

基于动态 QoS 的 Web 服务组合

孔维梁 刘清堂 杨宗凯 韩淑云

(华中师范大学国家数字化学习工程技术研究中心 武汉 430079)

摘 要 在 Web 服务组合中,现行的几种 QoS 衡量标准都将重点放在单个 Web 服务本身的质量上,而忽视了 Web 服务动态特性、组合特性以及服务组合中的网络特性。另外,在诸多服务组合的算法中,都只是强调组合服务的总体质量,却忽略了用户对某些质量属性的约束条件,从而导致服务重计算问题经常发生。为此,考虑了服务动态特性以及服务间的协作关系对组合服务质量的影响,提出了动态 QoS 模型;同时,综合了用户的质量约束以及组合服务的整体质量,将用户的质量约束引入服务组合流程中。最后通过实验证实了所提出的动态 QoS 模型能够根据服务实体的实时情况计算服务质量,同时将用户的质量约束引入服务组合流程中,有效地避免了服务重计算问题。

关键词 Web 服务,动态 QoS,Web 服务组合,服务质量,遗传算法

中图法分类号 TP181 **文献标识码** A

Composition of Web Services Based on Dynamic QoS

KONG Wei-liang LIU Qing-tang YANG Zong-kai HAN Shu-yun

(National Engineering Research Center for E-learning, Huazhong Normal University, Wuhan 430079, China)

Abstract In composition of Web services, several current standards of QoS focus on the quality of independent Web services, and ignore the dynamic, compositional, network-savvy characteristics of Web services. In addition, many combination algorithms emphasize the overall quality of compositional services and ignore the constraint conditions of users. This results in the recalculation of composition of Web services. Therefore, considering the dynamic characteristics of the service and the efforts of collaboration between the services, the author put forward a dynamic standard of QoS. At the same time, the author also took into account the constraint conditions of users and the overall quality of compositional services and applied the constraint conditions of users in the composition process. Finally, we proved that the dynamic QoS standard can compute the real service quality of service entity and that applying the constraint conditions of users in the composition process can avoid the service re-computing effectively.

Keywords Web service, Dynamic QoS, Composition of Web services, QoS, Genetic algorithm

1 引言

在互联网环境中,存在大量的 Web 服务被发布和共享,同时存在着数量众多的功能相同、模型结构相同而服务质量(QoS)各异的服务^[1]。如何能够根据用户需求,将分布于网络上的单个 Web 服务自动、有效地组织起来,正成为一个迫切需要解决的问题。在 Web 服务组合中,首先面临的问题就是 QoS 的衡量。国内外许多研究中都提出了有关评价 Web 服务质量的思路,强调服务 QoS 的重要性^[2-4]。但这些 QoS 模型都将重点放于单个 Web 服务的质量,而忽视了服务动态特性和网络特性;其次,在诸多服务自动组合的算法中^[5,6],都只是强调组合服务的总体质量,却忽略了用户对某些质量属性的约束条件。

针对以上问题,本文以 Zeng^[3]提出的模型为基础,综合

考虑了服务动态特性以及服务间的协作关系对组合服务质量的影响,提出了一个动态 QoS 模型;同时,综合了用户的质量约束条件以及组合服务整体的质量,将用户的质量约束引入服务组合流程,使得服务组合结果既满足用户的质量需求,有效避免了服务重计算问题,又使得组合服务具有较高的总体质量。本文第 2 节在总结前人经验的基础上,分别从基本服务和组合服务的角度描述了动态 QoS 模型;第 3 节介绍了基于动态 QoS 的服务组合,同时在组合过程中引入了用户约束条件;第 4 节给出了实验验证以及实验分析;最后是结论并指出下一步的研究内容。

