Web 服务组合综述*

李景霞[1,3]，侯紫峰[2]

(1. 中国科学院 计算技术研究所，北京 100080；2. 联想研究院，北京 100085；3. 中国科学院 研究生院，北京 100049)

摘 要：Web 服务技术的快速发展使得利用 Web 服务组合来满足应用的需求成为可能。首先介绍了 Web 服务体系结构和两种有代表性的 Web 服务描述方式，接着着重介绍了当前主流的三大类 Web 服务组合方法并分析了各自的优缺点，最后指出 Web 服务组合研究面临的挑战。

关键词：Web 服务；WSDL；语义 Web；OWL-S；Web 服务组合

中图法分类号：TP393 **文献标识码**：A **文章编号**：1001-3695(2005)12-0004-04

Research Survey of Web Service Composition

LI Jing-xia[1,3], HOU Zi-feng[2]

(1. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080, *China*; 2. *Lenovo Research Institute, Beijing* 100085, *China*; 3. *Graduate School, Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: Along with the rapid development of Web service technology, its possible to use composition of Web service to meet the needs of the application. In this paper, the architecture of Web service is introduced, then two representative description modes of Web service are presented. Then it focuses on discussing three leading methods of Web service composition and analyzes their merit and shortcoming respectively. At last the main challenges of Web service compositions research are pointed out.

Key words: Web Service; WSDL; Semantic Web; OWL-S; Web Service Composition

近年来 Web 服务技术得到快速发展和应用，Web 服务是基于网络的、分布式的、自描述的、模块化的组件，它执行特定的任务，遵循一定的技术规范，提供了面向 Internet 应用的统一服务注册、发现、绑定和集成机制，成为广域环境下实现互操作的一种主要机制，得到产业界和学术界的广泛认可。

1 Web 服务体系结构

文献[1,2]描述了 Web 服务的基本体系结构，Web 服务体系结构由三种角色和三种基本操作构成。三种角色包括：

(1) 服务使用者。它是一个应用程序，一个软件模块或需要服务的另一种服务。它发起对注册中心中服务的查询，通过传输绑定服务，并执行服务功能，服务使用者根据接口契约来执行服务。

(2) 服务提供者。它是一个可通过网络寻址的实体，接收和执行来自使用者的请求。将自己的服务和接口契约发布到服务注册中心，以便服务使用者发现和访问该服务。

(3) 服务注册中心。它是服务发现的支持者，包含一个可用服务的存储库，并允许感兴趣的服务使用者查找服务提供者接口。

Web 服务体系结构中的每个实体都扮演着服务提供者、使用者和注册中心这三种角色中的某一种(或多种)。Web 服务体系结构中的三种操作包括：

收稿日期：2005-03-28；修返日期：2005-05-13

基金项目：国家“863”计划基金资助项目(2003AA148020)

(1) 发布。为了使服务可访问，需要发布服务描述以使服务使用者发现和调用它。

(2) 发现。服务请求者定位服务，查询服务注册中心来找到满足其需求的服务。

(3) 绑定和调用。检索完服务描述之后，服务使用者根据服务描述信息来调用服务。

Web 服务体系使用一系列标准和协议来实现相关的功能。使用 WSDL(Web Services Description Language)来描述服务，使用 UDDI(Universal Description, Discovery and Integration)来发布、查找服务，使用 SOAP(Simple Object Access Protocol)来调用服务。

2 Web 服务的描述

Web 服务具有广泛的适应性和应用背景，对 Web 服务的描述也多种多样。下面介绍两种有代表性的 Web 服务描述。

2.1 WSDL

WSDL[3]是用来描述网络(Network)服务或终端(Endpoint)的一种 XML 语言，它用于定义 Web 服务以及如何调用它们。WSDL 文档可用于动态发布 Web 服务、查找已发布的 Web 服务以及绑定 Web 服务。

如图 1 所示，WSDL 将 Web 服务定义为服务访问点或端口的集合。在 WSDL 中，服务访问点和消息的抽象定义从具体的服务部署或数据格式绑定中分离出来。消息是对交换数据的抽象描述；而端口类型是指操作的抽象集合；用于特定端口类

