

用分类树方法实现 Web 服务测试例的自动生成

李 乔, 秦 锋, 郑 啸

(安徽工业大学 计算机学院, 安徽 马鞍山 243002)

摘 要:随着 Web 服务技术的不断发展和广泛应用,需要运用新的测试技术来保障 Web 服务的正确有效运行,而测试例的生成是 Web 服务测试的一项重要内容,它将直接关系到 Web 服务测试的效率和成本。基于 Web 服务的 WSDL 文档,根据输入域中子集间相互关系,利用分类树的方法(Classification Tree Method),提出 Web 服务测试例的一种自动生成方法,据此达到最少的数量最大限度地覆盖的目的,并结合具体的实例,使用 CTE 测试工具增强测试的自动化程度,从而提高软件测试的效率和软件的质量。

关键词: Web 服务;黑盒测试;测试用例;分类树方法;WSDL

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2007)11-0197-04

Automated Test Case Generation for Web Service by CTM

LI Qiao, QIN Feng, ZHENG Xiao

(School of Computer Science, Anhui Univ. of Tech., Ma'anshan 243002, China)

Abstract: With the development of technology, Web services (WS) is applied extensively and new testing technology is needed to make WS highly reliable and run effectively. Test case generation is an important topic in Web services testing. The quality of test case will influence the efficiency and cost during Web services testing. Based on the WSDL document for Web services, according to relationship of input domain subset and using classification tree method, presents a method of automated test case generation for Web services. Therefore, it is able to cover all the combinations of subsets to the greatest degree with the smallest scale of test suite. Combining the example, applying test tool (CTE), automatization degree of software test was built up, and accordingly, the efficiency of software and the quality of software were largely enhanced.

Key words: Web services; black-box testing; test case; classification tree method; WSDL

0 引 言

随着网络技术和面向对象技术、分布式计算的进一步的融合发展,从 COM, COM+ /DCOM, CORBA 到 Web 服务,网络应用正朝着 SOA(Service Objected Architecture)^[1]软件体系结构方向转变。网络上 Web 服务的增长速度越来越快,新应用技术出现的同时,也需要新的测试技术^[2],而在测试 Web 服务中,测试用例的生成与选择技术是软件测试的一个重要研究领域,测试用例的选择将直接决定软件测试的科学性和有效性。统计表明,在所有的软件测试的开销中,约 40% 花费在设计测试用例上,约 50% 花费在编写和编译测试脚本上,另外约 10% 花费在测试脚本的执行和配置管理上^[3],由此可见测试用例的设计占有相当重要的

地位。

当前,Web 服务测试主要依靠测试人员手工设计测试数据,这种方法费时、费力,且带有一定的盲目性和倾向性,而提高测试用例生成的自动化程度能够有效地减轻测试人员的劳动强度,提高软件测试的质量。因此,如何为 Web 服务自动生成有效的测试用例成为急需解决的问题。测试用例选择是指从所有的可用测试用例中选出少量典型的测试用例,以达到对测试域的最大限度覆盖,它是对一项特定的软件产品进行测试任务的描述,体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等,并形成文档。

多年以来,很多研究人员对之进行了广泛而深入的研究,并取得了许多研究成果^[4-7]。常用的基于黑盒测试用例选择方法是对系统每个输入子集采用等价类划分法、边际值分析法、因果图法等选取一组典型的值,然后在这些取值组合中随机选取一组测试用例,或者使用一些启发式方法从中进行筛选^[8,9],然而这些

收稿日期: 2007-01-10

基金项目: 安徽省教育厅自然科学研究资助项目(2006KJ064b)

作者简介: 李 乔(1980-),男,安徽马鞍山人,硕士研究生,研究方向为 Web 服务的测试;秦 锋,教授,研究方向为计算机网络。

方法的缺点是带有主观倾向性,不具有普遍性。针对 Web 服务测试是一种黑盒测试的特点,首次把分类树的方法运用到 Web 服务的测试当中,解决以往测试用例只能手工生成的问题。一般情况下,用户无法获得 Web 服务的设计和实现细节,不能基于其结构生成测试用例。

