维普资讯 http://www.cqvip.com

Web服务研究综述*)

丁长明 石 静 赵泽宇 薛向阳

(复旦大学计算机科学与工程系 上海200433) (复旦大学智能信息处理开放实验室 上海200433)

Review of Web Service Research

DING Chang-Ming SHI Jing ZHAO Ze-Yu XUE Xiang-Yang

(Dept. of Computer Science and Engineering, Fudan University, Shanghai 200433)

(Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433)

Abstract The importance of Web services has been recognized by industry and academic research. The two worlds concerned of two different dimensions. Industry has focused on modularization of service layers, while academic research has been mostly concerned with semantic expressiveness of service description. This paper gives a brief introduction to two worlds research and points out the challenge of research currently.

Keywords Web service, Web service architecture, Distributed computing, Semantic web

1 简介

当前的Web站点不仅能为用户提供静态的信息服务,它还能与用户进行交互,使用户不仅可以动态地访问Web上提供的软件服务和数据服务,而且可以通过Web访问远程计算机的硬件资源。

Web服务(Web Service)是下一代Web应用的机遇。作为分布式计算的一种新兴技术,相对于agent-based,grid-computing这两种分布式计算技术来说,Web服务出现得最晚,但是从学术研究走向商业应用的时间却是最短的。

那么,什么是Web服务呢?IBM从工业应用特点的角度给Web服务下的定义为:它是Web应用的一个新的种子(breed),Web服务是自包含的、自描述的、模块化的应用;Web服务提供了从简单请求到复杂商业过程的功能;一旦Web服务被部署,其他的应用(包括其它Web服务)就能够发现和调用部署的服务。

Web服务是模块化的应用。软件开发经历了无结构,面向过程,面向对象和面向组件的阶段。Web服务从某种程度上讲是更大规模的组件,Web服务的使用者可以根据全球各地提供的Web服务,合成具有更高商业价值的Web服务向其它使用者提供。

Web服务是松散耦合的、可重用的软件组件。Web服务内部实现技术对使用者来说是透明的。使用者只需要知道Web服务接口和语义即可,不需要了解其内部的实现细节。

Web服务是自治的服务。Web服务的提供者分布在全球各地,而每个提供者有自己的商业规则,因此不可能对服务提供者和服务使用者的系统平台做出任何限制,这样提高了Web服务开发的灵活性和各个团体开发的自主性,同时也增加了Web服务协调的难度。

目前对Web服务的研究有两个方向:工业应用研究和学术研究。其中工业应用研究的典型代表者是IBM。

在通用描述发现集成(Universal Description, Discovery and Integration, UDDI),简单对象访问协议(Simple Object Access Protocol, SOAP),Web服务描述语言(Web Service Description Language, WSDL),Web服务流语言(Web Services Flow Language, WSFL)等技术标准的支持下,工业界将Web服务的应用向前推进了一大步,其目标是:通过Web服务的应用获得商业的增值。工业应用的特点是采用封闭的服务描述词汇和标准的工业分类方法描述Web服务和检索Web服务。这使得Web服务应用的可扩展性和对服务功能的表达能力(对半结构化数据的描述,服务的约束条件,服务的类型和服务的继承关系等)受到限制,这样的Web服务的使用者是人而不是计算机。

相对于工业界的研究和应用,学术界的研究更注重对服务功能和服务描述的表达能力方面。它采用了语义Web的研究成果,使得对Web服务的自动发现,自动调用,自动合成和互操作,自动执行和监控成为可能。和工业研究相比,学术界的研究特点是:没有形成标准或事实标准;对Web服务语义的表达使得Web服务的使用者可以面向计算机。

本文主要比较工业界和学术界对Web服务技术的研究成果,重点阐述学术界的研究成果,并提出目前研究中存在的问题和解决问题的思路。

2 IBM Web服务体系结构

Web服务的典型过程如下:

