基于 OWL-S 的服务发现语义匹配机制

邱 田[1,2],胡晓惠[1],李鹏飞[1,2],马恒太[1]
(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: 本文提出一种基于 OWL-S 的语义 web 服务匹配系统的机制,将语义元素引入 UDDI 系统中,从而在 UDDI 中保存语义信息.系统通过语义服务匹配算法提高 web 服务匹配的准确度和召回率,并通过数据映射机制保证模型对于当前的服务发现标准 UDDI 基础架构的兼容以及 UDDI 标准操作接口的透明性.系统使用本体概念的索引机制提高服务发现的效率.而且,系统在建立和维护索引的过程、或服务的匹配过程中使用近似概念搜寻算法进一步提高本体概念搜寻和服务匹配的效率.

关键词: web 服务发现;基于 Web 服务的高层本体语言;统一描述、发现和集成(UDDI);语义匹配系统;数据映射机制

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112(2010)01-0042-06

A Semantic Matchmaking System Mechanism for Web Service Discovery Based on OWL-S

QIU Tian[1,2], HU Xiao-hui[1], LI Peng-fei[1,2], MA Heng-tai[1]
(1. *Institute of Software, The Chinese Academy of Sciences, Beijing* 100190, *China*;
2. *Graduate University of the Chinese Academy of Science, Beijing* 100190, *China*)

Abstract: The paper proposes an OWL-S based semantic web service matchmaking system mechanism, which imports semantic elements into UDDI, and preserves semantic information in UDDI storage. The system employs a semantic service matchmaking algorithm to improve precision and recall of service matching, and uses a data mapping mechanism to ensure the compatibility with present web service discovery standards, UDDI, and the transparency of UDDI APIs. The paper designs an ontology concept indexing mechanism for the system to improve the efficiency of service discovery. Moreover, in the processes of building and maintaining the indices, or alternatively, matching services, the system uses a similar concept searching algorithm to further enhance the efficiency of ontology concept searching and service matchmaking.

Key words: web service discovery; web Ontology language for services (OWL-S); universal description, discovery and integration (UDDI); semantic matchmaking system; data mapping mechanism

1 引言

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[1].建立在统一描述、发现和集成(Universal Description, Discovery and Integration, UDDI)[2]标准之上的 web 服务发现机制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,但在作为服务的发现机制方面仍存在缺陷.首先,UDDI 的数据不包含语义信息,不是机器可理解的.UDDI 不能根据关键字或 tModel 进行推理,因此基于句法的搜索导致了较低的召回率.其次,根据关键字的搜索并不适合 web 服务发现,因为某个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或其他属性的支持,这也使 UDDI 的搜索准确度受到限制.

语义匹配是一个提高服务发现效能的途径.针对 UDDI 存在的局限性,本文采用基于 web 服务的高层本体语言 OWL-S(Web Ontology Language for Services)[3]的语义服务匹配方法.OWL-S 提供了一个搜索服务的语义途径,能在召回率和准确度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.

收稿日期:2008-12-10;修回日期:2009-02-10
基金项目:国家自然技术研究发展计划(No.2006AA701416);中国科学院支撑技术项目(No.61501030)

本文提出一种语义 web 服务匹配系统的架构.这种系统的机制将 OWL-S 语义元素引入 UDDI 服务描述,从而在 UDDI 中保存语义信息,并通过基于 OWL-S 的 web 服务匹配算法提高 web 服务匹配性能.语义匹配系统根据该算法进行服务的匹配和搜索,并通过数据映射机制、索引机制与近似概念搜寻算法进一步提高服务发布和发现的准确度和召回率.匹配系统不需改变当前 UDDI 的基础架构,仍保持 UDDI 标准操作接口的透明性.

2　匹配系统

2.1　匹配系统架构模型

UDDI 是一个以标准基于 XML[3]数据结构的用于存储 web 服务以及服务提供者信息的注册系统.UDDI 的数据可根据关键字和分类信息进行服务的搜索与发现.OWL-S 是用于描述服务的上层语义本体.本文提出的语义匹配系统在 UDDI 中保存 OWL-S 语义信息,并能够通过语义推理进行服务发现.

