

中央民族大学“985工程”中国少数民族语言文化教育与边疆史地研究创新基地文库
国家语言资源监测与研究中心少数民族语言分中心计算语言学系列丛书

主编 ○ 戴庆厦 赵小兵

语义 Web 服务组合技术

邱莉榕 孙媛 赵小兵 / 著



中央民族大学出版社
China Minzu University Press

中央民族大学“985 工程”中国少数民族语言文化教育与边疆史地研究创新基地文库
国家语言资源监测与研究中心少数民族语言分中心计算语言学系列丛书

主编 © 戴庆厦 赵小兵

语义 Web 服务组合技术

邱莉榕 孙媛 赵小兵 著

中央民族大学出版社

图书在版编目 (CIP) 数据

语义 Web 服务组合技术/邱莉榕等著. —北京: 中央民族大学出版社, 2009. 7

ISBN 978 - 7 - 81108 - 701 - 7

I. 语… II. 邱… III. 语义网络—技术 IV. TP18

中国版本图书馆 CIP 数据核字(2009)第 103594 号

语义 Web 服务组合技术

作 者 邱莉榕 孙 媛 赵小兵

责任编辑 戴苏芽

封面设计 乌日恒

出 版 者 中央民族大学出版社

北京市海淀区中关村南大街 27 号 邮编:100081

电话:68472815(发行部) 传真:68932751(发行部)

68932218(总编室) 68932447(办公室)

发 行 者 全国各地新华书店

印 刷 者 北京九州迅驰传媒文化有限公司

开 本 880×1230(毫米) 1/16 印张:11.5

字 数 260 千字

版 次 2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷

书 号 ISBN 978 - 7 - 81108 - 701 - 7

定 价 28.00 元

版权所有 翻印必究

内 容 简 介

作为一种新兴的、面向 Internet 的分布式计算模式,面向服务的计算为构造松耦合、跨组织的集成应用提供了更好的使能技术。Web 服务作为当前最主要的一种服务实现技术,目的是为 Internet 上跨越不同地域、不同行业的应用提供更强大的互操作能力。单个的 Web 服务往往无法满足实际的用户需求,Web 服务组合就是研究如何协调和组织多个 Web 服务来构造新的 Web 服务或应用系统的技术,目标是提高服务组件及基本服务的可重用性和利用率,减少新应用的开发时间和费用。

异构系统中运行在不同平台之上的 Web 服务可能是以不同的方式创建、用不同程序语言实现、由不同供应商提供的,服务的请求需要根据特定的应用背景和需求进行合理的服务组合,这些都增加了服务组合问题的难度。

语义 Web 服务研究的主要任务就是对 Web 服务进行标记,使 Web 服务成为计算机可理解的、用户透明的和主体易处理的实体。具体来说,它包括自动的 Web 服务发现、执行、组合及互操作和执行监控。

本书的第 1-3 章由孙媛撰写,第 4-5 章由邱莉榕撰写,第 6 章由赵小兵撰写。主要内容如下:

1. 第一章详细介绍了计算环境的演变过程以及面向服务的计算环境和 Web 服务组合的研究历史。

2. 第二章讨论了语义 Web 服务研究的基本概念,详细介绍了语义 Web、Web 服务的发展根源,研究背景。

3. 第三章描述了服务应用的总过程,详细介绍了服务发现和服务匹配的主要技术细节和研究现状。

4. 第四章是本书的重点。该章节讨论了语义 Web 服务组合的相关研究技术和研究现状。对研究现状的介绍主要包括当前的 Web 服务组合方法和服务组合项目的介绍。针对提出的问题,从以下方面进行了探讨:

- 从信息类 Web 服务入手,借鉴基于义原及基于本体的词汇相似度的计算方法,获得服务参数相似度,从而实现基于参数相似度的信息类 Web 服务的组合算法。

- 介绍了基于动态描述逻辑的动作类 Web 服务组合。在深入分析动态描述逻辑的推理和系统的基础上,将改变世界状态的 Web 服务看成是一个原子动作或者复杂动作,利用动作理论来研究动作类 Web 服务组合问题。

- 介绍了既包含信息类 Web 服务又包含动作类 Web 服务的混合类 Web 服务组合问题,将物理服务注册为主体虚拟服务,提出了基于领域模板的半自动化组合方法。

