ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.5, May 2008, pp.1173−1185
DOI: 10.3724/SP.J.1001.2008.01173
© 2008 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于非确定性推理的网构软件服务质量动态评估方法*

吴国全[1,2+], 魏 峻[2], 黄 涛[2]

[1](中国科学技术大学 计算机科学与技术系,安徽 合肥 230027)
[2](中国科学院 软件研究所 软件工程技术研究中心,北京 100190)

A Dynamic QoS Assessment Approach for Internetware with Uncertainty Reasoning

WU Guo-Quan[1,2+], WEI Jun[2], HUANG Tao[2]

[1](Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)
[2](Technology Center of Software Engineering, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)
+ Corresponding author: E-mail: gqwu@otcaix.iscas.ac.cn

Wu GQ, Wei J, Huang T. A dynamic QoS assessment approach for Internetware with uncertainty reasoning. *Journal of Software*, 2008,19(5):1173−1185. http://www.jos.org.cn/1000-9825/19/1173.htm

Abstract: This paper proposes a dynamic QoS assessment approach for Internetware with uncertainty reasoning. The salient feature of this approach is that it's based on Bayesian network, and can correctly predict the component's service capability in various combinations of users' QoS requirements, especially to the entities with different service levels. Through online learning, it supports to update the Bayesian network dynamically and improves the accuracy of prediction result. The proposed QoS assessment model has been implemented in the developed Service Coordination Bus (Once-ESB). The experimental results show that it is effective.
Key words: Internetware; service differentiation; quality of service; Bayesian network; fuzzy logic

摘 要: 提出了网构软件环境下一种基于非确定性推理的构件服务质量动态评估方法.该方法基于贝叶斯网络,其主要特点在于考虑了用户对构件的 QoS 需求,可以预测在用户多种 QoS 需求下采用分级策略的构件服务能力,支持评估模型的动态更新,提高了评估结果的准确性.在自主开发的服务协同总线(Once-SCB)平台上进行了应用与验证,结果表明,该评估模型准确、有效,可以在用户多种 QoS 需求下为其选择最为合适的构件.
关键词: 网构软件;服务分级;服务质量;贝叶斯网络;模糊逻辑
中图法分类号: TP311 **文献标识码**: A

网构软件是未来开放软件协同环境(如 Internet)中的一种新型软件形态[1,2],其构建依赖于对开放环境中各种异构的、自治的构件实体间的有效协同.从软件基本模型来看,网构软件的基本计算单元是分布、自治、异构的"构件",我们将具备上述特点的"构件"称为网构构件.为了表述方便,文中提到的"构件"均指"网构构件".构件以自主的方式存在于Internet的各个节点,根据用户的需求以多种方式进行协作,从而对外提供软件服务[3].

* Supported by the National Natural Science Foundation of China under Grant No.60673112 (国家自然科学基金); the National Basic Research Program of China under Grant No.2002CB312005 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grand No.2006AA01Z19B (国家高技术研究发展计划(863))
Received 2007-11-16; Accepted 2008-03-11

© 中国科学院软件研究所 http://www.jos.org.cn

由于 Internet 环境的多变性、现有构件实现技术的制约以及运行环境等多种因素的影响,在实际的协同中构件可能无法提供其声明的服务质量;同时,为了吸引更多的消费者,构件提供者可能故意夸大其提供的服务质量,如何对构件的服务质量进行评估,为用户选择有质量保障的构件参与协同,最大程度地满足用户的 QoS 需求,已成为学术界关注的热点.

分布、自治的构件通常具有“黑箱”性质,无法通过分析内部实现对其提供的服务质量进行评估.现有的网构软件服务质量评估技术主要分为两类:一类基于用户反馈评级对构件提供的 QoS 进行评估,另一类采用监视方法记录构件在实际执行中提供的 QoS,以此为基础对所提供的 QoS 进行评估.第一类评估方法的主要缺陷在于评级主观,用户出于某种原因可能提供虚假的评级,并且缺乏很好的机制激励用户提供评级反馈,因此我们采用客观的基于监视的服务质量评估技术.

服务分级(service differentiation)作为一种灵活的资源管理方案在构件实现中已被广泛使用[4,5],构件提供者将其提供的服务质量划分为多个级别,每个级别代表一类具有相似 QoS 需求的用户[6],对于同一级别的用户 QoS 需求使用相同的分级策略,而对于不同级别的 QoS 需求提供不同分级的服务质量.用户的 QoS 需求信息通常记录在服务水平级协议(SLA)中[4].SLA 是请求者和提供者之间的契约,约定了在实际使用中构件必须提供的各种服务质量(例如,响应时间不得低于 3s,可用性大于 0.8 等)以及违反该协议时相应的惩罚.通过分析 SLA 协议,可以获取用户对构件 QoS 的各种需求信息.

对给定的用户 QoS 需求和一组候选构件,基于监视的服务质量评估根据构件在历史调用中的 QoS 信息,预测候选构件是否能够提供满足当前用户 QoS 需求的服务质量.目前评估方法的主要不足在于评估过程中忽略了用户的 QoS 需求,无法对采用分级技术的构件所提供的服务质量进行正确评估.例如,考虑一个采用分级策略的构件,若忽略用户的响应时间需求,对历史调用信息采用统计平均方法计算出构件的响应时间约为 3.2s,但该评估结果无法真实地反映构件提供的服务质量(实际上该构件的响应时间分级策略为:若用户对响应时间需求较高,则构件提供的响应时间约为 2s;若对响应时间的需求较低,则构件提供的响应时间约为 4s).造成评估结果不正确的原因在于评估过程忽略了历史 QoS 信息中构件对于不同级别的 QoS 需求提供的不同分级的服务质量.因此,对采用分级策略的构件,在评估时需要考虑用户的 QoS 需求,根据与当前用户 QoS 需求相关的历史信息评估构件提供的服务质量,以保证评估的正确性.进一步地,考虑用户对构件的多种 QoS 需求(如响应时间、可用性等),如何对构件提供的服务质量进行总的评价,从而选择满足用户多种 QoS 需求的最为合适的构件.此外,在开放动态的网络环境下构件所提供的服务质量易受多种因素影响发生变化,因此要求评估方法支持评估模型的动态更新以保证评估结果的准确性.

针对现有工作的不足,本文提出了一种基于非确定性推理的构件服务质量动态评估方法.该方法考虑了用户对构件 QoS 的多种需求,通过为候选构件建立基于贝叶斯网络的质量评估模型,选择满足用户 QoS 需求的最为合适的构件参与协同,并且基于监控信息引入模糊规则推理构件在实际执行中提供的服务能力,支持评估模型的动态更新,提高了评估结果的准确性.该方法可以作为一个独立的模块实现为网构软件选择满足用户 QoS 需求的构件参与协同,从而向用户提供有质量保障的软件服务.

