的新服务簇网的基本信息描述；

(3) $NF$ 表示组合得到的新服务簇网的功能语义描述；

(4) $NS = NS_1 \cup NS_2$；

(5) $LSN = (P, T, F, I, O, f_S, f_E)$，其中：

① $P = P_1 \cup P_2 \cup \{i, o, p_{1\sim5}\}$，$P_{\text{In}} = P_{\text{In1}} \cup P_{\text{In2}} \cup \{i\} - \{i_1, i_2\}$，$P_{\text{out}} = P_{\text{out1}} \cup P_{\text{out2}} \cup \{o\} - \{o_1, o_2\}$；

② $T = T_1 \cup T_2 \cup \{t_{1\sim6}\}$；

③ $F = F_1 \cup F_2$，新加控制变迁和控制库所的流关系如图 9 所示；

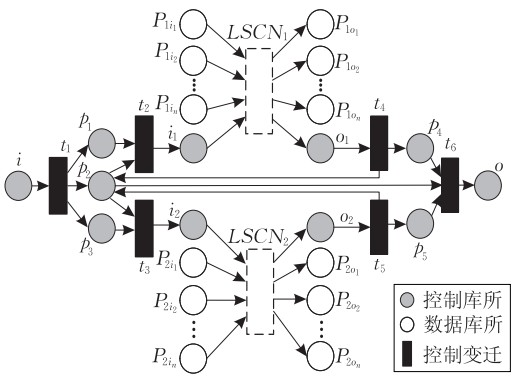



图 9 任意顺序运算模型

④ $I$ 是一个逻辑限制输入函数，使对 $\forall t \in T$：$I(t) = f_I$ 是一个逻辑输入表达式；

⑤ $O$ 是一个逻辑限制输出函数，使对 $\forall t \in T$：$O(t) = f_O$ 是一个逻辑输出表达式；

⑥ $f_S = f_{S1} \wedge f_{S2}(i_1 \rightarrow i, i_2 \rightarrow i)$；

⑦ $f_E = f_{E1} \wedge f_{E2}(o_1 \rightarrow o, o_2 \rightarrow o)$.

分析可以得到任意顺序运算 $SCN = SCN_1 \diamond SCN_2$ 等价于 $(SCN_1 \cdot SCN_2) \oplus (SCN_2 \cdot SCN_1)$，任意顺序运算可以通过调用顺序运算和选择运算得到. 然而，任意顺序运算仍然是有必要的，因为模型 $SCN = SCN_1 \diamond SCN_2$ 相对于 $(SCN_1 \cdot SCN_2) \oplus (SCN_2 \cdot SCN_1)$ 来说，即高效又简洁，尤其是在网系统庞大而复杂的情况下，优势更为突出.

由于参与任意顺序组合的两个服务不能有参数依赖关系，因此，组合过程中不涉及关联库所交互. 组合网能够继承组合子网的完备性.

**定理 5.** 设服务簇网 $SCN_1$，$SCN_2$，$SCN = SCN_1 \diamond SCN_2$，则 $SCN$ 是完备的，当且仅当 $SCN_1$ 和 $SCN_2$ 都是完备的.

证明. 充分性. 服务簇网 $LSN_1$ 和 $LSN_2$ 是任意顺序执行的，不妨设先执行 $LSN_1$ 到达状态 $M_1$，后执行 $LSN_2$ 到达状态 $M_2$. 由于 $SCN_1$ 是完备的，因此，一定有 $f_{E1} \mid M_1 = .T.$，$\forall p \in P_{C1}$，$M_1(p) = 0$，$f_{E2} \mid M_2 = .T.$，$\forall p \in P_{C2}$，$M_2(p) = 0$，$M_2[t_5, t_6\rangle M_e$. 根据图 9 所示模型，容易验证 $M_e(o) = 1$，$\forall p \in \{i, i_{1\sim2}, o_{1\sim2}, p_{1\sim5}\}$，$M_e(p) = 0$，由于 $f_E = f_{E1} \wedge f_{E2}(o_1 \rightarrow o, o_2 \rightarrow o)$，所以 $f_E \mid M_e = .T.$，因此，$LSN$ 是健壮的，即 $SCN$ 是完备的.

