

面向服务的语义知识发现研究

柳 巧 玲

(南京审计学院 南京 210029)

摘 要 企业在信息化过程中产生了大量的信息资源,为了将这些信息资源转变为企业的知识和财富,提出了构建面向服务的语义知识发现系统。论述了系统采用的技术提出了一种面向服务的语义知识发现模型,并对其实现的关键技术作了进一步的阐述。该模型为构建面向服务的语义知识发现系统以及解决系统构建中的关键问题提供了理论指导。

关键词 知识发现 本体 服务匹配 相似函数

中图分类号 TP311.13

文献标识码 A

文章编号 1002-1965(2011)01-0159-05

Research on Service-oriented Semantic Knowledge Discovery

LIU Qiaoling

(Nanjing Audit University, Nanjing 210029)

Abstract In the process of informatization, a lot of information is generated, in order to transform these information into knowledge for enterprise, the Service-oriented semantic knowledge discovery system is proposed in this paper. First, an overview of the SOA technology, then a Service-oriented semantic knowledge discovery model is put forward, and the key techniques for the system realization is elaborated. The proposed model provides theoretical guidance for building semantic knowledge discovery system and resolving the key problems involved.

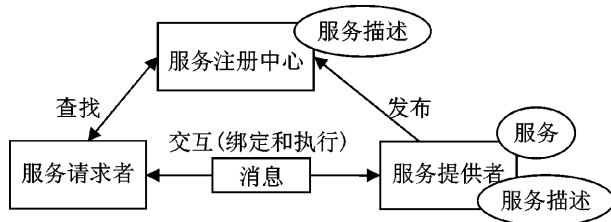
Key words knowledge discovery ontology service match similarity function

随着企业信息化的不断发展,企业中的信息资源也越来越丰富,而且具有结构复杂、动态变化等特点,出现了所谓的“信息过载”问题,如何利用新技术,高效、准确地将这些信息资源转变为企业的知识和财富,实现企业知识随处可得、随需可用,是企业信息化建设的一个核心问题和关键问题,然而,由于现有的企业资源的描述和表达是粗糙的,数据语义与内在逻辑没有明确表达或根本没有逻辑性,这就为进行有效的知识发现设置了巨大的障碍。而面向服务的架构(Services Oriented Architecture, SOA)和语义 Web 技术的出现为这一困境提供了新的解决思路。

1 SOA 技术框架

SOA 概念是 Gartner 公司在 1996 年首次提出,并将 SOA 定义为:“客户端/服务器的软件设计方法,一项应用由软件服务和软件服务使用者组成,SOA 与大多数通用的客户端/服务器模型的不同之处,在于它着

重强调软件组件的松散耦合,并使用独立的标准接口^[1]。”在基于 SOA 架构的系统中,具体应用程序的功能是由一些松散耦合且具有统一接口定义方式的服务组合构建起来的,接口是采用中立的方式进行定义的,它独立于实现服务的硬件平台、操作系统和编程语言,所有服务通过预先定义好的接口相互联系起来,根据需求对松散耦合的服务进行部署、组合和使用^[2]。SOA 采用如图 1 所示的注册、查找、绑定、执行的模式来完成对服务的访问与使用。

图 1 基本 SOA 模型^[3]

按照 SOA 基本模型设计的应用系统具有松散耦合性、平台无关性、位置透明性以及协议无关性等特

点,SOA 能够降低软件开发复杂度,便于应用系统集成,并提高系统的可复用性、灵活性以及可靠性等。因此,SOA 作为实现灵活的跨组织资源共享和应用集成的核心技术受到了广泛关注。当前广泛使用的 SOA 实现技术应该是 Web 服务,其服务注册中心以统一描述、发现和集成(Universal Description, Discovery, and Integration,UDDI)保存服务注册信息,使用通用的标准 Web 服务描述语言(Web Services Description Language,WSDL)对 Web 服务进行描述,并利用简单对象访问协议(Simple Object Access Protocol,SOAP)消息进行服务的绑定与调用。

现算法封装为算法服务,通过 UDDI 对外提供统一的访问接口,供算法控制子服务调用。知识发现服务包括目标控制子服务、数据控制子服务和算法控制子服务,其中目标控制子服务负责知识发现目标(如聚类分析、关联分析等)的设定和结果的管理;数据控制子服务负责与数据资源服务进行交互,完成对数据的选择以及数据的预处理工作;算法控制子服务负责与算法服务进行交互,完成对算法的选择和相关参数的设定。具体的语义知识发现过程如图 3 所示,用户通过语义知识发现服务接口发出一个知识发现请求任务,知识发现目标控制子服务负责提供常用的知识发现目

