

语义**Web**服务 及其合成方法的研究

黄雪娟 © 著

Research on Semantic
Web Service and
Its Composition Method



中国地质大学出版社有限责任公司
ZHONGGUO DIZHI DAXUE CHUBANSHE YOUXIAN ZEREN GONGSI

语义 Web 服务及其 合成方法的研究

黄雪娟 著



中國地質大學出版社有限公司
ZHONGGUO DIZHI DAXUE CHUBANSHE YOUXIAN ZEREN GONGSI

图书在版编目(CIP)数据

语义 Web 服务及其合成方法的研究/黄雪娟著. —武汉:中国地质大学出版社有限责任公司, 2013. 8

ISBN 978-7-5625-3247-7



中国版本图书馆 CIP 数据核字(2013)第 214670 号

语义 Web 服务及其合成方法的研究

黄雪娟 著

责任编辑:段连秀

策划编辑:段连秀 张华

责任校对:戴莹

出版发行:中国地质大学出版社有限责任公司

邮政编码:430074

(武汉市洪山区鲁磨路 388 号)

电 话:(027)67883511

传真:67883580

* E-mail: cbb@cug.edu.cn

经 销:全国新华书店

http://www.cugp.cug.edu.cn

开本:880 毫米×1 230 毫米 1/32

字数:210 千字 印张:6.25

版次:2013 年 9 月第 1 版

印次:2013 年 9 月第 1 次印刷

印刷:武汉教文印刷厂

印数:1—1 000 册

ISBN 978-7-5625-3247-7

定价:58.00 元

如有印装质量问题请与印刷厂联系调换

摘要

作为一种新兴的 Web 网络上的分布式计算模式,Web 服务提供了一种集成 Web 上数据与软件模块的机制,有效地解决了异构平台上共享数据与集成软件应用系统的互操作问题。语义 Web 服务是将 Web 服务概念与语义 Web 技术相结合而产生的崭新的服务计算实现形式,旨在解决基于 XML 的 Web 服务交互过程中因缺乏语义描述导致交互双方无法正确地相互理解的问题。语义 Web 服务通过共享与重用 Web 服务的语义知识,使得计算机程序能够理解 Web 服务交互过程中传递消息的内容,从而促进发现、选择、调用与合成 Web 服务的自动化实现。

Web 服务合成是指当单一的 Web 服务无法支持业务应用实现时,将多个 Web 服务有机地组合在一起来实现复杂的应用功能。服务合成中的发现、选择与协调各组成 Web 服务的实现过程均是对用户“透明的”。虽然当前国内外关于 Web 服务的研究工作已经提供了一些标准与规范,如“Web 服务描述语言(WSDL)”、“简单对象接入协议(SOAP)”,以及“统一描述、发现和集成(UDDI)”等,然而在开放的、分布式的网络环境下进行语义 Web 服务的描述、发现、选择与合成尚存在一些现实问题需要解决。

1. 存在的现实问题

(1) 如何找到满足用户需求的语义 Web 服务。解决这一问题实际上包括两方面的内容:第一,在较好地隐藏 Web 服务的实现技术细节的基础上,服务提供者如何清楚地描述其提供的 Web 服务实现的软件应用功能;第二,仅仅明白自己需要什么而不知道到哪里、使用什么方法发现 Web 服务信息的用户,如何能够顺利地找到满足其需求的 Web 服务来实现软件功能运行。针对上述 Web 服务描述需求,必须采取有效的方法实现 Web 服务的语义化描述。

(2) 如何实现语义 Web 服务描述与用户需求之间的正确匹配。现有的 Web 服务描述方法是通过描述调用 Web 服务过程中用户与 Web 服务之间进行的消息传递来说明 Web 服务的接口,而用户需求是从用户的角度阐述用户希望得到什么样的软件应用功能。语义 Web 服务通过增强 Web 服务描述的语义表述能力,减少了服务提供者与用户之间的理解误差,从而实现服务信息的机器可理解,然而当网络上候选 Web 服务较多时,语义信息将大大降低 Web 服务匹配的的实现效率。

