

基于 Java 的 SOA 集成测试模型研究

吴 蕾, 李心科, 汪 洪

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘 要:随着 Web 服务的兴起提出了 SOA 的软件体系架构,为 Internet 上的分布式计算提供了支持异构平台和多种语言的构件式程序架构,而目前几乎没有对 SOA 体系架构软件的集成测试方法及工具;文章利用植错测试技术,设计了在 Java 平台上对 SOA 架构下 Web 服务软件消息交互进行测试的 JT4SOA 模型。

关键词:Web 服务; 植错; 测试模型

中图分类号:TP311.5

文献标识码:A

文章编号:1003-5060(2007)01-0022-04

Research on testing of SOA integration based on Java

WU Lei, LI Xin-ke, WANG Hong

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: The Service Oriented Architecture(SOA) provides a component architecture which supports heterogeneous platforms and multiple languages for distributive computing on Internet. However there are few integration testing methods or tools for the SOA currently. In this paper, the JT4SOA model is designed to test message communication of Web services under the SOA on the Java platform by using the fault injection technology.

Key words: Web service; fault injection; testing model

0 引 言

面向服务的体系架构 SOA(Service Objected Architecture),由于它是松散耦合且基于已广泛接受的 Web 服务标准,从而提供了在每个不同的厂商解决方案间的交互性^[1]。

Web 服务将软件模块看成一种 Internet / Intranet 上的服务单元,借助 XML 和广泛应用的 Web 协议,如 SOAP、WSDL 和 UDDI 等^[2],实现分布式的计算和异构平台的信息集成。正因为 SOA 建立在异构平台的企业之间相互通信的能力基础之上,这使代码可以跨平台的 Java 成为开发 Web 服务及 SOA 集成的自然选择。

目前,对典型 SOA 结构的 Web 服务软件测试技术基本上还停留在对单个 Web 服务的测试上,少有对 SOA 体系结构软件的集成测试^[3]。而 SOA 强调的是 Web 服务之间的互操作,Web 服

务的调用过程可能引发消息和通讯等错误,因此如何测试 Web 服务间的交互关系,对 Web 服务软件进行总体测试十分重要。

1 SOA 的 Java 实现

1.1 SOA 的实现

SOA 的实现方法是一种自顶而下的设计方法,其体系结构思想中最重要的一条就是:“服务就是一切”。设计者在设计软件系统结构时,一切都是从服务的角度出发,首先考虑服务需求,进行系统服务装配,该软件过程划分如图 1 所示。当有大型服务库支持时,软件开发者不需发布自己的服务,可以跳过服务发布而直接查询服务。

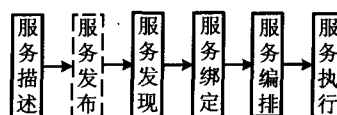


图 1 SOA 的实现过程

收稿日期:2005-12-02

基金项目:安徽省教育厅基金资助项目(2004kj136zd)

作者简介:吴 蕾(1978—),女,安徽合肥人,合肥工业大学硕士生。

1.2 Java 中的相关技术

SOA 中的服务依赖跨平台的消息交换技术作为其通信手段,XML(Extensible Markup Language,可扩展标记语言)在数据集成方面的能力已经使其成为 SOA 的基础。SOA 建立在使用不同计算平台的企业之间互相通信的能力基础之上,这使代码可以跨平台的 Java 成为实现 SOA 的自然选择。新的 XML Java Apis 的出现,使得在 Java 语言中使用 XML 越来越容易。

由于 XML 和 Java 平台之间相互配合得如此之好,两者已成为 SOA 实现的核心。开始者可以使用标准的 Java Apis 开发及部署 Web 服务,从而构建基于标准的 SOA。

这些 Java Apis 可划分为两大类:

(1) 面向文档的 Java Apis。处理 XML 的 Java API,简称 JAXP(Java API for XML Processing),使用几种不同的词法分析器如 SAX 和 DOM 直接处理 XML 文档。用于 XML 绑定的 Java 结构,简称 JAXB(Java Architecture for XML Binding),使用来自 Schema 的 JavaBeans 构建类自动处理 XML 文档。

