一种基于主体的语义Web服务模型

袁金平 姚 莉 鲍爱华 刘 芳
(国防科学技术大学信息系统与管理学院 长沙 410073)

摘 要 要实现Web服务组合自动快速准确性,必须对Web服务进行语义标注,使计算机能在"理解"服务的基础上实现快速准确查找和自动组合。Agent以其独特的性质,被普遍认为是可以代替人做出某些行为的实体。借鉴主动服务的思想,在传统面向服务体系架构(SOA)的基础上,将Agent技术引入到SOA模型中,提出了一种基于主体的语义Web服务模型,采用Agent来封装不同角色模块,使得Web服务的调用和组合过程可看成是代表各自角色的Agent之间的协商和协作解决问题的过程。

关键词 面向服务体系架构,主体,多主体系统,语义Web服务,语义Web服务组合

Agent Based Model for Semantic Web Services

YUAN Jin-ping YAO Li BAO Ai-hua LIU Fang
(School of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract In order to achieve the goal of finding services rapidly and accuratly and composing them automatically, Web services should be marked-up with semantics, which makes them computer-interpretable, use-apparent and agent-ready. Agent, with its fundamental characteristics, was recognized as an entity that can do work for man in some areas. In our work, after the research of Service Oriented Architecture (SOA) and Agent, with the idea of Active Service, a model for Semantic Web Services based on multi-Agent system was proposed. With this method, the course of invocation and composition of Web services can be regarded as a process of problem-resolving with negotiation and collaboration among multi-Agent.

Keywords Service oriented architecture, Agent, Multi-agent system, Semantic Web services, Semantic Web services composition

1 引言

在分布式计算领域,Agent与语义Web服务是重要研究方向,为网络上不同组织和个人的不同应用的互操作提供了可能性。Web服务是自包含自描述的模块化应用,它通过Web来发布、发现和调用。每个Web服务都有其独特的功能,也就是说能力水平是有限的。为了满足不同的用户(需求多样性),必须定义各种形形色色的能够独立完成相应任务的服务。然而,先不讨论用户需求是变化不断的,单从需求复杂度来看,很多情况下单个服务是无法达到用户需求目标的。为此,我们就需要Web服务组合技术,通过将多个独立功能的单个Web服务组合起来协调完成相对比较复杂的任务。实现Web服务组合的前提是能够发现合适的服务,然而随着越来越多的Web服务产生,在如此开放动态的环境下,快速准确地找到合适的服务变得越来越困难。Tim Berners-Lee在1998年提出的语义Web(亦称为Web3.0)是对当前Web的一种扩展,其中的信息被赋予明确定义的含义,使机器和人能更好地协同工作;它使人类从搜索相关资源的繁重任务中解放出来。自然而然,人们就想到将语义Web技术融入到Web服务中,提出了语义Web服务。它的提出主要是为了实现Web服务的自动发现、调用、组合及执行、监控。借助语义Web技术,对Web服务进行语义标注,使其成为计算机可理解、Agent可处理和用户透明的软件实体[1]。

有了语义Web技术,采用OWL-S(语义Web服务的描述本体)对Web服务进行语义标注,使其对计算机来说是可以完全"理解"的,从而可以实现基于内容的快速准确的查找服务。在Web服务组合过程中,总是需要用户的关注和频繁介入,才能较好地达到用户的意图。如何在减少用户干预的情况下,也能够较好地执行用户意图?

