ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.19, No.11, November 2008, pp.3011−3022
DOI: 10.3724/SP.J.1001.2008.03011

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 基于多重服务范例适应性调整的服务组合*

成睿星+, 杨放春, 苏 森

(北京邮电大学 网络与交换技术国家重点实验室,北京 100876)

## Service Composition Based on Adaptable Revision of Multiple Service Case

CHENG Rui-Xing+, YANG Fang-Chun, SU Sen

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

+ Corresponding author: E-mail: chengruixing@gmail.com, http://www.bupt.edu.cn

**Cheng RC, Yang FC, Su S. Service composition based on adaptable revision of multiple service case. *Journal of Software*, 2008,19(11):3011−3022.** http://www.jos.org.cn/1000-9825/19/3011.htm

**Abstract**: In this paper, a hierarchical semantic service case is outlined, which supports overall description of functionality, behavior restraints and features. An adaptable similarity degree measurement is proposed to retrieve a service case which is more similar to the target case and easier to be revised if necessary. Moreover, a method based on the interaction ontology is presented to realize dynamically revising service case to support Web service composition. The method of Web service composition proposed in this paper is proved to be feasible and effective by examinations.

**Key words**: Web service composition; case-based reasoning; multiple service case; adaptable revision; revision operator

**摘 要**: 提出了抽象多重服务范例来解决服务组合问题.同时,为了提高可适应性,给出一种用来选择适合调整的服务范例的可适应性的相似度测量方法,并提出了基于调整运算子的服务范例适应性调整方法.在这些工作的基础上实现服务组合.实验表明,该方法是可行和有效的.

**关键词**: Web 服务组合;范例推理;多重服务范例;适应性调整;调整运算子

**中图法分类号**: TP393 **文献标识码**: A

面对不断涌现的 Web 服务,需要根据用户的需求,将分布在不同环境、平台及企业间已经存在的 Web 服务合成一个粒度更大的、能够满足用户个性需求的增值服务.由于用户的需求具有多样性、动态性等特点,如何根据用户的需求,高效、正确地实现自动 Web 服务组合仍然是有待解决的问题.尽管工业界和学术界在 Web 服

* Supported by the National Natural Science Foundation of China under Grant No.60672121 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z164 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2003CB314806 (国家重点基础研究发展计划(973)); the Program for Changjiang Scholars and Innovative Research Team in University of China under Grant No.IRT0410 (长江学者和创新团队发展计划资助项目); the Beijing Education Committee Collaboration Science Program of China (北京市共建项目)

Received 2007-04-30; Accepted 2007-08-03

务组合方面进行了大量尝试[1],但是,在业务层面实现自动服务组合仍然是一个具有挑战性的问题[2].

基于范例推理的服务组合方法[3]是将范例推理技术用于服务组合,利用知识积累构建服务组合范例库和服务范例,并将现有的服务组合引擎的推理工作移至范例推理系统中,通过在范例库内检索和调整领域专家预先设计的服务范例,以及对用户以往的成功经验的复用,得到能够满足用户需求的组合服务逻辑集成,并提交服务执行引擎处理,实现服务组合.在一定条件[3]下,该方法能够提高服务组合的效率、成功率和可复用性.但是,在处理较为复杂且先验知识不足的服务组合时,现有的基于范例推理的服务组合方法[4,5]的成功率和效率远远不能满足用户的需求[2].在服务范例的结构方面,现有的基于范例推理的服务组合采用单一服务范例解决服务组合问题.由于单一服务范例的知识及能力的限制,仅仅使用单一的服务范例来解决一个复杂且庞大的服务组合问题是不合理且缺乏效率的.在服务范例的适应性调整方面,现有的基于范例推理的服务组合仅仅将范例推理用于服务组合的发现阶段,通过选取相似度较大的服务范例来满足用户的请求,没有考虑服务范例的适应性调整,使得这种方法的成功率不高,难以保证服务组合的性能和质量.

针对上述问题,我们在自己先前研究工作[3]的基础上,提出一种基于多重服务范例适应性调整的服务组合方法.用阶层式抽象服务范例的概念构造多重服务范例,同时给出一种可适应性的相似度测量算法,以选择适合调整的服务范例,并基于调整运算子的服务范例适应性调整方法将其调整成满足用户需求的服务范例,最终实现服务组合.

## 1 基于多重服务范例适应性调整的服务组合

由于基于范例推理的服务组合系统是利用过去解决问题的经验来解决新的问题,在实际的服务组合过程中,往往过去处理过的问题与新问题类似,但并不完全相同,以至于过去解决问题的服务范例并不完全适合于新问题.因此,过去解决问题的服务范例必须作适当的调整才能解决新的问题.在范例推理中,范例的调整(revise)是一个相当复杂且具有难度的任务[6].虽然已有许多范例调整技术[7,8]被提出来并应用于范例推理系统中,但是,这些范例调整的方式大多是以调整规则与限制条件来完成范例调整的任务,而且领域相关性很强,不同领域的范例推理系统都有属于自身所掌握的领域知识[9]及所要解决的领域问题[10],并且很难重复使用.

服务范例的适应性调整是指在服务组合过程中,系统自动地将不能满足用户需求的服务范例调整成能够适应用户需求的服务范例.服务范例的可适应性是指服务范例的适应性调整的能力.为了提高服务范例可适应性、可重用性以及服务范例库的可维护性,同时降低适应性调整的复杂程度,我们将服务范例划分为抽象服务范例和具体服务范例,并且依据不同的抽象程度将抽象服务范例划分至若干阶层,采用从上至下逐层精制的映射机制将不同阶层的抽象服务范例与具体服务范例相关联,构成一个可以反映业务层面服务组合需求并且与具体的服务组合解决方案相关联的大粒度、高适应性的服务范例.

当收到用户的请求及服务组合的约束条件后,首先对用户的请求进行语义标注,生成带语义标注的问题描述,并以此为索引进行检索.根据可适应的相似度测量算法在抽象服务范例层面进行检索,并且由抽象至具体逐层精制,直到具体服务范例.当检索到一个或者若干个相似度较高的候选服务范例时,范例推理处理器根据匹配算法和适应算法确定最适合的服务范例,然后判断是否需要进行适应性调整,如果需要进行适应性调整,则将该服务范例提交给调整运算子进行适应性调整,并且验证适应性调整的过程和结果.通过上述机制,将满足服务组合需求的服务范例提供给服务执行引擎执行,并将结果返回给请求者.

