第35卷 第11期 计 算 机 工 程 2009年6月
Vol.35 No.11 Computer Engineering June 2009
·软件技术与数据库· 文章编号：1000—3428(2009)11—0059—03 文献标识码：A 中图分类号：TP393.03

基于TTCN-3的Web Service测试方法

王恩雷，赵会群，尚思超
(北方工业大学信息工程学院，北京 100041)

摘 要：介绍Web Service技术的新特点和TTCN-3的相关知识，针对传统测试方法在测试Web Service应用时出现的问题，提出一种基于TTCN-3的Web Service测试框架及测试方法，并进一步说明使用TTCN-3测试Web Service的测试流程，给出一个Web Service的测试实例，证明该测试方法的可行性。

关键词：Web Service测试；被测系统适配器；编码解码器；TTCN-3抽象测试套

Web Service Test Method Based on TTCN-3

WANG En-lei, ZHAO Hui-qun, SHANG Si-chao
(College of Information Engineering, North China University of Technology, Beijing 100041)

【Abstract】This paper introduces new characteristics of Web Service and the related knowledge of Testing and Test Control Notation version 3(TTCN-3). Aiming at the problems when using traditional test methods to test Web Service applications, it proposes a framework and a method of Web Service test based on TTCN-3, and presents the process of testing Web Service with TTCN-3. By giving a case of Web Service test, it illustrates the feasibility of this test method.

【Key words】Web Service test; system under test adapter; codec; Testing and Test Control Notation version 3(TTCN-3) abstract test suite

1 概述

Web Service是一种基于SOAP(Simple Object Access Protocol)和XML技术的互联网应用框架。Web Service具有语言无关性和平台独立性的特点。然而，Web Service这种新型的互联网集成技术却给传统的测试方法带来了挑战：

(1)从客户端的角度来看，测试人员不能够完全掌握服务器端Web Service的具体信息，只能得到Web Service的接口，无法设计期望结果的测试用例。

(2)从服务器端的角度来看，由于Web Service的语言无关性和平台独立性的特点，使得在服务器端的测试人员无法根据客户端的具体实现完成测试。

文献[1]说明了如何通过XML DTD(Document Type Description)和XML Schema自动生成Testing and Test Control Notation version 3(TTCN-3)脚本实现自动化的Web Service测试。文献[2]给出了一种基于XML的Web Service测试框架，但并没给出具体实现。文献[3]提出了一种突变分析的方法应用于Web Service测试，主要对WSDL文件进行操作，但只定义了9种转换操作，测试不够全面。

2 Web Service和TTCN-3简介

Web Service是一个应用程序，它向外界发布如何调用自身功能和服务的说明，包括服务地址、所提供的服务接口和各个服务所需接受的参数等说明信息。

Web Service的体系结构描述了3个角色(服务提供者、服务请求者和服务中介者)与3个操作(发布、查找和绑定)。其中，3个操作使用3种截然不同的技术：

(1)发布服务使用UDDI(Universal Description, Discovery and Integration)；

(2)查找服务使用UDDI和WSDL(Web Service Description Language)的组合；

(3)绑定服务使用WSDL和SOAP。

TTCN-3即测试及测试控制表示法第3版。2007年2月最新发布的TTCN-3标准规范包含TTCN-3核心语言、TTCN-3运行时接口、TTCN-3控制接口等10个部分。TTCN-3主要用于描述在多种通信端口上的各种响应系统的测试，其典型的应用领域是协议测试(包括移动应用协议和Internet协议)、服务测试、模块测试和基于CORBA(Common Object Request Broker Architecture)平台的测试[4]。

3 基于TTCN-3的Web Service测试方法

3.1 基于TTCN-3的Web Service测试框架

对于Web Service的测试主要是功能测试和性能测试，而负载测试又是性能测试的重中之重。功能测试由TTCN-3语言编写的测试套模拟客户端来对单个或多个Web服务进行测试；负载测试则可以模拟多个客户端的功能操作来对单个或多个Web服务测试。

