

# Web 服务失败分类法

唐 渊, 金可音, 周 昆, 宋春阳

(湖南工业大学 计算机与通信学院, 湖南 株洲 412008)

**摘 要:** 在分析了当前失败分类法的基础上, 提出了一种 Web 服务分类方法来指明在 Web 服务中可能的失败, 并将失败分类处理。这样能够在失败发生时尽可能隔离失败, 减小失败带来或者可能导致的损失, 从而提高服务的可靠性。

**关键词:** Web 服务, 失败, 失败分类, 自适应

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1673-9833(2009)02-0073-04

## Fault Taxonomy Method for Web Services

Tang Yuan, Jin Keyin, Zhou Kun, Song Chunyang

(School of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan 412008, China)

**Abstract:** After analyzing current fault taxonomy, a new faults taxonomy about web service is introduced which can point out possible fault and deal with the fault with its method in web services. In this way, it can segregate fault and reduce losing as well as bring up its reliability of the service.

**Key words:** Web Services; fault; fault taxonomy; self-healing

## 1 Web 服务失败简介

Web 服务的研究重点在于发展语言 and 标准来明确和控制业务过程, 其中 BPEL4WS<sup>[1]</sup> (business process execution language for Web services) 是一种描述、定义及模型化业务过程的语言, OWL-S<sup>[2]</sup> (semantic Web ontology language) 是标记 Web 服务特征和性质的本体, 所有这些工作的目标都是希望 Web 服务能够自动发现、组合、执行及互操作。实际上使用者常会使用代理 (如 UDDI<sup>[3,4]</sup>—— universal description, discovery and integration) 及自适应和自管理方法来满足服务执行的松耦合性, 与此同时产生了大量失败。

针对出现的失败, 研究者们提出通过实时检测和互操作以达到组合的自适应<sup>[5]</sup>。此外, 还提出了服务质量模型<sup>[6,7]</sup>, 模型检测及核实<sup>[8]</sup>, 及失败修复<sup>[9]</sup>。现在在该方向主要研究集中在如何在 Web 服务执行过程中自动检测失败及修复失败, 即如何提高平台的自适应

性。因此, 在 Web 服务组合的研究工作中<sup>[10]</sup>如何辨别失败成为了一个关键点, 只有在分清楚失败的基础上才能够提供合适的修复机制。这样我们就需要了解所有可能发生的失败以及产生失败的原因, 以便对失败进行分类处理, 从而提高服务执行的可靠程度。

为此, 笔者在 SOA (service-oriented architecture) 的失败分类<sup>[11,12]</sup>基础上列举并执行 Web 服务的关键步骤, 针对各个关键步骤所产生的失败寻求它产生的原因, 并按其原因归类。本文的目标是提出一种新的分类法来捕获导致 Web 服务错误的原因, 在此, 将失败定义为组件出现的非正常情况或者组件的缺陷。失败比错误 (error) 低一个级别, 但是它可能导致错误<sup>[13]</sup>, 我们的分类法实例取自于文献<sup>[10]</sup>提供的平台。

## 2 Web 服务执行步骤中的失败

Web 服务主要步骤有发布 (publishing)、组合

收稿日期: 2008-12-26

基金项目: 湖南省自然科学基金资助项目 (05JJ30122), 湖南省教育厅高等学校科研基金资助项目 (04C720)

作者简介: 唐 渊 (1982-), 男, 湖南株洲人, 湖南工业大学硕士研究生, 主要研究方向为 WebServices。

E-mail: tto tto tto-1981@163.com

(composition)、绑定(binding)以及执行(execution),失败则可能出现在Web服务的每一步上,他们将导致错误而错误将导致调用失效等,如果Web服务系统结构有能力解决这些错误将会提高Web服务的可靠程度。以下是各执行步骤中可能出现的失败。

**发布失败(publishing faults):**在发布过程中,服务在服务器中发布其描述并展开才可以执行。服务描述失败与声明失败相似,它们发生在服务描述不正确的时候。描述本身可能有失败(描述不正确)或者描述不能够与展开服务匹配(服务描述不匹配)。