2 动态 QoS 模型

在 Web 服务组合中,现行的几种 QoS 衡量标准都将重点放于单个 Web 服务本身的质量,对于组合服务质量只是单

到稿日期:2011-03-09 返修日期:2011-06-03 本文受核高基重大专项(国产基础软件在数字教育领域的适配研究及重大应用示范,2010 ZX01045-001-005-3),博士后科学基金(构建数字教育语义网的若干关键技术研究,中博基字[2009]13号),国家科技支撑计划(音乐数字化服务关键技术与示范应用,2009BAH51B00)资助。

孔维梁(1985-),男,博士生,主要研究方向为数据挖掘、人工智能,E-mail:byta100@yahoo.cn;刘清堂(1969-),男,教授,博士生导师,主要研究方向为知识服务、人工智能、数字版权;杨宗凯(1963-),男,教授,博士生导师,主要研究方向为网络教育标准与技术、现代通信网络、智能信号处理;韩淑云(1982-),女,硕士生,主要研究方向为知识服务。

个服务质量的简单求和或取极值,从而忽视了 Web 服务动态特性、组合特性以及服务组合中的网络特性。为此,本文在前人经验的基础上,提出了动态 QoS 模型,综合考虑了服务动态特性以及服务间的协作关系对组合服务质量的影响。以下将分别从基本服务 QoS 和组合服务 QoS 两个方面描述动态 QoS 模型。

2.1 基本服务动态 QoS 定义

对于单个 Web 服务,考虑 4 种 QoS 参数,即时间 T (time)、费用 C (cost)、可靠性 Rel (reliability)、信誉度 Rep (reputation),在计算时考虑了服务动态特性和网络特性。参数的意义及计算方式见表 1。

表 1 基本服务动态 QoS 计算公式

质量	说明	计算公式
QoS _{tim}	服务请求从发送到完成时间	QoS _{tim} = T _{trans} + T _{wait} + T _{proc}
QoS _{cs}	调用 Web 服务的相关费用	由服务发布者提供
QoS _{rel}	服务能正确执行的程度	QoS _{rel} = w ₁ * rel _{Δt} + w ₂ * rel _{old}
QoS _{rep}	服务的可信程度,取决于用户的使用情况	QoS _{rep} = w ₁ * rep _{Δt} + w ₂ * rep _{old}

表 1 中,服务传输时间 T_{tran} 可以通过该服务执行的以往记录来确定,比如取几次传输时间的平均值;服务等待时间 T_{wait} 指服务从到达服务提供者端到服务开始执行的时间,由服务提供者端通过计算当时状态给出。一般而言,对于服务提供者端,处于平衡系统时,系统的输入应该等于输出,即 $I=O$, Δt 时间内,服务到达率=服务输出率, T_{wait} 计算公式如下:

$$T_{wait} = n * O_i; \text{当 } I < O$$

$$T_{wait} = \infty; \text{当 } I \geq O \text{ 且 } D \geq K$$

式中, n 为排在该服务之前的服务数, O_i 为服务端的输出率, D 为负载, K 为阈值;服务处理时间 T_{pro} 是由服务提供者给出;可靠性(或信誉度)计算中, w_i ($i=1, 2$) 为权值,分别代表最近 Δt 时间内服务执行和以往服务执行的可靠性(或信誉度)权重, $rel_{\Delta t}$ 与 rel_{old} 的计算如下:

$$rel_{\Delta t} = \frac{\sum_{i=1}^t Rel_i}{t}, rel_{old} = \frac{\sum_{i=1}^{old} Rel_i}{old}$$

若服务成功,则 $Rel_i = 1$, 否则 $Rel_i = 0$ 。同样地:

$$rep_{\Delta t} = \frac{\sum_{i=1}^t R_i}{t}, rep_{old} = \frac{\sum_{i=1}^{old} R_i}{old}$$