本文第 1 节概述构件服务质量动态评估方法.第 2 节介绍基于贝叶斯网络的构件服务质量评估模型.第 3 节介绍基于模糊逻辑推理服务能力(service capability)支持评估模型的动态更新.第 4 节介绍该评估方法的原型实现及其应用.第 5 节通过实验验证该方法评估结果的准确性.第 6 节介绍相关工作,最后是总结.

1 构件服务质量评估

构件服务质量评估主要用于网构软件的动态协同中,通过对候选构件提供的 QoS 进行评估,预测构件的服务质量,为用户选择满足其 QoS 需求的最为合适的构件参与协同.图 1 给出了基于非确定性推理的构件服务质量动态评估方法和构件选择过程:

① QoS 评估模块(BNs for QoS evaluation)基于构件在历史调用中用户的 QoS 需求信息和相应的服务能力信息(根据步骤②③④计算)为候选构件建立基于贝叶斯网络的评估模型,并根据评估模型预测在当前用户的多

© 中国科学院软件研究所 http://www.jos.org.cn

种 QoS 需求下候选构件的服务能力,服务能力最优的被选中参与网构软件的协同.

② 在构件使用过程中,通过 QoS 监视器(QoS monitor)记录构件的真实 QoS 信息.

③ 构件执行结束后,对于每种 QoS 属性,偏差值计算模块(compliance calculator)计算用户的需求和构件实际提供 QoS 的偏差值(compliance data).

④ 基于 QoS 偏差信息,引入模糊逻辑推理过程(fuzzy inference process)计算该构件实际执行中的服务能力,并使用该结果更新构件的 QoS 评估模型.

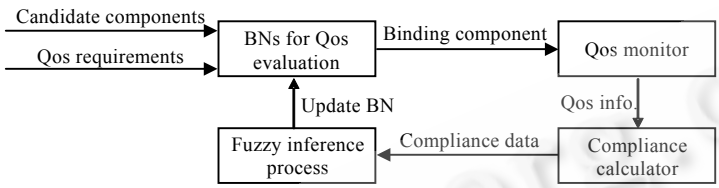


Fig.1 Dynamic QoS assessment approach for components with uncertain reasoning

图 1 基于非确定性推理的构件服务质量动态评估方法

2 基于贝叶斯网络的构件服务质量评估模型

定义 1(**服务能力模型**). 构件的服务能力模型(service capability model,简称 SCM)定义为函数 $SC=f(OP,Q_1,\ldots,Q_n)$.其中,服务能力(SC)是对构件提供的各种 QoS 质量的总的评估;f 为服务能力评估函数;参数 Q_i $(1\le i\le n)$表示用户的各种 QoS 需求,OP 表示用户请求的操作.服务能力的评估与用户的 QoS 需求有关,也和用户请求的操作相关,在实际的构件实现中可能包含多个操作,对于不同的 OP 提供的服务能力不同.目前,我们考虑 3 种通用的 QoS 属性:响应时间(DU)、可靠性(RE)和可用性(AV),其具体定义同文献[7].

为了求解在用户的多种 QoS 需求下的构件服务能力,我们将 SCM 模型映射到贝叶斯网络,提出一种基于贝叶斯网络的构件服务质量评估模型.我们选用贝叶斯网络进行服务质量评估的原因在于:1) 由于构件的分布自治特性,其提供的服务质量易受多种因素的影响,基于概率推理的方法更能真实地反映其提供的服务质量;2) 使用贝叶斯网络可以很方便地计算在用户多种 QoS 需求组合下的服务能力.

贝叶斯网络(Bayesian networks)也称为因果网络(causal networks),是一种概率推理模型[8],已在智能系统中广泛用于不确定性的知识表示和推理,预测类成员关系的可能性.贝叶斯网络是一个 DAG 图,规定了对于图中的每个节点 V_i,给定其双亲,V_i 条件独立于图中的非后继节点构成的节点子集,其理论基础基于贝叶斯规则:

$$p(H\mid X)=\frac{p(X\mid H)p(H)}{p(X)},$$

其中,$p(H|X)$是在条件 X 下 H 的后验概率,$p(H)$和 $p(X)$是先验概率,$p(X|H)$表示在条件 H 下 X 的后验概率.

定义 2(**构件服务质量分级评估模型**). 该模型是构件的服务能力模型 SCM 到朴素贝叶斯网络的一个映射,如图 2 所示.其中同名变量 SC 代表分级评估节点,为了细粒度区分服务能力,其取值划分为{*poor*,*moderate*,*good*,*excellent*}四个级别.节点 AV,DU,RE,OP 分别用同名变量表示构件的可用性、响应时间、可靠性以及构件所提供的操作,目前我们假设 QoS 属性之间无相互影响.节点的条件概率表 CPT(conditional probability table)以$\{P(Q_i|SC)\}$的集合形式表示,Q_i 取值分别为变量 OP,AV,DU,RE,$1\le i\le 4$.

对候选构件的服务能力评估依据条件概率 $P(SC|Q_1,\ldots,Q_i)$计算:

$$P(SC\mid Q_1,\ldots,Q_i)=\frac{P(Q_1,\ldots,Q_i\mid SC)\cdot P(SC)}{P(Q_1,\ldots,Q_i)} \quad \text{(贝叶斯规则)}$$

$$=\frac{P(Q_1\mid SC)\cdot\ldots\cdot P(Q_i\mid SC)\cdot P(SC)}{P(Q_1,\ldots,Q_i)}. \quad \text{(类条件独立假设)}$$

© 中国科学院软件研究所 http://www.jos.org.cn

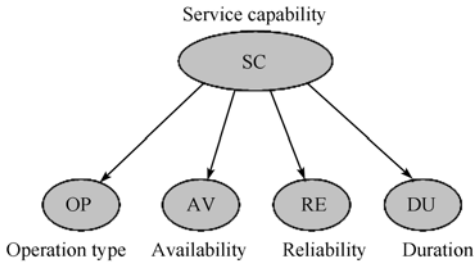


Fig.2 QoS assessment model for component with multi-levels

图 2 构件服务质量分级评估模型

对某些取值连续的 QoS 属性,为了使用贝叶斯网络对构件提供的服务质量进行评估,需要对其离散化.这里介绍 3 种连续变量离散化方法.第 1 种方法根据构件所属的不同领域,由领域专家事先为需要离散化的 QoS 属性指定不同的离散区间.例如,对响应时间属性离散化后为 0s –5s,5s –10s,>10s,分别记为 high,medium 和 low;第 2 种方法依据构件所声明的分级 QoS 信息.第 3 种方法使用文献[9]中介绍的递归的基于最小熵的划分方法(recursive minimal entropy partitioning),文中比较了几种常用的连续变量离散化方法,并通过实验得出基于熵的连续变量离散化方法可以大幅度提高贝叶斯分类算法的准确性.对于服务能力变量 *SC* 的取值,本文将其划分为 poor,moderate,good 和 excellent 四个级别,也可以进行更细粒度的划分.构件实际执行时的服务能力可以采用第 1.2.1 节的表 3 和表 4 的规则计算,我们将在后面进行详细介绍.