### 1.1 抽象服务范例与多重服务范例

我们可以通过阶层式表示法将一个服务组合目标分解成 1 个至数个子目标,表达目标与子目标的关系及子目标与子目标之间的关系,使得位于不同抽象阶层的目标得以连结.位于不同抽象阶层的目标即代表复杂度的不同,位于较高阶的目标较为复杂、抽象,位于低层的子目较为简单也更为具体.目标所对应的子目标及执行步骤也是以抽象阶层的方式来表示,主要分为两种不同的范例:抽象范例及具体范例.范例与范例之间的关系是以抽象联结的方式来表达范例阶层的关系,抽象连结的意义为上层抽象范例完成由下层较具体的范例来满足,范例的抽象阶层关系对于服务范例的调整具有重要意义.

下面我们首先讨论,分析基于范例推理的服务组合所面临的问题,然后提出基于抽象化服务范例与多重服务范例的概念构建范例的求解部分,使其能够解决较为复杂的 Web 服务组合问题.

由于所要解决的问题愈来愈复杂及庞大,传统的范例推理系统缺乏足够的效率来解决问题[6].单一范例均有解决问题的观点、知识及能力的限制,所以,仅仅使用单一的服务范例来解决一个复杂且庞大的服务组合问题是不合理且缺乏效率的.因此,我们可以利用抽象服务范例(abstract service case)的概念与多重(multiple)服务范例来解决复杂的 Web 服务组合问题.将一个复杂且庞大的 Web 服务组合问题由范例推理系统分解成数个不同层次的抽象阶层.抽象服务范例有助于服务范例的索引及相似度的测量,提升基于范例推理的服务组合系统在服务范例获取上的效率.使用抽象服务范例具有如下优点:

(1) 服务范例的抽象化可以减少服务范例的复杂度,而且抽象服务范例所描述的是问题重要且不被忽略的特征,且没有过于细节的描述,所以比一般单一含盖所有问题的服务范例表达上更具有弹性,且易于完成.

(2) 抽象服务范例有助于服务范例的相似度的测量,可以增加服务范例匹配(service case matching)的效率.

(3) 在服务范例的调整上可以减少调整知识的需求量.在此阶层架构中,包含所有种类的服务范例,最上层的服务范例表示其抽象程度(the level of abstraction)最高,服务范例阶层依次由上至下,越下层的服务范例描述得越具体,在抽象结构中,最低层的服务范例即是具有最具体的描述,整个抽象阶层中较上层的抽象服务范例都是由较底层的具体服务范例(concrete service case)建构而成.

当抽象服务范例被选取出来以后,其抽象解必须被系统加以精制(refinement),将原始的抽象解补足直到足够详细以解决所遇到的问题为止.而系统所选出的越下层的服务范例,其解就越具体、详细,将其解精制直到可以利用来解决问题的范围也就越小.在实际的 Web 服务组合过程中,有很多问题是很庞大且很复杂的,基于范例推理的服务组合系统则可以利用多重服务范例来解决此类问题.我们可以将此类问题分解成数个个别的子问题(sub-problems),将各个子问题视为单一问题,而系统针对每个子问题分别选取出符合的解,进而解决每个子问题,每个子问题的解我们称为部分的解(part solution).最后,将所有的部分解整合(integrate)成可以解决庞大且复杂的问题的解.

抽象服务范例由问题描述部分(description part)和求解部分(solution part)构成.抽象服务范例的求解部分是描述所要达成 Web 服务组合目标所对映的抽象解.抽象解所说明的是上层主要目标所分解的下层子目标,所以,我们可以把抽象解看成是将一个复杂目标分解成 1 个或数个子目标的说明.所以,一个抽象服务范例的完成可以由 1 个或是数个更具体的服务范例来完成.抽象服务范例的求解部分并不会涉及待解决问题的过于具体的细节部分,主要描述组成待解决问题片段的重要属性,如特征、限制、与下层服务范例的关系等内容,以减少其复杂度.抽象服务范例的求解部分可以视为包含有服务组合控制逻辑的虚服务.控制逻辑是描述服务组合的组合逻辑规划,包括服务组合类型与消息依赖,它是一个服务组合的虚服务,具有稳定性和永远不会失败的特性.一个服务组合的逻辑规划是不可执行的,需要与具体的服务相关联形成可执行的服务组合流程.抽象服务范例又可以根据不同的抽象粒度划分至不同的阶层.根据抽象粒度的不同,抽象服务范例可以分为领域服务层范例和应用层范例.领域服务层的范例是由领域专家通过对本领域的需求进行分析,结合行业背景特点以及以往的服务成功的经验归纳和总结出一些规范.根据这些规范及优化规则,领域服务层的范例抽象和封装了领域内业务服务逻辑及组合逻辑,并且通过服务范例的抽象化[2]将 1 个或者多个具体服务范例映射为一个领域服务类范例.

领域服务层范例可以如下表示:

*DC*:(*Name*,*Description*,*Solution*):

*Name*:领域服务层范例的名称;

*Description*:(*NameList*,*ServiceDescription*,*AInfo*):

*NameList*:(*serviceName*$_1$,*serviceName*$_2$,*serviceName*$_3$,…,*serviceName*$_n$) // *NameList* 表示领域服务名称列表,*serviceName*$_n$ 表示领域服务名称;

*ServiceDescription*:{*textDescription*,(*Input*,*Output*,*Constraint*)} // *ServiceDescription* 表示服务描

述,比如输入、输出参数及约束条件等:

*textDescription* 表示文本描述;

*Input* 表示输入属性;

*Output* 表示输出属性;

*Constraint* 表示用户对服务组合的约束条件,*Constraint* 包括 *Pre-contraint*,*Post-constraint*,*otherConstraint*.其中,*Pre-contraint* 表示服务行为前置条件,*Post-constraint* 表示服务行为后果,*otherConstraint* 表示其他约束条件,如时间、价格、可靠性(成功次数与总次数的比率)、用户偏好等;

*AInfo*:附加信息,如最近的调用信息、版本号、反馈的服务满意度记录等信息;

*Solution*:领域服务层范例的求解部分[2].