虽然 UDDI 不支持语义推理以及基于服务属性的匹配,但它目前仍是较完备的 web 服务注册和发现标准并已被广泛地接受.因此,UDDI 标准不应被直接替换,并且语义匹配系统的操作接口也需与原 UDDI 接口一致.本文的语义匹配系统遵循 UDDI 的标准和接口,图 1 为匹配系统的总体架构模型.

由图 1 可见,语义匹配系统包含三个模块,即消息转发模块(Message Forwarding Module)、映射模块(Mapping Module)和匹配模块(Matchmaking Module).其中,消息转发模块接收外部的 web 服务发布或查询请求.对于 UDDI 数据格式的请求消息,消息转发模块直接将消息传递至 UDDI 注册处并将结果返回.如果请求消息包含 OWL-S Service Profile 信息,此模块将消息传递至映射模块或匹配模块进行处理.映射模块通过将 OWL-S 数据模型转换为 UDDI 数据模型从而在 UDDI 中存储服务的语义信息,作者在文献[6]中对数据映射机制的细节做了详细描述.在搜索服务的过程中,匹配模块进行语义匹配处理.另外,为了提高系统效率,匹配模块还与两个数据索引表关联以减少搜索的耗时.当收到一个 OWL-S 发布请求时,语义转发模块将请求传递到映射模块.通过数据转换,映射模块将新的数据封装为新的 UDDI 数据请求消息,并将此消息继续传递到 UDDI 进行发布.对于 Service Profile 的查询请求,消息被直接转发至匹配模块,系统根据消息请求中的服务描述(包括服务名、输入、输出、服务参数和服务质量级别)进行匹配,匹配模块将匹配的服务指针通过消息转发模块从 UDDI 注册系统中获取服务信息.

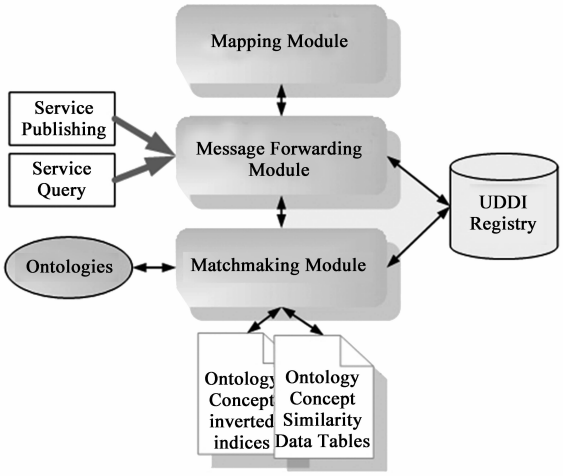

图1　语义匹配系统总体架构模型

2.2　匹配系统关键过程

在服务发现的过程中,如果 UDDI 中全部已发布服务都被访问并与请求服务进行匹配运算,将导致整个过程的低效.因此,系统使用和维护一套本体概念倒位索引和一套概念近似度数据表,以提高服务发现的效率.本节描述了基于语义匹配系统的服务发布与发现的关键过程.

2.2.1　服务的发布

语义匹配系统为每个本体维护一个本体概念倒位索引.每个本体概念树状结构都与一个索引记录关联,每个记录包括三部分倒位索引,它们将概念与其出现的位置(即已发布服务的服务名、输入或输出)关联.在发布服务时,语义匹配系统加载服务名、输入属性和输出属性所涉及到的本体,并更新本体的树结构.对于每个属性,匹配系统的匹配模块通过添加新的服务指针以更新该本体对应的索引.