(1)Web服务的提供者在Web上提供一个服务。他可以选择一个在线的注册中心注册服务。

(2)注册中心发布和定位一个服务。为了发现服务,注册中心提供了一种用来对服务进行分类的服务描述机制,其中包括服务的功能;服务的提供者;如何访问服务等信息。这些注册的信息由服务提供者提交给注册中心。

(3)服务的请求者在注册中心查找服务。查找服务的方式有两种:请求者或者通过浏览注册中心的方式查找服务,或者是提交一个查询。找到一个服务以后,请求者根据服务的描述信息设计应用程序绑定到服务并与之交互。

IBM Web服务体系结构如图1所示。可以看出SOAP,

*)自然科学基金(60003017)、国家863计划(2001AA114120,2002AA103065)、上海市政府科技发展基金(01QD14013,015115044)。丁长明 硕士研究生,主要研究方向为Web服务和语义Web。石 静 硕士研究生,主要研究方向为Web服务和语义Web。赵泽宇 硕士研究生,主要研究方向为Web服务和语义Web。薛向阳 教授,博士生导师,主要研究方向为多媒体信息处理和检索技术。

维普资讯 http://www.cqvip.com

WSDL,UDDI,WSFL是该体系结构应用的关键技术,它们各自的作用如下:

网络协议和简单对象访问协议(Simple Object Access Protocol,SOAP)是Web服务通讯的基础,其中SOAP封装了通讯中交换的消息。因为Web服务是自治的软件模块,因此需要有一个标准化的通讯方式表示服务之间通讯的消息,SOAP就发挥了这个作用。SOAP是一种简单的、轻量级的基于可扩展标记语言(Extensible Markup Language,XML)的机制,用于在网络应用程序之间进行结构化数据交换。SOAP既可以是面向文档也可以是面向远程过程调用(Remote Process Call,RPC)的交换消息封装。SOAP的通讯描述是抽象的,它必须被映射到具体的通讯协议上,如HTTP,FTP等。如今最常用的是映射到HTTP通讯协议上。

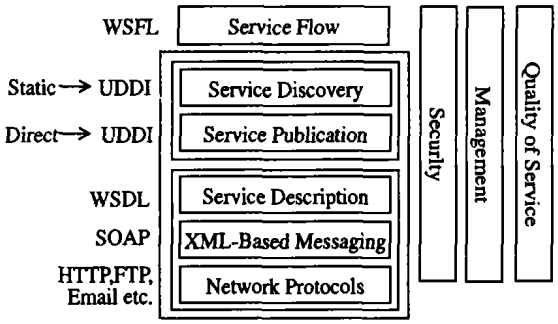

图1 IBM Web服务体系结构

Web服务描述语言(Web Service Description Language,WSDL)描述了服务的接口和与服务交互的其它相关信息。WSDL的扩展性有一定的局限性,WSDL的描述是完全基于元数据模型,不具备表达Web服务语义的能力。

通用描述发现集成(Universal Description,Discovery and Integration,UDDI)注册中心是IBM Web服务体系结构的核心。UDDI不仅包括了服务描述的数据结构,它还提供了一套应用程序接口(API)供服务的请求者访问服务的发布和服务的描述时使用。UDDI注册表的数据包括服务的白页,黄页和绿页。白页主要包括提供服务的公司名称、地址等与之联系的相关信息;黄页数据帮助对商业活动进行分类的信息,例如商业类型等信息。绿页数据则定义了提供的是何种服务以及如何与服务交互的信息。

Web服务流语言WSFL(Web Services Flow Language)定义了服务合成规范,它可以让用户通过合成Web服务从而构造商业上的增值服务。

其它相关的则是服务的安全、服务质量(QoS)和服务的管理等问题,这些问题贯穿服务的各个层次。

IBM Web服务体系结构是基于元数据模型的体系结构。然而由于缺乏有效的服务语义表达能力,这种体系结构可能会导致服务查询结果和用户需求之间产生较大的差异。语义Web服务的引入,较好地弥补了其语义描述方面的不足。

