

# 保障运行时适应性的 web 服务组合框架研究<sup>①</sup>

## Research on Framework for Run-time Adaptability of Web Services Composition

李昌盛 周 娅 黄桂敏 (桂林电子科技大学 计算机与控制学院 广西桂林 541004)

**摘 要:** 在 Web 服务组合过程中,几乎每时每刻都有新的 web 服务加入和退出,服务质量也处于动态变化中,本文分析了在 Web 服务组合过程中存在的问题,提出了保障运行时适应性的 web 服务组合框架,并给出了相应的服务替换算法和异构消息匹配算法,有效地解决了 Web 服务组合运行时的适应问题,并且通过扩展微软工作流框架实现了这一框架。

**关键词:** 适应性 Web 服务 组合 服务质量 运行时

各式各样的 web 服务的涌现,给企业提供了空前的机遇,随着 web 服务应用的发展,将单个 web 服务组合成新的复合服务,以适应用户不断变化发展的新需求,已经成为一个趋势。在这样一个开放的系统中,几乎每时每刻都有新的 web 服务加入和退出。而有些能够持续提供服务的 web 服务,其服务质量也因为各种原因(例如,底层网络情况,并发请求用户数目)处于不断的变化中。例如,在节假日旅游高峰期,预定机票的 web 服务可能会因为请求太多而暂时无法响应;流媒体服务可能会因为底层网络的变化而无法提供足够的传输速率。因此必须在 web 服务组合过程中保证参与服务的服务质量。

### 1 引言

文献<sup>[1]</sup>提出了一种分级的解决方案,通过在不同级生成多工作流,基于反馈机制和相配等级函数(suitable ranking)有选择的部署它们以实现适应性。文献<sup>[2]</sup>设计了一个基于策略的中间件 MASC, MASC 中间件的策略采用扩展后的 WS - Policy 描述,并实现在 .NET3.0 平台上。文献<sup>[3]</sup>提出了一种规则驱动的组装方法,实现了一个基于软件 Agent 的框架,它允许用户通过定义一组可插拔(pluggable)的自适应策略,扩展

流程对变化的适应能力。文献<sup>[4]</sup>给出了一种基于反射技术的 Web 服务失效处理方法,通过在 Web 服务的客户方和服务器方配置反射层使之捕捉到造成服务失效的各种影响状态,调整应用服务的内部结构和运行状态,使得服务适应环境的变化,以提高应用服务的适应性和健壮性。

本文提出了保障运行时适应性的 web 服务组合框架,重点关注 web 服务组合运行时 web 服务服务质量的变化。通过提出的服务替换算法,替换失效的和和服务质量不能达到要求的 web 服务。解决的问题可分为两种。(1)参与组合的 web 服务失效。web 服务失效可以分为绑定失效、调用失效和执行失效<sup>[5]</sup>,失效的原因是服务器失效或服务器资源受限。(2)参与组合的 web 服务提供的服务质量不能达到要求,例如,对于流媒体业务来说,数据率过低是不能忍受的。

我们将算法实现在微软的 Windows Workflow Foundation(WF)<sup>[6]</sup>平台上,WF 是一个通用的可扩展工作流框架,我们通过扩展 WF,将服务替换算法和异构消息匹配算法包装在定制服务中,从而实现了保障运行时适应性的 web 服务组合框架。

论文第 2 部分给出了框架结构,并描述了服务替换算法和异构消息匹配算法,框架实现在第 3 部分。

① 基金项目:广西区自然科学基金项目(桂科自 0447091)

最后是结束语。

## 2 保障运行时适应性的 web 服务组合框架

### 2.1 概述

保障运行时适应性的途径如下:

(1) 参与组合的 web 服务失效或者服务质量不能达到要求时, 查找功能相似的 web 服务集合, 针对 web 服务非功能属性<sup>[7]</sup>进行评估, 选择最优 web 服务。代替原服务完成组合过程。

