

基于本体的多约束服务发现

蒲国林¹ 邱玉辉²

(四川文理学院计算机学院 达州 635000)¹ (西南大学计算机与信息科学学院 重庆 400715)²

摘 要 随着 Web 服务的广泛应用,服务发现成为服务请求者和提供者之间的重要环节,服务发现机制的优劣直接关系到整个服务应用的质量。为了向服务请求者提供高质量的 Web 服务,提出一种基于本体的多约束服务发现机制,该服务发现机制在语义层面从多个角度选择满足服务请求者需求的最佳服务,从而提供更准确的服务,提高服务的性能;最后实验数据表明基于本体的多约束服务发现机制能在很大程度上提高查准率,进而可以明显地提高服务系统质量。

关键词 本体,推理,服务发现,多约束

中图分类号 TP311.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.5.040

Multi-constraints Service Discovery Based on Ontology

PU Guo-lin¹ QIU Yu-hui²

(School of Computer Science, Sichuan University of Arts and Science, Dazhou 635000, China)¹

(College of Computer and Information Science, Southwest University, Chongqing 400715, China)²

Abstract With the development of Web service, service discovery becomes an important link between the service requestor and service provider, and the quality of service applications is directly affected by the performance of the service discovery mechanism. In order to provide high-quality Web services to the service requester, multi-constraints service mechanism based on ontology was presented, which selects the best service to meet the needs of the service requestor from a semantic perspective with multi-constraints, thus providing a more accurate service and improving the performance of the service applications. Finally, the experimental results demonstrate that the proposed multi-constraints service mechanism based on ontology effectively improves the accuracy and the quality of service system.

Keywords Ontology, Reasoning, Service discovery, Multi-constraints

随着互联网的高速发展和广泛应用,互联网用户对服务的要求越来越高,被称为下一代 Web 范型的语义 Web^[1] 为 Web 信息查询提供了新的技术支持。语义网是对未来网络的一个设想,在语义网络中,信息都被赋予了明确的含义,机器能够自动地集成和处理网上可用的信息。语义网使用 XML 来定义定制的标签格式以及用 RDF 的灵活性来表达数据,下一步需要一种 ontology 的网络语言(比如 OWL)来描述网络文档中术语的明确含义和它们之间的关系。这些在研究领域都有了显著的研究成果^[2,3]。

基于语义的服务发现已经成为一个重要研究领域。文献[4]提出了一种功能语义的服务发现方法用于解决 Web 服务中知识的共享和重用;文献[5]提出了一种新颖的分布式服务发现协议,用具有语义层次结构的本体类和子类去描述服务;文献[6]对卫生保健领域的基于本体的服务发现机制做了深入的研究,对后续的研究工作产生了深远的影响;文献[7]提出了服务发现中需求匹配的服务质量问题,并做了简单的描述,然而却没给出具体的解决办法及可行方案;文献[8]提出

了一个涉及服务质量的服务发现的理论框架,并做了广泛的讨论。

针对现有工作的不足,本文提出一种基于本体的多约束服务发现机制,此服务发现机制从语义的角度选择满足服务请求者需求的服务,保证快速准确地查找到与其需求语义最匹配的服务,从而提供更佳的服务。本文实现了基于本体的服务发现框架,同时进行了实验分析和评估。实验结果表明,基于本体的多约束服务发现机制相对于一般约束的服务发现机制在查全率上相当,而在查准率上更有优势。

1 本体基础简介

本体(ontology)是语义 Web 应用的核心,是语义 Web 中基于语义的互操作能够得以实现的关键技术,它能使计算机理解 Web 信息真正包含的内容,从而实现 Web 信息的自动处理过程。更重要的是它还能通过推理发现本体信息中蕴含着的丰富的未直接表述的知识。

本体是共享的概念化规范,包括类、关系、实例和公理,可

到稿日期:2014-07-22 返修日期:2014-08-20 本文受国家自然科学基金:基于情感语义的全局均衡智能推荐理论与应用研究(61152003)资助。

蒲国林(1971—),男,博士,副教授,CCF 会员,主要研究方向为服务计算、人工智能;邱玉辉(1938—),男,教授,博士生导师,主要研究方向为人工智能。

形式化表示为 $O=(C,R,I,A)^{[9]}$ 。其中 C 为本体 O 中类的集合, I 为本体 O 中领域知识中所有对象实例的集合; R 为本体 O 中所有二元关系的集合; A 为本体 O 中所有公理的集合, 可能包含一般的逻辑公理和具体的领域公理。对于任意 $c \in C$, 如果实例 c 是 C 的实例, 则记为 $c \in C$ 。对于任意 $r \in R$, 若 $a, b \in I$ 且 $r=R(a, b)$, 则 r 表示实例间的二元关系; 若 $a, b \in C$ 且 $r=R(a, b)$, 则 r 表示类之间的包含关系。