式中, R_i 是用户对服务的评价值。

2.2 组合服务动态 QoS 定义

组合服务的质量模型基于基本服务的 4 种质量模型,可以定义为 $QoS(CWS) = \{QoS_{tim}(CWS), QoS_{cs}(CWS), QoS_{rel}(CWS), QoS_{rep}(CWS)\}$ 。为描述方便,本文以 1~4 来代表 4 种基本质量属性,于是组合 Web 服务的质量模型也可以描述为 $QoS(CWS) = \{QoS_i(CWS)\} (1 \leq i \leq 4)$ 。对于不同的服务组合结构, $QoS_i(CWS)$ 计算并不相同。在 Web 服务组合中,大部分组合 Web 服务都可以由单个 Web 服务以顺序结构、并发结构、选择结构、循环结构 4 种基本模型^[7-9] 组合而成。表 2 给出了这几种基本组合结构对应的 $QoS_i(CWS)$ 计算方法。

在 4 种 $QoS_i(CWS)$ 中,有一些质量属性是增量型的,即其值越高,表示组合服务的质量越高,如服务可靠性 Rel 。一些质量属性是减量型的,即其值越低,表示组合服务的质量越

高,如费用 C 。为了能更为统一和方便地表示组合服务的质量,需要将 4 种质量模型进行规一化处理,用以消除组合服务 $QoS(CWS)$ 中各个质量属性变化趋势的差异性。本文用 QoS_i^k 表示第 k 个组合服务的第 i 个质量属性的值 ($1 \leq i \leq 4$),对于增量型的质量属性和减量型的质量属性分别用如下公式进行规一化^[10]。

表 2 组合服务动态 QoS(CWS)计算公式

组合模型	时间 T	费用 C	可靠性 Rel	信誉度 Rep
顺序结构	$T_{cws} = \sum_{i=1}^n T_i$	$C_{cws} = \sum_{i=1}^n C_i$	$Rel_{cws} = \prod_{i=1}^n Rel_i$	$Rep_{cws} = \frac{\sum_{i=1}^n Rep_i}{n}$
并发结构	$T_{cws} = \text{Max}(T_1, \dots, T_n)$	$C_{cws} = \sum_{i=1}^n C_i$	$Rel_{cws} = \prod_{i=1}^n Rel_i$	$Rep_{cws} = \frac{\sum_{i=1}^n Rep_i}{n}$
选择结构	$T_{cws} = \sum_{i=1}^n T_i \alpha_i$	$C_{cws} = \sum_{i=1}^n C_i \alpha_i$	$Rel_{cws} = \sum_{i=1}^n Rel_i \alpha_i$	$Rep_{cws} = \sum_{i=1}^n Rep_i \alpha_i$
循环结构	$T_{cws} = n * T_i$	$C_{cws} = n * C_i$	$Rel_{cws} = \prod_{i=1}^n Rel_i$	$Rep_{cws} = \frac{\sum_{i=1}^n Rep_i}{n}$

$$Nor(QoS_i^k) = \frac{QoS_i^k - \text{Min}_{k=1}^m(QoS_i^k)}{\text{Max}_{k=1}^m(QoS_i^k) - \text{Min}_{k=1}^m(QoS_i^k)}$$

$$\text{Max}_{k=1}^m(QoS_i^k) - \text{Min}_{k=1}^m(QoS_i^k) \neq 0, \text{ 否则 } Nor(QoS_i^k) = 1.$$

$$Nor(QoS_i^k) = \frac{\text{Max}_{k=1}^m(QoS_i^k) - QoS_i^k}{\text{Max}_{k=1}^m(QoS_i^k) - \text{Min}_{k=1}^m(QoS_i^k)}$$

$$\text{Max}_{k=1}^m(QoS_i^k) - \text{Min}_{k=1}^m(QoS_i^k) \neq 0, \text{ 否则 } Nor(QoS_i^k) = 1.$$