必要性反证. 如果任意一子网不是完备的，不妨设 $SCN_1$ 不完备，则 $\exists M_0 \in Ms$，$M \in R(M_0)$ 使得 $\forall M' \in R(M)$，都有 $\Gamma_{P \rightarrow P1}(M') \notin M_{e1}$，因此，$M' \notin M_e$，即 $SCN$ 不是完备的. 证毕.

根据定理 5，如果服务簇网之间没有参数依赖的关系，则它们通过任意顺序运算组合之后是完备的当且仅当这两个服务簇网都是完备的.

### 5.3.5 并发运算

如果完成用户的请求，需要调用两个服务，且这两个服务之间互不影响，不存在依赖关系，此时，需要运用并发运算对这两个服务簇网进行组合. 并发运算用于描述两个服务簇网处于并发的关系，且运行时不存在交互的情形. 比如，旅游订票系统中，用户预订汽车票和预订旅馆的服务互不影响，不存在依赖关系，此时，需要运用并发运算将两种功能的服务簇网组合起来，并发运算模型如图 10 所示.

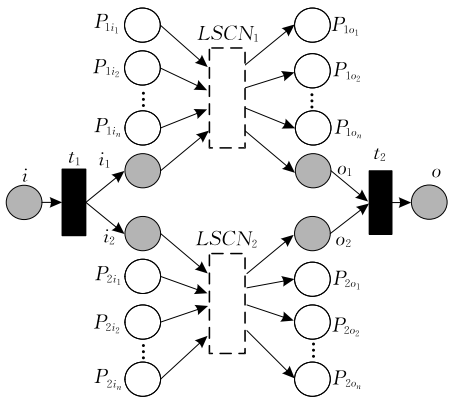



图 10 并发运算模型

**定义 20.** 并发运算. $SCN_1 \| SCN_2 = (NID, BI, NF, NS, LSN)$ 表示服务簇网 $SCN_1$ 和 $SCN_2$ 进行并发运算，其中：

(1) $NID$ 表示组合得到的新服务簇网的标识；

(2) $BI$ 表示组合得到的新服务簇网的基本信息描述；

(3) $NF$ 表示组合得到的新服务簇网的功能语义描述；

(4) $NS=NS_1\cup NS_2$；

(5) $LSN=(P,T,F,I,O,f_S,f_E)$，其中：

① $P=P_1\cup P_2\cup\{i,o\}$，其中，$P_{\mathrm{In}}=P_{\mathrm{In1}}\cup P_{\mathrm{In2}}\cup\{i\}-\{i_1,i_2\}$，$P_{\mathrm{out}}=P_{\mathrm{out1}}\cup P_{\mathrm{out2}}\cup\{o\}-\{o_1,o_2\}$；

② $T=T_1\cup T_2\cup\{t_1,t_2\}$；

③ $F=F_1\cup F_2\cup\{\langle i,t_1\rangle,\langle t_1,i_1\rangle,\langle t_1,i_2\rangle,\langle o_1,t_2\rangle,\langle o_2,t_2\rangle,\langle t_2,o\rangle\}$；

④ $I$ 是一个逻辑限制输入函数，使对 $\forall t\in T$：$I(t)=f_I$是一个逻辑输入表达式；

⑤ $O$ 是一个逻辑限制输出函数，使对 $\forall t\in T$：$O(t)=f_O$是一个逻辑输出表达式；

⑥ $f_S=f_{S1}\wedge f_{S2}(i_1\rightarrow i,i_2\rightarrow i)$；

⑦ $f_E=f_{E1}\wedge f_{E2}(o_1\rightarrow o,o_2\rightarrow o)$.

由于参与并发运算的两个服务簇网之间互不影响，不存在参数依赖关系，服务网运行时不存在参数交互，因此，在并发运算模型中，组合服务簇网仍然可以继承子网的完备性.

