doi:10.3969/j.issn.1000-436x.2014.09.006

基于组合上下文的服务替换方法

王海艳，李思瑞
(南京邮电大学 计算机学院，江苏 南京 210003)

摘 要：随着用户需求的多样性和网络环境的日益复杂性，组合服务的复杂程度越来越高，在对失效服务进行替换时，为了减少被替换服务的冗余信息和提高替换方法的准确性，提出以待替换服务的组合上下文为研究对象，通过以下 2 个步骤完成替换：第一，基于已有的着色 petri 网服务工作流建模方法，提出服务的组合上下文信息采集算法(CCICA, composition context information collection algorithm)，以服务的组合上下文为服务信息采集源；第二，提出基于编辑距离的服务替换方法(LDBSSM, levenshtein distance-based service substitution method)，并将服务接口的依赖关系加入到了算法中。仿真实验表明，该方法不仅较好地减少了被替换服务的信息冗余，而且有效提高了服务替换的准确性，具有更好的实用价值。

关键词：服务替换；组合上下文；着色 petri 网；编辑距离
中图分类号：TP399 **文献标识码**：A **文章编号**：1000-436X(2014)09-0057-10

Service substitution method based on composition context

WANG Hai-yan, LI Si-rui
(College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: With the variety of requirements from service requesters and the increasing complexity of network environments, service composition has become more and more complicated. In order to decrease redundant information of the substituted service and improve the accuracy of service substitution, the composition context was regarded as the matching criteria and tried to substitute failed service with the following two steps. First, a composition context information collecting algorithm (CCICA) was proposed based on the color petri net model of the service workflow, and regarded the composition context as the collecting source of service information. Second, a levenshtein distance-based service substitution method (LDBSSM) was given, which was integrated with dependencies between service interfaces. Simulation results show that the proposed LDBSSM method is practical because it can not only reduce redundancy of substituted service information but also effectively improve the accuracy of service substitution.

Key words: service substitution; composition context; colored petri net; levenshtein distance

1 引言

Web 服务是一种基于网络环境的自适应、自描述、模块化的应用程序，已经成为互联网中最为重要的一种计算资源和软件资产。单个 Web 服务通常无法满足用户的需求，需要组合多个原子服务形成增值的、更大粒度的服务或系统。然而，在组合服务的执行期间，假如服务中某部分由于硬件或软件环境等因素的变化而导致失效，此时服务替换机制需要被调用以及时、有效地对失效部分进行替换，确保服务的正常执行，服务替换对组合服务的可靠执行具有重要意义。

目前的替换方法从替换条件上大致可分为：基于功能、基于操作流程和基于行为接口的方法；从匹配信息采集来源上大致可分为：以失效服务描述模型和以失效服务组合上下文[1,2]的方法；从替换技

收稿日期：2013-06-10；**修回日期**：2013-09-15
基金项目：国家自然科学基金资助项目(61201163)；江苏省自然科学基金资助项目(BK2011072)
Foundation Items: The National Natural Science Foundation of China(61201163); The Natural Science Foundation of Jiangsu Province (BK2011072)

术角度大致可分为：基于语义本体理论的和基于形式化分析工具的方法。然而，为了满足用户需求的多样性以及网络环境的日益复杂性，发布的 Web 服务数量呈爆炸性增长，服务组合的粒度不断增大，其复杂程度也在不断提高，这些给服务的替换研究带来了新的挑战。

目前，研究方法中多以待替换服务为匹配对象[1,3,4,5]，须知服务的复杂度在不断提高，参与组合的服务可能只是这个服务的一部分[6]，此类方法并不能保证服务替换的准确执行；且替换大多以服务的功能相同或相似为替换依据[4,5]，很少考虑接口的匹配及接口的依赖关系[7]。因此普遍存在如下 2 个问题。

1) 失效服务的匹配信息不够准确。由于 Web 服务描述语言 (WSDL, Web service description languag) 是一种静态的描述方法，服务发布商使用其描述服务的功能、操作、输入、输出等信息，多用于解决服务的发现问题。目前并没有一种语言或模型用来记录目标服务在组合服务中的执行信息，因此，在服务发现的研究领域中，基于关键字或者语义本体的匹配方法以静态的服务描述为匹配的输入信息是比较合理的，而当用于解决服务替换的问题时则并不适合，就会出现匹配信息不够准确的问题。

2) 服务可替换性验证方法的准确性可进一步提升。由于服务的接口是存在依赖关系的，简单的查找或从推荐系统给出的一些功能相似的服务集中选择某个所谓最优的服务，而不考虑服务的接口问题，并不能保证可以用来有效地替换失效服务。而且目前针对服务接口匹配的服务可替换性分析方法中遵循的思想实质是“更少的输入得到更多的输出”[5]，这种思想自然合理，然而随着服务数量的激增和服务组合复杂度的提高，这种方法的判断条件就显得比较单一，只可以作为判断的必要条件。

假设图 1 中的 3 个服务 s_2、s_6、s_8 功能相似，3 个服务的输入输出依次为：I_2={*a*,*b*,*e*},O_2={*m*,*n*}; I_6={*a*,*b*,*c*},O_6={*m*,*n*};I_8={*a*,*b*,*e*},O_8={*m*,*n*};图中虚线表明了输出接口对输入接口的依赖，如服务 s_2 的输出 *m* 依赖于输入 *a* 和 *b*；假设服务 s_2 参与某个组合服务的执行，它在组合服务中具有输入 *a*、*b*，输出 *m* 的作用，但已失效，需要用 s_6 或 s_8 对其替换。它们功能相似，需判断接口匹配，传统方法以服务 s_2 所有的接口输入输出信息{*a*,*b*,*e*;*m*,*n*}作为匹配算法的输入，算法的结果会舍弃 s_6 而选择 s_8 进行替换；事实上，由图 1 可知，选择 s_6 比选择 s_8 更合适，因为 s_6 完全满足 s_2 的输入输出需求，而传统匹配算法舍弃 s_6 是因为服务 s_2 的接口信息中存在冗余的输入输出接口信息{*e*;*n*}，选择 s_8 是因为没有考虑服务输出对输入的依赖关系。

本文针对现有服务替换方法中存在的上述 2 个问题展开深入研究，主要贡献如下。

1) 在服务替换研究中，明确提出了以服务组合上下文为研究对象，从中采集失效服务相关信息以解决服务替换问题的思想。

2) 基于已有的着色 petri 网服务工作流模型，提出了一种服务信息采集方法，与目前方法相比，此方法并不依赖静态的 Web 服务描述语言或模型，而是从失效服务组合上下文的角度推算出相关信息，以牺牲较少的算法执行时间来大幅度提高匹配信息的准确性。