据上面的描述,组合 Web 服务的质量模型可以用下式表达:

$$QoS^k(CWS) = \sum_{i=1}^4 \omega_i * Nor(QoS_i^k)$$

式中, ω_i 为第 i 个基本质量属性的权重。

3 基于动态 QoS 的 Web 服务组合

3.1 Web 服务建模

为了消除服务组合过程中的语义冲突问题,本文将语义引入 Web 服务组合过程中。目前已有多种语言来描述 Web 服务(如 OWL-S^[11,12], SWIS, WSMO, SESMA 等)。本文以比较常用的,也是 W3C 推荐的 OWL-S 来描述 Web 服务。

定义 1 Web 服务可以定义为一个四元组 $WS = \langle P, M, C, Q \rangle$ 。其中 P 是服务的基本信息描述,它包括服务名称、简短描述、提供者信息等; M 是描述服务的模型,即描述服务是基本服务还是组合服务以及组合的方式; C 是服务功能描述,包括服务的接口参数、前置条件和后置条件以及服务执行之后的结果; Q 是服务的质量描述。

定义 2 服务的接口参数可以描述为一个二元组 $P = \langle I, O \rangle$,其中 I 是该服务的输入集合, O 是该服务的输出集合。

在本体中,有两个基本关系 sub-class-of 和 part-of:

sub-class-of: A sub-class-of B 表示 A 是 B 的子类。sub-class-of 具有传递性, C sub-class-of A , A sub-class-of B , 那么 C sub-class-of B , 此时 A 和 B 只能是抽象类。

part-of: A part-of B 表示 A 是 B 的一个部分或元素。

part-of 也具有传递性, A 可以是基本类、抽象类, 也可以是一个集合, B 只能是集合或抽象类。

定义 3(semi-belong-of, 语义属于) 假设有两个集合 A 、 B , 如果对于 A 中任意一个元素 a , 总能在 B 中找到一个元素 b , 使得 a, b 满足如下关系: $a=b$ 或 a part-of b 或 b sub-class-of a , 那么 A semi-belong-of B 。

本文使用到的集合操作中的属于均为语义属于。不属于亦类似。符号表述上使用常用集合表述符号, 然则意义却不尽相同。

定义 4(服务模型) 本文把从众多服务质量不同而功能相同、结构相同的服务中抽象出来的服务结构定义为服务模型, 把这些具体的服务称为该服务模型的候选服务。

3.2 用户质量约束条件

Web 服务组合可以描述为, 在保证用户质量约束要求的前提下, 使组合服务的整体质量最优。然而, 在诸多服务自动组合的算法中^[5,6], 都只是强调组合服务的总体质量, 却忽略了用户对某些质量属性的约束条件, 很难将用户约束条件应用于服务组合的过程中, 例如用户要求总的执行时间不超过 t 。一些基于图的组合算法中, 服务组合过程只考虑组合服务总的质量, 而没有考虑用户的约束。这样, 只有等到找到最优路径后, 才能知道该路径所表示的组合服务总执行时间是否超过 t 。但也有可能出现这种情况, 即该组合服务的整体质量很高, 执行时间却比较长。这时意味着, 这条路径有违用户约束要求, 需要标识该路径后, 重新再计算另一条路径, 这样效率会大大地降低。为此, 在服务组合过程中, 要全局考虑用户的质量约束条件。

设用户给定的组合服务质量约束要求为 R_i ($1 \leq i \leq 4$, 代表 4 种基本服务质量属性), Web 服务组合目标是找到一个组合服务 k , 在满足 $\frac{(-1)^N(QoS^k - R_i)}{R_i} \leq \alpha_i$ (α_i 为用户给出的质量 R_i 的容差度, 对于增量型质量 $N=-1$, 减量型质量 $N=1$) 的情况下, 使得 QoS^k 取得最大值。用户给出的质量约束要求可以是对所有质量属性的, 也可以是对部分质量属性的。例如, 用户要求组合服务总的执行时间小于 t , 总的费用也不大于 c , 然后在此条件下求得一个最优的组合服务。

