

文章编号:1001-9081(2006)11-2661-03

基于语义的 Web 服务发现核心技术研究

员红娟,叶飞跃,李霞,彭文滔

(上海大学 计算机工程与科学学院,上海,200072)

(iamyun0371@sohu.com)

摘要:在研究 Web 服务本体描述语言 OWL-S 缺乏对服务质量的明确刻画的基础上,引入服务质量本体 QoSOn 对 OWL-S 规范进行扩展。其次,提出了“两阶段”匹配模型,该算法首先经过基于服务分类的匹配剔除不相关的服务,然后引入服务相似度来度量候选广告服务和请求服务之间的相似程度;候选服务和请求服务之间的相似度是通过 OWL-S Profile 的功能相似度和 QoSOn 本体的服务质量相似度两个方面进行综合评估的。最后,通过对比实验证明了该匹配算法的可行性和有效性。

关键词:Web 服务发现;服务质量本体;匹配算法

中图分类号:TP311 **文献标识码:**A

Central technology of Web service discovery based on semantics

YUN Hong-juan, YE Fei-yue, LI Xia, PENG Wen-tao

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

Abstract: Because Web Ontology Language Service (OWLS) was lack of clear depiction to Quality of Service (QoS), QoSOn was introduced to extend OWL-S. Then, a matching model named "Two Phases" was proposed. In this model, the irrelevance services were eliminated by matchmaking based on classification, and similarity functions were introduced to rank semantic similarity between the candidate services and request service. Both OWL-S profile functional similarity and QoS similarity based on QoSOn were evaluated to measure the similarity between the candidate services and the request. The matching algorithm is proved to be feasible and effective by comparative experiments.

Key words: Web services discovery; Quality of Service(QoS) ontology; matching algorithm

0 引言

Web 服务发现旨在实现从众多的 Web 服务中定位到满足要求的服务,服务发现的效果直接关系到服务调用的质量,影响服务组合的相容性和可替换性,关系到能否真正实现服务的即插即用,而 Web 服务描述语言和匹配算法是实现有效的 Web 服务发现的两大核心技术。

Web 服务发现作为 Web 服务系统架构中的一个重要组成部分,引起了工业界和学术界研究者的关注。工业界主要以能够快速应用到商业系统中为目的制定了一系列标准,比如 SOAP、WSDL 和 UDDI,由于采用了语法级的服务描述语言和基于关键字的匹配算法,传统的 Web 服务发现机制查准率低。学术界主要致力于智能化的 Web 服务框架的研究,经典的研究有,Paolucci Massimo 提出的基于 DAML-S 描述语言的 augment UDDI 语义 Web 服务注册系统^[2],采用的匹配算法^[3]主要是利用本体概念之间的包含关系来实现 Web 服务的输入/输出的匹配,但是该算法仅仅实现了同一本体中的概念之间的匹配,并且没有考虑服务质量的匹配,导致匹配的范围过窄且有效性不足。美国乔治亚大学 LSDIS 实验室的 METEOR-S 系统^[4],采用语义本体论推理和 QoS 策略相结合的服务匹配和筛选方法,由于使用的是简单的基于服务模板的匹配算法,故而不能支持复杂的逻辑推理。国内的研究有中国科学院的史忠植等人研制的多主体服务环境 MAGE^[5]和

基于描述逻辑的主体服务匹配算法。国防科技大学的胡建强等人提出来的 QWSDL 描述语言和“多层次、五类型”的匹配模型^[6],主要通过基本描述、基调描述、服务质量来对 Web 服务进行描述,但对 Web 服务的分类等信息考虑不够。