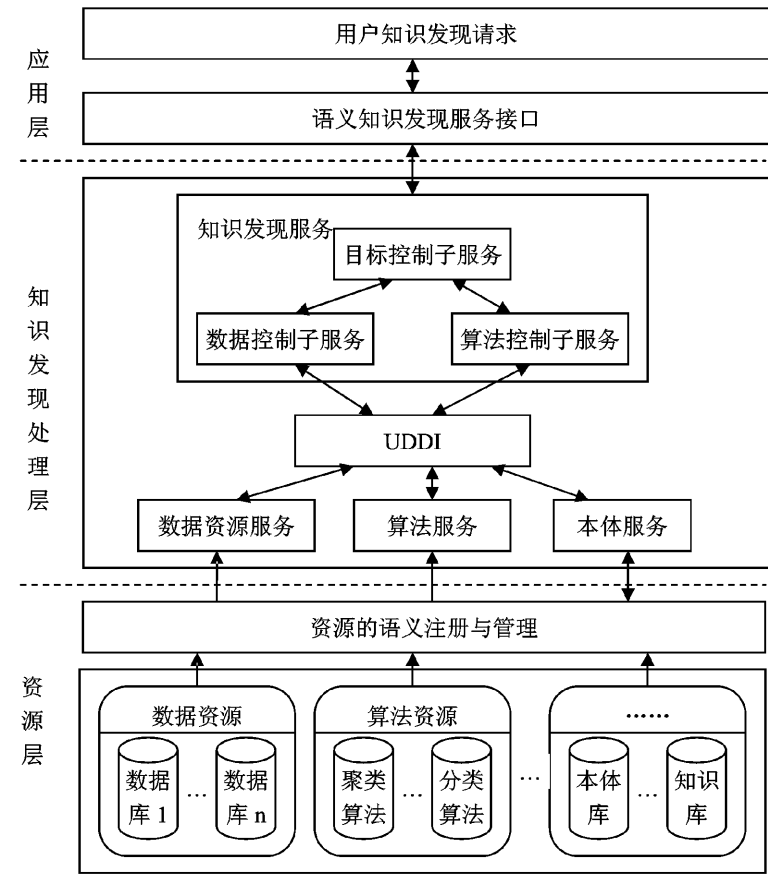


图 2 面向服务的语义知识发现模型结构图

2 面向服务的语义知识发现模型

2.1 面向服务的语义知识发现模型的构建 根据 SOA 技术框架,构建了如图 2 所示的语义知识发现的模型结构,在这个模型结构中,采用了分布式的组件架构和自适应的分布技术,在知识发现的整个过程中,用户不必关心资源的访问接口、存储管理方式和物理存储位置,在很大程度上简化和方便了用户的使用。从图 2 可以看出,分布式的、异质异构的、海量的企业信息资源可通过语义注册成为有语义支持的数据资源服务,并通过 UDDI 对外提供统一的访问接口,供数据控制子服务调用;同时,系统将算法提供者提供的知识发

标(如聚类分析、关联分析等)供知识发现用户选择,当知识发现用户选定目标类型后,知识发现目标控制子服务调用算法控制子服务,读取当前已注册的算法服务资源列表,并根据选定的目标类型自动推荐或人为选择合适的算法服务,知识发现目标控制子服务根据该算法服务的类型,让知识发现用户根据需求对算法的参数进行配置,并设定结果的满意阈值。同时,知识发现目标控制子服务调用数据控制子服务,读取当前已语义注册的数据服务资源列表,按用户需求选择数据资源,并对数据进行清洗、转换等预处理工作,数据资源和算法确定后,就可以执行知识发现任务,从而获取所需要的知识。

2.2 面向服务的语义知识发现的特点 面向服务的语义知识发现是建立在 SOA 等相关技术的基础上,在浩如烟海的企业信息资源中发现知识,在这一体系架构下,知识发现具有以下特点:

- a. 知识发现算法的动态加入与扩展。知识发现算法被封装成算法服务,通过服务注册、服务查询等机制可实现知识发现算法的动态发布/加入,以及算法的查找,实现了知识发现算法的动态性、开放性和可扩展性。
- b. 异质异构的企业信息资源的透明集成。企业中的信息资源往往具有不同的格式和访问接口,通过 SOA 和语义 Web 技术实现对这些分布式、异质异构的企业信息资源的透明集成和统一封装。用户不必关心这些企业信息资源的物理位置、具体的访问接口,就可以对他们进行访问,在很大程度上简化和方便了用户的使用。
- c. 知识的不断精化^[4]。面向服务的语义知识发现模型中运用了本体技术,本体的建立是一个不断完善

的过程,这一背景下语义集成的异质异构的企业信息资源,作为知识发现的源数据,其知识发现的输出本身也包含一定语义的知识。通过上一轮知识发现的结果,可以为本体的不断修正与完善提供参考,因此,下一轮知识发现,由于不同的企业信息资源是基于修正后的语义本体所集成的,知识发现的有效性相应地就会有一定程度的提高,从而实现知识的不断精化。

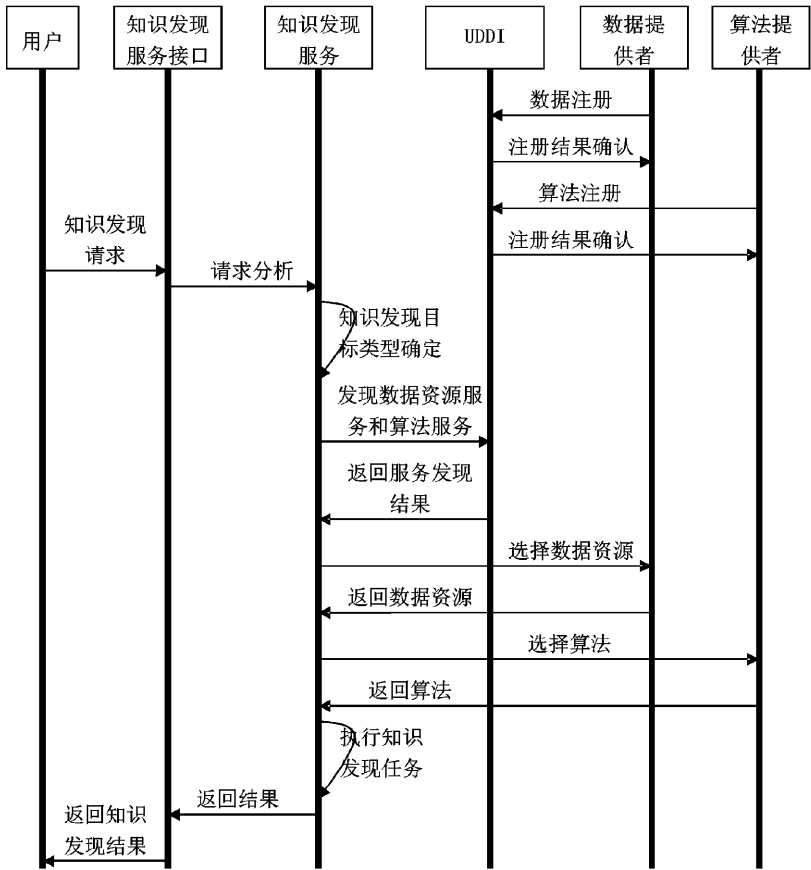


图 3 面向服务的语义知识发现的过程示意图

3 面向服务的语义知识发现模型实现的关键技术

3.1 语义服务建模 在面向服务的语义知识发现模型中,一切资源被封装为服务,并要求服务采用统一的语义描述机制,使不同的服务之间可以相互理解、相互协同。而传统的基于语法的 Web 服务主要是利用 WSDL 进行服务描述的,不能对服务进行语义描述,为了描述服务的功能和参数的语义信息,人们引入了本体^[5]。在描述语言上,采用本体来解决传统的基于语法的 Web 服务描述的异构性,增强对 Web 服务的功能、行为的语义描述。OWL-S^[6]是基于 OWL 语言的 Web 服务本体,可以将 Web 服务的属性和功能以精确和计算机可读的形式进行描述。