型的具体协议和数据格式规范构成了可再次使用的绑定。将 Web 访问地址与可再次使用的绑定相关联, 可以定义一个端口, 而端口的集合则定义为服务。

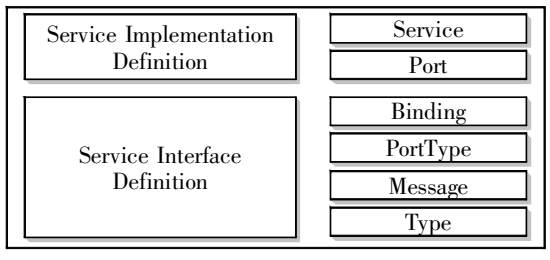

图 1　WSDL 服务实现和接口定义

语义网[4] (Semantic Web) 的发展要求可以通过 Web 资源的内容等语义信息来访问 Web 资源, 而不仅仅是通过数据类型匹配、关键词匹配来查找信息。Web 服务作为 Web 上的一类越来越重要的资源, 同样要求 Semantic Web 对其提供广泛的语义支持, 从而实现 Semantic Web Services。OWL-S 就是 Semantic Web 中一种用来描述 Web 服务的属性和功能的本体规范, 它使用一系列基本的类和属性来描述 Web 服务, 提供了一个可共享的框架。它的目标是使得 Web 服务成为计算机可理解的实体, 从而便于实现服务的发现、调用、互操作、组合、验证和执行监控等。

2.2　OWL-S

在 OWL-S[5] 中, 描述服务的基本信息主要有三类本体: ServiceProfile, ServiceModel 和 ServiceGrounding, 如图 2 所示。

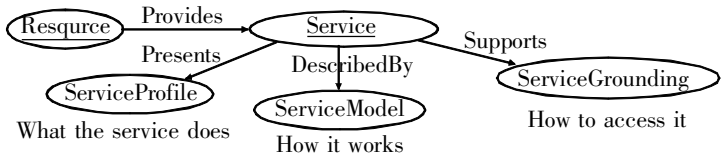

图 2　OWL-S 模型的三类本体

(1) ServiceProfile 描述了服务查询代理用来判断服务是否满足它要求的一组属性信息, 如服务提供的功能、服务的输入、输出、执行的前提条件、执行产生的结果等。它描述了服务做什么, 用于服务的自动发现。

(2) ServiceModel 描述了服务的过程模型(使用服务时所涉及的控制流和数据流) 。它描述了服务是如何工作的, 与 ServiceGrounding 配合用于服务的自动组合和执行。

(3) ServiceGrounding 将过程模型的描述和 WSDL 中的通信协议及消息描述联系起来。它描述了如何访问一个服务。

OWL-S 的 ServiceGrounding 将 OWL-S 与 WSDL 关联到一起。二者之间的关联有利于一个服务开发者受益于两种规范的优势, 一方面开发者能够利用 OWL-S 的过程模型, 另一方面开发者能重用 WSDL 的已有工作和基于 WSDL 的消息传递机制。

OWL-S 和 WSDL 是互补的, 因为两者实际上覆盖了不同的概念空间, 但在提供 WSDL 所谓的抽象类型规范方面确实存在重叠。WSDL 用 XML Schema 表达, 而 OWL-S 用 OWL 类表达。但是 WSDL 表达不了 OWL-S 类的语义, 而 OWL-S 表达不了 WSDL 的绑定信息。因而, 采用 OWL 类表达 WSDL 中抽象类型的声明, 而使用 WSDL 的绑定结构表达消息的格式。

3　Web 服务组合

随着 Web 服务技术的日益成熟, 越来越多的稳定易用 Web 服务共享在网络上。但单个的 Web 服务能够提供的功能有限, 为了更加充分地利用共享的 Web 服务, 有必要将共享的 Web 服务组合起来, 提供更为强大的服务功能, 加快系统开发的速度, 快速满足用户需求。