Agent是分布式计算与人工智能相结合的产物。它能够持续、自主地运行在特定的环境下,不需要人的干预,能够通过感知外部环境,自主决策采取相应的行为。自主性、协作性、适应性和社会性被认为是一个实体,是Agent的基本判别标准,从这4个特性中也可以看出,Agent具有人的一些特性,被认为是人类某些行为的最理想"代言人"。

语义Web与Web服务的结合方便了Agent访问调用。

到稿日期:2008-07-17 返修日期:2008-11-24 本文受国家自然科学基金(项目编号70371008)资助。

袁金平 男,博士研究生,主要研究智能决策技术、语义Web服务,E-mail:yuanjp@nudt.edu.cn;姚 莉 女,教授,博士生导师,主要研究知识管理、Agent技术、本体工程;鲍爱华 男,博士研究生,主要研究分布式人工智能、语义Web服务组合、本体进化;刘 芳 女,博士,主要研究本体技术、Web服务。

通过 OWL-S 对 Web 服务进行语义标注，使其对 Agent 来说是可以完全“理解”的。比较研究语义 Web 服务与 Agent，发现两者存在很大的共同特性：都是特定领域问题的求解器；都具有一定的语义处理能力；都可以通过组装完成更加复杂问题的求解等。同时，在语义 Web 服务组合过程中，有一系列的活动，如服务的发现、服务的调用、服务间的交互、执行及监控等，Agent 都能够自动完成操作。通过引入 Agent 技术，在 Agent 自动理解并处理语义 Web 服务的基础上，可以更方便地实现服务的自动查找、调用、组合和监控，主动地为用户提供良好稳定的组合服务。Agent 技术与语义 Web 的结合对提升 Web 服务的能力起了巨大的作用，使得 Web 服务能够提供更加有效优良的服务，也同时减少用户干预的频率[2]。

本文在语义 Web 服务中引入 Agent 技术，来解决上面的问题。比较 Agent 与 Web 服务的相似性，将 Agent 引入到 Web 服务的经典体系结构 SOA 中，提出了基于多 Agent 的语义 Web 服务框架。

2 Agent 与语义 Web 服务

2.1 Agent

“Agent 是一种在特定环境中持续、自主地运行的软件实体，通常与其他 Agent 一起，联合求解问题”[3]。即 Agent 是某一领域中的行为者，表现为它具有能够完成一项或多项任务的能力。对外界来说，它是一个独立自主的行为主体，通过感知外部环境自主决策采取什么样的行为，在必要的情况下能与其他的实体（Agent、用户、通信设施）进行交互。Agent 的基本特点是：自主性、协作性、适应性和社会性。一个 Agent 一般是针对特定领域的，解决特定的问题，其能力往往受其知识库和资源库的限制，因而无法适应开放、动态的分布环境。为了解决那些靠单个 Agent 是无法独立完成的复杂任务，人们提出了多 Agent 系统（MAS）。MAS 是由多个功能相对独立的 Agent 构成的集合，每个 Agent 都有特定领域、不同程度的问题求解能力，Agent 之间通过对话、协商、协作，共同完成单个 Agent 无法完成的问题求解任务。也就是说，MAS 是由分布在网络上的多个问题求解器松散耦合而成的大型复杂系统，这些问题求解器相互作用以解决由单一个体的能力和知识所不能处理的复杂问题[3]。这里的问题求解器可以称之为 Agent，同时也可以把它理解为封装好的 Web 服务。

2.2 语义 Web 服务

“Web 服务”最早出现在 1999 年微软旧金山中心的一次记者招待会上。它被描述为一组存在于 Internet 上的应用程序，向用户提供各种跨平台的信息发布、共享以及各种应用程序所规定的服务。典型的 Web 服务框架包含 3 种角色模型：服务提供者（Provider），服务注册中心（Broker）和服务请求者（Requestor）。服务提供者创建服务描述并将其发布服务到一个或多个服务注册中心；服务注册中心管理这些发布的信息供服务请求者查询；服务请求者向注册中心发出查询消息，当服务请求者通过评估发现了满意的服务时，便根据服务描述绑定该服务。