根据 OWL-S 的定义,面向服务的语义知识发现模型中的一个服务可由表示 (Presents)、被描述 (Describedby) 和支持 (Supports) 三个属性进行描述,每个属性对应着一个次高层的类,分别是服务简档 (Servi-

ceProfile)、服务模型 (ServiceModel) 和服务基点 (ServiceGrounding),这三个类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,如图 4 所示。因此,我们将面向服务的语义知识发现模型中的服务形式化定义为: $KDS = \{ ServiceProfile, ServiceModel, ServiceGrounding \}$ 。

服务简档 (ServiceProfile) 主要实现服务能力的描述,包括三类信息:服务提供者基本信息、功能描述信息和非功能属性信息,可将服务简档 (ServiceProfile) 定义为: $ServiceProfile = \{ BasInfo, FunInfo, QosInfo \}$,其中 BasInfo 是基本信息,即服务名称 (ServiceName) 和文本描述信息 (TextDescription); FunInfo 是服务功能描述信息,包括服务的输入参数 (Inputs)、服务的输出参数 (Outputs)、服务执行的前提条件 (Precondition) 和服务执行产生的预期效果 (Effects); QosInfo 是非功能属性描述,主要包括服务响应时间 (Time)、服务成本 (Cost)、服务可用性 (Availability) 和服务可靠性 (Reliability) 等非功能属性。

服务模型 (ServiceModel) 主要实现服务执行过程的描述,如服务的逻辑执行顺序等。

服务基点 (ServiceGrounding) 主要实现如何访问服务的描述,具体描述了服务的消息格式、通信协议以及服务访问方法等。

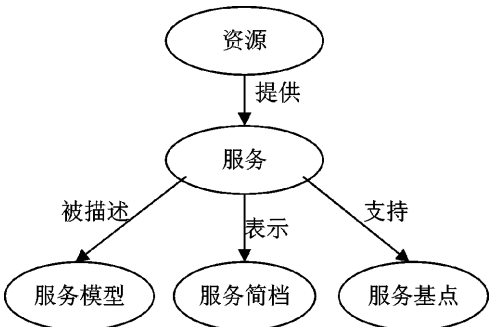


图 4 OWL-S 的顶层服务本体^[5]

3.2 基于语义的服务匹配

3.2.1 服务匹配模型的提出。服务匹配是面向服务的语义知识发现的关键技术之一,服务匹配的合理性和准确性直接影响着知识发现的质量。国内外学者和工业界对 Web 服务匹配方法进行了大量的研究,主要有两种,分别是基于语法的服务匹配和基于语义的服务匹配^[7]。如, Javier Gonzalez - Castillo 等利用 DAML+OIL 描述 Web 服务,扩展了服务匹配类型,但

没有定义输入/输出和服务质量等参数,服务匹配实用性差^[8]; Massimo Paolucci 等主要利用 DAML-S 中 Service Profile 对 Web 服务的描述,采用了服务基本信息和服务功能的匹配,但没有考虑服务质量的匹配,难以满足用户的非功能性要求,导致服务匹配有效性不足^[9];吴健等提出了基于本体论和词汇语义相似度的 Web 服务发现,该方法考虑了服务公共属性、服务专有属性、输入输出接口以及服务质量的匹配,但未能考虑服务执行的前提条件以及服务执行产生的预期效果等有关参数^[10]。

针对现有方法的不足,为了有效解决服务匹配中存在的问题,我们首先引入本体来充分表达语义信息,根据 $\text{ServiceProfile} = \{\text{BasInfo}, \text{FunInfo}, \text{QosInfo}\}$, 我们提出了“三层次”的语义匹配方法,在匹配过程中,采用了语义相似度,首先进行服务基本信息(如服务名称、文本描述等基本信息)的匹配,然后进行服务功能信息(如服务的输入参数、输出参数和服务执行的前提条件以及服务执行产生的预期效果)的匹配,最后进行服务质量信息(如服务响应时间、服务成本、服务可用性、服务可靠性等)的匹配,起到了层层过滤的作用,很大程度上缩短了服务匹配的时间,提高了服务匹配效率。