**定理 6.**　设服务簇网 $SCN_1,SCN_2,SCN=SCN_1\|SCN_2$，则 $SCN$ 是完备的，当且仅当 $SCN_1$ 和 $SCN_2$ 都是完备的.

证明.　充分性. 设 $SCN_1$ 和 $SCN_2$ 都是完备的，则 $LSN_1$ 和 $LSN_2$ 都是健壮的，$\forall M_0\in M_s$，有 $f_S|M_0=.T..$ 假设 $M_0[t_1\rangle M_1$，则 $M_1(i_1)=1$，$M_1(i_2)=1$，因为 $f_S=f_{S1}\wedge f_{S2}(i_1\rightarrow i,i_2\rightarrow i)$，所以 $\Gamma_{P\rightarrow P1}(M_1)\in M_{s1}$，$\Gamma_{P\rightarrow P2}(M_1)\in M_{s2}$，$LSN_1$ 和 $LSN_2$ 都是健壮的，因此，$\forall M\in R(M_1)$，都有存在 $M'\in R(M)$，且 $\Gamma_{P\rightarrow P1}(M')\in M_{E1}$，$\Gamma_{P\rightarrow P2}(M')\in M_{E2}$，假设 $M'[t_2\rangle M_e$，由于 $f_E=f_{E1}\wedge f_{E2}(o_1\rightarrow o,o_2\rightarrow o)$，因此 $f_E\mid M_e=.T.$，且 $\forall p\in P_{C1}\cup P_{C2}\cup\{i\}$，有 $M_e(p)=0$，因此，$LSN$ 是健壮的，即服务簇网 $SCN$ 是完备的.

必要性反证. 如果任意一子网不是完备的，不妨设 $SCN_1$ 不完备，则 $\exists M_0\in M_s$，使得 $M_0[t_1\rangle M_1$，$LSN_1$ 对标识 $\Gamma_{P\rightarrow P2}(M_1)$ 不是健壮的，因此，$\exists M\in R(M_1)$，使得 $\forall M'\in R(M)$，都有 $\Gamma_{P\rightarrow P1}(M')\notin M_{E1}$，由于 $f_E=f_{E1}\wedge f_{E2}(o_1\rightarrow o,o_2\rightarrow o)$，因此 $LSN$ 的终止标识不可达，即 $SCN$ 不是完备的.　　证毕.

本小节只给出了不含交互的并发运算的组合模型及其完备性分析，实际系统中存在含交互的并发运算的组合，可以先对两个服务簇网进行拆分，然后通过多次顺序运算和并发运算进行组合.

5.3.6　迭代运算

迭代运算用于描述重复执行一个服务操作的情形. 例如，旅客调用宾馆预订服务时，如果旅客对所订宾馆并不满意，需要再次调用宾馆预订服务重新预订，直到找到旅客满意的宾馆为止，迭代运算模型如图 11 所示.

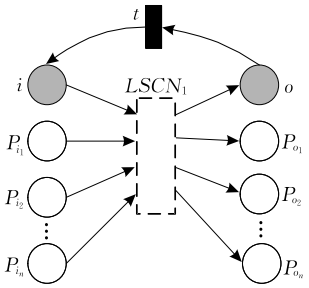



图 11　迭代运算模型

**定义 21.**　迭代运算. $SCN=\mu SCN_1=(NID,BI,NF,NS,LSN)$表示对服务簇网 $SCN_1$ 进行迭代运算，其中：

(1) $NID$ 表示组合得到的新服务簇网的标识；

(2) $BI$ 表示组合得到的新服务簇网的基本信息描述；

(3) $NF$ 表示组合得到的新服务簇网的功能语义描述；

(4) $NS=NS_1$；

(5) $LSN=(P,T,F,I,O,f_S,f_E)$，其中：

① $P=P_1$，其中，$i=i_1$，$o=o_1$，$P_{\mathrm{In}}=P_{\mathrm{In1}}$，$P_{\mathrm{out}}=P_{\mathrm{out1}}$；

② $T=T_1\cup\{t\}$；

③ $F=F_1\cup\{\langle o,t\rangle,\langle t,i\rangle\}$；

④ $I$ 是一个逻辑限制输入函数，使对 $\forall t\in T$：$I(t)=f_I$是一个逻辑输入表达式；

⑤ $O$ 是一个逻辑限制输出函数，使对 $\forall t\in T$：$O(t)=f_O$是一个逻辑输出表达式；

⑥ $f_S=f_{S1}$；

⑦ $f_E=f_{E1}$.

迭代运算在服务簇网中添加连接终控制库所 $o$ 和源控制库所 $i$ 的控制变迁 $t$，使得服务簇网由终止状态返回到初始状态，实现重复操作.