前面已经谈到，语义 Web 服务是语义 Web 与 Web 服务两种技术的结合（为了实现庞大服务计算机的自动化，这是分布式计算研究的必然趋势）。语义 Web 服务的框架结构基于典型的 Web 服务框架，对 Web 服务进行了相应的语义信息的标注，即通过 OWL-S 本体对一个服务进行描述。一般来说，如图 1 所示，OWL-S 从以下 3 个侧面来描述一个 Web 服务：ServiceProfile，ServiceModel，ServiceGrounding。ServiceProfile 描述 Web 服务的功能和交互信息；ServiceModel 描述 Web 服务的内部流程，即服务是如何工作的；ServiceGrounding 描述如何对 Web 服务进行访问（通信级信息）。有了 OWL-S 对 Web 服务的语义标注，Agent 能够在“理解”的基础上访问调用 Web 服务。

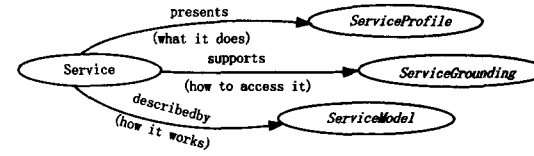

图 1 服务顶层本体

2.3 比较研究

通过比较研究 Agent 与语义 Web 服务，发现两者在分布式计算领域具有很大的相同特性，如都是封装好功能的应用模块；都是基于语义信息的处理；都可以集多个个体的能力协作完成复杂问题等。为此，有的研究认为可以实现 Agent 与语义 Web 服务的相互调用，他们认为通过提供合适的 WSDL 到 ACL 的映射，Web 服务就可以调用 Agent[4]。采用 Agent 调用语义 Web 服务是可行的，而 Web 服务调用 Agent 有待进一步的研究，因为 Web 服务的调用是在知道对方（Web 服务及其他）的功能及接口参数的前提下，也就是说 Agent 必须暴露其预先确定的行为，这与 Agent 的自治性相背离。Agent 具有“知识级”的通信及基于知识的推理能力，在语义 Web 服务中嵌入 Agent 技术，能够很好地处理语义 Web 服务，使得语义 Web 服务能够给人类提供更加优质和稳定的服务。

3 相关工作

随着 Web 服务数量的急剧增加，研究语义 Web 服务组合必然成为趋势。建立语义 Web 服务的目的就是在实现计算机“理解”Web 服务的前提下自动组合服务，以减少人的负担。Agent 具有的基本特性使其被普遍认为是可以代替人做出某些行为的最佳实体。通过 Agent 与语义 Web 服务的比较研究，人们认识到将 Agent 引入语义 Web 服务及其组合研究中，将大大方便服务组合，减轻用户的负担。文献[5]设计开发的智能会议系统 EasyMeeting 采用多 Agent 系统、语义 Web 本体、推理技术等实现在不同的会议环境提供相关的服务和信息给不同的与会者。结合了语义 Web 服务和多 Agent 系统各自特性，通过 Agent 对会议环境的感知，调用相应的服务提供给用户（与会者、会议设备等）。文献[6]将语义 Web 服务看成是独立的 Agent，组合服务则是多 Agent 系统。作者通过将一个个 Agent 表示为由 DAML-S（OWL-S 的前身）描述的具备一定功能的原子服务或组合服务，当组合服务被调用执行时，它们就可以被认为是一系列可利用的具有语义的行为（Agent），通过 FIPA 规范（Agent 的通信机制）进行交互通信。在文献[4]中，作者通过分析语义 Web 服务与 Agent 技术之间的关系，为了整合两种技术，设计实现了 Agent BDI 模型与语义 Web 服务的整合平台。作者认为，Agent 的

主要角色是封装用户的意图，协调用户的目标，并最终调用Web服务，完成相应的任务。当基于传统的工作流的Web服务组合技术无法满足人们的需求时，很多人提出了将Web服务组合技术与Agent技术结合，如文献[7]给出了基于多Agent技术的Web服务动态组合系统MAS-WS的框架。同时在文献[8]中，作者提出了基于QoS和Agent的动态Web服务选择、组合的方法，作者在SOA体系架构中的服务代理和服务消费者注入Agent技术，用于封装用户意图和存储QoS信息，从而提出了基于Agent的SOA系统结构，服务使用者与服务代理之间的交互表现为Agent之间的协作。