3 语义Web服务

3.1 语义Web服务简介

语义Web服务如今还没有被投入大规模具体的应用,基本上仍限于学术研究领域,我们现在所看到的是它的实验型的系统。

语义Web服务是语义Web的应用,语义Web解决的是如何让Web的语义能准确表示出来,从而能够实现基于语义的更强有力的搜索方法的相关问题。语义Web能够对Web服务进行更丰富的描述(如半结构化数据、数据类型、继承关系、语义约束等)。语义Web服务与IBM Web服务的比较如图2所示[1]。

属性	IBM Web服务	语义Web服务
服务类型	简单	可以合成
请求者	人	机器
提供者	注册服务	不需要注册服务
代理的角色	核心	削弱
服务描述	基于分类学	基于本体(ontology)
描述元素	封闭的	可以开放的
数据交换	基于语法	基于语义

图2 Web服务属性比较

语义Web服务和IBM Web服务相比最大的一个特点是:在IBM Web服务中处于中心控制地位的UDDI的代理机制有被削弱的趋势,取而代之的是具有智能化的agent。用户可以通过agent程序得到Web服务的描述。然后分析描述,并和服务绑定。

语义Web服务和元数据Web服务不同点在于:它强调对Web服务的自动发现,自动调用,自动合成和互操作,自动执行和监控。

自动Web服务发现:自动Web服务发现包括Web服务的自动定位。因为Web服务的语义描述被注册到某个注册中心,具有智能的可移动agent通过在不同的注册中心间移动而找到用户期望的服务,这可以通过有agent帮助的Web浏览方法实现。Letizia就是一个用户接口的agent,它能够根据用户预先配置好的服务查询配置文件浏览Web,这种方法也可以被使用在Web服务发现中[36]。例如:服务提供者可以在不同的层次上用不同的描述方法提供对服务的描述,而agent可以根据用户的配置文件在不同层次上查找用户期望的服务。

自动Web服务调用:目前许多的Web服务在执行过程中需要人的频繁干预。例如:在某个Web站点购书,用户需要填写购书的表单并提交。有时为了完成一次Web服务的执行,需要用户和Web服务之间多次交互。自动Web服务调用意味着一个软件的agent可以协助用户完成这些工作,而用户所要作的仅仅是让agent知道"我要购买一本名为《Web服务》的折扣最多的书"。agent会在网上的多个书店中选择一个满足用户需求的书店完成购书的任务。

自动Web服务合成和互操作:这个任务包括了Web服务的自动选择、合成和互操作来完成一个任务。复杂的Web服务一般由许多小的Web服务组合而成,例如:安排一次会议就可能包括了预定机票,预定客房等各种由不同服务提供者提供的服务。agent可以帮助用户协调各种服务的事务关系。

自动Web服务执行监控:在一个复杂的服务执行期间,用户经常需要跟踪和查询服务及其组件执行过程中状态的变化。

为了实现上述的Web服务的自动请求过程,软件agent需要对Web服务的描述,这种描述对计算机来说是可以解释的,同时还需要提供通过描述获取的访问Web服务的方法。

3.2 DAML-S的Web服务语义描述

DAML-S(DARPA Agent Markup Language - Service)

维普资讯 http://www.cqvip.com

是基于DAML+OIL的Web服务本体。它由DAML服务委员会提出的一套标记语言结构来描述Web服务的属性(properties)和服务的能力。DAML-S使得上述对Web服务的自动请求过程成为了可能。DAML-S是位于WSDL层之上的Web服务的语义描述语言。和WSDL相比,它不仅说明了服务和服务之间,服务与客户之间是如何发送信息,而且说明了发送什么(what)以及为什么发送(why)等语义方面的内容。它解决了WSDL并不具有或通过扩展WSDL后只是部分具有的对Web服务语义描述的能力以及Web服务发布和工作流等问题。DAML-S的描述包括关于服务的三部分信息:

ServiceProfile主要描述服务究竟是做什么的问题,这些问题是和商业相关的,也就是它向服务查找agent(service-seeking agent)提供关于服务的抽象的规范,服务查找agent可以通过这个规范确定该服务是否满足其需求。ServiceProfile的描述主要包括三部分:(1)服务的描述是对服务及服务提供者的描述。当用户浏览服务的注册中心时,这些信息被提供给用户;(2)服务功能行为的描述说明了服务的目的,它主要是服务的功能的信息,例如服务的输入,输出,服务执行的前提条件,后置条件等;(3)服务属性(functional attributes)的描述是对服务的补充性说明信息,例如:服务的响应时间限制,服务的开销,服务的地域范围等。ServiceProfile的作用是两方面的:对于服务提供者来说,可以通过ServiceProfile来发布他的服务;而对于服务的使用者来说可以通过构造ServiceProfile或其中一部分构造服务的查询请求。和UDDI以及WSDL相比,DAML-S的描述具有更强的表达能力,它更清晰地表达了在UDDI和WSDL中不具有的关于Web服务能力的描述,DAML-S的ServiceProfile和agent系统(如LARKS和OAA)中的描述语言一样关注的是服务的能力[37,38]。但同时DAML-S还充分利用了DAML+OIL本体及其推理能力而提高了更好定位相关服务的能力。ServiceProfile确保了服务的自动发现的实现。

ServiceModel指出了服务如何工作的问题。它是和过程相关的。服务查询agent可以通过ServiceModel更深入地了解关于服务的四个主要信息:(1)深入地分析一个服务是否满足agent的需求;(2)从多个服务的描述中合成一个任务的描述;(3)复杂服务执行期间各个组件的协调关系;(4)服务执行过程的监控。ServiceModel包括两个重要的组件:过程本体(Process Ontology)和过程控制本体(Process Control Ontology)。其中通过使用过程本体支持服务的自动调用、合成和互操作;通过使用过程控制本体而支持服务的自动执行监控。在目前的0.6版本的DAML-S中,过程控制本体没有发布。

ServiceGrounding规范了agent如何访问Web服务的技术方面的细节问题,例如通讯的协议、交换消息的格式、交换消息的序列化、传输端口以及服务的地址等。ServiceProfile和ServiceModel是关于服务的抽象描述,而ServiceGrounding处理关于服务的具体实现规范。

因为DAML-S对服务的描述是抽象的,它必须被映射到具体的服务描述上,服务才能被使用。目前的做法是将DAML-S映射到WSDL上,这样损失了DAML-S的部分语义,因此如何将DAML-S具体化仍然是一个研究课题。

3.3 语义Web服务体系结构

在语义Web服务中通过使用本体,服务描述的表达能力和可扩展性得到了极大的发挥,如表示半结构化数据、约束、类型支持和继承等。而IBM Web服务在确保对Web服务的可管理性,模块化等方面具有更强的优势,这两者的结合必然具有更大的益处。但是语义Web服务并没有规范的体系结构,大部分试验系统通过结合目前的IBM Web服务体系结构的工业标准设计实现。

对于Web服务的访问,基本有如下三种情况:(1)用户直接访问Web服务(图3中的箭头1所示),这种情况下要求用户对Web服务的描述和其位置都能够确定。(2)用户通过查询处于核心位置的注册中心获得关于Web服务的信息,然后和服务交互(图3中箭头2所示)。这种做法通常是通过扩展UDDI的TModel的方法,将DAML-S的描述映射到UDDI中心的数据模型上[9,10]。这样充分利用了UDDI作为工业标准的优势。(3)用户直接和agent交互,将请求交给agent,agent或者和UDDI注册中心交互,或者通过Crawling程序从Web上搜索DAML-S的描述,分析服务的DAML-S描述而获得语义Web服务的基本功能(图3中的箭头3和4所示)。

和工业标准的Web服务一样,agent间的交互也需要相应的通讯协议,目前通常是使用KQML(Knowledge Query and Manipulation Language)或者ACL(Agent Communication Language)作为通讯消息的抽象描述。