3) 提出一种基于编辑距离的服务替换方法，该方法中明确提出并实现将接口依赖问题加入服务可替换验证过程中，以提高方法的准确性，而且实现了接口级的匹配。

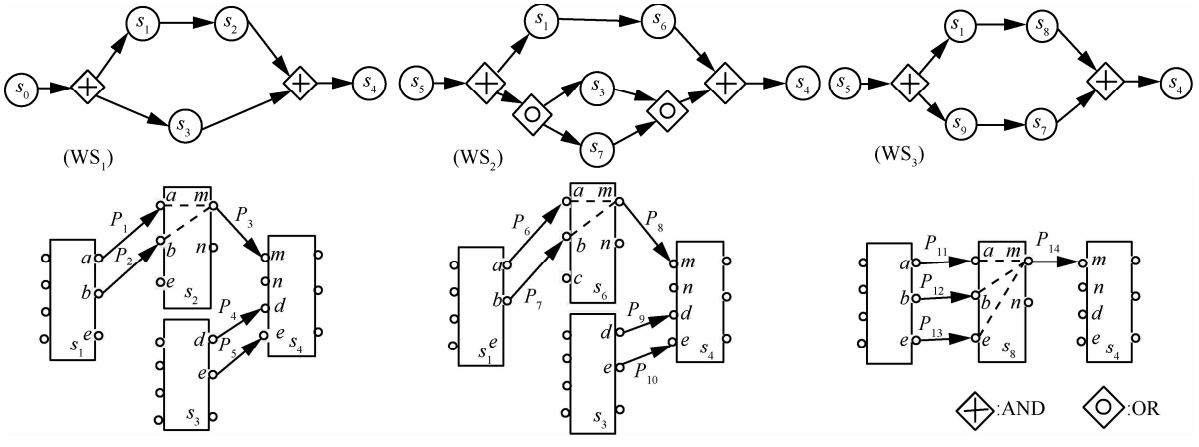


图 1 服务替换示例

2　相关工作

有关服务替换的问题是服务计算领域研究的热点问题之一。服务的替换问题与服务的发现问题密切相关，服务发现领域的一些研究成果为服务的可替换分析作出了重要贡献，主要集中在如何找出与需求功能相似的服务。然而，服务可替换分析与服务的发现不同，不仅需要功能相似，还需要考虑失效服务的组合上下文和服务的接口匹配问题。目前关于服务替换问题和服务组合上下文的相关研究，已有了较多的研究成果。

1) 服务替换相关研究

Marlon Dumas 等人[6]于 2010 年提出当服务失效时使用一种退化算法进行服务的查找替换，这种方法并不保证替换后的服务仍具有之前的所有功能，注重延长服务的使用时间而非致力于保证服务有更好的 QoS 性能，只要求替换后组合服务的功能和非功能属性在可接受的范围内，Marlon Dumas 等人的研究表明服务在组合时存在冗余的部分，而且随着组合服务粒度的提高，组合服务中存在冗余信息的子服务比例也会逐渐增多；Li Kuang 等人[3]将服务的行为使用 π-演算的方法形式化描述，基于强模拟和弱模拟理论，提出了实现服务功能和行为交互上的替换方法，此方法考虑到服务内部的输入输出接口的匹配，然而仍以失效服务为研究对象；邓水光等人[7]在研究服务发现时，考虑到目前服务描述的模型和协议不能满足要求，提出了另外一个解决的思路，即定义一种服务信息注册模型，要求模型中给出服务的接口依赖和语义相关信息，将此模型作为服务发现方法计算的数据来源，然而该方法要求服务开发者按照自己定义的模型注册服务，实施起来难度较大；刘莹等人[5]提出基于服务执行时的触发条件和结果判段服务是否可替换，实质是根据服务的输入输出判段服务功能的相似性，该方法没有考虑服务的组合上下文和服务接口依赖的问题。Zhang can 等人[8]针对目前服务和用户激增的问题，提出了一种动态的服务搜索方法以提高服务替换的效率。

2) 服务组合上下文相关研究

随着用户需求的逐渐提高和网络环境的不断变化，关于服务组合上下文的研究逐渐成为一个热点问题。Chan N N 等人[2]在 2011 年根据“判断某个人的好坏，向此人的母亲不如向此人的邻居获取信息”的思想提出考虑组合上下文进行服务的推荐，将服务组合流程图的历史信息形式化表示，提出了一种基于编辑距离的服务推荐方法。文中考虑到服务组合上下文对服务推荐的影响，但其形式化表示方法没有考虑服务的接口问题，不能直接用于解决服务替换的问题，但该工作为本文的服务替换研究提供了技术上的借鉴；Tan wei 等人[9]对如何将服务组合上下文流程的业务流程执行语言(BPEL, business process execution language)描述转变为着色 petri 网的服务工作流模型做出了研究，并详细地给出了转换的方法和过程。Zhang ming wei 等人[1]提出同一服务的 QoS 信息在不同的网络条件、网络通信数据量和服务器的前提下是不同的，当需要选择一个最优服务用于替换时，需要考虑服务在特定环境下的 QoS 信息，文中利用数据挖掘的相关理论提出 True QoS(TQoS)的采集算法，该文没有考虑服务的功能属性信息的采集问题。李喜彤等人[10]证明了当服务上下文组合流程具有良构性时，满足替换，但此法适用范围相对较小。以上关于服务组合上下文的研究为解决服务的替换问题提供了重要的依据。

3) 编辑距离应用的相关研究

编辑距离是一种计算字符串之间相似性的度量方法，近年来，不同研究领域对其的应用越来越多，也越来越成熟。范举等人[11]将编辑距离应用于空间数据的模糊查询问题中，综合考虑关键词的文本和空间距离，提出了一种支持查询容错的模糊查询方法；编辑距离在克隆检测问题中也有较多的应用，Thierry Lavoie 等人[12]与传统的基于编辑距离的克隆检测方法的性能进行比较和逼近，提出了一种新型的基于度量树和曼哈顿距离的克隆检测方法；编辑距离在数据库的数据查找问题中也有一定的应用，Sung-Hwan Kim 等人[13]提出了一种距离空间转换的方法，将待查询的一维的数据字符串转换成多维向量，利用空间数据索引技术解决查找问题，文中编辑距离的技术是其方法的基础。

