正交表方法在创建测试套件上应用

朱少民

http://blog.csdn.net/KerryZhu/archive/2006/10/10/1329087.aspx

http://blog.csdn.net/KerryZhu/archive/2006/10/15/1335852.aspx

在Zee的专栏里有篇文章 OATS正交表法用于测试用例设计 介绍正交表的应用。实际上，它还可以用在优化Test Suite (测试套件、测试用例组)，作用更大。

在测试中，特别是互联网应用，我们无法规定用户的环境（在一个单位局域网内，可以要求使用某种特定的操作系统、浏览器等），分布在世界各地的用户，其使用的环境是各种各样的。例如：

- 操作系统：Windows Vista, Windows XP Home, Windows XP professional, Windows 2000 pro, Windows 2000 server, Windows NT, Windows 98, Linux, Solaris 9, Solaris 10, Mac OS 9, Mac OS X
- 浏览器：IE 6.0, IE 7.0, FireFox 1.5, FireFox 2.0, Mozilla1.7, Mozilla1.4, Safari 1.3, Safari 2.0, NS 8.1
- 代理服务器/防火墙：ISA 2000, ISA 2004, Blue Coast, Cisco PIX, Linux squid, Checkpoint, ...
- 防火墙验证方式：无口令, 口令，Script, ...
- 传输协议：TCP, HTTP, SSL

除此之外，如果测试的对象是应用系统的服务器端软件平台，可能还涉及产品的客户端不同的版本、第 3 方产品的集成，以及本地化语言版本等兼容性测试：

- 客户端不同的版本： 1.0, 2.0, 3.0 <用户可能没升级，如MSN、Yahoo IM等客户端不同版本是同时存在的>
- 3方产品集成，如和OutLook/Lotus Notes的邮件/日历的集成，以Lotus Notes为例，就可能包括Notes 6.5.2、Notes 6.5.3、Notes 6.5.4、Notes 6.5.5、Notes 7.0、Notes 7.0.1
- 本地化语言：英文(EL)、中文简体（GB）/中文繁体(B5)、日文（JP）、德文(GE)、...

如果用一个完全的组合，将是爆炸性的组合，测试工作量将非常大。比如产品的功能测试用例为 1000 个，其完全组合数是

12 x 9 x 6 x 3 x 3 x 3 x 6 x 4 ＝ 419904 （40 多万）

- 操作系统 － 12
- 浏览器 － 9
- 代理服务器/防火墙 － 6
- 防火墙验证方式 － 3
- 传输协议 － 3
- 客户端版本 － 3
- 第3方产品集成，Lotus Notes － 6

- 本地化语言选4种：英文(EL)、中文繁体(B5)、日文（JP）、德文(GE)

要执行的测试用例数就从 1000 猛增到 419904000 (4 亿)个。假定每人每天(man-day)完成的测试用例数是 50，原来 1000 个测试用例，需要 20 man-day，即一个人一个月内可以完成。而 4 亿个测试用例，需要 8400000 manday， 也就是 3 万多人干一年。

进一步简化：

- 操作系统 - 考虑Windows系统的兼容性，Windows只选Windows Vista， Windows 2000，其它系统(Mac, Solaris, Linux)各选一种，共5
- 浏览器 — 各选一种，4
- 代理服务器/防火墙 — 3
- 防火墙验证方式 — 2
- 传输协议 — 1 （web 方式，只选HTTP+SSL）
- 客户端版本 — 3
- 第3方产品集成 — 3，选Lotus Notes — 6.5.3, 6..5.5, 7.0.1
- 本地化语言 — 选 2 种：英文(EL)、日文（JP）

则组合减少到 5 x 4 x 3 x 2 x 1 x 3 x 3 x 2 ＝ 2160，是原来的二百分之一的工作量（2160/419904），**但工作量依然很大，170 人干一年。但同时有带来比较大的风险，因为有许多环境，一点也没进行测试。**