- 讨论了服务组合中的语境信息类别,提出了基于 OWL-S 扩展的语境信息的语义模型 OWL-SC,模型从用户因素、服务因素以及所处环境等三个角度刻画了隐含在服务组合过程中并且不断变化的语境信息。介绍了语境信息优化的组合方法,以及语境信息的推理模型。

5. 第五章介绍了语义 Web 服务组合平台 SWSBroker 的实现,探讨该工具的结构、

核心组件的功能以及原型系统的使用细节。

6. 第六章描述了一个支持多民族语言文字的网络交流平台的应用实例。

本书受中央民族大学“985 工程”中国少数民族语言文化教育与边疆史地研究基地、国家语言资源监测与研究中心少数民族语言分中心、国家自然科学基金《基于动态流通语料库的汉语基本词汇特征曲线跟踪及自动提取方法研究》（项目号：60663008）、教育部科研课题《蒙古文编码转换系统》（项目号：MZ115-72）的资助。

在成书的过程中，我们得到了很多人的帮助，借此机会，特别在此感谢中国科学院计算技术研究所史忠植研究员；感谢中国科学院自动化研究所王飞跃研究员；感谢中央民族大学信息工程学院的杨国胜院长、曹永存副院长；感谢中央民族大学现代教育技术部肖波教授；同时感谢出版社领导与编辑的鼓励与支持。

由于时间仓促，水平有限，书中难免有所疏漏之处，在此敬请广大读者批评指正。

目 录

第 1 章 绪论	(1)
1.1 引言	(1)
1.2 计算环境的演变	(1)
1.2.1 计算环境的演变	(1)
1.2.2 分布式计算	(3)
1.2.3 面向服务的计算环境	(4)
1.3 Web 服务组合	(6)
1.4 本书主要内容	(7)
第 2 章 语义 Web 服务	(9)
2.1 Web 服务基础	(9)
2.1.1 Web 服务简介	(9)
2.1.2 面向服务架构 SOA	(10)
2.1.3 Web 服务特性	(12)
2.1.4 Web 服务规范	(14)
2.2 语义 Web	(22)
2.2.1 语义的引入.....	(22)
2.2.2 语义 Web 层次模型	(24)
2.2.3 本体.....	(29)
2.2.4 语义 Web 的基础软件	(33)
2.2.5 语义 Web 的应用	(37)
2.3 语义 Web 服务	(42)
2.3.1 Web 服务的语义描述	(42)
2.3.2 服务描述语言的分析与比较.....	(44)
2.3.3 OWL-S	(46)
2.3.4 OWL-S 与 WSDL 的关系	(49)
2.3.5 语义 Web 服务的相关研究	(50)
2.4 小结.....	(51)
第 3 章 Web 服务的发现与匹配	(53)
3.1 Web 服务应用的构造过程	(53)
3.2 Web 服务发现	(54)
3.3 Web 服务匹配	(55)
3.3.1 服务匹配流程.....	(56)
3.3.2 语义 Web 服务匹配	(57)

3.4 基于 BPEL4WS 的 Web 服务组合	(60)
3.5 小结	(61)
第 4 章 语义 Web 服务组合	(62)
4.1 概述	(62)
4.1.1 Web 服务组合方法	(62)
4.1.2 服务组合项目	(64)
4.1.3 主体技术在语义 Web 服务中的应用	(69)
4.1.4 语义 Web 服务组合策略	(75)
4.2 信息类 Web 服务组合	(76)
4.2.1 基本思想	(76)
4.2.2 信息类 Web 服务描述	(77)
4.2.3 词汇相似度	(78)
4.2.4 基于服务参数相似度的组合	(80)
4.2.5 核心算法	(84)
4.2.6 小结	(87)
4.3 动作类 Web 服务组合	(88)
4.3.1 基本思想	(88)
4.3.2 动态描述逻辑及其推理	(89)
4.3.3 语义 Web 服务的逻辑基础	(95)
4.3.4 基于 DDL 的服务组合算法	(100)
4.3.5 相关工作	(107)
4.3.6 小结	(108)
4.4 混合类 Web 服务的组合策略	(109)
4.4.1 半自动组合策略	(109)
4.4.2 基于主体的服务虚拟化	(114)
4.4.3 物理服务的注册和注销	(122)
4.4.4 相关工作	(126)
4.4.5 小结	(127)
4.5 组合过程中的语境信息优化	(127)
4.5.1 相关概念	(127)
4.5.2 组合过程中的语境优化模型	(133)
4.5.3 语境信息的推理	(137)
4.5.4 QoS 驱动的服务组合	(140)
4.5.5 相关工作	(146)
4.5.6 小结	(147)
第 5 章 服务组合工具 SWSBroker 的设计与实现	(148)
5.1 服务组合工具 SWSBroker 总体结构	(148)
5.1.1 MAGE 平台	(148)