**组合失败(composition faults):**对于单个服务来说可能就是没有找到服务或者是找到了错误的服务;对于组合服务来说,组合过程失败即是说组合代理返回结果是没有合适的组合(no valid composition),这类失败可能有很多原因造成,比如不兼容的组件、没有连接等等。

**绑定失败(binding faults):**在绑定过程中,服务申请者和服务提供商相互协调状态来执行服务。绑定失败发生在组件开发商否定该过程的时候。即发生提供商否定,证明失败或者账户问题时就会产生绑定失败。

**执行失败(execute faults):**执行失败发生在服务被执行但是结果不匹配预期输出的时候,如服务崩溃或者不正确的结果。

### 3 Web 服务失败的层次

Web服务的语言被进一步细化用来发现失败,都是寻找失败的效果以及可能产生该结果的原因,并将信息发给恢复系统,恢复系统针对不同层次的失败可以使用不同的方法进行恢复。明确失败的层次有助于抽象该问题。基于文献[13]的工作,我们将失败分为3个主要的相互交叉的层次级别:0级失败(物理失败),1级失败(开发性失败),2级失败(互操作失败)。

#### 3.1 0级失败

0级失败包括基础(物理)连接失败或者是服务主机不可用。举例:一个服务处于不活动状态,这个时候服务就会出现服务不可用。服务不可用有两种情况——服务关闭或者网络连接关闭。要能够精确的区分什么原因造成物理失败比较难,但是上述原因导致的失败一定会造成没有应答。只要没有应答就可以将他们分类到服务不可用这个失败中来,而且这个失败以及其它主机上的失败都只能由提供商来修复。

#### 3.2 1级失败

1级失败与系统的开发环境有关,特别是个人开发者,开发工具以及工具生产厂商都可能影响开发效果。1级失败可能导致服务部分的或者全部的错误,这种错误可能要到的时候才能够被发现。

参数不兼容是主要的一种1级失败,它是指服务

收到一个不正确的描述或者是一个不正确输入参数类型。在正常环境下,如果能够阻塞一个失败或者对失败进行异常修复,那么这种失败是可以避免的,而该失败就是可以避免的。

通常服务选取是基于接口或本体描述的。如果服务是基于接口的,那么在服务重新修改了接口之后或许会导致使用原有接口的后续队列产生失败。此时,将导致工作流程不一致(workflow inconsistency)问题。这种情况下服务不能被调用,因为实际调用的接口与工作流程中描述的接口不一致。如果服务使用者不能够发现服务接口改变,他们就会认为服务已经被破坏或不再提供服务。这种类型的失败和0级失败导致的结果非常相似,甚至可能被认为就是一种0级失败,除非服务在错误发生时能返回一个可感知的错误消息。

#### 3.3 2级失败

2级失败可以包含两个大类。一个大类是满意度失败:包括不正确的服务,不可理解的行为,应答错误,服务质量(quality of service, QoS)和服务级别(service level agreement, SLA)失败。满意度失败是基于服务描述的,当服务传输内容偏离了预期效果时,发生该失败。

应答失败和行为失败与不兼容的输入有很大的关系,Web服务如果有这样的失败就有可能在获得正确输入的情况下发送错误输出,这种错误输出可能是由服务内部错误的逻辑导致的,部分有误的服务甚至可能由一个申请产生随机的结果,他们通常导致不能理解的服务行为和后续服务错误的输入值。

Web服务的非功能方面包括服务级别(SLA)和服务质量(QoS)。这种非功能的属性能够在执行的时候捕获,也可以通过查看不同部分之间的接口来监视服务的表现。

SLA可以用来保留服务的计算效率<sup>[4]</sup>。在SLA中或者是由SLA引起的失败将导致申请者收到不正确的服务,SLA协商中所包含的部分都要接受提供商所提供的服务以及接口,如果提供商提供的服务不是通用的,那么就会产生SLA不一致的情况。QoS包括了服务质量所要求的传输速度和信息量,如果申请者对于这些部分都不是特别满意(比如速度很慢或者有服务应答延时)我们就认为服务的QoS产生失败。