3.3 服务组合流程

通常一个动态服务组合系统可以分为如下几个部分: 组合引擎、执行引擎、本体库、OWL-S 服务库、选择器, 如图 1 所示。系统工作时, 用户请求以及约束条件被发送到服务组合引擎。服务组合引擎针对这个需求通过领域本体库和服务选择器来寻找可用的服务。然后, 服务组合引擎根据用户的约束条件, 通过遗传算法对这些服务进行服务流程组合, 返回最佳的组合方案。随后, 服务组合引擎将此方案交给执行引擎执行, 最后将得到的结果返回给用户。整个组合过程对用户来说是透明的。

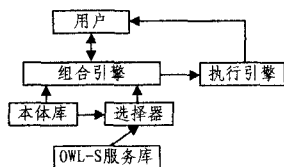


图 1 服务组合流程

在互联网环境中, 存在大量的 Web 服务被发布和共享, Web 服务组合首先要做的就是将可能有用的服务筛选出来, 这样能大大提高服务组合的效率。服务选择的总体思想是: 初始化一个集合 $AI=UI$ (UI 是用户输入集, UO 是用户要求输出集), 然后找出哪些服务的输入包含在 AI 内, 是用 semi-belong-of 进行的判断, 然后将这些服务的输出放入 AI , 进行下一次的匹配, 直到找不到更多的服务为止。算法如图 2 所示。

```

AI=UI, AO=∅, AS=∅
Repeat
  对于每一个服务 S, if(Ii semi-belong-of AI) && S ∉ AS
  {
    AI=AI∪Ii
    AO=AO∪Ii
    AS+=S
  }
Until 没有 S
If(UO semi-belong-of AO)
  执行组合算法
否则
  无解
  
```

图 2 服务选择算法

3.4 基于遗传算法的服务组合

在筛选出的服务中, 亦存在着数量众多的功能相同、模型结构相同而服务质量 (QoS) 等非功能特性各异的服务。如何在满足用户需求的条件下选取合适的服务进行组合, 众多学者做了不同深度的研究。一般有两种方法: 基于图的算法^[14]与遗传算法^[13]。然而, 在基于图的组合算法中, 对于用户提出的约束条件处理不足, 很难将用户约束条件应用于组合的过程中, 只有等到找到最优路径后, 才能知道是否符合用户的约束条件。相比之下, 遗传算法更适合解决这种带有全局约束条件的优化组合问题。为此, 本文也采用遗传算法来进行服务组合。

本文把服务模型 M 定义为基因位, 所有基因位一起构成了一个基因个体 (或称基因)。该基因的长度等于所有服务模型的个数 (包括起始服务和终止服务的模型), 每个基因位包含一个与之对应的候选服务集。由于每个基因的长度都相同, 因此本文采用整数定长编码的方式。为了方便表示用户的输入与要求输出, 本文引入了起始服务 S 和终止服务 E 。基因中第一个基因位总是起始服务 S , 最后一个基因总是终止服务 E 。基因中间每一个基因位对应一个服务模型, 其值为一个具体服务在该服务模型对应的候选服务集中的编号, 零表示该基因个体不包含此基因位, 也就是说该基因位对应的服务模型不参与此基因个体, 其结构如图 3 所示。在实际组合服务中, 包含的基因位数应该小于或等于总的基因位的总数。另外, 考虑到交叉和变异算子要表示选择型、并发型和循环型等服务组合类型, 以及个体是否是有效组合路径等问题, 本文采用文献^[7]的关系矩阵参与遗传操作。关系矩阵的主对角线元素表示个体, 其他元素表示任务间的位置关系。在进行交叉与变异操作时, 需要根据关系矩阵中任务间的位置关系检查新生成个体的路径合法性。遗传算法在解空间中进行并行全局搜索, 搜索所有组合路径 (基因个体) 中满足用户约束条件的一组最优解。其过程是将每一个组合路径编码

