

Web 服务组合方法综述

倪晚成, 刘连臣, 吴澄

(清华大学国家 CIMS 工程技术研究中心, 北京 100084)

摘要: Web 服务组合通过重用已有的 Web 服务满足用户高质量的应用需求, 是面向服务架构研究领域的热点问题。该文对当前 Web 服务组合的研究现状进行了综述, 分析了 WSC 的概念和实现框架。根据研究侧重点及其依赖的技术基础, 将 WSC 方法归为两大类别——基于 workflow、状态演算和进程代数模型描述的过程驱动的组合方法和基于语义描述的自动服务组合方法。对 WSC 的评价模型及其研究与应用中所面临的挑战进行了论述。

关键词: 面向服务架构; Web 服务; Web 服务组合

Survey on Web Services Composition Methods

NI Wan-cheng, LIU Lian-chen, WU Cheng

(National CIMS Engineering Research Center, Tsinghua University, Beijing 100084)

【Abstract】 Web service composition, as a new technology to fulfill the user's requirements using existing ones, has become a pop topic in the research of Service Oriented Architecture(SOA). The survey is made on current WSC methods. The concepts and implementing frameworks of WSC are discussed. And based on the research emphases and its dependent technology, WSC methods are classified into process driven WSC such as work-flow based, state-figure and process algebra based methods, and semantics based automatic WSC. The evaluation models and the challenges in research and implementation of WSC have been summarized.

【Key words】 Service Oriented Architecture(SOA); Web service; Web service composition

2004 年以来, Web 服务作为跨组织应用集成的支持技术得到业界的广泛支持, 面向服务架构(Service Oriented Architecture, SOA)成为信息技术的新热点。SOA 的一个重要理念是将“单独、独立、封装”的服务组合成为更大型的服务, 实现软件重用、发挥服务的潜力。因此, Web 服务组合受到学术界和企业界的极大关注, 涌现出大量的 WSC 研究。

1 WSC 概念

WSC 源于软件重用, 其基本思想是使用系统中已有的 Web 服务, 通过它们一定顺序的组合或组合顺序的改变, 创建出新的或更高质量的服务满足用户需求。

目前对 WSC 尚无统一定义, 研究者从不同角度和侧重点做了不同的定义。总的来说, 其定义可归纳为两个: (1) 基于过程模型: 从 WSC 内在因素的角度, 将其定义为一个依赖于特定控制流和数据流结合起来的、能够完成一定任务的 Web 服务集合, 如 AgFlow^[1]。(2) 基于构件单元: 从构件的角度, 将 WSC 定义为一个由自治且能相互协作的自描述单元所组成的系统, 如文献[2]。上述定义分别强调了 WSC 两个不同的研究角度。

2 WSC 实现框架

典型的 WSC 的实现框架包括 2 种用户角色(服务请求者和提供者)和 5 个部件(翻译器、组合管理器、执行引擎、服务匹配器和注册中心), 可选部件本体库为服务描述提供本体定义和推理支持, 如图 1 所示。

WSC 流程如下:

(1) 服务提供者通过服务注册将服务信息发布到注册中心的服务库。

(2~3) 服务请求者提交的服务需求经翻译器处理, 从自然

语言变为具有语义的需求描述传递给组合管理器。

(4) 组合管理器根据需求描述和来自服务库的服务描述, 生成满足服务需求的组合方案, 传递给执行引擎。

(5~7) 执行引擎将组合方案传递给服务匹配器, 服务匹配器根据服务描述选择最适合的 Web 服务, 将其句柄返回执行引擎。

(8) 执行引擎根据组合方案和 Web 服务句柄调用并监控 Web 服务执行。

(9) 最终将执行结果传递给服务请求者。

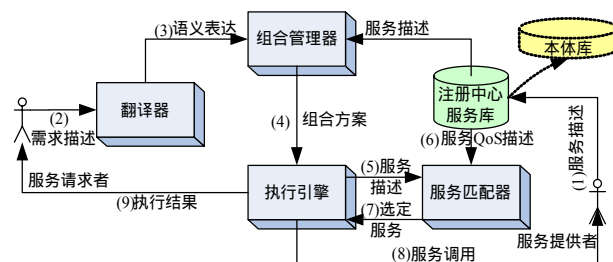


图 1 WSC 实现框架

根据消息在参与组合的服务之间的传递方式, 执行引擎分集中和分布两种模式, 如图 2 所示。集中模式下服务由执行引擎统一控制, 服务之间无交互, 该方式易于实现, 但瓶颈明显。分布模式中, 由前序服务启动后继服务, 通过服务间的消息传递减小系统通信负载、消除瓶颈。

作者简介: 倪晚成(1978 -), 女, 博士研究生, 主研方向: 网格计算, 动态服务组合; 刘连臣, 副研究员; 吴澄, 教授、博士生导师、中国工程院院士

收稿日期: 2007-02-22

E-mail: nwc01@mails.tsinghua.edu.cn

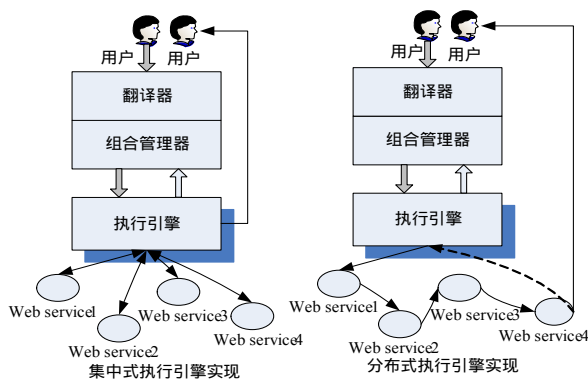


图2 执行引擎模式

3 WSC 方法分类

WSC 研究者有各自不同的角度和侧重点。一部分研究强调“自动化”：根据人工参与的程度将 WSC 分为人工、半自动和自动 3 大类；另一部分研究则强调“动态”：根据对参与组合的 Web 服务的绑定时间，将其分为静态 WSC(参与组合的 Web 服务在设计时确定)和动态 WSC(参与组合的 Web 服务在运行时绑定)。

上述分类各自涵盖了 WSC 的部分研究。这源于目前的 WSC 研究者来自两个派别：第一类研究者从语义网的角度出发，重视 Web 服务的自描述和语义；另一类研究者则从流程建模的角度出发，重视 WSC 中的数据流和控制流，强调对动态环境的处理。两种研究方向依赖于不同的技术基础，使用完全不同的 WSC 模型，这也是目前 WSC 存在两种定义的根本原因。

因此，本文根据 WSC 方法所依赖的技术基础，将其归纳为过程驱动的组合方法和语义驱动的组合方法。

3.1 过程驱动的 WSC

过程驱动的 WSC 基本思想是利用 WSC 与过程模型的相似性，使用较成熟的过程建模工具和语言对 WSC 业务过程进行建模，采用“替代”法将过程模型中的相关组件最终替换成为具体的 Web 服务，从而获得有效的、可执行的 WSC 方案。根据具体的建模工具和实现技术，过程驱动的 WSC 又可分为基于 workflow 模型、基于状态演算和基于进程代数的 WSC。

3.1.1 基于 workflow 模型的 WSC

workflow 模型由于发展较成熟且支持工具较多而成为过程驱动 WSC 中最常见的建模工具。WSFL、BPEL4WS 等扩展的 workflow 语言是该类方法的常用建模语言。

