

# 带情景约束和增强语义的 web 服务匹配方法<sup>①</sup>

孙珊珊 (湖南大众传媒职业技术学院 动漫与艺术设计系 长沙 410100); 贾金原 (同济大学 电子与信息工程学院 上海 200092)

**摘要:** 针对目前语义 web 服务匹配方法中缺少对服务情景约束的考虑及推理匹配的结果中同一级别缺少语义细分的缺点,设计了一种带情景约束和增强语义的相似度计算匹配方法。首先通过服务情景约束兼容性检测过滤掉不相关的服务;然后借助本体进行服务属性间的语义获取和语义相似度计算,得出待匹配服务的精确匹配度;最后对满足用户请求的服务进行反馈评价。实验结果表明,该方法较之传统的 OWL-S/UDDI 匹配方法在查准率和查全率方面得到了很大改进。

**关键词:** 语义 web 服务; 情景约束; 服务匹配; 语义相似度

## Matching of Semantic Web Service Based on Context Constrains and Enhanced Semantic

SUN Shan-Shan<sup>1</sup>, JIA Jin-Yuan<sup>2</sup> (1. Department of Animation and Art Designing, Hunan Mass Media Technical College, Changsha 410100, China; 2. School of Electronics and Information, Tongji University, Shanghai 200092, China)

**Abstract:** Aimed at lacking of considering the service context constraints and the deficiency of traditional reason matching whose result can't tell the semantic difference between the same degree in current matching algorithm of semantic web service, a new matching approach of semantic web service with service context constraints based on the reasoning and the enhanced computation of semantic similarity was proposed. First, this method filtered out the irrelevant services by checking compatibility of the service context constraints, then reasoning is applied to get different rank services, on the basis of rank services, the computation of semantic similarity was carried out by ontology to get more semantic information, in this way, the precise matching degree of services was gained; Finally, the evaluation of gained services which satisfied with the user's requirement was given. Compared to the traditional algorithm of OWL-S/UDDI, the experiment results showed that the approach could increase recall and precision ratio greatly.

**Keywords:** semantic web service; context constraint; service matching; semantic similarity

## 1 引言

目前国内外对语义 Web 服务匹配方法的研究越来越多的开始关注语义匹配和服务质量 QoS 的保证,语义匹配是指将语义信息加到服务匹配中来,比较具有代表性的是 Massim 等提出了一种基于 OWL-S 的服务匹配方法<sup>[1]</sup>,该方法通过领域本体基于推理进行服务匹配。文献<sup>[2]</sup>对上述方法进行了改进,在原有基础

上增加了一个匹配层次。文献<sup>[3]</sup>提出了一种基于语义相似度的匹配方法,在精确度上面有所提高,但是这种计算语义相似度的方法没有充分利用本体的语义信息。另一方面,服务质量 QoS<sup>[4-5]</sup>也备受关注,越来越多的应用到匹配方法中。但是目前语义 web 服务匹配方法中缺少对服务情景约束的考虑以及存在推理匹配的结果中同一级别缺少语义细分和语义信息利用不

<sup>①</sup> 基金项目:上海市重大攻关基金项目(085114501000)

收稿时间:2010-03-22;收到修改稿时间:2010-04-24

充分问题, 本文针对这些问题设计了一种带服务情景约束的基于推理和语义相似度计算的匹配方法。

本文的组织如下: 第二部分是叙述情景约束模型及其检测; 第三部分阐述基于语义推理和相似度计算策略; 第四部分给出带情景约束和语义增强的具体匹配算法; 第五部分是试验和结果分析。