由于服务簇网通过迭代运算组合之后，其逻辑服务网的内部结构并没有改变，重复执行的逻辑服务网能否到达终止标识依赖于网再次执行前的初始标识，因此，将迭代运算的完备性定义为一次执行的完备性，可以将迭代运算视为同一服务簇网的顺序组合，以下结论显然成立.

**定理 7.**　设服务簇网 $SCN_1,SCN=\mu SCN_1$，

则 $SCN$ 是完备的，当且仅当 $SCN_1$ 是完备的.

本节只讨论了服务簇网组合的顺序运算模型、选择运算模型、任意顺序运算模型、并发运算模型和迭代运算模型，并对其组合之后的完备性进行了分析. 更多复杂的组合结构，都可以通过以上 5 种组合模型得到. 对于多个服务簇网组合，可以通过不断地两两组合得到. 显然，以上 5 种代数运算对于服务簇网是封闭的.

### 5.4 代数性质

逻辑 Petri 网模型可以对服务簇网进行形式化建模、分析和验证. 而服务代数在服务簇网组合的结构层面上，对服务簇网的组合进行描述. 它可以清楚地刻画服务簇网组合的结构，对于代数性质的分析可以简化服务簇网系统规模，有效降低系统的复杂度，它对复杂结构的分析、修改和简化起着重要的作用. 表 1 中给出了组合运算的代数性质.

**表 1 组合运算的代数性质**

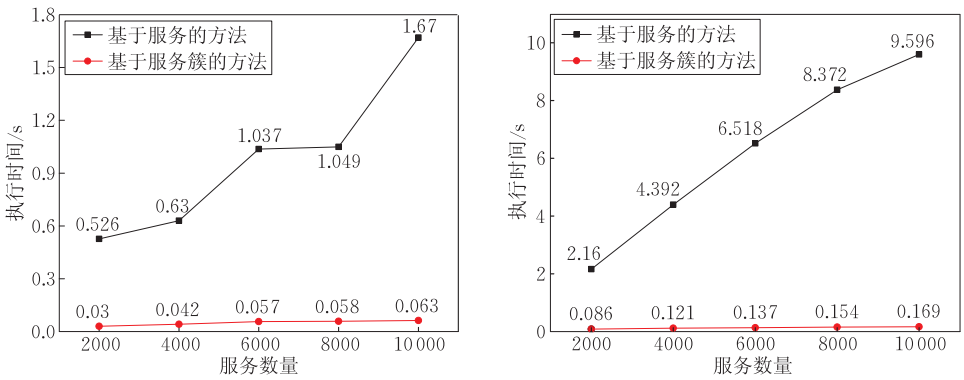
| | |
|---|---|
| 交换律 | |
| $SCN_1 \oplus SCN_2 = SCN_2 \oplus SCN_1$ | (1) |
| $SCN_1 \diamond SCN_2 = SCN_2 \diamond SCN_1$ | (2) |
| $SCN_1 \parallel SCN_2 = SCN_2 \parallel SCN_1$ | (3) |
| 结合律 | |
| $(SCN_1 \cdot SCN_2) \cdot SCN_3 = SCN_1 \cdot (SCN_2 \cdot SCN_3)$ | (4) |
| $(SCN_1 \oplus SCN_2) \oplus SCN_3 = SCN_1 \oplus (SCN_2 \oplus SCN_3)$ | (5) |
| $(SCN_1 \parallel SCN_2) \parallel SCN_3 = SCN_1 \parallel (SCN_2 \parallel SCN_3)$ | (6) |
| 分配率 | |
| $SCN_1 \cdot (SCN_2 \oplus SCN_3) = (SCN_1 \cdot SCN_2) \oplus (SCN_1 \cdot SCN_3)$ | (7) |
| $(SCN_1 \oplus SCN_2) \cdot SCN_3 = (SCN_1 \cdot SCN_3) \oplus (SCN_2 \cdot SCN_3)$ | (8) |
| $SCN_1 \parallel (SCN_2 \oplus SCN_3) = (SCN_1 \parallel SCN_2) \oplus (SCN_1 \parallel SCN_3)$ | (9) |
| $(SCN_1 \oplus SCN_2) \parallel SCN_3 = (SCN_1 \parallel SCN_3) \oplus (SCN_2 \parallel SCN_3)$ | (10) |
| $SCN_1 \diamond (SCN_2 \oplus SCN_3) = (SCN_1 \diamond SCN_2) \oplus (SCN_1 \diamond SCN_3)$ | (11) |
| $(SCN_1 \oplus SCN_2) \diamond SCN_3 = (SCN_1 \diamond SCN_3) \oplus (SCN_2 \diamond SCN_3)$ | (12) |
| $SCN_1 \cdot (SCN_2 \parallel SCN_3) = (SCN_1 \cdot SCN_2) \parallel (SCN_1 \cdot SCN_3)$ | (13) |
| $(SCN_1 \parallel SCN_2) \cdot SCN_3 = (SCN_1 \cdot SCN_3) \parallel (SCN_2 \cdot SCN_3)$ | (14) |
| 其他 | |
| $SCN \oplus SCN = SCN$ | (15) |
| $SCN_1 \diamond SCN_2 = (SCN_1 \cdot SCN_2) \oplus (SCN_2 \cdot SCN_1)$ | (16) |
| $SCN \diamond SCN = SCN \cdot SCN$ | (17) |