(3) 如何灵活地、有效地实现语义 Web 服务的合成。现实应用中经常会出现从网络上可用的服务资源中找不到单一的 Web 服务来支持复杂业务流程的情况,此时合成多个 Web 服务来生成一个新的支持更复杂应用功能的 Web 服务是一个可行的、有效的解决方法。然而应对开放的、多变的、不可预期的网络环境,协调各个组成 Web 服务之间的互操作,根据业务流程定义得到

合成服务过程描述说明等方面均使得语义 Web 服务合成成为一个富有挑战性的研究课题。

在讨论了语义 Web 服务合成存在的现实问题之后,本书通过研究分析 Web 服务的描述方法与描述语言、语义化标识,以及现有合成方法等存在的不足,提出了语义 Web 服务社区 SWSC 模型,说明了 SWSC 的概念模型、组成结构、服务语义描述方法、包含的活动等方面,并优化了现有 Web 服务发现与选择算法,提出了 SWSC 框架下语义 Web 服务合成的规则与实现方法。

2. 本书的研究工作

(1)讨论了 Web 服务描述包含的内容、描述方法、现有描述语言等,并阐述了语义 Web 服务是 Web 服务发展的必然趋势。接下来讨论了如何从语义层面上描述 Web 服务,并分析 Web 服务的功能与非功能属性的描述对于服务发现与选择过程的影响,得出了区分 Web 服务之间的相同点与不同点能够帮助提高服务发现与选择效率的结论。

(2)提出了语义 Web 服务社区 SWSC 模型,说明了在 SWSC 下描述语义 Web 服务的方法,并从概念模型、组成结构、包含的活动、实现规则、与其他语义 Web 服务模型相比具有的特性等方面阐述了 SWSC 模型的构造与运行。

(3)通过增强语义约束改进了现有的基于 OWL-S 的服务功能匹配算法,弥补了原始算法在服务查找与选择时存在的一类语义偏差。在此基础之上,本书描述了 SWSC 下查找与选择 Web 服务的基本思想与实现算法,并结合 SWSC 的组成结构说明了

SWSC 下查找与选择服务方法是如何通过缩小 Web 服务搜索范围来提高查找与选择效率。

(4) 在讨论了服务合成面临的现实问题以及现有的服务合成方法的基础上,提出了 SWSC 框架下语义 Web 服务的合成方法,同时介绍了在尽可能满足用户需求的前提条件下,促使 Web 服务资源与用户需求达到相对地供需平衡的协调机制。

3. 本书的创新之处

(1) 提出了语义 Web 服务社区 SWSC 模型,并阐述了其概念模型、组成结构以及服务语义化描述方法。

(2) 通过增强语义约束改进并完善了现有的基于 OWL-S 的服务功能匹配算法。

(3) 提出了 SWSC 框架下的服务合成规则与方法。

(4) 提出了 SWSC 框架下在尽可能满足用户需求的前提条件下,促使 Web 服务资源与用户需求之间达到并保持相对平衡的协调机制。

目 录

第一章 绪 论	(1)
1.1 研究领域概述	(1)
1.1.1 Web 的发展历史与演化趋势	(1)
1.1.2 服务计算与服务架构	(3)
1.1.3 Web 服务	(7)
1.1.4 Web 服务面临的机遇与挑战	(10)
1.2 国内外研究现状	(12)
1.2.1 Web 服务扩展架构与实现平台	(12)
1.2.2 语义 Web 服务	(15)
1.2.3 Web 服务发布、查找与发现	(17)
1.2.4 Web 服务合成	(21)
1.3 研究面临的主要问题	(23)
1.4 本书的主要工作	(25)
1.5 本书的组织形式	(25)
第二章 Web 服务的描述	(27)
2.1 Web 服务描述内容	(27)
2.1.1 用户对服务描述内容的需求	(28)