因此,结合以往黑盒测试用例选择方法,基于 Web 服务的功能说明,即 WSDL^[10]文档中的方法描述,按照对 Web 服务系统各输入子集进行组合全面覆盖的原则来生成测试用例,再根据他们之间的关系选出符合要求的测试用例,然后调用 CTE 工具自动产生一组达到一定覆盖充分度的测试用例。通过具体实例,表明该方法能够自动、有效地产生测试用例,能实现最大限度组合覆盖的特点,从而可以在提高 Web 服务测试质量的同时,降低成本,提高效果。

1 黑盒测试用例选择方法

1.1 黑盒测试的意义、概念

随着软件规模的扩大,复杂程度的不断提高,以及面向对象程序设计方法和工具的使用,软件测试的难度也进一步提高,测试质量更加难以度量。不完全、不彻底是软件测试的致命缺陷,任何程序只能进行少量而有限的测试,而对于新出现的 Web 服务测试来说,更是如此。测试用例的设计选择在此情况下产生,同时它也是软件测试系统化、工程化的产物,因此,该领域的研究就显得非常重要和有意义。

黑盒测试是功能测试,它是已知产品的功能设计规格,可以进行测试证明每个实现了的功能是否符合要求,不需要了解被测对象的内部结构,从这个方面来说,非常适用于 Web 服务的测试,Web 服务在被测试时被看成一个不能打开的黑盒子,在完全不考虑程序内部结构和内部特性的情况下,测试者在程序接口进行测试,它只检查程序功能是否按照需求规格说明书的规定正常使用,程序是否能适当地接收输入数据而产生正确的输出信息,并且保持外部信息的完整性。前人根据等价类划分、边值分析、因果图、正交实验设计法、判定表驱动等黑盒测试方法对测试用例进行选取,这种手动的划分效率非常低下,而且随着组合数的急剧增加,往往不能考虑和保持正确性,而用分类树的方法可以很好地解决这个问题。

1.2 黑盒测试的实现方法

1.2.1 等价类划分

等价类划分是测试用例设计的形式化方法,它将被测软件的输入输出划分成一些区间,被测软件对一个特定区间的任何值都是等价的。等价类可分两类:

有效等价类和无效等价类。有效等价类是指对被测对象的规格说明是有意义的、合理的输入数据所构成的集合;无效等价类是被测对象的规格说明是无意义的或不合理的输入数据所构成的集合。虽然等价类划分法能够比较全面、系统地设计黑盒测试的测试用例,但是没有注意选用一些“高效的”、“有针对性的”测试用例。

1.2.2 边界值分析

边界值分析的思想是假定错误最有可能出现在区间之间的边界,它是对等价类划分方法的补充。边界值分析使用在最小值、略高于最小值、正常值、略低于最大值和最大值处取输入变量值。它不是选择等价类的任意元素,而是选择等价类的边界。在设计测试用例时,边界值是最容易出错的地方。

1.2.3 因果图方法

等价类划分方法和边界值分析方法,虽然都注重输入子集的分类和划分,但未考虑输入子集之间的联系、相互组合等。若要检查输入子集间的联系及组合,则要把所有输入子集划分成等价类,即使这样,他们之间的组合情况也相当多,因此需利用因果图来设计测试用例。因果图方法最终可以生成判定表。

1.2.4 正交表方法

如果在此基础上再考虑保证任意两个子集之间各种组合的等概率覆盖,这样也能够精简测试用例,这种方法实际上就是正交实验设计。它是一种比较有效的测试用例选择方法,然而这种方法依赖于正交表,因为正交表的构造还存在很多未解决的难题,特别是对于混合型的正交表,目前还没有比较好的构造方法,所以这给正交试验设计^[11]在黑箱测试用例选择中的应用带来较大的局限。

2 WSDL 的解析

WSDL 是描述 Web 服务的标准 XML 格式,它用一种和具体语言无关的抽象方式定义了给定 Web 服务收发的有关操作和消息。WSDL 信息模型分离了服务接口定义(抽象接口)与服务实现定义(具体端点)。抽象接口规范描述了终端的处理能力,它在 WSDL 中表示为 portType。绑定机制用特定的通信协议、数据编码模型和底层通信协议,将 Web 服务的抽象定义映射至特定实现。如果绑定结合了实现的访问地址,抽象端点也就成为可供服务请求者调用的具体端点, WSDL 的 port 元素表示了这一结合。