3　相关概念介绍

目前，Web 服务的描述模型或协议中，并不记录服务在组合服务中的执行信息，而此信息是解决服务替换问题所需要的。本文选择从失效服务的组合上下文中获得所需的信息，而不是定义某种模型或协议，这样有利于方法的实施。

考虑到Web服务多样性的特点，本文使用着色petri 网对其进行形式化的统一描述，参考文献[14,15]中关于着色petri网的相关定义与理论介绍，本节给出定义1如下。

定义1 服务工作流网(SWN, service workflow net)

$SWN=(P, T, F, \Sigma, C, M_i, M_o)$，其中，$P$为有限库所集，$P=P^I\cup P^M$，$P^I$是服务流程内部库所集，$P^M$是服务与外界通信库所集，$P^I$和$P^M$互不相交；$P$中有2个特殊的库所，即起始库所$i$和终止库所$o$，满足${}^\bullet i=\varnothing$，$o^\bullet=\varnothing$即$i$的前集和$o$的后集均为空集。

T为有限变迁集，代表服务中执行的操作。

$F\subseteq P\times T\cup T\times P$为流关系。

Σ是类型的非空有限集，也称颜色集。

C为颜色函数，$C:P\to\Sigma$。

M_i和M_o分别为起始状态和终止状态。

除去库所集P^M，服务工作流网示例如图2所示，图中圆圈代表库所，长方形代表变迁，箭头代表链接库所和变迁的有向弧，每条有向弧的权重代表引起变迁的token和变迁发生所产生的token；服务开始执行时，根据起始状态M_i为每个库所分配token的数量和颜色（本文中token代表服务中的输入输出数据，token的颜色代表数据的类型），每个变迁因输入的token触发并产生新的token，当达到终止状态M_o时输出库所o中产生4个token，即{k，l，l，h}。

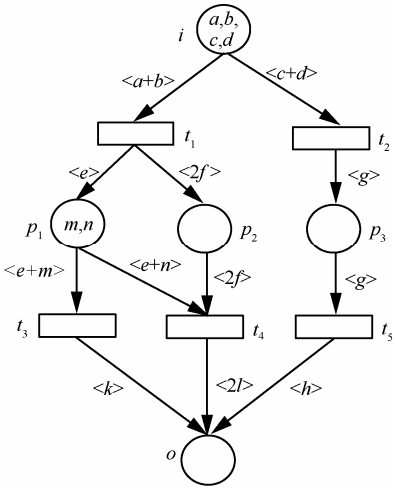


图2 服务工作流网示例

目前工业界已存在一些支持Web服务的工作流程建模语言，其中应用最多的是业务流程执行语言，该语言已被OASIS组织宣布为描述Web服务工作流程的标准。需要指出的是，基于BPEL标准描述的Web服务流程可以方便地转换为本文定义1中的形式化模型，如何将服务流程的BPEL描述转变为服务工作流网以及如何组合成更大粒度的服务工作流网可参考文献[9]，组合后的服务仍然满足定义1的SWN模型。本文着重介绍如何实现在定义1的基础上从组合上下文中采集服务的信息。

定义2 Web服务(WS, Web service)为四元组$WS=(ID,IO,CS,SWN)$ $WS=(ID,IO,CS,SWN)$，其中，ID是Web服务的唯一标识，可以是服务名称或标号。IO为Web服务输入输出集，$IO=\{I,O\}$，I和O分别为输入集与输出集，$I=\{i_0,i_1,\cdots,i_n,\ n=0,1,2,\cdots\}$，$O=\{o_0,o_1,\cdots,o_m,\ m=0,1,2,\cdots\}$。$CS$(component service)为原子和组合服务集；当$CS=WS.ID$时，WS为单个服务，否则WS为组合服务。$SWN=(P,T,F,\Sigma,C,M_i,M_o)$为有限服务工作流网的组合。

定义3 服务接口信息(SII, service interface information) $SII=(I',O',D)$，其中I'代表失效服务参与组合的实际输入，O'代表失效服务参与组合的实际输出，D代表输出集O'所依赖的输入集I'，满足：$I'\subseteq I, O'\subseteq O, D=\begin{cases}\varnothing, |I'|=|I|\\ I', |I'|<|I|\end{cases}$，$|I'|=|I|$表示输出对输入全局依赖的情况，$|I'|<|I|$表示输出对输入局部依赖的情况，只有当输出对输入存在局部依赖关系时，才会引起结果的偏差，因此本文仅考虑输出对输入局部依赖的情况。例如图1中服务WS_1的子服务s_2服务接口信息$SII=(a\ b,m,a\ b)$。

定义4 服务可替换度(Sub)用于定量表示候选服务是否能够用来替换失效服务的程度。可替换度Sub满足：$0\leqslant Sub\leqslant 1$，其值越大可替换性越好。当$Sub=0$时，服务的功能和接口依赖关系不满足要求，候选服务不能替换。一般情况下，由用户根据具体的组合服务特征及应用场景确定相应的阈值$\omega(0<\omega<1)$，当$Sub\geqslant\omega$时可以替换。特别地，当$Sub=1$时，功能和接口都满足替换要求，服务可替换。设$Sub(s_j,s_k)$表示服务s_k替换s_j的可替换度，例如图1中，$Sub(s_2,s_6)$表示服务s_6替换s_2的可替换度，第5节给出了服务可替换度的具体计算过程。

定义5 服务接口样式信息(SIPI, service interface pattern information)即服务的组合上下文中2个直接交互的服务的接口间的连接样式。为表述的方便，将服务接口样式信息SIPI简称为P。本文用元

样式集统一描述服务接口样式信息,形式化表示如下

$$P\left((i/o)_j,(i/o)_k\right)=\left((i/o)_j,p_1p_2\cdots p_n\right)$$

其中，*i/o* 代表服务组合上下文中某服务的输入或输出，选其一，关于选取方法视服务的相对位置而定；*p* 代表某一元样式，本文采用 6 种元样式[1]，即 {*sequence*, *AND-fork*, *AND-join*, *OR-fork*, *XOR-fork*, *OR-join*}。例如图 1 中：