(2) 面向编程过程的 Java Apis。基于 XML RPC 的 Java API,简称 JAX-RPC(Java API for XML-based RPC),经互联网向远程的伙伴发送 SOAP 方法调用并得到结果。处理 XML 消息的 Java API,简称 JAXM(Java API for XML Messaging),以标准的方式经互联网发送 SOAP 消息。用于 XML 格式注册中心的 Java API,简称 JAXR(Java API for XML Registries),提供标准的方式访问企业注册中心和共享信息。

XML 的 Java Apis 最重要的特性是支持工业标准,因此它保证了应用的互操作性。W3C Web 联盟等网络互操作性标准组织已经定义了这类工作的标准方法,因此遵循这些标准的企业数据和应用可以共同协同工作,这也正体现了 SOA 的核心思想。

1.3 基于 JWSDP 架构 SOA

Java 的 Web 服务开发包 JWSDP(Java Web Services Developer Pack)把所有 XML 的 Java Apis 包装在一个包中,JWSDP 支持行业标准,这就能够确保其与标准组织所发布的技术和规范具有互操作性。

JWSDP 还提供多种辅助工具,例如 WSDL stub 编译器,它能够生成一个 WSDL 文件,用于 Web 服务和独立的 Web 服务 UDDI 2.0 注册表。

JWSDP 还能和 Apache Tomcat 以及一些 Ant 的任务捆绑工作,这就让你能够在 Tomcat 内注册和管理 Web 服务。

SOA 提供了一种标准的编程模型,使得驻留在网络上的软件组件能够被发布、发现和调用,其主要包括 3 个主体:服务请求者、服务注册者和服务提供者。对应于 3 个行为主体,SOA 也包括 3 种主要的操作:查找服务、注册服务和绑定服务。作为 SOA 的一种实现方式,相应的 Web 服务体系结构也包括了同样的 3 个主体实现和 3 种操作接口。

SOA 作为一个概念上的模型,它将网络、传输协议以及安全等具体的细节都留给特定的实现。而 Web 服务体系结构实际上是 SOA 的一个特定的实现,充分体现了 SOA 其整体结构的分布性,松耦合的可集成性,标准的开放性以及应用过程的可管理性。

开发者用 UDDI 来注册 Web 服务,用 WSDL 来描述 Web 服务,用 SOAP 来处理 Web 服务的调用。

在 JWSDP 中,可以用 JAXR 在 UDDI 注册中心查找 Web 服务,用 JAX/RPC 来处理到外部 Web 服务的请求,用 JAXM 发送 SOAP 消息到外部 Web 服务,用 JAXP 和 JAXB 来把 Java 数据转换成适合于其他系统的 XML 格式,或把其他系统的数据转换成易于自己处理的 XML 格式,或者进行 XSLT 数据转换,如图 2 所示。

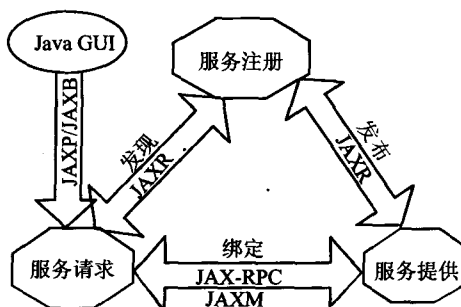


图 2 SOA 的 Java 实现

2 SOA 集成测试模型设计

由于 SOA 强调的是架构中各 Web 服务之间的互操作,所以对 SOA 进行集成测试关键是要测试 Web 服务之间交互时可能引发的消息及通信错误。这类通信错误主要包括:消息的删除、复制、延迟、重排序或讹传^[4]。

传统的分布式系统一般运行在 LAN 中并可以假定这一类型的错误可忽略,但是在 SOA 架

构中,如果在消息传递时出现这类错误就会被认为系统是不可靠的。而由于消息通信类错误的发生存在极大的偶然性,因此必须选择合适的测试技术才能很好地对通信类错误进行测试,从而维护系统的可靠性。研究表明,植错技术是解决以上问题的一个很好的方法^[5]。

(1) 植错技术。该技术作为一种测试技术是指按照特定的错误模型,用人为的、意识的方式产生故障,并施加于待测系统中,用以加速该系统的错误和失效的发生,同时观测并反馈系统对所注入故障的响应信息,通过分析,对系统进行验证和评价的过程^[6]。它给验证容错计算机和软件系统的可靠性提供了一种有效的解决方式^[7]。