为一个基因个体,通过基因个体之间的交叉、变异等重组操作,产生具有更高适应度的新基因个体。然后,新基因个体继续经过选择、交叉、变异等操作,产生下一个新的个体。该过程不断进行,直至找到一个最优解或满足搜索约束。具体算法如图4所示。

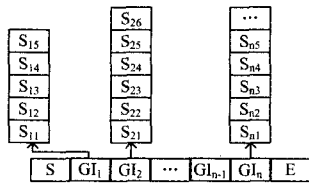


图3 基因结构

初始化种群 P 和进化代数 N

$t \leftarrow 0, best = P(0)$

$P'(t) \leftarrow \Phi$

While($t < N$)

P(t)适应度计算

$P'(t) \leftarrow \text{Intersect}(\text{Select}(P(t)))$

$P(t+1) \leftarrow \text{Evolove}(P(t), P'(t))$

Best = SelectBest(P(t+1))

$t \leftarrow t+1$

End

图4 遗传算法

其中,适应度计算需要基因编码与约束条件统筹兼顾。一般来说,由于所选服务构成的组合可能在某个方面会违背约束,编码可能对应着不可行解。为此,引入惩罚因子法,即将惩罚项添加到适应度函数中,从而迫使遗传搜索从解空间的可行和不可行两个方向接近最优解;选择操作采用轮盘赌选择策略,在选择时,将最优性方法嵌入轮盘赌选择来增强其在下一代中保持最优染色体的能力。交叉运算使用简单的单点交叉方法,变异采用随机扰动。由于初始种群采用随机方法产生,初始种群的多样性受到限制,再加上存在大量的不可行解,因此交叉概率和变异概率取值不可过低。最后, best 输出的组合路径即为最优基因,即是满足用户约束条件下的最优组合路径。

4 实验及分析

一些服务自动组合的方法,例如文献[15],采用的是基本的 QoS 计算方法,它们都是首先对服务进行语义扩充,然后使用“最短路径”算法获得最优的服务组合。本文将该方法称为 SPBQ(Shortest Path Based on QoS),它们虽然引入了语义,在服务组合成功率等方面比基于关键字的组合算法好很多,但此类方法仅考虑了基本 QoS 属性和组合服务的总体质量,而忽视了用户可能的质量约束条件。与已有的组合方法相比,本文提出的基于动态 QoS 组合方法综合考虑了 QoS 的动态性和网络特性以及用户服务质量约束条件,有效地避免了服务重计算问题,从而能够在尽可能短的时间内获得尽可能好的服务组合。本节实验将 SPBQ 方法与本文提出的基于动态 QoS 方法(记为 CBDQ)相比较。

4.1 实验环境

软硬件环境:此次实验都采用相同的实验软硬件环境,即

CPU 为奔腾 3.0GHz,内存 DDRII 2G,操作系统为 Windows server2003 标准版,开发语言为 Java。

实验数据:由于目前并没有标准的 Web 服务组合测试集,本文采用指定服务模型 20 个,对每个服务模型随机生成 5 个服务实体与之相对应。服务实体的 QoS 属性从时间、费用、可靠性、信誉度 4 个方面描述。每个服务实体的 QoS 属性值也分别在一定的取值范围内随机生成。

4.2 实验结果及分析

实验 1:访问负载实验

目的是说明本文提出的动态 QoS 模型对多个服务的访问负载产生的影响。服务端可提供服务并发的极值为 100,阈值 $K=80, \Delta t=5\text{min}$ 。在实验阶段,SPBQ 算法和本文提出的动态 QoS 算法(CBDQ)在 3 个测试周期内,对某服务模型的 5 个服务实体的访问负载各进行了 5 次随机抽样,然后对 5 次取样进行标准偏差计算。其访问负载的标准偏差如图 5 所示。