因此, portType 描述了 Web 服务是什么, binding 元素描述了如何使用 Web 服务, port 及 service 元素描述了 Web 服务的位置。WSDL 文档的 URL 就是 Web

服务本身的 URL 在末尾附加“? wsdl”。在客户端代码中请求 Apache Axis^[12]为我们生成 WSDL 文档,制定想要的具体服务、端口以及操作,而 Apache Axis 将能够从 WSDL 文档中抽取其他附属信息,不用自己定义。

通过上述方法,可以得到被调用的 Web 服务需要输入的各子域信息,这就为下一步用分类树的方法设计和选择测试用例提供了有力支持。

3 分类树的方法及分类树工具

在软件测试中,测试例的选择设计是非常关键的。基于规约的测试例确定,是一种功能测试。在工业领域功能性测试被广泛运用,并且对于验证系统非常重要,然而可用于产生相应测试例的方法和工具非常少。通常情况下,仅有的分割应用工具只有决策表或是 MS Excel,他们用于测试例的确定。分类树的方法对于功能性测试是一种非常有效的方法,它能够通过分类划分一个测试对象的整个输入域为各自独立的子集类,并且分类树编辑器(CTE)^[13]能够更高效地运用分类树方法。它是一种语法控制的、图形编辑器,通过使用分类树方法,能够对测试例的选择确定提供有力的支持。

分类树的主要思想是首先从被测对象中抽取具体测试数据,并且根据测试者考虑相关测试的方面,划分每个测试单元的数据输入域为分割的子集。通过组合不同的分类子集,产生测试例。分类树方法首先是划分测试对象的各个输入域的子集,这对于测试对象的功能性规范是相当重要的。对于每个单元对象,将他们划分成不相交的各个子集,这种分类划分应当是精确的、明晰的不同可能的输入。第二步就是在上述分类树划分的基础上,通过每个划分的单元子集的相互组合,来定义测试例。由于每个单元子集的组合形成的测试例将是非常巨大的一个数字,因此,第三步就是分析各个输入域集合的关系,借助分类树方法,用 CTE 图形化工具,进行等价类划分、因果关系定义、边界分析等,最终精简测试例数量。

4 测试用例生成实例

下面列举了一组测试用例生成的过程,用 CTE 生成工具产生了一组相应的测试用例表,测试用例的数量即测试用例表的行数。首先,解析 Web 服务中 WSDL 的描述,得到需要需要输入的参数,也就是输入域。本例的 Web 服务是输入产品零件的各个型号参数后得到该零件单价的一种服务。产品的型号规格有颜色、尺寸、形状、材料这四个输入参数,颜色可以输入

红、绿、蓝、黑任意一种,尺寸可选大、中、小当中的一种,形状则可能是三角形、长方形、圆形中的一种,材料可能是金、银、铜、铁、铝五种的一种。在 CTE 中把该类命名为零件,那么根据上述的情况,它有四个可以用于输入的子域,在 CTE 中编辑的树如图 1 所示,此时也就完成划分测试对象的各个输入域的子集。根据前面的定义,如果要遍历所有的组合,测试例的数目应该是 $4 \times 3 \times 3 \times 5$,即为 180 个。图 1 中左下方的测试例(Test Case)和右下方的一条横线相对应,而横线上的各个黑点则代表在每个输入域中所取的值。该 Web 服务中,存在下列多个相互关系。

关系 1:如果颜色是蓝和红时,零件的价格是一样的。在 CTE 中定义这种等价关系为:Red Or Green Or Black。

关系 2:只要有红颜色的零件,那么形状必然是大的。在 CTE 中可以定义这种因果关系为:Red = > Large。

关系 3:对于零件的形状是三角形的,那么它的材料就不可能是金、银两种。同样在 CTE 中定义因果关系为:

Triangle = > (Not Gold) And (Not Silver)。

关系 4:当零件的尺寸不是大的时候,它的材料必然是金的。在 CTE 中因果定义关系为:Small Or Middle = > Gold。

有了上述几种关系后,在 CTE 中进行关系的约束申明和定义,如图 2 所示,左下方测试例中将自动显示不满足要求的出黑点的测试例,可以看到,拿编号 Testcase1 测试例来说,它违背了上面定义的 1、4 关系,所以不能作为有效测试例,按照上面的过程操作,此时的满足上述要求的测试用例只有 47 个,精简了 73.8%,这将会省去人工手动运用判定表或因果状态迁移图划分测试例的时间,并且大大提高操作的准确性和效率。

对于本例遍历所有测试例是可以做到的,但是随着输入域的递增及输入子集的增加,测试例的数量将是非常巨大的,此时要想遍历完全是不可能的,而且也是不必要的,本例只是一个举例,如果推广到一般情况,当输入子集的输入是非离散值的时候,可以按照以下方法来处理:

设一个系统对外输入域由 n 个参数 N_1, N_2, \dots, N_n 组成。每个参数都有其指定的取值范围(约束): $N_1 \in D_1, N_2 \in D_2, \dots, N_n \in D_n$,其中 D_i 为参数 N_i 的合理取值范围。现在从这组输入域出发,决定选取测试用例来进行测试。先根据边界分析法选择 N_i 可以取到的边际值,因为边际值往往具有较强的暴露错误的能