基于贝叶斯网络的构件服务质量评估模型的优势在于,可以准确评估候选构件在实际执行中是否能满足用户提出的多种 QoS 需求.模型计算在用户多种 QoS 需求下 *SC* 为各种服务级别的后验概率,后验概率最高的服务级别即为用户当前 QoS 需求下的服务能力.下面给出了一个机票预定构件的服务能力评估方法:

假设该构件包含机票查询(query)、机票预定(book)和取消预定(cancel)三种类型的操作,*OP* 的取值为 {*query*,*book*,*cancel*}.当用户订阅机票,其可靠性需求为 high、响应时间需求为 medium 时,评估模型分别计算 *SC* 为 poor,moderate,good 和 excellent 四个级别的概率,其条件概率表达如下:

$P(SC=\text{“poor”}|OP=\text{“}book\text{”},RE=\text{“high”},DU=\text{“medium”})$;

$P(SC=\text{“moderate”}|OP=\text{“}book\text{”},RE=\text{“high”},DU=\text{“medium”})$;

$P(SC=\text{“good”}|OP=\text{“}book\text{”},RE=\text{“high”},DU=\text{“medium”})$;

$P(SC=\text{“excellent”}|OP=\text{“}book\text{”},RE=\text{“high”},DU=\text{“medium”})$.

条件概率的计算可以使用贝叶斯规则,在使用贝叶斯规则计算后验概率时,需要利用节点的条件概率表.在贝叶斯网络中,每个节点与一个条件概率表相关联.表 1 给出了节点 *RE*(可靠性)的条件概率表.

Table 1 CPT of relibility (*RE*)

表 1 可靠性(*RE*)的条件概率表

RE	*SC*			
	Poor	mod(erate)	good	exc(ellent)
low	$P(RE=\text{low}\|SC=\text{poor})$	$P(RE=\text{low}\|SC=\text{mod})$	$P(RE=\text{low}\|SC=\text{good})$	$P(RE=\text{low}\|SC=\text{exc})$
med(ium)	$P(RE=\text{med}\|SC=\text{poor})$	$P(RE=\text{med}\|SC=\text{mod})$	$P(RE=\text{med}\|SC=\text{good})$	$P(RE=\text{med}\|SC=\text{exc})$
high	$P(RE=\text{high}\|SC=\text{poor})$	$P(RE=\text{high}\|SC=\text{mod})$	$P(RE=\text{high}\|SC=\text{good})$	$P(RE=\text{high}\| C=\text{exc})$

其中,$P(RE=\text{“high”}|SC=\text{“excellent”})$表示该构件在历史执行中服务能力为 excellent 时,可靠性为 high 的概率:

$$P(RE=\text{“high”}|SC=\text{“excellent”})=\frac{P(RE=\text{“high”},SC=\text{“excellent”})}{P(SC=\text{“excellent”})}.$$

$P(RE=\text{“high”},SC=\text{“excellent”})=m/n$ 表示在 n 次历史执行中,服务能力为 excellent 且可靠性为 high 的概率.m 表示在 n 次执行中,服务能力为 excellent 且可靠性为 high 的调用次数.

$P(SC=\text{“excellent”})=k/n$ 表示在 n 次历史调用中,服务能力为 excellent 的概率.k 表示在历史执行中评估结果

© 中国科学院软件研究所 http://www.jos.org.cn

为 excellent 的次数,并且有:

$$\sum_{z\in S} P(SC=z)=1,\quad SC=\{\text{“poor”},\text{“moderate”},\text{“good”},\text{“excellent”}\}.$$

对于节点 *AV* 和 *DU*,其条件概率表的计算与 *RE* 类似.注意到,在实际的构件选择中,每种 QoS 属性对用户的重要性程度可能不同,为了表达用户的 QoS 属性偏好,可以对条件概率表进行扩展,增加一个布尔变量以表达该 QoS 属性对用户是否重要(受篇幅所限,没有给出扩展后的条件概率表).

3 动态更新QoS评估模型

3.1 基于模糊逻辑推理服务能力

由于构件的自治性以及运行环境的多变性,所提供的服务质量易受多种因素的影响而发生变化,为了保证评估结果的准确性,我们给出了一种评估模型的动态更新方法,即在构件每次执行结束后,依据构件在每种 QoS 属性上的表现计算该次执行中构件所提供的服务能力,并使用该结果更新相应的贝叶斯网络.我们通过引入“compliance”刻画构件在每种 QoS 属性上实际提供的 QoS 与用户需求的偏差,基于偏差值引入模糊逻辑计算总的服务能力.

假设 P 表示某个 QoS 属性,P_u 表示用户对 P 的 QoS 需求值.P_d 表示通过监控方法获取的服务实际提供的 QoS 值.构件执行结束后,属性 P 的 compliance 值 C_p 如下计算:

$$C_p=(P_d-P_u)/P_d.$$

理想情况下,当 $C_p=0$ 时,表示对该 QoS 属性构件实际提供的 QoS 与用户的需求一致,因此对属性 P,SC_p(构件在 P 上的服务能力)为 good.当 $C_P>0$ 时,表示构件实际提供的 QoS 大于用户的 QoS 需求,SC_p 为 excellent;当 $C_p<0$ 时,表示该构件实际提供的 QoS 低于用户的 QoS 需求,SC_p 为 poor.综上,对于 QoS 属性 P,C_p 与 SC_p 有如下关系:

$$C_p=0 \Rightarrow SC_p \text{ is good} \tag{1}$$

$$C_p<0 \Rightarrow SC_p \text{ is poor} \tag{2}$$

$$C_p>0 \Rightarrow SC_p \text{ is excellent} \tag{3}$$

但是,上面的对应关系存在不足.例如,考虑 $C_p=-0.001$,由式(2),SC_p 为 poor,但在实际应用中,-0.001 非常接近于 0,SC_p 为 good 更加合理,更符合用户的期望,因此需要寻求一种更加合理的对 compliance 的划分和评估方法.模糊逻辑提供一种很好的解决方案,模糊逻辑基于模糊集合(fuzzy set),将研究对象以 0,1 之间的数值来表示模糊概率的程度(称为隶属函数),使得计算出的结果更符合人的思考模式.