$P_1=(a,sequence),P_2=(b,sequence)$，

$P_3=(m,AND-join),P_4=(d,AND-join)$，

$P_5=(e,AND-join),P_6=(a,sequence)$，

$P_7=(b,sequence),P_8=(m,AND-join)$，

$P_9=(d,OR-join,AND-join)$，

$P_{10}=(e,OR-join,AND-join)$，

$P_{11}=(a,sequence),P_{12}=(b,sequence)$，

$P_{13}=(e,sequence),P_{14}=(m,AND-join)$。

定义 6　服务组合上下文样式信息(SCCPI, service composition context pattern information)。为表述方便，将服务组合上下文样式信息 SCCPI 简称为 SPI。其形式化表示如下

$$SPI_{WS}(s_j,s_k)=\{P_1,P_2,\cdots,P_n\},n=1,2\cdots$$

其中，*P* 为服务接口样式信息，表示组合服务 WS 中服务 s_j 和 s_k 之间的连接样式信息和接口信息。例如图 1 中服务 WS_1 的子服务 s_2 的服务组合上下文样式信息：

$SPI_{WS1}(s_2, s_1)=\{P_1, P_2\}=\{a, sequence\}$；

$SPI_{WS1}(s_2, s_4)=\{P_3\}=\{m, AND\text{-}join\}$；

$SPI_{WS1}(s_2, s_3)=\{P_3, P_4, P_5\}$。

借鉴文献[2]中将服务的组合流程图按邻居分层的思想，将第 *k* 层邻居和第 *k* 域邻居的定义重新列出。

定义 7　(第 *k* 层邻居)。服务 *s* 的第 $k(k\geqslant 0)$ 层邻居指所有与服务 *s* 之间有 *k* 个连接样式的服务；$N_{WS}^k(s)$ 表示组合服务 WS 中服务 *s* 的第 *k* 层邻居服务集，例如图 1 中 $N_{WS_1}^0(s_2)=\{s_2\}$，$N_{WS_1}^1(s_2)=\{s_1,s_4\}$，$N_{WS_1}^2(s_2)=\{s_3\}$；$N_{WS_2}^0(s_6)=\{s_6\}$，$N_{WS_2}^1(s_6)$，$N_{WS_2}^2(s_6)=\{s_3\}$；$N_{WS_3}^0(s_8)=\{s_8\}$，$N_{WS_3}^1(s_8)=\{s_1,s_4\}$。

定义 8　(第 *k* 域邻居)。服务 *s* 的第 $k(k\geqslant 0)$ 域邻居指所有与服务 *s* 之间有 $l(0\leqslant l\leqslant k)$ 个连接样式的服务；$\boldsymbol{N}_{WS}^k(s)$ 表示组合服务 WS 中服务 *s* 的第 *k* 域邻居，$\boldsymbol{N}_{WS}^k(s)=\bigcup_{i=0}^k N_{WS}^i(s)$，例如图 1 中，

$\boldsymbol{N}_{WS_1}^2(s_2)=N_{WS_1}^0(s_2)\cup N_{WS_1}^1(s_2)\cup N_{WS_1}^2(s_2)=\{s_2,s_1,s_4,s_3\}$；

$\boldsymbol{N}_{WS_2}^2(s_6)=N_{WS_2}^0(s_6)\cup N_{WS_2}^1(s_6)\cup N_{WS_2}^2(s_6)=\{s_6,s_1,s_4,s_3\}$；

$\boldsymbol{N}_{WS_3}^1(s_8)=N_{WS_3}^0(s_8)\cup N_{WS_3}^1(s_8)=\{s_8,s_1,s_4\}$。

4　基于着色 petri 网的服务信息采集方法

与文献[2]中的组合上下文的含义一致，本文中组合上下文是指组合服务中与被替换服务组合的相关服务。目前，对组合上下文的描述，大致分为 2 类：1) 利用数据挖掘的相关理论提出一些算法，从服务的组合上下文中得出自己需要的信息；2) 利用某种形式化分析的理论，对组合服务流程建模后进行相关研究的方法，主要有基于有限状态机的方法、基于进程代数（或 π-演算）的方法和基于 petri 网的方法等。由于着色 petri 网的形式化分析理论能有效模拟复杂系统的数据流和控制流，它不仅可以将组合服务流程图形化表示，还能模拟流程中的输入输出相关信息，故本文选择采用着色 petri 网的分析理论对组合上下文流程进行研究。

当组合服务在执行过程中失效时，已知服务总体的组合流程(目前主要使用 BPEL 描述)，用户请求的输入输出，及每个原子服务的 WSDL 描述文档，本文利用这些信息实现服务的组合上下文信息采集算法(CCICA, composition context information collection algorithm)，主要包括服务接口信息的采集算法(SIICA, service interface information collection algorithm)和服务组合上下文样式信息的采集过程 2 部分。

首先将基于 BPEL 的组合服务的流程转换成 petri 网服务工作流模型，关于着色 petri 网的详细内容可参考文献[14,15]。

4.1　服务接口信息采集

上文找出被替换服务参与组合的实际执行部分的相关信息是本文对组合上下文形式化研究的目的之一。首先需要得到服务接口信息，即 SII。如上节中介绍，对组合服务流程形式化描述后，得到的服务工作流网也是一种有向图 *G*，本文对有向图深度优先遍历(DFS)算法进行了改进，可以得到源节点到目的节点之间的所有路径，同时结合着色 petri 网变迁的触发规则[9]，得到本文的服务接口信息采集算法如下。

服务接口信息采集算法。

输入：服务工作流网的有向图 $G(V, E)$；

输出：失效服务接口信息 *SII*(*I*', *O*', *D*)；

1) BEGIN

2) FOR EACH place and *j* place.token= push (M_j.tokens)；

3) FOR EACH place and transition *p*,*t*∈ *G*

4) p.color=*t*.color=WHITE;*p*.parent=*t*.parent= NULL;

5) *v*=*i*;

6) NodeVisit(*G*,*v*)

7) END

NodeVisit(*G*,*v*)

1) IF(*v.tag* = 'end')

2) Depth.Push (*v*)；

3) IF *v*=*i*'

4) *SII.I*'=push(*token.type*)； *D*=*I*';

5) IF *v*=o'

6) SII.*O*'=push(token.type);

7) Depth.pop()；

8) ELSE

9) FOR each *v*∈ *G*.Adj[*u*]

10) IF (*v*==*node.place*)

11) IF (*v.color*==WHITE and *u.token.nu*>0)

12) Depth.Push (*v*)；

13) *v.parent*=*u*;

14) *v.oplace*=push(*tokens*);

15) *v.iplace*=pop(*tokens*);

16) ELSE

17) IF (*v.color*==WHITE)

18) Depth.Push (*v*)；

19) *v*.parent=*u*;

20) NodeVisit(*G*,*v*);