(2) 替换的 web 服务有可能存在消息异构<sup>[6]</sup>, 通过异构消息匹配算法解决上一级服务的输出和下一级服务的输入消息的异构问题。

web 服务组合框架应在数据库中记录每个 web 服务的每次组合的服务质量信息, 如果一个 web 服务多次不能提供合适的服务质量, 应该予以替换, 相反地, 如果故障率在某种情形下可以承受 (例如, 书店的书籍查询服务偶尔出现不能访问的故障), 则不予以替换。

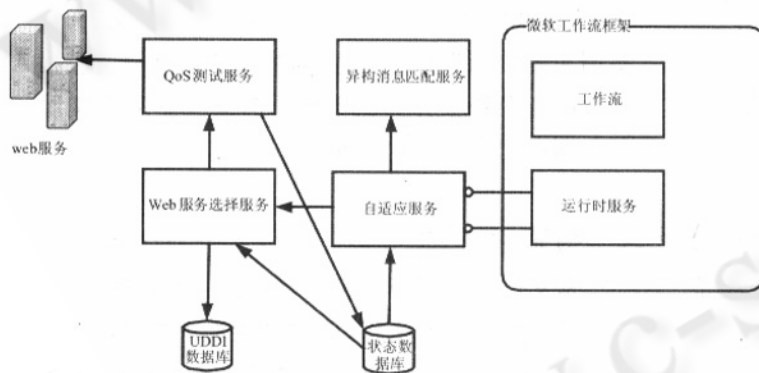


图 1 保障运行时适应性的 web 服务组合框架

如图 1 所示, 保障运行时适应性的 web 服务组合框架建立在微软的工作流基础框架之上, 通过扩展 WF, 将适应性服务添加到 WF 的运行时服务中。从而实现运行时的适应性。适应性服务包括三个子服务, 分别是: web 服务选择服务、QoS 测试服务和异构消息匹配服务, 这三个服务都包装为 web 服务。

适应性服务通过作为 WF 的运行时服务执行, 监控工作流运行时的情况, 得到当前运行的工作流的 web 服务失效, QoS 变化信息, 当出现服务失效和 QoS 无法达到要求的情况时, 适应性服务调用 web 服务选

择服务, web 服务选择服务通过服务替换算法, 首先查询 UDDI 数据库, 找到提供相似功能的 web 服务集合, 然后在集合当中选择最合适的 web 服务。提供给适应性服务。

如果替换服务的接口与原服务相同, 适应性服务可以直接将其替换到工作流中, 然而, 由于 web 服务是由不同机构和不同人员开发的, 虽然实现的是相同的业务, 但消息结构存在很大的差异。这时适应性服务调用异构消息匹配服务, 将异构消息匹配服务插入到工作流中替换的 web 服务前后。

状态数据库存放 web 服务组合过程的状态数据, UDDI 数据库还负责存放由 QoS 测试服务得到的各 web 服务的 QoS 信息, 这些数据可以用于后继的组合。

### 2.2 服务替换算法

替换算法执行的基本流程:

(1) 查找相似功能的 web 服务。这个过程得到的结果应该是一个包含相似功能 web 服务的集合。

(2) 选择最合适的 web 服务。选择的过程参考服务的 QoS 和其他非功能属性。

由于保障运行时适应性的 web 服务组合框架保证的是 web 服务组合运行时的适应性, 因此服务替换算法应保证正确性和效率。

我们采用基于服务类型和语义相似度计算的方法查找相似功能的服务, 首先进行服务类型的匹配, 筛选出和请求服务分类无关的服务, 得到候选服务集合, 通过计算待替换服务和网络中的候选服务在输入参数、输出参数、前置条件和结果的相似度, 得到总体的语义相似度。具体的相似度计算方法将在另一篇文章中叙述。

web 服务选择关注于如何从一系列基本符合用户功能需求的服务中选择出用户最满意的服务。本文的 web 服务选择算法主要解决了在选择 Web 服务的时候, 总体最优的服务可能存在单个属性值恶化的问题。有时候这种恶化是用户不能忍受的。