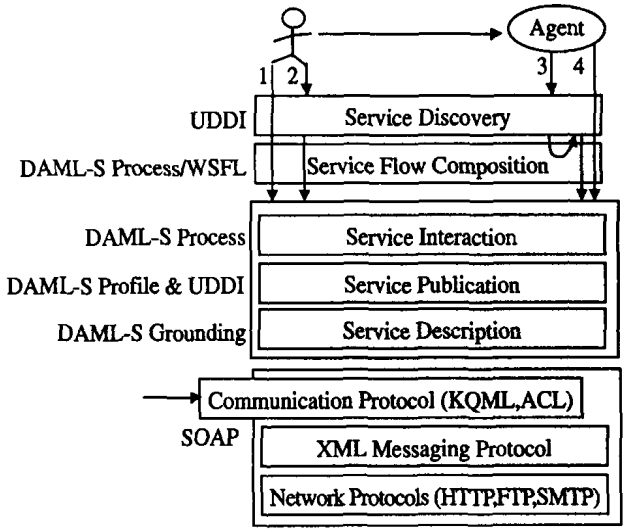

图3 语义Web服务体系结构

在语义Web服务中,DAML-S被作为Web服务的描述解决了WSDL中并不具有的对Web服务能力的描述,这种Web服务语义的描述对Web服务的自动发现、自动调用、自动合成和互操作、自动执行和监控是至关重要的。对于agent来说,只有掌握了Web服务的语义才能实现上述过程。

4 研究中的主要任务和挑战

从上述过程中,我们可以看出,工业标准的Web服务和语义Web服务特点的不同决定了对它们研究思路和方法的不同。

4.1 IBM Web服务的挑战

对于工业标准的Web服务,主要的特点是:结构化和模块化比较清晰,有成型的各种技术标准支持,应用的过程相对较容易。主要缺点是使用的描述语言缺乏对Web服务语义的描述,这使得Web服务的使用者面向人而不是计算机。也因为描述语言缺乏表达Web服务语义的能力,所以在研究和应用中有如下问题需要解决:

维普资讯 http://www.cqvip.com

4.1.1 服务合成 Web服务的合成有两种做法:静态合成(设计期间完成服务的合成)和动态合成(服务在运行过程中合成)。由于WSDL对Web服务语义表达能力的缺陷,以WSDL为基础的Web服务合成比较困难,基本上难以实现服务的动态合成。但是无论静态合成还是动态服务合成,它们都有共同的特点和共同的问题需要解决。

服务的合成必须满足:(1)协调合成的服务中服务的调用顺序管理。(2)服务之间数据流的管理。(3)服务的事务性管理。

除了上述要求以外,服务的合成必须要做到在不影响Web服务和使用者之间已经存在的关系的前提下,满足对Web服务合成的动态自适应性(服务提供的切换,已经存在的服务的钝化(deactive)和激活(active),新服务的创建(creation))。

服务合成中一个首要的问题是服务的异构性问题(相同的语义,不同的描述方式),这种异构性是由Web服务的自治性引起的。由于WSDL不包含对服务的语义描述,因此在合成时需要对WSDL的描述进行扩展,但是WSDL规范对WSDL的扩展有比较严格限制,因此需要考虑的是如何在WSDL层次之上进行扩展。不同研究组织提出了各自的框架来解决这个问题[4,39]。但是所有的工作解决问题的本质不外乎是服务操作的事务处理问题以及异构服务的均质化(homogenized)接口问题。我们认为可以通过在WSDL的描述层之上构建能表示部分语义的合成流语言来实现服务合成的方法。这样既可以充分利用已有工业标准的益处,也可以作为目前对Web服务合成的解决方案的补充。