Web 服务组合方法从组合方案生成方式来分有两大类: 静态组合和动态组合。静态组合意味着请求者应在组合计划实施前创建一个抽象的过程模型。抽象的过程模型包括任务的集合以及任务间的数据依赖关系, 每个任务包含一个查询的子句, 用来查找完成任务的真正的 Web 服务。因此这里的自动仅指 Web 服务的选择和绑定是由程序自动完成的。静态组合中最常用的是用图来描述过程模型。而动态组合不仅自动地选择、绑定 Web 服务, 同时更重要的是自动地创建过程模型。这需要请求者指定一些约束关系, 包括 Web 服务间的依赖关系、用户的偏爱等。

在此阐述服务组合方法, 我们采用按组合方法抽象层次由低到高来逐一介绍。现有的服务组合方法可分为四个层次: 硬编码模式、用特定的服务组合定义语言定义组合服务模式、模型驱动模式、根据目标推理服务组合方案模式, 如图 3 所示。硬编码模式将服务组合以硬编码的方式来实现, 是组合服务最原始的方法, 也最不灵活, 形成的组合服务适应性差。Web 服务是动态变化且自治的, 当某一服务组件产生错误时, 需要修改形成的组合服务工作量大, 另一方面该方法也不能适应应用需求的动态变化。

3.1　基于 BPEL4WS 的 Web 服务组合

BPEL4WS[6] (Business Process Execution Language for Web Services, Web 服务的商业流程执行语言) 是专为整合 Web 服务而制定的一项规范标准。BPEL4WS 的作用是将一组现有的服务整合起来, 从而定义一个新的 Web 服务。它能够将 Web 服务调用、操作数据、抛出故障或终止一个流程等不同的活动连接起来, 从而创造出复杂的流程, 这些活动可以嵌套到结构化活动中, 结构化活动定义了活动的运行方式和控制流程。BPEL4WS 可将一组现有的服务整合起来, 从而定义一个新的 Web 服务。

BPEL4WS 流程是一个流程图, 用来表达特定业务的处理逻辑和算法, 流程的每一步称为一个活动。BPEL4WS 主要利用 WSDL 使得服务的动态绑定成为可能, 但它没有提供具体方式来选取动态绑定时需要调用的服务, 并且 BPEL4WS 不支持在应用运行时的流程模型的调整。

与类似 BPEL4WS 这样来表达业务流程的还有 BPEL 等。查找 Web 服务时需要搜索巨大的服务空间, 比较、匹配满足条件的服务, 因此用来组建组合服务的方法复杂、动态以致难以手工处理。为了自动地开发组合服务, 需要同软件开发方法一样的系统的方法来分析需求, 将需求所涉及的活动模型化。改变服务开发方法使其适应服务组合开发的好处是我们能够深入了解组建服务组合流程以致更好地管理其实现。

3.2　模型驱动的 Web 服务组合

在用特定语言定义组合服务的基础上, 提出用模型驱动的方法来开发、管理动态服务组合, 模型驱动方法[7] 将软件开发方法学应用到服务组合中。该方法最大的特点是将组合逻辑与组合规范分离开, 使用 UML 描述服务组合, 使得能在更抽象的层次将 Web 服务组合模型化, 该组合定义随后可自动

地映射到特定的规范如 BPEL4WS 上,进而再关联到实际具体的 Web 服务上。模型驱动 Web 服务组合分以下五步,如图 4 所示。

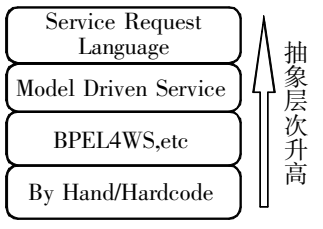

图 3 Web 服务组合方法

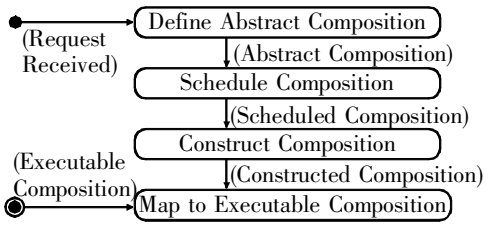

图 4 模型驱动组合生命周期

(1) 定义抽象组合。定义抽象组合服务包括指定组合服务的需求信息、约束条件、组成组合服务的 Web 服务组件及执行中可能发生的意外行为。

(2) 调度组合。决定怎样、何时运行服务,通过信息关联来表达数据间的依赖性,赋予各 Web 服务组件活动的执行优先权。该阶段产生供选择的组合调度方案,由应用开发者来选择。

