

基于组合的业务分析及自动生成用例的方法

马 珏

(江苏广播电视大学, 江苏 南京 210036)

摘 要: 给出了一种可自动生成用例的业务分析方法. 用组合表的方法分析业务, 充分考虑了相关联业务的各种取值及各种影响关系, 在避免重复用例的情况下大大增加了测试用例的覆盖度, 保证了被测系统的质量. 根据上面的算法, 在计算机中实现了软件原型, 在具体业务中, 将通过组合分析所产生的业务分析表导入到程序中, 通过自动分析, 得出基于组合和组合规则的测试用例.

关键词: 业务分析; 测试用例; 自动化生成用例; 组合; 用例覆盖

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2012)03-0121-04

Combination-Based Method of Business Analyses and Cases Generated Automatically

MA Jue

(Jiangsu Radio and Television University, Nanjing 210036, China)

Abstract: a kind of business analysis method is mentioned that can generate cases automatically, which analyzes business with the combination table. The method is a consideration of various kinds of associated values and influences. The method can significantly increase the testing coverage and ensure the quality of the tested system without repeating cases. According to the algorithm, We make a prototype of software which can produce the combination-based testing cases, by importing Business analysis form produced from combination-based analysis to it.

Key words: business analysis; test cases; automatic generation cases; combination; cases covered

1 引言

测试用例的设计, 在整个测试过程中占有举足轻重的作用, 测试用例的覆盖度直接影响到被测软件的稳定性. 而现在大多数测试用例的设计方法都采用按需求或概要设计, 得到软件功能划分图; 然后针对每个功能点, 采用等价类划分、临界值、因果图等方法来设计用例.

实际项目中, 撇开测试数据的设计不谈, 仅就测试项来说, 对同一个项目, 有经验的测试人员, 在写用例或测试时总会有更多的测试考虑点, 从而发现系统中存在的更多问题; 如果测试人员缺乏经验, 在表面上看, 测试用例的选择好象已经把页面所有信

息的测试都考虑到了, 实际上却还是遗漏了大量测试覆盖点, 导致其测试出来的程序总是比较脆弱^[1]. 特别是一些后台业务复杂, 而且在页面中并无太多体现的系统, 只要系统的条件有一点改变, 则页面的用例会大不相同, 在这种情况下, 我们必须列出与被测项相关联的业务条件及相互影响的关系, 来一一找到这些隐藏点并予以测试, 才能真正地保证我们的测试覆盖度.

本文给出了一种可自动生成用例的业务分析方法, 用组合表的方法分析业务, 充分考虑了相关联业务的各种取值及各种影响关系, 在避免重复用例的情况下大大增加了测试用例的覆盖度, 保证了被测系统的质量.

收稿日期: 2011-07-21; **修回日期:** 2011-08-22

基金项目: 江苏省高校自然科学基金(09KJB510021)

2 相关性研究

在测试过程中,常用的是黑盒测试的用例设计方法,主要有划分等价类、边界值分析法、错误推测法、因果图方法等。

(1)划分等价类:等价类是指某个输入域的子集合。在该子集合中,各个输入数据对于揭露程序中的错误都是等效的,并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试。该方法精减了用例,但是只针对单个字段,并未联系上下文的组合关系^[2]。

(2)边界值分析法:通常输入和输出等价类的边界,就是应着重测试的边界情况^[3]。应当选取正好等于,刚刚大于或刚刚小于边界的值作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据。

以上等价类划分法和边界值分析方法都是着重考虑输入条件,但没有考虑输入条件的各种组合、输入条件之间的相互制约关系。这样虽然各种输入条件可能出错的情况已经测试到了,但多个输入条件组合起来可能出错的情况却被忽视了。

(3)错误推测法:基于经验和直觉推测程序中所有可能存在的各种错误,从而有针对性的设计测试用例的方法。错误推测法适合于经验丰富且对被测系统很熟悉的测试人员。

(4)因果图方法:一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例。因果图法考虑了输入条件及输出条件的组合关系,用例覆盖完整,但对于规模比较大,条件较多的程序来说,很难直接用因果图来画出业务关系,效果不清晰,且产生的冗余用例较多。

