

# 服务关系及其在面向服务计算中的应用

陈世展 冯志勇 王 辉

(天津大学计算机科学与技术学院 天津 300072)

**摘 要** 在面向服务计算模式中,传统的 UDDI、ebXML 服务注册中心无论是服务发现,还是组合与交互方面都面临着信息共享和过程中介的难题.文中结合语义 Web(Semantic Web)和社会化网络(Social Networking)技术,显式地定义服务间的交互关系(简称服务关系),并将它们融入到服务注册中心的组织、构建及服务计算过程中,将可用 Web 服务组织成具有丰富语义信息、基于业务上下文和交互关系的服务生态系统.文章首先分析并定义了参数、操作和服务三个层级的服务关系;接着详细讨论了服务关系挖掘的技术细节,包括 Web 服务的语义标注、服务关系提取、利用等;最后是相关实验数据及结果分析.将语义推理和关系演算引入到服务发现、组合、交互过程中,可以促进和便利服务计算,也是朝向服务互联网(Internet of Services)迈出的重要一步.

**关键词** 服务网络;服务关系;服务挖掘;面向服务计算;Web 服务

**中图法分类号** TP311 **DOI 号**: 10.3724/SP.J.1016.2010.02068

## Service Relations and Its Application in Services-Oriented Computing

CHEN Shi-Zhan FENG Zhi-Yong WANG Hui

(School of Computer Science and Technology, Tianjin University, Tianjin 300072)

**Abstract** In Services-Oriented Computing (SOC) paradigm, traditional Web service registry, like UDDI and ebXML, has been facing great difficulty in information sharing and process mediation when processing service discovery, service combination and interaction. Combining Semantic Web and Social Networking technology, this paper organizes available Web services into service ecosystem, named Service Network (SN), which explicitly merges abundant semantic information and socialized interactive relations, i. e. Service Relation (SR), into SOC paradigm. Firstly, three different granularities of Web service, including parameter level, operation level and Web-service level, are analyzed and defined. Secondly, the details of Service Mining, which includes Web service annotation, service relations mining and its application and leverage on SOC, are discussed. Last comes the experimental results and analysis. By organizing available Web services into service discovery, composition, and interoperation with abundant semantic information, business context and social relations, SN is firmly believed to be a new infrastructure for SOC and also key effort toward the implementation of "Internet of services".

**Keywords** service network; service relation; service mining; services-oriented computing; Web service

## 1 引言

Web 服务 (Web Service)<sup>[1]</sup> 是松散耦合的、可复用的软件模块;建立在 WSDL (Web Services Description Language)<sup>[2]</sup>、SOAP (Simple Object Access Protocol)<sup>[3]</sup>、UDDI (Universal Description Discovery&Integration)<sup>[4]</sup>、XML (Extensible Markup Language)<sup>[5]</sup> 等技术基础上,使用“发现”机制定位服务、接口描述文档定义如何使用服务以及标准的传送格式进行通讯;其目的是为在 Internet 上不同操作系统、硬件平台和编程语言间集成应用软件提供支持,方便应用的实现和发布. 随着核心 Web 服务标准及补充规范被广泛采纳和实现,Web 服务已成为当前解决异构环境中应用集成、实现面向服务架构 (Service-Oriented Architecture, SOA) 的最佳实践<sup>[6]</sup>. 开发者可以在“描述、发现、交互”三原则基础上无缝地访问众多分布式服务,并按照最适合特定问题需求的方式链接在一起来解决复杂的问题,这促进了应用的快速开发、服务的重用和复杂服务的实现,成为当前分布式软件系统构建的主流技术.

面向服务计算 (Service-Oriented Computing, SOC)<sup>[7-9]</sup> 是针对分布式系统的新型计算模式,为解决资源异构问题以及构造动态、开放环境下松耦合的集成化应用带来了诸多便利,是一组典型体现 SOA 应用的概念、原理和方法的集合. 由 IEEE Computer Society (CS) 官方批准,并于 2004 年 11 月正式成立了服务计算技术委员会 (Technical Committee on Services Computing, TCSVC)<sup>①</sup>,将服务计算的内涵扩展为弥合业务服务和 IT 服务鸿沟的科学与技术的交叉学科,包括 Web 服务和 SOA、云计算、业务咨询方法和工具、业务流程建模、转化和集成. 服务计算的目标是使 IT 服务和计算技术更有效地支持业务服务,它涵盖了服务创新研究的整个生命周期,是面向服务计算的新发展.

当前, SOC 在互操作 (interoperability)、自动化 (automation) 和效率 (efficiency) 方面面临着重大挑战<sup>[10-13]</sup>. 主要表现在:互操作仍然是基于语法而不是语义的;服务组合更多的是一项静态事务而不是动态自主地构建灵活的业务流程;并且服务发现、组合等的效率也是个大问题,尤其是在较大规模的应用场景下.

没有人否认,目前导致 SOC 交互困境的主要原因在于缺乏计算机可理解的语义信息,这恰恰是语义 Web (Semantic Web) 和语义 Web 服务 (Semantic Web Service, SWS)<sup>[14]</sup> 的出发点. 但是,缺乏建立

在交互 (interaction) 和事务上下文 (transaction context) 基础上的服务间的关系信息才是更深层次上的原因. 本质上,通过 WSDL 文档,一个服务仅向外界通告了它自己能够做什么,但是它并不知道在 Internet 上 (或者服务注册库中) 是否存在其它服务? 也不知道那些服务是否会在功能与自己存在相似性而导致竞争或者互补性而存在协作的可能? 总之,服务间关系信息 (relational information) 的缺失,导致服务之间缺乏了解、互不感知,无形中助长了服务交互、组合的难度.

针对这一状况,我们将社会化网络 (Social Networking) 和语义 Web 技术融入到服务注册库中,提出并实现了一个名为服务网络 (Service Network, SN) 的面向服务计算的基础设施<sup>[15]</sup>,将语义推理和关系演算动态地引入到服务发现、组合、交互、管理过程中,促进和便利服务计算.

本文第 2 节综述服务计算相关研究的进展;接着在第 3~6 节详细讨论服务关系的定义、挖掘、利用、实验结果及分析. 其中,第 3 节从 Web 服务的 WSDL 模型出发,定义服务关系;第 4 节则详细讨论服务关系挖掘算法的设计和实现;第 5 节是服务关系的利用,即服务关系在服务发现、组合中的作用;第 6 节利用从 Internet 上随机下载的真实 Web 服务,对服务关系及其挖掘算法进行简单的验证和分析;第 7 节叙述服务间交互信息方面的工作;最后部分总结全文,并展望下一步的工作.

## 2 相关研究进展

在服务计算模式的愿景中,服务提供者 (service provider) 供应松耦合的服务;而服务请求者 (service requestor) 仅需要花费很少的代价就可以轻松灵活地将它们组装成动态业务流程.

如何充分地利用广泛分布于网络上的服务,将其组合起来协同工作完成更大粒度的任务,即服务发现、组合及复杂交互自然就成为 SOC 领域的研究热点<sup>[16-18]</sup>. 而当前的研究也主要聚焦在能够为联邦式服务发现、动态服务组合、服务监控和管理、服务安全与隐私、服务语义计算提供良好支持的服务建模、互操作性及其数学基础等几个主题上.

### 服务注册

Web 服务注册中心,也被称为服务库 (Web services registry) 或服务代理,是 SOA 架构的重要

① <http://tab.computer.org/tcsc/>

基石<sup>[19-20]</sup>. 典型的 Web 服务注册机制包括一组接口和一个知识库. 从体系结构方面看, Web 服务注册中心可分为集中式(Centralized)、分布式(Decentralized)和混合式(Hybrid)三类<sup>[20]</sup>.

服务注册中心的设计与服务发现机制紧密联系在一起, 互相影响, 图 1 扩展了文献[21]中的服务注册机制及其对应的服务发现技术. 从服务发现的视角看, 传统的服务发现大多是基于 Web 服务接口的结构化特征和关键词匹配的. 为了提高服务发现的精确性和自动化程度, 运用本体和推理技术、基于语义相似度的语义服务发现方法得到广泛的认可<sup>[22-30]</sup>.

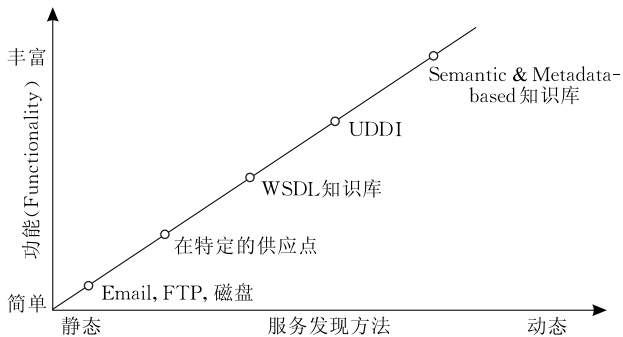


图 1 服务注册库类型

### 服务组合

面向服务计算的一个核心理念就是将服务作为基本的构建模块来创造可以跨越组织边界和计算平台的动态业务流程或敏捷应用<sup>[31]</sup>. 如何充分地利用广泛分布于网络上的服务, 将其组合起来协同工作完成更大粒度的任务, 即 Web 服务组合和复杂交互自然就成为 SOC 领域的研究热点.

从组合策略的角度, 服务组合可以分为两类. 一类是在设计时刻基于服务组合和编排规范, 如 WS-BPEL(Web Services Business Process Execution Language)<sup>①</sup>或 WS-CDL(Web Services Choreography Description Language)<sup>②</sup>, 静态地汇集多个服务构建业务流程. Microsoft Biztalk<sup>③</sup> 和 Oracle WebLogic<sup>④</sup> 是静态服务组合引擎的典型代表. 另一类则是在人工智能(AI)和语义 Web 技术基础上, 动态地组合可用服务满足用户需求. 在以计算机可理解的形式增加描述 Web 服务的属性和性能的语义信息方面, 人工智能和语义 Web 社区进行了大量的研究. 一些研究的目的是使用形式语言和本体来对服务的所有相关方面进行语义描述, 从而有助于通过 Web 进行的服务自动发现、汇聚和调用<sup>[32-33]</sup>. 另一些研究主要集中于利用 AI 规划技术来自动地发现和组合 Web 服务<sup>[32-36]</sup>、面向服务应用的验证<sup>[37-38]</sup>和运行时刻的监控<sup>[39]</sup>上.