5.1.2 KMSphere 平台	(150)
5.1.3 SWSBroker 的总体结构	(151)
5.2 SWSBroker 的推理模块	(153)
5.2.1 规划器	(153)
5.2.2 DDL 推理机	(154)
5.3 服务组合工具 SWSBroker 实现	(156)
5.3.1 服务的语义标注	(156)
5.3.2 语境信息管理器	(156)
5.3.3 可视化的组合模板编辑模块	(158)
5.3.4 SWSBroker 的使用	(159)
5.4 小结	(161)
第 6 章 应用实例：多民族语言语料库网络管理平台的设计	(162)
6.1 问题的提出	(162)
6.2 主要思路	(163)
6.2.1 解决方案	(163)
6.2.2 主要实现目标	(164)
6.3 民族语言生活状况语料库网络平台建设	(165)
6.3.1 系统架构	(165)
6.3.2 系统子模块	(165)
6.3.3 系统设计	(166)
6.4 基于网络的少数民族语言资源交流平台	(166)
6.4.1 基于动态网站的平台架构	(166)
6.4.2 平台功能及特点	(167)
6.4.3 平台技术特色	(167)
参考文献	(168)

第 1 章 绪 论

1.1 引 言

任何新兴的技术，检验其优劣的标准无疑是应用，而推动其进步和发展乃至普及的动力则是市场需求。面向服务的计算环境 SOA 正是在企业信息系统遭遇应用集成“瓶颈”的背景下出现的。

面向服务的计算环境并不是一个新概念，有人就将 CORBA 和 DCOM 等组件模型看成 SOA 架构的前身。早在 1996 年，Gartner Group 就已经提出了 SOA 的预言，不过那个时候仅仅是一个“预言”，当时的软件发展水平和信息化程度还不足以支撑这样的概念走进实质性应用阶段。到了近一两年，面向服务计算环境的技术实现手段渐渐成熟了，在 IBM 等软件巨头的极力推动下，慢慢风行起来。

一方面，由于市场的变化日趋迅速，需要企业及时做出响应，因此现代企业的业务流程不可能是一成不变的。而基于传统软件架构模式构建的软件信息系统是依据事先的需求分析和系统架构设计来完成的，这种需求反映的往往只是企业过去某一时间点的业务流程。等到系统设计开发完成，企业的需求已经发生了变化，这也就是为什么结构越复杂的系统，实施成功率越低的原因，因为结构越复杂，实施周期就越长，系统架构与业务流程的偏差就越大，失败的几率自然就增加。企业需要容易实施，并可以方便地改变业务流程的系统。

另一方面，随着互联网技术的成熟，无论人与人之间，还是企业与企业之间的沟通变得更加容易。但是这还不够，企业的管理者们希望通过有效的手段帮助企业的应用系统加强交互，以改变传统的商务谈判和交易模式，实现真正意义上的电子商务。

这要求企业的应用系统间不但要突破部门间的界限，同时必须打通企业间的界限，提高异构系统间以及企业间应用的互操作性。

1.2 计算环境的演变

1.2.1 计算环境的演变

计算环境由一组计算机、软件平台和相互联通的网络组成，这个环境能够处理和交换数字信息，允许外界访问其内部的信息资源。不同的计算环境有不同的计算风格和编程模型，由一些特定于该计算环境的技术来支撑。如何在一个计算环境中分割和部署计算能力、数据资源，如何让各个部分相互通信和协作，如何在概念上对问题域进行建模，然后映射到该计算环境，都会受到计算环境的影响和制约。因此，了解一下计算环

境的历史，会帮助我们理解面向服务的计算环境是如何演变而来的。

计算环境的演变经历了若干个阶段，在早期的主机时代，绝大多数的计算功能和系统的组成部分，都包括在一台机器里。在 20 世纪 80 年代，随着 PC 的繁荣，计算环境发生很大的变化。通过局域网相互连接的计算设备构成客户/服务器计算环境，计算资源和数据资源被适当地分割，客户和服务器通过网络协议、远程调用或消息等方式来相互协作，完成计算。