## 2 服务情景约束模型及其检测

本文使用 OWL-S 作为描述语言, 为了支持服务情景约束, 我们对 OWL-S 进行扩展, 在 Profile 中加入描述服务情景约束的信息。

### 2.1 服务情景约束模型

情境指的是用来刻画用户和服务所处环境的信息, 用于尽可能全面准确的描述用户的基本信息、个性化偏好和 web 服务上下文。为此我们给出如图 1 所示模型。

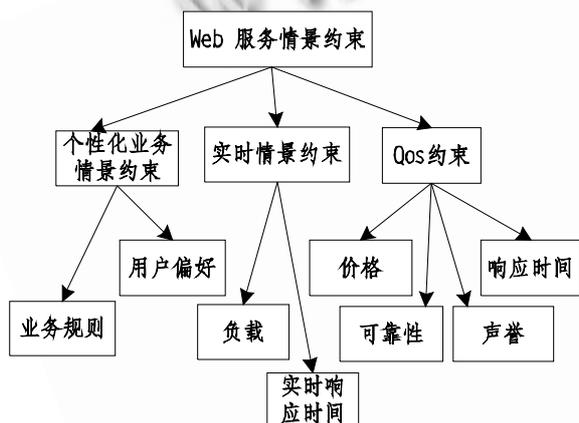


图 1 Web 服务情景约束模型图

服务情景约束模型从三个层面进行刻画, 其中用户偏好和业务规则描述个性化服务需求, 比如宽带服务开通业务中, 用户选择是先开通后付费还是先付费后开通, 不同的偏好对应不同的业务规则; 实时情景约束主要是针对实时性要求较高的服务请求, 实时情境属性由挂靠在服务端的传感器感知得到, 包括负载: 当前时刻 Web 服务并行处理的请求数, 以及实时响应时间: 当前时刻 Web 服务响应用户请求所花费的时间。服务质量 Qos 主要是质量保证、质量等级、质量评价等属性的描述, 本文采用服务价格(cost)、服务响应时间(time)、服务可靠性(reliability)、声誉(reputation)四个属性。

### 2.2 服务情景约束的兼容性检测

定义 1(Web 服务). 本文中一个 Web 服务用一个五元组(Is,Os,Ps,Es,CCs)来表示, 其中:

Is、Os、Ps、Es、CCs 分别表示服务输入、输出、前提、效果、情景约束的参数所引用的语义概念集合。

定义 2(情景约束兼容). 两个 web 服务 WS1 和 WS2 是情景约束兼容的, 当且仅当对于这两个服务所有的任何一个同种类型的约束 cck 不发生冲突。

服务的情景约束通过引用外部语义模型(如本体)中的概念, 将情景约束的信息加入到描述服务的本体中, 像输入输出参数一样, 以实时情景中的负载为例, 服务情景约束在 OWL-S 中的描述为:

```

<profile:hasContext rdf:resource=" Load" />
< profile:hasconstraint rdf:resource=" 4" />
  
```

如果存在个性化业务情景约束和实时情景约束, 通过预处理提取出这些约束信息, 然后进行服务匹配, 遇到不能匹配的约束 cck, 停止匹配并返回匹配度 0。如果这两种类型的约束 cck ( $1 \leq k \leq n$ )都是兼容的, 说明这两个服务是情景约束兼容的。考虑到服务质量 Qos 约束在很多情况下不像前两种约束那样严格, 必须要满足, 所以本文先利用前两种约束进行服务的筛选过滤, 然后通过相似度计算选择出良好的 Qos 服务列表供请求者选择。服务情景约束的检测算法可描述为:

```

Compatible(CCsreq,CCspro){
  if CCsreq=CCspro return true
  if CCsreq < CCspro return true
  else return false
}
  
```

## 3 基于语义推理和相似度计算的匹配策略

通过对服务情景约束进行兼容性检测, 过滤掉不能满足预期的服务之后, 接下来需要进行语义推理, 来得到服务匹配的等级, 对可能匹配的服务进行相似度计算<sup>[6]</sup>。

### 3.1 两个服务间的推理匹配

本文将服务匹配分为三个等级, 完全匹配用 Exact 表示, 匹配度为 1; 完全不匹配用 Fail 表示, 匹配度为 0; 有可能匹配的情况用 Possible 表示, 匹配度为 0-1 之间的某个值。对于输入的匹配具体做

法是将  $Ispro$ ,  $Isreq$  两个集合中的概念元素  $isk(1 \leq k \leq n)$  进行描述逻辑中的包含关系和相等关系推理, 结果会出现以下几种情况: (1) **Exact**:  $Isreq$ ,  $Ispro$  两个集合中的概念元素  $isk(1 \leq k \leq n)$  相同, 或者  $Ispro$  集合中的元素是  $Isreq$  集合中元素的直接子类; (2) **Fail**:  $Isreq, Ispro$  两个集合中的概念元素  $isk(1 \leq k \leq n)$  之间不存在任何包含关系; (3) **Possible**: 上述两种情况的之外的匹配, 对于这种情况将进一步进行概念间的语义相似度的计算。