3.2.2 服务匹配的相似度函数的构造。在面向服务的语义知识发现模型中,设 RS 为用户请求的服务,PS 为 UDDI 中已发布的服务,在进行服务匹配之前,RS 和 PS 首先要进行语义标准化操作。给定服务请求者 RS,算法服务、数据资源服务等根据 UDDI 中的服务资源和服务请求者 RS 进行匹配,匹配的结果可以有多个,这时需要计算匹配结果与服务请求者 RS 之间的语义相似度,取语义相似度最大的服务作为最终匹配结果。根据提出的“三层次”服务匹配模型,构造服务请求者 RS 与已发布的服务 PS 的语义相似度函数为:

$$\text{Sim}(\text{RS}, \text{PS}) = \eta_1 \times \text{Sim}_{\text{bas}}(\text{RS}, \text{PS}) + \eta_2 \times \text{Sim}_{\text{fun}}(\text{RS}, \text{PS}) + \eta_3 \times \text{Sim}_{\text{qos}}(\text{RS}, \text{PS}), \sum_{i=1}^3 \eta_i = 1, 0 \leq \eta_i \leq 1, i = 1, 2, 3 \quad (1)$$

其中 $\text{Sim}_{\text{bas}}(\text{RS}, \text{PS})$, $\text{Sim}_{\text{fun}}(\text{RS}, \text{PS})$ 和 $\text{Sim}_{\text{qos}}(\text{RS}, \text{PS})$ 分别是基本信息匹配语义相似度、服务功能匹配语义相似度和服务质量匹配语义相似度; η_i 是语义相似度函数对应的权值,一般来说, $\eta_1 > \eta_2 > \eta_3 > 0$ 。

根据公式(1),首先计算服务基本信息匹配语义相似度,服务基本信息包括服务名称和服务文本描述等信息,则定义服务基本信息匹配语义相似度函数为:

$$\text{Sim}_{\text{bas}}(\text{RS}, \text{PS}) = \alpha_1 \times \text{Sim}_{\text{name}}(\text{RS}, \text{PS}) + \alpha_2 \times$$

$$\text{Sim}_{\text{text}}(\text{RS}, \text{PS}), \sum_{i=1}^2 \alpha_i = 1, 0 \leq \alpha_i \leq 1, i = 1, 2 \quad (2)$$

其中, $\text{Sim}_{\text{name}}(\text{RS}, \text{PS})$ 和 $\text{Sim}_{\text{text}}(\text{RS}, \text{PS})$ 分别是服务名称匹配的语义相似度和文本描述匹配的语义相似度, α_i 是语义相似度函数对应的权值,体现服务请求者对服务名称或文本描述的重视程度,权值的大小可以根据专家经验确定,但这种方法随意性强,不适应大规模的真实文本处理,可以采用基于统计的方法进行文本特征提取。

其次,计算服务功能匹配语义相似度,服务功能信息包括服务的输入参数集、输出参数集和服务执行的前提条件以及服务执行产生的预期效果,则定义服务功能匹配语义相似度函数为:

$$\text{Sim}_{\text{fun}}(\text{RS}, \text{PS}) = \beta_1 \times \text{Sim}_{\text{io}}(\text{RS}, \text{PS}) + \beta_2 \times \text{Sim}_{\text{pe}}(\text{RS}, \text{PS}), \sum_{i=1}^2 \beta_i = 1, 0 \leq \beta_i \leq 1, i = 1, 2 \quad (3)$$

其中 $\text{Sim}_{\text{io}}(\text{RS}, \text{PS})$ 和 $\text{Sim}_{\text{pe}}(\text{RS}, \text{PS})$ 分别是服务的输入参数/输出参数匹配的语义相似度和服务执行的前提条件/服务执行产生的预期效果匹配的语义相似度, β_i 是语义相似度函数对应的权值。一般来说,用户输入/输出的要求高于前提/效果,即一般 $\beta_1 > \beta_2$ 。

服务的输入参数/输出参数包括服务的输入参数集和服务输出参数集,则定义服务的输入参数/输出参数匹配的语义相似度函数为:

$$\text{Sim}_{\text{io}}(\text{RS}, \text{PS}) = u_i \times \text{Sim}_{\text{input}}(\text{InRS}, \text{InPS}) + u_2 \times \text{Sim}_{\text{output}}(\text{outRS}, \text{outPS}), \sum_{i=1}^2 u_i = 1, 0 \leq u_i \leq 1, i = 1, 2 \quad (4)$$