另外,当加载新的本体时,系统将创建或更新一个与该本体对应的概念近似度数据表.一个本体的概念近似度数据表将每个本体概念与它近似的概念及其近似度相关联."近似的概念"是一个具有模糊度的概念,在这里可定义为具有大于某一特定值的近似度的概念.在数据表中,近似的概念以降序排列从而使服务发现的过程更有效率,譬如寻找近似度大于特定值的概念.表中记录的结构为某概念到与其近似概念的映射,因此定义每条记录起始的概念为主键概念.

系统使用近似概念搜寻算法(在第 3 节描述)来更新和维护一个本体中所有概念近似度数据,这个过程是非常耗时和耗资源的,所以系统在注册工作负荷相对低的时候执行本体概念近似度表的维护和更新,以避免影响其它进程如服务请求的处理.

2.2.2　服务的发现

当接收到一个包含 Service Profile 的服务发现请求

时,匹配模块首先从 Service Profile 中获得服务名、输入属性和输出属性.对于每个属性,匹配模块从概念近似度数据表中查询近似的概念并获得具有可接受的语义近似度的概念.对于每个近似概念,匹配系统将访问该概念对应的本体概念倒位索引并在对应的属性索引中试图找到已注册的服务,从而使系统获得全部的匹配结果.

当系统获得局部匹配的已注册服务以及这些服务与请求服务的服务名、输入和输出属性近似度值后,系统通过交运算获得在三种关键属性上都匹配的服务.利用作者在文献[7]中提出的服务匹配算法,系统可获得集合中每个服务与请求服务间的匹配度,排序并返回结果.另外,如果任何所需的近似概念在匹配系统查询时尚未更新,系统将直接使用近似概念搜寻算法来寻找近似概念.但这种情况发生的概率较低.

3 近似概念搜寻算法

如前所述,语义匹配系统维护与所有本体树状结构对应的概念近似度数据表,当处理请求时若所需的近似概念在匹配系统查询时尚未得到更新,系统也需要使用与更新数据表同样的算法来寻找近似概念.

对于本体树结构中的某个特定的概念,为了提高系统效率,这里提出的近似概念搜寻算法用以在某概念(假设为图 2 中的 $c17$)的"周围"寻找其近似概念.图 2 所示的本体概念树状结构示例可以用来更清晰地表述算法的细节.

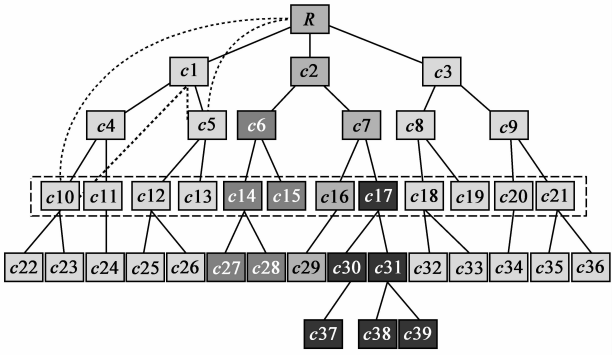

图2 一个本体概念树状结构的示例

在算法中多次出现"同辈(compeer)"的概念,在这里先予以定义.

定义 1 对于某一特定的树节点概念 n、p、a 和 o,当满足以下条件:

p 与 n 具有同样的深度;a 是 p 和 n 的共同祖先;p 不属于节点 o 的后代,

则称节点 p 是 n 在 a 之下且在 o 之外的同辈节点.如果 o 未确定,则称 p 是 n 在 a 之下的同辈节点.

以形式化的方式,对于根节点 $root$ 以及其全部后代节点,令 T_x 表示以概念 x 为根节点的子树,l_x 表示节点 x 的深度,Γ_n 表示全部与节点 n 具有同样深度的节点概念的集合,P_n 表示概念 n 的同辈节点的集合,那么有 $\Gamma_n = \{c \mid c \in T_{root}, l_c = l_n\}$,并且有 $P_n = \{T_a \cap \Gamma_n - T_o\}$.

例如,在图 2 中,$c37$ 是 $c38$ 的在 $c17$ 之下和 $c31$ 之外的唯一同辈,而 $c37$、$c38$ 和 $c39$ 在 $c17$ 之下互为同辈节点.