21) Depth.pop()；

RETURN

此算法中包含对库所节点和变迁节点的递归遍历，对变迁节点的遍历算法中需要加入变迁的触发规则；因服务工作流网的组合仍满足定义 1 模型，此算法仍适用。此算法由图的深度优先遍历算法改进而来，时间复杂度为 $O(V^2)$，算法虽牺牲了一定的计算时间，然而获取了大幅减少冗余后的服务信息，对服务可替换度的计算有重要意义。

不失一般性，以图 2 服务工作流网为例，当服务接口信息采集算法执行结束时，如果库所 *i*,p_3,*o* 和变迁 t_2,t_5 是一条有效的执行路径，则 *SII*=(*c d*,*h*,*c d*)。需要说明着色 Petri 网服务工作流中的信息均有自己的颜色即类型，所以 SII 中得到的是输入输出接口的类型，而非某一次执行实例的输入输出信息。

4.2 服务组合上下文样式信息采集

由定义 6 可知，服务组合上下文样式信息由服务接口样式信息组成，即服务的组合上下文中 2 个直接交互的服务的接口间的连接样式，如定义 5 中介绍本文使用 6 种元样式进行形式化表示，当且仅当采集的服务接口信息中包含某接口时，才需要对此接口的连接样式进行形式化表示，将 2 个直接交互的服务间所有接口形式化表示后，按定义 5~定义 8 即可得到服务组合上下文信息，形式化表示为：

$$SPI_{WS}(s_i,s_j)=\{P_1,P_2,\cdots,P_n\}$$

其中，*P* 为服务接口样式信息，*n* 为服务 WS 中子服务 s_i 与 s_j 之间服务接口样式信息的个数。例如图 1 中 $SPI_{WS_1}(s_2,s_1)=\{P_1,P_2\}$, $SPI_{WS_1}(s_2,s_4)=\{P_3,P_4\}$，$SPI_{WS_1}(s_3,s_4)=\{P_4,P_5\}$。

5 基于编辑距离的服务替换方法

在信息论中，编辑距离是指 2 个字符串之间由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。例如，kitten 和 sitting 的编辑距离是 3，gumbo 和 gambol 的编辑距离是 2 等。基于这个计算方法，本文把服务接口样式信息中每个元素当成一个字符，那么服务接口样式信息即为一个字符串，它们间的匹配度就可采用编辑距离的方法方便地计算出来。

在已知 SII 和 SPI 的条件下，介绍基于编辑距离的服务替换方法(LDBSSM, levenshtein distance-based service substitution method)，在这一方法的计算过程中，本文加入了服务接口的依赖关系约束，需要说明的是本文的服务可替换度是考虑失效服务的实时信息和服务的接口依赖关系条件下计算而来的，可以用作服务的可替换判断的依据，而服务发现和查找领域的服务相似度仅用于判断服务是否满足用户的请求，并不能用于判断服务是否满足替换的要求，故本文的服务可替换度与服务相似度是不同的。

5.1　服务接口样式信息匹配

因为服务组合上下文由服务接口样式组成，所以为了计算服务的组合上下文匹配度，需要先计算服务接口样式的匹配度，由上文，服务接口样式信息是 2 个服务直接交互的接口之间的信息，已经使用 6 种元样式对其形式化描述，本文采用编辑距离的方法计算它们之间的匹配度。具体给出 2 个服务接口样式信息如下

$$P\left((i/o)_j,(i/o)_k\right)=\left\{(i/o)_j,p_1p_2\cdots p_n\right\}$$

$$P'\left((i/o)'_j,(i/o)'_k\right)=\left\{(i/o)'_j,p'_1p'_2\cdots p'_m\right\}$$

它们之间的匹配度由式(1)进行计算。

$$M_P(P,P')=1-\frac{LenvenshteinDistance(P,P')}{\mathrm{Max}(n,m)+1} \tag{1}$$

其中，$0\leqslant M_P\left(P,P'\right)\leqslant 1$，当 $P=P'$ 时，$M_P\left(P,P'\right)=1$；当 $P\subset P'$，$m\geqslant n$ 时，$M_P\left(P,P'\right)=\dfrac{n+1}{m+1}$。

需要指出的是当连接 2 个接口的有向弧方向相反时，匹配度为零。即

$$M_P\left(P((i/o)_j,(i/o)_k),P'((i/o)'_k,(i/o)'_j)\right)=0$$

$$M_P\left(P((i/o)_k,(i/o)_j),P'((i/o)'_j,(i/o)'_k)\right)=0$$

5.2　服务组合上下文信息(FSP)匹配

有了服务接口样式的匹配度，本节计算服务组合上下文匹配度，因服务组合上下文信息由服务接口样式信息集合而来，且集合中每个元素的地位是平等的，所以本文采用将服务接口样式信息匹配度求和取平均的方法计算服务组合上下文信息匹配度。

具体给出 2 个服务组合上下文信息如下

$$SPI_{WS}(s_j,s_k)=\{P_1,P_2,\cdots,P_n\}$$

$$SPI'_{WS'}(s'_j,s'_k)=\{P'_1,P'_2,\cdots,P'_m\}$$

则它们之间的匹配度计算如下

$$\bar{M}_{SPI}(SPI,SPI')=\frac{\sum_{t=1}^{x}M_P(P_t,P'_t)}{x},x=\mathrm{Min}(n,m) \tag{2}$$

其中，$x=\mathrm{Min}(n,m)$ 表示当 2 个集合中元素个数不相同时，以 2 个集合中元素较少者为计算的标准，舍弃掉元素较多的集合中无法对应的服务接口样式信息的匹配度计算，选择舍弃掉并不会影响计算的结果。

原因是在 5.4 节的服务替换方法中对接口的匹配做出了判断，需要满足的必要条件是：1) 替换服务输出集包含被替换服务输出集；2) 被替换服务输入集包含替换服务输入集。假设失效服务和待替换服务信息为：$SII(I,O,D)$ 和 $SII'(I',O',D')$，则接口匹配需满足的条件形式为：$I\supseteq I',O\subseteq O',D\supseteq D'\Rightarrow|I|\geqslant|I'|,|O|\leqslant|O'|,|D|\geqslant|D'|$。为了避免重复考虑带来的问题，式(2)中不再考虑多出接口的信息匹配问题。

5.3　服务的可替换度计算

上文中根据服务组合上下文信息（SPI）计算出了服务组合上下文匹配度，本节基于组合上下文匹配度，提出了计算服务的可替换度的方法，方法中需考虑服务接口的依赖关系的问题和为不同邻居层的服务分配权重的问题。