本文在研究目前常用的 Web 服务本体描述语言(Web Ontology Language Service, OWL-S)存在的没有提供服务质量度量的基础上,引入服务质量本体 QoSOn 对 OWL-S 规范进行了扩展。并且提出了一种有效的“两阶段”服务匹配算法。该算法首先进行基于服务分类的匹配,筛选出与请求服务分类相关的服务作为候选服务集,然后进行服务的相似度匹配,候选服务和请求服务之间的相似度是 OWL-S Profile 的功能相似度和 QoSOn 本体的服务质量相似度的加权平均值。最后,通过对比试验证明了该算法与经典算法^[3]相比具有较高的查全率(86%)、查准率(91%),并且匹配到的服务既能满足用户在功能方面的需求也能满足在非功能方面的需求。

1 Web 服务描述语言

OWL-S^[7]是一种基于本体的描述 Web 服务属性和功能的规范,OWL-S 本体结构是由 Service Profile, Service Model 和 Service Grounding 三个属性构成的,这三个属性分别描述了服务具备的功能、服务如何执行、服务如何访问的语义信息,其中 Service Profile 提供 Web 服务的功能信息供服务请求者判断该服务是否符合其需求。虽然 OWL-S 的 Profile 中有指定

收稿日期:2006-05-12;修订日期:2006-07-24

作者简介:员红娟(1977-),女,河南郑州人,硕士研究生,主要研究方向:计算机网络、数据库;叶飞跃(1959-),男,江苏泰兴人,教授,主要研究方向:数据库、模式识别、数据网格;李霞(1980-),女,山东淄博人,硕士研究生,主要研究方向:数据库;彭文滔(1981-),女,湖南长沙人,硕士研究生,主要研究方向:信息抽取。

服务质量参数的部分,如质量评估,质量等级等属性,但是规范中并没有提供服务质量描述的和属性的详细定义,也没有提供一种有关 QoS 度量的计算和评价方法。本文引入服务质量本体 QoSOn 对 OWL-S 规范进行扩展。

1.1 QoS 属性

服务质量显式地表达了一个产品或服务满足使用者需求的能力。根据领域的异同,可以分为领域独立的 QoS 和领域具体的 QoS。领域具体的 QoS 是指应用于特定领域的服务质量。领域独立的 QoS 可以从很多方面来描述,如响应时间、服务价格、性能、可靠性、可用性、安全性、信誉度、吞吐量、并发处理能力等。本文主要从最能表达服务质量的四个方面:响应时间、服务价格、可靠性、信誉度来对 Web 服务的质量进行描述,建立通用的领域独立的 QoS 本体。

响应时间 (ResponseTime) 指服务请求者提交请求到获得服务响应的的时间间隔,包括延迟时间和处理时间。

服务价格 (Cost) 指使用 Web 服务的相关费用。

可靠性 (Reliability) 表示 Web 服务能够正常运行的概率,是成功执行次数和总的调用次数的比率,由于 Web 服务是按次进行调用的,可以采用统计的方法来计算 Web 服务的可靠性。使用 $Reliability = s/t$ 表示,其中, s 表示成功运行的次数, t 表示总的调用次数。

信誉度 (ReputationDegree) 用于表征 Web 服务的信誉程度,反映了 Web 服务使用者对服务的评价。用 $[0,1]$ 的实数来表示,计算公式如下:

$$ReputationDegree = \begin{cases} 0, & T = 0 \\ \sum_{i=1}^n r_i/n, & T = 1 \end{cases} \quad (1)$$

其中, T 表示服务描述的真实性,它的值为 0 或 1。 r_i 表示从历史记录中获得的某个客户的评价, n 表示选取的客户数目。服务描述的真实性是影响语义 Web 服务的信誉度的首要因素。当 $T = 0$ 时,即服务提供者造假,则该服务的信誉度为 0,具有最差的信誉度;当 $T = 1$,即该服务的描述是真实的。服务的信誉度由客户的历史评价来衡量。