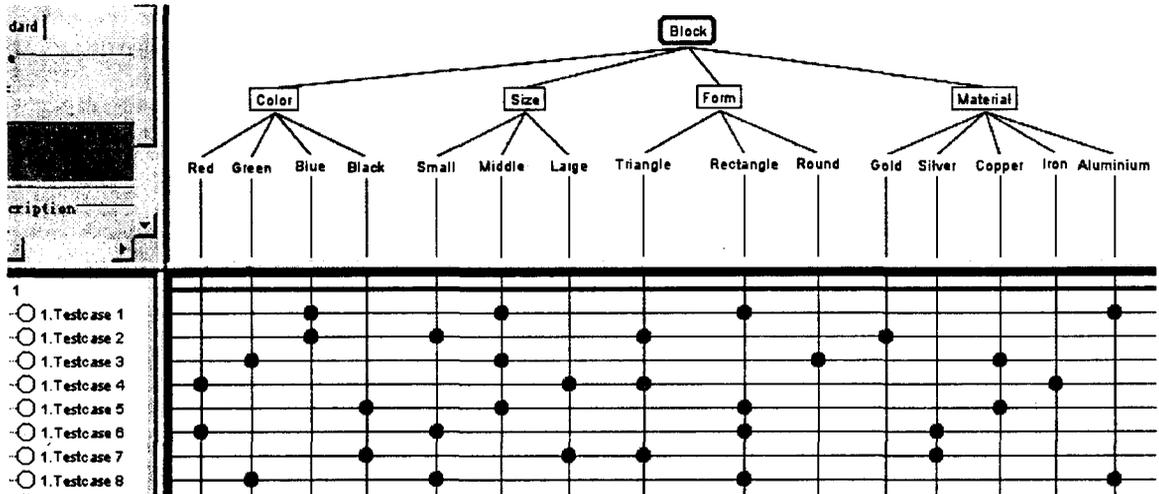


图1 初始测试用例的生成

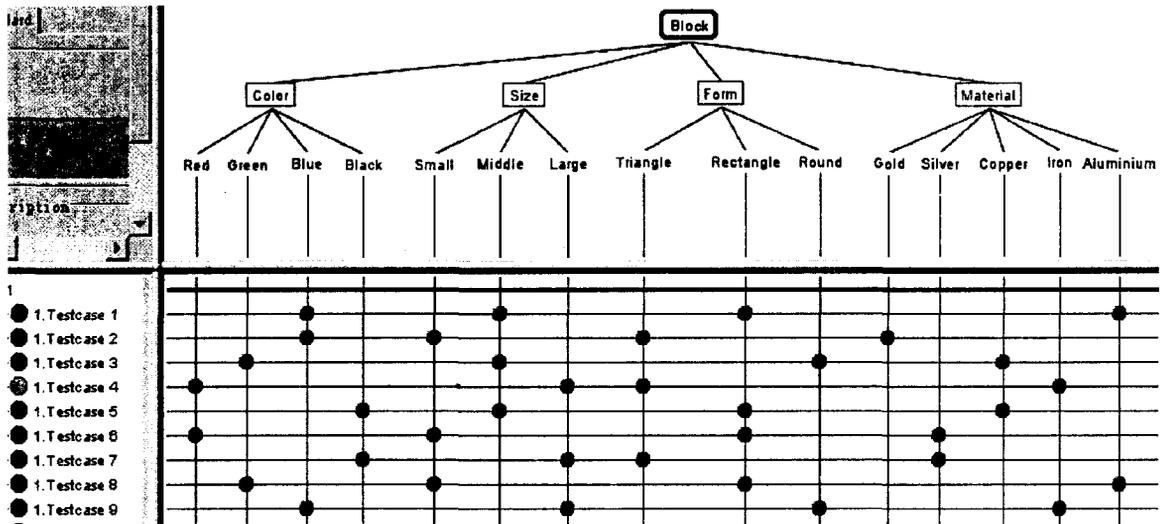


图2 精简以后的测试用例

力,再根据等价划分法补充一些取值点,使得 N_i 的取值更具有代表性。经过这样处理后,参数 N_1, N_2, \dots, N_n 都取一组离散值 L_1, L_2, \dots, L_n 。其中 L_i 表示参数 N_i 可取的有限离散点集,并假设符号 l_i 表示第 i 个参数 N_i 可取值的个数,即 L_i 中元素的个数: $l_i = |L_i|$ 。经过以上处理,需要 $S = l_1 \times l_2 \times \dots \times l_n$ 个测试用例,数量还是相当大,需要进一步精简,此时就可以用分类树的方法在 CTE 中进一步精简了,我们就能够用挑选的测试用例最少的数量最大限度地对上面各种可能性因素进行覆盖。综合考虑测试用例的规模、覆盖能力以及测试费用,如果采用了分类树的方法,实际上在多数情况下,生成工具产生的测试用例数量远远小于全面覆盖测试要求的测试用例数,因此会大大提高效率和测试例的质量。

5 结束语

提出了一种基于 CTM 的 Web 服务测试用例自动

生成方法,用实例验证了该方法的可行性,列举了该法实现的实际效果,说明了该法产生的测试用例具有数量少、高度覆盖等优点。以后还将就一般情况下如何生成最优的测试用例做进一步的研究。Web 服务测试是一项新的研究领域,虽然以往有很多的测试技术和测试方法,但针对 Web 服务测试,目前还没有完全形成一个统一的行业标准,有很多的测试方法很难在实践工程中进行应用,灵活的运用分类树测试方法和 CTE 工具,有效地组织测试人员和安排测试任务,并且尽量使用软件测试工具增强软件测试的自动化程度,可以帮助软件开发和测试人员提高测试效率和软件的质量。

参考文献:

[1] Sprott D, Wilkes L. Understanding Service-Oriented Architecture[EB/OL]. 2004-02. <http://msdn.microsoft.com/architecture/soa/default.aspx>.

(8) 剩余电流动作保护器远程自动监控。远程设置剩余电流动作保护器的整定值, 显示低压出线零序电流值; 远程判断故障类型, 进行合闸或闭锁操作。

(9) 数据自动管理和智能评估。对低压配网系统内配变和低压出线的运行数据进行自动分类管理; 对系统的供电可靠性、电压合格率、低压线损率等进行智能分析; 对运行状态、供电质量、供电能力等作出科学评估。

(10) 与 GSM 通信网络手机用户进行交互。手机用户既可与监控中心进行信息互传, 也可充当移动式监控中心对指定的一台或多台下位管理机进行移动式管理。

5 结论

讨论了低压配电网多功能无线远程监控的实现和基于 GSM 通信网络、计算机网络、客户/服务器结构与浏览/服务器结构结合的体系结构的管理信息系统的技术构成, 将管理信息系统应用于低压配电网管理自