图 3 给出了对响应时间属性 compliance 的模糊表示.图中定义了 4 个隶属函数,分别用 low compliant,less compliant,compliant 和 high compliant 表示.隶属函数的形式会随使用者的定义而有所不同,但在实际应用通常使用规格化的标准隶属函数.常用的 4 种标准隶属函数有 Z-type,Λ-type(lambda),Π-type(pi)和 S-type.例如隶属函数 low compliant 使用 Z-type 表示,less compliant 使用Λ-type 标准隶属函数.定义 compliance 使用的隶属函数类型及数目依赖于所需要表示的 QoS 属性的特征和所需要的划分数目,对于不同的 QoS 属性,其 compliance 可以有不同的模糊表示.

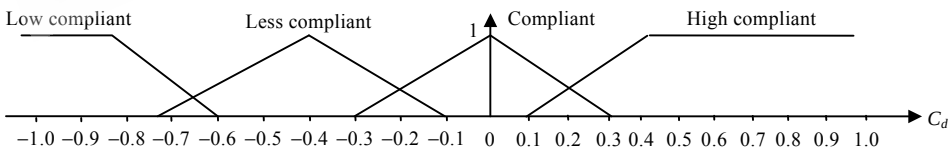


Fig.3 Fuzzy function of compliance value (*DU*)

图 3 响应时间(*DU*)偏差值模糊函数

© 中国科学院软件研究所 http://www.jos.org.cn

对于服务能力,对应于前面给出的服务能力级别 poor,moderate,good 和 excellent,我们定义了 4 种相应的隶属函数,如图 4 所示(假定对服务能力的评估范围为 1~5).

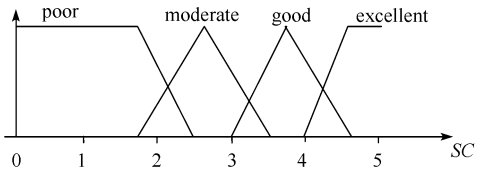


Fig.4 Fuzzy function of *SC*

图 4 服务能力(*SC*)的模糊函数

在给出 QoS 属性偏差值和服务能力的模糊表示后,我们定义一组模糊推理规则以计算总的服务能力.推理规则的前件是每种 QoS 属性的偏差值,后件是相应的 *SC* 级别.我们在 第 3.1 节中指出,对于不同的,用户每种 QoS 属性的重要性是不相同的,因此,我们区分无用户偏好的推理规则和基于用户偏好的推理规则.表 2 给出了无用户偏好的推理规则集,表 3 给出了基于用户偏好的推理规则集(C_d:响应时间的偏差值;C_a:可用性偏差值;C_r:可靠性偏差值).

在推理过程中,QoS 属性的偏差值可能满足多条规则的前件,为了计算总的服务能力,我们使用了 Mamdani min-max 模糊推理方法[10],规则的输出使用 fuzzy union 方法进行聚合,得到最终的服务能力模糊集表示. Mamdani min-max 推理过程如下:首先对规则前件各命题的函数值取其逻辑与(min)作为前件部的适合度,然后将此适合度与规则后件的命题作逻辑与(min)运算,作为每一个规则的结果程度值.接着对整个规则库作逻辑和(max),所得即为整个模糊推论的结果.

Table 2 Fuzzy reasoning rules without user preference

表 2 无用户偏好的模糊推理规则

Rule	Antecedent			Consequent	Rule	Antecedent			Consequent
	C_d	C_a	C_r	*SC*		C_d	C_a	C_r	*SC*
R1	high compliant	high compliant	high compliant	excellent	R2	high compliant	high compliant	compliant	excellent
R3	high compliant	compliant	high compliant	excellent	R4	compliant	high compliant	high compliant	excellent
R5	compliant	compliant	high compliant	excellent	R6	compliant	high compliant	compliant	excellent
R7	high compliant	compliant	compliant	excellent	R8	compliant	compliant	compliant	good
R9	high compliant	high compliant	less compliant	good	R10	high compliant	less compliant	high compliant	good
R11	less compliant	high compliant	high compliant	good	R12	less compliant	less compliant	less compliant	moderate
R13	compliant	compliant	low compliant	moderate	R14	compliant	low compliant	compliant	moderate
R15	low compliant	compliant	compliant	moderate	R16	high compliant	low compliant	low compliant	moderate
R17	high compliant	low compliant	low compliant	moderate	R18	high compliant	low compliant	low compliant	moderate
R19	low compliant	low compliant	low compliant	poor	R20	less compliant	low compliant	low compliant	poor
R21	low compliant	less compliant	low compliant	poor	R22	low compliant	low compliant	less compliant	poor

© 中国科学院软件研究所 http://www.jos.org.cn

Table 3 Fuzzy reasoning rules with user preference

表 3 基于用户偏好的模糊推理规则

Bias: Duration is more important	
Rule 1	if C_d is high compliant=>*SC* is excellent
Rule 2	if C_d is compliant=>*SC* is good
Rule 3	if C_d is less compliant=>*SC* is moderate
Rule 4	if C_d is low compliant=>*SC* is poor
Bias: Availability is more important	
Rule 5	if C_a is high compliant=>*SC* is excellent
Rule 6	if C_a is compliant=>*SC* is good
Rule 7	if C_a is less compliant=>*SC* is moderate
Rule 8	if C_a is low compliant=>*SC* is poor
Bias: Reliability is more important	
Rule 9	if C_r is high compliant=>*SC* is excellent
Rule 10	if C_r is compliant=>*SC* is good
Rule 11	if C_r less compliant=>*SC* is moderate
Rule 12	if C_r low compliant=>*SC* is poor
…	
Bias: Reliability is more important	
Rule m1	If C_d, C_a are high compliant=>*SC* is excellent
Rule m2	If C_d, C_a are compliant=>*SC* is good
Rule m3	If C_d, C_a are low compliant=>*SC* is poor
Rule m4	If C_d, C_a are less compliant=>*SC* is moderate
Rule m5	If C_d is high compliant, C_a is low compliant=>*SC* is moderate
Rule m6	If C_d is low compliant, C_a is high compliant=>*SC* is moderate
Rule m7	If C_d is low compliant, C_a is low compliant=>*SC* is moderate
Rule m8	If C_d is compliant, C_a is low compliant=>*SC* is poor
Rule m9	If C_d is low compliant, C_a is compliant=>*SC* is poor

例如,考虑无用户偏好的推理规则,响应时间(*DU*)的偏差值为 0.05,可用性(*AV*)的偏差值为−0.15,可靠性(*RE*)的偏差值为 0.25(假设 *AV* 和 *RE* 使用和 *DU* 相同的模糊表示),满足规则 R5 和规则 R8 的前件,使用 Mamdani min-max 模糊推理方法,得到作用于规则 R5,R8 后服务能力的模糊表示,如图 5 右下所示.再通过各种解模糊方法(defuzzification),如面积中心法、最大中心法等得到解模糊化后的值,在本例中为 4.3.模糊推理引擎检测完规则 R1~R20 以后,得到总的服务能力.