## 6 实验分析

文献[16]提出了一种基于前向链的服务组合方法. 该方法首先将用户可提供的输入参数集合作为可达参数集，搜索基于该可达参数集可用的服务，然后将这些可用服务产生的输出参数加入到可达参数集中，继续搜索可用服务，直到可达参数集包含用户需要的所有输出参数时算法终止，表示已经得到用户需要的组合服务. 本节根据该前向链方法设计基于服务簇的前向链算法. 服务组合时，首先应用前向链算法对服务簇进行组合，然后从各服务簇中选择最优服务进行绑定. 本节将以前向链算法为例，对基于单个服务和基于服务簇的服务组合方法进行比较，以证明基于服务簇的服务组合方法的可行性和有效性.

实验首先构造本体概念树，以产生实验所需要的服务和参数. 通过本体概念树构造 20 个服务簇，每个服务簇中包含 100～500 个服务，先将每个服务包含的输入参数和输出参数数量设为 3～8 个，然后将其扩展到 5～15 个. 分别用基于单个服务和基于服务簇的服务组合方法对两组测试集进行实验，对两种方法的算法执行时间进行比较，实验结果如图 12 所示.

图 12(a)为服务参数数量为 3～8 个时两种服务组合方法的执行时间. 通过实验结果可以看出，基于服务簇的组合方法的执行效率是基于单个服务的组合方法的 15 倍. 两种方法都随着服务数量的增多呈近似线性增长的趋势. 当服务数量增加时，基于单个服务的组合方法的执行时间增长较快. 图 12(b)为服务参数数量为 5～15 个时两种算法的执行时间. 随着服务数量的增长，基于单个服务的算法需要耗费大量时间，且执行时间增长较快.




(a) 服务参数数量为3～8个时的执行时间 (b) 服务参数数量为5～15个时的执行时间

图 12 执行时间对比实验结果

## 7 总　结

本文分析了现有的静态服务组合和动态服务组合的缺陷，提出了在业务模型层面上对服务簇进行组合的思想. 用逻辑 Petri 网对服务簇和服务簇网进行了图形化描述，提出了服务簇网的顺序、选择、任意顺序、并发和迭代 5 种基本组合方法，并分别对服务簇网的完备性继承做了分析. 由于顺序运算模型涉及关联库所的不确定性交互，因此重点分析了顺序运算的完备性继承条件. 给出了服务簇网组合的代数运算性质. 最后，通过实验证明了服务簇网组合运算的可行性和有效性.

本文对于服务簇网组合的研究还处于初始阶段，还需要不断完善，下一步将要研究服务请求的处理、业务模型层上服务簇网的发现、服务簇网动态组合算法以及服务簇网到具体服务的绑定等问题，并建立一套完整的服务簇网建模、组合、优化、分析验证工具.