由于得到的服务信息（SII(*I*,*O*,*D*)）中包含有服务接口的依赖关系信息，故可在算法中直接做出比较判断。

对于为各邻居层服务分配权重的问题，直观上，层数越大，对目标服务的影响就越小，相应的权重就越低，而且考虑的总层数不同，每层的权重也不同，所以本文与文献[2]中权重分配方法一致，即按照式 $w_l=\dfrac{k-l}{k}$ 分配权重方法，其中，k 表示算法考虑的总层数，l 表示需分配权重的当前层数。加入权重后的服务可替换度计算公式为

$$Subs_{WS_a,WS_b}\left(s_i,s_j\right)=\frac{\sum_{l=0}^{k-1}\dfrac{k-l}{k}\sum\bar{M}_{SPI}}{\left|E^k_{WS_a}(s_j)\right|} \tag{3}$$

其中，$\sum\bar{M}_{SPI}$ 表示某邻居层中所有服务组合上下文样式信息匹配度的求和，$\left|E^k_{WS_a}(s_j)\right|$ 表示组合服务 WS_a 中服务 s_j 的第 k 域邻居范围内所有组合上下文信息个数。

5.4　基于编辑距离的服务替换方法

基于编辑距离的服务替换算法如下。

输入：服务 WS_a 中失效服务 s_j 的接口信息（SII_i）和组合上下文样式信息集（SPI_1,SPI_2,…）；服务 WS_b 中待替换服务 s_k 的接口信息（SII_j）和组合上下文样式信息集（SPI_1',SPI_2',…）；

输出：满足替换要求的服务集 WS-set；

BEGIN

1) IF(|SII_i.I|<|SII_j.I|)

2) RETURN *Sub*=0;

3) ELSE IF($|SII_j.O`|<|SII_i.O`|$)

4) RETURN Subs=0;

5) ELSE IF($|SII_i.D|<|SII_j.D|$)

6) RETURN *Sub*=0;

7) ELSE {

8) CALCULATE $Subs=\dfrac{\sum_{l=0}^{k-1}\dfrac{k-l}{k}\sum \bar{M}_{SPI}}{\left|E^{k}_{WS_a}(s_j)\right|}$;

9) IF($Sub\geqslant\omega$)

10) WS-set.PUSH(*S*, *Sub*);

11) }

END

步骤 1)~6)为服务可替换必要条件的判断过程；步骤 7)为不同邻居层分配权重；步骤 8)为服务可替换度的计算过程，步骤 9)~10)将满足用户设定阈值的服务及其可替换度保存到满足条件的服务集中。本文取可替换度最高的进行失效服务的替换。如图 1 中 Sub(s_2,s_6)≈0.69，Sub(s_6,s_2)≈0.65，Sub(s_2,s_8)≈0，算法的结果会选择 s_6 而舍弃 s_8。

6 仿真实验与结果分析

本文的服务信息采集方法是针对目前的替换方法以现有的服务描述模型和协议为失效服务信息采集来源所带来的匹配信息存在冗余的问题而提出的，此冗余会导致某些满足替换的服务无法被召回，本文方法中以服务组合上下文为失效服务的信息来源，以减少冗余，使满足条件的候选服务能够更好地被召回；此外，从服务的组合上下文中还得到了失效服务的接口依赖信息，此信息用在本文提出的基于编辑距离的服务可替换度的计算过程中，进行接口依赖的判断，从而提高服务替换的准确性。

因此本节验证服务信息采集方法对提高候选替换服务集召回率的影响，以验证本文方法减少失效服务冗余信息的有效性，和基于编辑距离的服务可替换性分析方法对提高候选替换服务集准确率的验证。召回率是指从服务集中返回的满足替换要求的服务数占服务集中满足替换要求的服务总数的比例；准确率是指从服务集中返回的所有服务中，能满足替换要求的服务数占返回的所有服务数的比例。例如，若对于某个失效服务替换请求，服务集中存在满足替换要求的服务数量为 n，使用服务替换方法从该库中返回 m 个，但这 m 个中只有 k 个真正满足用户请求，则该方法的召回率为 k/n，而准确率为 k/m。

6.1 实验准备

为了能够方便地对服务集的参数根据实验需要进行配置，本文使用应用程序模拟生成服务集。首先生成 10 组子服务集，每个子服务集包含 5 个功能相似的子服务，总共 50 个子服务；然后由这些子服务组合成 10 个组合服务，每个组合服务包含 4~6 个子服务，每个子服务包含 2~3 个输入和输出接口，输入输出接口的类型从 1~10 中随机选取。实验中将本文方法和文献[2]中基于组合上下文匹配的服务推荐方法(CCMSR)和文献[3]中基于 π-演算的服务行为替换分析方法(ABSWS)进行了比较仿真，在生成服务集的过程中需要配置的参数包括：服务集中存在冗余信息的服务比例记为 r_1 和存在服务接口局部依赖的比例记为 r_2。

表 1 r_2=0, r_1 不同的服务集

服务集	r_1/%	服务集	r_1/%
S-1	0	*S*-6	50
S-2	10	*S*-7	60
S-3	20	*S*-8	70
S-4	30	*S*-9	80
S-5	40	*S*-10	90

表 2 r_1=0, r_2 不同的服务集

服务集	r_2/%	服务集	r_2/%
S'-1	0	*S*'-6	50
S'-2	10	*S*'-7	60
S'-3	20	*S*'-8	70
S'-4	30	*S*'-9	80
S'-5	40	*S*'-10	90

本实验硬件环境为 Intel(R)Core(TM) i3-2310M CPU @2.10 GHZ 2.10 GHz，2.00 GB 内存。软件环境为 Win7 ultimate、eclipse-SDK-3.5.2-win32，仿真程序用 Java 语言编写。

6.2 实验与结果分析

实验 1 服务信息采集方法有效性验证

为了验证本文服务信息采集方法对提高失效服务候选替换服务集的召回率的有效性，实验 1 中

从表 1 的每种服务集中随机选取 5 个服务作为失效服务，取阈值 ω=0.8，将所计算的召回率求和取平均，图 3 给出了仿真测试结果。结果表明：在不考虑服务接口局部依赖即 r_2=0 的情况下，随着 r_1 的增加，ABSWS 方法未考虑组合上下文致使召回率趋于降低，而本文方法和 CCMSR 均考虑了组合上下文，有效地减少了失效服务的冗余信息，体现为召回率较稳定，且有升高趋势。因本实验中设定 r_2=0 且仅从召回率的角度进行比较，在此特定情况下，CCMSR 方法也能较好地考虑组合上下文，所以本文方法和其没有明显差别，这一实验结果与理论预期也是相符合的，仍能说明本文所采用的考虑组合上下文的 LDBSSM 服务替换方法中使用的服务信息采集方法是有效的。