为了增加基于服务应用的适应能力和灵活性, 一些研究将上下文和服务质量的概念引入到服务组合中, 形成了上下文感知(context-aware)<sup>[40]</sup>和服务质量感知(QoS-aware)<sup>[41]</sup>的组合方案, 在某种程度上, 也提高了组合过程的自动化程度.

### 语义扩展及服务本体(Service Ontology)

传统的 Web 服务是基于 XML 的, 在描述需求和 Web 服务的能力方面缺乏语义表达能力, 不利于计算机间的自动交互. 其扩展是将语义 Web(Semantic Web)技术(如 OWL-S<sup>[42]</sup> 和 METEOR-S<sup>[43]</sup> 等)应用到 Web 服务领域, 借助本体和语义标注技术扩展 Web 服务的语义描述, 通过良好定义和具有明确含义的共享本体, 使计算机能够读、发布、交换信息和知识.

在语义 Web 背景下的 Web 服务研究呈现多样化, 包括自动语义标注<sup>[44-49]</sup>、基于本体<sup>[50]</sup>和推理的自动服务组合<sup>[51]</sup>等; 另外一些研究工作侧重于对现有解决方案进行语义扩展, 如向 WSDL、BPEL 中添加语义, 改善服务的互操作. 通过良好定义和具有明确含义的本体扩展 Web 服务的语义描述, 使计算机能够读、发布、交换信息和知识. 然而, 要想实现人-计算机、计算机-计算机之间通过服务流畅地进行交互, 上述工作仍然在业务流程语义和服务的自治与能力两个方面面临挑战<sup>[52]</sup>.

Internet of Services (IoS)<sup>[53]</sup> 是今天互联网朝服务化方向的演化, 其核心就是通过 Internet 提供服务, 致力于消除当前情况下服务访问中的障碍和低效率, 促进用户/服务交互和跨组织的无缝协作. 在该领域, 最著名的研究项目就是最初由德国联邦经济和技术部发起的 THESEUS<sup>⑤</sup>, 其目标是开发一种新型的基于 Internet 的基础设施以便更好地利用 Internet 上的知识. 该项目同时关注技术和经济方面的问题. 另外还有一些研究项目和活动关注于该主题的特定方面, 如服务价值网(Service Value Networks)、利用 SaaS 模式来提供企业应用、IoS 中的信息组织等. 为应对 IoS 的挑战, Ferrario<sup>[54]</sup> 和 Oberle<sup>[55]</sup> 等人的工作正朝着建立公共的基础性服务本体迈出坚实的一步.

## 3 服务关系(Service Relation, SR)

服务网络(Service Network, SN)是关于 Web

① <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

② <http://www.w3.org/TR/ws-cdl-10/>

③ <http://www.microsoft.com/biztalk/en/us/default.aspx>

④ <http://www.oracle.com/appserver/weblogic/enterprise-edition.html>

⑤ <http://theseus-programm.de/en-us/home/default.aspx>

上可用服务的社会网络,核心是带有丰富语义信息的服务和服务间的交互关系(简称服务关系, Service Relation, SR). 其形式化的定义可表述如下:

$$SN(V, E) = Graph_{SN} \langle V_{SN}, SR_{SN} \rangle,$$

其中,

$$V_{SN} = \{ \text{Abstract Services} \} \cup \{ \text{Actual Web Services} \},$$

$$SR_{SN} = V_{SN} \times V_{SN}.$$

即以服务为结点,服务关系为边的社会图(Social Graph). 服务结点是基于 OWL DL<sup>①</sup> 语言构建、用于描述 Web 服务及其特征的良好定义的本体模型,分为抽象服务结点(Abstract Web Service)和具体服务结点(Actual Web Service). 服务间的社会化交互关系,简称服务关系,是从服务结点  $V_i$  到  $V_j$  的(有向)边,代表了服务之间操作上、结构上或语义上的约束及关联. 记作

$$SR_{Type} \langle V_i, V_j \rangle,$$

其中,  $V_i, V_j \in V_{SN}$ ,  $Type = \{ \text{exa, plu, sub, sim, seq1, seq2, pre1, pre2, Community, MemberOf, Nonfunctional} \}$ .

服务关系是良好定义的、建立在服务交互和事务上下文基础上的二元关系,描述了服务之间的功能性和非功能性约束;是关于 Web 服务间接触、联络、关联、群体依附和聚会等方面的关系数据,它们把一个服务与另一个服务联系在一起,形成基于服务关系的 Web 服务生态系统.

从 WSDL 文档的视角,Web 服务可被抽象为由服务(Service)、操作(Operation, 在 WSDL 新版中被称为 Interface)和参数(Parameter) 3 个层次的组件构成的概念模型,如图 2 所示. 自然地,服务关系的定义也被分为 3 种粒度:参数级关系(Parameter-level)、操作级关系(Operation-level)和服务级关系(Service-level).

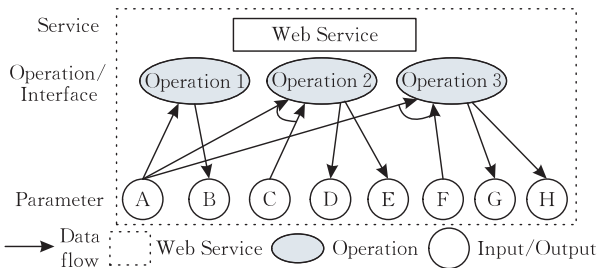


图 2 Web 服务模型

建立在本体概念基础上,参数级关系(如表 1 所示)仅定义了一种,即计算不同服务的输入或输出参数之间的等价(Equal)关系.

表 1 参数级服务关系

类型(categories)	描述(description)
Equal	两个参数在结构上或者语义上等价,因而可以相互替换

服务  $WS_i$  的操作可被抽象为  $OP_i \langle I_i, O_i \rangle$  或者  $OP_i \langle I_i, O_i, P_i, E_i \rangle$  参数对的形式,其中  $I_i, O_i, P_i, E_i$  依次为服务的输入(Input)、输出(Output)、前提条件(Precondition)和效果(Effect). 自然地,操作级关系即表示两个参数对之间的关系,具体定义了等价(Equal)、完全匹配(Full-matching)和部分匹配(Partial-matching) 3 种类型,如表 2 所示. 仅以完全匹配关系为例,如果操作  $OP_2$  的输入  $I_2$  包含于操作  $OP_1$  的输出  $O_1$ ,即操作  $OP_1$  的输出可以完全满足操作  $OP_2$  的输入,则表明操作  $OP_1$  与操作  $OP_2$  之间存在完全匹配类型的关系, $OP_1$  可以独自驱动或调用(invok)  $OP_2$ ,或者说  $OP_1$  是驱动  $OP_2$  的充分条件.

表 2 操作级服务关系

类型(categories)	描述(description)
Equal	操作 $OP_1 \langle I_1, O_1 \rangle$ 和操作 $OP_2 \langle I_2, O_2 \rangle$ 等价,因为 $I_1$ 与 $I_2$ , $O_1$ 与 $O_2$ 分别等价
Full-matching	$\exists O_1 \supseteq I_2 \Rightarrow OP_1 \rightarrow OP_2$
Partial-matching	$\exists O_1 \subset I_2 \Rightarrow OP_1 \cup \dots \rightarrow OP_2$

服务级关系侧重于描述服务间的交互和约束,是操作级关系在服务粒度层面上的体现. 在 Web 上,多个服务在功能上等价或相似,它们之间必然存在竞争:即在某些应用场景下,用户或者选择服务 A 或者选择服务 B,而没有必要同时“雇用”多个竞争性的服务(在某些特定场景下用户可能会同时调用竞争的多个关系!). 另一方面, SOC 模式的内在特性决定了服务之间存在交互和协作,这也是复合服务和业务流程中控制流的基础. 在 SN 中共定义了竞争关系(Competitive relations)、协作关系(Collaborative relations)和其它类型(Other relations) 3 大类 11 种类型的服务级关系,如表 3 所示. 以 Sequential-part 为例,服务  $WS_i$  中的操作  $OP_{im}$  产生的输出仅仅是  $WS_j$  中  $OP_{jn}$  的输入的真子集,单独的  $OP_{im}$  自然不可能驱动  $OP_{jn}$ ,  $OP_{im}$  仅可以看作是驱动  $OP_{jn}$  的必要条件.  $WS_i$  需要与其它服务,如  $WS_k$ ,或多个服务同步后才能共同驱动  $WS_j$  中的操作  $OP_{jn}$ ,即构成服务组合中的同步(synchronization)模式.

① [http://www.w3.org/TR/2004/REC-owl-guide-20040210/#OWL\\_DL\\_tag](http://www.w3.org/TR/2004/REC-owl-guide-20040210/#OWL_DL_tag)

