一种基于组合测试的 Web 应用兼容性故障诊断方法*)

许 蕾[1,2] 聂长海[1,2] 徐宝文[1,2,3]

(东南大学计算机科学与工程系 南京 210096)[1] (江苏省软件质量研究所 南京 210096)[2]

(武汉大学软件工程国家重点实验室 武汉 430072)[3]

摘 要 Web 应用兼容性故障诊断涉及到种类繁多的软硬件设备、数目庞大的设备品牌和型号以及各种情况的组合,如何有效而又快速地对故障定位是一项很重要的工作。针对 Web 应用兼容性测试的特性以及组合测试的基本模型和特征,我们对组合测试的结果进行分析,根据初步分析结果补充一些附加测试用例进行重新测试,并对其结果作进一步分析和验证,从而迅速将故障原因锁定在很小的范围内,为 Web 应用兼容性的调试和测试工作提供方便、有价值的线索和参考。

关键词 Web 应用,兼容性测试,组合测试,故障诊断

A Fault Diagnosis Method for Web Application's Compatibility Based on Combinatorial Testing

XU Lei[1,2] NIE Chang-Hai[1,2] XU Bao-Wen[1,2,3]

(Department of Computer Science & Engineering, Southeast University, Nanjing 210096)[1]

(Jiangsu Institute of Software Quality, Nanjing 210096)[2]

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)[3]

Abstract The fault diagnosis for Web Application's compatibility is concerned with multiple kinds of software and hardware facilities, enormous numbers of equipments, and the combination of all the situations. How to orient the location of the fault effeciently is an important task. Combined with the characters of Web applications' compatibility testing and the model and characters of combinatorial testing, we analyze the combinatorial testing results based on our previous work. Then we retest with some complementary test cases based on the elementary analytic results, and make further analysis and validation with the retesting results. Thus we obtain the factors that cause the errors in a very small range, and we also could provide efficient and valuable guidance for the debugging and testing work.

Keywords Web appplications, Compatibility testing, Combinatorial testing, Fault diagnosis

1 引言

Web 应用以其广泛性、交互性和易用性等特点得到快速发展。但由于 Web 应用的组成成分、网络的拓扑结构、接入方式、物理配置等存在差别,带来异构、自治等问题。为了保证 Web 应用在各种不同的浏览器及用户端配置下都能够正常运行,需要进行兼容性测试[3],主要是在不同的客户端环境下对 Web 应用的显示作测试。在测试完成以后,还需要对测试结果进行分析,以诊断、定位并排除故障[4]。这些工作非常繁琐,需要数量适宜的测试用例和有效快速的故障诊断策略。

目前,国内外对 Web 测试进行研究并取得了一些研究成果[3,5],我们在 Web 测试的建模、测试方法和技术等相关方面进行了一定的尝试[7~9]。文[8]中提出了一种基于组合测试的 Web 应用兼容性测试用例生成方法,使用多个参数表示客户端的软硬件种类,根据需要选择单因素覆盖法或两两组合覆盖法,能够最大限度地以最少数量的测试用例实现对各种可能配置情况的覆盖。根据取得的测试初步结果,本文将着重研究如何利用组合测试方法实现快速有效的故障诊断。

为此,本文首先分析了 Web 应用兼容性测试的特性以及组合测试的基本模型和特征;并结合我们先前工作取得的成果对 Web 应用兼容性测试结果进行分析,然后补充一些附加测试用例进行重新测试,对其结果作进一步分析和验证,从而将故障原因锁定在很小的范围内。

2 相关工作

2.1 Web 应用兼容性测试的特性

Web 应用页面文件的解析和显示离不开浏览器的支持。浏览器有各种版本的 Netscape、Internet Explorer、AOL、Opera 等,可应用于 Windows、Macintosh、Unix、Linux 等平台,可能会出现:不同的浏览器对同一内容的解析不同;不同类型的计算机在提供的字体类型和字体大小上会有差别;不同的屏幕尺寸对页面的显示也有影响;不同浏览器具有不同的容错性。因而要在不同的客户硬件配置、客户操作系统、浏览器类型和版本以及浏览器插件的组合使用情况下进行 Web 应用兼容性测试,规模之大难以想象。为此,在进行 Web 应用兼容性测试前,需要很好地设计测试用例,既要使其数目在一

*)本研究得到国家自然科学基金(60425206,90412003,60503033)、国家重点基础研究发展规划 973 资助项目(2002CB312000)、江苏省自然科学基金(BK2005060)、江苏省高技术研究项目(BG2005032)、武汉大学软件工程国家重点实验室开放基金资助。许 蕾 博士,讲师,研究方向:Web 应用分析测试。聂长海 博士,副教授,研究方向:软件工程和软件测试技术、模糊信息处理等。徐宝文 博士,教授,博导,研究方向:程序设计语言、软件工程、并行与网络软件等。