应用层范例是由 1 个或者多个领域服务层范例抽象而成,并且根据以往的成功经验确定领域层范例间的逻辑关系.应用层范例通常描述一个完整的应用.应用类范例与领域类范例结构类似,只是抽象程度不同,受篇幅所限,本文不再详细说明.

具体服务范例为最底层的范例,其求解部分可以是要完成目标计划的 Web 服务的 WSDL 文件.问题描述部分由一组与求解部分相关的特征组成,其叙述是以任务导向的方式,涉及到所应用的领域范围.一个具体服务范例的求解部分可以被定义为一个三元组:$CC=\{SC,SA,SP\}$,其中:*SC* 代表 Web Services 的服务能力,描述了服务提供的功能;*SA* 代表 Web Services 的具体访问技术细节,描述了客户端与服务的消息交换协议、服务的底层绑定协议等;*SP* 代表 Web Services 的服务属性,描述了服务的提供方等相关信息.

### 1.2 可适应的服务范例相似度测量与选择

考虑到服务组合的复杂性,从范例库中查找到的服务范例不一定能够完全满足用户需求,往往需要进行适应性调整.这就要求查找到的范例不仅相似度较高,而且还要易于适应性调整.但是,当前的许多相似度测量算法并不考虑服务范例的可适应性,使得查找出来的范例不易调整或者无法调整,大大降低了服务组合的可适应性和准确性.针对该问题,本节在兼顾效率和准确性的基础上提出一种考虑服务范例可适应性的相似度测量算法.

为了满足服务范例的相似度算法的准确性和可调整性的要求,我们提出一种 3 层次服务范例相似度测量算法.首先定义以下相似函数:

基本描述:采用元素 *serviceName* 和 *textDescription* 分别定义 Web 服务名称和文本描述;

设 *ServiceSet* 是范例 *C* 包含的服务集合,*ServiceSet′* 是范例 *C′* 所包含的服务集合.

基本相似函数:

$$Sim_{base}(C,C')=\frac{|\{S_i\,|\,(S_i\in ServiceSet)\cap(S_i\in ServiceSet')\}|-\rho\,|ServiceSet'-\{S_i\,|\,(S_i\in ServiceSet)\cap(S_i\in ServiceSet')\}|}{|ServiceSet|},\ \rho<1 \quad (1)$$

其中,$\rho$是松弛系数,$Sim_{base}(C,C')$针对基本描述进行相似度计算,体现服务请求者对服务基本描述的重视程度.

在实际的服务组合过程中,在使用相似度测量算法来选取服务范例需要调整的情况下,基本相似度最高的范例往往不易进行适应性调整或者无法调整,这时,需要在相似度测量算法中考虑输入-输出与约束条件描述的因素.

输入-输出描述:通过元素 Input 和 Output 描述类型和方向来定义服务功能.设 $e^p,g^q$ 是分别来自本体 $p$ 和 $q$ 的实体类,$Sim_\alpha,Sim_\beta$和 $Sim_\chi$分别表示本体类基于同义词、属性和语义关系的相似函数,$\varpi_\alpha,\varpi_\beta$和$\varpi_\chi$是权值,则 $e^p,g^q$ 的实体类相似函数为

$$Sim(e^p,g^q)=\varpi_\alpha\times Sim_\alpha(e^p,g^q)+\varpi_\beta Sim_\beta+\varpi_\chi Sim_\chi,\varpi_\alpha,\varpi_\beta,\varpi_\chi\geq 0 \quad (2)$$

基于 Tversky[11]模型和 Andrea[12]相似函数及集合论,设 $E,G$ 是 $e^p,g^q$ 本体类描述集合(同义词集、属性和语义关系集),$\cap$,| |分别是集合的并、模运算,$\mu$是非公共特性的相对重要程度,则 $Sim_\alpha,Sim_\beta$和 $Sim_\chi$通过下式计算:

$$Sim_{\alpha,\beta,\chi}(e^p,g^q)=\frac{|E\cap G|}{|E\cap G|+\mu|E/G|+(1-\mu(e^p,g^q))|G/E|},0\le\mu\le1 \tag{3}$$

输入-输出描述的相似度测量需要计算输入、输出类型的相似度,可如下定义输入-输出相似函数:

$$Sim_{in\text{-}out}(C,C')=\lambda_1\times Sim_{in}(C.input,C'.input)+\lambda_2\times Sim_{out}(C.output,C'.output),\ \sum_i\lambda_i=1,\ 0\le\lambda_i\le1,\ i=1,2 \tag{4}$$

其中,$Sim_{in}(C.input,C'.input)$,$Sim_{out}(C.output,C'.output)$分别是两个服务关于输入集(input set)、输出集(output set)的相似函数,$\lambda_i$ 是权值.可以通过公式(3)和以下的递归公式(5)、公式(6)来计算:

$$Sim_{in}(C.input,C'.input)=Sim_{in}(C.input-e^{C.input},C'.input-g^{C'.input})+\lambda_i\times Sim_{in}(e^{C.input},g^{C'.input}) \tag{5}$$

$$Sim_{out}(C.output,C'.output)=Sim_{out}(C.output-e^{C.output},C'.output-g^{C'.output})+\lambda_i\times Sim_{in}(e^{C.output},g^{C'.output}) \tag{6}$$

其中,$\sum_i\lambda_i=1,0\le\lambda_i\le1$.