另一个大类是计时失败:包括不正确的提供者,超时,不能理解的业务流程。

计时失败与服务访问时间或者服务传输时间有关,它可以导致系统中的结果偏离原有正常应答时间。组合服务的执行需要联系所有被组合的Web服务,在各个服务之间进行信息传输的过程中,信息包到达的一个服务可能并不是我们要的服务,这就是不正确提供者引起的失败,这种情况通常是由慢速网络

造成的。在组合服务中, 服务通常依靠相互协作来得到预期的结果, 当信息发送给不正确的提供者时就有可能返回非预期结果。

超时可能是由慢速网络导致的失败。如果使用一个适当恢复机制, 超时失败和连接媒体失败造成的数据丢失都能够按时被接收到。例如, 在 BPEL 中的阻塞

异常处理器就能够处理这样的失败。

### 4 失败分类及恢复策略

本文将 Web 服务执行步骤中可能出现的失败进行归类, 结果如图 1 所示。

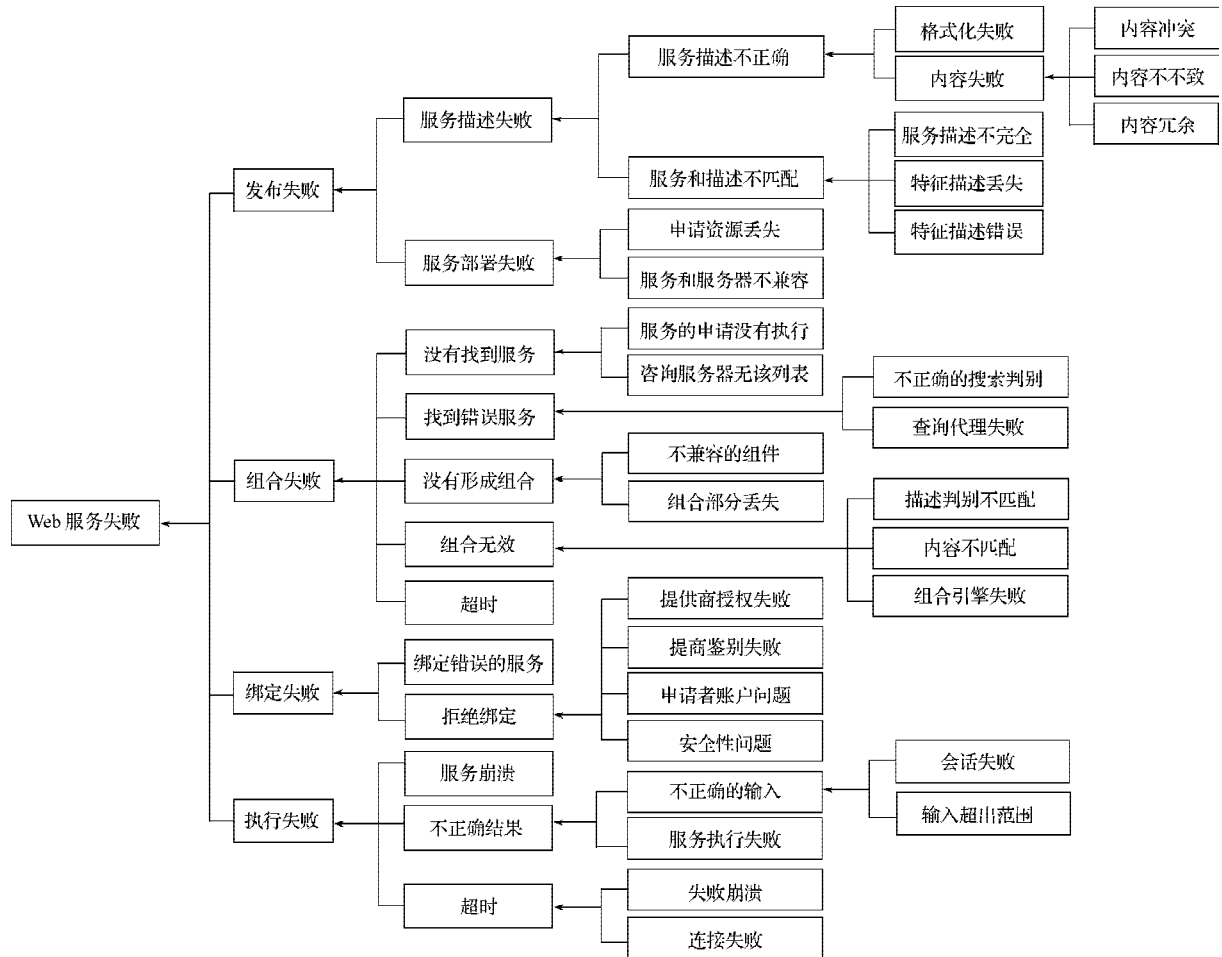


图 1 Web 服务失败分类图  
Fig. 1 The classification of Web services fault

表 1~3 详细描述了 Web 服务失败的基本分类和失败所属层次, 分析了失败产生的原因及可能导致的结果, 并提出了相应的恢复策略。表中的恢复策略是指在 Web 服务出现失败时可以采用的措施: 重试、替换、局部重构 (局部重构是指将受到单个失败影响的流程进行部分的重新构造)。同时, 表中选取了 Avižienis A 提出的失败基本分类<sup>[13]</sup>中的 4 种, 作为 Web 服务失败

的基本分类。其含义表述如下:

- 开发失败: 发生在系统开发或维护时产生的失败
- 过程失败: 发生在服务传输时产生的失败
- 内部失败: 由系统内部所引起的失败
- 外部失败: 由系统外部所引起的失败并且能够通过互操作或者接口繁殖到系统的其它服务中。

表 1 Web 服务发布和绑定过程中的失败  