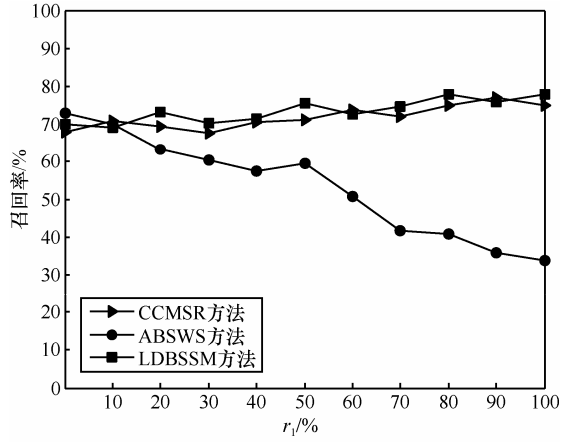


图 3　服务信息采集方法有效性验证

实验 2　服务替换方法有效性验证

为了验证基于编辑距离的服务可替换性分析方法中考虑接口局部依赖对提高失效服务候选替换服务集的准确率的有效性，实验中从表 2 的每种服务集中随机选取 5 个服务作为失效服务，取阈值 ω=0.8，将所计算的准确率求和取平均，图 4 中的仿真结果表明：在不考虑服务存在冗余信息的情况下，随着 r_2 的增加，因 CCMSR 未考虑接口局部依赖，使准确率趋于降低，而本文 LDBSSM 和 ABSWS 考虑了接口局部依赖因素，则不受影响，且有升高的趋势。因本实验中设定 r_1=0 且仅从准确率的角度进行比较，此情况下，ABSWS 方法也能较好地解决接口局部依赖关系的问题，所以本文方法和其没有明显差别，然而仍可得出结论：本文所采用的基于编辑距离的服务可替换性分析方法中考虑接口局部依赖关系是有效的。

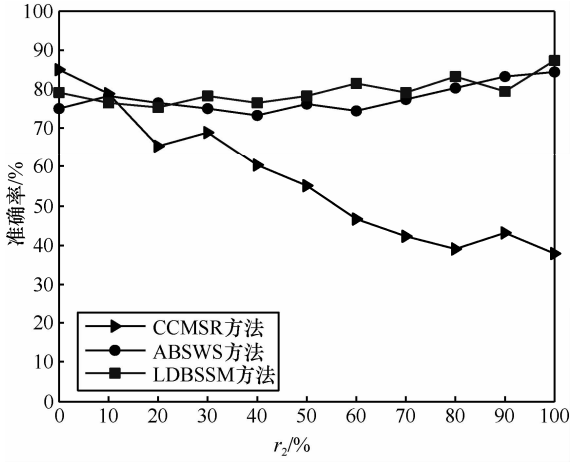


图 4　基于编辑距离的服务替换方法有效性

实验 3　服务替换方法 LDBSSM 对阈值 ω 的适用性验证

为了验证阈值 ω 的选取对本文方法的影响，实验中取失效服务第 k=3 域邻居服务作为匹配计算的组合上下文，取 r_1=20%, r_2=20%。

图 5 中反应了 ω 不同取值下，服务数量的分布比例，当 ω 取 0～0.5 时，本文 LDBSSM 方法返回的服务数量比例占 25%，CCMSR 方法返回的服务数量比例占 17%，ABSWS 方法返回的服务数量比例占 20%；当 ω 取 0.5～1.0 时本文 LDBSSM 方法返回的服务数量比例占 75%，CCMSR 方法返回的服务数量比例占 83%，ABSWS 方法返回的服务数量比例占 80%。可见与 CCMSR 和 ABSWS 方法相比，由本文 LDBSSM 方法计算的满足高阈值的服务数量较少，相反，满足低阈值服务数量较多，这是由于本文考虑了服务组合上下文和服务接口局部依赖的原因。

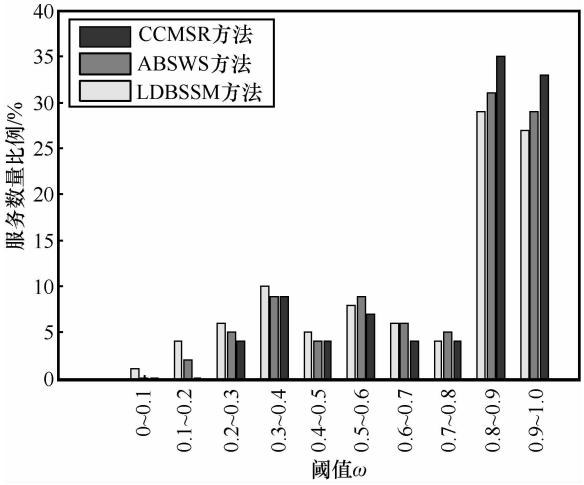


图 5　满足不同阈值的服务数量分布

图 6 中反应了随着阈值的不同本文 LDBSSM 方法与 CCMSR 方法和 ABSWS 方法返回服务的准确率高低的比较结果。结果可知，虽然本文方法使得可替换度高的服务数量减少了，但是当替换的阈值较高时，得到的候选替换服务集的准确率却相对提高，说明本文 LDBSSM 方法在阈值较高时更加有效，具有更好的实用价值。

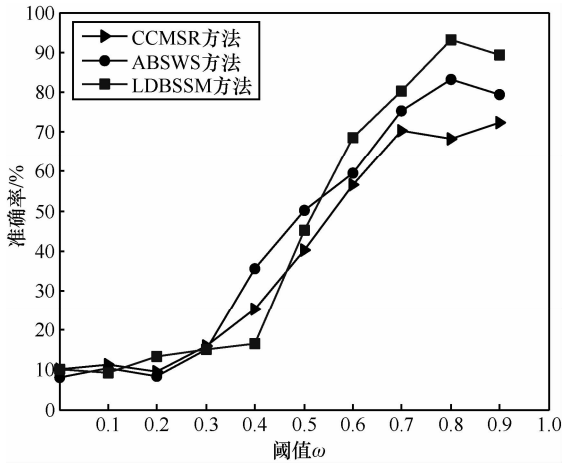


图 6 阈值 ω 对 3 种方法的影响

7 结束语

获得准确的候选替换服务集是保证服务替换正确执行的重要前提，本文从 2 个方面解决这个问题。第一，以服务的组合上下文为研究对象，基于转换后的着色 petri 网服务工作流模型，提出了一种服务信息采集方法，仿真实验表明该方法有效地去除了冗余的待替换服务描述信息，提高了候选替换服务集的召回率；第二，将服务接口的依赖关系问题加入到服务可替换验证方法中，提出了一种基于编辑距离匹配的服务替换分析方法，有效地提高了候选替换服务集的准确率，弥补了现有服务替换相关研究对此问题考虑较少的空缺。下一步的工作是将服务的非功能属性加入到服务替换的方法中[16~18]。