2.1.2	服务提供者提供的服务描述内容	(30)
2.1.3	服务描述的语义	(32)
2.2	Web 服务的功能描述	(33)
2.2.1	描述方法分类	(34)
2.2.2	Web 服务描述语言 WSDL 的描述方法	(34)
2.2.3	Web 服务本体语言 OWL-S 的描述方法	(40)
2.2.4	显性与隐性相结合的 Web 服务功能的描述方法	(44)
2.3	Web 服务社区 WSC 模型	(46)
2.3.1	Web 服务间的异同	(46)
2.3.2	Web 服务社区 WSC 中描述服务的方法	(47)
2.4	小结	(51)
第三章	语义 Web 服务	(52)
3.1	下一代 Web 服务	(52)
3.1.1	基于 XML 的 Web 服务	(52)
3.1.2	Web 服务发展的趋势	(54)
3.2	语义 Web	(55)
3.2.1	语义 Web 的发展	(55)
3.2.2	本体与语义 Web	(58)
3.2.3	资源描述框架 RDF	(62)
3.2.4	语义 Web 本体语言 OWL	(66)
3.2.5	语义 Web 的规则	(68)
3.3	语义 Web 服务	(72)

3.3.1	Web 服务的语义	(73)
3.3.2	语义 Web 服务的描述	(76)
3.3.3	语义 Web 服务的实现工具	(81)
3.4	语义 Web 服务与语义 Web 服务社区 SWSC 模型 ...	(85)
3.5	小结	(94)
第四章	语义 Web 服务模型	(96)
4.1	语义 Web 服务建模面临的现实问题	(96)
4.2	语义 Web 服务社区 SWSC 模型	(99)
4.2.1	SWSC 概念模型	(99)
4.2.2	SWSC 组成结构	(102)
4.2.3	SWSC 中语义信息的数据结构	(108)
4.2.4	SWSC 中 Web 服务的语义描述	(109)
4.2.5	SWSC 包含的活动	(111)
4.2.6	SWSC 的实现规则	(116)
4.3	语义 Web 服务社区模型的特性	(118)
4.4	小结	(121)
第五章	语义 Web 服务合成	(123)
5.1	Web 服务合成	(123)
5.2	SWSC 下语义 Web 服务的查找与选择	(127)
5.2.1	语义 Web 服务的查找与选择方法	(127)
5.2.2	基于 OWL-S 的 Web 服务功能匹配算法的改进	(132)
5.2.3	SWSC 下语义 Web 服务查找与选择方法	(137)

5.3 SWSC 下语义 Web 服务合成	(143)
5.3.1 语义 Web 服务动态合成过程中存在的问题	(143)
5.3.2 SWSC 下语义 Web 服务的合成规则	(145)
5.3.3 SWSC 下语义 Web 服务的合成过程	(148)
5.3.4 SWSC 下 Web 服务资源与用户需求的协调机制	(153)
5.4 小结	(155)
第六章 实验	(156)
6.1 实验平台	(156)
6.2 实验设计	(157)
6.3 实验分析	(160)
6.4 小结	(167)
第七章 结束语	(169)
7.1 本书的主要研究成果	(169)
7.2 后期工作展望	(171)
参考文献	(172)
ABSTRACT	(184)
致谢	(189)

第一章 绪 论

1.1 研究领域概述

作为一个通用的共享与传递数据的信息空间,Web 网络历经 20 年的迅速发展已经成为人类学习、工作和生活中不可缺少的一部分。服务计算(Service Oriented Computing, SOC)和服务架构(Service Oriented Architecture, SOA)等概念的提出使用户能够通过传递标准化的消息,实现网络异构环境下的软件功能调用与集成。这意味着 Web 网络对于人类用户而言不再仅仅是获取与传递数据的空间,而将成为共享与复用各种各样的软件应用功能的重要媒介。这一变化标志着 Web 世界正走在从数据型网络向服务型网络转变的道路上。

1.1.1 Web 的发展历史与演化趋势

Tim Berners-Lee^[1] 在 1990 年首次提出了万维网(World Wide Web)的概念:使用超文本(Hypertext)和统一资源标识符(URL)技术来创建一个信息关联的网络,通过网络连接来自于世界各地的各种文档资料,从而使用户能够远程访问这些文献数据。最初提出的万维网概念是非常简单的,只是将其设定为一个用于共享文字与图表等数据信息的平台,关注的也仅仅是数据显示格式与对象链接方法。

Tim Berners-Lee 于 1994 年在麻省理工大学创立了万维网联盟(World Wide Web Consortium, 简称 W3C)。W3C 的目标是规范在 Web 网上使用的协议和技术,从而使全球尽可能多的人可以访问 Web 站点并浏览 Web 页面内容。W3C 陆续发布了许多规范文档,包括 CSS、XHTML、XML、XQuery 和 XSLT 等。