为了便于后文的描述,这里先定义了一系列符号.令 T_x 为以节点 x 为根的树形拓扑结构的所有概念集合,l_b^a 为在以节点 a 为根的子树结构中 b 的深度,l_b 等价于 l_b^{root},$m_{a,b}$为在以节点 a,b 的最近共同组先节点为根的子树结构中 a 的深度,$d_{a,b}$为节点 a,b 的距离,$Des(a)$表示节点 a 的后代节点概念集合,或者说,节点 a 之下的节点集合,$Chi(a)$表示节点 a 的子节点集合,$Ans(a)$表示节点 a 的祖先节点集合,$Sib(a)$表示节点 a 的兄弟节点集合,$cans(a,b)$表示节点 a,b 的最近共同祖先节点,$par(a)$表示节点 a 的父节点,$Cmp(a)$表示节点 a 在根节点之下的同辈概念集合,$Cmp^x(a)$表示节点 a 在节点 x 之下的同辈概念集合.即 $Cmp(a) \cap Des(x)$,$Cmp^{x|y}(a)$表示节点 a 在节点 x 之下且在节点 y 之外的同辈概念集合.即 $Cmp(a) \cap Des(x)\text{-}Des(x) \cap Des(y)$.

本体概念树状结构的不同节点间的概念近似度具有以下定理描述的特性.其中语义近似度函数 Similarity 在文献[7]中定义.

定理 1 在一个本体概念树形拓扑结构的所有概念集合 T_{root} 中,若概念 a 和 b 满足 $a, b \in T_{root}$, $b \in Cmp(a)$,那么若 $\forall p \forall p'(a \in Des(p) \wedge b \in Des(p') \wedge l_p = l_{p'})$,则 $p' \in Cmp(p)$;若 $\forall p \forall p'(p \in Des(a) \wedge p' \in Des(b) \wedge l_p = l_{p'})$,则 $p' \in Cmp(p)$;若 $\forall p \forall p'(p \in Sib(a) \wedge p' \in Sib(b))$,则 $p' \in Cmp(p)$.

定理 2 一个本体概念树形拓扑结构的所有概念集合 T_{root} 中,如果概念 a 和 b 满足 $a, b \in T_{root}$, $b \in Cmp(a)$,若 $\forall p \forall p'(p \in Des(a) \wedge p' \in Des(b) \wedge l_p = l_{p'})$ 则 $\text{Similarity}(a,p) = \text{Similarity}(b,p')$.

定理 3 在一个本体概念树形拓扑结构的所有概念集合 T_{root} 中,如果概念 a 和 b 满足 $a, b \in T_{root}$, $b \in Cmp(a)$,若 $\forall p \forall p'(a \in Des(p) \wedge b \in Des(p') \wedge l_p = l_{p'})$,则 $\text{Similarity}(a,p) = \text{Similarity}(b,p')$.

定理 4 在一个本体概念树形拓扑结构的所有概念集合 T_{root} 中,如果概念 a 和 b 满足 $a, b \in T_{root}$, $b \in Cmp(a)$,若 $\forall p \forall p'(p \in Sib(a) \wedge p' \in Sib(b) \wedge l_p = l_{p'})$,则 $\text{Similarity}(a,p) = \text{Similarity}(b,p')$.

定理 5 在一个本体概念树形拓扑结构的所有概

念集合 $\boldsymbol{T}_{root}$ 中,如果概念 a 和 b 满足 $a, b \in T_{root}$, $b \in Cmp(a)$,概念 $n = cans(a, b)$,那么,对于 $\forall p(p \in Out(n))$,有 $\text{Similarity}(a, p) = \text{Similarity}(b, p)$.