4.1.2 Web服务的查询和匹配 目前UDDI的查询是基于框架(frame-based)的查询技术,图4所示是该方法的一个例子。也有学者在框架查询方法的基础上部分采用了语义的方法(例如LARKS),根据服务I/O的泛化和特殊化的关系来查找服务[40]。这些方法的特点是查准率比较高,但是代价是每一个服务都必须被模型化为框架,手工的工作量比较大。在框架查询的基础上发展的还有推理检索方法,这种方法中所有服务的属性(输入,输出,功能,性能)都必须用逻辑学的方法将其形式化(图5所示)。

Description	a service to find the average of a list of integers
Input	integer
Output	real
Execution Time	number of inputs * 0.1 msec

图4 对整数排序服务基于框架的描述

Name:	set-insert
Syntax:	set-insert(Elem,Old,New)
Input-types:	(Elem:Any),(Old:SET)
Output-types:	(New:SET)
Semantics:	
Precond:	¬member(Elem,Old) member(Elem,New)
Postcond:	∧∀ x(member(x,Old)→member(x,New)) ∧∀ y(member(y,New)→(member(y,Old) ∨y=Elem))

图5 推理检索的服务描述

这样检索就是查找那些能够被证明可以获得查询中的功能的单项。可以考虑如果我们预先定义一些非冗余的逻辑谓词以及相关服务和查询属性的形式化的方法,那么从理论上来讲,这种查询的方法可以获得比较好的查准率和查全率(recall),但是这种方法面临一个严重的问题:对于非专业的用户来说,用形式逻辑模型化一个查询或服务是一件比较困难的事情。更不用说那些规模比较大的服务(例如对Microsoft Word的行为形式化)。服务查询方法的设计既要考虑性能问题,还要考虑可行性。

4.2 语义Web服务的挑战

4.2.1 Web服务的描述 DAML-S是目前发展最成熟的对Web服务语义的描述方法,但是正如上述,它目前没有提出解决过程控制本体的方案,因此还不能解决Web服务自动执行监控的问题。

在语义Web中通常和DAML-S结合的KQML和ACL通讯协议虽然在目前使用比较广泛,但是DAML-S是通过定义的本体来描述Web服务,而通讯协议又不具有表达本体的能力,因此在通讯层上(图3)不能反映Web服务和agent对本体的需求,因此有必要考虑对通讯层的扩展。

DAML-S在目前需要被映射到WSDL才能投入具体的应用,映射的过程损失了DAML-S中的语义;对于复杂的Web服务(合成的Web服务)映射过程比较困难。

总体来说在服务描述方面仍然需要研究抽象描述如何转化为包含准确语义的具体实现方法。

4.2.2 Web服务的查询和匹配 语义Web服务最大的特点是能够表达服务的语义,这使得我们可以采用语义匹配的方法来查询服务。用来描述服务的本体反映了服务的语义,因此可以设计基于本体的查询方法[4,40]。一种方法是以过程模型为核心,预先建立过程模型的本体来捕获过程的知识。将服务的功能看作一个过程模型,为服务模型建立索引(将服务模型和其构件插入到正确的本体段);查询也被表示成过程模型,这样可以根据过程本体中的语义关系查询服务。查询的设计要同时考虑查全率和查准率的问题。同时提高查全率和查准率固然是最理想的状态,但考虑到实际的应用,需要做一些折中的工作,一般情况下语义Web服务的查询都是基于本体设计的,这种查询对于非专业的用户来说比较困难,因此如何将查询的构造对大多数用户可行也是研究中值得关注的问题。

查询中一个关键的问题是对Web服务分类的问题。分类可以通过手工的方法实现,也可以通过仔细定义相似度实现半自动或自动的Web服务分类。DAML-S的基础-过程模型(Process Model)以流程图的方式建立,这种流程图是标准的程序开发的一部分,因此在多数情况下,我们需要建立索引的服务已经有了相应的过程模型,而这种模型在某些情况下可以被认为是某一种过程本体的具体化。和上述基于框架的模型相比,过程模型对语义的表达能力更强,这就意味着如果能够为其定义相似度,那么这种相似度可以比传统的相似度具有更强的分类表达能力。我们可以考虑用语义网络(semantic network)结合词频统计的方法计算自然语言元素的相似度,同时结合分类推理及图论的相似度计算方法来设计一种用于Web服务自动分类的相似度计算方法。