其中, $\text{Sim}_{\text{input}}(\text{InRS}, \text{InPS})$ 和 $\text{Sim}_{\text{output}}(\text{OutRS}, \text{OutPS})$ 分别表示服务输入参数集匹配的语义相似度和服务输出参数集匹配的语义相似度, InRS 和 InPS 分别表示服务请求 RS 描述的输入参数集和已发布的服务 PS 描述的输入参数集, OutRS 和 OutPS 分别表示服务请求 RS 描述的输出参数集和已发布的服务 PS 描述的输出参数集。 u_i 是语义相似度函数对应的权值,可由领域专家决定,如果对权值没有要求,则可采用默认的平均权值法,即这两个方面同等重要。

设 InRS 参数个数为 m , InPS 参数个数为 n , 若 $m \geq n$, 并且对于 InPS 中的任意一个参数 InPS_j , 在 InRS 的参数中都会存在 1 个或多个参数 InRS_i 与之概念语义相似度大于 0, 取其中的概念语义相似度最大值, 记 InRS_i 为 InRS 的第 i 个输入参数, InPS_j 为 InPS 的第 j 个输入参数, 且 $1 \leq i \leq m, 1 \leq j \leq n$, 则定义服务输入参数集匹配的语义相似度函数为:

$$\text{Sim}_{\text{input}}(\text{InRS}, \text{InPS}) =$$

$$\begin{cases} \frac{1}{n} \sum_{j=1}^n \max_{i=1, \dots, m} (\text{Sim}_c(\text{InPS}_j, \text{InRS}_i)) & m \geq n > 0 \\ 0 & \text{其他} \end{cases} \quad (5)$$

其中, $\text{Sim}_c(\text{InPS}_j, \text{InRS}_i)$ 表示 InPS 中的第 j 个输入参数 InPS_j 与 InRS_i 中的第 i 个输入参数 InRS_i 相匹配的语义相似度。

设 OutRS 参数个数为 p , OutPS 参数个数为 q , 若 $p \leq q$, 并且对于 OutRS 中任意一个参数 OutRS_i , 在 OutPS 参数中都会存在一个或多个参数 OutPS_j 与之概念语义相似度大于 0, 取其中的概念语义相似度最大值, 记 OutRS_i 为 OutRS 的第 i 个输出参数, OutPS_j 为 OutPS 的第 j 个输出参数, 且 $1 \leq i \leq p, 1 \leq j \leq q$, 则定义服务输出参数集匹配的语义相似度函数为:

$$\begin{cases} \frac{1}{p} \sum_{i=1}^p \max_{j=1, \dots, q} (\text{Sim}_c(\text{OutRS}_i, \text{OutPS}_j)) & 0 < p \leq q \\ 0 & \text{其他} \end{cases} \quad (6)$$

其中, $\text{Sim}_c(\text{OutRS}_i, \text{OutPS}_j)$ 表示 OutRS 中的第 i 个输出参数 OutRS_i 与 OutPS 中的第 j 个输出参数 OutPS_j 相匹配的语义相似度。

公式(5)和公式(6)说明, 如果已发布服务 PS 的服务能力满足请求服务 RS 的服务请求, 则 RS 的服务输入参数必须包含 PS 的服务输入参数, 同时, PS 的服务输出参数也必须包含 RS 的服务输出参数, 即请求服务 RS 要有能力供给已发布服务 PS 的输入, 使之能够满足运行服务所需要的所有输入参数, 而已发布服务 PS 要有能力满足请求服务 RS 的输出, 使之能够获得满意的服务结果。

$\text{Sim}_{\text{PE}}(\text{RS}, \text{PS})$ 的计算过程与 $\text{Sim}_{\text{IO}}(\text{RS}, \text{PS})$ 相似, 在此不再赘述。

最后, 计算服务质量匹配语义相似度。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程度无法判断出匹配到的服务在服务响应时间、服务成本等非功能方面是否能满足用户的需求。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既能满足功能方面的需求, 也能满足服务质量的需求, 还需要考虑服务质量的语义相似度。服务质量信息主要考虑服务成本、服务响应时间、服务可用性和服务可靠性(reliability)四个方面, 则定义服务质量匹配语义相似度函数为:

$$\begin{aligned} \text{Sim}_{\text{QOS}}(\text{RS}, \text{PS}) = & \omega_1 \text{Sim}_c(\text{RS}, \text{PS}) + \omega_2 \text{Sim}_l(\text{RS}, \\ & \text{PS}) + \omega_3 \text{Sim}_a(\text{RS}, \text{PS}) + \omega_4 \text{Sim}_r(\text{RS}, \text{PS}), \sum_{i=1}^4 \omega_i = 1, 0 \\ & \leq \omega_i \leq 1, i = 1, 2, 3, 4 \end{aligned} \quad (7)$$