表 3 服务级服务关系

类别 (Types)	类型 (Categories)	描述 (description)
竞争关系 (Competitive relations)	Exact $SR_{\text{exa}} \langle WS_i, WS_j \rangle$	$\forall OP_{ik} \in WS_i, \exists OP_{jm} \in WS_j \Rightarrow OP_{ik} = OP_{jm}$ $\forall OP_{jk} \in WS_j, \exists OP_{im} \in WS_i \Rightarrow OP_{jk} = OP_{im}$ } i. e. $WS_j = WS_i$
	Plugin $SR_{\text{plu}} \langle WS_i, WS_j \rangle$	$\forall OP_{ik} \in WS_i, \exists OP_{jm} \in WS_j \Rightarrow OP_{ik} = OP_{jm}$ i. e. $WS_i \subset WS_j$
	Subsume $SR_{\text{sub}} \langle WS_i, WS_j \rangle$	$SR_{\text{sub}} \langle WS_i, WS_j \rangle \Leftrightarrow SR_{\text{plu}} \langle WS_j, WS_i \rangle$ i. e. $WS_i \supset WS_j$
	Intersection $SR_{\text{sim}} \langle WS_i, WS_j \rangle$	两个服务在功能上存在相似性, 但又不属于上述 3 种情形.
协作关系 (Collaborative relations)	Sequential-total $SR_{\text{seq1}} \langle WS_i, WS_j \rangle$	$\exists OP_{im} \in WS_i, \exists OP_{jn} \in WS_j \Rightarrow OP_{im}^{\text{out}} \supseteq OP_{jn}^{\text{in}}$ i. e. $WS_i \xrightarrow{\text{invoke}} WS_j$
	Sequential-part $SR_{\text{seq2}} \langle WS_i, WS_j \rangle$	$\exists OP_{im} \in WS_i, \exists OP_{jn} \in WS_j \Rightarrow OP_{im}^{\text{out}} \subset OP_{jn}^{\text{in}}$ i. e. $\left. \begin{matrix} WS_i \\ WS_k \end{matrix} \right\} \xrightarrow{\text{synchronizing, invoke}} WS_j$
	Precedence-I $SR_{\text{pre1}} \langle WS_i, WS_j \rangle$	$SR_{\text{pre1}} \langle WS_i, WS_j \rangle = SR_{\text{seq1}} \langle WS_j, WS_i \rangle$
	Precedence-II $SR_{\text{pre2}} \langle WS_i, WS_j \rangle$	$SR_{\text{pre2}} \langle WS_i, WS_j \rangle = SR_{\text{seq2}} \langle WS_j, WS_i \rangle$
其它关系 (Other relations)	Community	对于任意两个抽象服务 $S_{01}$ 和 $S_{02}$ , $(S_{01}, S_{02}) \in R_0$ 当且仅当存在领域功能本体 $FO_i \in FOS$ , 使得 $S_{01}.FC \in FO_i$ 且 $S_{02}.FC \in FO_i$ . 领域功能本体 (简称功能本体) 是指在领域本体基础上定义的, 描述该领域内动作功能关系的本体.
	MemberOf	每一个具体服务都要隶属于某一个抽象服务.
	Nonfunctional relations	公司实体层次上的关系, 如公司 A 和公司 B 间的联盟关系 (alliance relation) 势必将会影响到服务的选择.

服务关系的 3 种粒度不是相互独立的, 而是一个有机整体: 参数关系是最细粒度的, 也是计算操作级和服务级关系的基础, 如图 3 所示 (见下页). 服务 Location Information 中的输出消息 Postcode 与服务 Postcode to Zipcode Converter 中的输入消息 Postcode 表达同一概念, 因而在这两个消息间存在 Equal 类型的参数级关系. 在操作层面, 操作 GetPostcode 的输出完全匹配操作 PostcodeToZipcode 的输入, 即操作 GetPostcode 可以独立驱动操作 PostcodeToZipcode. 最后, 可以说在服务 Location Information 和服务 Postcode to Zipcode Converter 存在顺序关系, 即在特定业务流程中, 服务 Postcode to Zipcode Converter 只有在服务 Location Information 执行成功后才可执行.

## 4 服务关系挖掘

将分散于 Web 各个离散站点上的服务组织成富有语义的、便于应用程序访问的 SN, 传统的服务注册过程被分解为扩展服务语义信息的服务标注和提取服务之间交互信息的关系挖掘两大阶段, 如图 4 所示. 首先是服务标注, 包括服务接口文档验证、解析、服务分类、基于领域本体的语义标注以及将服务加入/注册到知识库中; 其次是服务关系的挖掘以及服务网络分析等工作.

### 4.1 服务分类和标注 (Service Annotation)

利用本体 (ontology) 扩展、改善和描述 Web 服

务的语义信息, 是当前解决和弥补 SOC 中“语义鸿沟”问题的普遍做法.

#### 服务解析验证

服务接口文档的验证、解析是服务分类、标注的预处理, 其目的是对利用网络爬虫从 Internet 上抓取的 Web 服务 (WSDL 文档) 进行过滤, 将不符合 WSDL 文档规范的“无效”服务筛选掉.

#### 服务分类

服务分类的目的并不是对 Web 服务集合的一个划分, 而是帮助待标注的服务快速找到合适的领域本体. 服务分类的过程大致上分为 3 步:

- (1) 从 Web 服务的描述文件 (即 WSDL 文档) 中提取描述该服务的特征向量;
- (2) 在 WordNet<sup>①</sup> 语义词典和 JWordNetSim<sup>②</sup> 基础上, 计算这个特征向量与领域中心向量 (每一个领域本体被归结为一个中心向量) 的语义距离;
- (3) 根据计算的结果, 选择合适的领域作为分类.

WSDL 描述文档可看作是服务开发者对 Web 服务最原始的注解, 其中包括服务的解释性信息, 是服务分类的重要依据. 此外, WSDL 中各元素在文档中所占的比重是不一样的, 例如, 操作接口是服务的功能单元, 根据一般的命名习惯, 服务的操作名中往往包含了服务的功能和分类信息. 因此, 本文从 WSDL 文档出发、重点关注操作接口名称来提取

① <http://wordnet.princeton.edu/>

② [http://en.pudn.com/downloads124/sourcecode/windows/detail52\\_8580\\_en.html](http://en.pudn.com/downloads124/sourcecode/windows/detail52_8580_en.html)

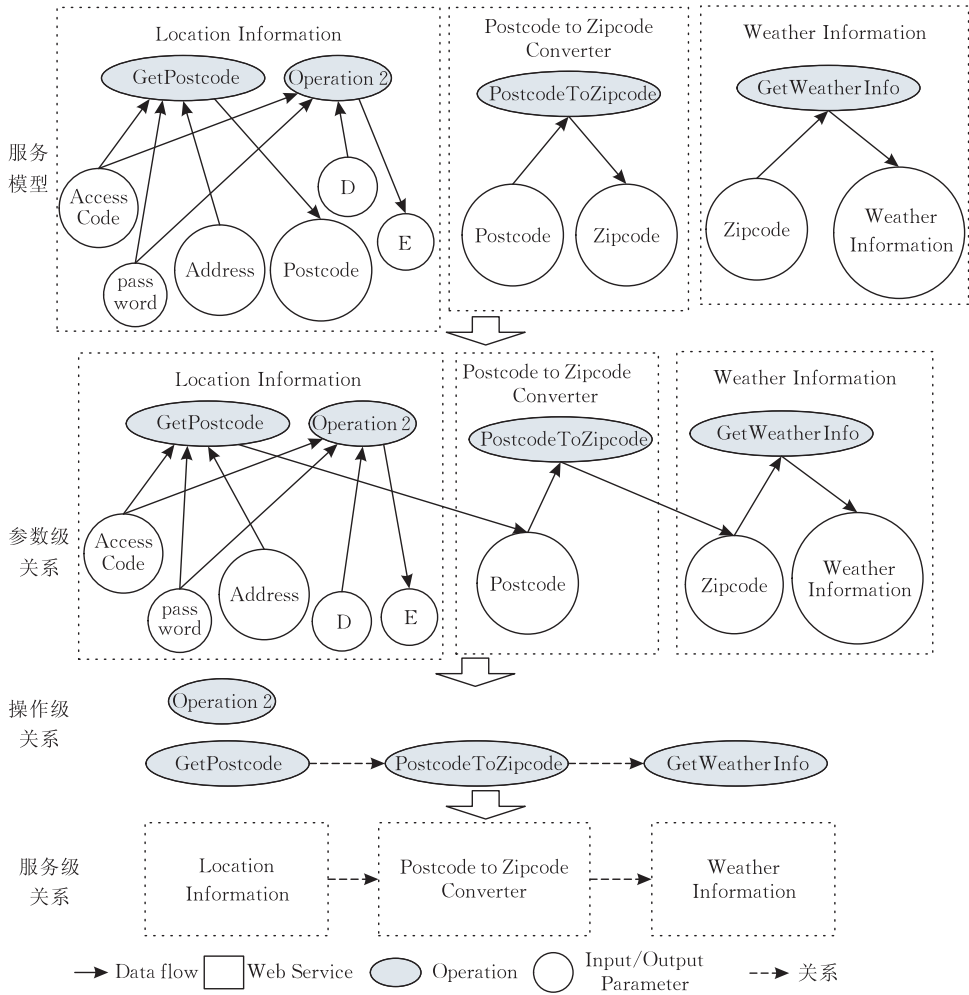


图 3 服务关系

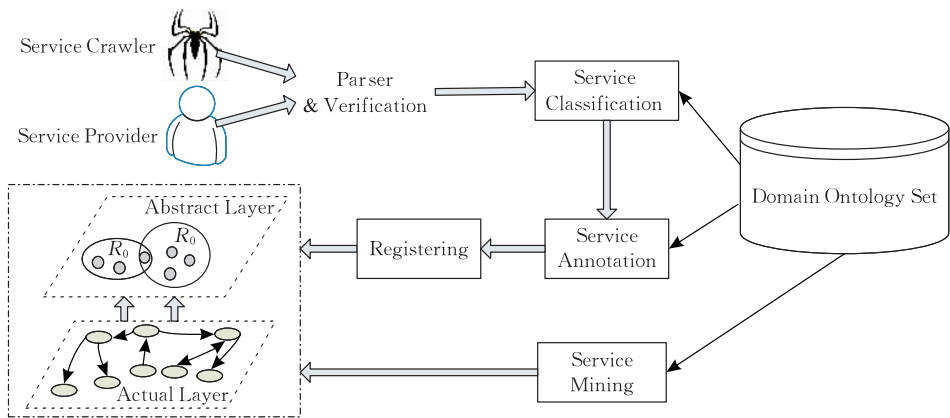


图 4 服务关系挖掘示意图

特征向量. 考虑到在大多情况下 Web 服务的操作名并不是规范的单词, 而是一些连写或缩写 (如 GetWeatherByZip), 本文借助文本处理的方法, 以大写字母和下划线来对这些短语进行分词、缩写词还原、去掉无意义词等处理. 通过设置停用词表 (Bad Word Table, BWT), 可以排除诸如 Get 和 By 这类对于服务分类没有帮助以及会同时出现在很多 WSDL 文件中 (如 information 等) 类别区分度很小

的单词干扰. 对于 WSDL 操作名中经常会出现的缩写词, 如 Geo<sup>①</sup> 等, 设置专门的缩写词表 (Abbreviation Word Table, AWT), 将其还原为原始的单词. 为了提高分类的准备性, 还需要根据单词所包括分类信息的多少, 应用 TF-IDF (Term Frequency-

① 一种在 HTML 和 XHTML 中标记 WGS84 地理坐标 (经度、纬度) 的徽格式

Inverse Document Frequency)算法对特征向量的单词赋与不同的权值. 对于每个单词,其权值为

$$W_i = tf_i \times idf_i,$$

其中,  $tf_i$  表示该词项在文档出现的频率,  $idf_i$  表示该词项的反比文档频率. 取  $idf_i = \log\left(1 + \frac{N}{n_i}\right)$ ;  $N$  是整个文档集的文档个数,  $n_i$  是包含该词项的文档个数.

$$SD_i = W_i \times \max(f_{SD}(word_i, CVword_j)),$$

其中  $W_i$  是  $word_i$  的单词权重,  $CVword_i$  是中心向量中的单词,  $f_{SD}$  是计算英文单词语义相似度的函数. 在此基础上, 向量距离可以表示为

$$VD = \sum_{i=1}^n SD_i,$$

其中,  $n$  是特征向量中单词的个数.