还有一些组合覆盖用例的方法中,也只是利用边界值分析、等价类划分等方法将参数离散化,生成最终的测试用例^[4]。

上述一些方法是本文的一个基础,本文在单个控件取值中采取了等价类划分法和边界值分析方法,在业务分析上借鉴了因果图的思路,用 EXCEL 表格作为分析工具,加上了前置条件、页面条件及验证点,再将这些条件及验证点进行组合的时候,再用一个组合规则,将不必要的用例进行约束,从而产生的用例无论是从字段输入值,到业务条件组合,再到用例冗余,都做到了既精减又覆盖全面。

3 基于条件组合的业务分析

在很多被测系统中,界面比较简单,但其业务特

别复杂,而且业务的各种条件之间相互影响,基于条件组合的业务分析能够清晰的反映待测系统的各种组合情况,并能够根据业务分析轻松的生成各种不同有针对性的测试用例,在避免重复用例的情况下最大限度的扩充用例。

基于条件组合的业务分析,主要有三个部分组成:第一部分要列出待测场景的各个前置条件,前置条件就是在测试场景中没有明显出现的字段,但是又与本场景有着密切的关系,前置条件的改变,可能导致被测场景的用例取值的完全改变。第二部分是页面条件,页面条件包含页面上可以直接看到的字段,页面条件的取值可以按照一般黑盒测试用例的设计方法去设计取值,如等价类化分法,边界值化分法等。第三部分是验证点,验证点就是在场景中进行操作后,对系统产生了哪些影响。

用一个表可以简单的描述业务分析的过程,如表 1。

表 1 业务分析表

	取值	取值	取值
前置条件 A	TA1	TA2	
前置条件 B	TB1	TB2	TB3
...			
页面条件 A	PA1	PA2	PA3
页面条件 B	PB1	PB2	
...			
验证点 A	CA1	CA2	
验证点 B	CB1		
...			

上述业务分析表实现了对测试场景所关联的所有条件进行了组合,按照笛卡尔积推导 $D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) | d_i \in D_i, i=1, 2, \dots, n\}$, 如果按上表已知条件进行组合覆盖,则会产生 72 个用例。事实上,不是每个条件都相互影响,那么就会有許多冗余的用例产生,所以我们还要对上表进行细分,对相关的条件进行组合,可以细分这多个组合表。例如前置条件 A 只会影响页面条件 A,而前置条件 A 的取值和改变与页面条件 B 无关,则在组合时没有必要把它们放在一起。这样可以简单的把上述业务分析细分为表 2 和表 3。

细分之后的两个组合表,则会产生的用例个数

为 $2 \times 3 \times 2 \times 1 + 3 \times 3 \times 2 \times 2 \times 1 = 48$, 则减少了 24 个冗余的用例。在实际应用中, 可能还会有更复杂的情况产生, 比如上述表 2 中前置条件 B 中的三个取值, 和页面条件 A 中的三个取值, 只有部分组合是有意义的。那么这种情况, 可以列出组合规则二维表来明确直观的表达。表 4, 把有意义的组合用 T 来表示, 没有意义的组合用 F 来表示。

表 2 组合表一

	取值	取值	取值
前置条件 A	TA1	TA2	
页面条件 A	PA1	PA2	PA3
验证点 A	CA1	CA2	
验证点 B	CB1		

表 3 组合表二

	取值	取值	取值
前置条件 B	TB1	TB2	TB3
页面条件 A	PA1	PA2	PA3
页面条件 B	PB1	PB2	
验证点 A	CA1	CA2	
验证点 B	CB1		

表 4 组合规则表

	TB1	TB2	TB3
PA1	T	T	F
PA2	T	T	F
PA3	F	T	F

这样, 在业务分析阶段, 就能清晰的知道, 需要多少个测试用例去覆盖, 不会有遗漏的测试点及验证点, 在实现测试用例最大化覆盖的同时又优化了测试用例^[5]。根据这样的业务分析表格, 可以很轻易的生成用例。