以输出为例, 对每一对输出的算法可描述为:

```
degreeOfMatch(Osreq, Ospro){
  if Osreq Ospro then return exact
  if Osreq Ospro then return exact
  if (Osreq Ospro) || (Ospro Osreq)
  then return fail
  else return possible
}
```

### 3.2 Possible 级的语义相似度计算

本文中的概念是以本体的形式进行组织。一般而言两个概念间的语义距离越近, 其语义相似程度越高; 另一方面, 在本体中概念自上而下可以继承信息, 且包含共同祖先节点的数目越多, 概念的语义重合程度就越高, 其相似度也越大。

定义 3(概念的语义差异度). 用层次结构树中概念间包含相同上位概念的个数来度量, 用公式表示如下:

$$Diff(A, B) = \frac{1}{|sum(R, A) \cap sum(R, B)|}$$

$|sum(R, A)|$  和  $|sum(R, B)|$  分别表示节点  $A$  和  $B$  到根节点  $R$  的最短路径上节点的数目。语义差异度表示两个概念间的语义重合程度, 概念的语义差异度越小, 其相似度越大。

定义 4 (语义距离). 两个概念  $A, B$  的语义距离  $Dist(A, B)$  为连接它们的最短路径上  $n$  条边的权值的总和。用公式表示如下:

$$Dist(A, B) = \sum_{k=1}^n weight_k$$

其中  $weight_k$  表示连接  $A, B$  的最短路径上第  $k$  条边的权值。两个概念间的语义距离越小, 其相似度越高, 在层次树中离根较远的概念间的相似度要比离根近的

概念间相似度大些, 也即处于树中不同深度的边应该赋给不同的权值, 根较远的概念间的相似度要比离根近的概念间相似度大些。

定义 5 (概念  $C$  在树中的深度). 一个概念  $C$  在树中的深度  $Dep(C)$  等于连接该概念与根节点  $Root$  的最短路径所包括的边数, 设定根节点的深度为 1, 则非根节点的深度可以表示为:

$$Dep(C) = Dep(Parent(C)) + 1$$

定义 6 (概念  $C$  的密度). 用  $Wid(C)$  表示,  $Wid(C)$  的大小为其直接孩子节点的数目。

在层次结构树中, 概念密度越大, 其向下的分类就越多, 每个连接边分流的权值相应就越小, 因此深度和密度对权值的影响, 可以用下面的计算公式来计算权值:

$$Weight(A, B) = Weight(A) = \begin{cases} 1/wid(A) & \text{if } A \text{ is root} \\ 0.5 \times Weight(Parent(A)) / wid(A) & \text{else} \end{cases}$$

考虑到概念属于同义集合的情况, 我们借助 **WordNet**, 根据同义词在 **WordNet** 中的语义距离来确定其相似度。根据待比较概念的同义词是否存在公共上位词( $psw$ )两种情况来分别对待, 因此最终的语义距离度量如下所示:

$$Dis(A, B) = \begin{cases} Dist(A, B) & \text{if } A, B \text{ 不具有相同的公共上位词} \\ Dist(A, psw(A, B)) + Dist(B, psw(A, B)) & \text{else} \end{cases}$$

层次树中的概念包含了层次分类细化程度的信息, 而且这种信息自上而下可以传播和继承, 也就是子节点包含有祖先节点的信息。概念信息量是概念信息的一个量化值, 概念信息越丰富, 概念的信息量越大。概念共享的信息越多, 其相似度越大。

定义 7 (概念  $C$  的信息量). 概念信息量  $Info(C)$  用来度量概念所包含的语义价值, 用如下公式表示:

$$Info(c) = 1 - \log(hypo(c) + 1) / \log(hypo(root) + 1)$$

其中  $hypo(c), hypo(root)$  分别代表概念  $c$  所有子孙节点个数与本体树中的所有节点个数。从该公式中可以看出, 如果以树根节点为上, 从上往下看, 概念信息量是一个递增过程, 孩子节点的信息量大于父亲节点。