动化中, 设计开发了低压配电网多功能无线远程监控管理信息系统。并结合实际, 介绍该系统的设计方案、实现技术和功能特点。现场运行表明, 该系统具有监控装置接线简单、配置灵活、抗干扰能力强、可靠性高, 管理信息系统技术先进成熟、运行维护方便等显著优点, 有力地促进了低压配电网自动化的发展^[5], 值得在当前电力企业低压配电网管理中大力推广运用。

参考文献:

- [1] 唐涛. 国外变电站无人值班与综合自动化技术发展综述[J]. 电力系统自动化, 1995, 19(10): 10-17.
- [2] 黄晓晨, 阙建飞, 周志祥, 等. 电力调度管理信息系统的研究[J]. 电力系统通信, 2001, 12(8): 21-23.
- [3] 徐玮, 尹宝林, 李昭原. 企业信息系统业务构件设计研究[J]. 软件学报, 2003, 14(7): 1213-1225.
- [4] 王敏, 李静, 范中磊, 等. 一种虚拟化资源管理服务模型及其实现[J]. 计算机学报, 2005, 28(5): 856-863.
- [5] 唐涛. 电力系统厂站自动化技术的发展与展望[J]. 电力系统自动化, 2004, 28(4): 92-97.

(上接第 196 页)

参考文献:

- [1] 胡广书. 数字信号处理——理论、算法与实现[M]. 北京: 清华大学出版社, 1997.
- [2] Texas Instruments. TMS320VC33 Digital Signal Processor [M]. Texas: Texas Instruments Incorporated, 2000.
- [3] 姚文刚, 余国强, 孟小锁. 基于多 DSP 架构的电机控制系统

[J]. 计算机技术与发展, 2006, 16(6): 48-53.

- [4] Texas Instruments. TMS320C3x User's Guide[M]. Texas: Texas Instruments Incorporated, 2001.
- [5] Texas Instruments. TMS320C3x general-purpose applications user's guide[M]. Texas: Texas Instruments Incorporated, 2001.
- [6] 王金友. 用 CPLD 实现 DSP 与外设芯片的速度匹配[J]. 电子测量技术, 2006(4): 73-75.

(上接第 200 页)

- [2] Schieferdecker I, Stepien B. Automated Testing of XML/SOAP based Web Services[C]//Proceedings of the GIFachtagung "Kommunikation in Verteilten Systemen", KIVS. Leipzig, Germany: [s. n.], 2003: 43-54.
- [3] Patton R, 周予滨, 姚静, 等. 软件测试[M]. 北京: 机械工业出版社, 2003.
- [4] DeMillo R A. Constraint-based automatic test data generation[J]. IEEE Transactions on Software Engineering, 1991, 17(9): 900-910.
- [5] Grabowski J, Wiles A, Willcock C, et al. On the design of the new testing language TTCN-3[C]// In: Ural H eds. Testing of Communicating Systems. Kluwer: Academic Publishers, 2000: 161-176.
- [6] Schieferdecker I, Grabowski J. Conformance testing with TTCN[J]. Elektronik, 2000, 96(4): 85-95.
- [7] Offutt J. An integrated automatic test data generation system [J]. Journal of Systems Integration, 1991, 1(3): 391-409.

- [8] Chen T Y, Lau M F. A new heuristic for test suite reduction [J]. Information and Software Technology, 1998, 40(5-6): 347-354.
- [9] Walter T, Grabowski J. A framework for the specification of test cases for realtime distributed systems[J]. Information and Software Technology, 1999, 41(11-12): 781-798.
- [10] W3C note, WSDL (Web Service Description language)[EB/OL]. 2001-03. <http://www.w3.org/TR/wsdl.html>.
- [11] Hedayat A S, Sloane N J A, Stufken J. Orthogonal Arrays: Theory and Applications[M]. New York: Springer Verlag, 1999.
- [12] Apache Software Foundation. Axis User's Guide[EB/OL]. 2004-02-08. <http://ws.apache.org/axis/java/user2guide.html/>.
- [13] Classification Tree Editor[EB/OL]. 2005-08-09. <http://www.systematic-testing.com/>.