

# Web 服务组合事务处理研究与实现

管有庆,程 强

(南京邮电大学 物联网学院 信息技术研究所,江苏 南京 210003)

**摘要:**通过重用已有的服务进行服务组合来实现应用集成;而保证服务组合的事务特性是实现服务组合的关键。文中提出一个 Web 服务组合事务处理模型,将 BPEL (Business Process Execution Language, 业务流程执行语言) 的流程设计与 WS-C/T (Web Services Coordination/Transaction, Web 服务协调/事务) 规范结合。该模型主要由客户端、BPEL 执行引擎、事务协调器和参与者四个部分组成。客户端发起请求, BPEL 执行引擎响应请求并开启事务,事务协调器和参与者根据事务协调算法完成事务协调。

**关键词:**服务组合;业务流程执行语言;事务;协调

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2013)11-0077-05

doi:10.3969/j.issn.1673-629X.2013.11.020

## Research and Implementation of Web Service Composition Transaction Processing

GUAN You-qing, CHENG Qiang

(Institute of Information Network Technology, College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** To realize application integration, existing services are used to achieve service composition, and the key to achieve service composition is to guarantee transactional properties of service composition. In this paper, a Web service composition transaction model is introduced, which combines BPEL (Business Process Execution Language) process design with transaction protocol based on WS-C/T (Web Services Coordination/Transaction). This model is chiefly composed of clients, BPEL execution engine, transaction coordinator and participants. The client initiates a request, waiting for BPEL execution to give response and to start a transaction. Then the transaction coordinator and the participants complete the coordination by means of transaction coordination algorithm.

**Key words:** service composition; BPEL; transaction; coordination

### 1 概述

随着 Internet (因特网) 技术的快速发展,传统单一的 Web 服务已经难以满足应用的需求,将多个服务组合起来完成跨系统、跨平台的复杂的业务逻辑越来越受到关注。这种通过将多个 Web 服务整合成为一个功能更为完善的服务的行为被称为 Web 服务组合<sup>[1]</sup>。为了保证 Web 服务组合数据一致性,Web 服务组合应遵守 Web 服务事务规范。Web 服务事务规范有 BTP (Business Transaction Protocol, 业务事务协议)<sup>[2]</sup>, WS-C/T (Web Services - Coordination /Transaction, Web 服务协调/事务) 规范<sup>[3]</sup> 和 WS-CAF<sup>[4]</sup> (Web Services

Composite Application Framework, Web 服务组合应用框架)。BTP 采用两阶段提交协议,以确保事务的一致性。WS-C/T 由 WS-C 与 WS-T 组成,其中 WS-C 协议提供了一个协同 Web 服务达成事务一致性的框架,WS-T 协议包含 WS-AT (Web Services Atomic Transaction, Web 服务原子事务) 与 WS-BA (Web Services Business Activity, Web 服务业务活动),它可以分别用于处理原子事务和业务事务。尽管 WS-C/T 能够用于处理 Web 服务组合事务,然而它并没有对 Web 服务组合中的业务流程进行描述。WS-CAF 类似于 WS-C/T,但在实际应用系统中很少用到。由

收稿日期:2013-01-20

修回日期:2013-04-23

网络出版时间:2013-08-28

基金项目:江苏省高校自然科学基金研究计划项目(05KJD520146)

作者简介:管有庆(1963-),男,副研究员,硕士生导师,研究方向为数据库、通信软件和下一代网络等;程 强(1988-),男,江西宜春人,硕士研究生,研究方向为软件技术在通信网络中的应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130828.0837.027.html>