为了构造更加完备的本体信息以支持自动推理, W3C 又推出了 OWL (Web Ontology Language)。OWL 在 RDF 和 RDFS 的基础上通过增加更多建模原语来描述类、特性、类的实例, 以及它们之间的关系^[11], 并针对特性提供了更加丰富的类型定义: 属性描述以及强有力的推理机制。OWL 用来描述 Web 文件或 Web 应用中使用的实体类和各类之间的关系, 以便支持描述 Web 环境中文件和应用的语义。它是在 DAML+OIL 的基础上发展起来的。推理是指依据一定的规则从已有的事实推出结论的过程。本体推理机主要是针对本体进行推理, 当前基于本体的推理越来越多地集中在标准的本体语言上, 如 RDFS/RDF、DAML、OWL 等, 在本文的实验中使用的 jena 就是基于 OWL 本体语言的本体推理机。

2 基于本体的多约束服务发现

基于本体的服务发现主要包含两大模块: 领域本体的建立和服务发现策略。

2.1 领域本体建立

为了更好地进行服务发现, 首先需要用本体语言对服务进行描述。本文用一个打印服务发现的例子来说明本体的定义。使用本体构建工具 Protégé 构建一个打印本体, 模式如下:

本体 Printer(打印机类型 hasPrintSytle, 纸张类型 canPrint, 处理现有任务时间 waitTasksTime, 打印机位置 hasLocateIn, 能否 Usable)。

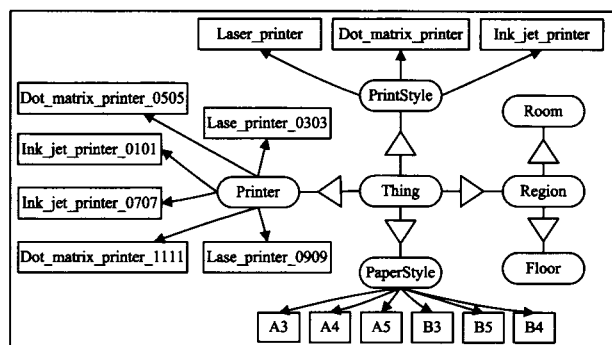


图1 本体组织结构概图

图1为Printer本体信息的图形化表示。Printer本体涉及了打印机方面的一些知识, 定义了打印机等一些类、属性和实例。例如Printer、PaperStyle、PrintStyle等均为类, Ink_jet_printer_0101就是类Printer的一个实例, hasPrinterStyle、canPrint、Usable等均为其属性。类似地, 本体定义了由7个打印机、6种类型纸张、3种打印机类型组合而成的实验测试样本集合。现取实例Ink_jet_printer_0101的代码片段如图2所示。

<!-http://www.semanticweb.org/ontologies/2011/11/PaintOnolo-

```
gy.owl # Ink_jet_printers_0101 -->
<rdf:Description
  Rdf: about = "http://www.semanticweb.org/ontologies/2011/11/
  Printer.owl # Ink_jet_printer_0101">
  <rdf:type
    Rdf: resource = "http://www.semanticweb.org/ontologies/2011/11/
    Printer.owl # Printer"/>
  <rdf:type
    Rdf: resource = "http://www.w3.org/2002/07/
    owl # NamedIndividual"/>
  <Printer:canPrint
    Rdf: resource = "http://www.semanticweb.org/ontologies/2011/11/
    Printer.owl # A5"/>
  <Printer:hasPrintStyle
    Rdf: resource = "http://www.semanticweb.org/ontologies/2011/11/
    Printer.owl # Ink_jet_printer"/>
  <Printer:hasLocateIn
    Rdf: resource = "http://www.semanticweb.org/ontologies/2011/11/
    Printer.owl # Room_1"/>
  <Printer:waitTasksTime>http://www.semanticweb.org/ontologies/2011/11/
    Printer.owl # 1</Printer:waitTasksTime>
  <Printer: Usable> http://www.semanticweb.org/ontologies/2011/11/
    Printer.owl # true</Printer:Usable>
</rdf:Description>
```