计算环境的演变经历了若干个阶段：

1. 主机时代。其特点是，几乎所有的计算功能和系统的组成部分，都包括在一台机器里，很多人分时的共同使用一台计算机。各个主机之间的数据、功能很难共享和相互调用。

2. 客户/服务器计算环境。在 20 世纪 80 年代，随着 PC 的繁荣，计算环境发生很大的变化。通过局域网相互连接的计算设备构成客户/服务器计算环境，计算资源和数据资源被适当地分割，客户和服务器通过网络协议、远程调用或消息等方式来相互协作，完成计算。

3. 基于多层架构和中间件的分布式计算环境。为了满足更高的可伸缩性需求，多层架构出现，数据和计算功能分布多样化，分布在多台计算机，与企业中原来已经存在的计算环境，尤其是主机及其遗留系统之间的集成也变得越来越重要。中间件迅速发展，开始出现分布式对象、组件和接口等概念，用于在计算环境中更好地分割运算逻辑和数据资源。计算环境中不同部分之间的交互，也从原有相对低层的网络协议、远程调用和消息机制的基础上，发展到支持分布式对象、组件和接口之间的交互，这种交互在名字服务（Naming Service）等的支持下，通常是位置透明的。但由于缺乏普遍的标准支持，很难做到技术透明，系统是紧耦合的

4. 面向服务的计算环境。也就是基于标准、开放的互联网技术，以服务为中心的计算环境。这是一个以服务为基本单位和抽象手段的世界。随着互联网（Internet）的发展，开放和标准的网络协议被普遍支持，所有底层计算平台都开始支持这些标准和协议，这导致一个计算环境内部和各个计算环境之间交互的藩篱被打破。数据和功能的表示与交互在 XML、WEB 服务（Web Service）技术与标准的基础上，保证了通用性和最大的交互能力，这使得计算环境发展到一个全新的阶段—基于标准、开放的互联网技术的计算环境。在这样的计算环境中，各个部分可以采用异构的底层技术，它们使用 XML 来描述和表示自己的数据和功能，采用开放的网络协议（如 HTTP）来握手，在此之上，基于 Web 服务来互操作和交换数据。在这里，一个很重要的新概念是“服务”，它是一个自包含的功能，使用者通过明确定义的接口（契约）来与一个服务交互，这个接口的描述基于 WSDL（Web Service Description Language）这样的开放标准。对象和组件重在表示一个事物本身的组成部分和相互关联（也就是 WHAT “THINGS” ARE 的问题），而服务则表示一个事物做什么（也就是 WHAT “THINGS” DO 的问题）。Web 服务是实现服务的技术手段，就如同各种编程语言中的对象是实现对象的技术手段，J2EE 中的 EJB 是实现组件的技术手段一样。

1.2.2 分布式计算

分布式计算是近年提出的一种新的计算方式。所谓分布式计算就是在两个或多个软件互相共享信息，这些软件既可以在同一台计算机上运行，也可以在通过网络连接起来的多台计算机上运行。分布式计算比起其他算法具有以下几个优点：

1. 稀有资源可以共享。
2. 通过分布式计算可以在多台计算机上平衡计算负载。
3. 可以把程序放在最适合运行它的计算机上。

其中，共享稀有资源和平衡负载是计算机分布式计算的核心思想之一。

分布式计算与人类由于现代人类各个课题学科繁多，涉及面广，而分类又细。而当今的每个学科似乎都需要进行大量的计算。天文学研究组织需要计算机来分析太空脉冲(pulse)，星位移动；生物学家需要计算机来模拟蛋白质的折叠(protein folding)过程；药学家想要研制克服艾滋病(AIDS)或非典(SARS)的药物；数学家想计算最大的质数和圆周率的更精确值；经济学家要用计算机分析计算在几万种因素考虑下某个企业/城市/国家的发展方向从而宏观调控。由此可见，人类未来的科学，时时刻刻离不开计算。而分布式计算(Distributed Computing)，以其独特的优点——便宜、高效而越来越受到社会的关注。

