

一个基于概念语义近似度的 Web 服务匹配算法

邱 田^{1,2}, 李鹏飞^{1,2}, 林 品^{1,2}

(1. 中国科学院软件研究所, 北京 100190; 2. 中国科学院研究生院, 北京 100190)

摘 要: 随着 web 服务的迅速发展和广泛应用, 高效的服务发现逐渐成为一个关键问题. 目前的 web 服务发现标准 UDDI(Universal Description, Discovery, and Integration), 存在不支持语义推理的局限性, 而且无法根据服务的能力和属性进行搜索, 限制了服务发现的效能. 针对这个问题可以采用在服务发现中引入 OWL-S 的方法. 本文提出了一个基于概念语义近似度的 web 服务语义匹配算法, 通过对服务的不同属性进行语义匹配计算, 从而提高服务发现的查准率和查全率. 算法基于一个关键的语义近似度量函数, 用以计算本体实体概念的语义近似度值. 测试算法的实验结果显示了算法的有效性.

关键词: web 服务发现; UDDI; OWL-S; 语义匹配算法; 近似度量函数

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2009) 02-0429-04

A Web Service Matching Algorithm Based on Semantic Similarity of Concepts

QIU Tian^{1,2}, LI Peng-fei^{1,2}, LIN Pin^{1,2}

(1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;

2. Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Web service discovery has been a key problem with the development of web services. The current service discovery standard, UDDI, has the limitation of lack of semantic inference support, and the mechanism which cannot search based on service capabilities and properties leads to a limited performance. The problem can be addressed by adopting OWL-S in web service discovery. In this paper a matching algorithm for service discovery is proposed based on semantic similarity of concepts, aiming at enhancing both recall and precision of web service search. The algorithm is based on a key function of semantic similarity measure that produces numeric similarity degree of ontology entity concepts. Experimental results are presented to show the effectiveness of the algorithm.

Key words: web service discovery; UDDI; web ontology language for services (OWL-S); semantic matching algorithm; similarity measure function

1 引言

Web 服务为 web 平台的整合与互操作问题提供了解决方案. 目前的 web 服务发现机制建立在统一描述、发现和集成 (Universal Description, Discovery, and Integration, UDDI)^[1] 的标准之上. UDDI 是基于可扩展置标语言 (EXtensible Markup Language, XML)^[2] 的 web 注册机制, 而 UDDI 对于服务搜索请求是通过关键字和分类信息来处理的. 虽然 UDDI 作为一个 web 服务基础架构标准已被广泛接受, 但在作为服务发现机制方面仍存在缺陷. 首先, XML 数据不包含语义信息, 因而不是机器可理解的. 句法不同的请求关键字可能在语义上是等同的. 因此 UDDI 基于句法的搜索导致了较低的查全率 (recall), 这是 UDDI 性能受限制的一个方面. 其次, 根据关键字

和分类信息的搜索并不适合 web 服务发现, 因为某个特定的关键字可能出现在完全不相关的服务描述中. 另外, UDDI 的关键字匹配过程并不提供对服务能力 (capability) 或其它属性的支持, 这也使 UDDI 搜索的查准率 (precision) 受到限制.

针对 UDDI 存在的局限性, 本文采用了 OWL-S (Web Ontology Language for Services)^[3]. OWL-S 是基于 Web 本体语言 (Web Ontology Language, OWL)^[4] 的用以描述 web 服务的高层本体语言. OWL-S 为服务描述定义了一套完整的用以描述服务的语义概念, 提供了一个搜索服务的语义途径, 能在查全率和查准率两个方面提高服务搜索的性能. 而且, OWL-S 根据服务能力和其他属性进行推理和语义匹配, 因而这种途径能够产生比基于关键字的搜索方法更准确的结果.