如果用正交表法，工作量可进一步大大降低，相对后一种优化风险还处在较低的水平（前一种完全测试，没有风险，也不需要动脑筋）。

在上部分我们讨论了问题和面临的挑战，现在开始解决这个问题。

软件测试的目的就是发现缺陷，从理论上看，试图发现 100%的缺陷，就要对各种组合进行测试。而我们的目标没有必要设在 100%，而是通过一个优化组合去发 现 99.5%～99.9%以上的缺陷，而且产品的所有主要特性得到验证，就足够的，因为我们测试的对象是一个应用系统。从正交试验方法来看，我们将所有这 些因素（操作系统、浏览器、代理服务器/防火墙、传输协议、客户端版本、第 3 方产品集成和本地化语言）考虑进去，然后根据因素之间的强度和对测试结果的影 响程度，选定不同水平的正交表进行试验，通过分析找出对结果有影响的组合，就是要进行测试的组合。

这里正交试验方法所考察的“结果”，就是对测试所发现的缺陷的影响——每一种组合可以或可能发现的缺陷，实际也就是产品的特性对任一种组合的敏感度。我们再 将“试验”概念扩展一下，就是以前完成的测试所积累下来的数据可以看作是这种试验的可用结果之一。另外，这里主要考虑功能特性，对于非功能特性（如性能、安全性），有另外一套特定的测试用例，而且影响因素也是不一样的，如性能集中在代理服务器/防火墙、传输协议上。

为了使问题简单些，我们可以将因素分类，先在各个类别进行正交组合试验和分析，然后再进行大类别的正交组合设计，使之整个测试套件的设计策略得到最优实现。

1．测试用例的分类

为了更准确分析组合并降低风险，可以将测试用例从两个角度进行分类：用例的优先级和用例的目标域。

用例按优先级（Priority）可以分为 3 类：

Cp1－最高级，这些测试用例所发现的缺陷是致命的——必须在当前版本修正后

Cp2－中级，这些测试用例所发现的缺陷是较严重的——必须在下一个版本修正

Cp3－最低级，这些测试用例所发现的缺陷是一般的——有时间最好修正

用例按目标域（Objectives）可以分为 3 类：

Cu – 测试客户端功能界面的用例

Cl – 测试客户端功能逻辑的用例

Ci – 测试客户端集成（接口）的用例

当然，针对产品的不同特点/规模等，可以进行不同的、更多的层次分类。

2．操作系统和浏览器的正交性试验结果

OS\Browser	IE7.0	IE6.0 SP1	Firefox 2.0	Firefox 1.5	Mozilla 1.4	Mozilla 1.7.13	NS 8.1	Safari 1.3	Safari 2.0
Windows									
Windows Vista	Y	Y	Y			Y	Y		
Windows XP SP2	Y	Y	Y	Y		Y	Y		
Windows 2000 SP4 Update Rollup 1		Y		Y		Y	Y		
Windows 2003 Server R2		Y		Y		Y			
Mac									
Mac OS X Panther (10.3)			Y	Y	Y		Y	Y	
Mac OS X 10.4 (Power PC)			Y	Y			Y	Y	Y
Mac OS X 10.4.6 (Intel native)			Y	Y			Y		Y
Solaris									
Solaris 9			Y	Y	Y	Y			
Solaris 10			Y	Y		Y			
Linux									
Red Hat Enterprise Linux 4			Y	Y	Y	Y	Y		
SuSE Linux 10			Y	Y	Y	Y	Y		

其次可以根据操作系统和浏览器在市场的分额、所发现的缺陷等数据来优化这个组合。根据市场数据，假定操作系统和浏览器在市场的分额是：

Windows Vista	5%		IE 7.0	5%
Windows XP SP2	60%		IE 6.0 SP1	55%

Windows 2000 SP4 Update Rollup 1	**20%**		**Firefox 2.0**	**5%**
Windows 2003 Server R2	**1%**		**Firefox 1.5**	**15%**
Mac OS X Panther (10.3)	**2%**		**Mozilla 1.4**	**3%**
Mac OS X 10.4 (Power PC)	**2%**		**Mozilla 1.7.13**	**6%**
Mac OS X 10.4.6 (Intel native)	**1%**		**NS 8.1**	**5%**
Solaris 9	**1%**		**Safari 1.3**	**2%**
Solaris 10	**2%**		**Safari 2.0**	**4%**
Red Hat Enterprise Linux 4	**3%**			
SuSE Linux 10	**8%**			