4 基于主体的语义Web服务模型

4.1 模型设计

本文以经典的Web服务模型SOA为基础，引入Agent来封装用户的意图和处理用户间交互的信息。借鉴主动服务[9]的思想，将Agent嵌入到SOA模型中，提出了基于Agent的语义Web服务模型，如图2所示。Agent技术的引入，使得SOA更具智能和自动能力。

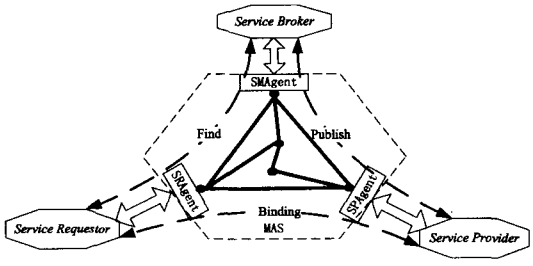

图2 基于多Agent的语义Web服务模型

本模型包含4个角色模块：服务提供者(Service Provider)，服务注册中心(Service Broker)、服务请求者(Service Requestor)和MAS。其中服务提供者，服务注册中心、服务请求者与经典的Web模型中各角色完成相同。而引入的MAS内核结构如图3所示，包含服务请求者Agent(SRAgent)、服务提供者Agent(SPAgent)、服务管理Agent(SMAgent)、目标分解Agent(GDAgent)、服务组装Agent(SCAgent)以及模型库、知识库和推理机RACER等。MAS中Agent的主要功能如下。

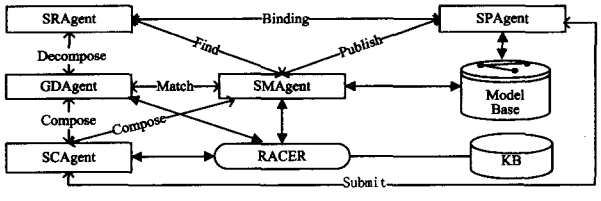

图3 MAS内核结构

(1)SRAgent，SPAgent，SMAgent

SRAgent与SPAgent用于封装用户(分别面向服务请求者和提供者)意图，SMAgent面向注册中心，负责服务的管理和发现任务。

它们分别代表服务提供者、使用者和注册中心，使得两两之间的交互表现为Agent之间的协商和协作。即服务发现的过程是根据SRAgent的意图，SMAgent搜索可用的服务及模型的过程；服务发布过程是SPAgent将描述好(OWL-S规范)的服务提交给SMAgent组织管理；服务绑定则是SRAgent与SPAgent的对话与协作，实现调用其服务求解问题。

(2)GDAgent

GDAgent能够将复杂的任务分解为相对比较简单能够实现的子任务。在SMAgent发现没有现成可用的服务及模型时，通过GDAgent与SRAgent，SMAgent的协作，将SRAgent的意图(目标)分解为多个子目标，直到模型库中匹配到可用的服务模型，从而实现复杂问题分解求解。

(3)SCAgent

将多个功能相对独立的简单服务，根据要求组装起来形成能够解决复杂问题的组合服务。在GDAgent分解及SMAgent匹配到可用服务后，SCAgent根据SMAgent意图将这些简单服务组装为新组合服务模型(分解子目标的逆过程也就是服务的组合过程)，并提交给SPAgent予以发布。

通过上面的分析，该模型中主要包含的操作有：发布、查找、绑定、分解、匹配、组装、提交。其中发布、查找和绑定与传统Web服务模型基本一致，不过它们都是通过封装各自意图的Agent来完成。在本模型中，查找首先是查找现有的服务模型库，看是否存在现有的能够直接满足请求者需求的服务。分解是在查找结果显示没有现有的服务供使用时，启动GDAgent根据服务请求者的目标并通过SMAgent挖掘模型库现有服务，将其分解为多个能够实现的子目标；匹配是分解过程中的一个操作，通过匹配GDAgent才知道需不需要进一步对子目标进行分解；组装则是SCAgent与GDAgent协作完成服务的组合，形成能够满足请求者的新的组合服务模型；提交是SCAgent将新的服务模型及相关信息提交给SPAgent发布(这是一个全新的服务，提交给服务提供者发布到注册中心和模型库)，以备下次使用。