定理 6　一个本体概念树形拓扑结构 T_{root} 的所有概念集合 C 中,如果概念 n、o、a 和 p 满足 $n, o, a, p \in C$, $n \in Des(o)$, $o \in Chi(a)$,若有 $\forall p_1 \forall p_2(p_1, p_2 \in C \wedge p_1 \neq p_2 \neq a \neq o \neq n \wedge p_1, p_2 \in Cmp^{a \mid o}(n))$,则 $\text{Similarity}(p_1, n) = \text{Similarity}(p_2, n)$.

根据定义 1,有 $p_1 \in Cmp^{a \setminus o}(n) \Rightarrow l_n = l_{p1} \wedge p_1, n \in Des(a) \wedge p_1 \notin Des(o), n \in Des(o) \wedge o \in Chi(a) \Rightarrow p_1, n \in Des(a) \wedge p_1 \notin Des(o) \wedge n \in Des(o) \wedge o \in Chi(a) \Rightarrow a = cans(n, p_1), a = cans(n, p_1) \wedge l_n = l_{p1} \Rightarrow (m_{p_1, n} = m_{n, p_1} = l^a_{p_1} = l^a_n = l_{p_1} - l_a) \wedge (d_{p_1, n} = m_{p_1, n} + m_{n, p_1} = 2 \times (l_{p_1} - l_a))$.

同理 $(m_{p_2, n} = m_{n, p_2} = l^a_{p_2} = l^a_n = l_{p_2} - l_a) \wedge (d_{p_2, n} = m_{p_2, n} + m_{n, p_2} = 2 \times (l_{p_2} - l_a))$.

根据概念间语义近似度函数的定义描述,有 $\text{Similarity}(p_1, n) = \text{Similarity}(p_2, n)$.

图 3 为近似概念搜寻算法的主控制环路,算法搜寻与概念 reqConcept 近似的概念,另一参数 minSim 是所寻找的概念与所比较的概念近似度的最小可接受值.算法首先试图横向寻找最近的已在概念近似度数据表中作为主键概念保存过的同辈,如果找到任何一个,则根据已保存的同辈数据获得当前概念寻找近似概念.否则将根据近似度函数进行搜寻.所有找到的概念按近似度排序并被返回.另外,已获取的近似概念以"概念-近似度"映射对的方式保存在变量 simMap 中.

如图 3 所示,算法首先判断是否存在已保存近似度记录的同辈节点.例如,若寻找 $c17$ 的同辈,算法找到 $c7$ 之下且在 $c7$ 之外的同辈 $c16$.如果 $c16$ 无法找到已存记录,则 $c14$ 与 $c15$ 作为在 $c2$ 之下且在 $c7$ 之外的同辈被查询.若仍无法找到,则查找 R 之下且在 $c2$ 之外的同辈,即 $c10$-$c13$ 以及 $c18$-$c21$.

```
main(reqConcept,minSim){
    referenceConcept=findPreservedCompeerConcepts();
    if(referenceConcept!=null){
        get commonAncestor of referenceConcept and reqConcept;
        findSimilarConceptsBasedOnSavedData
                                (reqConcept,referenceConcept,commonAncestor);
    }
    else
        findSimilarConceptsBasedOnSimilarity(reqConcept);
    return sort(simMap);    //sort the map items by the similarity.
}
```

图3　近似概念搜寻算法的主环路

假设 $c14$ 作为 $c17$ 的同辈被找到并已作为主键概念保存在概念近似度数据表中,那么根据定理 1-5,与 $c17$ 近似的概念以及对应的近似度能够根据 $c14$ 与近似的概念及近似度而得到.例如,$c14$ 与 $c15$ 的近似度对应于 $c17$ 与 $c16$ 的近似度,而 $c14$ 与 $c27$ 的近似度对应于 $c17$ 与 $c30$ 的近似度等.在 $c2$ 之外的所有概念与 $c14$ 的近似度分别等同于它们与 $c17$ 的近似度.另外,如果 $c14$ 缺少与 $c17$ 等同的一些关系节点,譬如兄弟节点或 $c17$ 更远的后代节点,那么这些节点与 $c17$ 的近似度根据近似度函数单独计算.