4.2.3 Web服务的合成 有了对Web服务的语义描述,就可以实现Web服务的动态合成。DAML-S已经包含了Web服务合成的控制结构标记和相应的语义。但是如何能使合成后的服务能够正常地执行(避免死锁),合成服务的执行必须能在一个安全的状态下进行[5]。这必须从理论上进行研

维普资讯 http://www.cqvip.com

究。首先可以通过使用谓词逻辑形式化 DAML-S 中相关子集的语义，然后可以用 Petri 网理论形式化 Web 服务的描述，并用 Petri 网理论解决合成的 Web 服务的执行是否有潜在进入不安全状态的可能性。

因为语义 Web 服务的描述可以表达 Web 服务的语义，所以可以在 Web 服务描述基础上实现 Web 服务的动态合成，有合成就必须有查询，要查询服务就要能够有描述服务能力的方法。因此针对语义 web 服务的特点，SWWS(Semantic Web Working Symposium)提出了如下的工作任务：(1)提供计算机可以理解的 Web 服务的描述框架。(2)定义 Web 服务的发现框架。(3)提供可扩展的 Web 服务的协商框架[6]。

结束语 Web 服务目前在国内的应用尚处在初始阶段，尤其是语义 Web 服务仍然处在研究阶段，这些研究工作有其哲学的基础。虽然目前通过引入哲学中的本体论解决了部分语义表示的问题，但是要将其投入实际的应用，仍然有许多工作要做。

参 考 文 献

1 Sollazzo T, Handschuh S, Stabb S, Frank M. Semantic Web Service architecture---Evolving Web Service Standards toward to the Semantic Web. www. aifb. uni-karlsruhe. de/WBS/sst/Research/Publications/sub-flairs2002. pdf

2 McIlraith S, Son T C, Zeng H. Semantic Web services, IEEE Intelligent Systems. Special Issue on the Semantic Web, 2001, 16(2):46～53

3 Ankolekar A, et al. DAML-S: Semantic markup for web services. In: Int. Semantic Web Working Symposium, 2001. 411～430

4 de Figueiredo Pires P. WebTransact: A Framework for specifying and coordinating reliable web services compositions. http://citeseer. nj. nec. com/context/15289/0

5 Narayanan S, McIlraith S A. Simulation, Verification and Automated Compositon of Web Services. Artificial Intelligence

6 Semantic Web Working Symposium. http://www. semanticweb. org/

7 Hendler J, McGuinness D L. Darpa agent markup language. IEEE Intelligent Systems, 2001, 15(6):72～73

8 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001, 284(5):34～33

9 Paolucci M, Kawamura T, Payne T R, Sycara K. Semantic matching of web services capabilities In ISWC2002, 2002

10 Paolucci M, Kawamura T, Payne T R, Sycara K. Importing the Semantic Web in UDDI. http://citeseer. nj. nec. com/540826. html

11 http://www-3. ibm. com/software/solutions/ webservices/pdf/WSFL. pdf

12 DAML Joint Committee. Daml + oil (march 2001) language. http://www. daml. org/2001/03/daml-+oil/index. html, 2001

13 Christensen E, Curbera F, Meredith G, Weerawarana S. Web Services Description Language (WSDL) 1. 1. http://www. w3. org/TR/2001/Note-wsdl-20010315, 2001

14 Martin D, Cheyer A, Moran D. The Open Agent Architecture: A framework for Building Distributed Software Systems. Applied Artificial Intelligence, 1999, 13(1-2):92～128

15 Peng Y, Ivezic N. Semantic resolution inf multi-agent systems. In: Proc. Of Goddard/JPL Workshop On Radical Agent Concepts. 2002