特征向量与中心向量的距离意味着此 WSDL

文档与领域描述的相似度. 对于特征向量中的每一个单词  $word_i$ , 分别计算其与中心向量元素的最大相似度:

考虑到本文所讨论的 Web 服务分类是为了找到合适的领域本体而对 Web 服务进行标注的, 在实践中, 根据计算的 VD 值:

- (1) 如果 VD 值大于 0.6 的领域超过 3 个, 则只选取前 3 位的领域进行标注;
- (2) 如果 VD 值都不大于 0.6, 则只选取一个最大值的领域进行标注;
- (3) 否则, 只选取 VD 大于 0.6 的领域进行标注.

### 服务标注

服务标注的整体框架如图 5 所示, 虚线框外左右两侧则是标注过程中的所用到的辅助数据库, 虚线框内的部分是概念匹配的基本流程, 分别针对简单概念和复合概念进行标注.

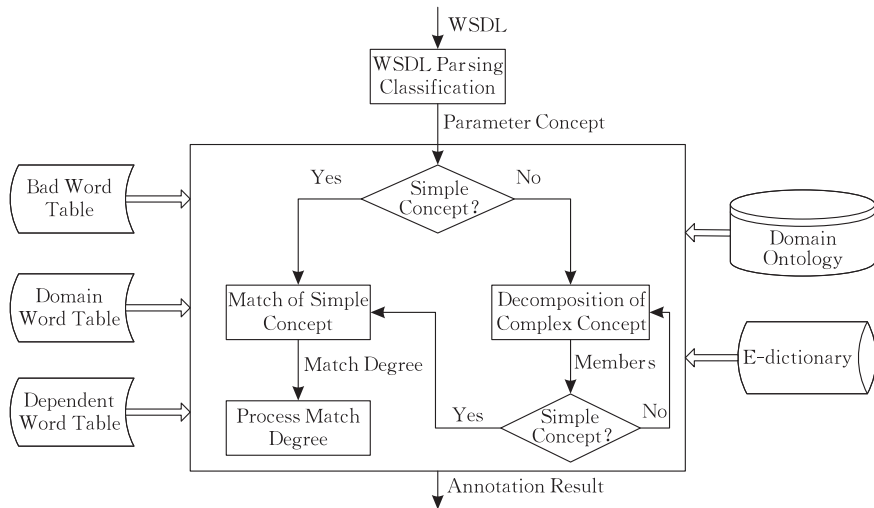


图 5 服务标注框架

考虑到有些单词(缩写)在不同的领域中意义完全不同, 在缩写词表基础上, 设置表示“领域”的字段来区分不同领域的缩写词, 将缩写词表进一步扩展为领域专有词表(Domain Word Table, DoWT). 此外, 对于有些单词, 如 name, 单独拿出来时没有太明显的表意, 只有和其它词放在一起时才有明确意思. 本文将这类词放到依赖词表中(Dependent Word Table, DeWT).

当两组词中有相同的依赖词时, 将这些词去掉之后再比较. 如“countryName”分别与“personName”和“countryCode”计算相似度, 如果直接比较, 分别得到相似度为

$$Sim(\text{“countryName”}, \text{“personName”}) = 0.63,$$

$$Sim(\text{“countryName”}, \text{“countryCode”}) = 0.56.$$

结果表明“countryName”与“personName”的相似度更大, 这显然是不合理的. 而事实上, “countryName”

与“personName”的相似度的真实相似度完全取决于 person 和 country 的相似度. 通过设置依赖词表, “countryName”与“personName”比较时, 将 name 去掉之后再比较, 此时得到的相似度是 0.26. “countryName”与“countryCode”的相似度值不变(country 不是依赖词), 依然是 0.56. 从而能够选择“countryCode”与“countryName”匹配, 这也符合人的判断.

算法 1 给出了 Web 服务自动语义标注的基本流程. 即, 对于每一个 WSDL 参数概念 PC(Parameter Concept), 从领域本体中搜寻与之最大匹配的本体概念(Ontology Concept) OCmax, 以此作为 PC 的标注概念.

### 算法 1. 服务标注流程.

L1 Function *annotateWebService*

L2 Input  $\forall pc_i \in WS$

```

//对于每个参数概念 PC,分别找出最大匹配的本体概念 OC
L3 Output OCmax
L4 Begin
L5 Float maxMD=0; //存放最大匹配度值
L6 OntologyConcept ocmax;
//存放最大匹配概念,即最终标注所用的概念
L7 for (OntologyConcept oci; OCSet){
//在概念集中寻找最合适的本体概念
L8 Float temp=getMatchDegree(pc,oci);
//获取概念的匹配度
L9 if (temp>maxMD){
L10 maxMD=temp;
L11 ocmax=oci;}}
L12 End

```

整个服务标注过程的核心在于如何求得概念匹配度(即算法 1 中 L8 行所示的函数),即计算 WSDL 参数概念与本体概念的匹配度,也可以表述为是本体概念对 WSDL 概念覆盖程度,如算法 2 所示.

#### 算法 2. 概念匹配算法.

```

L1 Function getMatchDegree
L2 Input ParameterConcept pc, OntologyConcept oc;
//输入为参数概念和本体概念实例
L3 Output MD //输出综合匹配度
L4 Begin
L5 Float MD, textSim, typeSim;
L6 if (isSimpleType(pc)){
//简单概念匹配;如果是简单概念则直接比较
L7 textSim=getTextSimDegree(pc.N,oc.N);
//文本匹配
L8 typeSim=getTypeSimDegree(pc.S,oc.S);
//结构匹配
L9 MD=getIntegratedMD(textSim,typeSim);
//得到综合匹配度}
L10 else {
//复合概念匹配;如果是复合概念,则本质上是一个
//递归过程
L11 textSim=getTextSimDegree(pc.N,oc.N);
L12 for (ParameterConcept sub_pc; pc.Elements){
//分别求得 PC 子概念与 OC 子概念的最大匹配
L13 Simi=
max(getMatchDegree(sub_pc,oc.Element));}
L14 TypeSim= $\frac{1}{N} \sum_{i=1}^N Sim_i$ ;
L15 MD=getIntegratedMD(textSim,typeSim);
//得到综合匹配度}
L16 End

```

匹配算法主要包括了两大部分:简单概念匹配(算法 2 中的 L6~L9 行)和复合概念匹配(算法 2 中的 L11~L15 行).

简单概念匹配分为两个层次:文本匹配(L7 行)和结构匹配(L8 行).简单概念的类型信息即为 string、double 等基本类型,这些与本体中的数据类型属性相同,因而在标注的时候可以直接比较.本文对于类型大小有如下约定:

```

boolean<int<float<double<string;
int<time<date<datetime<string.

```

对于 WSDL 中的枚举和数组两种约束,由于我们关注的是本体概念对 WSDL 概念的覆盖,因而仍可以将其还原为原始类型,如 ArrayOfString,虽然在 WSDL 标识为复合类型,但仍以 string 类型对待,为简单类型.而对于 string 上定义的枚举,也以 string 类型看待.复合概念的类型匹配实质上是一个递归的过程(L13~L16 行),将复合概念分解,依次去匹配其成员概念.最终根据根据 TextSim 和 TypeSim 的值得到综合匹配度:

$$MD = w_1 \times TextSim + w_2 \times TypeSim,$$

其中  $w_1, w_2$  为权值,且  $w_1 + w_2 = 1$ .

对于简单概念,  $w_1, w_2$  取值为

$$\begin{cases} w_1 = 0.8, w_2 = 0.2, & \text{当 } TextSim > 0.8 \text{ 时} \\ w_1 = 0.2, w_2 = 0.8, & \text{当 } TypeSim > 0.8 \text{ 时} \\ w_1 = 0.5, w_2 = 0.5, & \text{其它情况} \end{cases}$$

而对于复合概念,  $w_1, w_2$  取值为(其中  $N$  为成员概念的个数,当成员概念的个数多于 4 个时,  $w_1, w_2$  的取值只与  $N$  有关,即成员概念个数越多,结构信息所占比重就越大):

$$\begin{cases} w_1 = 0.8, w_2 = 0.2, & \text{当 } TextSim > 0.8, \\ & \text{且 } N \leq 4 \text{ 时} \\ w_1 = 0.2, w_2 = 0.8, & \text{当 } TypeSim > 0.8, \\ & \text{且 } N \leq 4 \text{ 时} \\ w_1 = \frac{1}{N+1}, w_2 = \frac{N}{N+1}, & \text{当 } N \geq 5 \text{ 时} \\ w_1 = 0.5, w_2 = 0.5, & \text{其它情况} \end{cases}$$

## 4.2 关系挖掘

经过自动标注,服务的元素与本体概念建立了对应关系并加入到 SN 中.但此时的 SN 是相当稀疏的,仅在抽象服务层(Abstract Service Layer)与具体服务层(Actual Service Layer)之间存在联系,而各层之内的关系则几乎没有.关系挖掘的本质可看作是服务注册库中的知识发现(Knowledge Discovery in Service Network, KDSN),即从大量的 Web 服务中识别和获取有意义的有关语义交互和逻辑约束的知识、“挖掘”服务之间的关联<sup>[56]</sup>,使原本松散的 SN 变得更加紧密.具体说来,服务关系挖掘程序应用本体理论对本体进行推理,找出隐含的概念之间的关系,要完成以下两个功能:

(1) 对于抽象参数(Abstract Parameter, AP)关系的挖掘:不同本体标注的抽象参数概念,通过挖掘程序将相同的概念合并,在合并的过程中,不同的领域本体(Domain Ontology)之间建立了联通桥梁,同时还可以减少服务网络的冗余。