分布式计算包含集群计算和网格计算，两者之间的不同之处在于在集群中资源是由单个实体自治管理的，而在网格中，资源在地理上是分布的，并且它采用多自治方式管理。集群和网格不同的另外一个关键在于它们的调度策略方式不同。在集群系统中，调度策略集中于提高整个系统的性能和可用性。而在网格中，调度集中于以满足用户需求为基础提高特定应用的性能。因此根据它们不同的构建，就提出了网格计算和集群计算。而网格受到大多数人的欢迎，它相对于集群计算就更“醒目”些了。网格技术是基于分布式的，但是比传统的分布式(DCOM, CORBA)要复杂得多。当今的网格基于Web服务，有大量的服务去实现资源管理、安全、协同合作、性能分析、容错等等大量的复杂工作。

分布式计算是利用互联网上计算机的CPU闲置处理能力来解决大型计算问题的一种计算科学。分布式计算工作过程如下：

首先，要有一个需要非常巨大的计算能力才能解决的问题。这类问题一般是跨学科的、极富挑战性的、人类亟待解决的科研课题。其中较为著名的是：

1. 解决较为复杂的数学问题，例如：GIMPS（寻找最大的梅森素数）。
2. 研究寻找最为安全的密码系统，例如：RC-72（密码破解）。
3. 生物病理研究，例如：Folding@home（研究蛋白质折叠，误解，聚合及由此引起的相关疾病）。
4. 各种各样疾病的药物研究，例如：United Devices（寻找对抗癌症的有效的药物）。
5. 信号处理，例如：SETI@Home（在家寻找地外文明）。

从这些实际的例子可以看出，这些项目都很庞大，需要惊人的计算量，仅仅由单个

的电脑或是个人在一个能让人接受的时间内完成计算是绝不可能的。在以前，这些问题都应该由超级计算机来解决。但是超级计算机的造价和维护非常的昂贵，这不是一个普通的科研组织所能承受的。随着科学的发展，一种廉价的、高效的、维护方便的计算方法应运而生——分布式计算！

随着计算机的普及，个人电脑开始进入千家万户。与之伴随产生的是电脑的利用问题。越来越多的电脑处于闲置状态，即使在开机状态下 CPU 的潜力也远远没有被完全利用。一台家用的计算机将大多数的时间花费在“等待”上面。即便是使用者实际使用他们的计算机时，处理器依然是不计其数的等待（等待输入，但实际上并没有做什么）。互联网的出现，使得连接调用所有这些拥有限制计算资源的计算机系统成为现实。那么，一些本身非常复杂的但是却很适合于划分为大量的更小的计算片断的问题被提出来，然后由某个研究机构通过大量艰辛的工作开发出计算用服务端和客户端。服务端负责将计算问题分成许多小的计算部分，然后把这部分分配给许多联网参与计算的计算机进行并行处理，最后将这些计算结果综合起来得到最终的结果，由此网络计算就应运而生。

当然，通过网络实现一个大型的计算任务，看起来也似乎很原始、很困难，但是随着参与者和参与计算的计算机的数量的不断增加，计算计划变得非常迅速，而且实践证明这是可行的。目前一些较大的分布式计算项目的处理能力已经可以达到甚而超过目前世界上速度最快的巨型计算机。

您可以选择参加某些项目来捐赠自己的 CPU 内核处理时间，您将发现您所提供的 CPU 内核处理时间将出现在项目的贡献统计中。您可以和其他的参与者竞争贡献时间的排名，您也可以加入一个已经存在的计算团体或者自己组建一个计算小组。这些方法很利于调动参与者的热情。随着民间的组队逐渐增多，许多大型组织（例如公司、学校和各种各样的网站）也开始了组建自己的战队。同时，也形成了大量的以分布式计算技术和项目讨论为主题的社区，这些社区多数是翻译制作分布式计算项目的使用教程及发布相关技术性文章，并提供必要的技术支持。

1.2.3 面向服务的计算环境

计算环境由计算机、相关的软件平台以及连通这些计算机和相关软件平台网络组成。不同的计算环境有不同的计算风格和编程模型，由一些特定于该计算环境的技术来支撑。

面向服务的计算环境（SOA）已经成为企业 IT 系统实施的新时尚，无论是对旧有信息系统得改造，还是对企业新 IT 架构的设计，面向服务的体系结构都往往成为首选方案。

面向服务的体系结构（SOA）是一种由相互独立，分布和协作的服务组件构成系统的体系结构。这种服务可以分布在组织的物理边界和安全域内外，可以存在于不同的平台上，也可以由不同的编程语言实现。业界给出了如下的定义：