16 UDDI. The UDDI Technical White Paper. http://www. uddi. org/, 2000

17 W3C. Extensible markup language (xml) 1. 0 (second edition). http://www. w3. org/TR/2000/REC-xml-20001006, 2000

18 W3C. Soap version 1. 2, w3c working draft 17 december 2001. http://www. w3. org/TR/2001/WD-soap12-part0-20011217/, 2001

19 Sycara K, Klusch M, Widoff S, Lu J. Dynamic service matchmaking among agents in open information environments. ACM SIGMOD Record (Special Issue on Semantic Interoperability in Global Information Systems), 1999, 28(1):47～53

20 DAML-S Coalition. Daml-s 0. 6 draft release (December 2001). http://www. daml. org/services/daml-s/2001/10/, 2001

21 DAML-S Coalition. DAML-S: Web service Description for the Semantic Web. Forthcoming in Proceedings of the First International Semantic Web Conference (ISWC01), 2002

22 Chandrasekaran S, et al. Composition, Performance analysis and Simulation of Web Services. http://lsdis. cs. uga. edu/lib/download/CMSA+02. pdf

23 Austin D, Barbin A, Ferris C, Garg S. Web Service Architecture Requirements. http://www. w3c. org/TR/wsa-reqs, accessed 22 Sep. 2002

24 Berners-Lee T, Handler J, lassila O. The Semantic Web. http://www. sciam. com/article. cfm?articleID = 00048144-10D2-1C70-84A9809EC588EF21, accessed 22 Sep. 2002

25 Bartolini C, et al. A Framework for Automated Negotiation. www. hpl. hp. com/research/tesl/Publications/ tech%20reports/HPL-2001-90. pdf

26 Moreau L. Agent for the Grid: a Comparision with Web Services (Part I: Transport Layer). In IEEE International Symposium on Cluster Computing and the Grid, Berlin, Germany, May 2002

27 Avila-Rosas A, Moreau L, et al. Agents for the Grid: A comparison with Web Services (Part II: Service Discovery). In: IEEE Intl. Symposium on Cluster Computing and the Grid, Berlin, Germany, May 2002

28 van Harmelen F, Patel-Schneider P F, Horrocks I. Reference Description of the DAML + OIL Ontology Markup Language, March 2001. http://www. daml. org/2000/12/reference. html

29 Barros F A, Goncalves P F, Santos T L V L. Providing context to web searches: The use of ontologies to enhance search engine's accuracy. Journal of the Brazilian Computer Society, 1998, 5(2):45～55

30 Luke S, Spector L, Rager D, Hendler J. Ontology-based web agents. In: Proc. of the 1st Intl. Conf. Autonomous Agents, 1997. 59～66

31 Maedche A, Staab S. Ontology learning for the semantic web. IEEE Intelligent Systems, 2001, 16(2)

32 Horrocks I, et al. The Ontology Inference Layer OIL, Aug. 2000

33 http://www. scientificamerican. com/article. cfm? articleID = 00048144-10D2-1C70-84A9809EC588EF2

34 Lu S, Dong M, Farshad F. The Semantic Web: Opportunities and Challenges for Next-Generation Web Applications. http://informationr. net/ir/7-4/paper134. html

35 Seely S. SOAP: Cross Platform Web Service Development Using XML. 机械工业出版社, 2002

36 Lieberman H. Letizia: An Agent that assists web browsing

37 Martin D, Cheyer A, Moran D. The Open Agent Architecture: A Framework for Building Distributed Software Systems. Applied Artificial Intelligence, 1999, 13(1-2):92～128

38 Sycara K, Klusch M, Widoff S, Lu J. Dynamic Service Matchmaking among agents in open information environments. Journal ACM SIGMOD Record, 1999, 28(1):47～53

39 Paulo F, Pires M, et al. Building Reliable Web Services Compositions. http://www. jeckle. de/ws-rsd/abstracts/

40 Klein M, Bernstein A. Searching for Services on the Semantic Web Using Process Ontologies. http://www. semanticweb. org/SWWS/program/full/paper2. pdf