1.2 QoS 本体

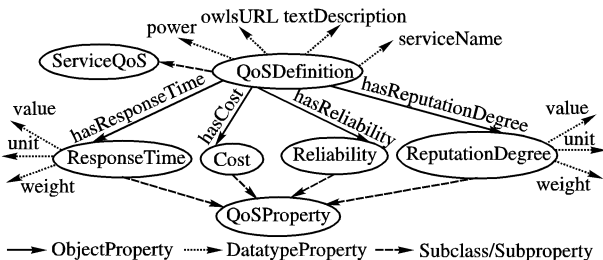


图1 QoSOn 的类层次结构

为了使不同 Web 服务的服务质量能够在共享语义的基础上进行比较,本文引入服务质量本体 QoSOn。QoSOn 采用两层模型来定义,即包括质量定义层 (QoS Definition Layer) 和质量属性度量层 (QoS Property Metric Layer)。

质量定义层 是一个 QoSDefinition 类,具有 4 个数据属性:serviceName 表示 QoS 本体所描述的 Web 服务的名字, textDescription 是 QoS 本体的一个简单的文字描述, owlsURL 是所描述的 Web 服务的 OWL-S 的 URL, power 表示服务质量的权值。对象属性:具有质量属性 (hasProperty), hasProperty 的取值范围为类 QoSProperty。

质量属性度量层 该层主要定义了一些通用的质量属

性, QoSProperty 类是所有质量属性的超类,通过子类继承关系来定义各服务质量的属性。QoSProperty 类有 5 个数据属性:属性名称 (propertyName)、文本描述 (textDescription)、权值 (weight)、单位 (unit)、值 (value),其中值取非负整数或非负小数。

QoSOn 使用 OWL 本体描述语言进行描述, QoSOn 的类层次结构如图 1 所示。

1.3 扩展后的 OWL-S 本体

QoS 在 Web 服务的发现、组合、调用、集成等过程中都将发挥一定的作用,因此有必要对 Web 服务的 QoS 进行明确地描述。由于 OWL-S 规范对 QoS 的支持不足,本文引入 QoSOn 对 OWL-S 规范进行扩展,扩展后的 OWL-S 结构如图 2 所示。

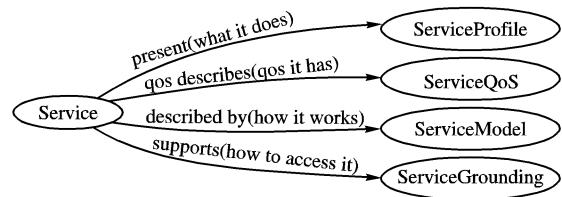


图2 扩展后的 OWL-S 本体结构

2 Web 服务匹配模型

在面向服务的架构 (Service-Oriented Architecture, SOA) 中,服务请求者向服务注册中心发出查询请求,服务注册中心将服务请求描述与广告描述进行匹配,服务发现的问题就转化为请求描述与广告描述之间的匹配问题。匹配算法是实现有效的 Web 服务发现的核心技术,服务匹配应该保证较高的查准率、查全率、良好的灵活性,同时应该具有较高的效率。本文提出了一种有效的匹配算法,基本思想是:进行基于服务分类的匹配,筛选掉和请求服务分类无关的广告服务;搜索所有的候选服务,计算每个候选服务与请求服务之间的服务相似度;如果服务相似度大于设定的阈值,则将该服务添加到结果列表中;根据服务相似度的大小对结果列表中的服务进行排序并返回。