图1表示TTCN-3的Web Service功能测试的GFT图，其中，Service1和Service2表示一个Web Service应用的2个服务；PtcType代表并行测试成分；Myport代表用TTCN-3语言描述的客户端。客户端可以对Service1发送Request1进行功能请求，也可以对Service2进行功能请求，请求发送后

基金项目：国家“863”计划基金资助项目“基于网格的海洋环境数据共享与信息服务技术研究”(2006AA09A139)；北京市自然科学基金资助项目“基于模型的测试方法及网络游戏软件测试关键技术研究”(4062012)

作者简介：王恩雷(1983－)，男，硕士研究生，主研方向：协议一致性测试；赵会群，教授、博士后；尚思超，硕士研究生

收稿日期：2008-12-04 **E-mail：**wangenlei@gmail.com

可返回功能请求是否成功的回应。

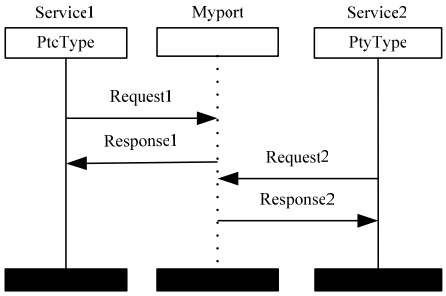

图 1　基于 TTCN-3 的 Web Service 功能测试框架

图 2 表示的是基于 TTCN-3 的 Web Service 负载测试 GFT 图，其中 *N* 表示并发用户数。客户端可以在本地对远端服务器的服务进行负载测试。

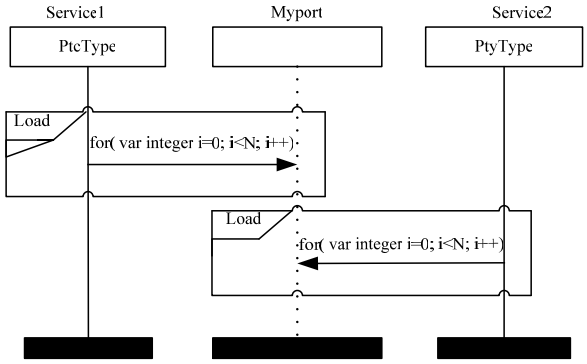

图 2　基于 TTCN-3 的 Web Service 负载测试框架图

以上的测试框架最重要的一方面在于其通用性，可以很大程度上独立于具体的 Web Service。除此之外，这一框架也适用于单独服务的性能测试、压力测试和整合服务测试。

3.2　基于TTCN-3 的 Web Service 测试方法

传统测试方法使得测试人员在客户端和服务器端都无法承担测试 Web Service 的任务。为了保证完成 Web Service 测试，测试用例的描述与设计应当在服务器端完成，然后将测试用例的描述发送到所有的客户端，最后在客户端完成测试实现并且执行测试用例。综上所述，测试 Web Service 的测试用例应以一种抽象的描述方式实现，TTCN-3 作为一种抽象测试描述语言，其语言无关性的特点正适用于解决传统测试方法在 Web Service 测试中的问题。