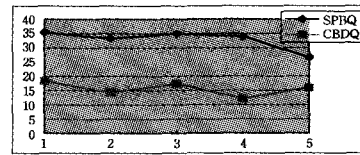


图5 访问负载标准偏差

由图 5 可见,使用基本 QoS 标准和动态 QoS 标准,在标准偏差上有明显的差异。使用基本 QoS 标准的标准偏差要比使用动态 QoS 标准的标准偏差大,这是由于使用基本的 QoS 衡量标准时,不能及时根据实时的负载情况重新计算服务实体的质量,因此每个服务模型的负载主要集中于最优的服务实体,而使用本文提出的 QoS 模型,能够根据服务实体的实时情况计算服务质量,在一定程度上平衡了各服务实体间的负载。

实验 2:组合效率实验

目的是说明用户约束条件引入服务组合过程对组合效率的影响。在未引入用户约束和引入用户约束两种组合策略下,分别以计算出符合用户要求的组合服务的组合时间(单位:ms)作为组合效率,这个时间中包含服务重计算时间。在实验阶段,一个测试周期内,对 5 个服务组合要求进行 3 次取样,然后求平均。其组合效率如图 6 所示。

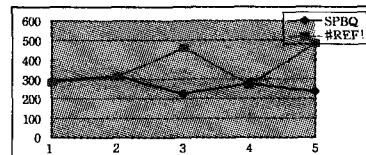


图6 服务组合效率

由图 6 可见,在一般情况下,两种组合方法所用的时间基本相当,SPBQ 方法效率稍高一点。但在未考虑用户约束时,在一定程度上会面临服务重计算,如图 6 中的 3 就是由于重计算所引起的时间延长。

通过实验结果及其分析,可以看出本文提出的动态 QoS 计算方法具有以下优点:(1) 可以使服务的负载更加均衡;(2) 可以在很大程度上避免服务重计算问题的发生。

结束语 为了充分利用共享的 Web 服务,如何将现有的 Web 服务连接与合并起来,生成复杂的复合 Web 服务,从而提供更强大、更完整的商业功能,成为现有服务集合增加价值的**关键**。本文在前人研究的基础上,提出了 Web 服务动态 QoS 模型。与已有的其它 Web 服务 QoS 模型相比,动态 QoS 模型考虑了服务的动态性以及服务组合中的网络特性,使得服务的 QoS 属性更加真实地反映 Web 服务当前的状态。另外,本文将用户约束引入到组合过程中,最大限度地避免了服务重计算现象的发生。但是,Web 组合服务自适应性方面和 Web 服务的动态监控方面还有许多要研究的工作,如何在组合算法中增加自适应性,以及如何更准确地监控 Web 服务,都是今后工作的重点。

参 考 文 献

[1] 王勇,代桂平,侯亚荣.信任感知的聚合动态选择方法[J].计算机学报,2009,32(8):1668-1675

[2] Aggarwal R, Verrna K, Miller J, et al. Constraint driven Web service composition in METEOR-S[C]//Proc of the IEEE International Conference on Services Computing. Shanghai, China; IEEE,2004:23-30

[3] Zeng L Z, Benatallah B, Ngu A H H, et al. QoS-aware middleware for Web services composition[J]. IEEE Trans on Software Engineering,2004,30(5):31-327

[4] 杨胜文,史美林.一种支持 QoS 约束的 Web 服务发现模型[J].计算机学报,2005,28(4):589-594

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

[6] Hashemian S V. A Graph-based Approach to Web Services

Composition[C]// Proc of the 2005 Symposium on Applications and the Internet Society. Trento, Italy; IEEE/IPSJ,2005:183-189

[7] 刘书雷,刘云翔,张帆,等.一种服务聚合中 QoS 全局最优服务动态选择算法[J].软件学报,2007,18(3):646-656

[8] Peer J. Bringing Together Semantic Web and Web Services[C]// Proc of the 1st International Semantic Web Conference (ISWC 2002). London, UK; Springer-Verlag, Lecture Notes in Computer Science, Vol. 2342,2002:279-291