此类方法中，workflow 被用作分布活动的协调引擎或服务组合的建模定义的工具。该类方法^[3]基于扩展 workflow 模型，存在抽象服务(Abstract Service, AS)和实体服务(Concrete Service, CS)两个概念。AS 是一个完全抽象的定义，它被作为“占位符”替代具体活动来建立 workflow 模型；CS 与 AS 相对，对应系统中的一个实际 Web 服务。该类方法包括两个阶段的内容：(1)WSC 设计者为 workflow 中每个活动设定一个 AS，形成抽象服务流程定义；(2)执行前通过中间环节(如图 1 中执行引擎)进行服务的匹配和绑定，将每个 AS 替换成为一个 CS，将抽象服务流转换成实际的 WSC 方案。AS 也称服务模版，因此该方法又称“基于服务模板的组合方法”。

由于参与组合的任一服务不可用都可能导致整个组合方案崩溃，因此研究如何克服动态环境的影响，在基于 workflow 模型的 WSC 方法中十分重要。现有的方法是尽可能保证生成

的组合方案有效，如 CAFISE^[4]采用“逐步演化”法：将服务匹配环节推迟到某个服务节点启动时进行，服务匹配操作查询当前可获得的最佳目标服务，以保证每个被选取的 Web 服务当前时刻最佳并且可获得，从而较大幅度地降低执行中的异常出现；同时将抽象 workflow 到实际组合方案的转换工作分散到各个活动执行前进行，也能降低执行引擎在服务查找和绑定时的开销。

基于 workflow 模型的 WSC 通常以半自动的方式实现：系统提供图形化界面，服务设计者通过人机交互定义抽象服务流；服务匹配和绑定则由系统自动处理。这种方式充分利用了设计者的领域知识降低系统复杂度，又减少了设计者工作量，可实现性强，并具有较好的动态性和灵活性。

3.1.2 基于状态演算的 WSC

基于状态演算的 WSC 方法基本思想是建立 WSC 描述和服务状态图模型的一一映射关系，建立形式化的 WSC 模型，通过形式化模型和工具分析 WSC 系统。其中，Petri 网模型的“安全、死锁、可达”等定义在描述并发、冲突、同步等流程现象上极具优势，并具有良好的形式化语义和直观的图形化描述，因此被较多地应用于对 WSC 方案的可行性分析和验证。

文献[2]将 Web 服务的操作对应于 Petri 网的变迁；服务状态对应库所；用库所与变迁之间的箭头表征状态间因果关系；文献[5]则是将一个服务节点映射为 Petri 网中的库所。面向对象 Petri 网和着色 Petri 网也在 WSC 建模中被使用^[6]。

基于状态演算的 WSC 方法本质上仍是基于 workflow 模型的方法，但它侧重于对 WSC 的形式化建模，主要目的在于验证 WSC 方案的有效性和可行性。

3.1.3 基于进程代数的 WSC

状态演算适用于静态系统描述，难以处理 WSC 的动态特性。用于描述和分析并发、异步、非确定和分布式等系统行为的进程代数，是对动态实体进行建模的正式语言，具有严密的形式化语义，因此被引入到 WSC 研究中。其中支持通道名称传递的 π 演算由于适用于描述结构动态变化的并发系统，而成为进程代数在 WSC 研究中应用的代表。

π 演算首先应用在构件系统中描述构件组合：Pahl^[7]基于 π 演算的“迁移性”和构件“演变性”的相似之处，为基本的构件组合及替换原则建立了基于 π 演算的描述框架。之后 π 演算与 WSC 结合起来，文献[8]通过建立 Web 服务描述与 π 演算中进程描述的对对应关系，实现 WSC 的形式化描述。

由于 π 演算相关工具可用于推演 WSC 的系统行为，并能为分析验证 WSC 方案提供支持，因此也受到 W3C 的青睐：XLANG 和 WS-CDL 都是通过建立 π 演算和 WSDL 基本元素的对应关系而提出的 WSC 描述语言。

基于 π 演算的方法缺乏如 Petri 网的直观图形化支持，但它对动态演化系统的灵活描述使它更适合刻画 WSC 的动态行为。目前对基于 Petri 网和 π 演算的 WSC 方法仍存在争议。