(2) 对于抽象接口(Abstract Interface, AI)关系的挖掘:以领域本体为基础,借助本体推理等手段,挖掘抽象服务/接口之间的关联关系,同时将相同领域的抽象服务聚类,将描述同样功能的抽象服务合并,形成相互交叉的抽象服务聚类。

在服务网络中,抽象参数都已经和领域概念本体中的概念之间建立起了映射,可以认为,对抽象参数的操作事实上就是对本体的操作. 在同一个领域本体内部,以本体概念之间的关系为基础,结合词义相似度计算 Web 服务的参数概念之间的关系. 由于服务的参数可能是用不同的领域本体标注的,还需要跨领域的挖掘. 跨领域的概念关系挖掘可以看作是本体整合问题。

基于“如果两个概念的名字相似,则两个概念很可能相似”和“如果两个概念的属性相似,则两个概念很可能相似”的原则<sup>[57]</sup>,两个抽象参数之间相似度计算包括名字相似度和属性相似度两部分. 本文采用 HP 开发的 Jena<sup>①</sup> 包,所实现的概念关系挖掘过程如图 6 所示。

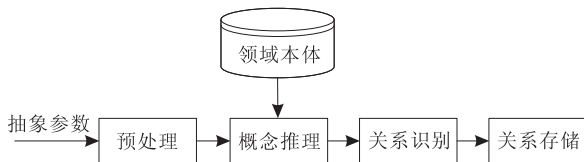


图 6 参数关系挖掘过程示意图

根据定义,抽象接口之间的关系是建立在输入输出参数的基础上的,因此,抽象接口之间的关系挖掘需要用到参数挖掘的结果,既要考虑到两个接口的输入参数(集合)之间的关系,又要考虑输出参数(集合)间的关系,如图 7 所示,最终完成服务关系的建立。

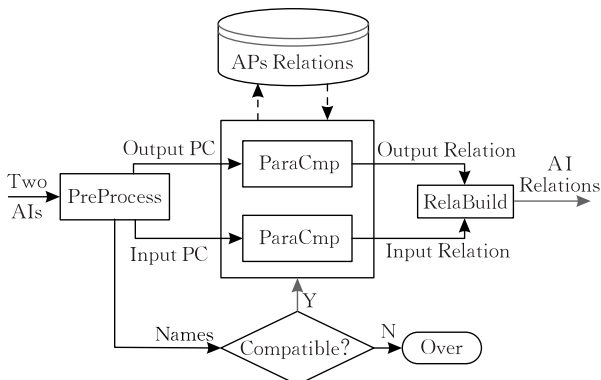


图 7 抽象接口关系挖掘过程示意图

由于服务网络中的 Web 服务或者是主动地从互联网上抓取,或者是被动地等待服务提供者注册,因而对服务网络中服务挖掘也可以认为是对互联网上服务资源的挖掘。

除了上述这种“在完全没有先验知识、从服务描述本身的特性出发,应用 Schema 匹配、本体推理等手段找出其隐含的服务关系,为服务计算提供参考”外,服务关系挖掘的另一个侧重点就是根据过往的使用记录,挖掘服务之间的关联关系,这一点与流程挖掘很类似。

#### 4.3 网络分析(Service Network Analysis)

社会网络(social network)起源于 Migram 的“六度分隔理论”<sup>[58]</sup>,是个体(人或者组织)间各种社会关系的数学模型. 最近几年,将社会网络模型融合 Internet 和 Web 的研究越来越受到重视,而社会网络也在语义 Web 的诸多领域(如知识管理(knowledge management)、信息检索(information retrieval)、普适计算(ubiquitous computing)等领域)突显出非常重要的作用. 对 Web 数据应用社会网络分析(Social Network Analysis, SNA),用于理解人们是怎样以计算机为媒介相互联系的,被认为是在 Web 上实现社会智能(social intelligence)的最具潜力的方法<sup>[59]</sup>. 还有一些研究利用搜索引擎从 Web 中挖掘社会网络<sup>[60]</sup>。

服务网络本质上是由 Web 上可用服务构成的社会化网络,包含描述服务自身能力的属性数据(如接口、参数等)和非独立的描述两个或多个服务之间多层次关系的关系数据. 借鉴 SNA 的方法和技术,对服务网络进行分析,试图对服务间各种关系进行量化分析,进而促进服务的共享、交互和合作,并改善 SN 系统的性能、可扩展性和可用性;并对设计基于关系的服务发现和组合算法提供指导。

## 5 服务关系利用

服务关系将一个服务与其它服务联系在一起,形成了基于关系的 Web 服务生态系统(如图 8 所示),从而具有一些结构化的特征. 服务链(service chain)是两个或两个以上的服务结点通过边顺次相连构成的一条路径,记作  $P = \langle W_0 R_0 W_1 R_1 \dots W_{m-1} R_{m-1} W_m \rangle$ ,表示了一个结构化的服务流(service flow),是构建复合服务(业务流程)的潜在候选者. 具有公共根结点的多个服务构成服务丛(service bundle). 服务丛是隶属于同一个抽象服务

① <http://jena.sourceforge.net/>

的 Web 服务集合,从结构上看,每个服务丛都是一棵以抽象服务结点为根、以成员(MemberOf)关系为边的生成树. SN 可以看作是一种特殊的森林:包含很多服务丛、不同服务丛上的结点通过关系铰链构成的多维网络. 服务群(service cluster)是 SN 中

的一个凝聚子集(cohesive subgroup),也可看作是 SN 的一个子图(sub-graph);其中含有服务链;也可能是复合服务及其成员;还可能由其它关系构成的小集团/派系(clique).

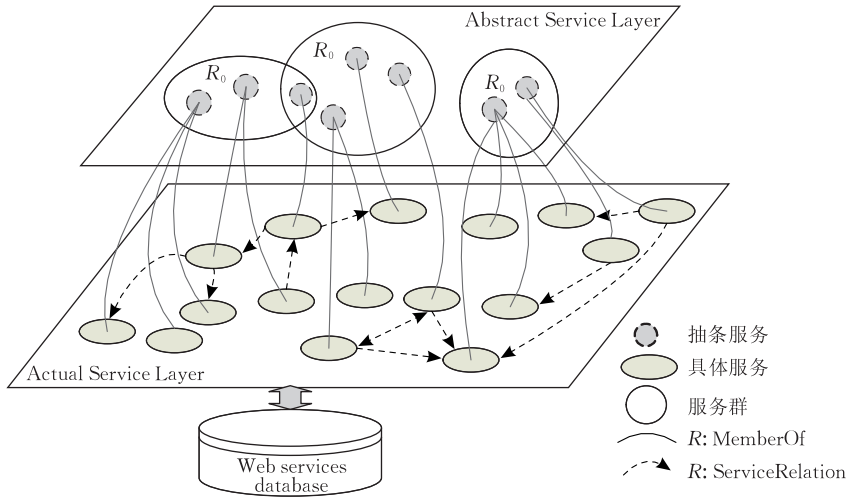


图 8 基于关系的服务生态视图

从社会网络的视角, SN 是服务为中心的个体网络(egocentric networks)所构成的集合. 每个服务都有自己的社交圈子,它知道:周围的服务哪些是自己的朋友和伙伴,可以相互协作完成更大粒度的业务功能;哪些又是竞争对手;相互间是怎样连接起来的;以及为了实现 SOC 愿景,相互间又能提供什么样的支持.

### 5.1 服务发现 (Web Service Discovery)

服务发现是服务计算模式中的核心任务,通过以下两种途径将服务关系引入到服务发现的过程中:(1)利用等价、相似等关系发现与目标服务“相关的”服务. 在保证查准率的前提下,可以提高查全率和查找的效率;(2)充分利用前驱、后继等关系,当在 SN 中查找不到能够满足用户请求的单一服务时,如果能够在 SN 中查找到可满足用户需求的服务链,那么也可以认为查找到了满足用户需求的服务. 即将服务发现由发现单个服务扩展到发现服务流(或业务流程). 该问题可以形式化如下:

目标:在 SN 中搜索可满足用户请求的服务,其中用户请求表示成  $Request = \{REQ^{in}, REQ^{out}\}$ ,其中  $REQ^{in}$  是用户的初始输入,  $REQ^{out}$  是用户期望的输出即请求结果.

状态:由于借助标注和本体技术,具体服务与抽象服务间建立起了对应关系,所以搜索空间就不再是具体服务构成的服务世界,而是 SN 中的抽象服务.

初始状态:从降低搜索空间的角度出发,指定能

够与用户输入  $REQ^{out}$  相匹配的抽象服务为初始状态,即  $AS^{out} \supseteq REQ^{out}$ .

后继函数:通过前驱后继关系能够到达的服务.

目标测试:检测抽象服务是否与用户输入  $REQ^{in}$  相匹配,即  $AS^{in} \subseteq REQ^{in}$ .

充分利用 SN 中的前驱和后继关系,采用双向搜索为主要策略的服务发现算法如算法 3 所示.

算法 3. 服务发现算法.

```

L1 Function ServiceDiscovery ( $REQ^{in}, REQ^{out}, limit$ )
returns a solution or False
L2 Input  $\{REQ^{in}, REQ^{out}\}$  //用户输入
      Limit //服务链长度限制,避免搜索代价过高
L3 Output 长度  $\geq 1$  的抽象服务链, or False
L4 Static FIFOQueue //初始化为空的先进先出队列
L5 SpanningTree //初始化为以  $REQ^{out}$  为 root 的树
L6 Begin
L7 ChildNodes  $\leftarrow$  GetAbstractServices( $REQ^{out}, SN$ );
L8 If Empty?(ChildNodes) then return False;
L9 FIFOQueue  $\leftarrow$  InsertQueue(ChildNodes,
      FIFOQueue);
L10 SpanningTree  $\leftarrow$  InsertTree(ChildNodes,  $REQ^{out}$ ,
      SpanningTree); //都加为  $REQ^{out}$  的孩子
L11 For each node in FIFOQueue do{
L12 If Empty?(FIFOQueue) then return False;
L13 Node  $\leftarrow$  RemoveFirst(FIFOQueue);
L14 If Depth(node, SpanningTree)  $>$  limit then continue;
      //限制返回服务链的长度不超过 limit
L15 If Test(node,  $REQ^{in}$ ) then return solution(node);

```

```
//solution 返回在SpanningTree上从node到root(REQout)
//但不包括 root 的路径
L16 ChildNodes ← Expand(node, 前驱关系, SN);
L17 FIFOQueue ← InsertQueue(ChildNodes,
                             FIFOQueue);
L18 SpanningTree ← InsertTree(ChildNodes, node,
                              SpanningTree);
L19 End
```

## 5.2 服务组合 (Web Service Composition)

上述面向业务流程的服务发现,已经有了简单的服务组合能力.然而,只有充分地利用服务关系以及关系之间的推理和演算,作为基于语义和社会化关系的服务计算基础设施,SN的威力才充分发挥出来.采用“以空间换时间”的策略,预先挖掘并存储服务关系,将服务链等结构化特征和关系推理引入到基于语义和 AI 推理的服务组合算法中,实现在功能层面上的服务组合.组合的结果即为一个抽象的工作流.只有基于服务的 QoS 等信息动态绑定后才成为可执行的业务流程.这不仅可以照顾到不同用户的个性化需求,也提高了业务流程的可靠性.