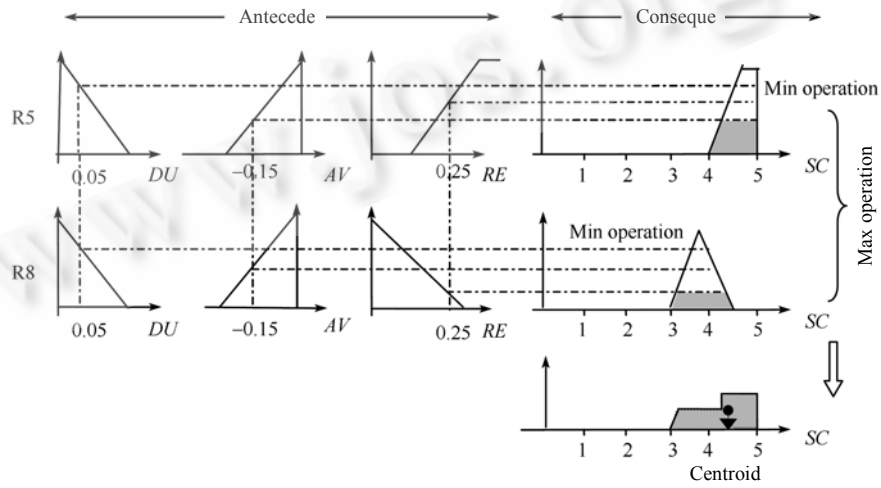


Fig.5 An example of mamdani min-max fuzzy reasoning

图 5 基于 Mamdani min-max 的模糊推理方法示例

© 中国科学院软件研究所 http://www.jos.org.cn

3.2 评估模型更新

根据推理引擎推理出的服务能力,我们提出评估模型的动态更新方法,具体的更新步骤是对贝叶斯网络中每个节点的条件概率表进行更新.记在构件执行结束后根据第 1.2.1 节推理得到 SC 为 y,变量 X 的取值为 x,m 表示在历史调用中 $SC=y$ 的次数,m'为更新后 $SC=y$ 的次数($m'=m+1$),则更新后的条件概率 $P'(X=x|SC=y)$可由先验概率 $P(X=x|SC=y)$按下式计算:

$$P'(X=x\mid SC=y)=\frac{P(X=x\mid SC=y)\times m+1}{m'}.$$

条件概率表中变量 X 取其他值的条件概率按下式更新:

$$P'(X=x'\mid SC=y)=\frac{P(X=x'\mid SC=y)\times m}{m'}.$$

例如,考虑节点 RE 的条件概率表更新,当 RE 取“high”时,得到 SC=“excellent”,则更新后的条件概率 $P'(\cdot|SC=y)$如下(假定初始情况下变量 AV 的取值有相同的概率分布):

$$P'(RE=\text{high}\mid SC=\text{excellent})=\frac{0.33\times1+1}{2}=0.67,$$

$$P'(RE=\text{low}\mid SC=\text{excellent})=\frac{0.33\times1}{2}=0.165,$$

$$P'(RE=\text{medium}\mid SC=\text{excellent})=\frac{0.33\times1}{2}=0.165.$$

考虑到在实际的网构软件协同中构件所提供的服务质量不是固定不变的,可能受多种因素影响而发生变化,根据构件最近执行所计算出的服务能力更能真实反映构件当前的服务质量,可以为条件概率表中的每一项加入一个随时间衰减的系数,并且采用归一化处理以保证条件概率表中的每一列的和为 1 以提高评估结果的准确性.

目前,我们只考虑 3 种通用的 QoS 属性,更多的可以通过监控方法获得的领域相关的 QoS 属性都可以添加到评估模型中,例如对于提供文件下载服务的构件,可以将下载速率作为影响服务能力的一个 QoS 属性加入到该评估模型中.

4 原型实现与应用

基于非确定性推理的构件服务质量动态评估模块包含 3 个重要组成:

- 偏差值计算模块:负责计算服务实际监控的 QoS 信息与用户的 QoS 需求之间的偏差.
- 模糊推理引擎:根据偏差值模块计算出的偏差值,使用模糊推理过程计算总的服务能力,并更新相应的贝叶斯网络,我们使用 MATLAB 7.0 实现了模糊推理引擎.
- 基于贝叶斯网络的质量评估和推荐模块:通过评估用户多种 QoS 需求下候选构件集的服务能力,为用户选择最为合适的服务.

目前我们将该评估模块用于自主开发的服务协同总线平台 Once-SCB 中,Once-SCB 负责管理服务请求者和服务提供者之间的交互协同逻辑,支持服务间依赖关系的动态确定和服务交互质量的动态保障,其总体架构如图 6 所示.服务请求者通过 Request Broker 提交对服务的功能性和非功能性需求,Request Broker 在内部生成相应的 BSR(business service request)实例负责与服务注册中心交互查找候选服务集合,根据评估模块的评估结果选择满足用户 QoS 需求的最合适的服务作为绑定服务,并构造该服务在 Once-SCB 上的访问端点.中介处理链(mediation framework)提供消息的监控、路由及适配逻辑等,监控方法基于拦截器模式.服务提供者通过 Provider Broker 和 BSP(business service provider)进行服务的发布和注册.