2.1 基于服务分类的匹配

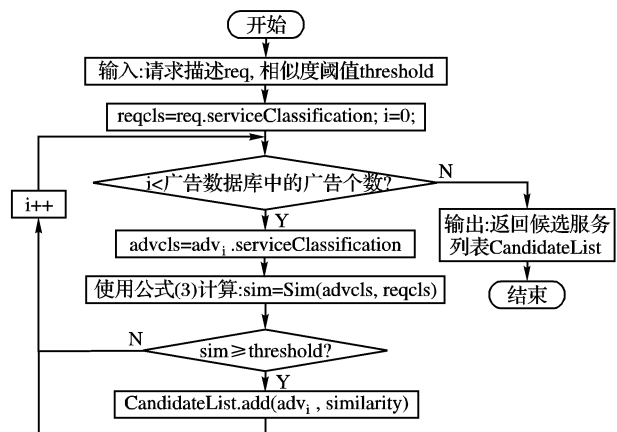


图3 基于服务分类的匹配算法流程

服务分类匹配可以筛选掉和请求服务分类无关的广告服务,明显降低了整个算法的时间复杂度,节省服务匹配的时间,提高匹配引擎的效率。本文使用 OWL-S Profile 规范中的 serviceClassification 来描述 Web 服务的分类信息,即 Web 服务的 serviceClassification 信息。通过计算请求服务和广告服务的 serviceClassification 相似度,返回相似度大于给定阈值的广告

列表,作为下一步匹配的候选服务。

2.2 服务相似度匹配

服务相似度可以用来衡量请求和广告服务之间的关联性,主要是采用数值计算的方法在语义层面上灵活地度量服务之间的相似程度。假设广告和请求分别用 adv , req 表示,则它们之间的相似度计算流程如下:

1) 若 $adv \in CandidateList$, 则计算 $s = Sim_{fun}(adv, req)$;

2) 若 $s \geq threshold$, 则计算 $S = (1 - \mu) * s + \mu * Sim_{QoS}(adv, req)$ 。

其中 $Sim_{fun}(adv, req)$, $Sim_{QoS}(adv, req)$ 分别表示广告和请求的功能相似度以及它们的服务质量相似度, μ 取请求服务 QoS 描述中的服务质量的权值(power)。

2.2.1 功能相似度

OWL-S Profile 规范通过 Input(输入)、Output(输出)、Precondition(前提条件)和 Effect(结果)4 个属性来描述服务的功能。广告和请求的功能相似度的计算公式如下:

$$Sim_{fun}(adv, req) = w_1 Sim_{in}(req_{in}, adv_{in}) + w_2 Sim_{out}(adv_{out}, req_{out}) + w_3 Sim_{pre}(adv_{pre}, req_{pre}) + w_4 Sim_{e}(adv_e, req_e), \sum w_i = 1 \quad (2)$$

通过公式(2),功能的相似度计算就细化为输入(Sim_{in}),输出(Sim_{out}),前提条件(Sim_{pre}),以及结果(Sim_e)集合的相似度计算。下面以输入集合的相似度计算为例,其他集合的相似度计算方法类似。假设 req 的输入集合有 m 项,用 $in_{11}, in_{12}, \dots, in_{1m}$ 表示, adv 输入集合有 n 项,用 $in_{21}, in_{22}, \dots, in_{2n}$ 表示。

输入集合相似度计算算法如下:

1) 如果 $m = n = 0$, 则 $Sim_{in}(req_{in}, adv_{in}) = 1$, 算法结束;否则,继续。

2) 如果 m 和 n 只有一个为 0, 则 $Sim_{in}(req_{in}, adv_{in}) = 0$, 算法结束;否则,继续。

$$3) \text{ 构造相似度矩阵 } \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ r_{21} & r_{22} & r_{23} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ r_{m1} & r_{m2} & r_{m3} & \cdots & r_{mn} \end{bmatrix}$$

其中, r_{ij} 表示 in_{i1} 与 in_{2j} 两个实体类的相似度,计算方法见公式(3)。

4) 找到相似度的最大值 r_{ij} , 删除 r_{ij} 所在的行与列得到矩阵 \mathbf{R}' 。重复执行 4), 直到 \mathbf{R}' 为空, 得到相似度序列 s_1, s_2, \dots, s_k , 其中 $k = \text{Min}(m, n)$ 。该步骤的目的是找到两个输入集合的相似度配对结果中相似度最大的序列。

5) 计算 $Sim_{in}(req_{in}, adv_{in}) = \sum_{i=1}^k \mu_i * s_i, \sum \mu_i = 1$, 算法结束。