Table 1 The fault in process of Web services publishing and binding

失败名称	基本分类	所属层次	失败原因	导致的结果	恢复策略
服务描述失败	开发, 内部	1 级失败	服务描述不正确, 服务 / 描述不匹配	组合失败, 绑定错误的服务, 返回错误的结果	替换
服务部署失败	开发, 内部	1 级失败	申请资源丢失, 服务器不兼容		替换
绑定错误服务	过程, 内部	2 级失败	描述错误	服务崩溃, 返回错误结果	替换
拒绝绑定服务	过程, 内部	2 级失败	授权失败, 识别失败, 账户问题, 安全性问题	服务无法执行	重试、替换

表2 服务组合过程中的失败

Table 2 The fault in process of Web services composition

失败名称	有无子类	基本分类	失败层次	失败原因	导致的结果	恢复策略
没有发现服务	有	过程, 内部	2级失败	申请没有执行, 查询不到列表	没有返回结果	重试
发现错误服务	有	过程, 外部	2级失败	查询判别失败, 查询代理失败	返回错误的结果, 组合失败	替换
没有组成服务	有	过程, 内部	2级失败	组件不兼容, 组件部分丢失	服务无法执行	局部重构
组合失败	有	过程, 外部	2级失败	内容不匹配, 判别不匹配, 组合引擎失败	组合崩溃, 服务中断执行	局部重构
超时	无	过程, 外部	2级失败	慢速网络, 不正确输入	没有形成组合	重试、替换

表3 服务执行过程中的失败

Table 3 The fault in process of Web services execution

失败名称	有无子类	基本分类	失败层次	失败原因	导致的结果	恢复策略
服务崩溃	无	开发, 内部	0级失败 1级失败	服务器错误, 软件开发错误	无法执行	替换、局部重构
不正确结果	有	过程, 外部	2级失败	不正确的输入, 服务出错	返回错误结果	替换
超时	有	过程, 外部	0级失败 1级失败	服务崩溃, 连接失败	不正确结果, 没有返回结果	重试、替换