参考文献：

[1] ZHANG M W, ZHU Z L, ZHANG L, *et al.* An execution context aware approach for Web service substitution[A]. Proc of the International Conference on Advanced Information Management and Service[C] .Jeju,Korea,2011.13-18.

[2] CHAN N N, GAALOUL W, TATA S, *et al*. Composition context matching for Web service recommendation[A]. International Conference on Services Computing[C].Washington DC,USA,2011.624-631.

[3] LI K, XIA Y J,WU J, *et al*. Analyzing behavioral substitution of Web services based on π-calculus[A].IEEE International Conference on Web Services[C].Miami,FL,USA,2010.441-448.

[4] FREDJ M, GEORGANTAS N,ZARRAS A, *et al*. Dynamic service substitution in service-oriented architectures[A].IEEE Congress on Services[C]. Honolulu, HI, USA, 2008.101-104.

[5] 刘莹,张一川,张斌等.基于行为效果的服务可替换性分析[J].计算机研究与发展,2010,47(8): 1442-1449.
LIU Y, ZHANG Y C, ZHANG B, *et al*. Analysis of service replaceability on behavior effect[J].Journal of Computer Research and Development,2010,47(8): 1442-1449.

[6] DUMAS M, YANG Y,ZHANG L. Improving Web service survivability via gracefully degraded substitution[A].International Conference on Web Intelligence and Intelligent Agent Technology[C]. Toronto, Canada, 2010. 597-600.

[7] 邓水光,尹建伟,李莹等.基于二分图匹配的语义 Web 服务发现方法[J]. 计算机学报, 2008,31(8):1364-1375.
DENG S G, YIN J W, LI Y, *et al*. A method of semantic Web service discovery based of bipartite graph matching[J].Chinese Journal of Computers, 2008,31(8):1364-1375.

[8] ZHANG C, CHEN H P,DU J B. A Tabu search approach for dynamic service substitution in SOA applications[A]. IEEE Asia Pacific Services Computing Conference[C].Jeju, Island, Korea, 2011.284-289.

[9] TAN W,FAN Y S,ZHOU M. A petri net-based method for compatibility analysis and composition of Web services in business process execution language[J]. IEEE Transactions on Automation Science and Engineering,2009,6(1):94-106.

[10] 李喜彤,范玉顺.Web 服务流程相容性和相似性分析[J].计算机学报,2009,32(12):2429-2437.
LI X T, FAN Y S. Analyzing compatibility and similarity of Web service process[J]. Chinese Journal of Computers, 2009,32(12):2429-2437.

[11] 胡俊,范举,陈姗姗等.空间数据上 Top-*k* 关键词模糊查询算法[J].计算机学报,2012,35(11):2237-2246.
HU J, FAN J, CHEN S S, *et al*. Top-*k* fuzzy spatial keyword search[J]. Chinese Journal of Computers, 2012, 35(11):2237-2246.

[12] LAVOIE T,MERLO E. An accurate estimation of the levenshtein distance using metric trees and manhattan distance[A].International Workshop on Software Clones[C].Zurich,2012.1-7.

[13] SUNG-HWAN K,JONG-KYU S,HWAN-GUE C. Flexible and efficient string similarity search with alignment-space transform[A]. Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication[C].New York, NY, USA,2013.

[14] JENSEN K. An introduction to the theoretical aspects of colored petri nets [J]. Lecture Notes in Computer Science, 1994, 803: 230-272.

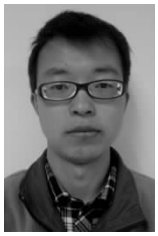
[15] 郭玉彬,杜玉越,奚建清.Web 服务组合的有色网模型及运算性质[J].计算机学报,2006,29(7):1067-1075.
GUO Y B, DU Y Y,XI J Q. A CP-net model and operation properties

（下转第 78 页）

[12] LEE C W, CHEN L M, CHEN M C, *et al*. A framework of handoffs in wireless overlay networks based on mobile IPv6[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(11): 2118-2128.
[13] IEEE Standard for Local and Metropolitan Area Networks-Part 21: Media Independent Handover Services[S]. IEEE 802.21, 2008.
[14] LEE K, HONG S, KIM S J, *et al*. SLAW: self-similar least-action human walk[J].IEEE/ACM Transactions on Networking, 2012, 20(2): 515-529.

赵季红（1963-），女，陕西西安人，博士，西安交通大学教授、西安邮电大学教授、博士生导师，主要研究方向为宽带通信网、新一代网络的管理与控制。

作者简介：



任堪晔（1990-），男，甘肃庆阳人，西安交通大学博士生，主要研究方向为异构网络中的无线资源管理。

曲桦（1961-），男，陕西杨凌人，博士，西安交通大学教授、博士生导师，主要研究方向为现代通信网、计算机网络体系结构。

（上接第 66 页）

for Web service composition[J]. Chinese Journal of Computers, 2006, 29(7): 1067-1075.
[16] ZHENG Z B, LYU M R. Optimal fault tolerance strategy selection for Web services[J].International Journal of Web Services Research,2010, 7(4):21-40.
[17] ZHENG Z B, LYU M R. An adaptive QoS-aware fault tolerance strategy for Web services[J]. Empirical software Engineering, 2010,15(4): 323-345.
[18] SANTHANAM G R,BASU S,HONVAR V. Web service substitution based on preferences over non-functional attributes[A].IEEE International Conference on Services Computing[C]. Bangalore,India, 2009. 210-217.

作者简介：

王海艳（1974-），女，江苏东台人，博士，南京邮电大学教授，主要研究方向为计算机网络、大数据应用技术、可信计算技术等。

李思瑞（1987-），男，安徽阜阳人，南京邮电大学硕士生，主要研究方向为基于网络的计算机软件应用技术等。