在 1998 年, Tim Berners-Lee 提出所有 W3C 的技术工作必须围绕“持续演化”与“可操作性”这两点展开,并强调数据与语言的演化是万维网发展的两大重点问题^[2]。分析 Tim Berners-Lee 所讨论的内容,不难发现其所言的“演化”更接近于是指网络的可扩展性与灵活应用性。另一个关于 Web 演化的重要讨论是在 2006 年 WWW 国际会议的小组讨论会上由五位著名的 Web 研究学者 Ron Brachman、Dan Connolly 和 Rohit Khare 等给出的。这次讨论会的描述文档^[3]中说明了一切关于 Web 的发展都是源于人类最基本的要求。例如,人们希望网上信息结构清晰、意义明确,即对应地要求 Web 数据内容必须使用结构化的表述方法,并且具有良好定义的语义,因此开发用于描述 Web 内容含义的语言成为 Web 演化的核心问题之一,推动着 Web 网络朝着语义 Web 的方面发展。将网络与人类社会相关联,2007 年的语义技术大会给出了如下的 Web 发展阶段说明^[4],预示了 Internet 的演变趋势。

◎ 1990—2000 年, Web1.0 (Web, 万维网, 作用: 连接数据), 主要包括网页搜索引擎、网站、数据库、文件服务器等。

◎ 2000—2010 年, Web2.0 (Social Web, 社会网, 作用: 连接知识), 引入了社区、RSS、Wiki、社会化书签、社会化网络等概念。

◎ 2005—2020 年, Web3.0 (Semantic Web, 语义网, 作用: 连接知识), 由本体、语义查询、人工智能、智能代理、知识结点、语义

知识管理等构成。

◎ 2015—2030 年, Web4.0 (Ubiquitous Web, 无处不在的网络, 作用: 连接情报), 具体内容不定, 其含义关键在于 Web 网能够实现任何时候、任何位置提供用户所需。

虽然没有明显迹象表明从 Web1.0 到 Web2.0 的转变已经完成, 但有一点可以肯定的是: Web 的发展最初是由人类对共享传递信息的需求所驱动的, 其以后的演变过程必定也将持续地一步步地模拟人类社会发展而进行^[5]。正如同人类社会从商品型经济向服务型经济的市场转变一样, Web 世界也正在经历着从基于信息的“数据仓库”向基于服务的“虚拟社会”的转变。这一转变源于人类对 Web 网络的期待不再仅仅局限于共享传递数据, 而是希望能够从 Web 上获得对其学习、工作、生活有所帮助的软件服务。E-business, E-learning, E-marketplace 及 E-Science 等基于 Web 网络的研究热点的不断涌现很好地体现了这一转变。

1.1.2 服务计算与服务架构

复杂性与兼容性是信息技术经常需要面对的两大问题。无论是构建新的应用系统、替换现有应用系统, 还是处理各种维护与改进, 都必须应对现实应用需求本身所固有限制情况以及不同软件设计模式的局限性所带来的各种复杂状况。另外, 随着软件业的迅速发展, 各企业组织内部为繁杂的业务需求开发了大大小小无数的程序。虽然某些程序可能实现的是相同的业务功能, 但它们在开发平台、编程语言、数据结构等方面存在或多或少的差异, 而在万维网环境下这些差异使得这类程序在企业组织协同工作的时候变成了“鸡肋程序”。降低复杂性, 克服兼容问题, 是在 Web 环境下发展软件业不得不面临的重大挑战。

服务计算 SOC(下文中出现的 SOC 均是指服务计算)是一种使用服务(Service)作为基本组成部分的计算模式,其目标是希望能够在异构环境下实现快速的、低成本的、易于合成的分布式应用系统的软件开发^[6]。服务计算模式是对面向对象、面向过程、面向消息以及面向数据库等软件开发方式的补充。基于服务的计算之所以能够得到广大软件厂商的接受,是因为其具有以下优点^[7]:

- **重用**:通过对现有程序的封装使其具有能够被其他应用系统复用。

- **效率**:通过组合现有多个服务能够快速创建实现新的复杂应用功能的软件系统。

- **与技术的松耦合**:软件建模独立于底层实现环境与技术细节。

- **软件开发职责的划分**:使业务人员与技术人员能够分别只关注于业务问题和技术问题,两组人员之间通过服务契约进行协同工作。

面向服务与面向对象都具有重用与松耦合特点,但两者之间存在以下显著区别:

- ◆ **定义方式**:服务是通过描述与其他服务交换的消息(Messages)来定义的,而对象的定义继承了传统的软件程序定义方法,是由方法的型构(Method Signature)决定的,即程序名以及调用参数等。

- ◆ **抽象层次**:服务定义所在的抽象层次高于对象定义所在的抽象层次,因为这样才可以将服务定义映射到某种面向过程的语言(如 COBOL)、消息排队系统(如 JMS)或面向对象系统(如 J2EE 或 .NET)上。

◆**调用方式**:服务是通过遵循预先定义好的服务契约进行消息交换来实现调用的,而对象是通过程序名进行调用的(此差异来源于定义方式的不同)。

◆**粒度(Granularity)**:两者虽都能定义粗粒度的接口,但服务主要用来解决应用间的互操作问题,通过合成能够快速实现新的应用,而对象是为实现具体的业务逻辑创建组件而设计的。

◆**效率**:服务计算虽然减少了需要部署的代码量,减轻了管理、维护和支持方面的负担,但服务的单次调用因需要进行执行环境的映射和处理 XML 文档等工作,反而会消耗更多的计算资源。

简而言之,服务计算更多关注于解决应用间的互操作性问题以及如何有机地集成现有的各种业务功能模块来实现复杂的应用功能。

服务计算必须依赖于服务架构 SOA(下文出现的 SOA 均是指服务架构)形式。基于服务的架构为软件系统开发与集成提供了一种将各种业务逻辑功能的实现进程组装成服务的方法,这些服务通过标准化的接口进行接入与交互,从而实现跨部门、跨组织、异构环境下的协同工作。从面向应用的角度,基本的服务架构定义了三种基本角色,角色之间通过消息交互来实现对服务的发布、查找与调用^[8],如图 1-1 所示。

基于上述服务架构来说明“服务”(Service)概念。“服务”是指在某一具体软件执行环境下,程序支持的业务功能的实现,以及服务运行时用于传递消息的接口的正式说明文档。服务的用户不必清楚服务运行的底层实现环境与技术细节,只需根据服务提供者在服务描述文档中说明的服务接口形式进行服务的绑定,然后发出服务调用请求,就能实现服务支持的业务功能逻辑。服

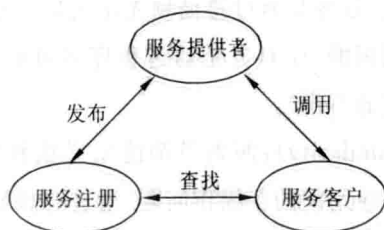


图 1-1 基本的服务架构

务定义时所采用的“接口与实现分离”的思想,在 J2EE、COM 或 DCE 中已有所体现,但不同于过程调用或参数传递等形式的分离,SOA 主要是通过文本文件来实现功能与执行环境的分离,故能获得更“松散的”分离与更强的互操作性。

从软件应用逻辑、运行处理、实现技术三方面对 SOA 与传统的客户/服务(Client/Server,C/S)端、分布式网络架构(Distributed Internet Architecture)分别进行比较^[9],可以得到以下结论:

(1)应用逻辑:C/S 与分布式网络架构一般将大部分的应用逻辑放置在服务(S)端,而 SOA 将应用逻辑划分成独立的功能单元,松散地分布在服务提供者的物理实体上。

(2)运行处理:C/S 中的服务(S)端的工作站承担着应用系统大部分的运行处理工作;分布式网络架构以构件(Component)为单位实现应用功能的运行处理,而构件间的通信需要部署适当的通信协议来实现;SOA 中服务的运行是通过传递消息实现,所以服务的调用涉及序列化、传输、抽取与解析等与消息相关的处理操作。

(3)实现技术:C/S 的实现技术依赖于编程语言、执行环境,实现互操作的难度较大;分布式网络架构的实现技术包括构件、服务端脚本以及一些初始的 Web 技术,如 HTML 和 HTTP 等,