

# 基于 EPN 的学习资源组合研究

高 旻 李 华 吴中福  
(重庆大学计算机学院 重庆 400044)

**摘 要** 提出基本 Petri 网 (EPN) 的资源组合方法, 用于对学习资源和组合流程进行建模和分析。首先对基于 Petri 网的组合方法进行综述, 这些方法推动了资源组合的发展, 但其所描述的资源及变迁规则不适合学习资源。为此提出了基于 EPN 的资源组合, 以解决这一问题。改进 EPN 的变迁规则, 把学习资源封装为 Web 服务, 使用改进后的模型对服务进行建模, 然后提出组合算法根据已知条件和变迁规则组合学习资源。最后, 给出学习资源组合的实例验证方法的可行性。

**关键词** 学习资源组合, Petri Net, Web 服务, 基本 Petri 网

**中图法分类号** TP311 **文献标识码** A

## EPN-based Approach for E-learning Resource Composition

GAO Min LI Hua WU Zhong-fu  
(College of Computer Science, Chongqing University, Chongqing 400044, China)

**Abstract** In e-learning domain, learning resource composition is essential to improve the efficiency of courseware development. This paper presented an EPN-based approach to e-learning resource composition flows modeling and analysis. Till now, most methods are based on high-level Petri nets (HPN), but the resources HPN described are not suitable for learning resource composition. That leads to the inefficiency in modeling and analyzing composition flows. We proposed a EPN-based approach to the issue. We described technique for transition rules representing, algorithm, and workflow, so that resource composition can be automatically carried out. Finally, an example shows the method is suitable to the resource composition.

**Keywords** Petri net, Learning resource composition, Web services, EPN

## 1 引言

近年来, 随着各级精品课和网络课程资源的发布, 网上共享的学习资源越来越多, 自动组合现有资源形成新的个性化学习课件的需求与日俱增。目前对 Web 服务组合技术的研究较多<sup>[1]</sup>, 为借鉴这些研究, 本文将学习资源封装为 Web 服务后进行组合。目前, 服务组合的研究主要包括基于工作流的动态组合、基于线性逻辑的组合、基于 Petri 网及 PI 演算的组合。对服务的描述主要包括面向 Web 服务的框架描述语言 WSADLs (Web Service-oriented Architecture Description Languages)、Web 服务流程语言 WSFL (Web Services Flow Language)、Web 服务业务流程执行语言 BPEL4WS (Business Process Execution Language for Web Services)、Web 服务协作接口 WSCI (Web Service Choreography Interface) 和 Petri 网。基于 Petri 网的服务描述和组合能显式地描述系统的状态和事件 (过程), 有丰富的分析方法, 有利于对服务组合动态性质的分析和测试<sup>[2-4]</sup>。因此, 基于 Petri 网的组合成为研究热点。

很多学者对基于 Petri 网的服务组合进行了初步研究, 例

如, GUO<sup>[2]</sup> 提出基于有色 Petri 网的 Web 服务组合模型, Hamad<sup>[3]</sup> 提出基于 Petri 网对 Web 服务控制流进行建模, Chi<sup>[4]</sup> 提出基于 Petri 网对服务组合元模型进行描述和形式化定义。这些研究推动了服务组合的发展。但目前基于 Petri 网的服务组合强调对服务建模方法、运算性质及性能的分析, 主要采用库所/变迁 (P/T) 系统和高级网系统 HPN (Pr/T、颜色网、随机网等), 而描述事件间基本关系和行为的 EPN 却没有被使用。本文在文献<sup>[1-6]</sup>基础上, 1) 改进 EPN 的变迁规则, 2) 使用改进后的 EPN 模型对学习资源建模, 3) 提出动态资源组合算法。

本文第 2 节讨论基于 Petri 网的组合方法的研究现状和不足; 第 3 节提出改进的 EPN 模型, 对学习资源进行建模; 第 4 节提出组合算法和案例; 最后对全文进行总结。

## 2 研究背景及问题分析

Web 服务是一种松耦合、采用标准协议、统一接口定义方式和通信协议的软件组件, 是自包含、自描述、可以在网络中共享的资源<sup>[1]</sup>。学习资源可以被封装成 Web 服务, 用 Web 服务的标准和语言描述学习资源的属性, 用服务的输入输出