本文采用文献[8]中提出的实体类相似性度量方法,从同义词、属性、语义关系三个方面来衡量两个实体类的相似程度。设 a^p, b^q 分别是来自本体 p 和 q 的实体类, Sim_w, Sim_u 和 Sim_n 分别表示本体类基于同义词、属性和语义关系的相似函数, w, u 和 n 是对应的权值, 则 a^p, b^q 的实体类相似函数为公式(3):

$$Sim_o(a^p, b^q) = w \times Sim_w(a^p, b^q) + u \times Sim_u(a^p, b^q) + n \times Sim_n(a^p, b^q) \quad (3)$$

2.2.2 服务质量(QoS)相似度

在进行服务匹配时,可能出现多个服务满足需求的情况,同时单纯从功能的相似性也无法判断出匹配到的服务在非功

能方面是否能满足用户的需求,比如:有的服务可能价格过高或者响应时间过长,有的服务可靠性不好或者提供者蓄意夸大服务的功能。为了给用户提供一种评价服务的手段,同时保证匹配到的服务的质量,剔除那些过时的服务或者信誉不好的服务,需要综合考虑服务的功能相似度和服务质量相似度,以使匹配到的服务既能满足功能方面的需求,也能满足服务质量方面的需求。服务质量相似度公式如下:

$$Sim_{QoS} = \mu_1 Sim_{RT}(adv, req) + \mu_2 Sim_C(adv, req) + \mu_3 Sim_R(adv, req) + \mu_4 Sim_{RD}(adv, req) \quad (4)$$

其中 $Sim_{RT}(adv, req), Sim_C(adv, req), Sim_R(adv, req), Sim_{RD}(adv, req)$ 分别表示广告和请求之间的响应时间相似度、价格相似度、可靠性相似度以及信誉度相似度,取值范围为 $[0, 1]$ 。 μ_i 取请求服务 QoS 描述中每个质量属性相应的权值(weight)。广告的价格可以由提供商具体给出,其他的三项从 QoS 历史数据库(QoSDB)中获得。

请求者期望广告的响应时间越小越好,响应时间相似度与请求的响应时间成正比,与广告的响应时间成反比。假设 QoSDB 中该广告的响应时间的平均值是 rt_{avg} 、最小值是 rt_{min} , 请求中设置的响应时间是 rt_{req} , 则响应时间相似度为:

$$Sim_{RT}(adv, req) = \begin{cases} 0, & rt_{req} \leq rt_{min} \\ \frac{1}{2} \left(\frac{rt_{req} - rt_{min}}{rt_{avg} - rt_{min}} + \frac{rt_{req}}{rt_{avg}} \right), & rt_{min} < rt_{req} < rt_{avg} \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

请求者期望价格越低越好,本文假设服务均为不免费的,假设广告的价格用 c_{adv} 表示,请求中设置的价格为 c_{req} , 价格相似度的计算公式如下:

$$Sim_C(adv, req) = \text{Min} \left(1, \frac{c_{req}}{c_{adv}} \right) \quad (6)$$

请求者期望广告的可靠性越大越好,可靠性相似度与请求的可靠性成反比,与广告的可靠性成正比。假设 QoSDB 中该广告的可靠性的最大值是 r_{max} 、平均值是 r_{avg} , 请求中设置的可靠性是 r_{req} , 则可靠性相似度为:

$$Sim_R(adv, req) = \begin{cases} 0, & r_{req} \leq r_{max} \\ \frac{1}{2} \left(\frac{r_{max} - r_{req}}{r_{max} - r_{avg}} + \frac{r_{req}}{r_{avg}} \right), & r_{avg} < r_{req} < r_{max} \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