4 根据业务分析生成用例的算法

基于上面的业务分析表格, 不再需要手工编写用例, 组合的业务分析方法及细化的条件让我们可以轻而易举的实现用例生成的自动化, 大大减少了人力成本, 下面给出自动生成用例的算法流程, 如图 1 所示。

(1) 首先根据业务分析表读出组合表和组合规

则表放入数组;

(2) 根据组合表个数进行单个循环;

(3) 用递归算法对组合表进行遍历, 当遍历到最后一层(最后一个组合条件)时, 该遍历路径就是一个用例, 最后一层数据遍历完之后, 再返回上一个组合条件的下一个取值。递归具体算法如下(VB):

```

'参数 shCreate: 写入用例的 EXCEL 页面对象
'参数 lev: 规则中条件的下标
'参数 route: 数组, 记录遍历路径
Public Function CreateTestcase (shCreate, lev, byref route)
    For aa=1 to UBound(rule(lev))
        route(lev)=rule(lev)(aa)
        If lev=UBound(rule) Then
            WriteExceptReqRow shCreate, route
        Else
            CreateExceptGZ shCreate, lev+1, route
        End If
    Next
End Function

```

(4) 在用例写入测试用例文件之前, 还需判断一下该用例是否在组合规则表中, 若不在组合规则表中, 说明没有限制该组合, 则写入用例文件; 在组合规则表中若为 T, 写入用例, 若为 F, 则丢弃该用例。

5 实践效果

根据上面的算法, 在计算机中实现了软件原型, 在具体业务中, 将通过组合分析所产生的业务分析表导入到程序中, 通过自动分析, 得出基于组合和组合规则的测试用例。

在实践过程中, 自动生成用例的工具不仅避免了在编写测试用例过程中出现的人为错误或遗漏, 而且为程序测试员节约了大量的时间, 在用例复杂的测试环境中显示出较大的优势。

下面是某证券交易系统三个页面在测试用例数量上的分析, 在原有用例的基础上, 使用本文所述业务分析方法, 对系统页面进行重新分析, 扩充原有用例, 选了三个测试场景进行比较, 页面一到页面三是从简单到复杂的页面业务条件, 见表 5。

表 5 组合规则表

被测页面	业务点	未使用组合分析前用例数	使用组合分析后用例数	使用组合分析加上限制规则后用例数
页面一	3	5	16	10
页面二	7	12	56	19
页面三	12	32	259	89