1. IBM 认为：“SOA 一直主要是关于如何正确地进行构建，如何创建一种体系结构蓝图，该体系结构蓝图允许进行可重用的构建，允许以更加松散耦合的方式工作，

SOA 是真正支持使用可重用的组件或服务装配业务流程的体系结构，这些组件或服务独立于应用程序和它们运行的平台。”

2. Service-architecture.com 将 SOA 定义为：“SOA 本质上是服务的集合，这些服务彼此间相互通信，这种通信可能是简单的数据传送，或者是两个或更多的服务协调进行某些活动。服务间需要某些方法进行连接。”

3. 学术界认为“SOA 是设计和构建松散耦合软件的解决方案，能够以程序化的，可访问软件服务的形式公开业务功能，以使其他应用程序可以通过已发布和可发现的接口来使用这些服务。”

SOA 的关键是服务所实现的功能通过使用标准的接口描述语言暴露给外部，实现细节则被隐藏在服务内部，外部应用通过其所暴露的接口来调用其功能。Web 服务是一种实现 SOA 的方式。

面向服务是一种创建分布式系统的方法。在它最抽象的层面，面向服务作为一个服务提供程序，包含了一切——从大型机应用程序到打印机到码头工作人员到隔夜交货公司。服务提供程序通过接口公开了功能。面向服务的体系结构与这些功能和接口进行了映射，这样它们就可以编制到流程里。这种服务模型是“不规则的”：新形成的流程本身就是一个服务，它公开了一种全新的聚合功能。

在面向服务的计算环境（其实更准确地说，是面向服务架构的计算环境）中，系统可以是高度分布、异构的。它一般包括服务运行时环境（Service Runtime）、服务总线（Service Integration Infrastructure）、服务网关（Service Gateway）、服务注册库（Service Registry）和服务组装引擎（Service Choreography Engine）等。

我们知道，面向服务的计算机环境是和传统的主机模式的计算环境、客户机-服务器模式的计算环境、基于组件的计算环境、基于分布式网络的计算环境和混合 Web 服务的计算环境是有区别的，面向服务的计算环境具有以下特点：

1. 在整个 IT 生命周期中（包括应用的构建和运营）和各个阶段，都是以服务为中心的，服务是核心概念。

2. 软件系统和业务对齐，软件系统具有足够的灵活性，也就是说软件系统对业务需求和业务需求的变化响应是否可以达到业务人员的期望。

3. 软件系统本身是松散耦合的。松散耦合是指信息的消费者和信息的提供者之间是否是松散耦合的，如果以服务为中心的话，那其实就是指的是，服务的消费者和服务的提供者之间是否是松散耦合的。这包括：在服务契约的设计上，是否通过抽象设计减少了技术依赖性；在服务调用层面上，是否通过 Web Service 技术保持服务调用的平台的中立性。

4. 软件功能的暴露或者说功能的接口是粗粒度的。

5. 位置和传输协议是透明的。所谓位置的透明，就是指不论服务组件的实际位置 URL 如何变化，客户端的调用程序的 URL 都不需要改变。同理，目前的服务组件如 EJB、Web 服务、JMS 都只能接收特殊的传输协议，如 EJB 只接收 RMI 的传输协议，Web 服务只接收 SOAP 的传输协议，JMS 组件只接收 JMS 的传输协议。这样导致客户端调用这些服务组件时，也必须采用相应的传输协议才能调用，一旦组件的传输协议改

变了, 客户端也必须要修改相应的传输协议。这就是传输协议的不透明。所谓传输协议的透明, 就是指不管服务组件的传输协议如何变化, 客户端的调用程序的传输协议都不需要改变。

面向服务的体系结构可以基于现有的系统投资来发展, 而不需要彻底重新创建系统。如果企业将开发力量集中在创建服务、利用现有的技术、结合基于组件的方法来开发软件上, 将获得如下几方面好处:

1. 利用现有的资产。SOA 提供了一个抽象层, 通过这个抽象层, 企业可以继续利用它在 IT 方面的投资, 方法是把这些现有的资产包装成为提供企业功能的服务。企业可以继续从现有的资源中获取价值, 而不必重新从头开始构建。