描述学习资源的前驱(先学知识)和后继(后学知识)。服务组合是根据用户需求选择相关服务形成组合流程的过程, Petri网适合处理组合流程<sup>[1-5]</sup>。

## 2.1 Petri Net

Petri网<sup>[5]</sup>是 workflow模型的数学表示形式,可以描述物质资源的库所(P)和变迁(T)表示的行为。有向弧F表示P与T之间的关系。P、T和F的集合形成网Net。库所P有0个或多个Token,Token在P之间传递。Token可以触发变迁T,T被触发后,其输入库所和输出库所中的Token值会随之变化。Net是Petri网的基本结构<sup>[5]</sup>,由(P,T,F)表示,其中 $P \cap T = \emptyset, P \cup T \neq \emptyset$ ,并且 $F \subseteq P \times T \cup T \times P$ 。其形式化描述<sup>[5]</sup>为:

$$Net(P, T, F) \Leftrightarrow P \cap T = \emptyset \wedge P \cup T \neq \emptyset \wedge F \subseteq P \times T \cup T \times P \wedge dom(F) \cup cod(F) = P \cup T \quad (1)$$

其中,  $dom(F) \cup cod(F) = P \cup T \mid dom(F) = \{x \mid \exists y, (x, y) \in F\}, cod(F) = \{y \mid \exists x, (x, y) \in F\}$

Petri网有3种主要类型:库所/变迁网(PN)、高级Petri网(HPN)和EPN。

PN描述分布式系统中资源的动态变化过程,用六元组 $\Sigma = (P, T, F, K, W, M_0)$ 表示。K是Net的容量函数,W是Net的权函数,M是Net的标识, $M_0$ 是初始标识。PN形式化描述为: $PN(P, T, F, K, W, M_0) \Leftrightarrow Net(P, T, F) \wedge K: P \rightarrow N \cup \{\infty\} \wedge W: F \rightarrow N \wedge M_0: P \rightarrow N_0 \wedge \forall p \in P: M_0(p) \leq K(p)$ <sup>[5]</sup>。

高级Petri网是PN网的扩充,用于描述复杂系统。成熟的HPN有Pr/T<sub>-</sub>(predication and transition system),CPN(Colored Petri Net),Hierarchical Petri Net和TPN(Timed Petri Net)等。HPN中一个库所P表示多种资源,一个变迁T元素表示多种变迁。例如Pr/T<sub>-</sub>是9元组 $\Sigma = (P, T, F, D, V, A_P, A_T, A_F, M_0)$ ,其中(P,T,F)是Net,D是 $\Sigma$ 中的有限元素,V是D中的元素集合, $A_P: P \rightarrow \pi, \pi$ 是表示论断的可变集合, $A_T: T \rightarrow f_D, f_D$ 是D中公式的集合, $A_F: F \rightarrow f_S, f_S$ 是D的符号集合, $M_0: P \rightarrow f_S$ 。HPNs比PN更加复杂。

EPN可以描述分布式系统的基本性质

$$EPN = (B, E, F, C) \quad (2)$$

其中,B是条件集合,类似于式(1)中的库所;E是事件集合,类似于式(1)中的变迁;F是B与E中元素的关系集合,由有向弧表示;C是状态集合, $c_m$ 表示初始状态;事件输入输出分别由 $\cdot e$ 和 $e \cdot$ 表示:

$$\cdot e = \{x \mid (x, e) \in F\} \text{ or } \cdot e = F^{-1} \cap (B \times E)$$

$$e \cdot = \{y \mid (e, y) \in F\} \text{ or } e \cdot = F \cap (B \times E)$$

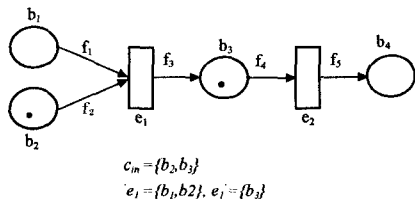


图1 EPN模型

如图1所示,当状态集合c满足 $\cdot e \subseteq c \wedge e \cdot \cap c = \emptyset$ 时,事件e被触发,由 $c[e]$ 进行标识。 $c[e]$ 表示当且仅当e的输入包含于状态集合c,并且e的输出与c没有交集事件时,e可被触发。