可以得到新的量化的正交表，可以消除所有低于 0.2%的组合：

OS\Browser		IE7.0	IE6.0 SP1	Firefox 2.0	Firefox 1.5	Mozilla 1.4	Mozilla 1.7.13	NS 8.1	Safari 1.3	Safari 2.0
Windows		5%	55%	5%	15%	3%	6%	5%	2%	4%
Windows Vista	5%	**0.25%**	**2.75%**	**0.25%**			**0.30%**	**0.25%**		
Windows XP SP2	60%	**3.00%**	**33.00%**	**3.00%**	**9.00%**		**3.60%**	**3.00%**		
Windows 2000 SP4 Update Rollup 1	20%		**11.00%**		**3.00%**		**1.20%**	**1.00%**		
Windows 2003 Server R2	1%		**0.55%**		0.15%		0.06%			
Mac										
Mac OS X Panther (10.3)	2%			0.09%	**0.26%**	0.05%		0.09%	0.03%	
Mac OS X 10.4 (Power PC)	2%			**0.10%**	**0.30%**			0.10%	0.04%	0.08%
Mac OS X 10.4.6 (Intel native)	1%			0.06%	0.18%			0.06%		0.05%
Solaris										
Solaris 9	1%			0.05%	0.15%	0.03%	0.06%			
Solaris 10	2%			0.10%	**0.30%**		0.12%			

OS\Browser		IE7.0	IE6.0 SP1	Firefox 2.0	Firefox 1.5	Mozilla 1.4	Mozilla 1.7.13	NS 8.1	Safari 1.3	Safari 2.0
Linux										
Red Hat Enterprise Linux 4	3%			0.15%	**0.45%**	0.09%	0.18%	0.15%		
SuSE Linux 10	8%			**0.40%**	**1.20%**	**0.24%**	**0.48%**	**0.40%**		

进一步，根据产品在相应的历史缺陷数据(所占比率——百分比值)，可以知道测试用例目标域各个分类（Cu、Cl、Ci 等）在各个平台的正交数据，进一步优化组合。

3．网络协议和代理服务器的正交性试验

按同样道理，可以得到相应的矩阵。由于篇幅所限，不再详细叙述。

4．客户端版本、第 3 方产品集成和本地化语言正交性试验

对于客户端版本的构成，可以 1.0 版本为基线，2.0 版本可以看作“1.0 版＋新增功能＋修改/影响区域”，所以对于 2.0 版本的测试集中在“新增功能”、“修改/影响区域”。对于 3.0，同样类推。

对于第 3 方产品、本地化语言进行分析，可以和客户端软件“1.0 版本、2.0 新增功能、2.0 修改/影响区域、3.0 新增功能、3.0 修改/影响区域”5 项因素构成正交表。

5．综合形成一个集成的测试套件

根据上述 4 个方面的分析，可以构成优化的组合。为了更有效地提高测试效率，还可以将测试组合分解到多个阶段（如新功能验证阶段、第 1 回归测试阶段、第 2 回归测试阶段），例如：

新功能验证阶段：

- 英文 ＋ 3.0新增功能 ＋ 协议/防火墙强组合 ＋ 操作系统和浏览器强组合
- 英文 ＋ 3.0修改/影响区域 ＋ 协议/防火墙中组合 ＋ 操作系统和浏览器中组合

第 1 回归测试阶段：

- 中文 ＋ 2.0新增功能 ＋ 协议/防火墙强组合 ＋ 操作系统和浏览器中组合
- 中文 ＋ 2.0修改/影响区域 ＋ 协议/防火墙中组合 ＋ 操作系统和浏览器强组合

第 2 回归测试阶段：

德文 ＋ 1.0（基本功能）＋ 协议/防火墙弱组合 ＋ 操作系统和浏览器弱组合

组合强弱，可以按照缺陷数概率和本身（协议/防火墙或操作系统和浏览器）组合概率的乘积获得的值来决定，值越高，说明其组合强度越高，应该优先得到测试。