图2 Ink_jet_printer_0101实例片段

2.2 基于本体的多约束服务发现策略

本体 OWL 语义提供了丰富的推断规则, 本文利用语义 Web 推理工具(如 jena), 通过 sparql 查询语言可以对形成的三元组进行查询, 最后得到用户真正需要的服务中最佳的服务。

首先, 通过用户界面获得服务请求者的需求 Q (打印机类型 printer Q , 纸张类型 paper Q , 请求者位置 location Q);

然后, 编写用于推理的相关规则:

规则 1: if 打印机 p 能打印 A3 类型的纸张, then p 能打印 B3 类型的纸张;

规则 2: if 打印机 p 能打印 B3 类型的纸张, then p 能打印 A4 类型的纸张;

规则 3: if 打印机 p 的处理现有任务时间 time P 为 0, then p 是能提供的时间较短的服务之一;

规则 4: if 打印机 p 的处理现有任务时间 time P 为 0, 并且 if 打印机 q 的处理现有任务时间 time P 为 1, then p 比 q 能提供时间上更佳的服务;

规则 5: if p 比 q 能提供时间上更佳的服务, 并且 q 比 r 能提供时间上更佳的服务, then p 比 r 能提供时间上更佳的服务;

规则 6: if 服务请求者 a 位置在 m_n (m 楼 n 号房间), 打印机 p 位置为 x_y , 则 a 和 p 的空间距离 L 为 $|x-m| * k +$

$|y-n|, k$ 为参数, 实验中设为 2;
最后, 将上述规则结合本体数据建立推理机。当用户发出请求时, 可以编写查询语句, 发现机制对请求和服务进行语义上的服务匹配, 其中发现机制主体伪代码如图 3 所示。

```
//主体伪代码
model=Read(ontology file);
ruleString=Read(rules file);
rule=Load(rule base);
parseResult=rule.parse(ruleString);
reasoner=Build(parseResult);
infModel=CreateInfModel(resoner, model);
queryString=Create(query);
executQueryResultSet=Execur(queryString, infModel);
results=executQueryResultSet.execSelect();
//部分查询语句:
queryString=
    Select printerP, timeP, locationP, usableP
    where printerP canPrint paperQ
        printerP hasPrintStyle printerQ
        printerP Usable
        printerP min(waitTasksTime)
        printerP min(L)
        printerP canPrint A4
```

图 3 主体程序伪代码

3 系统体系结构与实验结果分析评估

3.1 系统体系结构

本系统为服务请求者提供满足功能需要的最佳服务, 如图 4 所示。其中主要模块的功能如下: 用户控制台, 能够解析用户请求信息, 并把解析结果传给推理机, 同时还能将推理机的处理结果呈现给用户; 推理机, 其中包含空间选优、时间选优、功能选优 3 个模块, 这 3 个模块综合推导出在空间、时间、功能上最优的服务来满足用户; 推理空间, 由规则库和事先构建的文本组成, 规则库包含通用规则和具体服务发现过程所需要的自定义规则, 该模块能在推理机的作用下找到需要的服务; 随意操作器, 为了模拟本体实例随时间不断变化的现实, 此模块能随机地改变某些本体实例中属性值, 来满足实验的需求, 最后为服务请求者提供简单明了的用户界面, 如图 5 所示。