约束条件描述:约束条件(constraint)包括 *Pre-contraint*,*Post-constraint*,*otherConstraint*.其中,*Pre-contraint* 表示服务行为前置条件,*Post-constraint* 表示服务行为后果,*otherConstraint* 表示其他约束条件,如时间、价格、可靠性(成功次数与总次数的比率)、用户偏好等.约束条件相似函数是相似元的数量以及每个相似元对系统相似度影响权系数等因素的函数.设服务范例 $C$ 的问题描述部分(description part)中的约束条件部分由 $m$ 个特征项(feature)组成,服务范例 $C'$的问题描述部分(description part)中的约束条件部分由 $n$ 个特征项组成,服务范例 $C$,$C'$之间存在 $k$ 个相似特征项,构成 $k$ 个相似元,每个相似元的值记为 $sim(f_i^X,f_i^Y)$.每一相似元对相似系统相似程度的影响权重为$\omega_i$,则系统 $A$ 与 $B$ 的约束条件相似函数为

$$Sim_{constraint}(C,C')=\frac{k}{m+n-k}\sum_{i=1}^{k}\varpi_i sim(f_i^C,f_i^{C'}) \tag{7}$$

在该式中,$m$,$n$,$k$ 分别表示相似服务范例 $C$ 问题描述部分(description part)的约束条件部分和服务范例 $C'$问题描述部分(description part)的约束条件部分的各自组成特征项的数量和它们公有的相似特征项的数量,而 $\frac{k}{m+n-k}$ 项表示服务范例 $C$ 和 $C'$之间问题描述部分的约束条件部分的相似特征项数量 $k$ 的多少对系统相似度的影响.将每个语义标注过的特征项都作为一个概念,其对应的文本信息作为一个实例.语义范例相似度算法是遍历树型结构的范例,每一棵树表示类层次的本体元素,如果 $f_i^{X'}$ 与 $f_i^{Y'}$ 前 $j$ 个父类相同,则相似度 $sim'(f_i^{X'},f_i^{Y'})=1$;如果 $f_i^{X'}$ 与 $f_i^{Y'}$ 前 $j$ 个父类不同,即从叶子节点到其父节点无匹配项,则相似度 $sim'(f_i^{X'},f_i^{Y'})=0$.相似度计算公式为

$$sim(f_i^X,f_i^Y)=\sum_{i=1}^{b}W_i l_i\Big/\sum_{i=1}^{a}W_i l_i \tag{8}$$

其中,$W_i$ 表示第 $i$ 层的权重,$\sum_{i=1}^{b}W_i l_i$ 表示选定的遍历路径上匹配的节点深度 $b$ 的加权和;$\sum_{i=1}^{a}W_i l_i$ 表示选定的遍历路径上节点的最大深度 $a$ 的加权和.除了考虑基本描述的相似性以外,为了使选取出来的范例更准确、更便于适应性调整,有时也需要输入-输出的相似度和考虑约束条件的相似度,同时可以作为服务范例选择与优化的依据,由此,我们给出服务范例相似度测量函数:

$$Sim(C,C')=\theta_1 Sim_{base}(C,C')+\theta_2 Sim_{in\text{-}out}(C,C')+\theta_3 Sim_{constraint}(C,C'),\ \sum_i\theta_i=1,\ 0\le\theta_i\le1,\ i=1,2,3 \tag{9}$$

其中,$\theta_i$ 是相似函数对应的权值,一般由领域专家来确定.$Sim_{base}(C,C')$,$Sim_{in\text{-}out}(C,C')$,$Sim_{constraint}(C,C')$分别是基本相似函数、输入-输出相似函数、约束条件相似函数.下面以例 1 为一个目标范例 $C$,例 2 为一个源范例 $C'$,说明如何使用函数(9)进行计算.

例 1:一个旅行领域服务范例,该服务的描述为:订一个从北京到上海的商务旅行的服务,并且要求乘坐飞机,当订飞机票服务成功后再订宾馆,总费用在 2 000~3 000 元以内,时间在 4 ~5 天以内.

例 2:用户只要求订一个从北京到上海的商务行程,包括开会、旅游等服务,并且要求乘坐飞机,当订飞机票服务成功后再订宾馆,总费用在 5 000 元以内,时间为 6 天.

(1) 基本相似度:采用函数式(1),设$\rho$=0.1,可以得到 $Sim_{base}(C,C')$=0.96.

(2) 输入-输出相似度:设$\lambda_i$=0.5,根据函数式(2)、函数式(3)、函数式(5)得到 $Sim_{in}(C.input,C'.input)$=0.71;根据函数式(2)、函数式(3)、函数式(6)得到 $Sim_{out}(C.output,C'.output)$=0.92;根据函数式(4)得 $Sim_{in\text{-}out}(C,C')$=0.815.

(3) 约束条件相似度:根据函数式(7)、函数式(8)得到 $Sim_{constraint}(C,C')$=0.7016.

(4) 服务范例的相似度:设$\theta_1$=0.5,$\theta_2$=0.25,$\theta_3$=0.25,根据函数式(9)及函数式(1)~函数式(3)的结果得到:

$$Sim(C,C')=\theta_1 Sim_{base}(C,C')+\theta_2 Sim_{in\text{-}out}(C,C')+\theta_3 Sim_{constraint}(C,C')=0.85915.$$

在我们的原型中,我们采用阶层式相似度计算的算法,如果某一层次的目标范例与源范例相似度超过某一重用门限值 $S$,则确认源范例为可重用范例,此时,源范例的求解部分(solution)中就包含下一层的范例.依此类推,直至具体服务层,最终返回的结果为可执行的服务组合.对于相似度接近但未达到重用门限值的源范例,算法终止,则确认为不可重用范例,将该范例及该范例所对应的上一层范例返回给范例推理处理器,判断是否重用该抽象服务,或者提交调整运算子进行适应性调整.当出现相似度相近的多个可重用范例时,通过比较服务记录中最近一段时期内成功服务的次数来确定目标范例.例如,用户需要旅行服务,包括订飞机票服务和订宾馆服务,如果没有其他约束条件,先订飞机票后订宾馆的范例与先订宾馆后订飞机票的范例的相似度是一样的,这时就需要比较服务范例附加信息(AInfo)中的范例的调用记录.我们定义用户的非期待值,成功次数越高的组合服务范例表明其用户非期待值就越低,即被用户接受的可能程度越高;反之则越低.具体的选择算法如下:

首先定义服务范例的用户非期待值:$\eta$,初始值$\eta_0(Case)$为 1.当该范例第 $n$ 次调用后,第 $n$+1 次调用成功,则$\eta_{n+1}(Case)=\eta_n(Case)-\alpha^{successNumber}$.其中,$\alpha\in(0,1)$是加强因子,$successNumber$ 是调用成功的次数.如果该范例第 $n$ 次调用后,第 $n$+1 次调用失败,则$\eta_{n+1}(Case)=\eta_n(Case)+\alpha^{failureNumber}$,其中,$failureNumber$ 是 $n$ 次调用中失败的次数.对于包含 $n$ 个子范例的范例,其用户非期待值为 $\eta(Case)=\sum_{i=0}^{n}\eta_i(Case)$ .由于最近时间内的调用记录中连续的成功次数或者失败次数对服务范例的选择至关重要,算法突出了最近一段时间内 Web 服务的连续成功调用或者服务的连续失败调用对服务质量的影响.算法定义的公式在连续成功或者失败后,对服务范例用户非期待值的调整量呈指数方式衰减.