Web 服务组合问题可简单地形式化为:在 SN 中查找到多个服务  $\{AS_1, AS_2, AS_3, \dots, AS_n\}$ ,且满足

(1) 对于每一个  $AS_i (i \geq 2)$ ,其输入  $AS_i^{\text{in}}$  有

$$AS_i^{\text{in}} \subseteq REQ^{\text{in}} \cup AS_1^{\text{out}} \cup \dots \cup AS_{i-1}^{\text{out}}.$$

(2)  $REQ^{\text{out}} \subseteq AS_1^{\text{out}} \cup AS_2^{\text{out}} \cup \dots \cup AS_n^{\text{out}}.$

在 SN 基础上,我们已经提出并构建了一个二阶段服务组合的框架<sup>[61]</sup>,基本思想是首先利用服务关系和抽象接口基础上实现基于功能的服务组合模板,然后才是根据 QoS 信息的服务绑定构建具体的组合服务流程.在 SN 中,具有相同功能的多个 Web 服务隶属于同一个抽象接口,第 1 阶段的基于功能的服务组合使得组合算法的计算复杂度维持在一个相对较低的水平;在进入第 2 阶段后,候选服务集缩小,充分利用 QoS 等信息来达到更高的准确度和业务流程的灵活性.

## 6 实验及分析

由于缺乏公认的 Web 服务测试集,本文利用自定义的聚焦网络爬虫 (focused crawler) 从互联网上随机抓取了 326 个与旅游、天气相关的 Web 服务.测试服务集的选取基于如下原则:(1) 服务数量.从 Seekda!<sup>①</sup>网站可以看出,与旅游、天气相关的 Web 服务数量在 300~500 左右,相较于只有十几、几十个服务的领域,选取旅游、天气领域在服务数量上有一定的代表性,也更有说服力;(2) 对应的领域本

体.与旅游、天气相关的领域本体也相对比较成熟,在进行服务表祝贺关系挖掘时,可以尽可能降低领域本体对结果的影响;(3) 尽可能接近实际.采用“随机式”抓取与旅游、天气相关的 Web 服务,而不是人工有目的的筛选,是希望结果尽可能地更接近 Internet 上可用 Web 服务的真实情况.

经解析验证,从下载来的 326 个服务中筛选出 147 个无重复的服务<sup>②</sup>;利用旅游领域本体<sup>③</sup>对其中的 3692 个 WSDL 参数概念进行标注,标注结果如表 4 所示.

表 4 WSDL 标注结果

概念类型	准确率/%
复合概念	80.4
简单概念	89.1

从结果可以看出,对于随机选取的 Web 服务,本文的标注算法依然具有很高的标注正确率.在这里,简单概念的标注正确率要比复合概念更高.这是因为 WSDL 复合概念的定义随意性较大,很容易出现干扰项,且这些干扰项没有固定特征,比较难于用程序消除.而简单概念只要命名不是无意义的字符串,在 WordNet 和专有词表的帮忙下,基本都可以标注成功.与文献[62]相比,在不需要人工修正的帮助下,本文方法就可以接近其修正文件的正确率;而对于互联网上的随机服务,本文结果要远高于其独立文件的标注正确率.

经分析,造成标注错误的原因主要存在以下几个方面:

(1) 领域本体不够完善,使得有些 WSDL 概念确实无法找到与之匹配的领域概念.分标注失败比例占 31.1%.

(2) 当未知的缩写词出现时,会干扰标注结果.占标注失败 4.4%.

(3) WSDL 定义的随意性而导致的标注失败,占总标注失败的 55.6%.可见,由 WSDL 定义的随意性而导致的标注失败占了很大比例,这部分错误有些可以通过改进标注算法来提高.更主要的是需要规范化开发服务的接口文档.

(4) 其它原因占 8.9%.

随着本体库的完善和专有词表、停用词表和依赖词表的扩充,一、二类错误将越来越少.对于三、四类错误,可以通过引入文本处理技术、从上下文中获取信息、规范化服务接口描述等方式加以改善.此

① <http://webservices.seekda.com/>

② <http://ikse.tju.edu.cn/download/services.rar>

③ <http://ikse.tju.edu.cn/download/travel.owl>

外,有些标注失败是当前的标注算法所不能及的,如 GlobalWeather<sup>①</sup> 服务中的某些操作,其输入参数为字符串型的“地址编码”,输出为布尔值.对于这样的 Web 服务,仅靠计算机程序在概念参数层标注是不可能达到期望的目的的,需要进一步在操作功能层面进行标注.

以服务为单位统计标注结果如图 9 所示,其中有 67 个服务标注全部正确;在存在标注错误的 80 个服务中,其中 80% 的错误又集中出现在其中的 28 个服务.与文献[47]中标注错误基本上呈均匀分布相比,这 28 个服务中出现的错误数占了总标注错误的 80%,而只占被标注服务总数的 19.04%.基本上符合“二八现象”,即 80% 的错误出现在 20% 的服务中.而在出现错误的 28 个服务,参数概念标注错误的出现也不是均匀出现的(如图 10 所示).那么,未来的服务标注工作重点应该是在如何识别这 20% 的服务,并有针对性的设计标注算法,而不是泛泛的改进标注方法.利用本文定义的服务挖掘算法,对上述标

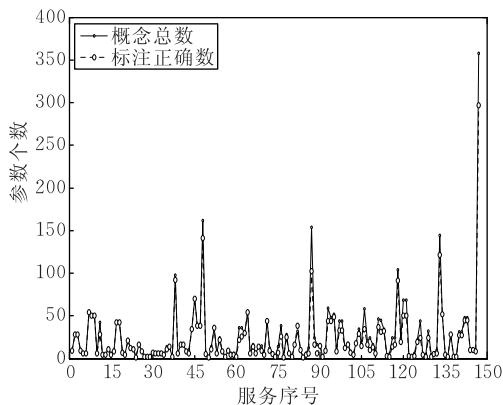


图 9 服务标注结果统计

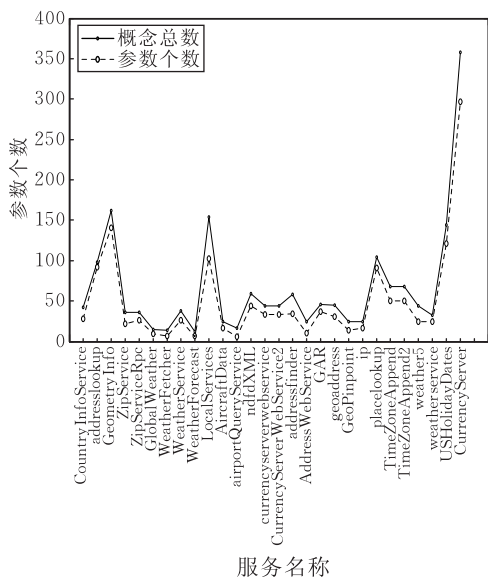


图 10 服务标注错误分析

注后的 147 个真实 Web 服务进行关系挖掘构建一个服务网络实例,并利用 Pajek<sup>②</sup> 软件进行分析.

本文所讨论的服务关系挖掘算法侧重于应用本体相关技术进行推理,找出隐含的概念之间的关系,其正确性在很大程度上依赖于服务标注的结果.另外,抽象接口的名字相容性判断也会影响关系挖掘的正确性.但在本文所选取的测试服务中,尚未发现因为名字而引起的错误.因此下述结果(服务之间关系等)基本反应了互联网上存在的旅游领域 Web 服务的关系状况,并在某种程度上反映了互联网上可用服务的关系状况.

(1) 147 个 Web 服务共包含 1566 个具体操作,生成的抽象接口(功能操作)376 个,将仅有输入而没有输出、仅有输出而没有输入(共 34 个)以及输入与输出等价(50 个)这 3 种特殊情况过滤掉,保留良好定义的抽象接口(功能操作)292 个,其中共涉及到旅游领域本体中的概念 103 个.可以看出,通过将能够完成同样功能的服务操作归纳为抽象接口(功能)后,搜索空间下降为原来的 20% 左右.即,对于给定的功能需求,在 Internet 上平均有 4~5 个候选服务.在服务发现时,通过抽象参数、抽象接口(功能操作)而不是直接通过服务的接口参数匹配,搜索的速度效率会有大的提高.

表 5 抽象接口层部分关系统计

Plugin/Subsume	TotalPrecedence	PartPrecedence
69	1439	549

注:根据服务关系的定义,竞争类中的等价关系体现为具体操作与抽象接口间的映射. Plugin 和 Subsume 是互逆关系,两者在数量上相等.在协作类关系中, Precedence 和 Sequence 也是互逆,只统计一种关系.

(2) 在抽象接口层挖掘得到部分服务关系统计结果如表 5 所示.平均而言,每个抽象接口的 Plugin/Subsume 度为 0.236, TotalPrecedence 度为 4.92, PartPrecedence 度为 1.88.抽象接口间也存在一定的可替换性;最重要的是,透过 TotalPrecedence 和 PartPrecedence 关系,在利用规划技术进行服务组合时,规划树的分枝数一般为 5~8.