综合来说，服务提供者构建的Web服务通过SPAgent与SMAgent协作将Web服务发布到服务注册中心；当服务请求者有问题求解任务时，创建SRAgent后，通过SRAgent发布请求者意图给SMAgent，由SMAgent搜索模型库得到能够满足请求者的Web服务；当模型库中不存在合意的Web服务时，SRAgent再将请求者的意图发布给GDAgent，此时GDAgent在匹配现有可用服务的基础上，对用户的目标进行分解；目标分解后，通过SCAgent将各个子服务按照一定的流程(如目标分解的逆过程)组装形成新任务求解的组合服务，最后将这个新的Web服务提交给SPAgent以发布。

4.2 实例描述

下面以车主卖车为例，来阐述该模型的服务处理过程。

我们知道卖方的意图(目标)只有一个：即卖出车，获得更多的钱。为了达到目标，车主可能在网络上寻求相应的服务：汽车估价服务(Car Valuation Service，WS1)、汽车拍卖服务(Car Auction Service，WS2)、在线支付服务(Online Financial Service，WS3)等等。为了更清楚地描述模型的处理过程，我们采用UML协作顺序图来刻画各个Agent的协作顺序关系，如图4所示。

需要说明的两点是：1)假定3个服务由同一个SPAgent发布，现实中它们可能来自不同的服务提供者；2)图中的交互不仅仅是消息的传递调用，它代表了Agent之间的协作。

要达到卖主的目标，需要将汽车估价服务、汽车拍卖服

(下转第177页)

EON	76%	84%	91%	95%
Russia	72%	83%	80%	85%

表 2　SNAX_Map 与 SNAX_Map+的 F 值对比

	SNAX_Map	SNAX_Map+
EON	0.82	0.89
Russia	0.75	0.84

相似度传播公式是一个迭代的公式，迭代收敛速度也是判断迭代公式好坏的一个因数，一般来说，迭代收敛速度越快，迭代公式相对也较好。本文以 SNAX_Map+系统不再产生新映射对为迭代终止记号。从表 3 中看出 SNAX_Map+系统的相似度传播迭代算法在收敛速度上还是比较好的，基本在迭代 3 次以内实现稳定，从表中也可以看出，迭代的次数跟本体的规模也有关系，规模越大，迭代次数也相应增加。

表 3　SNAX_Map+在 EON 与 Russia 数据集上的平均迭代次数

	EON	Russia
Iteration times	2.78	3

结束语　相似度传播在本体概念相似度计算中有着广泛的运用，然而目前的相似度传播算法却并未对相似度传播值进行合理定量分析。本文分析了对于已匹配的节点，其概念信息量的大小对于相似度传播的影响关系，提出了基于概念信息量的相似度传播算法，从而能够进行更精确的相似度传播。实验验证，该算法对系统的映射性能有一定的提高。

参 考 文 献

[1] Jean-Mary YR, Kabuka MR. ASMOV Results for OAEI 2007 [C]// International Semantic Web Conference (ISWC). Busan, Korean, 2007: 141-151

[2] Introduction of Falcon-AO[EB/OL]. http://xobjects.seu.edu.cn/project/falcon/matching/index.html. 2006

[3] Li Y, Zhong Q, Li J, et al. Result of Ontology Alignment with Rimom at OAEI 2007 [C]// OAEI. 2007: 227-235

[4] Melnik S, Garcia-Molina H, Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm[C]// The 18th International Conference on Data Engineering. San Jose, California, USA, February 26th-March 1st, 2002: 112-126

[5] Hu W, Jian NS, Qum YZ, et al. GMO: A Graph Matching for Ontologies[C]// K-CAP Workshop on Integrating Ontologies. Banff, Alberta, Canada, October 2005: 1-8