### 1.3 基于调整运算子的阶层式服务范例适应性调整

根据目标及计划的阶层式表示方式及分析方法,我们可以将目标分解成 1 个或数个子目标.以一个旅游服务的目标为例,我们可以将一个旅游服务(traval)目标分成 3 个子目标,分别为 Book_Airline_Ticket 服务、Book_Hotel 服务和 Bank_Payment 服务,每个服务对应一个具体范例.在有新的服务请求需要被执行以前,根据新请求描述的特征属性,系统会先从范例库中选取出与目前最相似于新请求的服务范例.在执行服务范例选取的程序时,会执行范例索引及相似度测量的动作来选择并选取出最相似也最适合调整的服务范例[2].

#### 1.3.1 调整运算子

调整运算子是将已选取出的服务范例调整成能够解决当前 Web 服务组合问题的新服务范例的方法,它封装了服务范例调整的一系列内部操作,成为一个与外部环境自主通信的实体.通过这种方式,一个服务范例调整的程序就成为服务范例处理实体与调整运算子一连串的交互程序,通过与调整运算子的交互决定服务范例调整的结果.我们所采用的调整运算子操作的概念是将服务范例调整的程序分成数个交互通信过程.首先,从范例库中选取出相似度较高且易调整的服务范例,评估与待解决问题之间描述的差异部分,调整运算子根据差异的类型执行不同的操作,所以,调整运算子只会响应正确的调整请求.换而言之,当调整运算子执行操作时,一定会先去判断前提条件是否被满足,当前提条件被满足时,调整运算子将会执行相应的调整操作,通过一连串的调整操作来完成服务范例调整的程序,直到调整成满足待解问题的新服务范例为止.这种过程是一种自主通信,即为了完成调整自身功能与交互环境中的其他实体(比如知识库)进行知识传递的交互.

采用调整运算子的优点在于,便于对服务范例调整过程建模实现调整需求驱动的服务范例的自动调整.通过分析调整运算子的交互环境,可以定义交互环境本体,并在此基础上给出调整运算子能力描述框架,在此基础上建立调整运算子的概念化能力到形式化的进程表示的自动转换,从而实现并且验证服务范例的自动调整.

1.3.2 基于调整运算子的阶层式调整

在阶层式调整是以各个子服务范例对映的执行步骤分别作调整的程序,最后整合成所需要的一个完整的调整执行步骤.

如图 1 所示,对一个新的服务请求我们视其为一个新的问题(实线椭圆),而虚线的部分为完成请求所需要的子服务范例(虚线椭圆)及步骤(虚线方形).根据新问题的描述特征,从范例库和知识库中分别选取出一个最相似的服务范例及其对应的子服务范例和确定相应的调整运算子和调整规则.调整运算子采取相应的动作完成服务范例的调整.以下为服务范例调整的步骤:

(1) 接收调整请求.调整运算子收到调整请求以后,首先比较由用户服务请求生成的请求范例(目标范例)与范例库内选取出来的最佳服务范例之间是否存在特征项差异.如果存在特征项差异,则判断新的目标范例与所选取的抽象服务范例描述部分所对应的差异特征项的差异类别,调整运算子调用相应的调整操作响应调整请求.

(2) 对待调整的服务范例执行删除操作.将需要调整的抽象服务范例求解部分中的无效解删除,同时将其所对应的具体范例进行删除.需要说明的是,此时的抽象服务范例是通过可适应性的相似度测量算法从范例库中选取出来的相似度最大且需要调整的服务范例,该范例求解部分的解是抽象解.

(3) 与知识库交互.根据特征差异类型、差异特征项及知识库所对应的解答,确定待调整的抽象范例求解部分中需要置换的解.

(4) 与范例库交互.以需要置换的解为特征执行增加操作,将适合的子服务范例从范例库中选取出来.

(5) 对待调整的服务范例执行置换操作.根据新目标所描述的特征执行置换操作,将所选取的子服务范例的求解部分进行置换的动作.

(6) 对待调整的服务范例求解部分整合,将适合的范例(具体范例或抽象范例)整合进执行步骤中.每个子目标的调整方式都相同,当每一个需要调整的子目标都完成调整的程序时,即完成整个执行步骤的调整.

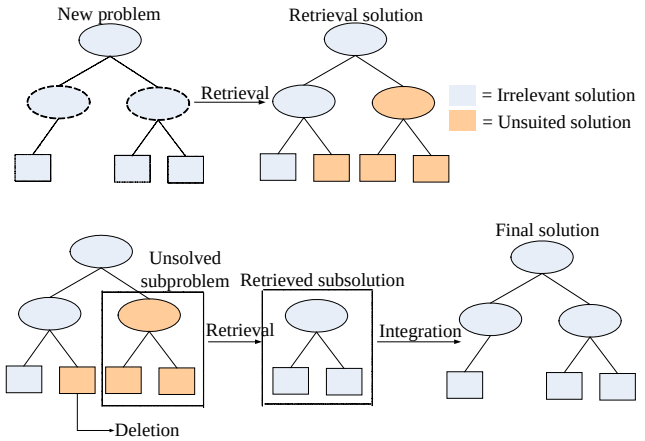


Fig.1 Hierarchical revision

图 1 阶层式调整

图 2 反映了抽象服务范例调整的过程.由于调整运算子的调整过程是一个与周围交互环境自主通信的过程,基于这个特点,我们提出在交互本体的基础上对调整运算子的能力进行描述,由此构建调整运算子能力框架,并由该框架导出半形式化的服务范例调整状态迁移图.这样就可以将修改运算子的操作用现成的遍历图算法自动完成,从而实现服务范例适应性调整.