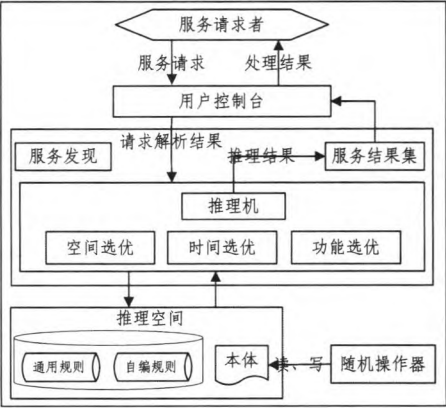


图 4 系统体系结构

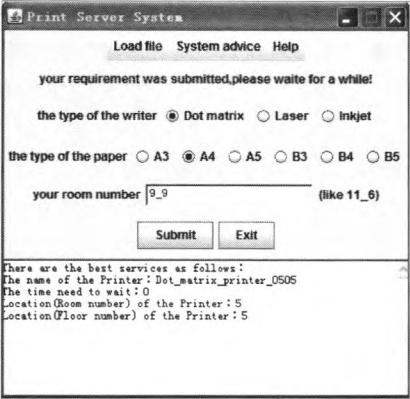


图 5 系统用户界面

3.2 实验结果分析

3.2.1 实验数据集和指标

实验数据采集开发的 Printer 本体包括 6 个类、6 个属性。分别从功能、时间、空间、可用性等角度对打印机做了比较详细的描述, 通过组合从而形成了一个包含 26 个实例的实验样本集合。

查全率和查准率的定义分别为:

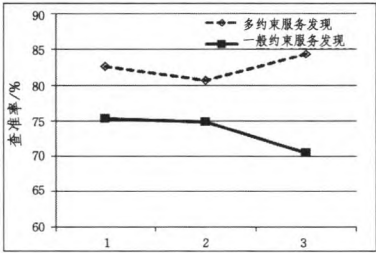
查准率 $P = \frac{CorrectNumber}{RetrievedNumber} \times 100\%$

查全率 $P = \frac{RetrievedNumber}{IndividualNumber} \times 100\%$

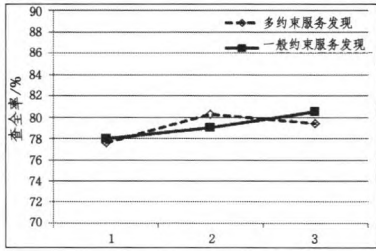
其中, $CorrectNumber$ 为 $RetrievedNumber$ 中包含的最佳服务的个数; $RetrievedNumber$ 为服务发现系统返回的服务个数; 而 $IndividualNumber$ 为整个样品集合包含的所有服务个数。

3.2.2 实验数据分析

首先采用基于本体的多约束服务发现机制, 实验做 3 次记录, 每次记录结果的查准率和查全率连续, 执行 10 次并取其平均值; 然后用一般约束服务发现方法, 记录 3 次, 每次记录结果的查准率和查全率, 连续执行 10 次并取其平均值, 分别得到图 6(a) 和图 6(b) 的实验比较结果。



(a) 查准率实验对比结果



(b) 查全率实验对比结果

图 6

3.2.3 实验数据结果分析

使用本文介绍的基于本体的多约束服务发现机制, 其查

全率和查准率都有不同程度的改善,特别是查准率有着明显的优化。这说明,基于本体的多约束服务发现能够提供更高精度的服务发现,这对于提高整个服务系统的性能有重要的意义。

3.2.4 实验结果解释

实验结果表明基于本体的多约束服务发现机制在查准率方面有很明显的改善,同时我们在实验过程中发现了导致这种结果产生的两类比较典型的数据,如表1和表2所列。

表1 第1类数据

| 说明 | 请求者需求 | 多约束服务 发现结果 | 一般服务 发现结果 |
|-----------|-------|---------------|--------------|
| 打印机类型 | 喷墨式 | 喷墨式 | 喷墨式 |
| 纸张类型 | A5 | A3 | A5 |
| 待服务时间 | 无要求 | 0 | 0 |
| 距离 | 无要求 | 2.2 | 4.4 |
| 打印机/请求者位置 | 5_5 | 7_7 | 1_1 |

表2 第2类数据

| 说明 | 请求者需求 | 多约束服务 发现结果 | 一般服务 发现结果 |
|-----------|-------|---------------|--------------|
| 打印机类型 | A4 | B3 | A4 |
| 纸张类型 | 无要求 | 0 | 1 |
| 待服务时间 | 无要求 | 2.2 | 4.4 |
| 距离 | 7_7 | 9_9 | 3_3 |
| 打印机/请求者位置 | A4 | B3 | A4 |

注:①实验基于一个假设:A3、B3、A4、B4、A5、B5能打印纸张类型为前者的打印机就能打印纸张类型为后者的纸张,比如,能打印B3类型纸的打印机就能打印A4、B4、A5、B5类型的纸张;②实验实例x_y中x表示楼层号,y表示房间号,如7_7表示7楼7号房间对应的打印机。