## 2.2 基于Petri网的Web服务组合方法和问题分析

目前,已有多项基于Petri网的服务组合的研究。文献[2]提出基于有色Petri网的Web服务组合模型和分析方法。该模型能表示服务之间的逻辑关系。模型中用有色Petri网的变迁表示服务或操作,用库所表示服务输入和输出参数。库所具有容量和当前标记数属性,这些属性值和弧函数控制变迁的发生。

文献[3]提出基于Petri网的方法,用于对Web服务控制流(workflow)进行建模。文中Web服务被描述为基于行为的Petri网模型Moap。Moap能捕捉复杂服务组合中的语义,描述服务之间的动态关系。

文献[4]提出基于Petri建模的服务组合平台,利用P/T<sub>-</sub>网对服务组合元模型进行描述和形式化定义,为服务组合过程的检测和优化提供了基础。

这些研究有效推进了Web服务组合的发展,但仍存在问题。它们都是基于PN或HPN的研究。在这些网系统中,每个库所代表一种或多种物质资源空间,它的容量函数K(s)是容纳s类资源的能力,权函数W(t,s)和W(s,t)分别是变迁t发生时占用和释放资源的数量。变迁的触发规则主要取决于容量函数、权函数以及库所中的Token(资源数目)。但容量函数和权函数对Web服务/学习资源的组合意义不大,还会使变迁的触发规则变得复杂,降低组合的效率<sup>[6]</sup>。而EPN的库所只有“真”与“假”状态,适合描述Web服务资源(学习资源)的“有”和“无”两种状态,因此EPN更适合对服务的建模、组合及分析。

## 3 基于EPN的组合建模

服务资源/学习资源不会因为使用而减少或消失,这与EPN的变迁规则不同,因此本文重新定义EPN的事件变迁规则。

### 3.1 改进的EPN事件变迁规则

定义1(触发与变迁规则,TR) 事件 $e \in E$ 可以被触发,当且仅当状态集合c满足

$$\cdot e \subseteq c \quad (3)$$

标记为 $c[e]$ ,表示c可以使e发生,当且仅当e的输入是状态集合c的子集。

如果 $c[e]$ ,e的输入不会发生改变,但是e的输出变为真,c变为c的后续状态集合 $c'$ 。这个过程被标记为 $c[e]c', c' = c \cup e \cdot$ 。说明当一个事件触发时,事件输入库所的Token不发生变化,而事件输出库所会产生Token。相应地,事件之间关系的定义也随着发生改变。

定义2(事件之间关系的定义) 假设 $e_1, e_2 \in E$ ,事件 $e_1$ 和 $e_2$ 在状态集合c有两种关系:

顺序关系: $e_1$ 可以在c发生,而 $e_2$ 不能,但 $e_1$ 发生后 $e_2$ 可以发生。形式化表示为 $e_1$  and  $e_2$  are in sequence at c iff  $c[e_1] \wedge \neg(c[e_2]) \wedge c'[e_2]$  where  $c[e_1]c'$ 。

并发关系: $e_1$ 和 $e_2$ 在c可以同时发生,没有特别的顺序关系。形式化表示为 $e_1$  and  $e_2$  concurrently occur at c iff  $\cdot e_1 \cup \cdot e_2 \subseteq c$ 。

### 3.2 基于EPN的服务(资源)模型

Web服务可以看作一系列操作的偏序集。因此,可以将其映射到改进的EPN服务组合模型WSC。在该模型中,事

件  $e$  描述 Web 服务,行为  $b$  描述服务的状态。有向箭头  $f$  表示它们之间的关系。WSC 的形式化表示为:

$$WSC = \{B, E, F, C\} + TR \quad (4)$$

其中,  $\{B, E, F\}$  是有向网,包括行为、服务和有向弧;  $C$  是状态集合,详见式(2)。

$B: b \rightarrow \{true, false\}$  是行为的有限集合,表示服务的输入输出,即学习资源的先学和后继学习资源。

$E = \{s_1, s_2, \dots, s_n\}$  是服务(资源)的有限集合。

$F \subseteq \{\langle \cdot s, s \rangle, \langle s, s \cdot \rangle\}$  表示两种有向弧,一种表示从先驱资源到资源,一种表示从资源到后继资源。