我们将 Web 服务非功能属性分为过滤性属性和排序性属性<sup>[7]</sup>, 并分别加以处理。为了保证在选取用户最满意的 Web 服务的时候, 某个属性不至于太过恶劣。引入了目标达成度<sup>[9]</sup>。目标达成度的计算公式

如下:

设  $f_i^{min} = \min_{x \in X} f_i(x), i=1,2,\dots,N$ , 则有:

$$\mu_i(f(x)) = \frac{\Delta}{\bar{f}_i - f_i^{min}} \quad (1)$$

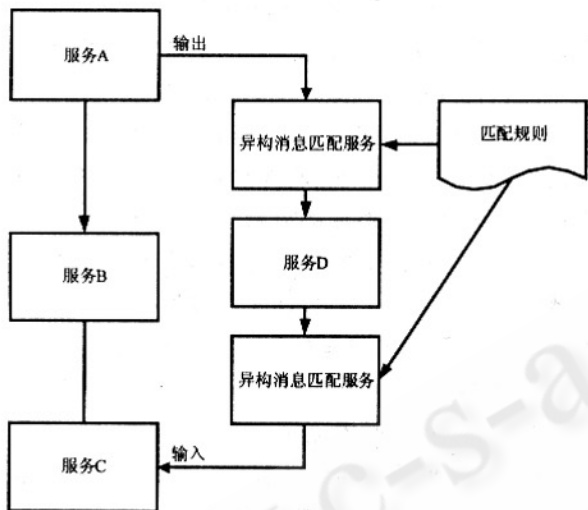


图 2 异构消息匹配示意图

为目标方案  $f(x)$  在目标  $f_i$  的达成度<sup>[9]</sup>, 其中  $f_i$  为某个目标,  $\bar{f}_i$  为目标  $f_i$  的期望水平。

通过计算各服务与目标向量的欧氏距离, 计算公式如下:

$$d_i = \sqrt{\sum_{j=1}^n (a_{ij} - a_j)^2} \quad (2)$$

其中  $a_{ij}$  代表服务  $i$  的第  $j$  属性,  $a_j$  是用户设定的第  $j$  属性的目标值。算法将选择出与目标向量的欧氏距离最小, 并且符合目标达成度的服务。

### 2.3 异构消息匹配算法

由于参与组合的服务是由不同的组织和不同的人员开发的, 因此会出现消息异构问题, 替换算法选择出来的服务的输入消息与上一级服务的输出消息, 其输出消息与下一级服务的输入消息, 都可能存在异构。

在图 2 中, 服务 B 失效, 替换它的服务 D 的输入消息与服务 A 的输出消息存在异构, 服务 D 的输出消息与服务 C 的输入消息也存在异构, 因此需要在服务 D 的前后调用异构消息匹配服务。

异构消息匹配服务是一个基于 XML 的转换服务, 它发布为 web 服务, 因而可以很方便的调用。消息的匹配本质上是不同格式的 XML 数据之间的匹配, 通过 XSLT (Extensible Stylesheet Language - Transformer) 语

言<sup>[10]</sup>可以方便的进行转换, 它是声明性语言, 本身只包含了一些转换规则的文档, 这些文档可以被递归地应用到匹配过程中。

包装为 web 服务使得异构消息匹配服务可以独立于 web 服务组合的流程, 并不限于这一个框架之中。它可以被其他支持 web 服务的流程多次调用。

### 3 框架实现

Microsoft Windows Workflow Foundation (WF) 是一个可扩展框架, 用于在 Windows 平台上开发 workflow 解决方案, Microsoft Windows Workflow Foundation 是一个广泛且通用的 workflow 框架, 并且从下到上、在每个级别都针对可扩展性进行了设计。我们的方法是扩展 Microsoft Windows Workflow Foundation 的运行服务, 通过自定义的适应性服务, 保障 workflow 实例运行时的适应性, 其原理如图 3 所示。