2. 更易于集成和管理复杂性。在面向服务的体系结构中, 集成点是规范而不是实现。这提供了实现透明性, 并将基础设施和实现发生的改变所带来的影响降到最低限度。通过提供针对基于完全不同的系统构建的现有资源和资产的服务规范, 集成变得更加易于管理, 因为复杂性是隔离的。当更多的企业一起协作提供价值链时, 这会变得更加重要。

3. 更快的响应和上市速度。从现有的服务中组合新的服务的能力为需要灵活地响应苛刻的商业要求的组织提供了独特的优势。通过利用现有的组件和服务, 可以减少完成软件开发生命周期包括收集需求、进行设计、开发和测试所需的时间。这使得可以快速地开展新的业务服务, 并允许组织迅速地对改变做出响应和减少上市准备时间。

4. 减少成本和增加重用。通过以松散耦合的方式公开业务服务, 企业可以根据业务要求更轻松地使用和组合这些服务。这意味资源副本的减少, 以及重用和降低成本的可能性的增加。而这一切都依赖于的优势高可复用性、灵活性, 以及更好的扩展性和可用性。

1.3 Web 服务组合

Web 服务结合了面向组件的方法和 Web 技术的优势, 利用标准网络协议和 XML 数据格式进行通信, 能实现不同平台上各种语言编写的服务之间的交互电子商务的迅速发展, 对跨企业的应用集成以及企业内部应用的集成提出了需求。而 Web 服务的出现正好为这些应用的集成提供了一个很好的解决方案, 使企业内和企业之间应用的集成成为可能, 而且使应用的集成更加快捷和方便。因此, Web 服务在企业的电子商务系统中得到了广泛的运用。

单个 Web 服务通常只提供唯一的调用函数完成单一的功能, Web 服务组合能够利用 Internet 上分布的易于执行的轻量级服务创建功能丰富易于用户定制的复杂服务, 将松散耦合的相关 Web 服务有机组织成更为可用的系统。

服务组合作为一个满足业务需求的流程规划, 涉及基本服务和复合构件服务的协同, Web 服务组合需要解决的主要问题有: 采用怎样的 Web 服务组合模型作为组合研究的框架基础; 在可替换的服务中如何根据 QoS 要求进行质量驱动的服务选择; 建立

怎样的代价模型以评估 Web 服务组合的代价；怎样定义各个原子服务之间的关系；如何提供更为精确和通用的语义信息的支持；怎样验证和测试组合 Web 服务；如何对 QoS 或其他行为进行监控，保证在满足需要的同时能够高效的利用资源。

从软件工程的角度看，以上问题涉及服务组合的设计、开发、运行和管理等的软件生命周期。从相关技术看，涉及了对形式化方法、语义学和运筹规划等领域知识，使其成为一个跨学科的研究领域。

围绕服务组合问题，已经有大量的研究者进行了相当广泛和深入的研究。当前解决服务组合的问题主要有基于工作流（workflow）的方法、基于 AI Planning 的方法和基于软件工程的方法。

1. 基于 AI Planning 方法：主要有基于逻辑程序设计语言 Golog 方法、基于适合过程模型的分层任务网络 SHOP2 方法和使用基于规则的专家系统构建复合服务的开发工具 SWORD 方法。目前，利用 AI 规划技术求解语义 Web 服务的自动组合问题不能够对组合服务中各种动态关系，特别是并发关系进行刻画；

2. 基于工作流方法：主要包括基于 Web 服务业务流程执行语言 BPEL4WS/BPEL 方法、基于业务流程建模语言 BPML 方法和基于 Web 服务编排定义语言 WS-CDL 方法。基于工作流相关方法往往需要大量人工操作事先定义该商业进程，不是真正意义上动态的自动的服务组合；

3. 基于软件工程的方法，主要方法为借鉴软件工程中的原则的 Web 构件组合方法。

1.4 本书主要内容

本书的核心内容从语义 Web 服务组合的相关概念、关键技术等方面进行了研究，其主要工作和创新性贡献如下：

1. 详细介绍了面向服务的计算环境、Web 服务、语义 Web、服务组合等主要相关概念，并围绕语义 Web 服务组合的研究现状及存在的问题，总结服务组合方法与理论。

2. 提出了将语义 Web 服务分成信息类 Web 服务和动作类 Web 服务的分类方法，并针对不同的服务类型采用不同的组合策略。针对信息类 Web 服务，提出基于参数相似度的服务组合算法，由于获取信息类 Web 服务不会导致外部状态的变化这一特性，设计动态组合算法，该方法可以根据用户请求实现信息类服务的动态组合。