另一方面,如果 $c17$ 没有已保存的同辈节点,那么将采用根据近似度函数搜寻的子算法.仍从 $c17$ 开始,首先 $c17$ 与其后代节点的近似度以由近至远的方式被度量和保存,然后是 $c17$ 与其祖先节点的近似度以距离由近至远的方式被度量和保存,直到近似度值不可接受或者到达根或叶节点.接下来算法进入一个循环:首先以 $c17$ 的父节点 $c7$ 为关注,$c17$ 在 $c7$ 之下且在 $c17$ 之外的同辈概念只有 $c16$,于是 $c16$ 所有的后代与 $c17$ 的近似度以由近至远的方式被度量和保存.值得注意的是,$c16$ 的后代中同深度的概念节点互为同辈,因此根据定理 6,它们与 $c17$ 的近似度是相同的,因此对于每个深度只用度量一个节点.接着概念 $c16$ 自身与 $c17$ 的近似度,以及 $c16$ 在到达 $c7$ 之前的父节点(只有 $c7$)与 $c17$ 之间的近似度被度量和保存.其次,算法关注的节点移至 $c7$ 的父节点 $c2$.同样地,$c17$ 在 $c2$ 之下且在 $c7$ 之外的同辈包括 $c14$ 和 $c15$,它们以及它们的后代与 $c17$ 之间的近似度被度量和保存.在到达 $c2$ 之前,它们有两个父节点 $c6$ 和 $c2$,二者与 $c17$ 的近似度以距离由近至远的方式被度量和保存.最后,算法循环的关注点移至整个树的根节点 R,在此之后,$c17$ 在整个树结构的近似概念都被度量和保存.

4　实验分析

以及评估服务匹配系统的机制、匹配算法与近似概念搜寻算法,针对服务搜索准确度、召回率和可扩展性的实验已实现和记录.实验在一台 Intel Pentium 4 (3.00GHz)处理器、内存 1GB 的 IBM PC 服务器上进行,操作系统为 Microsoft Windows XP, UDDI 环境为 jUDDI v2.0 ,OWL 推理工具为 Pellet.

首先,服务搜索的准确度和召回率是服务发现的关键要素.实验使用 Protégé 软件设计了若干 OWL 本体,并从互联网下载和选择了大量本体.这些本体包括动植物、生物化工、食品、地域、科技专业领域等多个范畴.利用这些本体,创建了 1000 个假想服务的 OWL-S 描述和对应的 UDDI 数据格式的描述,而且确保每个本体相关服务数量不少于 50 个.其次,构造 1000 组语义服务发现请求属性条件集,每组都随机地定义了输入和输出参数以及服务质量的匹配条件,部分请求属性集中包含服务名和服务属性的请求.对于这 1000 组语义服务发现请求属性条件集再定义一一对应的传统服务发现关键字条件集.实验系统把请求的语义属性条件集和关键字条件集分别封装为 OWL-S 服务发现请求

和 UDDI 服务发现请求.在进行实验之前,人工分析每组条件所对应的符合匹配的服务,剔除无意义(不存在与搜索请求中包含的服务描述相匹配的服务)的请求,并记录每个服务请求对应匹配的服务.

针对服务匹配的可扩展性的实验首先需从 1000 个服务中随机地分为 10 组 G_1-G_{10},每组 100 个服务,并将每组服务标注序号,然后产生 10 个服务样本集 S_1-S_{10}.其中 $S_n = \sum_{i=1}^{n} G_n$;将服务样本集 S_1 注册,并保证本体概念倒位索引和本体概念近似度数据表的完整;使用传统的服务匹配方法进行服务发现,并记录服务发现的时间;使用语义服务匹配方法进行服务发现,并记录服务发现的时间.更新服务样本集,分别用 S_2-S_{10}重复以上步骤,实验并记录.

对于语义服务发现,在以上所有实验中,采用 $\lambda = 0.6$ 作为搜索中可接受的匹配度阈值.