本文提出了一种基于 OWL-S 的 web 服务匹配算法用以提高 web 服务搜索的性能,该算法包括不同服务属性的匹配子算法以及一个核心的本体实体概念语义近似度量函数。

2 服务匹配算法

Web 服务匹配算法使用 OWL-S 作为上层本体.在算法中,服务搜索请求中的服务描述也被认为是 Service Profile 形式,与服务注册系统中的服务描述 Profile 进行匹配.由于权重的分配以及子匹配算法存在的差异,匹配算法中服务属性被重新分为五类:输入、输出、服务名、服务参数和服务质量(QoS).这里的服务参数包括 OWL-S 服务描述中的 service parameter 参数,但不包括服务质量参数.服务匹配算法包括匹配五个服务属性的子算法。

首先,一个已发布的服务与服务搜索请求中包含的服务描述匹配的前提是两者的服务名、输入和输出都匹配,只有这些关键属性都满足匹配,服务参数和服务质量的匹配度才被计算.否则,若三者中任何一个属性不能匹配,算法将返回零匹配度.所有服务属性的匹配自算法都依赖一个实体概念语义近似度量函数,该函数将在第 3 节中详细描述.算法也包括服务参数和服务质量的匹配子算法。

对服务名匹配子算法仅包括对服务名的语义近似度量函数的调用。

对于服务的输出,一个搜索请求与一个已发布服务输出属性的匹配需要满足:请求中的服务的每个输出属性都可在该已发布服务的输出属性中找到对应的匹配.而对于输入属性,需要满足的匹配顺序则相反,即请求中服务的输入属性包含该已发布服务的每个输入属性对应的匹配.输出属性的匹配子函数如图 1 所示.输入属性的匹配子函数除了匹配顺序以外,与输出属性的匹配过程相同。

```

matchOutputs(req,adv){
  outputsMatchDegree=0;
  similarityList=empty List;
  forall reqOutputs of req do{
    sim=0;
    forall advOutputs of adv do
      sim=max(sim,similarity(reqOutput,advOutput));
      if(isAcceptable(sim)) similarityList.add(sim);
      else return 0;
  }
  return average(similarityList);
}

```

图1 输出属性的匹配子算法

服务参数可能包含附加的服务信息.服务参数匹配子算法关注那些在服务搜索请求和已发布服务中都包含(即语义上可匹配)的属性.然而,如果服务搜索请求中并未定义任何服务参数,算法返回 100% 的匹配度.图 2 描述了该子算法.在最新的 OWL-S 上层本体中,profile 属性 Quality Rating 已被建议除去,而与服务质量等级相关的服务参数被采用.由于服务质量在匹配

算法中单独考虑,因此在该子算法中,与服务质量相关的属性被忽略。

```

matchServiceParams(req,adv){
  if(numberOfReqSP==0) return 1;
  List commonSPValues= empty List;
  forall reqSPs of req do{
    sim=0;
    if(isAcceptable(similarity(advSP,"QualityRating"))
      ignore this service parameter.
    forall advSPs of adv do
      sim=max(sim,similarity(reqSP,advSP));
    if(isAcceptable(sim))
      commonSPValues.add(similarity(reqSPValue,advSPValue));
  }
  return SUM(commonSPValues)/numberOfReqSP;
}

```

图2 服务参数的匹配子算法

如前所述,服务质量在目前的 OWL-S 的 Service Profile 中,一般以 Quality Rating 参数或服务参数的方式描述,本文假设服务质量属性仍以与 Quality Rating 类似的结构展现.类似于服务参数的匹配过程,在服务质量的匹配中,如果请求中并未定义与服务质量相关的属性,匹配子算法将返回 1 的质量匹配度值.该子算法在图 3 中描述.它包括匹配所有的服务质量属性值并返回平均的匹配度值.服务搜索请求的服务质量属性在服务的搜索中属于准则型属性,因此在匹配中如果已发布服务的对应属性的值不低于请求中的属性值,则该属性具有 100% 的匹配度,否则是 0 的匹配度。

```

matchQoS(req,adv){
  if(numberOfReqQoSProps==0) return 1;
  List qosPropValues= empty List;
  forall reqQoSProps of req do{
    sim=0;
    forall advQoSProps of adv do
      sim=max(sim,similarity(reqQoSProp,advQoSProp));
    if(isAcceptable(sim)){
      if(advQoSPropValue NOT lower than reqQoSPropValue)
        qosPropValues.add(1);
      else
        qosPropValues.add(0);
    }
  }
  return SUM(qosPropValues)/numberOfReqQoSProps;
}