传统的植错技术由于实时性限制等原因并不适用于分布式应用系统的测试;而目前利用植错技术对分布式系统的测试研究工作也仍然聚焦在基于紧密耦合的系统^[5],这对于测试 SOA 显然是不合适的。

此外现存的植错工具在测试网络消息丢失毁坏时很有用,但是在更详细地测试系统的其他性能时是无效的。所以要通过对 SOAP 消息中的 XML 文档进行解析,植入有意义的错误才能更好地、更有针对性的测试 SOA 集成时的消息通信错误。

(2) 扰动。是企图在不改变现存代码的前提下强行修改程序状态。这通常通过代码插入来实现。执行代码(称为扰动函数)以函数调用的形式被加入系统中来修改内部程序值,这些修改的值或者基于初始值随机产生,或者使用一个修改的常量值。

为了在 SOAP 消息中植入特定的消息通信错误,可以将扰动技术扩展为在网络消息中产生有意义的扰动,如可以在一个消息序列中关注某一个参数。这种参数扰动比标准的代码插入技术具有更小的破坏性,因为它对系统的改变更小更集中。

(3) 网络层植错。其关注的是网络接口中网络消息的毁坏、丢失或重排序。可以使用软件执行错误注入(SWIFI)工具,通过配置消息发送方操作系统的协议堆栈来注入错误^[4],但是这样就有可能被消息接收方系统协议堆栈检测到并拒绝接收。

因此人们使用一种标准网络层植错技术的变种,作为确定系统可靠性的手段,这项技术主要是通过修改 SOAP 消息中的参数来进行状态扰动。

虽然本文意图在 Web 消息中植入错误但无法直接这么做,因为消息在构造好之后会被加密/验证。加密就是确保消息在传输中不被读取和干预,消息一旦被加密就不能访问。因此,只能在消息中植入随机错误而不允许在特定的层中对消息进行所需要的操作。所以选择在应用层和协议栈顶之间的 API 边界植入错误,这是在任何加密和签名之前最低且最容易访问的错误植入点。这样,克服了消息被加密的问题并允许控制人们所需要的层。

(4) 基于 Java 的测试模型 JT4SOA。SOA 集成最重要的就是实现架构中 Web 服务互操作的平台无关性和语言无关性,利用 JAX-RPC^[8],非 Java 语言编写的客户端可以访问在 Java 平台上开发和部署的 Web 服务。

反过来说,利用 Java 语言编写的客户端也可以访问其他平台开发和部署的 Web 服务。而由于 JAX-RPC 遵守 SOAP 标准并支持 WSDL,所以能很好地实现跨平台的 Web 服务通信及互操作。

JAX-RPC 能在客户端将服务请求者的 RPC(远程方法调用)转化成 SOAP 消息,并利用 wscompile 工具使用 WSDL 文档建立 stub;而在服务器端将 SOAP 消息翻译为 RPC 并利用 wsdeploy 工具创建 tie。这一切都为在 SOA 架构中实现 Web 服务集成创造了先决条件。

利用植错技术对 SOA 集成进行测试,必须首先对通信中的 SOAP 消息进行解析,再对经过解析的 XML 文档中的某些参数进行有意义的状态扰动,从而模拟一定的通信错误,达到测试可靠性的目的。

在 JWSDP 中处理 XML 文档可由 JAXP 来完成,而 JAXP 本身并不提供词法分析功能,它要通过 SAX 或 DOM 来进行词法解析。由于 SAX 只能读出 XML 文档中的数据却不能修改数据,由于不仅需要得到 XML 中的数据,还需要对所得数据中的某些参数进行状态扰动,所以只能使用 DOM 文档对象模型来处理 SOAP 消息中的 XML 文档。

建立 DOM 后,就可以使用 DOM 中的方法 insert() 和 remove() 方法对 DOM 树进行灵活处理。

SOA 中的消息交互可以通过 JAXM API 来完成,而 JAXM 的连接有 2 种类型,一种是 javax.xml.soap.SOAPConnection 表示由消息发