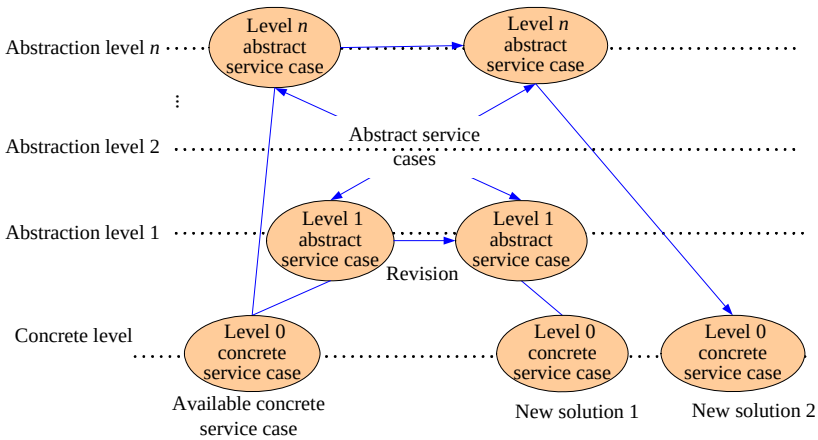


Fig.2 Procedure of the hierarchical revision of abstract service case

图 2 抽象服务范例的阶层式调整过程

1.3.3 基于本体的调整运算子能力描述

交互环境是一个相对的概念,是实现交互行为的现实世界,软件行为的后果可以在它的交互环境中被观察和评估[13].对于一个调整运算子来说,其交互环境中的行为描述就可以通过描述自己的行为来预测.接受事件也是同样的道理,可以通过调整运算子对外部事件的响应来进行预测和描述.

调整运算子与环境的交互包括调整运算子与当前服务范例处理实体之间的交互和调整运算子与其他资源之间的交互.调整运算子与当前服务范例处理实体之间的交互是为了调整当前的服务范例与发出调整请求的服务范例处理实体之间进行知识的传递;而调整运算子与其他资源之间的交互则是指调整运算子主动获取知识库的信息或者改变范例库资源的状态.由于这些资源不是自主的实体,调整运算子对资源的行为需要通过与该资源的功能主体进行交互才能完成.调整运算子对交互环境中资源的行为也可以归结为范例推理系统中资源操作实体之间的交互.

本节中的交互采用述行成分(performative)表示,每个操作至少包含 4 个参数,按如下格式定义:

(performative

: content

: handler

: communication

: precondition

: effect

: ontology)

其中,content 表示交互的内容,可以有两种形式:一种是 description,表示交互的是服务范例的描述部分或者知识库的规约描述;另一种是 solution,表示交互的是服务范例的求解部分.handler 表示执行该动作的执行者,是一个交互环境中的自主通信的实体;communication 表示可以与所描述动作进行通信的动作,它对与服务范例之间行为的交互进行了约束;precondition 表示执行当前动作的前因,此参数是可选的;effect 则表示执行当前动作的后果,此参数是可选的;ontology 描述行为概念所属的本体,只有在同一个本体约束下,概念的匹配才有意义.

本文的顶层交互本体[2]包含两类述行成分:一类是输入型(input),表示调整运算子从交互环境中获取信息;另一类是输出型(output),表示调整运算子向交互环境中输出信息.其中,Input 类型的操作包括:Receive,Get,Retrieve 和 Demand.顶层交互本体中,Output 类型的操作包括:Send,Request,Return 和 Provide.以 Retrieve 为例说明述行成分的表示:

(Retrieve

: content description

: handler OperatorA

: communication {Request}

: effect {Return}

: ontology http://cbrframeworkhost:33125/IntrOntology/Revision.owl)

表示 Retrieve 是 Revision.owl 本体[2]中的概念,由交互实体 OperatorA 执行,接收对方服务范例执行实体发出的查询新的服务范例求解部分的请求,将以 Return 发出信息实现对 Request 的响应.

在对服务范例的调整过程中,交互环境中的信息和实体资源的状态迁移 $Trans_1$,$Trans_2$ 之间存在以下关系:

**定义 1(顺序关系(SQR))**. SQR 定义了状态迁移发生的必然的前后顺序关系.

**定义 2(竞争关系(RAR))**. PRD 定义了状态迁移发生的竞争关系,当交互资源到达某个状态时,可以向多个目标状态迁移,此时,哪个迁移的条件先被满足,下一步就将发生哪个迁移.调整运算子的交互环境中往往不是仅包含一种交互资源,各种资源状态迁移之间存在着相互制约,交互环境中的事件也会对迁移产生影响.

**定义 3(条件关系(CNR))**. 交互资源 $r_1$ 的迁移 $Trans_1$ 不仅依赖于本资源的迁移,还依赖于其他资源到达某个状态或者进行某个迁移.

**定义 4(触发关系(EVR))**. 迁移 *Trans* 的发生依赖于外部的触发事件.

1.3.4 调整运算子的状态迁移图

调整运算子的状态迁移图包含 4 种图元:椭圆形表示变迁结点,(*r*::*s*:*a*,*b*)表示交互资源 *r* 的属性 *s* 的值从 *a* 变迁到 *b*;圆圈表示事件结点;结点间的连接边表示结点间的关系,包括 4 种:*SQR*,*RAR*,*CNR*,*EVR*,所以,状态迁移图是一个有向标记图,边的方向表示关系之间的依赖与被依赖关系;虚框表示调整运算子的边界,虚框里面的结点是由该调整运算子产生的,包括操作资源的变迁和调整运算子触发的事件;虚框外面的结点中,是由交互环境中的调整运算子产生的,包括访问资源的变迁和交互环境对该调整运算子的触发事件(在图中用大写单词表示).

图 3 是一个旅行服务范例适应性调整的状态迁移图.状态迁移图模型可以由我们定义的调整运算子能力描述框架[2]导出,框架中的迁移和事件对应状态迁移图中的结点,而事件和交互的描述形式蕴含了状态迁移图结点之间的依赖关系.这种半形式化的扩展的状态迁移图模型可以直接使用现成的图遍历算法来进行服务范例的调整操作,具体的图遍历算法不在本文的讨论范围内.