TR 决定哪些服务可以在状态集合  $c$  被触发,详见式(3)。

事件  $s \in E$  可以在状态集合  $c$  被触发,当且仅当  $\cdot s \subseteq c$ , 标识为  $c[s]$ 。

如果  $\text{If } c[s]$ , 则  $s$  的输入保持不变,但是输出变为真,状态集合  $c$  将转变为状态集合  $c'$ 。这个过程标识为  $c[s]c'$ , 其中  $c' = c \cup s \cdot$ 。

对服务(学习资源)组合后,可以对组合模型进行分析。例如,可以分析工作流程中学习资源之间的关系,或者对工作流程的活性进行检测。

**定义 3 (服务间的顺序和并行关系)** 设  $s_1, s_2 \in E$  为  $N$  的任意两个服务,顺序关系表示  $s_1$  和  $s_2$  顺序相关,条件丛只能授权  $s_1$  发生,  $s_2$  在其后继丛有发生权;并发关系表示  $s_1$  和  $s_2$  不存在制约关系。以下定义是定义 2 的扩展。

1) 如果  $c[s_1]$ , 但  $\neg c[s_2]$ , 而  $c'[s_2]$ , 其中  $c'$  是  $c$  的后继:  $c[s_1]c'$ , 则  $s_1$  和  $s_2$  在  $c$  有顺序关系。形式化表示为:

$\text{Iff } c[s_1] \wedge \neg c[s_2] \wedge c'[s_2] \wedge c[s_1]c'$  则  $s_1, s_2$  在状态集合  $c$  具有顺序关系, 记作  $s_1 * s_2$ 。

2)  $s_1$  和  $s_2$  在条件丛  $c$  并发的充分必要条件是:  $\cdot s_1 \cup \cdot s_2 \subseteq c$ 。形式化表示为:

$s_1$  和  $s_2$  具有并行关系  $\text{iff } \cdot s_1 \cup \cdot s_2 \subseteq c$ , 记作  $s_1 || s_2$ 。

**定义 4 (WSC 的活性)** WSC 的活性用来表示组合后的服务是否有死锁。

假设  $r = \{(c, c') \mid c, c' \in C \wedge \exists s \in E: c[s]c'\}$ ,  $r^*$  表示一种可达关系,  $r^* = r^0 \cup r^1 \cup r^2 \cup \dots = \bigcup_{i=0}^{\infty} r^i$ 。

如果  $\forall c \in C, \forall s \in E, \exists c' \in C: c r^* c \wedge c'[s]$ , 则 WSC 是活的。

## 4 基于 WSC 的动态组合和案例分析

### 4.1 学习资源模型

在组合前,将一个学习资源封装成一个服务。每个服务只有一个操作,使用 WSC 对服务建模,服务被建模为事件,  $\cdot s$  和  $s \cdot$  分别表示服务的输入输出(代表学习资源的先学和后继资源)。在简单服务(非组合服务)的模型中只存在两种有向弧(见式(4)中的  $F$ )和一个事件  $s$ , 所以服务可以简单地表示为  $s(\cdot s; s \cdot)$ 。

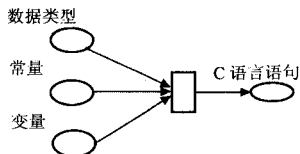


图2 学习资源的 WSC 模型

例如关于“表达式”知识点的学习资源,其先学知识点包

括数据类型、常量、变量,后继知识点为“C 语言语句”,则此资源被封装为服务,描述为表达式(数据类型、常量、变量; C 语言语句)。其 WSC 模型如图 2 所示。

建模后根据用户的输入参数和需求选择服务进行组合。其中,输入参数和需求分布对应输入参数集合和目标参数集合。通过输入参数集合与 WSC 的输入库所匹配判断服务是否可以被触发,可触发的服务产生的输出加入输入参数集合中,这样循环下去,直到用户需求完全包含在目标参数集合中,复合 WSC 的建模成功,资源的组合方案也随之形成。

### 4.2 启发式向前链组合算法

组合中采用正向推理的启发式搜索算法,步骤如下:

1) 为每个候选服务创建 WSC 模型。

2) 把用户的输入作为初始状态集合  $c_{in}$ , 如果  $b \in c_{in}$ , 在  $b$  中标识 Token, 表明它的状态是 true, 将  $c_{in}$  赋值给状态集合  $c$ 。