其中, ω_i 是语义相似度函数对应的权值, 可由领域专家决定, 如果对权值没有要求, 则可采用默认的平均

均权值法, 即这四个方面同等重要。 $\text{Sim}_c(\text{RS}, \text{PS})$, $\text{Sim}_l(\text{RS}, \text{PS})$, $\text{Sim}_a(\text{RS}, \text{PS})$ 和 $\text{Sim}_r(\text{RS}, \text{PS})$ 分别表示服务请求与已发布服务相匹配的成本相似度、响应时间相似度、可用性相似度和可靠性相似度, 具体定义如下:

$$\text{Sim}_c(\text{RS}, \text{PS}) = \begin{cases} 0 & \text{RSc} < \text{PSc} \\ \frac{|\text{PSc} - \text{RSc}|}{\text{PSc}} & \text{RSc} \geq \text{PSc} \end{cases} \quad (8)$$

其中, RSc 和 PSc 分别是服务请求和已发布的服务所要求的成本。

$$\text{Sim}_l(\text{RS}, \text{PS}) = \begin{cases} 0 & \text{RSt} < \text{PSt} \\ \frac{|\text{PSt} - \text{RSt}|}{\text{PSt}} & \text{RSt} \geq \text{PSt} \end{cases} \quad (9)$$

其中, RSt 和 PSt 分别是服务请求和已发布的服务的响应时间。

$$\text{Sim}_a(\text{RS}, \text{PS}) = \begin{cases} 0 & \text{RSa} > \text{PSa} \\ \frac{|\text{PSa} - \text{RSa}|}{\text{PSa}} & \text{RSa} \leq \text{PSa} \end{cases} \quad (10)$$

其中, RSa 和 PSa 分别是服务请求和已发布的服务的可用性。

$$\text{Sim}_r(\text{RS}, \text{PS}) = \begin{cases} 0 & \text{RSr} > \text{PSr} \\ \frac{|\text{PSr} - \text{RSr}|}{\text{PSr}} & \text{RSr} \leq \text{PSr} \end{cases} \quad (11)$$

其中, RSr 和 PSr 分别是服务请求和已发布的服务的可靠性。

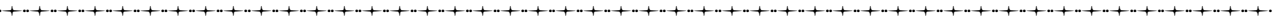
通过上述有关公式就可以计算出服务请求与已发布服务的语义相似度, 根据语义相似度从大到小排序, 返回前 N 个匹配服务作为候选服务供服务请求者调用。

4 结 语

知识经济时代, 如何利用新技术, 高效、准确地将企业在信息化过程中产生的大量信息资源转变为企业的知识和财富, 实现知识随处可得、随需可用, 是企业信息化建设的一个核心问题和关键问题, 为此, 本文给出了一种面向服务的语义知识发现系统框架, 为构建面向服务的语义知识发现系统以及解决系统构建中的关键问题提供了理论指导。

参 考 文 献

- [1] SOA, 引领软件发展新方向[EB/OL]. <http://www.e-works.net.cn/tbbd/soa/x1.htm>
- [2] 曹宝香, 刘 阳. 基于中间件的企业计算模型[J]. 计算机应用研究, 2007(2): 69-72
- [3] 喻 坚, 韩燕波. 面向服务的计算——原理和应用[M]. 北京: 清华大学出版社, 2006
- [4] 吴朝晖, 陈华钧. 语义网格: 模型、方法与应用[M]. 杭州: 浙江



(上接第 163 页)

大学出版社,2008

[5] 谢储晖. 空间知识网格基础与应用[M]. 徐州:中国矿业大学出版社,2005

[6] OWL-S 1.2[EB/OL]. <http://www.daml.org/services/owl-s>,2008-12

[7] 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

[8] JavierGonzalez-Castillo, David Trastour, and Claudio Bartolini. Description Logics for Matchmaking of Services[C]. In: Pro-

ceedings of the KI-2001 Workshop on Application of Description Logics, Vienna, Austria,2001:582-586

[9] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, et al. Importing the Semantic Web in UDDI[C]. In: Proceedings of Web Services, E-business and Semantic Web Workshop, Toronto, Canada, 2002:225-236

[10] 吴 健,吴朝晖,李 莹等. 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

(责编:刘影梅)