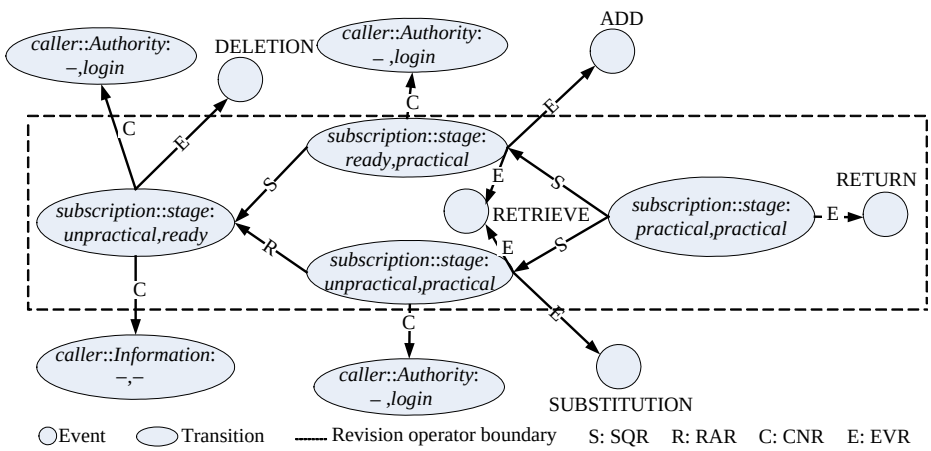


Fig.3 State transition graph of the substitution revision of travel service case

图 3 Travel service case 置换调整的状态迁移图

## 2 实验及对比分析

实验环境如下:CPU 为 Pentium IV 2.8GHz,内存为 512MB,操作系统为 Window XP SP2,开发语言采用 Java,运行的 Java 虚拟机为 J2SE 1.5.0_04,使用 Jena2.1.

**实验 1**. 与现有的基于范例推理的服务组合方法的比较实验.

为了检验本文所提出的服务组合方法 ARMSC(adaptable revision of multiple service case)的有效性,我们将 ARMSC 与现有的基于范例推理的服务组合方法 WeSCo_CBR[4],CBRE[5]进行比较.首先进行效率实验,由于没有标准的测试数据集,我们的实验选取 Internet 上 Web 服务实例,如 Google,Amazon,Global Weath 等知名服务共 5 个领域 100 个实例,根据这些实例构造源范例描述部分的特征项空间.为了便于比较,将动作、变量[4]均视为特征项.构造抽象服务范例,控制抽象服务的粒度,使其阶层数为 2.选取部分特征项构造目标服务范例.分别设置不同的服务范例相似度门限值 $S$[2]和每个服务组合包含服务的数量.比较在门限值和服务数量的不同条件下,3 种服务组合方法的成功率和效率.以运行 100 个实例的平均成功率作为成功率比较的依据,设置每次服务组合时间上限 30 000ms,以在时间上限内完成服务组合的平均时间作为效率比较的依据.

图 4 给出了与现有的基于范例推理的服务组合方法比较实验的结果.从实验结果可以看出,随着每个服务组合所包含的服务数量的不断增加,WeSCo_CBR 和 CBRE 的响应时间增加得最快,ARMSC 的响应时间最短.随着门限值 $S$ 的升高,WeSCo_CBR 和 CBRE 的成功率下降得最快,ARMSC 的成功率最高.这是由于 WeSCo_CBR 和 CBRE 这两种基于范例推理的服务方法都不具备适应性调整能力,因此,在服务场景比较复杂的情况下,成功率不高并且响应时间较长.

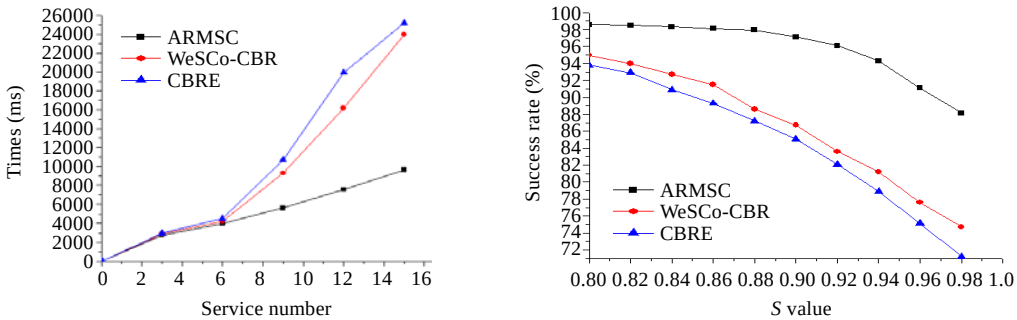


Fig.4 Experimental results about service composition, varying the value of three different current methods based CB

图 4 现有的 3 种基于范例推理的不同服务组合方法实验结果

**实验 2**. 与其他服务组合方法的成功率的比较实验

由于没有标准的测试数据集,我们构造特征项空间,采用随机生成的模拟抽象服务、具体服务及服务之间的相似度作为测试用例,进行多次仿真实验.为了突出对比内容,实验基于以下假设:

所有的服务请求都能找到对应的抽象服务范例和具体服务范例;不考虑服务执行引擎执行服务组合逻辑的时间;不考虑网络传输时间.

随机地从服务范例库的服务组合规划层抽象范例特征项空间中内选取不完全一致、包含若干个特征项的 100 个抽象服务范例作为测试集.设置抽象服务范例的门限值 $S$ 为 0.97,抽象服务范例的阶层数为 2.再为每个抽象服务范例配置不同数目的具体服务,使得每个特征项对应于一个 Web 服务.范例库所包含的服务总数分别为 200,400,600,800,1000.具体服务范例的用户非期待值设置为区间(1,2)的随机数.忽略 Web 服务之间除顺序执行关系以外的其他时序关系.分别做 100 次实验,取其平均成功率.

图 5 给出了 3 种不同方法进行服务组合的平均成功率.本文的方法 ARMSC 成功率大于基于关键字匹配的服务组合方法(KC)及基于服务语义的服务组合方法(SC).