3) 将  $c_{in}$  和  $\cdot s$  进行匹配, 如果不存在  $\cdot s \subseteq c$ , 则没有服务可以在  $c$  处被执行  $\rightarrow \exists c[s]$ , 组合失败, 算法结束; 如果存在  $c[s]$ , 根据无回溯策略, 添加服务的 WSC 模型到组合 WSC 模型, 通过  $\cdot s$  判断模型之间的连接, 添加  $s \cdot$  到  $c$ , 执行第 4) 步。

4) 用状态集合  $c$  匹配用户的目标集合, 如果目标被满足, 则推理成功, 算法结束, 否则执行第 3) 步。

**算法 NonbacktraceServicesComposition**

UserInputs; 输入集;

UserOutputs; 目标集;

C; 已知状态集;

$c_{in}$ ; 初始状态集;

ServicesSet( $\cdot s; s \cdot$ ); 服务集;

UsedServices; 组合服务集;

NonbacktraceServicesComposition (UserInputs, UserOutputs, ServicesSet)

```
{
  To model every candidate service, form ServicesSet( $\cdot s; s \cdot$ ) to a WSC set;
  Initial  $c_{in} = \text{UserInputs}$ , UsedServices = Null;
  Choose one service  $s$  from WSC set where  $\cdot s \subseteq c_{in}$ ;
  Repeat
  {
    Choose one service  $s$  from WSC set where  $\cdot s \subseteq c_{in}$ ;
    If ( $s$  does not exist) then {Output; Composing is failed; break};
    Else if ( $e \cdot \subseteq c$ ) then delete  $s$  from WSC set;
    Else
    {
      Add WSC model of  $s$  to the composed WSC model, connected them by  $\cdot s$ ;
      Add all  $e \cdot$  to C,  $C = C + e \cdot$ ;
      Add  $s$  into UsedServices;
      Delete  $s$  from WSC set;
    }
  }
  Until UserOutputs  $\subseteq C$ ;
  Output; UsedServices is the service set that meet the user's requirement;
}
```

### 4.3 学习资源组合案例

假设某学生在某课程下有如下知识点(线性表、顺序表、链表和有序表)需要学习。已知他们的前驱后继关系如表 1 所列。假设“数据结构基础”是该学生的已学知识点,“有序

表”是他的目标知识点。

表1 前驱后继关系表

| 依赖关系集 |        |     |
|-------|--------|-----|
| 序号    | 前驱     | 后继  |
| 1     | 数据结构基础 | 线性表 |
| 2     | 线性表    | 顺序表 |
| 3     | 线性表    | 链表  |
| 4     | 顺序表    | 有序表 |
| 5     | 链表     | 有序表 |

这些学习资源可以描述为:

数据结构基础:  $S_0(\emptyset; \text{线性表})$ ;

线性表:  $S_1(\text{数据结构基础}; \text{顺序表}, \text{链表})$ ;

顺序表:  $S_2(\text{线性表}; \text{链表})$ ;

链表:  $S_3(\text{线性表}; \text{有序表})$ ;

有序表:  $S_4(\text{顺序表}, \text{链表}; \emptyset)$ 。

首先构建 WSC 模型(模型略)。根据用户需求,  $UserInputs$  是{数据结构基础},  $UserOutputs$  是{有序表}。与此对应, 初始标识集合为{数据结构基础}, 目标集合为{有序表}。组合步骤如下:

(1) 由于  $c_m = UserInputs = \{\text{数据结构基础}\}$  仅能触发  $S_1$ , 添加  $S_1$  到  $WebServicesUsed$  集合。

(2) 执行  $S_1$ , 顺序表和链表将存在  $token$ , 表明它们的值已经被获得。添加它们到状态集合  $c$ , 并从  $ServicesSet$  移除  $S_1$ 。

(3) 因为  $c$  不能满足用户需求  $UserOutputs: \{\text{有序表}\}$ , 继续寻找哪些服务可以被触发,  $S_2$  和  $S_3$  都可以被触发, 将其加入  $UsedServices$ 。

(4) 执行  $S_2$  和  $S_3$ , 链表和有序表中将有  $token$ 。添加它们到状态集合  $c$ ,  $c$  变为{顺序表, 链表, 有序表}, 可以满足用户需求  $UserOutputs$ , 组合成功。得到组合的结果和学习流程。