送者到接收者的直接连接,由于它是点对点的,所以比较容易使用;另一种是 javax.xml.messaging.ProviderConnection 表示到消息提供者(Messaging Provider)的连接,这种连接方式需要消息发送者和消息接收者通过消息提供者来交互,消息提供者负责传递消息,它可以在消息到达最终接收者之前把消息转发到一个或多个中间目标,只需在消息的 SOAPHeader 对象中制定即可。由于采用植错的测试技术,所以必须使消息在传输的过程中经过植错模块,因此只能采用 JAXM 的第2种连接类型。

在测试模型 JT4SOA (Java Testing for SOA)中,是把植错模块作为一个单独的过程来处理的,这样构造的原因是:

(1) 简化植错模块的设计。

(2) 便于把这个模块安排在一个单独的机器上运行,再通过 JAXM 的消息提供者来处理消息传递和路由服务。在这个过程中时间主要花费在对 SOAP 消息的解析上,所以网络传输时间可以忽略不计。

(3) 这使多个节点使用相同的测试用例成为可能,具体的测试模型如图3所示。

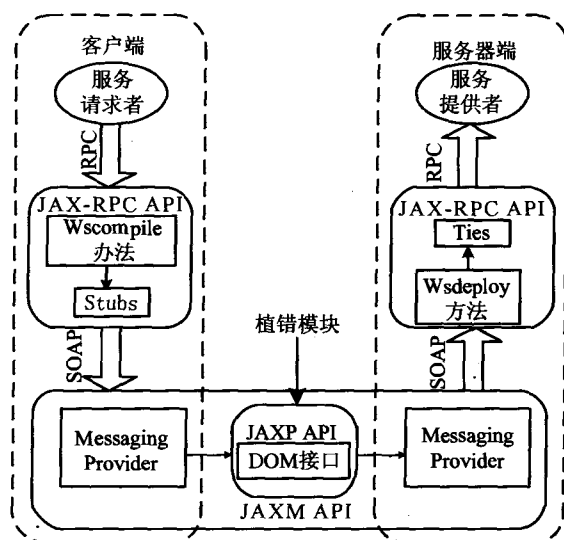


图3 JT4SOA 模型结构

年,SOA 将成为占有绝对优势的软件工程技术,因此进行 SOA 集成测试具有极其重要的理论和实践意义。

本文所提出的 JT4SOA 经过实验论证是一个切实可行的模型架构,下一步工作就是要将该模型转化为具体的测试工具,并通过对消息通信的错误模型进一步分析得出测试脚本,编写测试用例,从而最终实现自动测试。

参 考 文 献

- [1] 蔡月茹,柳西玲. Web Service 基础教程[M]. 北京:清华大学出版社,2005:8-120.
- [2] Curbera F, Duftler M, Khalaf R, et al. Unraveling the Web services Web: an introduction to SOAP, WSDL, and UDDI[J]. IEEE Internet Computing, 2002, (6): 86-93.
- [3] 黄 宁,余 莹,张大勇. Web 服务软件测试技术的研究与实现[J]. 计算机工程与应用, 2004, 40(35): 147-149.
- [4] Looker N, Xu J. Assessing the dependability of SOAP-RPC-Based Web services by fault injection[A]. 9th IEEE International Workshop on Object-oriented Real-time Dependable Systems[C]. Copri Island, Italy, 2003. 163-170.
- [5] Marsden E, Fabre J-C, Arlat J. Dependability of CORBA systems: service characterization by fault injection[A]. Proc 21st Int Symposium on Reliable Distributed Systems (SRDS02)[C]. Osaka, Japan; IEEE CS Press, 2002: 276-285.
- [6] 屠海澄,吴芳美. 面向软件黑箱测试的仿真环境嵌入故障研究[J]. 软件学报, 1999, (5): 516-520.
- [7] 王乐春,龚正虎,陈建荣. 基于错误注入技术的协议实现鲁棒性测试体系结构[J]. 计算机工程与应用, 2003, (22): 139-141.
- [8] Sun Microsystems Ltd. Java API for XML-Based RPC (JAX-RPC) [EB/OL]. <http://java.sun.com/xml/jaxrpc/index.jsp>, 2005-10-08.
- [9] 李心科,刘宗田,何允如. 一个面向对象软件测试工具实现及其实验研究[J]. 小型微型计算机系统, 2002, 26(5): 592-595.

(责任编辑 张秋娟)

3 结束语

SOA 的提出者 Gartner Group 预测到 2008