请求者期望服务的信誉度越高越好,信誉度相似度的计算公式和公式(7)类似。

3 试验结果与分析

为了验证算法的有效性,开发了一个原型系统 SWSC(语义 Web 服务发现系统),本系统运行在 Windows 2000, JDK 1.5.0.6, Wordnet2.0 环境下,运行的硬件环境的技术指标为: CPU 为 P4 2.66GHz, 768M 内存。参考 OWLS-TC V2 服务测试数据集^[9], 选取了 100 个 service 并对部分 service 文件做了修改,加入了一些 precondition 和 effect 条件和分类等信息,制作了测试数据集 SWSC-TC, 测试了查准率、查全率两项技术指标,查准率定义为查询结果中与查询相关的服务数目和返回的服务总数目之比;查全率定义为查询结果中与查询相关的服务数目和广告数据库中查询相关的服务数量之比。

(下转第 2666 页)

为 $O(|U|^2|AT|)$ 。4) 求最小集合覆盖复杂度在 3) 的基础上需要考虑最多 $|AT|$ 个属性的区分集合, 复杂度为 $O(|U|^2|AT|^2)$, 整个算法的时间复杂度为 $O(|U|^2|AT|^2)$ 。

5 实例分析

如表 1 的不完备信息系统, $U/SIM(AT) = \{X1, X2, X3, X4, X5, X6\}$, 其中 $X1 = \{1\}$, $X2 = \{2, 6\}$, $X3 = \{3\}$, $X4 = \{4, 5\}$, $X5 = \{4, 5, 6\}$, $X6 = \{2, 5, 6\}$ 。差别矩阵如表 2。

表 1 不完备信息系统示例

Car	P	M	S	Ma
1	High	High	Full	Low
2	Low	*	Full	Low
3	*	*	Compact	High
4	High	*	Full	High
5	*	*	Full	Hige
6	Low	High	Full	*

表 2 差别矩阵

U_1	P	M	S	Ma	U_1	P	M	S	Ma
$U1 = \{X1, X2\}$	1	0	0	0	$U9 = \{X2, X6\}$	0	0	0	1
$U2 = \{X1, X3\}$	0	0	1	1	$U10 = \{X3, X4\}$	0	0	1	0
$U3 = \{X1, X4\}$	0	0	0	1	$U11 = \{X3, X5\}$	0	0	1	0
$U4 = \{X1, X5\}$	1	0	0	1	$U12 = \{X3, X6\}$	0	0	1	1
$U5 = \{X1, X6\}$	1	0	0	1	$U13 = \{X4, X5\}$	1	0	0	0
$U6 = \{X2, X3\}$	0	0	1	1	$U14 = \{X4, X6\}$	1	0	0	1
$U7 = \{X2, X4\}$	1	0	0	1	$U15 = \{X5, X6\}$	1	0	0	1
$U8 = \{X2, X5\}$	1	0	0	1					

由相关矩阵我们可以看到属性 M 这一列全为零, 这说明属性 M 对保持 $U/SIM(AT)$ 不变没有任何贡献, 可以直接约去。

(上接第 2663 页)

通过仿真性能测试与 augment UDDI Registry^[2] 做了一些比较。试验结果表明, 本系统的查准率 (91%) 略逊于 augment UDDI Registry 系统 (97%), 在查全率方面本系统 (86%) 明显高于 augment UDDI Registry (68%)。从分析得知, augment UDDI Registry 系统由于采用了基于 Prolog 推理的匹配算法, 返回的结果是经过精确匹配的服务, 所以具有很高的查准率。但是由于算法本身的匹配范围窄的局限, 造成了查全率不高。本系统采用了“两阶段”匹配算法, 综合考虑了服务的功能和服务质量的相似性。第一阶段的服务分类匹配剔除了与请求服务分类无关的广告服务, 明显降低整个算法的时间复杂度, 减少了第二阶段的匹配服务的个数, 节省匹配时间, 提高匹配引擎的效率, 并且不会对整个算法的查全率和查准率造成影响, 具有较好的匹配效果。

4 结语

本文通过引入服务质量本体, 对现有的 Web 服务描述语言 OWL-S 进行了扩展, 为基于语义相似度匹配算法提供了有效的服务描述语言。提出了“两阶段”匹配算法, 该算法应用在原型系统 SWSC 上, 试验结果表明较 augment UDDI Registry 系统的匹配方法能更大范围的定位服务, 并且在效率方面有所提高, 改善了 Web 服务发现的性能。