```

图3 服务质量匹配子算法

Web 服务的匹配度是根据服务名、输入、输出、服务参数和服务质量属性的匹配度计算的.如果使用 SN 、 I 、 O 、 SP 和 QoS 来表示五种属性类型,并用 M 表示匹配算法对应的函数, M_{SN} 、 M_I 、 M_O 、 M_{SP} 和 M_{QoS} 分别表示五种属性的匹配子函数, w_1 、 w_2 、 w_3 、 w_4 和 w_5 分别表示五种属性的匹配度在整个服务匹配度计算中的权重.一个查找请求的服务描述 req 与一个已发布服务 adv 间的匹配度定义为每个部分匹配度的加权。

服务匹配算法的主回路将注册中心的所有已发布服务与搜索请求中的服务描述相匹配,并将匹配结果以有序方式排列。

算法中多次出现的 *similarity* 函数将在下一节中作详细阐述.关于 *isAcceptable* 函数,它将其参数与一个可接受的匹配度或近似度阈值进行比较,该阈值在实际系统实现上可由服务请求者指定.另外,如 M Paolucci 等^[5]所建议,为了提高算法的效率,服务搜索中近似概念的搜寻可限制在目标概念周围的小范围内,该范围以外的概念将视作无关或不匹配的概念。

3 概念语义近似度函数

在五种服务属性的匹配子算法中都引用了一个近似度量函数,用以计算两个实体的语义近似度.这个函数基于代表本体中概念上下级关系的树状结构,如图4所示.函数返回值为一个介于0和1之间的数值,代表近似度值.函数使用一系列关于两个实体概念的树节点的结构参数,包括两个树节点之间的最小距离,节点深度,以及两个节点相对其最近共同祖先节点的深度.设 R 为树状结构的根节点, a 和 b 分别为两个实体概念, d 为概念 a 和 b 之间的最小距离, l_a 和 l_b 分别为两个概念节点在树状结构中的深度, m_a 和 m_b 相对它们最近共同祖先节点的深度,这些参数变量都是非负整数,即 $d, l_a, l_b, m_a, m_b \in \mathbb{Z}^*$.

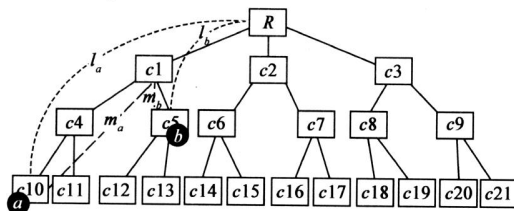


图4 一个本体树结构的例子

以下为这几个参数变量的近似度计算属性.

- (1) 对于给定 l_a 和 l_b , 概念 a 和 b 之间的语义近似度随着 d 的增加单调递减;
- (2) 对于给定 d 和限定值的 $|l_a - l_b|$, 概念 a 和 b 之间的语义近似度随着 $(l_a + l_b)$ 的增加单调递增;
- (3) 对于给定 d 和限定值的 $(l_a + l_b)$, 概念 a 和 b 之间的语义近似度随着 $|l_a - l_b|$ 的增加单调递减;
- (4) 对于给定 d, l_a 和 l_b , 概念 a 和 b 之间的语义近似度随着 $\max(m_a, m_b)$ 的增加单调递增.