最后,  $UsedServices = \{S_1, S_2, S_3\}$ 。图3为组合后的结果, 用户的学习需求可以由  $S_1, S_2$  和  $S_3$  组合所完成。用第3节的方法对组合模型进行分析, 可以检测组合的活性、可达性、服务之间的执行顺序:

该组合是活的, 并不存在死锁, 组合流程中的每一服务都有可能被执行。

对可达性的检测表明目标状态集合{有序表}是可以实现的。

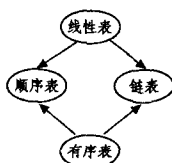


图3 组合结果

$c_m[S_1]$ , 但  $\neg c_m[S_2] \wedge \neg c_m[S_3]$  而  $c[S_2] \wedge c[S_3]$ ,  $c$  是  $c_m$  的后继从  $c_m[S_1] > c$ , 说明  $S_3$  与  $\{S_2, S_3\}$  有顺序关系;  $S_2 \cup S_3 \subseteq c_m$ , 说明  $S_2$  和  $S_3$  之间可以并行执行。最终形成的组合执行方案的数学表达方式为:  $S_3 \cdot (S_1 || S_2)$  ( $\cdot$  表示顺序执行关系,  $||$  表示服务并行执行关系), 图形表达如图3所示。

该实例表明, 我们的建模方法和组合算法是简单、切实可行的。

**结束语** 本文提出基于 EPN 的 WSC 模型对学习资源和组合进行建模并对组合后的模型进行分析的方法。对每个资源建模后, 根据库所的匹配情况进行组合。文中将组合问题转化成经典的 AI 问题, 并通过启发式向前链算法加以解决。最后, 本文通过实例对组合过程进行说明。WSC 模型对学习资源的组合有推动作用, 但是研究还存在后续问题需要解决。例如, 分析组合算法效率, 将时间概念引入变迁规则描述服务和组合的时间性, 将前提、结果和资源质量引入 EPN 的状态集合等。

## 参考文献

- [1] Fu Y, Dong Z, He X. An Approach to Web Services Oriented Modeling and Validation[C]//IW-SOSE, Shanghai, China, 2006
- [2] 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[J]. 计算机学报, 2006, 29(7): 1067-1074
- [3] Hamadi R, Benatallah B. A Petri Net - based Model for Web Service Composition[C]//Conferences in Research and Practice in Information Technology. Adelaide, Australia, Australian Computer Society, 2003
- [4] Chi Y-L, Lee H-M. A formal modeling platform for composing web services[D]. Expert Systems with Applications, 2007
- [5] 袁崇义. Petri 网原理与应用[M]. 北京: 电子工业出版社, 2005
- [6] Gao M, Sun L, Wu Z F, et al. An EPN-based method for web service composition[C]//Proceedings of the 2008 IEEE International Conference on Networking, Architecture, and Storage. IEEE Computer Soc. Los Alamitos, 2008: 163-164
- [7] Aalst W V D. Don't go with the flow: Web service composition standards exposed[J]. IEEE Intelligent System, 2003, 15: 72-85
- [8] Charif Y, Sabouret N. An Overview of Semantic Web service composition Approaches[J]. Electronic Notes in Theoretical Computer Science, 2006, 146: 33-41
- [9] Koehler J, Srivastava B. Web service composition: Current solutions and open problems[C]//Proceedings of the 13th International Conference on Automated Planning & Scheduling. Trento, Italy, 2003
- [10] Lee J, Wu C-L. A Possibilistic Petri-Nets-Based Service Discovery[C]//Proceedings of the IEEE International Conference on Networking, Taipei, Taiwan, Sensing & Control, 2004
- [11] Milanovic N, Malek M. Current solutions for Web service composition[J]. Internet Computing, IEEE, 2004, 8: 51-59
- [12] Murata T. Petri Nets; Properties[C]//Analysis and Applications. Proceedings of the IEEE. 1989: 541-580
- [13] Paolucci M, Kawamura T, Payne T R, et al. Importing the semantic Web in UDDI[C]//Proceedings of Web services, E-business and semantic Web workshop. Toronto, Canada, 2002
- [14] 钱柱中, 陆桑璐, 谢立. 基于 Petri 网的 Web 服务自动组合研究[J]. 计算机学报, 2006, 29(7): 1057-1066