关于匹配系统的准确度和召回率实验结果统计见表 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.其中,*K* 代表传统的基于关键字的匹配方法,*S* 代表本文提出的语义匹配算法.

表 1 实验结果

匹配方法	实验数量	平均准确度	平均召回率	平均 F1-Measure 值
K	600	31.3%	27.0%	0.268
S	600	77.4%	86.9%	0.796

针对服务发现的可扩展性的实验结果对比如图 4 所示.其中,*TK* 代表传统的基于关键字的服务发现方法的服务发现响应时间统计;*TS*1 代表有完整本体概念近似度数据表和本体概念倒位索引的语义服务发现方法的服务发现响应时间统计;*TS*2 代表仅有完整本体概念近似度数据表的语义服务发现方法的服务发现响应时间统计;*TS*3 代表仅有完整本体概念倒位索引的语义服务发现方法的服务发现的响应时间统计.

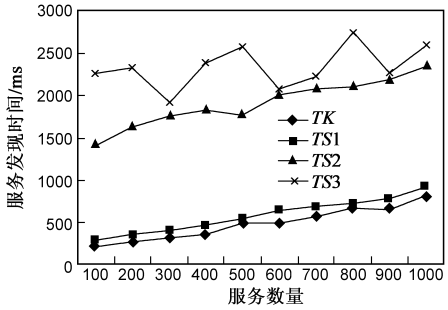

图4 不同的服务发现方法的可扩展性对比

实验结果表明,语义服务发现的方法响应时间随着服务注册中心数据库中服务数量的增加成线性增加,而且平均响应时间比传统的服务发现方法高,但在可接受的范围之内.当本体概念倒位索引被移除的情形下,服务发现子系统遍历所有服务并对每个服务进行匹配计算,从而导致服务发现响应时间大幅度增加.而如果系统不包含完整的本体概念近似度数据表时,服务发现系统响应时间超过 2000ms,并且波动很大.也就是说,当服务发现子系统本体概念倒位索引或本体概念近似度数据表不完整时,系统效率可能大幅度降低.

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[6,8]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,以及之后在文献[9]中又提出和描述了 UDDI/OWL-S Matchmaker,这些工作的部分成果已作为本文的研究基础.Klusch 等[10]的研究在文献[9]的基础上做出了新的拓展,即利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[11]介绍了使用 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比较的方法,这种方法能够根据本体概念的树结构识别一系列匹配度.Benatallah 等[12]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.基于 OWL-S 的其他服务匹配相关工作包括 Mandell[13]、Klein[14]等的研究.另外,web 服务发现和语义匹配的相关研究也包括其它技术和方法如 Web 服务建模本体(WSMO)[15]、义原相似度匹配[16]、P2P[17]以及服务过程模型[18]等.

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服务匹配要素.另外,文献[9,11,12]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.本文的研究以文献[10]的工作为基础,但在以下几个方面有所创新.首先,本文的方法使用更加细粒度的匹配算法产生匹配度数值;其次,本文提出的近似概念搜寻算法将搜寻范围限制在目标概念附近,以提高搜索效率.而且语义匹配系统使用概念近似度数据表和近似概念倒位索引,将有较大时间消耗的近似概念搜寻的过程从服务的发布和发现过程中分离.

6 结束语

本文提出了一个基于 OWL-S 并结合 UDDI 的 web 服务语义匹配系统,在保留 UDDI 标准接口的基础上将 OWL-S 匹配机制引入 web 服务的注册与发现过程.目的在于提高服务搜索的召回率和准确度从而提高服务发现的性能.论文展示了系统的整体架构和模块细节,描

述了语义匹配系统中服务发布和发现的过程,并分析了系统的性能和效率,其中对系统的服务发现过程使用的关键算法进行了详细介绍.

性能和效率是语义 web 服务发现的重要问题,与其相关的深入研究工作仍在进行中.本文的匹配算法使用 OWL 上下级关系规则用以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.