3.2 语义驱动的 WSC

与过程驱动的 WSC 不同，语义驱动的 WSC 强调 Web 服务的自描述，其基本思想是通过为 Web 服务描述和服务请求描述添加能被计算机系统所理解的语义，使 WSC 方案通过推理自动生成。

3.2.1 服务描述

服务描述对于语义驱动的 WSC 十分重要，最基本的服务语义包括四元组

$\{I(\text{数据输入}), O(\text{数据输出}), P(\text{条件输入}), E(\text{输出影响})\}$

常见的服务描述有：

(1)OWL-S：由描述服务能力、服务工作过程和服务访问细节的 ServiceProfile, ServiceModel 和 ServiceGrounding 3 部分组成。ServiceProfile 涵盖了 $\{I, O, P, E\}$ 以及服务 QoS 语义，满足计算机推理需求；ServiceModel 则定义了 WSC 的内部流程。因此被当前的绝大部分的 WSC 研究所使用。

(2)WSDL：描述服务 URL 和命名空间、服务类型、有效函数、函数参数、参数类型以及函数返回值和返回值类型等 Web 服务的通信和调用方法的 XML 文件。它是 Web 服务的事实标准，但它仅通过“端口”规约了服务的功能特性，必须扩充增加 $\{P, E\}$ 和 QoS 语义才能支持 WSC。

(3)自定义语义：为实现WSC推理而定义的非标准简单描述语言。如Sword^[9]中基于规则的服务描述——将服务描述成在特定输入下产生特定输出的“规则”。随着OWL-S的发展，这种服务描述已逐渐淡出。

3.2.2 推理与选择

语义驱动的 WSC 通过对服务组合图的搜索，生成组合方案。

服务组合图是一个改进的状态图：以用户请求的输入为起点，输出为终点；图中状态代表一个服务；变迁代表前后服务之间的语义关联(包括 $\{I, O\}$ 数据相似度， $\{P, E\}$ 匹配度，以及 QoS 要求的符合度)。

WSC 推理就是在服务组合图上的反向搜索过程：以用户需求为起点，首先寻找输出与用户需求目标一致的 Web 服务；再寻找输出与该服务输入匹配的上一个服务；如此类推直到找到输入描述与用户提供输入一致的服务为止。如果搜索获得多个满足用户需求的组合方案，则需要通过评价选择一条最优的方案执行。

在服务组合图上，最优组合方案表现为从起点到终点相邻服务语义关联程度总和最大的一条路径，“最短路径”及其改进算法是方案选择使用的普遍方法。

4 WSC 评价模型

完善的指标体系对于评价WSC方案的优劣具有重要的意义。目前的研究主要通过用户满意度或QoS评价WSC。用户满意度作为一种主观指标难以量化，DOSCOM^[10]使用组合方案的语义相似程度表征用户满意度。更多的研究使用 AgFlow^[11]提出的五维QoS模型。该模型包括执行成本、执行时间、服务声誉、可靠性和可用性 5 个非功能性指标，如表 1、表 2 所示。

表 1 单个服务 QoS 模型

指标	单个服务指标函数
执行成本	$q_{price}(s) = q_{price}(s, op)$ $q_{price}(s, op)$ - 服务 s 操作 op 的执行成本
执行时间	$q_{du}(s) = q_{du}(s, op) =$ $T_{process}(s, op) + T_{trans}(s, op)$
服务声誉	$q_{rep}(s) = \frac{\sum_{i=1}^n R_i}{n}$ R_i - 用户对服务的声誉打分 n - 用户个数
可靠性	$q_{rel}(s) = N_c(s)/K$ $N_c(s)$ - 要求时间内执行成功次数 K - 总调用次数
可用性	$q_{av}(s) = T_a(s)/\theta$ $T_a(s)$ - 服务 s 在 θ 时间内的可获得时间