使用 TTCN-3 测试 Web Service 的具体测试流程如图 3 所示。

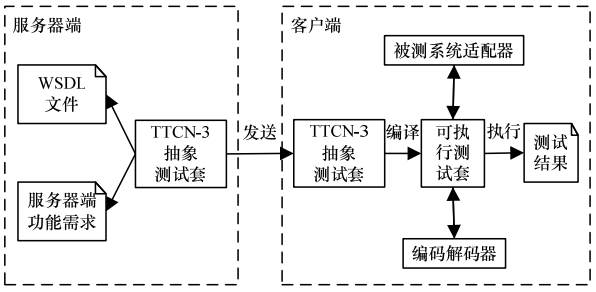

图 3　基于 TTCN-3 的 Web Service 测试流程

基于 TTCN-3 的 Web Service 测试流程如下：

(1)根据服务器端的功能需求和 WSDL 文件描述的 Web Service 信息完成 TTCN-3 抽象测试套的设计。

(2)将 TTCN-3 抽象测试套通过网络发送到所有的客户端，并在客户端通过 TTCN-3 编译工具编译生成可执行测试套。

(3)使用 Java 或 C++编写对应的被测系统适配器和编码解码器，配置完成后在客户端执行测试。

(4)在 TTCN-3 运行时环境中，执行测试用例得到测试结果。

该测试流程为 Web Service 的测试带来了 4 个好处：

(1)测试用例的设计在服务器端实现，保证了测试用例的质量。

(2)由于 TTCN-3 抽象测试套具有语言无关性和平台独立性的特点，使得测试人员只需要描述测试逻辑(例如，测试结果的验证)，而不用考虑测试的具体实现。

(3)在服务器端所开发的抽象测试套可由所有客户端共享。

(4)可开发通用的被测系统适配器实现测试部件的可重用，提高了测试效率。

4　一个Web Service的测试实例

本文使用 www.webservicex.net 网站提供的 TranslateService 作为 Web Service 测试实例。TranslateService 服务是一个翻译引擎，它可以为各种语言提供 Web Service 翻译服务，包括英译德、英译法、德译英、法译英等。该服务需要使用者提供 2 个参数，分别是翻译模式和翻译文本。TranslateService 的客户端是由 J2ME(Java 2 Micro Edition)和 kSOAP 实现的手机网络应用程序。

测试过程包括 4 步：

(1)测试用例的设计；

(2)抽象测试套的设计；

(3)测试实现；

(4)测试执行。

其中前 2 步在服务器端完成，后 2 步在客户端完成。

4.1　测试用例的设计

TranslateService 服务接收翻译模式的参数“英译德”和翻译文本的参数“Hello”。如果服务返回“Hallo”，则测试通过；如果返回其他字符串，则测试失败。为了测试提供服务的服务器负载大小，模拟 200 个客户并发调用 TranslateService 服务，得到负载测试结果。

4.2　功能测试的抽象测试套设计

由图 1 框架来设计测试套，按照 TTCN-3 的命名约定。

(1)定义服务请求的自定义数据类型，描述所提供服务的方法和参数：serviceName 指要请求的服务名称；languageMode 代表翻译模式；翻译文本用 text 来表示：

```
type record WsRequestType {
  charstring serviceName,
  charstring languageMode,
  charstring text
}
```

(2)根据自定义的服务请求数据类型，定义测试数据模板，用来描述测试输入和测试期望的返回结果：

```
template WsRequestType m_Request_001 := {
  serviceName: = "Translate",
  languageMode := "EnglishTOGerman",
  text: = "Hello"
```

```
}
template charstring m_Response_001 := "Hallo";
```

(3)定义测试配置用来描述构建测试结构的通信端口和通信组件，用基于消息的端口 WsPortType 来访问 Web Service，并通过该端口传输发送和接收的消息；不同的并行测试成分(PTCs)表示单独的服务测试或整合服务测试：

```
type component PtcType {
    timer tc_localTimer := 5.0;
    port WsPortType ptcPort;
}
```

(4)创建一个函数来描述并行测试组件的行为，在这一部分“wsRequest”请求将会被发送，如果接收到期望消息“wsResponse”，则测试被认为成功(pass)，其他消息或超时操作将会被挂起并提示失败(fail)：

```
function (in WsRequestType wsRequest, charstring wsResponse)
runs on PtcType {
    ptcPort.send (wsRequest);
    alt {
        [] ptcPort.receive(wsResponse) {
            tc_localTimer.start;
setverdict(pass);
        }
…
    }
}
```

(5)利用先前定义的函数创建测试用例并在控制描述部分执行测试例。v_ptc 是创建的并行测试组件(PtcType)：

```
testcase TC_001() runs on MtcType system SystemType {
    map(v_ptc:ptcPort, system:systemPort);
    v_ptc.start(f_ptcBehaviour(m_Request_001,
m_Response_001));
    v_ptc.done;
}
```