## 参　考　文　献


[1] Nayak R, Lee B. Web service discovery with additional semantics and clustering//Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. Washington, America, 2007: 555-558

[2] Sudha R, Thamarai S. Semantic grid service discovery approach using clustering of service ontologies//Proceedings of the IEEE Technology Conference in the Asia-Pacific Region 2006. Hong Kong, China, 2006: 1-4

[3] Sun Ping, Jiang Chang-Jun. Web service discovery based on ontology and similarity of works. Chinese Journal of Computers, 2008, 31(8): 1340-1353(in Chinese)
(孙萍，蒋昌俊. 利用服务聚类优化面向过程模型的语义 Web 服务发现. 计算机学报，2008，31(8)：1340-1353)

[4] Du Yu-Yue, Qi Liang, Zhou Meng-Chu. A vector matching method for analyzing logic Petri nets. Enterprise Information Systems, 2011, 5(4): 449-468

[5] Du Yu-Yue, Jiang Chang-Jun. On the design and temporal Petri net verification of grid commerce architecture. Chinese Journal of Electronics, 2008, 17(2): 247-251

[6] Feller A L, Wu T, Shunk D L, et al. Petri net translation patterns for the analysis of ebusiness collaboration messaging protocols. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2009, 39(5): 1022-1034

[7] Du Yu-Yue, Jiang Chang-Jun, Zhou Meng-Chu. Modeling and analysis of real-time cooperative systems using Petri nets. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2007, 37(5): 643-654

[8] Du Yu-Yue, Jiang Chang-Jun, Zhou Meng-Chu. A Petri net-based model for verification of obligations and accountability in cooperative systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2009, 39(2): 299-308

[9] Murata T. Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 1989, 77(4): 541-580

[10] Wu Zhe-Hui. Petri Net Introduction Theory. Beijing: China Machine Press, 2006(in Chinese)
(吴哲辉. Petri 网导论. 北京：机械工业出版社，2006)

[11] Du Yu-Yue, Jiang Chang-Jun. Modeling real-time cooperative systems with workflow nets. Chinese Journal of Computers, 2004, 27(4): 471-481(in Chinese)
(杜玉越，蒋昌俊. 基于工作流网的实时协同系统模拟技术. 计算机学报，2004，27(4)：471-481)

[12] Hamadi R, Benatallah B. A Petri net-based model for web service composition//Proceedings of the 14th Australasian Database Conference. Adelaide, Australian, 2003: 191-200

[13] Xiong Peng-Cheng, Fan Yu-Shun, Zhou Meng-Chu. A Petri net approach to analysis and composition of Web services. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2010, 40(2): 376-387

[14] Guo Yu-Bin, Du Yu-Yue, Xi Jian-Qing. A CP-net model and operation properties for Web service composition. Chinese Journal of Computers, 2006, 29(7): 1067-1074(in Chinese)
(郭玉彬，杜玉越，奚建清. Web 服务组合的有色网模型及运算性质. 计算机学报，2006，29(7)：1067-1074)

[15] Van der Aalst W, Hee K, Massuthe P, et al. Compositional service trees//Franceschinis G, Wolf K eds. Application and Theory of Petri Nets: 30th International Conference on Petri Nets. Berlin: Springer Berlin Heidelberg, 2009: 283-302

[16] Thakkar S, Knoblock C, Ambite J, et al. Dynamically composing Web services from on-line sources//Proceedings of the 8th International Conference on Artificial Intelligence. Edmonton, Canada, 2002: 1-7


**WU Hong-Yue**, born in 1988, M. S. candidate. His research interests include service-oriented computing, Petri net theory and application.

**DU Yu-Yue**, born in 1960, Ph. D., professor, Ph. D. supervisor. His research interests include CSCW, software engineering, formal technology, Petri net theory and application.

**Background**

Service-oriented architecture has evolved to become a promising technology for integration of disparate software components. However, as the number of services on the web rising sharply, service composition is facing enormous pressure and challenges. The existing static service composition can't be capable enough to fulfill the various requirements and dynamic service composition methods show inefficiency to compose service components dynamically according to specific requirement.

This paper integrates the advantages of static composition and dynamic composition methods and proposes a service composition approach based on service cluster paradigm which is formally modeled by logical Petri net. This work is partially supported by the National Natural Science Foundation of China under Grant No. 61170078, National Basic Research Program (973 Program) of China under Grant No. 2010CB328101, the Doctoral Program of Higher Education of the Specialized Research Fund of China under Grant No. 20113718110004, the Shandong University of Science and Technology Research Fund of China under Grant No. 2011KYTD102, and the Basic Research Program of Qingdao City of China under Grant No. 13-1-4-116-jch.