观察发现,基于本体的多约束服务发现机制的返回结果受两个方面的影响:1)该机制能够发现在语义上满足服务请求者的服务,这导致有较高的召回数量;2)该机制能从多方面综合考察某个服务是否为优质服务导致较少的召回数量。两个因素综合导致基于本体的多约束服务发现机制相对于一般约束的服务发现机制在查全率上相当,而在查准率上更有优势。

结束语 从Web服务发现的角度看,如何从大量满足功能的服务中选出最佳的服务,是提高服务质量的关键问题。本文提出的基于本体的多约束服务发现机制从不同于关键词匹配的语义匹配角度出发引入多约束的概念来发现最佳的服务,使服务发现精度提高,大大提高了服务系统的性能。如何

结合已有方法实现查全率的进一步提升将成为后续的研究内容。

参考文献

[1] Berners-Lee T, Hendler J, Lassila O. The semantic Web [J]. Scientific American, 2001, 284(5): 35-43

[2] Corby O, Dieng-Kuntz R, Faron-Zucker C. Searching the semantic web: approximate query processing based on ontologies [J]. IEEE Intelligent Systems, 2006, 21(1): 20-27

[3] Ji J, Bu F, Cai H, et al. Ontology model for semantic web service matching [M] // Information Computing and Applications. Springer Berlin Heidelberg, 2010: 181-188

[4] Yang Z, Chen J, Wu B. A new ontology-based service matching algorithm [C] // The 6th World Congress on Services. IEEE, 2010: 170-171

[5] 叶蕾, 张斌. 基于功能语义的 Web 服务发现方法 [J]. 计算机研究与发展, 2007, 44(008): 1357-1364

[6] Chakraborty D, Joshi A, Yesha Y, et al. Toward distributed service discovery in pervasive computing environments [J]. Transactions on Mobile Computing, IEEE, 2006, 5(2): 97-112

[7] Caceres C, Fernández A, Ossowski S, et al. Agent-based semantic service discovery for healthcare: an organizational approach [J]. Intelligent Systems, IEEE, 2006, 21(6): 11-20

[8] Kritikos K, Plexousakis D. Requirements for QoS-based Web service description and discovery [J]. Transactions on Services Computing, IEEE, 2009, 2(4): 320-337

[9] Zhou C, Chia L T, Lee B S. Semantics in service discovery and QoS measurement [J]. IT professional, 2005, 7(2): 29-34

[10] 刘志忠. 多本体环境下服务发现的相关技术研究 [D]. 武汉: 国防科学技术大学研究生院, 2007

[11] Meditskos G, Bassiliades N. Structural and role-oriented Web service discovery with taxonomies in OWL-S [J]. Transactions on Knowledge and Data Engineering, IEEE, 2010, 22(2): 278-290

[12] Segev A, Sheng Q Z. Bootstrapping ontologies for web services [J]. Transactions on Services Computing, IEEE, 2012, 5(1): 33-44

[13] Lacasta J, Nogueras-Iso J, Béjar R, et al. A Web Ontology Service to facilitate interoperability within a Spatial Data Infrastructure: Applicability to discovery [J]. Data & Knowledge Engineering, 2007, 63(3): 947-971

(上接第199页)

[13] Ramanarayanan R, Gupta M, Chakraborty S S, et al. Harnessing partial vectorization in Open64 compiler [C] // 2014 IEEE International Advance Computing Conference (IACC). IEEE, 2014: 813-824

[14] 魏帅, 赵荣彩, 姚远. 面向国产 CPUSW-1600 的向量重组 [J]. 计算机应用与软件, 2012, 28(11): 230-233

[15] Wu P, Eichenberger A E, Wang A. Efficient simd code generation for runtime alignment and length conversion [C] // International Symposium on Code Generation and Optimization, 2005

(CGO 2005). IEEE, 2005: 153-164

[16] Allen R, Kennedy K. Optimizing compilers for modern architectures: a dependence-based approach [M]. San Francisco: Morgan Kaufmann, 2002

[17] Leupers R. Code selection for media processors with SIMD instructions [C] // Proceedings of the Conference on Design, Automation and Test in Europe. ACM, 2000: 4-8

[18] Sreeraman N, Govindarajan R. A vectorizing compiler for multimedia extensions [J]. International Journal of Parallel Programming, 2000, 28(4): 363-400