定义 8 (边的强度). 有向边的强度  $Edge(A, B)$  用连接该边的概念的信息量之差表示:

$$Edge(A, B) = |Info(A) - Info(B)|$$

定义 9 (概念间的信息量). 连接两个概念间最短

路径上边的强度之和。用公式表示为:

$$Info(A, B) = \sum_{k=1}^n Edge_k$$

综合考虑以上影响概念间语义相似度的因素, 提出相似度的计算公式:

$$Sim(A, B) = 1 - \sqrt{\frac{Diff(A, B) \times \frac{|Dep(A) - Dep(B)|}{Dep(A) + Dep(B)} \times Dist(A, B) \times \frac{1}{Info(A, B)}}{1}}$$

对于服务质量 Qos 的匹配, 我们用下面的公式进行计算:

$$Sim_{Qos}(SR, SA) = \sqrt[4]{Time(SR, SA) \times Cost(SR, SA) \times Rel(SR, SA) \times Rep(SR, SA)}$$

其中,  $Time(SR, SA) = 1 - \frac{|SR.time - SA.time|}{SR.time}$ ;  $Cost(SR, SA) = 1 - \frac{|SR.cost - SA.cost|}{SR.cost}$ ;

$$Rel(SR, SA) = 1 - \frac{|SR.rel - SA.rel|}{SR.rel}; Rep(SR, SA) = 1 - \frac{|SR.rep - SA.rep|}{SR.rep}$$

需要指出的是, 服务质量方面可能具有欺骗性, 存在恶意服务, 导致服务质量数据存在很大的不可信, 为此我们采用基于客户评价的声誉这一指标来权衡质量可信度的方法。为了防止客户的恶意打分, 使用方差来计算服务的声誉与期望值的偏差, 其中  $\overline{rep}$  是用户评价该服务的声誉平均值:

$$S^2 = \frac{1}{n} \sum_i (rep_i - \overline{rep})^2$$

在相同声誉得分下, 方差越小, 则该服务的声誉越可信, 如被选中则采用激励机制给该声誉一个适当的增量。

#### 4 带情景约束和增加语义的匹配算法

通过第 3 部分所述, 服务匹配的相似度计算可用以下公式来计算:

$$Sim(S_{req}, S_{pro}) = k_1 Sim_{IO}(S_{req}, S_{pro}) + k_2 Sim_{PE}(S_{req}, S_{pro}) + k_3 Sim_{Qos}(S_{req}, S_{pro})$$

其中,  $k_1 + k_2 + k_3 = 1$ 。由于用户的输入输出的要求高于前提和效果, 也高于 Qos, 因此  $k_1 > k_2$  且  $k_1 > k_3$ 。

算法. 带服务情景约束和增强语义的服务匹配算法

输入: 请求服务的五元组 ( $IS_{req}, OS_{req}, PS_{req}, ES_{req}, CCS_{req}$ )

输出: 提供的服务及匹配度

Step1. 进行预处理, 得到标准化的请求服务的五元组, 并置匹配列表 machList 为空;

Step2. 计算 Compatible( $CCS_{req}, CCS_{pro}$ ) 的结果, 如果为 true, 转 step3, 否则返回匹配度 0, 转 step5;

Step3. 计算 degreeOfMatch( $S_{req}, S_{pro}$ ), 如果返回 exact, 则加入该服务到 machList, 返回匹配度 1, 转 step5; 返回是 fail, 返回匹配度 0, 转 step5; 否则转 step4;

Step4. 进行相似度计算: 如果 ( $Sim_{IO} > t_1$  &&  $Sim_{PE} > t_2$ ), 判断 req.Qos 是否为 null, 若为 null, 则直接把该服务加入到 machList, 若不为 null, 进而判断是否  $Sim_{Qos} > t_3$ , 如果大于, 则把该服务加入 machList。其中  $t_1$ 、 $t_2$ 、 $t_3$  为匹配阈值, 然后对 machList 排序, 转 step5;