(3) 创建组合。创建可执行的 Web 服务组合,输入调度组合的方案,输出可执行的模式,如由 BPEL4WS 表示的可执行流程。

(4) 映射为可执行的组合。系统将可执行的流程转变成编程语言中具体的对象,运行 Web 服务组合。

(5) 监视组合的执行。系统监视执行流程,确保上述步骤的正确性。

模型驱动法描述业务流程建立在 UML 基础之上,不依赖特定的规范,适用范围广。其缺点是该方法需要集成规划工具和服务组合服务器等子系统,较为复杂,难以实现。

3.3 基于 AI 规划的 Web 服务组合

OWL-S 的出现为基于 AI 规划的组合方法提供了可能,将 Web 服务看成 AI 中的动作,通过输入输出参数、前提和结果等来描述 Web 服务。在服务组合时,只要将 Web 服务的这些描述映射为动作形式化描述,在 Web 服务空间中以构造 Web 服务组合为目标,通过形式化的推理来得出 Web 服务的组合序列,动态形成服务组合方案,同时也能够保证规划结果的正确性和完整性。基于 AI 的规划方法包括情景演算规划法、层次任务网规划法和基于规则的规划法。

3.3.1 情景演算

情景演算是一种形式化的规划方法。它是一种一阶逻辑语言,主要用来表示动态变化的世界,是一种关于状态、动作和动作作用于状态结果的形式化谓词演算。在状态演算中,世界上的所有变化都是动作执行的结果。我们把知识表达为关于状态和动作的公式,然后用一阶谓词演算的演绎系统来回答这样的问题:“存在满足一定目标属性的状态吗? 如果存在,现在的状态如何通过动作才能被转换为目标状态呢?”。对这样的一个询问的回答是构造一个到达目标状态的计划,这个计划可以简单地表示为动作的序列。

McIlraith[9,10] 将 Golog 作为自动构建 Web 组合服务的语言,Golog 是一种建立在情景演算基础之上的逻辑编程语言,通过提供更高层次的过程和用户定制的约束描述来解决 Web 服务组合问题。Golog 被采纳成为一种表示、推理问题的自然的、形式化方法。这种方法的主要思想是软件代理可以对 Web 服务推理,从而自动地执行 Web 服务的发现、执行、组合以及交互。用户提出的需求和约束可以通过情景演算的一阶谓词语言来描述。将每一 Web 服务看成一个活动——原始活动或复杂活动。原始活动可以看成是改变事物状态的可感知活动,或是改变代理状态的信息收集活动;复杂活动是多个单个服务的组合。在此运用基于情景演算的逻辑语言来描述 Web 服务前束条件及预期结果,描述组合服务的目标及相关约束,在 Web 服务空间中运用推理来发现一组 Web 服务序列满足服务请求。组合服务是由过程编程语言结构连接的原子服务的集合。

3.3.2 层次任务网规划法

Hierarchical Task Network(HTN) Planning[11] 利用 OWL-S 作为 Web 服务的描述语言,通过任务分解的方法来组建组合服务。与其他规划方法不同的是,HTN 的核心不是状态,而是任务。一个基于 HTN 的规划系统将目标任务分解为子任务的集合,这些子任务再分解成更小的子任务,一直循环这个过程,直到所有的子任务仅包含原子 Web 服务,即现有的 Web 服务。在每一轮任务的分解过程中,都会检验是否违反给定的条件(如子任务数不能超过一定数目)。如果目标任务在不违反任何给定的条件下,能够被成功分解成原子任务集,则规划问题成功求解。SHOP2[12] 就是利用 HTN 方法实现 Web 服务组合的系统。

3.3.3 Rule-based 规划

Medjahed[15] 提出了一种从高层次声明描述产生组合服务的技术,这一方法使用组合规则来决定两个服务是否可以组合。组合方法包括以下四个阶段:①使用组合服务说明语言描述目标服务;②利用组合规则匹配服务组合规划;③如果在匹配阶段,产生多于一个的规划方案,则服务请求者在组合质量参数的基础上选择一个规划方案;④方案产生阶段,一个详细描述组合服务的方案产生并提交给服务请求者。