个可以接受的范围，又要能覆盖尽可能多的客户端软硬件配置情况。

组合测试技术[1,2,4,6]能够以较少的测试用例实现对被测系统的科学有效的测试，特别适用于由多个参数以及参数间相互作用而影响系统状态的系统。相应地，Web 应用的客户端通常由浏览器、操作系统、显卡、声卡、打印机等软硬件设备组成，一种设备可以用一个对应的参数来表示，各参数可以有多种取值，从而可以应用组合测试方法来生成高质量的测试用例。

执行测试用例后，如果有故障产生，需要进一步分析故障原因以定位并排除故障，即查找引起故障的参数或参数组合，具体、有效的故障诊断方法可通过分析组合测试的模型和特征而得到。

2.2 组合测试的基本模型和特征

应用组合测试方法进行测试时，要求待测系统的测试场景能够由一组参数决定。假设某待测软件 SUT(Software under Test)中有 n 个参数 $c_1,c_2,\cdots,c_n$，每个参数分别有 T_1，$T_2,\cdots,T_n$ 种取值。

定义 1 称 n 元组 $(v_1,v_2,\cdots,v_n)(v_1\in T_1,v_2\in T_2,\cdots,v_n\in T_n)$ 为 SUT 的一个测试用例。

定义 2 对 SUT 只取定某 $k(1\leqslant k\leqslant n)$ 个位置上的参数值后形成的 n 元组 mod=(-,…,v_{i1},-…,v_{i2},-…,v_{im},-…,-)，称 mod 为 SUT 的一个 k 值模式，其中"-"表示相应位置上参数的取值待定。

在应用测试用例集 $T(|T|=m)$ 对待测软件系统进行测试时，如果其中有 l 个测试用例在执行中发现某个故障(记为集合 A)，而其它 $m-l$ 个测试用例都正常运行(记为集合 B)，那么需要进一步分析这个故障可能是由哪些取值模式引起的。我们首先给出定理 1 及其证明。

定理 1 若待测试系统的某个故障只由某个 k $(1\leqslant k\leqslant n)$ 值模式 mod=(-,…, v_{i1},-…, v_{i2},-…,v_{im},-…,-)引起，则对任意 $x\in A$，x 中一定含有模式 mod，且对任意 $y\in B$，y 中一定不会出现模式 mod。

证明：若某 k 个参数的一个取值组合是导致系统某个故障的原因，而这个组合在另外某个没有发生该故障的测试用例 NF 中出现，则在测试时，NF 一定会引发故障，这与条件矛盾，得证。

由定理 1 可以推得：①如果待测试系统的某个故障由某个 $k(1\leqslant k\leqslant n)$ 值模式引发，那么包含该 k 值模式的所有测试用例将导致系统出现故障；②如果待测试系统的某个故障不只是由某 $k(1\leqslant k\leqslant n)$ 个参数的某个取值组合引起，那么引起该故障的这些取值组合一定只会出现在集合 A 中，而不会出现在集合 B 中；③如果我们能从集合 A 中找到公共的模式，即某个参数(某组参数)的某个取值(取值组合)在集合 A 中都出现，而在集合 B 中不出现，那么这些公共模式是最有可能引起故障的原因；④凡是在集合 A 中出现过而在集合 B 中没有出现的模式，都可能是引起这个故障的原因，在集合 B 中出现的任何模式都不可能是引起这个故障的原因。

基于以上定理和推论，要确定故障原因，需要从集合 A 中找到公共模式，并且这些模式在集合 B 中不出现，将这组公共模式记为集合 M。该集合包含了最有可能引发这个故障的原因。

为了进一步确定出是哪些参数的取值组合导致了软件故障，还需要为每个发生故障的测试用例设计 n 个测试用例：$T_{at}=\{(*,v_{2i_2},\cdots,v_{m_n}),(v_{1i_1},\cdots,v_{m_n}),\cdots,(v_{1i_1},v_{2i_2},\cdots,v_{(n-1)i_{n-1}}*)\}$，其中 * 号表示我们可以用任意不同于原来测试用例 $(v_{1i_1},v_{2i_2},\cdots,v_{m_n})$ 相应位置参数的取值来代入。这样需要产生 nl 个附加测试用例。运行这组测试用例，根据运行结果，可以将导致故障的因素锁定到很小的范围。如果发现多个故障，可以分别对各个故障使用以上原理及推论。