IBM 和 Microsoft 联合提出了一项专为整合 Web 服务的规范 BPEL (Business Process Execution Language, 业务流程执行语言), 提供了 Web 服务事务规范中所不具有的工作流的框架。文献 [5] 提出利用 BPEL 实现 Web 服务的互操作。David Beech 等人 [6] 提出将工作流的概念引入到 Web 服务当中, Web 服务和 workflow 技术相结合, 允许 workflow 中任意的任务用 Web 服务的方式来实现, 并通过 workflow 实现 Web 服务的组合来完成特定的业务逻辑。文献 [7] 中提出使用更为复杂的协调补偿机制替换 BPEL 规范中的本地补偿机制, 通过 SOAP (Simple Object Access Protocol, 简单对象访问协议) 消息完成服务间的交互, 从而取代类似于 WS-BA 的协调作用, 然而该方法并没有充分利用 WS-BA 机制协调事务的优势, 并且实现协调补偿机制较为复杂。Chintan Patel 等人 [8] 提出了在已有业务流程执行过程中动态绑定具体服务来满足用户需求的方法, 它降低了因服务的动态变化而导致系统执行失败的可能性, 提高了系统的自适应性, 但是该方法没有提及因业务流程执行失败而引起的撤销所带来的影响。

## 2 Web 服务组合事务模型

### 2.1 Web 服务组合事务概念

BPEL 是由 Microsoft 提出的 XLANG (XML-based extension of Web Services Description Language, 基于 XML 的 Web 服务描述语言扩展) 和 IBM 提出的 WSFL (Web Services Flow Language, Web 服务流语言) 两种 workflow 语言演化而来的。BPEL 提供了 receive (接收)、invoke (调用) 等基本活动以及 sequence (顺序)、if (分支)、flow (并行)、pick (选择)、while (循环) 等控制流程, 组合这些活动与流程可以描述面向服务的业务流程, 实现 Web 服务之间的交互。BPEL 还提供了包含有事务性质的 scope (作用域), 通过作用域机制, BPEL 可以对作用域中已完成的操作进行补偿, 达到撤销的效果。但是作用域机制缺乏灵活性, 在作用域定义了允许一起被撤销的活动集, 活动集中的活动要么全部完成, 要么全部补偿。而在实际的应用中往往需要允许部分参与者完成, 部分参与者补偿。所以仅仅通过 BPEL 的流程不能满足快速、灵活的服务组合事务的需求。

WS-BA 是 WS-C/T 中用于处理业务事务的子协议。它为业务事务提供了两种完成模式: 参与者完成模式和协调者完成模式。另外, WS-BA 引入了补偿机制, 参与者在事务执行前定义并实现补偿逻辑, 当事务在撤销时, 参与者可以依据补偿逻辑恢复到事务执行前的状态。然而 WS-C/T 并没有提供对业务流程的描述, 在实际应用中采用硬编码的方式实现业务流程,

将增加实现难度 [9]。WS-C/T 具备服务组合中进行事务协调的优势, 而 BPEL 满足服务组合中对业务流程描述的需求, 所以将两者结合描述 Web 服务组合事务模型。

### 2.2 Web 服务组合事务模型

Web 服务组合事务模型由客户端、BPEL 执行引擎、事务协调器、参与者四个部分组成, 其中, BPEL 执行引擎与事务协调器协同扮演协调者的角色。图 1 为 Web 服务组合事务模型结构图。

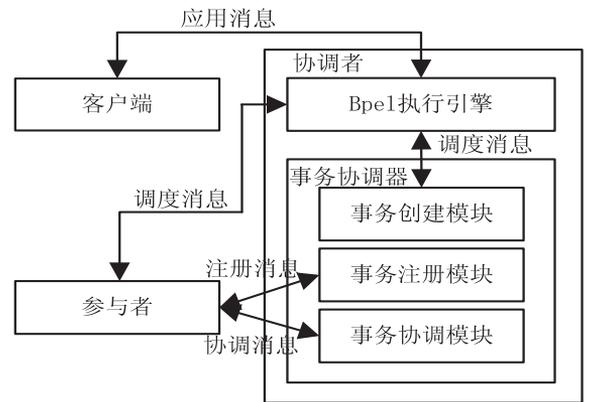


图 1 Web 服务组合事务模型结构图

1) 客户端: 由客户端应用程序通过应用消息向 BPEL 执行引擎发起事务请求。

2) BPEL 执行引擎: BPEL 执行引擎响应客户端请求并开启事务, 与事务协调器、参与者进行消息交互, 完成事务执行过程。

3) 事务协调器: 事务协调器包含了按照 WS-C 实现的三个功能模块, 这三个功能模块将与 BPEL 执行引擎共同协调完成事务。三个功能模块具体描述如下:

(a) 事务创建模块: 提供激活服务。用于接收 BPEL 执行引擎创建事务的请求, 为新事务生成一个包含事务 ID 号、协调类型和注册服务地址等信息的协调上下文, 然后将协调上下文返回给 BPEL 引擎。

(b) 事务注册模块: 提供注册服务。用于接收参与者发来的注册消息, 事务注册模块根据注册消息为参与者注册, 并根据注册情况向参与者返回注册结果。

(c) 事务协调模块: 提供事务协调服务。参与者根据自身服务运行状态得到一个处理结果, 并将它作为子事务状态返回给事务协调器, 最后由事务协调器根据所有子事务状态决定提交或补偿事务。

4) 参与者: 提供具体的 Web 服务。多个参与者一起实现 Web 服务组合。

### 2.3 事务协调过程

Web 组合服务是一个复杂的服务, 它是由一系列 Web 服务按照一定的业务逻辑组合而成的。由于这些服务具有自治的, 跨组织, 松散耦合并长周期运行的特

性,Web 服务事务不能严格遵循 ACID(Atomicity, Consistency, Isolation, Durability)属性。Web 服务事务分为原子事务和业务事务。

定义 1:原子事务(Atomic Transaction, AT)是类似传统的 ACID 事务。它用来协调短周期的操作。原子事务要求所有的参与者全部提交或全部中止。在提交之前,AT 将锁定资源。

定义 2:业务事务(Business Activity, BA)用于协调长周期运行的事务。它不像 AT 在提交之前锁定资源,BA 的参与者可以提交自己的子事务,则已提交的子事务的结果可被其他应用访问。

目前跨组织间协作的组合服务运行往往需要一定的时间周期,所以 BA 应用更为广泛。文中将 BA 协调机制与 BPEL 执行引擎结合,提出 Web 服务组合事务协调过程,它主要包括协调者协调和参与者协调, BPEL 执行引擎与 BA 事务协调器结合共同构成协调者,具体的 Web 服务构成参与者。图 2 为协调者状态转换图(椭圆表示状态,有向弧表示状态转换,有向弧上的标记表示发出或收到消息)。

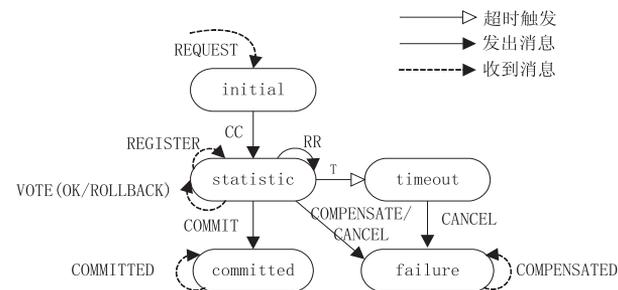


图 2 协调者状态转换图

协调者收到用户应用请求后开启事务,协调者进入 initial (初始化)状态,设置定时器 T (最大等待时间),通过协调者内的激活服务获得 CC (Coordinator-Context, 协调上下文),将 CC 发送到事务中的参与者,协调者进入 statistic (统计)状态,它接收参与者发来的 REGISTER (注册)消息,注册成功则返回 RR (RegisterReturn, 注册返回)消息。此时协调者等待参与者根据自身完成的情况发来的 VOTE (投票)消息并进行统计投票,协调者根据最终投票情况对参与者发出相应的协调消息:如果参与者的投票情况都为 VOTE (OK) (成功执行),则向参与者发送 COMMIT (提交)消息,协调者进入 committed (提交完成)状态,确认参与者 COMMITTED (提交完成)消息;如果有一个参与者的投票情况为 VOTE (ROLLBACK) (执行失败),则向已完成的参与者发送 COMPENSATE (补偿)消息,向未完成的参与者发送 CANCEL (取消)消息,协调者进入 failure (失败)状态,确认参与者 COMPENSATED (补偿完成)消息;如果定时器超时,则协调者进入