适应性服务监控 workflow 的运行状态, 当一个对 Web 服务的调用出现失效, 或者其 QoS 无法忍受时, 适应性服务调用服务替换服务将其替换, 并在数据库中记录相关信息。当出现接口不匹配的情况时, 还需要调用异构消息转换服务, 异构消息转换服务被插入到 workflow 中新服务和上下级服务之间。从而实现了 Web 服务组合运行时的适应性。

### 4 结束语

本文讨论了 Web 服务组合运行时的自适应问题, 重点关注服务失效和服务无法提供要求的 QoS 两个问题, 并给出了一个保障运行时适应性的 Web 服务组合框架。当发生服务失效和服务无法提供要求的 QoS 的问题时, Web 服务组合框架调用 Web 服务替换服务, 替换掉出现问题的服务, 并且给出了选择最优 Web 服务和消息异构的解决方案。相应算法均实现在微软 workflow 框架 (WF) 之上。

下一步的工作是扩展 Web 服务组合框架, 使其支持更复杂的自适应行为, 进一步研究服务替换算法和异构消息匹配算法。

### 参考文献

- 1 Girish Chafle, Koustuv Dasgupta, Arun Kumar. Adaptation in Web Service Composition and Execution, IEEE

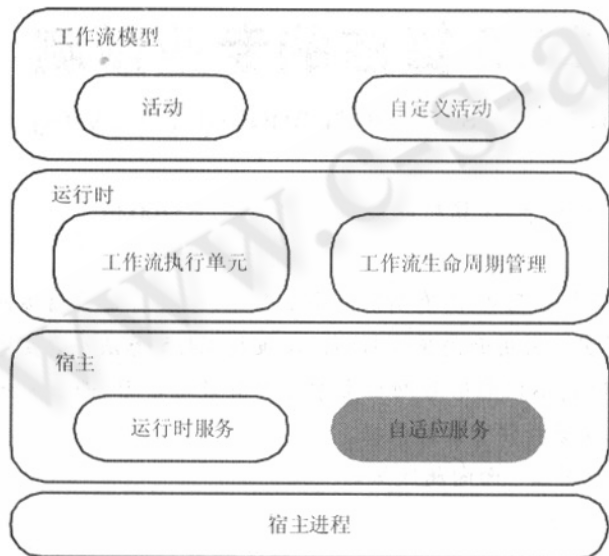


图 3 框架实现

- International Conference on Web Services (ICWS06).
- 2 Erradi, V Tomic, P Maheshwari. MASC - .NET - Based Middleware for Adaptive Composite Web Services. IEEE International Conference on Web Services (ICWS07).

- 3 孙熙、刘谭哲、焦文品、黄罡、梅宏, 一种规则驱动的网络服务组装机, 计算机学报, 2006 年 7 月.
- 4 徐新卫、周良、丁秋林, Web 服务失效处理的反射中间件技术应用与实现, 系统工程、电子技术, 2007 年 8 月.
- 5 徐新卫、王有远、曹永忠等, 基于反射技术的 Web 服务失效处理, 计算机工程与应用, 2007, 43(12).
- 6 Windows Workflow Foundation <http://msdn2.microsoft.com/en-us/netframework/aa663328.aspx>.
- 7 M Kerrigan. Web service selection mechanisms in the Web Service Execution Environment (WSMX), Proceedings of the 2006 ACM symposium on Applied computing, 2006.
- 8 李涛、李涓子、王克宏, Web 服务异构消息匹配及重用, 计算机学报, 2006 年 7 月.
- 9 蒋尚华、徐南荣, 基于目标达成度和目标综合度的交互式多目标决策方法, 系统工程理论与实践, 1999 年 1 月.
- 10 XSL Transformations (XSLT) Version 1.0. <http://www.w3.org/TR/xslt>