3 Web 应用兼容性故障诊断的过程和方法

为了便于说明问题，我们在 Web 应用兼容性测试[8]的基础上说明如何进行故障诊断。为简单起见，只考虑五种软硬件设备，并假设每种设备有 2 个主流品牌的软硬件供选择，如表 1 所示。

表 1 兼容性测试所需的软硬件类型表

显卡	声卡	打印机	浏览器	操作系统
A1	B1	C1	D1	E1
A2	B2	C2	D2	E2

对系统进行测试后，如果没有发现故障，则无需进行故障诊断。若有 l 个测试用例运行时发生某个故障，故障诊断通常分三步进行：1)对集合 A 中的测试用例本身的分析；2)生成附加测试用例以辅助故障原因定位；3)根据附加测试用例的运行情况重新进行分析验证。

3.1 Web 应用兼容性测试结果分析

如文[8]所示，只需 6 个测试用例就可实现对表 1 中各因素的两两组合覆盖，其测试用例集 T={(A1,B1,C1,D2,E1),(A1,B1,C2,D1,E2),(A1,B2,C1,D1,E1),(A2,B1,C2,D1,E2),(A2,B2,C1,D2,E2),(A2,B2,C2,D2,E1)}。当我们用这组测试用例对 Web 应用进行测试后，如果没有发现问题，那么就说明 Web 应用功能在这个水平上的测试是合格的，不需要进一步分析。如果这组测试用例中有一个或几个运行时发生了故障，则需要分析故障原因并进行故障的定位和排除工作。

如前所述，要确定故障原因，需要寻找公共模式 M。不失一般性，当用 T 中这组测试用例对 Web 应用进行测试时，我们以只有一个测试用例在系统运行时发生故障的情况为例作讨论。不妨设这个测试用例为：(A2,B2,C2,D2,E1)，即 T 中第六个测试用例。在这种情况下，单个参数不是导致系统出错的原因，因为单个参数的某个取值多次出现在其它测试用例中；2-值模式(A2,-,-,D2,-)，(A2,-,C2,-,-)，(A2,B2,-,-,-)，(-,B2,-,D2,-)，(-,B2,-,-,E1)，(-,-,-,D2,E1)也不是导致系统出错的原因，因为这些模式分别出现在其它不引起故障的测试用例中，即用例 5、用例 4、用例 5、用例 5、用例 3 和用例 1；其它情况下都有可能导致故障产生。

因此，M={(A2,-,-,-,E1),(-,B2,C2,-,-),(-,-,C2,D2,-),(-,-,C2,-,E1),(A2,B2,C2,-,-),(A2,B2,-,D2,-),(A2,B2,-,-,E1),(A2,-,C2,-,E1),(A2,-,C2,D2,-),(A2,-,-,D2,E1)(-,B2,C2,D2,-),(-,B2,C2,-,E1),(-,B2,-,D2,E1),(-,-,C2,D2,E1),(A2,B2,C2,D2,-),(A2,B2,C2,-,E1),(A2,B2,-,D2,E1),(A2,-,C2,D2,E1),(-,B2,C2,D2,E1),(A2,B2,C2,D2,E1)}。集合 M 中的元素都是可能导致系统出现故障的因素，而这些可能因素的数目较大，需要进一步缩小范围，以迅速锁定故障位置。

(下转封四)

(上接第 269 页)

3.2 补充附加测试用例进行重新测试

为每个发现故障的测试用例生成相应的 n 个测试用例，即前文所述的附加测试用例 T_{at}。然后用这些测试用例重新对系统进行测试，如表 2 所示，其中 * 号表示相应位置的参数值取任意一个不同于原来出错测试用例在相应位置的值(在本例中取括号内的值)。

表 2 附加测试用例表

显卡	声卡	打印机	浏览器	操作系统
*(A1)	B2	C2	D2	E1
A2	*(B1)	C2	D2	E1
A2	B2	*(C1)	D2	E1
A2	B2	C2	*(D1)	E1
A2	B2	C2	D2	*(E2)

用这组测试用例再对 Web 应用进行测试，根据测试结果我们就可以确定出导致故障的原因。即进一步缩小故障可能原因范围，在原先 M 集合的基础上，去掉在附加测试用例中未发生故障的测试用例内出现过的所有模式，则剩下的模式将是有可能导致故障的原因。如果所有的附加测试用例运行时都发现故障，那么一定是附加测试用例引入了新的导致故障的模式。

3.3 对结果作进一步分析和验证

执行附加测试用例后，根据故障的具体产生情况，举例说明如何确定故障原因。

情形 1 运行这五个附加测试用例都没有发生故障。根据前述方法，M 集合将缩减为{(A2,B2,C2,D2,E1)}，此时说明取值组合(A2,B2,C2,D2,E1)是导致故障的原因。