4.3 负载测试的抽象测试套设计

负载测试的测试套只需在测试例 TC_002()中修改通信组件的测试配置，创建一个并行测试组件数组。自定义的数据类型和测试数据模板的定义与功能测试抽象测试套中的设计一致。根据图 2 的框架可知，通过循环控制并发用户数可以对服务器进行负载测试，客户端执行相应的操作，从创建、连接、开始到完成，其中 NUMBER_OF_PTCS 是并发用户数，以下是开始操作的代码，创建、连接、完成与之类似：

```
for (i:=0; i<NUMBER_OF_PTCS; i:=i+1) {
v_ptcArray[i].start(f_ptcBehaviour(m_Request_001,
m_Response_001));}
```

4.4 测试实现

抽象测试套包含了服务器端的所有测试逻辑信息，在客户端需要将 TTCN-3 抽象测试套通过 TTCN-3 编译器生成可执行测试套。本文使用德国 Testing Tech 公司开发的 TTthree 作为 TTCN-3 编译器。TTthree 可以将 TTCN-3 抽象测试套编译为可执行测试套。除了可执行测试套，完整的测试实现还需要被测系统适配器和编码解码器。它们是 TTCN-3 测试系统中的标准实体。

被测系统适配器负责测试执行的控制、测试事件的记录、定时器的实现、与被测系统的通信以及测试系统用户接口的提供。被测系统适配器中最重要的接口方法是 triSend。triSend 创建一个 SOAP 消息，并向 TranslateService 发送请求，然后监听返回的 SOAP 消息并解析。编码解码器负责 TTCN-3 数据的编码和解码。

功能测试与负载测试通过使用相同的被测系统适配器和编码解码器完成了测试实现。这表明抽象测试套的设计体现了被测系统适配器和编码解码器的可重用性。

4.5 测试结果

在一个基于 Eclipse 的 TTCN-3 集成开发环境[5]中执行测试。测试结果如表 1 所示，测试用例 TC_001 的期望输出与实际输出一致，表示测试通过。测试用例 TC_002 的测试结果表示服务器端可以同时为 195 个客户提供 TranslateService 服务。

表 1 测试结果

测试用例	测试输入	期望输出	实际输出
TC _001	serviceName:="Translate", languageMode:="EnglishTOGerman", text:="Hello"	"Hallo"	"Hallo"
TC _002	serviceName:="Translate", languageMode:="EnglishTOGerman", text:="Hello" NUMBER_OF_PTCS:=100	190 个以上测试通过	195 个测试通过

5 结束语

本文用基于TTCN-3的Web Service测试方法来解决传统的测试方法在测试 Web Service 时的不足的问题。并且说明了使用 TTCN-3 测试 Web Service 的测试流程，通过测试部件的可重用性提高了测试效率。目前，TTCN-3 测试技术已经在除协议测试外的多种测试领域得到了成功应用，随着 TTCN-3 测试技术的日渐成熟，TTCN-3 将会在更广泛的测试领域中得到应用和发展。

参考文献

[1] Schieferdecker I, Stepien B. Automated Testing of XML/SOAP Based Web Services[C]//Proc. of the 13th Fachkonferenz der Gesellschaft f¨ur Informatik(GI) Fachgruppe KiVS. Leipzig, Germany: [s. n.], 2003: 43-54.

[2] Tsai W T, Ray P, Song Weiwei, et al. Coyote: An XML-based Framework for Web Services Testing[C]//Proc. of the 7th IEEE International Symposium on High Assurance Systems Engineering. Tokyo, Japan: [s. n.], 2002: 173-174

[3] Siblini R, Mansour N. Testing Web Services[C]//Proc. of the 3rd ACS/IEEE International Conference on Computer Systems and Applications. Los Alamitos, CA, USA: IEEE Computer Society Press, 2005: 135-142 .

[4] European Telecommunications Standards Institute. ETSI ES 201 873-1 V3.2.1-2007 Methods for Testing and Specification(MTS)[S]. 2007.

[5] Ma Yunfeng, Zhao Huiqun. Design and Implementation of TTCN-3 IDE Based on Eclipse[C]//Proc. of ISMATM’05. Beijing, China: [s. n.], 2005: 190-193.

编辑 任吉慧