(3) 对服务网络中结点的可达距离进行统计,测得两个可达结点之间最长的距离为 8,而平均距离为 3.2829.统计距离长度为 1、2、3、4 的结点对占可达节点总对数的 75.19%;如果将距离为 5 的结点对也考虑进去,则该数字上升为 93.94%.这说明,在旅游领域内,将近 94% 的服务组合可以在五步之内完成,或者说,一般情况下在旅游领域内的业

① <http://www.webservices.com/globalweather.asmx?WSDL>

② <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>

务流程其组件服务一般不超过 6 个。

(4) SN 的聚类系数 (clustering coefficient) 为 0.052861348, 远大于信息服务研究领域概念网络<sup>[63]</sup>聚类系数 0.0012496, 这表明服务网络的小集团性程度较高, 从而基于等价、相似等服务关系的服

务发现算法会具有较高的效率和查全率。

图 11 显示了用本文方法构建的服务网络中以 GetWeather 为中心的一个小片段, 其中文字表示服务, 连线表示服务之间的关系, 详细内容可参见 <http://ikse.tju.edu.cn/sn>.

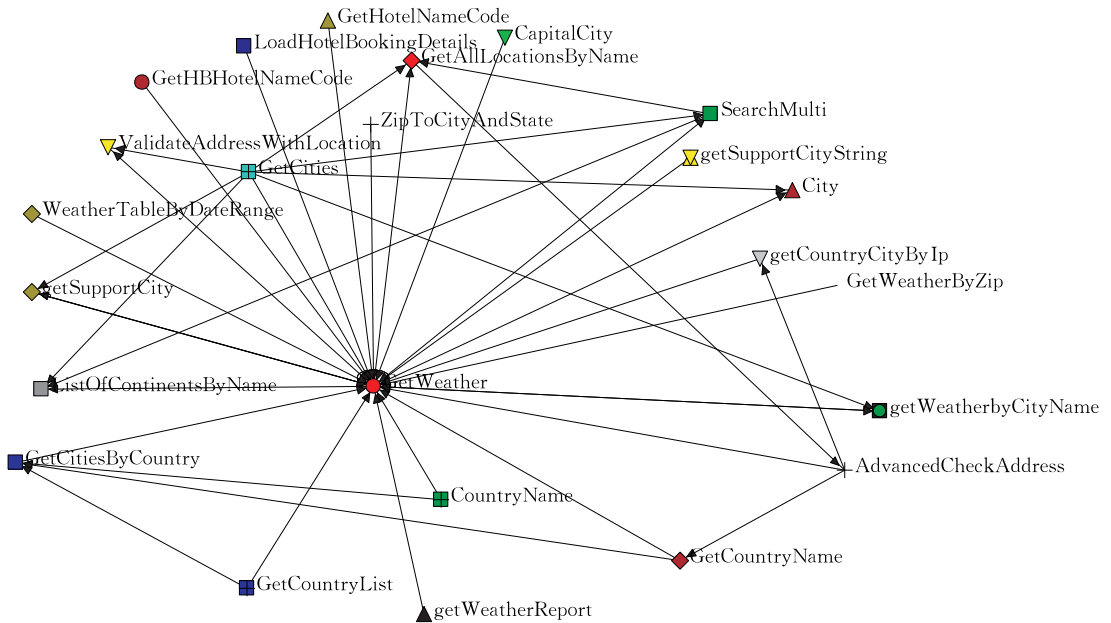


图 11 服务网络片段

## 7 总 结

自底向上的服务挖掘为服务组合提供强力支持. 从已有的 Web 服务出发, 挖掘其隐含的逻辑或业务关系, 为服务发现和复杂的服务组合提供更大粒度的素材, 降低服务组合的难度. Gombotz 等人<sup>[64]</sup>提出 Web 服务交互挖掘架构 (Web Services Interaction Mining Architecture, WSIMA) 来分析服务使用者和服务提供者之间的交互. Liang 等人<sup>[65]</sup>通过对服务使用记录的分析来挖掘服务的行为模式. 受药物发现过程启发, 文献<sup>[66]</sup>提出了一个自底向上的网络服务挖掘框架, 在此框架下, 非预期的、用户感兴趣的服务可以自动组合.

xCRL (eXtensible Customer Relationships Language)<sup>[67]</sup>是 OASIS (Organization for the Advancement of Structured Information Standards) 制定的、侧重于从单一客户/伙伴的视角实现不同系统、过程和平台间互操作的关系语言, 识别三类客户关系: 组织-组织 (organization-to-organization)、组织-个人 (organization-to-person) 和个人-个人 (person-to-person). WSRL (Web Services Relationship Lan-

guage)<sup>[68]</sup>则是基于 XML 语言的业务关系描述语言, 试图捕获业务实体 (business entity)、Web 服务 (Web service) 和操作 (operation) 3 种不同粒度的关系, 将这些关系看作选择和组合满足客户需求的正确服务集的重要辅助信息. 然而, xCRL 侧重于描述服务提供者间的关系, 没有考虑如何描述服务注册中心 (如 UDDI) 中的服务之间的关系. WSRL 可以通过链接的形式嵌入到 WSDL 文档中, 但仅侧重于服务间的静态关联信息, 对动态的关系和 QoS 信息等关注也不够, 而这些信息恰恰对服务的发现、动态组合和管理有着重要的影响.

本文结合社会网络和语义 Web 技术, 定义并挖掘 Web 服务间基于语义和交互的关系, 将可用 Web 服务组织成具有丰富语义信息、业务上下文和交互关系的服务生态系统; 将语义推理和关系演算引入到服务的发现、组合等过程中, 是朝向服务互联网 (Internet of Services) 迈出的重要一步.

下一步的工作将集中在以下两个方面: (1) 对大数据量下的 SN 结构分析, 以期进一步洞察服务间的社会化、结构化特征, 进而指导设计实现更有针对性的、高效的服务发现及组合算法; (2) 根据现有的实验结果, 有针对性的改进服务标注和服务挖掘算法.

致谢 感谢审稿人提出的评审意见,这些评审意见对提高论文水平有很大帮助!

### 参 考 文 献

- [1] Ferris C, Farrell J. What are Web services? Communications of the ACM, 2003, 46(6): 31-31
- [2] W3C. Web Services Description Language (WSDL) 1.1, 2001. <http://www.w3.org/TR/wsdl>
- [3] W3C. Simple Object Access Protocol (SOAP) 1.2, 2003. <http://www.w3.org/TR/SOAP/>
- [4] OASIS. UDDI specification Version 3.0.2. 2004. <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>
- [5] W3C. Extensible Markup Language (XML) 1.0, 2008. <http://www.w3.org/TR/REC-xml/>
- [6] Newcomer E, Lomow G. Understanding SOA with Web Services (Independent Technology Guides). Reading, Massachusetts, USA: Addison-Wesley Professional, 2004
- [7] Papazoglou P M. Service-oriented computing: Concepts, characteristics and directions//Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISE 2003). Roma, Italy, 2003: 3-12
- [8] Huhns M N, Singh M P. Service-oriented computing: Key concepts and principles. IEEE Internet Computing, 2005, 9(1): 75-81
- [9] Papazoglou M P, Traverso P, Dustdar S et al. Service-oriented computing: State of the art and research challenges. Computer, 2007, 40(11): 38-45
- [10] Benatallah B, Dumas M, Sheng Q Z. Facilitating the rapid development and scalable orchestration of composite Web services. Distributed and Parallel Databases, 2005, 17(1): 5-37
- [11] Zhang L J. Special issue: Web services discovery and composition—Editorial preface. International Journal of Web Services Research, 2007, 4(1): I-I
- [12] Margaria T. The semantic Web services challenge: Tackling complexity at the orchestration level//Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2008). Belfast, Northern Ireland, 2008: 183-189
- [13] Petrie C, Margaria T, Lausen H. Semantic Web Services Challenge: Results from the First Year (Semantic Web and Beyond). Berlin Heidelberg: Springer, 2008: V-VI
- [14] McIlraith S A, Son T C, Zeng H. Semantic Web services. IEEE Intelligent Systems, 2001, 16(2): 46-53
- [15] Chen Shi-Zhan. Service network: An infrastructure based on semantics and social relations for Web services computing [Ph. D. dissertation]. Tianjin University, Tianjin, 2009 (in Chinese)  
(陈世展. 服务网络: 基于语义和社会化关系的 Web 服务计算基础设施[博士学位论文]. 天津: 天津大学, 2009)
- [16] Srivastava B, Koehler J. Web service composition-current solutions and open problems//Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS'2003). Trento, Italy, 2003: 28-35
- [17] Milanovic N, Malek M. Current solutions for Web service composition. IEEE Internet Computing, 2004, 8(6): 51-59
- [18] Dustdar S, Schreiner W. A survey on Web services composition. International Journal of Web and Grid Services, 2005, 1(1): 1-30
- [19] Tsai W T, Zhou X Y, Chen Y N et al. Roadmap to a full service broker in service-oriented architecture//Proceedings of the IEEE International Conference on e-Business Engineering. Hong Kong, China, 2007: 657-660
- [20] Dustdar S, Treiber M. A view based analysis on Web service registries. Distributed and Parallel Databases, 2005, 18(2): 147-171
- [21] Graham S, Daniels G, Davis D et al. Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI. 2nd Edition. Upper Saddle River, NJ, USA: Pearson Education, 2004
- [22] Cardoso J, Sheth A. Semantic e-workflow composition. Journal of Intelligent Information Systems, 2003, 21(3): 191-225
- [23] Klein M, Bernstein A. Searching for services on the semantic Web using process ontologies//Proceedings of the International Semantic Web Working Symposium (SWWS). 2001: 159-172
- [24] Wu J, Wu Z H. Similarity-based Web service matchmaking//Proceedings of the 2005 IEEE International Conference on Services Computing. Orlando, Florida, USA, 2005, 1: 287-294
- [25] Hau J, Lee W, Darlington J. A semantic similarity measure for semantic Web services//Web Service Semantics Workshop at the 14th International World Wide Web Conference (WWW2005). Chiba, Japan, 2005
- [26] Verma K, Sivashanmugam K, Sheth A et al. METEOR-S WSDI: A scalable P2P infrastructure of registries for semantic publication and discovery of Web services. Information Technology and Management, 2005, 6(1): 17-39
- [27] Pathak J, Koul N, Caragea D et al. A framework for semantic Web services discovery//Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management. Bremen, Germany, 2005: 45-50
- [28] Fox J, Borenstein J. Semantic Discovery for Web Services. 2003. <http://soa.sys-con.com/node/39718>
- [29] Zhou B, Huang T. Semantic WEB service discovery search with ontology learning//Proceedings of the 2008 International Conference on Computer Science and Software Engineering. Wuhan, China, 2008: 1048-1051
- [30] Michael K, Ruben L, Axel P et al. A logical framework for Web service discovery//Proceedings of the ISWC 2004 Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications. 2004: 119
- [31] Papazoglou M P, Traverso P, Dustdar S et al. Service-oriented computing: A research roadmap. International Journal of Cooperative Information Systems, 2008, 17(2): 223-255
- [32] McIlraith S, Son T C. Adapting Golog for composition of semantic Web services//Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002). 2002: 482-493