这里我们主要关注组合规则,它定义了如何产生组合方案。组合规则考虑了 Web 服务的句法和语法特性,句法规则包括操作模式规则和交互绑定协议规则。语法规则包括下列子集:①消息组合定义了两个 Web 服务可组合当且仅当一个 Web 服务的输出和另一个 Web 服务的输入是兼容的;②操作语义组合定义了两个 Web 服务领域的、类别的和目标的兼容度;③质量组合定义了请求者对组合服务操作质量的偏好;④组合合理性定义了服务的组合是否合理。笔者引入组合模板的概念来定义不同种类服务间的依赖关系。Sword[16] 就是利用基于规则服务组合方法的一个组合服务的开发工具集。

基于 AI 的服务组合方法能动态地生成 Web 服务组合方案、动态地链接具体实现的 Web 服务,能适应 Web 服务的动态变化,更能满足业务不断变化的需求。但如何高效、准确地在 Web 服务空间中查找适合的 Web 服务,如何评价生成的 Web 服务组合方案与需求吻合度等一系列问题都是亟待深入研究的问题,以便使基于 AI 的 Web 服务组合能更好地为应用需求服务。

4 总结与展望

Web 服务体系结构是面向对象分析与设计的一种合理发展,同时也是电子商务解决方案中,面向体系结构、设计、实现

与部署而采用的组件化的合理发展,这两种方式在复杂的大型系统中经受住了考验。与面向对象系统一样,封装、消息传递、动态绑定、服务描述和查询也是 Web 服务中的基本概念,而且 Web 服务另外的一个基本概念就是: 所有东西都是服务,这些服务发布一个 API 供网络中的其他服务使用,并且封装了实现细节。Web 服务的一个主要思想就是未来的应用将由一组应用了网络的服务组合而成。如何基于语义准确地发现 Web 服务、如何准确产生满足应用需求及相应约束的组合方案,如何对 Web 服务组合方案进行评估、优化,以及监控 Web 服务组合的执行和相应的容错处理都是 Web 服务组合领域有待深入研究的问题。

参考文献:

[1] P Papazoglou, D Georgakopoulos. Service-Oriented Computing [C]. Service Oriented Computing Communications of the ACM, 2003.

[2] Dipanjan Chakraborty, Anupam Joshi. Dynamic Service Composition: State-of-the-Art and Research Directions[EB/OL]. Technical Report. http://research.ebiquity.org/re/papers.html, 2001-11.

[3] Web 服务: WSDL 专题[EB/OL]. http://www-900.cn.ibm.com/developerworks/cn/Webservices/ws-intwsdl/part1/index.shtml.

[4] Joachim Peer. Bringing Together Semantic Web and Web Services [C]. Lecture Notes in Computer Science Archive Proceedings of the 1st International Semantic Web Conference on the Semantic Web Table of Contents. 279-291.

[5] OWL-S 1.1 Release[EB/OL]. http://www.daml.org/services/owl-s/1.1.

[6] Web 服务: BPEL4WS 专题[EB/OL]. http://www-900.cn.ibm.com/developerWorks/cn/Webservices/ws-theme/ws-bpel.shtml.

[7] Bart Orriëns, Jian Yang, *et al*. Model Driven Service Composition [EB/OL]. http://maximus.uvt.nl/sigsoc/pub/Orriens%20et%20al%20-%20model%20driven%20service%20composition.pdf.

[8] David Skogan, Roy Grønmo, Ida Solheim. Web Service Composition in UML[C]. The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC), Monterey, California, 2004.

[9] Sheila McIlraith, Tran Cao Son. Adapting Golog for Composition of Semantic Web Services[C]. Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002), 2002. 482-493.

[10] Sheila McIlraith, Tran Cao Son, *et al*. Semantic Web Services[C]. IEEE Intelligent Systems, Special Issue on the Semantic Web, 2001.

[11] Evren Sirin, Bijan Parsia, Dan Wu. HTN Planning for Web Service Composition Using[EB/OL]. http://www.mindswap.org/papers/SHOP-JWS.pdf.