表 2 组合服务 QoS 模型

指标	组合服务指标函数
执行成本	$Q_{price}(p) = \sum_{i=1}^N q_{price}(s_i, op_i)$
执行时间	$Q_{du}(p) = CPA(q_{du}(s_1, op_1), \dots, q_{du}(s_N, op_N))$ CPA - 关键路径算法
服务声誉	$Q_{rep}(p) = \frac{1}{N} \sum_{i=1}^N q_{rep}(s_i)$
可靠性	$Q_{rel}(p) = \prod_{i=1}^N (e^{q_{rel}(s_i) \times z_i})$ $Z_i = \begin{cases} 0-s, \text{不在关键路径上} \\ 1-s, \text{在关键路径上} \end{cases}$
可用性	$Q_{av}(p) = \prod_{i=1}^N (e^{q_{av}(s_i) \times z_i})$ $Z_i = \begin{cases} 0-s, \text{不在关键路径上} \\ 1-s, \text{在关键路径上} \end{cases}$

该模型对 WSC 的 QoS 进行了明确量化，但在服务提供者利益不可忽视的情况下，仅能通过服务使用者和提供者是否达成一致协议来判断 WSC 方案是否可行，缺乏评价模型。

5 结束语

本文从 WSC 的概念、实现框架、典型方法和评价模型 4 方面对研究现状进行了综述。根据 WSC 方法所依赖的技术基础，将其分为：基于工作流、状态演算以及进程代数模型的过程驱动的 WSC 方法和语义驱动的 WSC 方法。

WSC 作为计算机科学与软件工程的新领域，尚存在两个层面的基础问题：(1)实现层。尽管 OWL-S 进展迅速，但 WSC 仍缺乏统一的、语义良好的语言规范，跨组织的 WSC 实现缺乏基础支持。(2)评价层。WSC 缺乏完善的评价模型。此外，WSC 在应用实现中须处理动态变化的资源环境，不仅要研究 WSC 方案的生成，更要确保组合方案的有效和高效性，如在组合方案执行中对突发状态(如资源失效)的协调处理等。

参考文献

- [1] Zeng Liangzhao, Benatallah B, Ngu A, et al. QoS-aware Middleware for Web Services Composition[J]. IEEE Trans. on Software Engineering, 2004, 30(5): 311-327.
- [2] Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition[C]//Proceedings of the 14th Australasian Database Conference on Database Technologies. Adelaide: ACM Press, 2003.
- [3] Casati F, Ilnicki S, Jin Lijie, et al. Adaptive and Dynamic Service Composition in eFlow[C]//Proc. of the International Conference on Advanced Information Systems Engineering. Stockholm: Springer-Verlag, 2000: 13-31.
- [4] Han Yanbo, Zhao Zhuofeng, Li Gang, et al. CAFIS: An Approach to Enabling Adaptive Configuration of Service Grid Applications[J]. Journal of Computer Science and Technology, 2003, 18(4): 485.
- [5] 唐宇, 何凯涛, 陈萃, 等. 空间信息栅格体系与服务聚合技术[J]. 国防科技大学学报, 2005, 27(2): 46-51.
- [6] 陶晓峰, 孙健. 基于面向对象 Petri 网的 Web 服务合成[J]. 计算机应用, 2005, 25(6): 1424-1426.
- [7] Pahl C. A PiCalculus Based Framework for the Composition and Replacement of Components[EB/OL]. (2001-01-09). <http://www.cs.iastate.edu/~leavens/SAVCBS/papers-2001/cover.pdf>.
- [8] 廖军, 谭浩, 刘锦德. 基于 Pi-演算的 Web 服务组合的描述和验证[J]. 计算机学报, 2005, 28(4): 635-643.
- [9] Ponnekanti S R, Fox A. SWORD: A Developer Toolkit for Web Service Composition[EB/OL]. (2002-10-22). <http://swig.stanford.edu/pub/publications/sword/www11.pdf>.
- [10] 李曼, 王大治, 杜小勇, 等. 基于领域本体的 Web 服务动态组合[J]. 计算机学报, 2005, 28(4): 644-650.