timeout (超时)状态,协调者向所有参与者发送 CANCEL 消息,协调者进入 failure 状态。

参与者的状态转换图如图 3 所示。

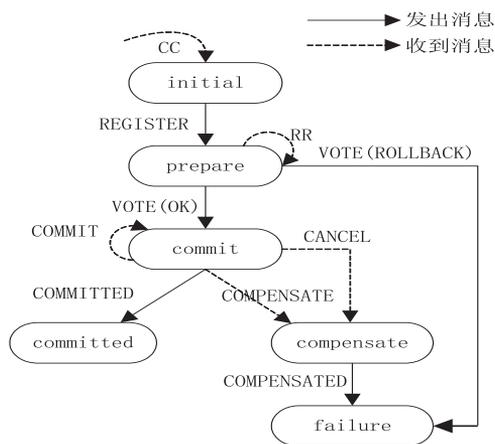


图 3 参与者状态转换图

参与者收到 CC 时,进入 initial 状态,根据 CC 中提供的注册服务地址向协调者发送 REGISTER 消息,参与者进入 prepare (准备)状态。参与者收到 RR 消息后,根据自身服务完成情况向协调者发送投票消息,若自身服务执行失败则发送 VOTE (ROLLBACK) 消息,参与者直接退出事务。若成功完成则发送 VOTE (OK) 消息,参与者进入 commit (提交)状态。此时参与者则根据协调者发来的消息进行相应的操作:如果收到的是 COMMIT 消息,则参与者进行确认,向协调者返回 COMMITTED 消息,进入 committed 状态;如果收到的是 COMPENSATE 消息,则依据自身补偿服务进行补偿操作,进入 compensate (补偿)状态,向协调者返回 COMPENSATED 消息,进入 failure 状态;如果收到的是 CANCEL 消息,参与者中止自身操作,进入 compensate 状态,依据自身的补偿服务进行补偿操作,向协调者返回 COMPENSATED 消息,进入 failure 状态。

### 2.4 事务处理流程

Web 服务组合事务处理流程如图 4 所示。

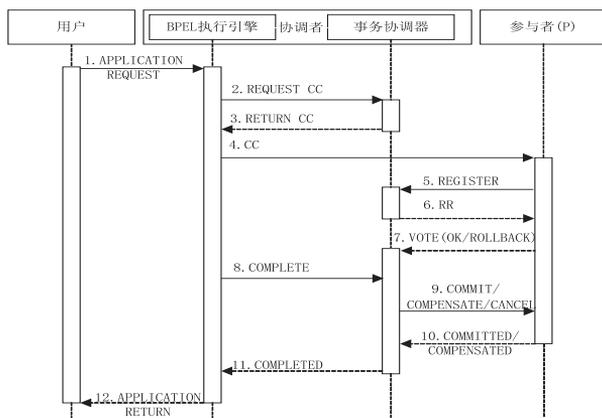


图 4 流程事务处理时序图

1) APPLICATION REQUEST: 用户向 BPEL 执行引



1) 事务完成 (Transaction is COMMITTED): 假设用户向旅行代理发送预定申请, 旅行代理成功完成了预定旅馆、预定机票和银行支付三个子事务, 整个业务事务成功提交, 并持久记录在数据库中。

2) 事务失败 (Transaction is ROLLED BACK): 假设用户向旅行代理发送预定申请后, 旅行代理完成了预定机票、银行支付两个子事务, 但执行预定旅馆子事务失败, 导致整个业务事务失败, 最后撤销整个事务, 并对预定机票、银行支付进行了补偿。

#### 4 结束语

文中将业务流程执行语言的流程设计与 WS-C/T 事务协议结合, 对 Web 服务组合事务模型进行研究与实现, 该模型能够对 Web 服务进行可靠、有效地集成, 并且具有事务补偿和故障恢复能力, 保证了事务执行结果的一致性。提出的模型支持业务事务, 将来还要进一步对混合事务 (业务事务作用域内嵌套原子事务) 进行研究, 实现复杂的事务管理; 另外可进一步研究动态的 Web 服务组合以提高服务质量<sup>[10]</sup>。

#### 参考文献:

[1] 武云鹏, 包卫东, 张维明, 等. Web 服务组合系统研究综述 [J]. 计算机科学, 2011, 38(9): 1-4.  
 [2] OASIS. Business transaction protocol (BTP) version 1.0 [EB/OL]. 2002-06. <https://www.oasis-open.org/committees/>

business - transaction/documents/primer/Primerhtml/BTP%20Primer%20D1%2020020602.html.