[12] Dan Wu, Evren Sirin, James Hedler. Automatic Web Services Composition Using SHOP2[EB/OL]. http://www.mindswap.org/papers/ICAPS03-SHOP2.pdf.

[13] Evren Sirin, Bijan Parsia. Planning for Semantic Web Services[C]. Semantic Web Services Workshop at the 3rd International Semantic Web Conference (ISWC2004), 2004.

[14] Claus Pahl. A PiCalculus-based Framework for the Composition and Replacement of Components[C]. Proc. Conference on Object-Oriented Programming, Systems, Languages, and Applications(OOPSLA'2001), 2001.

[15] Brahim Medjahed, Athman Bouguettaya, Ahmed K Elmagarmid. Composing Web Services on the Semantic Web[EB/OL]. http://www.cs.purdue.edu/homes/ake/pub/composition_VLDBJ.pdf.

[16] Shankar R Ponnekanti, Armando Fox. Sword: A Developer Toolkit for Web Service Composition[EB/OL]. http://www2002.org/CDROM/alternate/786/.

[17] Vadim Ermolayev, Natalya Keberle, Oleksandr Kononenko. Towards a Framework for Agent-enabled Semantic Web Service Composition [EB/OL]. http://www.cs.jyu.fi/ai/papers/IJWSR-2004.pdf.

[18] Evren Sirin, James Hendler, Bijan Parsia. Semi-automatic Composition of Web Services Using Semantic Descriptions[EB/OL]. http://www.mindswap.org/papers/composition.pdf.

作者简介:

李景霞(1976-),女,博士研究生,主要研究方向为 Web 服务组合、分布式计算; 侯紫峰,中科院计算所研究员,首席研究员,博士生导师,主要研究领域为移动计算、数字信号处理、网格计算等。

(上接第 3 页)

可以认为,嵌入式移动操作系统的研究在以下方面将继续深入: 客户和实体移动性的结合带来新的模式和方法; 适应网络情况和应用逻辑的高度灵活的系统模型; 大量遗留应用向移动操作系统的迁移; 针对网络非连接、弱连接和短路动态变化所造成问题的解决; 针对移动环境上下文信息获取与表示的一些新方法以及嵌入式 Linux 的完善等。无线通信技术的发展促使了移动应用的增长、常规分布系统向移动环境的迁移。由于移动环境的若干特殊性始终存在并有所变化,如何保持对应用的透明性、减少应用开发者对移动特性的关注、继承大量遗留的应用,将是移动终端操作系统长期而主流的需求。

参考文献:

[1] mielinski T, Korth H F. Introduction to Mobile Computing[A]. Mobile Computing[M]. Kluwer Academic Publishers, 1996. 1-43.

[2] M Satyanarayanan. Fundamental Challenges in Mobile Computing [C]. Proc. ACM SigMobile, 1997. 144-169.

[3] 文远保,周俊杰. 移动计算环境中自适应体系结构的研究[J]. 计算机应用研究, 2002, 19(8): 32-34.

[4] Martin Tasker, *et al*. Mobile Solutions on the EPOC Platform[M]. Wrox, 2000.

[5] Jerry Epplin. Linux as an Embedded Operating System[A]. Embedded Systems Programming[C]. 2000. 12-36.

[6] Guanling Chen, David Kotz. A Survey of Context-Aware Mobile Computing Research[R]. Dartmouth Computer Science Technical Report TR2000-381, 2000. 124-156.

[7] A Joseph, J Tauber, M Kassboek. Mobile Computing with the Rover Toolkit[J]. IEEE Transactions of Computer, 1997, 46(3): 337-352.

[8] Benini L, Bogliolo A, De Micheli G. A Survey of Design Techniques for System-Level Dynamic Power Management[J]. IEEE Trans. VLSI Syst., 2000, 8: 299-316.

作者简介:

王继刚(1978-),男,安徽马鞍山人,博士研究生,主要研究方向为嵌入式系统、移动计算、第三代移动通信; 谢鑫(1972-),男,四川成都人,三级主任工程师,硕士,主要研究方向为计算机操作系统、嵌入式系统; 顾国昌(1946-),男,上海人,教授,博士生导师,主要研究方向为嵌入式系统、人工智能。