Step5. 返回满足要求的带精确匹配度的列表 machList 给请求用户, 并对服务进行评价和声誉打分。

## 5 实验及结果分析

### 5.1 测试数据集及评价标准

本算法使用公共测试集 OWLS-TC<sup>[7]</sup>中提供的数据进行测试, 该数据集涉及 7 个领域, 其中有 135 个 education 服务, 52 个 medical 服务, 25 个 food 服务, 106 个 travel 服务, 29 个 communication 服务, 206 个 economy 服务, 25 个 weapon 服务。评价标准使用信息检索领域的查全率<sup>[8]</sup>:  $R = A / (A + C)$  和查准率:  $P = A / (A + B)$ 。

### 5.2 对比实验及结果分析

记本文的匹配方法为 CSM, 下面是有关匹配结果的比较, 表中的表示匹配度, 通过对比基于关键字、OWL-S/UDDI 和 CSM 来观察三种方法的性能表现。通过实验, 规定匹配度大于 0.6 为满足服务请求的阈值, 小于该值认为不匹配。经过反复实验得到查全率、查准率对

比如下:

表 1 各系统间的查准率比较

System	Key_Based	OWL-S/UDDI	CSM
$s = 1$	0.26	0.81	0.96
$s > 0.6$	0.24	0.76	0.92
average	0.25	0.785	0.94

表 2 各系统间的查全率比较

System	Key_Based	OWL-S/UDDI	CSM
$s = 1$	0.28	0.86	0.88
$s > 0.6$	0.25	0.81	0.80
average	0.265	0.835	0.84

从实验结果可以看出,基于关键字无论在查全率还是查准率方面效果都是最差的,这是由于服务属性间的语义信息极少被利用。本方法在查准率方面得到了很大的提高,特别是当匹配率为 1 的时候,我们的方法达到了 0.96,而传统的方法只有 0.81,本文提出的匹配方法在性能方面得到了很大的提高,由于有了服务情景约束的兼容性检测,在推理匹配进行之前就可以淘汰大量的无用的服务,这里无用的服务是指那些通过推理匹配得到的结果是可以匹配的,但是实际上是不可以用的,比如说服务的用户个性化需求(选择先付费还是后付费)不满足等原因导致请求的服务与提供的服务不匹配。这样就可以大大减少系统运行时间,这是提高性能一个方面。另一方面是给出了一种有效的语义增强的相似度计算方法,这弥补了推理过程中语义信息不充分的缺点,同时返回请求服务与提供服务之间的精确匹配度。至于查全率方面,两种方法相差不大,由于我们并没有把 Qos 强制加入到情

景兼容性检测中,仅仅淘汰了确定无法满足约束的不相关服务,因此没有牺牲查全率来保证查准率,从另一个角度说明了本方法可取性。而且,本文对最终匹配得到的服务进行基于声誉方差来评价服务的可靠性,并适当的给予客观的声誉激励和惩罚,从而在一定程度上保证了 Qos 的性能。

#### 参考文献

- 1 Paolucci M, Kawamura T, Payne TR, Katia S. Semantic matching of web services capabilities. Proceedings of the 1st International Semantic Web Conference, 2002:34-43.
- 2 Lei Li, Ian Horrocks. A software framework for match Making based on semantic web technology. Proceedings of the 12th International Conference on World Wide Web, 2003:331-339.
- 3 彭晖,史忠植,邱莉榕等. 基于本体概念相似度的语义 Web 服务匹配算法. 计算机工程, 2008,34(15):51-53.
- 4 Kritikos K, Plesousakis D. Evaluation of Qos-Based Web Service Matchmaking Algorithms. Congress on Services, 2008:567-574.
- 5 RAN, SHUPING. A Model for Web Services Discovery with QoS. ACM SIGecom Exchanges, 2003,4(1):1-10.
- 6 Jian Wu, Zhaohui Wu. Similarity-based Web Service Matchmaking. Proceedings of the 2005 IEEE International Conference on Services Computing, 2005:287-294.
- 7 <http://www.dfki.de/~klus/owls-mx>.
- 8 张庆军. 本体映射中的推理修正与抽取算法研究 [硕士学位论文]. 长沙:中南大学, 2009.