[9] Zhang Lian-jie, Li Bing, Chao Tian, et al. On demand Web services-based business process composition[C]//Proc of the IEEE International Conference on System, Man, and Cybernetics. Washington, USA; IEEE,2003:4057-4064

[10] 代钰,杨雷,张斌,等.支持组合服务选取的 QoS 模型及优化求解[J].计算机学报,2006,29(7):1167-1178

[11] Martin D, Burstein M, Denker G, et al. OWL-S: Semantic markup for Web services[EB/OL]. <http://www.ai.sri.com/daml/services/owl-s/1.2/>,2006

[12] Harmelen F, Hendler J, Horrocks I, et al. OWL Web Ontology Language Reference [EB/OL]. <http://www.w3.org/tr/owl-ref/>,2004

[13] 张成文,苏森,陈俊亮.基于遗传算法的 QoS 感知的 Web 服务选择[J].计算机学报,2009,29(7):1029-1037

[14] 刘峰,谭庆平,杨艳萍.基于图论的 Web 服务合成算法[J].华中科技大学学报:自然科学版,2005,12(3):202-204

[15] Zhang R, Arpinar B, Aleman Meza B. Automatic composition of semantic Web services[C]//Proceedings of International Conference on Web Services. Las Vegas, USA,2003:38-41

(上接第 254 页)

[4] Hu X H, Cercne N. Learning in relational databases: A rough set approach[J]. International Journal of Computational Intelligence,1995,11(2):323-338

[5] Wang G Y, Yu H, Yang D C. Decision table reduction based on conditional information entropy[J]. Chinese Journal of Computer,2002,25(7):759-766

[6] Wang G Y. Calculation method for core attributions of decision table[J]. Chinese Journal of Computer,2003,26(5):611-615

[7] Wang G Y. Rough reduction in algebra view and information view [J]. International Journal of Intelligent System,2003,18(4):679-688

[8] Zhang W X, Mi J S, Wu W Z. Knowledge reductions in inconsistent information systems [J]. Chinese Journal of Computer,2003,26(1):12-18

[9] 腾书华,魏荣华,孙即祥,等.基于不可区分度的启发式快速完备约简算法[J].计算机科学,2009,36(8):196-199

[10] 代劲,何中市.基于云模型的决策表规则约简[J].计算机科学,2010,37(6):265-267

[11] 徐久成,史进玲,孙林.一种基于相对粒度的决策表约简算法[J].计算机科学,2009,36(3):205-207

[12] Xu Z Y, Yang B R, Song W. Comparative study of different at-

tribute reduction based on decision table[J]. Chinese Journal of Electronics,2006,15(4A):953-956

[13] Xu Z Y. Comparative Research of Different Attribute Reduction Definitions[J]. Journal of Chinese Computer Systems,2008,5:848-853

[14] 叶东毅,陈昭炯.一个新的差别矩阵及其求核方法[J].电子学报,2002,30(7):1086-1088

[15] 赵军,王国胤,吴中福,等.一种高效的属性核计算方法[J].小型微型计算机系统,2003,24(11):1950-1953

[16] 聂红梅,周家庆.一个新的差别矩阵及其求核方法[J].四川大学学报:自然科学版,2007,44(2):277-283

[17] 周创德,田卫东.基于约束函数的差别矩阵及其求核算法[J].计算机工程,2008,34(15):60-63

[18] 葛浩,杨传健,李龙澍.基于可分辨矩阵的快速求核算法[J].计算机工程与设计,2009,30(5):1021-1024

[19] 汪小燕.一种改进的差别矩阵及其求核方法[J].安徽工业大学学报:自然科学版,2009,26(1):86-90

[20] Bondy J A, Murty U S R. Graph Theory[S]. ISBN: 978-1-84628-969-9. Springer

[21] 苗夺谦,胡桂荣.知识约简的一种启发算法[J].计算机研究与发展,1999,36(6):681-684