语义近似度函数定义为一个符合以上计算属性的指数隶属函数 $S(a, b)$, 如式(1)所示. 在该函数定义中, t, α, β 均为实际可调的参数, 并且满足 $0 < t < 1, 0 < \alpha < 1, 0 < \beta < 1, \gamma > 0$. 其它符号均与前文定义的符合.

$$S(a, b) = t^d \times \alpha^{\max(|l_a - l_b|, \gamma)} \times \beta^{\max(m_a, m_b)} / (l_a + l_b + \gamma) \quad (1)$$

由函数定义可见,该函数还存在以下特性:

- (1) $0 \leq S(a, b) \leq 1$
- (2) $\forall a, S(a, a) = 1$
- (3) $\forall a, b, S(a, b) = S(b, a)$

第一个特性描述了函数的值域,即两个完全无关的概念的近似度为0,而两个完全等同的概念的语义近似度为或1.第二个特性描述了函数的自反性,即每个概念与它自身的语义近似度为1,或概念与其自身是等同的.最后一个特性描述了函数的对称性.这三个特性与概念语义近似度的共识相符^[6].

另外,语义近似度匹配函数假设两个参与比较的实体概念来自于同一个本体,当进行 web 服务匹配时,服务搜索请求中的服务描述可能包含来自于不同本体的概念,对于每个概念,需要与已发布的服务描述中的对应本体概念作比较,若不存在则返回0近似度.

函数的参数分配在较大程度上取决于所使用的本体树结构.根据实验数据,参数 t, α, β 和 γ 的建议赋值分别为 0.1、0.7、0.44 和 1.0. $isAcceptable$ 函数也根据采用的本体指定可接受的近似度阈值.

4 实验结果

为了测试概念的语义近似度以确定隶属度函数,以及评估服务匹配的精确度,语义近似度函数的相关实验已根据近似度函数和服务匹配算法实现和记录.实验关注点为服务搜索的查准率和查全率,这也是服务发现的关键要素.实验设计了多个 OWL 本体,包括动植物、食品、地域、质量、价格等.利用这些本体,创建了 139 个假想服务的 OWL-S 描述和对应的 UDDI 数据格式的描述.基于关键字的搜索模式和语义搜索都在实验中被测试.对于语义搜索,实验采用 0.6 作为搜索中可接受的匹配度阈值.另外,虽然实验中涉及的服务均为假想,但在实验中服务的请求者与服务提供者以及本体设计者不同,以保证实验数据的准确性和客观性.

在大量测试后,无意义(不存在与搜索请求中包含的服务描述相匹配的服务)的实验结果被剔除,并随机抽取 100 个关键字搜索结果和 100 个语义搜索结果的查准率,查全率和 F_1 -Measure 值.实验结果的统计见表 1.实验结果显示了语义服务搜索方法相对于关键字搜索的明显优势和有效性.

表 1 实验结果

匹配方法	测试数量	平均查准率	平均查全率	平均 F_1 -Measure 值
K^*	100	32.3 %	28.6 %	0.281
S^{**}	100	77.1 %	85.1 %	0.785

* K —传统的基于关键字的匹配方法

** S —本文提出的语义匹配算法

5 相关工作

近年来,在 web 服务发现和语义匹配方面已有相当多有意义的研究工作,其中也包括基于 OWL-S 的相关工作. Paolucci 等^[5]提出的语义匹配算法主要关注服务能力与本体概念的上下级关系,该算法根据本体树结构中的概念最小距离定义了四个匹配度值. Klusch 等^[7]的研究在文献[5]的基础上做出了新的拓展,即使用基于逻辑的推理和信息获取技术进行服务的匹配.另外, Li 与 Horrocks^[8]介绍的方法使用 DAML-S(OWL-S 的前

身)本体和描述逻辑推理工具 RACER 进行基于本体服务描述的比较,这种方法能够根据本体概念的树结构识别一系列匹配度。Hau 等^[6]提出了一种根据 OWL 概念的共有信息量进行概念语义近似度度量的方法,并可用于 web 服务的比较和匹配。Benatallah 等^[9]提出了基于 DAML-S 本体的 web 服务发现的自动化方法,这种方法将服务的发现视为一个重写(rewriting)的过程,并且能够发现符合请求的最佳的服务组合。Kifer 等^[10]基于 Web 服务建模本体(Web Service Modeling Ontology, WSMO)^[11]提出了使用形式化语义服务描述逻辑框架进行服务发现的方法。