## 5 结语

本文主要针对 Web 服务失败进行了定性的分析, 并在此基础上提出了一种新的 Web 服务的失败分类法, 该分类法涉及到失败导致的错误以及失败的层次问题, 然后指出了失败恢复的部分策略。下一步的研究重点在于: 1) 尽可能完善失败的分类, 以及他们在 Web 服务运行中的影响; 2) 失败恢复技术中的失败策略的运用, 以及如何将他们整合到恢复系统中。我们希望将来的系统能够显示可能发生的错误和错误可能导致的的影响, 还能够指导用户进行错误的恢复。

### 参考文献:

- [1] Andrews Tony, Curbera Francisco, Dholakia Hitesh, et al. Web Services Business Process Execution Language Version. 1.1[EB/OL]. [2007-02-16]. <http://www.ibm.com/developerworks/library/specification/ws-bpel/>.
- [2] David Martin, Burstein Mark, Hobbs Jerry, et al. OWL-S 1.1 Release[EB/OL]. [2004-11-08]. <http://www.daml.org/services/owl-s/1.1/>.
- [3] Ariba Inc, IBM Corporation, Microsoft Corporation. UDDI Technical White Paper[EB/OL]. [2000-09-02]. [http://www.uddi.org/pubs/Iru\\_UDDI\\_Technical\\_White\\_Paper.pdf](http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf).
- [4] Ariba Inc, IBM Corporation, Microsoft Corporation. UDDI Executive White Paper[EB/OL]. [2001-09-15]. [http://www.uddi.org/pubs/UDDI\\_Executive\\_White\\_Paper.pdf](http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf).
- [5] Baresi L, Ghezzi C, Guinea S. Towards Self-Healing Services Compositions[EB/OL]. [2007-06-09]. <http://home.dei.polimi.it/baresi/papers/PRISE04.pdf>.
- [6] Cardoso J, Sheth A P, Miller J A. Quality of Service for Workflows and Web Service Processes[J]. Web Semantics, 2004, 1(3): 281-308.
- [7] 杨胜文, 史美林. 一种支持 QoS 约束的 Web 服务发现模型

[J]. 计算机学报, 2005, 28(4): 589-594.

Yang Shengwen, Shi Meilin. A Model for Web Service Discovery with QoS Constraints[J]. Chinese Journal of Computers, 2005, 28(4): 589-594

- [8] Fu X, Bultan T, Su J. Analysis of Interacting BPEL Web Services[C]//Proc. 13<sup>th</sup> ACM Intl. Conf. on the World Wide Web. New York: Association for Computing Machinery, 2004: 621-630.
- [9] Fan X, Umaphy K, Yen J, et al. Team-Based Agents for Proactive Failure Handling in Dynamic Composition of Web Services[C]//Proc. IEEE Intl. Conference on Web Services. San Diego California: The IEEE Technical Community for Services Computing, 2004: 782-785.
- [10] Steyn J. Approaches to Failure and Recovery in Service Composition[EB/OL]. [2007-07-20]. <http://polelo.cs.up.ac.za/publications.html>.
- [11] Smidts C. Integrating Software into PRA. A Software-Related Failure Mode Taxonomy[J]. In Risk Analysis, 2006, 26(8): 997-1012.
- [12] Bruning S, Weibleder S, Malek M. A Fault Taxonomy for Service-Oriented Architecture[C]//Humboldt-University at Zu Berlin, Tech. Rep. 215. Washington: IEEE Computer Society, 2007: 367-368.
- [13] Avižienis A, Laprie J C, Randell B, et al. Basic Concepts and Taxonomy of Dependable and Secure Computing[J]. IEEE Transaction on Dependable and Secure Computing, 2004, 1(1): 11-32.
- [14] Ludwig H, Gimpel H, Dan A, et al. Template-Based Automated Service Provisioning-Supporting the Agreement-Driven Service Life-Cycle[C]//Proc. Intl. Conf. Service Oriented Computing. New York: Springer Verlag, 2005: 283-295.

(责任编辑: 李玉珍)