3. 介绍了描述逻辑以及动态描述逻辑 DDL 系统。DDL 结合描述逻辑与动作理论，支持对服务过程方面的建模和推理。DDL 明确的描述了环境与服务之间的交互的内涵语义，填补基于语义内容的推理和基于服务的推理之间的鸿沟。利用 DDL 的推理能力，给出了基于 DDL 的语义 Web 服务之间的执行关系的定义，并给出基于 DDL 的组合算法。通过这种形式化的描述，可以在服务组合过程中，去除等同服务、相似服务等，缩小服务搜索空间，提高组合效率。

4. 针对混合类 Web 服务，提出了半自动化的服务组合方法。该方法采用基于主体

的服务虚拟化技术,将动态不稳定的物理 Web 服务注册为主体虚拟服务,并使用主体虚拟服务来构建组合的领域模板。给出了主体服务描述语言以及主体服务匹配算法,介绍了基于情景演算的主体虚拟服务的形式语义。利用多主体协作领域中已有的优化算法,可以更好地解决服务组合问题。

5. 服务组合过程中的语境信息有多种类型,根据在实现过程中的共性特点进行抽象,提出了基于 OWL-S 扩展的语境信息的语义模型 OWL-SC,该模型从用户因素、服务因素以及所处环境等三个角度刻画了隐含在服务组合过程中并且不断变化的语境信息。使用语境信息定位用户偏好,具有较高的自动性和组合效率,并且用户满意度有明显提高。

6. 上述工作分别从理论和应用的角度对语义 Web 服务组合进行了深入研究,初步构建了基于主体的服务组合原型系统 SWSBroker,为了将服务组合实用化,最后介绍了多民族语言语料库网络交流平台的设计,使用结果表明了本书所述的工作的可行性和有效性。

第 2 章 语义 Web 服务

2.1 Web 服务基础

2.1.1 Web 服务简介

SGML (Standard Generalized Markup Language) 作为一种通用的文档结构描述的符号化语言, 主要用来定义文献模型的逻辑和物理类结构, 在 20 世纪 60 年代后期就已经存在了。但是由于其过于复杂, 没有得到大规模的应用。进入到 20 世纪 90 年代, HTML 作为面向呈现的标记语言得到迅猛发展, 但 HTML 结构固定, 难以扩展, 缺乏必要的语义信息, 不适用于信息交互。

为了克服 HTML 和 SGML 的弊端, 1996 年 W3C (World Wide Web Consortium) 专家组对 SGML 进行了裁减, 形成了 SGML 子集, 即为 XML (eXtensible Markup Language), XML 作为一种扩展性标记语言, 成为描述电子商务、多媒体演示数据等各种各样数据应用语言的基础语言。通过 XML 的应用, 开发者可以附加语义和上下文信息, 并可实现跨越互联网协议传输。XML 的出现和发展为未来的 Web 服务奠定了很好的技术基础。

从上个世纪 90 年代到本世纪初, Web 应用主要围绕 Internet 上的数据展开。在几乎所有的应用中, 这种面向数据的应用总是由使用者通过使用 Web 浏览器等工具的方式完成。

XML 技术以及其他相关技术的快速发展改变了 Web 应用发展的格局: 它将 Web 应用从信息交互的领域扩展到了服务交互的领域, 这种服务交互就是当前引起工业界和学术界关注的 Web 服务 (Web Services)。

Web 服务的基本思想就是使应用程序也具有 Web 分布式编程模型所具有的松散耦合性。Web 服务提供一个建立分布式应用的平台, 使得运行在不同操作系统和不同设备上的软件、用不同的程序语言 and 不同厂商开发的软件都可以利用这个平台实现分布式计算的目的。

下面我们介绍 Web 服务的相关概念。服务, 它代表了给用户提供的功能, 以帮助用户解决其问题。在面向服务计算中, 服务是指具有规范封装的软件或资源, 可以对它进行规范地使用。目前服务的具体形式有 Web 服务 (Web Services)、Grid 服务 (Grid services)

等。为了更好地理解服务的概念, 我们给出关于服务的一些权威的定义。

- 第一届面向服务的计算国际会议上的定义

服务是一类自治的、平台独立的计算元素, 采用基于 XML 的方式可以对其进行描述、发布、发现及组合, 从而能够支持大规模分布式应用的构造。