这些研究工作提出的方法都没有将服务属性如服务参数以及服务质量评估作为匹配要素。另外,文献[5,7]和[8]的研究将服务匹配度划分为若干个固定级别,但却不能产生更细粒度和准确的数字结果,这就限制了服务发现的有效性。

6 结束语

本文提出了一个基于 OWL-S 的 web 服务匹配算法,目的在于提高服务搜索的查全率和查准率从而提高服务发现的效能。这个方法使用 Service Profile 描述对服务进行匹配,其中使用的关键服务属性包括服务名、输入属性、输出属性、服务参数和服务质量相关的属性。算法包括一个语义近似度度量函数用以度量服务属性中的概念间的近似度,该函数根据本体树结构的上下级关系返回数值近似度结果。服务匹配算法能够返回细粒度的服务匹配度值,从而提供了一种用于 web 服务匹配和发现的灵活和准确的方法。

本文的匹配算法根据 OWL 上下级关系规则进行服务属性的语义近似度的度量,下一步的研究工作将在语义近似度的度量和服务匹配算法中引入更多的 OWL 规则。另外,不同本体的概念实体间语义近似度的度量也是另一个值得深入研究的主题。

参考文献:

- [1] L Clement, A Hatley, C von Riegen, T Rogers. UDDI Version 3.0.2 [EB/OL]. <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>, 2004.
- [2] T Bray, J Paoli, C M. Sperberg-McQueen, E Maler, F Yergeau. Extensible Markup Language (XML) 1.0 (Fourth Edition) [EB/OL]. <http://www.w3.org/TR/2006/REC-xml-20060816/>, 2006.
- [3] D Martin, M Burstein, J Hobbs, O Lassila, D McDermott, S McIlraith, S Narayanan, M Paolucci, B Parsia, T Payne, E Sirin, N Srinivasan, K Sycara. OWL-S: Semantic Markup for Web Services [EB/OL]. <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>, 2004.
- [4] D L McGuinness, F van Harmelen. OWL Web Ontology Lan-

guage Overview [EB/OL]. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, 2004.

- [5] M Paolucci, T Kawamura, T R Payne, K Sycara. Semantic matching of web services capabilities [A]. Proceedings of the First International Semantic Web Conference (ISWC) [C]. London: Springer-Verlag, 2002. 333 - 347.
- [6] J Hau, W Lee, J. Darlington. A semantic similarity measure for semantic web services [A]. Web Service Semantics Workshop of the 14th International World Wide Web Conference 2005 [C]. Chiba, Japan, 2005.
- [7] M Klusch, B Fries, K. Sycara. Automated semantic web service discovery with OWLS-MX [A]. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems [C]. New York: ACM, 2006. 915 - 922.
- [8] L Li, I Horrocks. A software framework for matchmaking based on semantic web technology [A]. Proceedings of the Twelfth International Conference on the World Wide Web (WWW 2003) [C]. Budapest: ACM, 2003. 331 - 339.
- [9] B Benatallah, M-S Hacid, C Rey, F Toumani. Request rewriting-based web service discovery [A]. The Semantic Web-ISWC 2003 [C]. Berlin: Springer, 2003. 242 - 257.
- [10] M Kifer, R Lara, A Polleres, C Zhao, U Keller, H Lausen, D Fensel. A logical framework for web service discovery [A]. ISWC 2004 Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications [C]. Hiroshima, Japan, 2004.
- [11] H Lausen, D Roman, U Keller. Web Service Modeling Ontology-Standard (WSMO Standard) [EB/OL]. Digital Enterprise Research Institute (DERI), <http://www.wsmo.org/2004/d2/v0.2/>, 2004.

作者简介:



邱田男, 1979年生于河南南阳, 中国科学院软件研究所博士研究生。研究领域为语义 web 和 web 服务。

Email: qutian@ios.cn



李鹏飞男, 1974年生于河北定州, 中国科学院软件研究所博士研究生。研究领域为 web 服务与移动代理的安全协议。

Email: lipf@ios.cn