参考文献:

[1] TSALGATIDOU A, PILIOURA T. An Overview of Standards and

各属性的区分集合分别为: $S_P = \{U1, U4, U5, U7, U8, U13, U14, U15\}$; $S_M = \emptyset$; $S_S = \{U2, U6, U10, U11, U12\}$; $S_{Ma} = \{U2, U3, U4, U5, U6, U7, U8, U9, U12, U14, U15\}$ 。

由于集合 S_P, S_S, S_{Ma} 三个集合是集合 U_1 的最小集合覆盖, 根据算法则对应的三个属性组成的集合 $\{P, S, Ma\}$ 就是原不完备信息系统的最小属性约简。

参考文献:

- [1] 张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法 [M]. 北京: 科学出版社, 2001. 1-40.
- [2] WONG SKM, ZIARKO W. On optimal decision rules in decision tables [J]. Bulletin of Polish Academy of Sciences, 1985, 33(11/12): 693-696.
- [3] HU XH, CERCONE N. Learning in relational database: A rough set approach [J]. International Journal of Computational Intelligence, 1995, 11(2): 323-338.
- [4] JELONEK J, KRAWIEC K, SLOWINSKI R. Rough set reduction of attributes and their domains for neural networks [J]. International Journal of Computational Intelligence, 1995, 11(2): 339-347.
- [5] GUAN JW, BELL DA. Rough computational methods for information systems [J]. Artificial Intelligences, 1998, 105(1/2): 77-103.
- [6] 苗夺谦, 胡桂荣. 知识约简的一种启发式算 [J]. 计算研究与发展, 1999, 36(6): 681-684.
- [7] 王珏, 王任, 苗夺谦, 等. 基于 Rough Set 理论的“数据浓缩” [J]. 计算机学报, 1998, 21(5): 393-399.
- [8] 周献中, 黄兵. 基于粗糙集的不完备信息系统属性约简 [J]. 南京理工大学学报, 2003, 27(5): 630-635.
- [9] 黄兵, 周献中, 张蓉蓉. 基于信息量的不完备信息系统属性约简 [J]. 系统工程理论与实践, 2005, 25(4): 55-60.

Related Technology in Web Service [J]. Distributed and Parallel databases, 2002, 12(2/3): 135-162.

- [2] PAOLUCCI M, KAWAMURA T, PAYNE TR, et al. Importing the semantic Web in UDDI [A]. Proceedings of Web Services, E-business and Semantic Web Workshop (CAiSE Workshop) [C]. Toronto, Canada, 2002. 225-236.
- [3] PAOLUCCI M, KAWAMURA T, PAYNE TR, et al. Semantic Matching of Web Services Capabilities [A]. Proceedings of the 1st International Semantic Web Conference (ISWC) [C]. Sardinia, 2002. 333-347.
- [4] PATIL A, OUNDHAKAR S, SHETH A, et al. METEOR-S Web service Annotation Framework [A]. The Proceedings of the Thirteenth International World Wide Web Conference [C], 2004. 553-562.
- [5] 史忠植, 蒋运承, 张海俊, 等. 基于描述逻辑的主体服务匹配 [J]. 计算机学报, 2004, 27(5): 625-635.
- [6] 胡建强, 邹鹏, 王怀民, 等. Web 服务描述语言 QWSDL 和服务匹配模型研究 [J]. 计算机学报, 2005, 28(4): 505-513.
- [7] The OWL Services Coalition. Semantic markup for Web services (OWL-S) [S], 2004.
- [8] ANDREA RM, EGENHOFER MJ. Determining semantic similarity among entity classes from different ontologies [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(2): 442-456.
- [9] <http://projects.semwebcentral.org/frs/download.php/255/owlstc2.zip> [EB/OL], 2006.