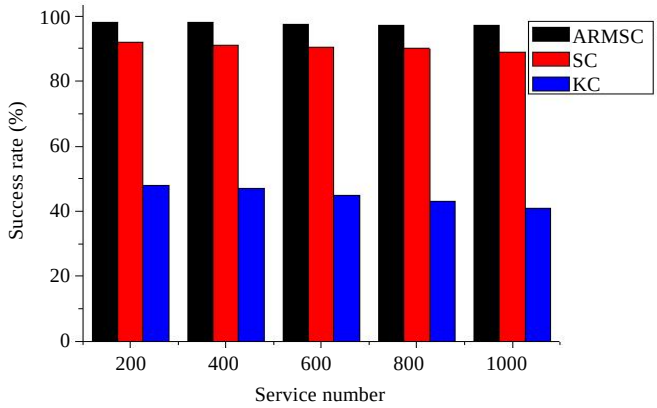


Fig.5 Experimental results about service composition success rate, varying the value of three different three methods

图 5 3 种不同的服务组合成功率实验结果

## 3 结 论

我们提出使用具有较好的可适用性的抽象阶层式服务范例来实现服务组合,提高了服务组合的成功率和服务范例的可适用性.与一般方法相比,我们的方法有更好的实用意义:

(1) 抽象阶层式的服务范例可以在较高的应用层面上构造服务组合,这种大粒度的构造方法避免了在较低的具体服务层面上从头构造的麻烦,减少了服务组合的代价和人工的参与程度.使用可适应的服务范例相似度测量方法,在服务范例选择时考虑了可适应性的需求,降低了服务范例调整的难度,提高了适应性调整的成功率.

(2) 与基于模板的服务组合方法[14]相比,在服务组合的过程中降低了人工参与程度,提高了服务组合的自动化程度,能够有效地利用成功服务组合的经验,克服在知识缺乏或知识不良定义的情况下用户设计正确的应用的困难,从而提高了应用的可靠性.

(3) 与其他基于范例推理的服务组合方法相比,使用了语义范例描述语言和可适应的服务范例相似度测量与选择方法,提高了范例的描述能力、服务范例查询的准确率和效率以及可适应性.基于调整运算子的服务范例适应性调整方法的提出和使用,提高了服务范例的可重用性和可适应性,提高了处理复杂服务组合的能力.

(4) 随着服务成功次数的增加,服务范例库内能够满足用户需求的范例也不断增加,能够提高服务组合的成功率.

目前,我们的工作还有待完善,今后的工作包括将范例推理与归纳算法相结合,加强系统的学习能力;提高服务范例适应性调整的自动化程度;完善服务范例的抽象化算法和服务范例的动态更新机制.

**References**:

[1] Srivastava B, Koehler J. Web service composition: Current solutions and open problems. In: Proc. of the Int'l Conf. on Automated Planning & Scheduling (ICAPS) 2003 Workshop on Planning for Web Services. 2003. 28−35.

[2] Cheng RX. Research on services composition supported by case-based reasoning [Ph.D. Thesis]. Beijing: Beijing University of Posts and Telecommunications, 2007 (in Chinese with English abstract).

[3] Cheng RX, Su S, Yang FC. Using case-based reasoning to support Web service composition. In: Proc. of the 6th Int'l Conf. on Computational Science (ICCS 2006), Part IV. LNCS 3994, Reading: Springer-Verlag, 2006. 87−94.

[4] Lajmi S, Ghedira C, Ghedira K, Benslimane D. WeSCo_CBR: How to compose Web services via case based reasoning. In: Proc. of the IEEE Int'l Conf. on e-Business Engineering (ICEBE 2006). 2006. 618−22.

[5] Limthanmaphon B, Zhang YC. Web service composition with case-based reasoning. In: Proc. of the 14th Australasian Database Conf. Adelaide: Australian Computer Society, 2003. 201−208.

[6] Leake DB, Kinley A, Wilson D. Learning to improve case adaptation by introspective reasoning and case-base reasoning. In: Proc. of the 1st Int'l Conf. on Case-Based Reasoning. 1995. 229−240.

[7] Bergmann R, Wilke W. Building and refining abstract planning cases by change of representation language. Journal of Artificial Intelligence Research, 1995,3(1):53−118.

[8] Leake DB, Kinley A, Wilson D. A case study of case-based CBR. In: Proc. of the 2nd Int'l Conf. on Case-Based Reasoning. 1997. 371−382.

[9] Gao JJ, Deng GS. A semantic Web-based case representation and retrieval. In: Proc. of the 6th World Congress on Intelligent Control and Automation. 2006. 4335−4339.

[10] Li M, Wang DZ, Du XY, Wang S. Dynamic composition of Web services based on domain ontology. Chinese Journal of Computers, 2005,28(4):644−650 (in Chinese with English abstract).

[11] Tverskey A. Features of similarity. Psychological Review, 1977,4(84):327−352.

[12] Rodríguez MA, Egenhofer MJ. Determining semantic similarity among entity classes from different ontologies. IEEE Trans. on Knowledge and Data Engineering, 2003,15(2):442−456.

[13] Hou LS, Jin Z, Wu BD. Modeling and verifying Web services driven by requirements: An ontology based approach. Science in China (Series E: Information Sciences), 2006,36(10):1189−1219 (in Chinese with English abstract).

[14] Hu HT, Li G, Han YB. An approach to business-user-oriented larger-granularity service composition. Chinese Journal of Computers, 2005,28(4):694−703 (in Chinese with English abstract).

**附中文参考文献:**

[2] 成睿星.基于范例推理的服务组合的研究[博士学位论文].北京:北京邮电大学,2007.

[10] 李曼,王大治,杜小勇,王珊.基于领域本体的 Web 服务动态组合.计算机学报,2005,28(4):644−650.

[12] 侯丽珊,金芝,吴步丹.需求驱动的 Web 服务建模及其验证:一个基于本体的方法.中国科学(E 辑:信息科学),2006,36(10):1189−1219.

[13] 胡海涛,李刚,韩燕波.一种面向业务用户的大粒度服务组合办法.计算机学报,2005,28(4):694−703.

**成睿星**(1977－),男,河南洛阳人,博士,主要研究领域为下一代网络,软交换技术,移动通信技术.

**杨放春**(1957－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为通信软件,智能网,下一代网络.

**苏森**(1971－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布式计算.