[6] Jiang J J, Conrath D W. Semantic Similarity Based on Corpus Statitics and Lexical Taxonomy[C]// Proceedings of International Conference Reaserch on Computational Linguistic. Tai-Wan, 1997: 1-15

[7] Ortega JM, Rheinboldt WC. Iterative Solution of Nonliner Equations in Several Variables[M]. New York: Academic Press, 1970

[8] Zhang ZW, Xu DZ, Zhang T. Ontology Mapping Based on Conditional Information Quantity[C]// Proceedings of ICNSC 2008. Sanya, 2008: 587-591

(上接第 173 页)

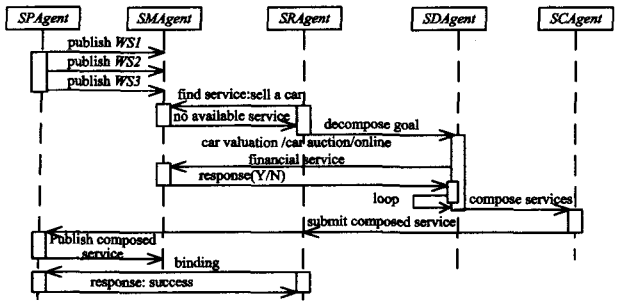

图 4　卖车服务协作顺序图

务、在线支付服务组装起来形成组合服务(3 个服务组装是个简单的顺序结构)，即汽车估价服务给卖主估计汽车可能卖出的价格；这一信息提供给拍卖行后，拍卖行就在此基础上给定拍卖价格；拍卖完成后，买卖双方在线完成支付，最后完成了这个交易。

结束语　本文通过研究 Agent 技术与 Web 技术相互间的关系和相似性，将 Agent 引入到 Web 服务的经典模型 SOA 中，提出了一种基于多 Agent 的语义 Web 服务组合框架。在该框架中，使用 Agent 来封装 SOA 中 3 个不同的角色，使得角色之间的交互表现为 Agent 的协商和协作，同时可以使用多 Agent 协作来搜索发现服务资源，发现用户意图，分解用户目标及动态规划组合服务。有了以多 Agent 为内核的 Web 服务组合管理模块，语义 Web 服务能够很好地适应当今极其开放动态的分布式计算环境，给用户提供高质量稳健的服务，同时我们认为有 Agent 封装用户意图，就可以基于用户偏好，更有针对性地提供相应的服务。下一步需要解决的是基于 Agent 的目标分解及目标-服务匹配等问题。

参 考 文 献

[1] McIlraith S, Son T C. Semantic Web Services[J]. IEEE Intelligent Systems, 2001(Special Issue on the Semantic Web)

[2] Handler J. Agents and the Semantic Web[J]. IEEE Intelligent System, 2001, Mar/Apr: 30-37

[3] 姚莉，张维明. 智能协作信息技术[M]. 北京：电子工业出版社，2002

[4] 叶云，李舟军，李梦君，等. 整合 Agent 与语义 Web 服务[J]. 计算机科学，2007，34(5)：144-146

[5] Chen H, Finin T, Joshi A, et al. Intelligent Agent meets the Semantic Web in Smart Space [J]. IEEE Internet Computing, 2004, Nov/Dec: 69-79

[6] Buhler P, Vidal J M. Semantic Web Services as Agent Behaviors[M]. Agentcities: Challenges in Open Agent Environments, LNCS/LNAI, Berlin: Springer-Verlag, 2003

[7] 任磊，李玉忱，李璟. 基于多 Agent 的 Web 服务动态合成的研究[J]. 计算机应用，2005(4)：802-804

[8] 姚世军. 基于 Agent 和 QoS 动态选择服务的方法[J]. 计算机应用，2005(12)：345-346

[9] 张尧学，方存好. 主动服务—概念、结构与实现[M]. 北京：科学出版社，2005

[10] Payne T R. Web Services from an Agent Perspective[J]. IEEE Intelligent System, 2008, Mar/Apr: 12-14