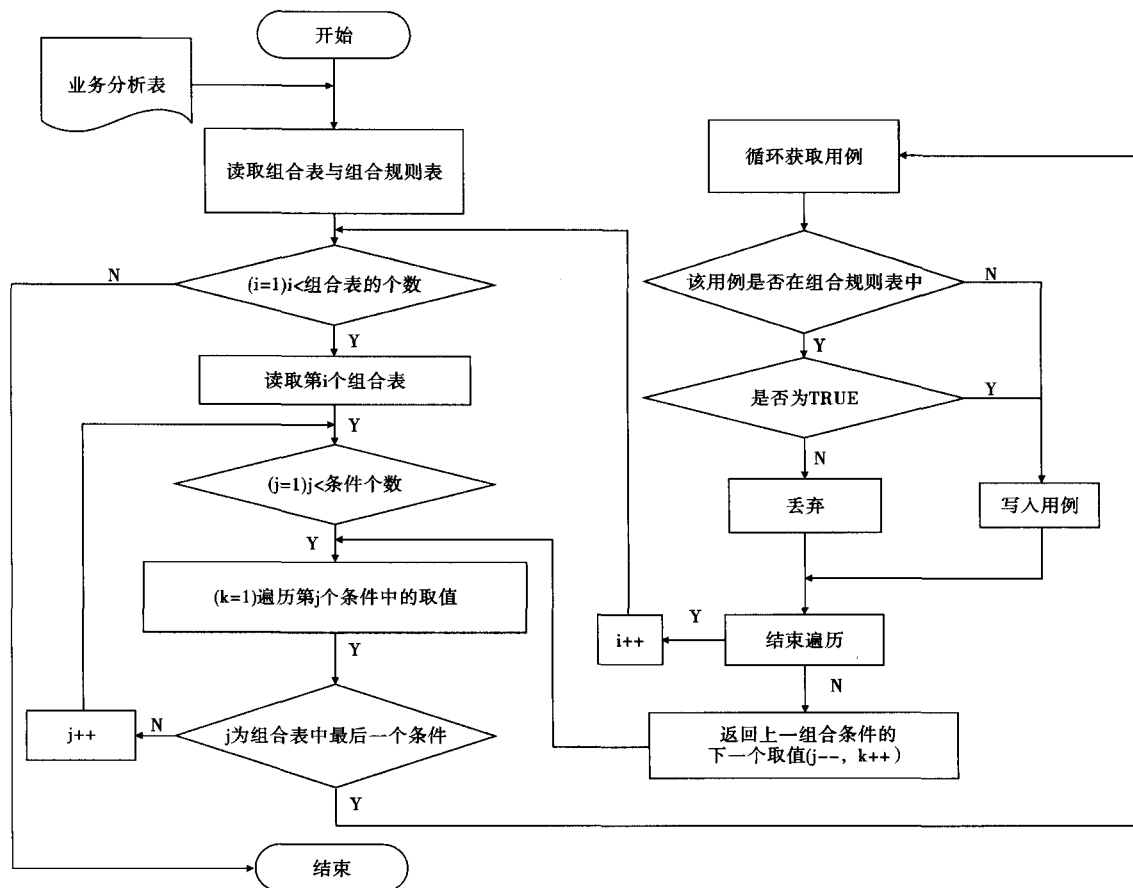


图1 自动生成用例的算法流程图

可以明显的看出,在使用了组合业务分析,再加上限制规则后,可以在精减用例的基础上,全面覆盖测试点,在复杂的业务环境中(页面三)更能显示出它的优势,减少了不必要的测试用例,但又不遗漏每一个测试功能点。

6 结束语

本文给出了基于组合的测试用例分析及自动生成用例的算法. 在具体的输入取值方面,延用了等价类划分,边界值等方法,保证了基本输入值的覆盖面及精减度,在业务分析方面,采用了组合表的方法对前置条件、页面条件及验证点进行各个可能性的组合,保证了与页面相关的各个条件及各个条件的组合都不会被漏测,在精减用例方面,为了避免冗余用例及无效用例的产生,加上了组合规则表进行限制,做到了测试用例的精简与全面覆盖并存。

自动生成测试用例的方法,减少了人力成本,节约了时间,避免错误的发生. 在实践中,该方法投入成本小,用例覆盖全面精简,在业务较复杂的测试中

取得了不错的实际效果。

参考文献:

- [1] 尹文洁, 崔冬华. 测试用例集的约简方法研究[J]. 电脑开发与应用, 2011, 24(5): 18-20.
- [2] 李健, 石冬琴. 软件黑盒测试方法研究及应用[J]. 中国高新技术企业, 2011, 18(4): 27-29.
- [3] 李革, 郑立峰. 几个 RIT 软件的测试与比较[J]. 计算机仿真, 2007, 24(1): 114-119.
- [4] 张卫祥, 刘文红. 一种基于组合覆盖的黑盒测试用例自动生成方法[J]. 飞行器监测学报, 2008, 27(5): 53-56.
- [5] 崔霞, 高建华. 一种新的测试集简化的测试覆盖准则[J]. 计算机科学, 2009, 36(1): 244-246.
- [6] 肖利琼. 软测之魂: 核心测试设计精解[M]. 北京: 电子工业出版社, 2011.

作者简介:

马珏女, (1976—), 讲师, 硕士. 研究方向为软件工程和网络技术。