情形 2 只有一个测试用例运行时发生故障，例如表 2 中第二个测试用例。M 集合将缩减为{(A2,-,C2,D2,E1),(A2,B2,C2,D2,E1)}。

情形 3 有某两个测试用例在运行时出现故障，例如表 2 中的第二个和第四个测试用例。M 集合将缩减为{(A2,-,C2,-,E1),(A2,B2,C2,-,E1),(A2,-,C2,D2,E1),(A2,B2,C2,D2,E1)}。

类似地，我们可以讨论有多个测试用例使系统出错的情况。但如果有多个附加测试用例均产生故障，则相应的分析过程比较复杂且分析效果也比较差。不过在多数情况下，导致系统出错的测试用例数量很少(这种测试用例对软件测试来说是高质量的测试用例)，反之则说明系统质量存在很大的问题，需要重新进行广泛的分析和检查。

结束语 为了保证 Web 应用程序在各种不同的用户端配置下的正常运行，需要进行 Web 应用兼容性测试；在执行完测试用例后，还需要对测试结果进行深入分析，以迅速诊断出故障原因。

本文从归纳、分析 Web 应用兼容性测试的特性入手，并结合组合测试的基本模型和特征说明 Web 应用兼容性故障诊断的思想和步骤；在此基础上，结合我们先前工作取得的成果对 Web 应用兼容性测试结果进行分析，然后在初步分析结果的基础上补充一些附加测试用例进行重新测试，并对其结果作进一步分析和验证，从而迅速将故障原因锁定在很小的范围内。

未来的工作一方面要进行基于组合测试的 Web 应用故障诊断算法的优化、推广等工作，形成相应的测试辅助工具；另一方面继续密切关注 Web 应用发展动态，考虑新的发展热点和趋势对 Web 应用程序的影响，从而对 Web 应用兼容性测试和故障诊断工作起更好的指导和促进作用。

参 考 文 献

1 Cohen D M, Fredman M L. New Techniques for Designing Qualitatively Independent Systems. Journal of Combinational Designs, 1998, 6(6): 411～416

2 Kobayashi N, Tsuchiya T, Kikuno T. A New Method for Constructing Pair-wise Covering Designs for Software Testing. Information Processing Letters, 2002, 81(2): 85～91

3 Liu C H. A Formal Object-Oriented Test Model for Testing Web Applications:[Doctor Dissertation]. 2002

4 Nie Changhai, Xu Baowen, Shi Liang. Software Fault Diagnosis Method Based on Combinatorial Testing. Journal of Southeast University, 2003, 33(6): 681～684

5 Ricca F, Tonella P. Web Site Analysis: Structure and Evolution. In: Proc. of Int. Conf. on Software Maintenance, 2000. 76～86

6 Tai K C, Lei Y. A Test Generation Strategy for Pairwise Testing. IEEE Trans on Software Engineering, 2002, 28(1): 109～111

7 Xu Lei, Xu Baowen, Chen Zhenqiang, Jiang Jixiang, Chen Huowang. Regression Testing for Web Applications Based on Slicing. In: Proc. of the 27th Annual Int. Computer Software & Applications Conference (COMPSAC'03), 2003. 652～656

8 Xu Lei, Xu Baowen, Nie Changhai, Chen Huowang, Yang Hongji. A Browser Compatibility Testing Method Based on Combinatorial Testing. In: Proc. of the 3ird Int. Conference on Web Engineering (ICWE2003). 310～313

9 许蕾，徐宝文. Web 应用测试框架研究. 东南大学学报，2004, 34(6): 751～755

计算机科学

(1974 年 1 月创刊)

第 33 卷 (卷终) 第 12 期 (月刊)

2006 年 12 月 25 日出版

国际标准连续出版物号 ISSN 1002－137X

国内统一连续物出版号 CN50—1075/TP

定价：30.00 元 国外定价：5 美元

邮发代号：78—68

发行范围：国内外公开

主管单位：国 家 科 学 技 术 部

主办单位：国家科技部西南信息中心

编辑出版：《计 算 机 科 学》杂 志 社

重庆市渝中区胜利路 132 号 邮政编码：400013

电话：(023) 63500828 E-mail: jsjkx@swic.ac.cn

网址：www. jsjkx. com

社 长：牟炳林

总 编：彭 丹

主 编：朱宗元

主编助理：徐书令

印 刷 者：重 庆 科 情 印 务 有 限 公 司

总发行处：重 庆 市 邮 政 局

订 购 处：全 国 各 地 邮 政 局

国外总发行：中国国际图书贸易总公司 (北京 399 信箱)

国外代号：6210—MO