© 中国科学院软件研究所 http://www.jos.org.cn

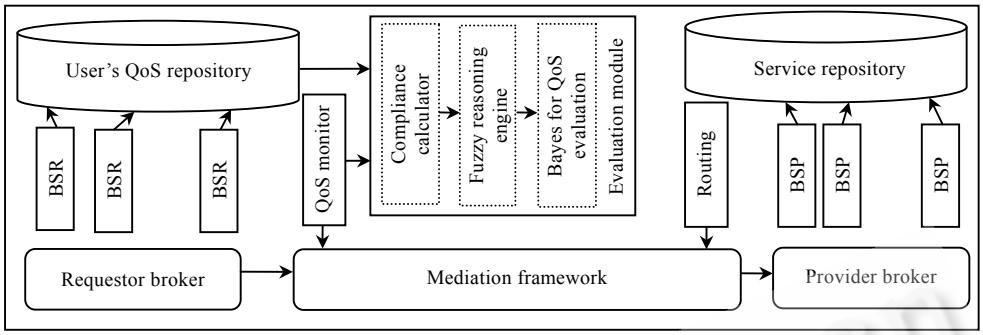


Fig.6 Once-SCB service coordination bus

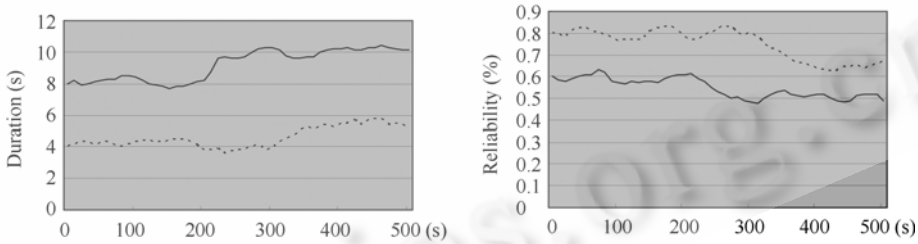
图 6 Once-SCB 服务协同总线

5 实验及分析

我们进行两组实验以分析所提出的构件服务质量评估方法的有效性.第 1 组实验验证评估模型的评估结果是否与服务的分级策略一致.第 2 组实验将该评估模型用于服务选择,验证该模型是否能够选择最为合适的服务.

前两组实验的环境如下:服务用例部署在 P4 2.8GHZ,500RAM 的机器上,所使用的操作系统 Windows XP professional sp2.客户端使用 Apache 的 JMeter 2.1 作为负载生成工具模拟多个并发线程发送请求,并发度为 20.每次调用的 QoS 请求根据 QoS 变量离散化后的结果随机生成,为了方便实验的进行,这里采用了人为指定连续变量离散化的方法.

在第 1 组实验中,我们开发了具有分级能力的 Web 服务,为了更加符合实际应用,服务的分级策略随时间变化.图 7(a)给出了响应时间分级策略的时间变化曲线,图 7(b)给出了服务可靠性分级策略的时间变化曲线.



(a) Two levels of *DU*

(a) 两种级别的响应时间

(b) Two levels of *RE*

(b) 两种级别的可靠性

Fig.7

图 7

表 4 和表 5 分别给出了在不同时刻,贝叶斯评估模型的预测结果.

Table 4 Prediction result when *t*=100s (%)

表 4 *t*=100s 时的预测结果(%)

	DU: 3.5s~4.5s, *RE*: 0.75~0.85	*DU*: 7.5s~8.5s, *RE*: 0.55~0.65	*DU*: 1s~3s, *RE*: 0.9~1.0
poor	0.00	0.15	10.40
moderate	1.72	1.91	0.82
good	20.81	21.52	0.50
excellent	0.92	0.85	0.21

从表 4 可以看出,评估模型的预测结果与服务在该时段提供的分级策略一致.如对第 1 列的用户请求(3.5s<*DU*<4.5s,0.75<*RE*<0.85),评估模型的预测结果为“good”;对于第 3 列,评估模型的预测结果为“poor”,从图 9

© 中国科学院软件研究所 http://www.jos.org.cn

和图 10 可以看出,服务的分级策略不支持该组 QoS 请求,评估结果与服务的分级策略一致.

Table 5 Prediction result when *t*=280s (%)

表 5 *t*=280s 时的预测结果(%)

	DU: 3.5s~4.5s, *RE*: 0.75~0.85	*DU*: 7.5s~8.5s, *RE*: 0.55~0.65	*DU*: 1s~3s, *RE*: 0.9~1.0
poor	0.02	15.35	9.40
moderate	1.83	5.91	0.72
good	22.32	9.52	0.45
excellent	0.97	0.75	0.11

在 *t*=280s 时刻,从表 5 可以看出,对于第 1 列,虽然服务的分级策略发生了变化,但是对于用户的需求(3.5s<*DU*<4.5s,0.75<*RE*<0.85)仍能满足;对于第 2 列,评估的结果为“poor”,而图 7 表明在该时间段服务提供的 QoS 已不能满足用户的需求,评估结果与服务的分级策略一致;对于第 3 列,在该时间段服务提供的分级策略仍不支持该组 QoS 请求,与评估模型的预测结果一致.

在 *t*=450s 时刻,从表 6 可以看出,对于第 1 列,评估的结果为“poor”,原因在于服务的分级策略发生了变化,该组的 QoS 需求已经无法得到满足.同理,第 2 列和第 3 列的 QoS 请求也无法得到满足.

Table 6 Prediction result when *t*=450s (%)

表 6 *t*=450s 时的预测结果(%)

	DU: 3.5s~4.5s, *RE*: 0.75~0.85	*DU*: 7.5s~8.5s, *RE*: 0.55~0.65	*DU*: 1s~3s, *RE*: 0.9~1.0
poor	18.83	15.35	9.40
moderate	9.76	5.91	0.72
good	8.32	9.52	0.45
excellent	0.97	0.75	0.11

在第 2 组实验中,我们将该评估模型用于 QoS 驱动的服务选择,通过开发两个具有相同功能但采用不同分级策略的 Web 服务,验证该评估模型是否能为用户选择合适的服务.ws1 和 ws2 提供的分级策略如图 8 所示.

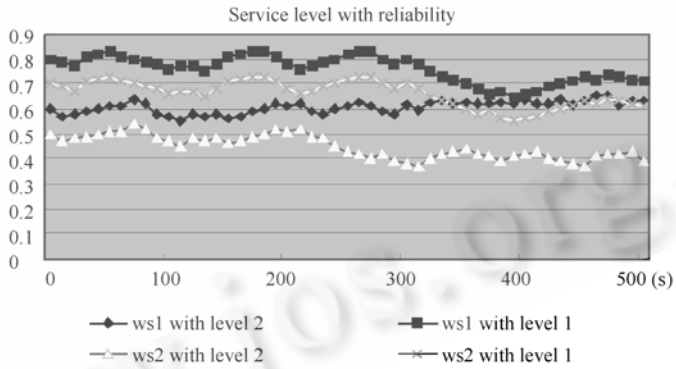


Fig.8 Differentiation policy of ws1, ws2 (*RE*)

图 8 ws1 和 ws2 的可靠性(*RE*)分级策略

图 8 给出了 ws1,ws2 可靠性(*RE*)分级策略的时间变化曲线;图 9 给出了 ws1,ws2 响应时间(*DU*)的时间变化曲线.