- [3] BEA, IBM and Microsoft. Web services coordination (WS-Coordination) [EB/OL]. 2009-02. <http://docs.oasis-open.org/ws-tx/wstx-wscoord-1.2-spec-os/wstx-wscoord-1.2-spec-os.html>.  
 [4] Aljuna, Fujitsu, IONA, et al. Web services composite application framework (WS-CAF) [EB/OL]. 2003-07. <http://xml.coverpages.org/WS-CAF-Primer200310.pdf>.  
 [5] Mandell D J, McIlraith S A. Adapting BPEL4WS for the semantic web; the bottom-up approach to web service interoperability [C]//Proc of ISWC. [s. l.]: Springer, 2003: 227-241.  
 [6] Thompson H S, Beech D, Maloney M, et al. XML schema-part 1: structures second edition [EB/OL]. 2004-10. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.  
 [7] Sauter P, Melzer I. A comparison of WS-business activity and BPEL4WS long-running transaction [C]//Proc of KiVS. [s. l.]: Springer, 2005: 115-125.  
 [8] Patel C, Supekar K, Lee Y. Provisioning resilient, adaptive Web services-based workflow: a semantic modeling approach [C]//Proc of IEEE international conference on web services. [s. l.]: IEEE, 2004: 480-487.  
 [9] 何演, 管有庆. 基于 WS-C/T 协议的 Web 服务业务事务处理研究与实现 [J]. 计算机技术与发展, 2011, 21(4): 90-93.  
 [10] 吴钊. 保证服务质量的动态 Web 服务组合及其性能分析研究 [D]. 武汉: 武汉大学, 2007.

(上接第 76 页)

(1): 25-38.  
 [5] Ruud S, Owen H, Janet B. Ant-like agents for load balancing in telecommunications networks [C]//Proceedings of the first international conference on autonomous agents. Marina del Rey, California, United States: [s. n.], 1997.  
 [6] Caro G D, Dorigo M. AntNet: a mobile agents approach to adaptive routing [R]. Bruxelles: Universite Libre de Bruxelles, 1997.  
 [7] Hussein O H, Saadawi T M, Jong L M. Probability routing algorithm for mobile ad hoc networks resources management [J]. IEEE journal on selected areas in communications, 2005, 23(12): 2248-2259.  
 [8] Gomez J F, Khodr H M, de Oliveira P M, et al. Ant colony system algorithm for the planning of primary distribution circuits [J]. IEEE transactions on power systems, 2004, 19(2): 996-1004.  
 [9] Zecchin A C, Simpson A R, Maier H R, et al. Parametric study

for an ant algorithm applied to water distribution system optimization [J]. IEEE transactions on evolutionary computation, 2005, 9(2): 175-191.

- [10] Giannakis G B, Hua Y B, Stoica P, et al. Signal processing advances in wireless and mobile communications, volume 1: trends in channel estimation and equalization [R]. Upper Saddle River, N J: Prentice Hall PTR, 2001.  
 [11] Ding Z, Li Y. Blind equalization and identification [M]. New York: Marcel Dekker, 2000.  
 [12] Slock D. Blind fractionally-spaced equalization perfect-reconstruction filter banks and multichannel linear prediction [C]//Proc of ICASSP. [s. l.]: [s. n.], 1994: 585-588.  
 [13] Ding Z. Matrix outer-product decomposition method for blind multiple channel identification [J]. IEEE transactions on signal processing, 1997, 45(12): 3054-3061.  
 [14] 张志涌, Bai Erwei. SIMO 含公零点信道的直接盲序列检测 [J]. 电子学报, 2005, 33(4): 671-675.

# Web服务组合事务处理研究与实现

作者: 管有庆, 程强, GUAN You-qing, CHENG Qiang  
作者单位: 南京邮电大学 物联网学院 信息网络技术研究所, 江苏 南京, 210003  
刊名: 计算机技术与发展

---

ISTIC

英文刊名: Computer Technology and Development

---

年, 卷(期): 2013(11)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201311021.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201311021.aspx)