参考文献:

[1] 岳昆,王晓玲,周傲英.Web 服务核心支撑技术:研究综述[J].软件学报,2004,15(3):428 – 442.
Yue K, Wang XL, Zhou AY. Underlying techniques for Web services: A survey[J]. Journal of Software, 2004, 15(3): 428 – 442. (in Chinese)

[2] L Clement, A Hately, C von Riegen, T Rogers. UDDI Version 3.0.2 [EB/OL]. http://uddi.org/pubs/uddi-v3.0.2 – 20041019.htm, 2004.

[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] 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.

[5] M Paolucci, T Kawamura, T R Payne, K Sycara. Importing the Semantic Web in UDDI[A]. Web Services, E-Business, and the Semantic Web[C]. Berlin/Heidelburg: Springer, 2002. 815 – 821.

[6] Qiu T, Li L, Li P. Web service discovery with UDDI based on semantic similarity of service properties[A]. Proceedings of the Third International Conference on Semantics, Knowledge and Grid (SKG 2007)[C]. Xi'an, 2007. 454 – 457.

[7] 邱田,李鹏飞.一个基于概念语义近似度的 Web 服务匹配算法 [J].电子学报,2009,(2)37:429 – 432.
Qiu T, Li P. A Web service matching algorithm based on semantic similarity of concepts[J]. Acta Electronica Sinica, 2009, 37(2): 429 – 432. (in Chinese)

[8] M Paolucci, T Kawamura, T R Payne, K Sycara. Semantic matching of Web services capabilities[A]. The Semantic Web-ISWC 2002[C]. Berlin/Heidelberg: Springer, 2002. 333 – 347.

[9] N Srinivasan, M Paolucci, K Sycara. An efficient algorithm for OWL-S based semantic search in UDDI[A]. Semantic Web Services and Web Process Composition[C]. Berlin/Heidelberg: Springer, 2005. 96 – 110.

[10] M Klusch, B Fries, K Sycara. Automated semantic Web service eiscovery with OWLS-MX[A]. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems[C]. New York: ACM, 2006. 915 – 922.

[11] 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.

[12] B Benatallah, M-S Hacid, C Rey, F Toumani. Request rewriting-based Web service discovery [A]. The SemanticWeb-ISWC 2003 [C]. Berlin/Heidelberg: Springer, 2003. 242 – 257.

[13] D J Mandell, S A McIlllraith. A bottom-up approach to automating Web service discovery, customization, and csmantic translation[A]. Proceedings of the Twelfth International Conference on the World Wide Web (WWW 2003)[C]. Budapest: ACM, 2003.

[14] M Klein, B Koenig-Ries. Coupled signature and specification matching for automatic service binding[A]. Proceedings of European Conference on Web Services[C]. Erfurt: Springer, 2004. 183 – 197.

[15] U Keller, R Lara, H Lausen, A Polleres, D Fensel. Automatic location of services[A]. The Semantic Web: Research and Applications[C]. Berlin/Heidelberg: Springer, 2005.

[16] 吴健,吴朝晖,李莹,邓水光.基于本体论和词汇语义相似度的 Web 服务发现[J].计算机学报,2005,28(4):595 – 602.
Wu J, Wu C, Li Y, Deng S. Web service discovery based on ontology and Similarity of Words. Chinese Journal of Computer, 2005, 28(4): 595 – 602. (in Chinese)

[17] F Banaei-Kashani, C-C Chen, C Shahabi. WSPDS: Web service peer-to-peer discovery service[A]. Proceedings of International Symposium on Web Services and Applications (ISWS)[C]. Hiroshima, Japan, 2004. 733 – 743.

[18] A Bernstein, M Klein. Discovering services: Towards high-Precision service retrieval[A]. Web Services, E-Business and The Semantic Web[C]. Berlin/Heidelberg: Springer, 2002. 260 – 275.

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.E-mail:qiutian@ios.cn

胡晓惠　男,1960 年生,中国科学院软件研究所综合信息系统技术国家级重点实验室研究员,博士生导师.研究领域为信息系统集成与仿真技术. E-mail:hxh@ios.cn