- [33] Medjahed B, Bouguettaya A, and Elmagarmid A K. Composing Web services on the Semantic Web. *The VLDB Journal*, 2003, 12(4):333-351
- [34] Rao J H, Dimitrov D, Hofmann P et al. A mixed initiative approach to semantic Web service discovery and composition; SAP' s guided procedures framework//Proceedings of the IEEE International Conference on Web Services (ICWS 2006). Chicago, IL, 2006: 401-408
- [35] Lazovik A, Aiello M, Papazoglou M. Planning and monitoring the execution of Web service requests. *International Journal on Digital Libraries*, 2006, 6(3): 235-246
- [36] Wu D, Sirin E, Hendler J et al. Automatic Web services composition using SHOP2//Workshop on Planning for Web Services at the 13th International Conference on Automated Planning & Scheduling (ICAPS'2003). Trento, Italy, 2003
- [37] Narayanan S, McIlraith S A. Simulation, verification and automated composition of Web service//Proceedings of the 11th International World Wide Web Conference (WWW' 2002). Honolulu, Hawaii, USA, 2002: 77-88
- [38] Kazhamiakin R, Pistore M, Santuari L. Analysis of communication models in Web service compositions//Proceedings of the 15th International Conference on World Wide Web. Edinburgh, Scotland, 2006: 267-276
- [39] Barbon F, Traverso P, Pistore M et al. Run-time monitoring of instances and classes of Web service compositions//Proceedings of the IEEE International Conference on Web Services (ICWS 2006). Chicago, IL, 2006: 63-71
- [40] Colombo E, Mylopoulos J, Spoletini P. Modeling and analyzing context-Aware composition of services//Proceedings of the 3rd International Conference on Service Oriented Computing (ICSOC 2005). Amsterdam, 2005: 198-213
- [41] Wang G, Chen A, Wang C et al. Integrated quality of service (QoS) management in service-oriented enterprise architectures//Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference. Monterey, California, 2004: 21-32
- [42] Martin D, Burstein M, Hobbs J et al. OWL-S: Semantic Markup for Web Services. 2004. <http://www.w3.org/Submission/OWL-S/>
- [43] Patil A A, Oundhakar S A, Sheth A P et al. Meteor-S Web service annotation framework//Proceedings of the 13th International Conference on World Wide Web. New York, NY, USA, 2004: 553-562
- [44] Burstein M. Ontology mapping for dynamic service invocation on the Semantic Web//Proceedings of the AAAI Spring Symposium on Semantic Web Services. Palo Alto, 2004
- [45] Heß A, Johnston E, Kushmerick N. ASSAM: A tool for semi-automatically annotating semantic Web services//Proceedings of the 3rd International Semantic Web Conference (ISWC2004). Hiroshima, Japan, 2004: 320-334
- [46] Heß A, Kushmerick N. Learning to attach semantic metadata to Web services//Proceedings of the 2nd International Semantic Web Conference (ISWC2003). Sanibel Island, Florida, USA, 2003: 258-273
- [47] Patil A A, Oundhakar S A, Sheth A P et al. METEOR-S Web service annotation framework//Proceedings of the 13th International Conference on World Wide Web (WWW' 2004). New York, NY, USA, 2004: 553-562
- [48] Oldham N, Thomas C, Sheth A et al. METEOR-S Web service annotation framework with machine learning classification//Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition. San Diego, CA, USA, 2005: 137-146
- [49] Sabou M. From software APIs to Web service ontologies: A semi-automatic extraction method//Proceedings of the 3rd International Semantic Web Conference (ISWC'2004). Hiroshima, Japan, 2004: 410-424
- [50] Arpinar I B, Aleman-Meza B, Zhang R et al. Ontology-driven Web services composition platform//Proceedings of the IEEE International Conference on E-Commerce Technology, 2004: 146-152
- [51] Sheshagiri M. A planner for composing services described in DAML-S//Proceedings of the AAMAS Workshop on Web Services and Agent-based Engineering. 2003
- [52] Sycara K. Untethering semantic Web services, in semantic Web services//Martin D, Domingue J eds. *IEEE Intelligent Systems, Part 2*, 2007, 22(6): 11-13
- [53] Buxmann P, Hess T, Ruggaber R. Internet of services. *Business & Information Systems Engineering*, 2009, 1(5): 341-342
- [54] Ferrario R, Guarino N. Towards an ontological foundation for services science//Future Internet — FIS 2008: First Future Internet Symposium. Vienna, Austria. LNCS 5468. Springer-Verlag, 2009: 152-169
- [55] Oberle D, Bhatti N, Brockmans S et al. Countering service information challenges in the Internet of services. *Business & Information Systems Engineering*, 2009, 1(5): 370-390
- [56] Chen S Z, Feng Z Y, Wang H et al. Building the semantic relations-based Web services registry through services mining//Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS' 2009). Shanghai, China, 2009: 736-743
- [57] Wei Z X. Research on ontology integration and system implement [M. S. dissertation]. Tianjin University, Tianjin, 2009(in Chinese)  
(魏哲雄. 本体整合方法研究及系统实现[硕士学位论文]. 天津大学, 天津, 2007)
- [58] Milgram S. The small world problem. *Psychology Today*, 1967, 2: 60-67
- [59] Finin T, Ding L, Zhou L et al. Social networking on the semantic Web. *The Learning Organization*, 2005, 12(5): 418-435
- [60] Matsuo Y, Morib J, Hamasakia M et al. POLYPHONET: An advanced social network extraction system from the Web. *Journal of Web Semantics*, 2007, 5(4): 262-278
- [61] Hu J, Feng Z Y. Automated composition of semantic Web services using case-based planning//Proceedings of the 2009 International Forum on Information Technology and Applications. Chengdu, China, 2009: 246-251
- [62] Xu Bin, Li Juan-Zi, Wang Ke-Hong. Web service semantic annotation. *Journal of Tsinghua University (Science and Technology)*, 2006, 46(10): 1784-1787(in Chinese)

(许斌, 李涓子, 王克宏. Web 服务语义标注方法. 清华大学学报(自然科学版), 2006, 46(10): 1784-1787)

- [63] Wang Jian-Dong. Domestic information services research concept network analysis based on complex network method. *New Technology of Library and Information Service*, 2009, 25(10): 56-61(in Chinese)  
(王建冬. 基于复杂网络方法的国内信息服务研究概念网络分析. 现代图书情报技术, 2009, 25(10): 56-61)
- [64] Gombotz R, Dustdar S. On Web services workflow mining// *Proceedings of the International Workshops at Business Process Management*. Nancy, France, 2006: 216-228
- [65] Liang Q A, Chuang J-Y, Miller S et al. Service pattern dis-

covery of Web service mining in Web service registry-repository//*Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'06)*. Shanghai, China, 2006: 286-293

- [66] Zheng G, Bouguettaya A. Service mining on the Web. *IEEE Transactions on Services Computing*, 2009, 2(1): 65-78
- [67] OASIS. extensible Customer Relationships Language (xCRL). 2003 <http://www.oasis-open.org/committees/ciq/ciq.shtml#8>
- [68] Zhang L-J, Zhang J, Cai H. *Services Computing*. Berlin Heidelberg: Springer, 2007: 114-133



**CHEN Shi-Zhan**, born in 1975, Ph. D., lecturer. His current research interests include services computing, SOA.

**FENG Zhi-Yong**, born in 1965, Ph. D., professor, Ph. D. supervisor. His current research interests include knowledge engineering, services computing, security software engineering.

**WANG Hui**, born in 1981, Ph. D.. His current research interests include services computing, SOA.

## Background

As an emerging paradigm for architecting and implementing business collaborations within and across organizational boundaries, Service-Oriented Computing (SOC) is currently of interest to both software vendors and scientists. In SOC paradigm, service provider provides loosely coupled services while service requestor can easily compose these services into dynamic business processes with little cost. On the one hand, there are considerable effects and activities, which are based on AI planning and semantic Web technology, concentrated on Web service discovery, interoperation based on Semantics, service management and dynamic (or automatic) composition (sometime also called service orchestration or choreography). On the other hand, Since services are created by various providers using different programming languages, running in distinct hardware platforms and host machines, supporting different operating systems, there exist grand challenges in SOC. Traditional Web service registry, like UDDI and ebXML, has been facing great difficulty in information sharing and process mediation when processing service discovery, service combination and interaction. The main problems are as follows. First is ignoring semantics among services, i. e., semantic relations based on interaction and transaction context. And second, complexity, reliability and efficiency are big questions. Today, we are still on the road of syntactic, not semantic interoperation. Service composi-

tion is largely a static affair, not automatic composition of flexible business processes. And the efficiency of service discovery and composition is also a big question especially in large-scale environment. To overcome these limits, the authors propose to combine Semantic Web and Social Networking technology to explicitly merge abundant semantic information and socialized interactive relations, i. e. Service Relation (SR), into SOC paradigm. The authors propose the concepts of Services Network (SN) based on semantics and socialized interaction relations, introducing semantic reasoning and relational calculus to facilitate service discovery, composition and interaction in SOC. The authors ultimate goal is to realize "the Internet of Services". Firstly, three different granularities of Web service, including parameter level, operation level and Web-service level, are analyzed and defined. Secondly, the details of Service Mining, which includes Web service annotation, service relations mining and its application and leverage on SOC, are discussed. Last comes the experimental results and analysis. Duo to organize available Web services into service discovery, composition, and interoperation with abundant semantic information, business context and social relations, SN is firmly believed to be a new infrastructure for SOC and also key effort toward the implementation of "Internet of services".