表 7 给出了在 *t*=100s 时,对于 3 组不同的 QoS 请求,ws1,ws2 评估模型的预测结果,从表中可以看出,对于第 1 列的 QoS 请求(3.5s<*DU*<4.5s,0.75<*RE*<0.85),ws1 评估模型预测结果为“good”,ws2 评估模型的预测结果为“poor”,因此 ws1 提供的服务质量满足该组 QoS 需求,与服务在该时段提供的分级策略一致.而对于第 2 列的 QoS 请求(7.5<*DU*<8.5,0.55<*RE*<0.65),根据 ws1,ws2 的评估模型的评估结果,选择 ws1 进行服务调用,注意,虽然 ws2 的分级策略 level 1 也完全满足该组 QoS 请求,但考虑到在实际使用中,其提供的服务分级越高,使用该服务的费用也越高,因此选择 ws1 最为合适;对于第 3 列 QoS 请求,ws1 的评估结果为“moderate”,ws2 的评估结果为

© 中国科学院软件研究所 http://www.jos.org.cn

“poor”,与 ws1,ws2 实际提供的分级策略相一致,ws1,ws2 提供的服务质量都无法满足该组 QoS 请求.

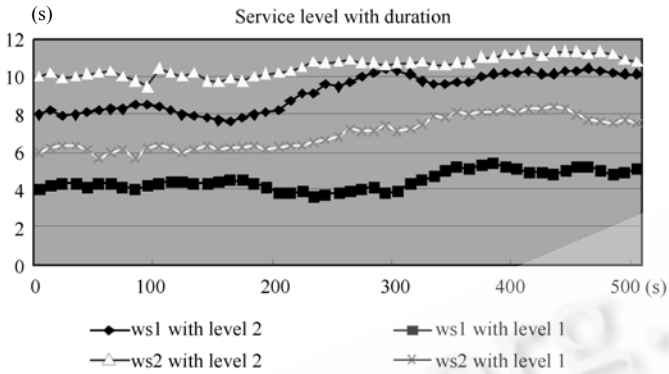


Fig.9 Differentiation policy of ws1, ws2 (*DU*)

图 9 ws1 和 ws2 的响应时间(*DU*)分级策略

Table 7 Prediction result of ws1, ws2 when *t*=100s

表 7 *t*=100s 时,ws1,ws2 的预测结果

	DU: 3.5s~4.5s, *RE*: 0.75~0.85	*DU*: 7.5s~8.5s, *RE*: 0.55~0.65	*DU*: 1s~3s, *RE*: 0.9~1.0
ws1	good	good	moderate
ws2	poor	moderate	poor

表 8 给出了在 *t*=400s 时,对于 3 组不同的 QoS 请求,ws1,ws2 评估模型的预测结果,从表中可以看出,对于第 1 组 QoS 请求,ws1,ws2 提供的分级服务都无法满足.对于第 1 组 QoS 请求,ws2 的评估结果为“good”,ws1 的评估结果为“moderate”,与 ws1 和 ws2 提供的分级策略一致.对于第 3 组 QoS 请求,两个评估模型的评估结果均为“poor”,都无法满足用户的 QoS 需求,与 ws1 和 ws2 提供的分级策略相一致.

Table 8 Prediction result of ws1, ws2 when *t*=400s

表 8 *t*=400s 时,ws1,ws2 的预测结果

	DU: 3.5s~4.5s, *RE*: 0.75~0.85	*DU*: 7.5s~8.5s, *RE*: 0.55~0.65	*DU*: 1s~3s, *RE*: 0.9~1.0
ws1	moderate	poor	poor
ws2	poor	good	poor

从实验结果我们可以得出,所提出的 QoS 动态评估方法能够对候选服务的 QoS 进行准确评估,识别服务提供的分级策略,选择满足用户 QoS 需求的最合适的服务进行协作,从而向用户提供有质量保证的复合服务.

6 相关工作

近年来,QoS 评估方面的研究已经受到越来越多的关注.在多主体系统领域(agent domain)已有多种评估模型提出来,从简单的平均值法到复杂的离散信任模型、流模型等[11−13],其主要思想是通过聚合来自用户的评级对软件实体提供的质量进行评估.文献[14]提出了一个用于度量软件服务间信任关系的信任评估模型,信任被抽象成一个由信任评估主体对客体的主观期望和客观经验共同作用的函数,并综合了直接经验和第三方的推荐经验.针对网构软件的动态协作特点,文献[15]提出了一个适用于网构软件的信任度量及演化模型.该模型不仅对信任关系度量过程和信任传递及合并过程进行了合理抽象,还提供了一种合理的方法,用于促进协同实体间信任关系的自动形成与更新,为解决开放网络环境下网构软件的可信性问题提供了很好的解决思路.

随着 Internet 的快速发展,目前以 Web 服务为代表的构件及构件协同已成为一种新兴的 Web 应用形态.针对 Web 服务,文献[16]采用了一种个性化的 QoS 评估方法,该方法考虑了用户对服务的期望,认为有相同期望的用户对服务的评价更加一致.但是,该方法的缺陷在于需要用户显式地指定期望值,并且用户使用完服务后需要提供对各种 QoS 的评级信息.尽管评级信息反映了用户对服务质量的满意度,但目前还没有很好的机制来激励

© 中国科学院软件研究所 http://www.jos.org.cn

用户提供评级信息,并且用户从自身的利益出发,可能提供虚假的评级信息,同时这类系统还易遭受到联合欺骗等方式的攻击,从而影响服务质量评估的准确性.文献[17]提出了一种基于 QoS 监控的服务质量评估模型,通过引入 *verity* 度量服务声明的 QoS 与其实际提供 QoS 的一致性,*verity* 和用户的评级共同作为对服务质量评估的依据.文献[6]提出了一种基于多质量空间映射(multiple QoS-space mapping)的 Web 服务质量评估方法,其缺陷在于仅考虑了单个 QoS 属性,没有解决在用户多种 QoS 需求的情况下对服务者提供的服务质量进行评估的问题.在复合服务的质量保障方面,文献[7,18]基于服务所声明的 QoS 质量分别使用整形规划和启发式算法进行服务选择保证用户提出的 QoS 需求得到满足,但其不足在于忽略了在开放的网络环境下,服务可能无法提供其所声明的服务质量,因此构造出的复合服务在实际运行中可能无法满足用户的 QoS 需求.由于 Web 服务的分布自治特性,文献[19]提出了基于马尔可夫决策(MDP)的服务组合方法,在策略计算时使用贝叶斯网络表示马尔可夫决策过程,选择最优的行为并更新相应的转换函数,而我们首先将构件的服务能力评估建模为朴素贝叶斯网络,并通过更新贝叶斯网络的方法使得评估结果更加真实地反映构件的服务能力.

7 总 结

目前大部分的研究工作将基于服务质量的构件选择归结为多条件约束满足问题,基于构件声明的服务质量选择满足用户 QoS 需求的构件参与协作,忽略了构件的分布自治特点,构件在实际执行中无法提供其所声明的服务质量.本文提出了网构软件环境下一种基于非确定性推理的构件服务质量动态评估方法.该方法基于贝叶斯网络能够预测用户多种 QoS 需求下的构件服务能力,从而为用户选择合适的构件参与网构软件协作.实验结果表明,该评估模型准确、有效.

进一步的研究工作包括:1) 模糊规则集的大小对评估结果正确性的影响;2) 目前考虑了单个构件的服务质量评估技术,由于网构软件本身也可以作为一个软件服务对外发布,在现有工作的基础上,我们将进一步考虑协作结构研究网构软件的服务质量评估技术.

References:

[1] Yang FQ. Thinking on the development of software engineering technology. Journal of Software, 2005,16(1):1−7 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1.htm

[2] Yang FQ, Mei H, Lü J, Jin Z. Some discussion on the development of software technology. Acta Electronica Sinica, 2002,30(12A): 1901−1906 (in Chinese with English abstract).

[3] Lü J, Ma XX, Tao XP, Xu F, Hu H. Research and progress on Internetware. Science in China (Series E), 2006,36(10):1037−1080 (in Chinese with English abstract).

[4] Keller A, Ludwig H. Defining and monitoring service level agreements for dynamic e-Business. In: Proc. of LISA 2002: The 16th Systems Administration Conf. Berkeley: USENIX Association, 2002. 189−204.

[5] Huang T, Chen NJ, Wei J, Zhang WB, Zhang Y. OnceAS/Q: A QoS-enabled Web application server. Journal of Software, 2004, 15(12):1787−1799 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/12/1787.htm

[6] Shercliff G, Shao JH, Gray WA, Fiddian NJ. QoS evaluation of providers with complex behaviours: A expectation-based approach with confidence. In: Dan A, Lamersdorf W, eds. Proc. of the 1st Int'4 Conf. on Service Oriented Computing (ICSOC 2006). Berlin/Heidelberg: Springer-Verlag, 2006. 378−389.

[7] Zeng LZ, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. IEEE Trans. on Software Engineering, 2004,30(5):311−329.

[8] Russel S, Norvig P. Artificial Intelligence: A Modern Approach. 2nd ed., Prentice Hall, Inc., 2002.

[9] Dougherty J, Kohavi R, Sahumi M. Supervised and unsupervised discretization of continuous features. In: Prieditis A, Russell SJ, eds. Machine Learning: Proc. of the 12th Int'l Conf. ACM Press, 1995. 194−202.

[10] Nakamura K, Sakashita N, Nitta Y, Shimomura K, Tokuda T. Fuzzy inference and Fuzzy inference processor. IEEE Micro, 1993, 13(5):37−48.

[11] Josang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. Decision Support System, 2005.

© 中国科学院软件研究所 http://www.jos.org.cn

url://citeseer.ist.psu.edu/738255.html

[12] Resnick P, Sakashita N, Nitta Y, Shimomura K, Tokuda T. Reputation systems. Communications of the ACM, 2000,43(12):45−48.

[13] Screenath RM, Singh MP. Agent-Based service selection. Journal on Web Semantics (JWS), 2004,1(2):141−154.

[14] Xu F, Lü J, Zheng W, Cao C. Design of a trust valuation model in software service coordination. Journal of Software, 2003,14(6): 1043−1051 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1043.htm

[15] Wang Y, Lü J, Xu F, Zhang L. A trust measurement and evolution model for Internetware. Journal of Software, 2006,17(4): 682−690 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/682.htm

[16] Deora V, Shao JH, Gray WA, Fiddian NJ. A quality of service management framework based on user expectations. In: Orlowska ME, Weerawarana S, Papazoglou MP, eds. Proc. of the 1st Int'l Conf. on Service Oriented Computing (ICSOC 2003). Berlin/ Heidelberg: Springer-Verlag, 2003.104−114.

[17] Kalepu S, Krishnaswamy S, Loke SW. Reputation=*f* (User ranking, Compliance, *Verity*). In: Zhang LJ, Heights Y, eds. Proc. of IEEE Int'l Conf. on Web Services. Washington: IEEE Computer Society, 2004. 200−209.

[18] Yu T, Zhang Y, Lin KJ. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Trans. on Web, 2007,1(1).

[19] Doshi P, Goodwin R, Akkiraju R, Verma K. Dynamic workflow composition using Markov decision processes. In: Proc. of the IEEE Int'l Conf. on Web Services. Washington: IEEE Computer Society, 2004. 576−584.

附中文参考文献:

[1] 杨芙清.软件工程技术发展思索.软件学报,2005,16(1):1−7. http://www.jos.org.cn/1000-9825/16/1.htm

[2] 杨芙清,梅宏,吕建,金芝.浅论软件技术发展.电子学报,2002,30(12A):1901−1906.

[3] 吕建,马晓星,陶先平,徐锋,胡昊.网构软件研究与进展.中国科学(E 辑),2006,36(10):1037−1080.

[5] 黄涛,陈宁江,魏峻,张文博,张勇.OnceAS/Q:一个面向 QoS 的 Web 应用服务器.软件学报,2004,12(15):1787−1799. http://www.jos.org.cn/1000-9825/12/1787.htm

[14] 徐锋,吕建,郑玮,曹春.一个软件服务协同中信任评估模型的设计.软件学报,2003,14(6):1043−1051. http://www.jos.org.cn/1000-9825/14/1043.htm

[15] 王远,吕建,徐锋,张林.一个适用于网构软件的信任度量及演化模型.软件学报,2006,17(4):682−690. http://www.jos.org.cn/1000-9825/17/682.htm

吴国全(1979－),男,安徽合肥人,博士生,主要研究领域为软件工程,网络分布式计算.

黄涛(1965－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为软件工程,网络分布式计算.

魏峻(1970－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为软件工程,网络分布式计算.

© 中国科学院软件研究所 http://www.jos.org.cn