

笔记本电脑自动化测试的任务分配模型与方法^①



黄永锋¹, 孙自诚¹, 罗保国²

¹(东华大学 计算机科学与技术学院, 上海 201620)

²(中国科学院 软件研究所, 北京, 100190)

通讯作者: 黄永锋, E-mail: 741239226@qq.com

摘要: 笔记本电脑的软件和硬件测试是笔记本在大规模生产前一个非常重要的环节. 由于笔记本电脑是一个集成度较高的产品, 测试过程的重复性和复杂性就成了笔记本测试的一个主要特征. 针对这一特点设计了一种测试任务模型并实现了相应的自动化分配算法. 首先根据测试要求将相应的测试任务进行整合, 形成一个由多种测试条件组合的数据表; 然后再根据每台笔记本的状态进行自动任务分配. 测试员只需根据提示完成相应测试任务即可, 无需对任务进行分类测试这一繁琐的过程. 通过多台机器的实际测试, 验证了分配算法的实用性, 使测试效率有了明显提高.
关键词: 笔记本测试; 自动化测试; 任务模型; 分配算法

引用格式: 黄永锋, 孙自诚, 罗保国. 笔记本电脑自动化测试的任务分配模型与方法. 计算机系统应用, 2019, 28(6): 183-188. <http://www.c-s-a.org.cn/1003-3254/6960.html>

Task Assignment Model and Method for Laptop Automated Testing

HUANG Yong-Feng¹, SUN Zi-Cheng¹, LUO Bao-Guo²

¹(Computer Science and Technology Academy, Donghua University, Shanghai 201620, China)

²(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: The software and hardware testing of laptop is a very important part of its production before mass production. Because the laptop is a highly integrated product, the repeatability and complexity of the test process has become a major feature of its testing. For this feature, an automated test task model is designed and the corresponding allocation algorithm is implemented. First, according to the test requirements, the corresponding test tasks are integrated to form a data table composed of a plurality of test conditions; then, the automatic task assignment is performed according to the state of each laptop. The tester only needs to complete the corresponding test task according to the prompts, without the tedious process of classifying the task. Through the actual test of multiple machines, the practicality of the allocation algorithm is verified, and the test efficiency is obviously improved.

Key words: laptop test; automated test; task model; allocation algorithm

引言

随着计算机技术的飞速发展, 笔记本电脑的功能越来越强大, 结构也越来越复杂, 各功能出现的错误种类也随之增加, 这无疑为其软硬件的测试带来诸多不便^[1]. 传统的手工测试已逐渐不能满足日益增多的测试任务需求, 因此自动化任务分配就成了笔记本软硬件

测试的必然方向, 是笔记本高质量、高可靠性的重要保障^[2,3]. 在以往的手工测试过程中, 测试员需要自主区分笔记本的工作状态对每个检查点进行测试, 这无疑是对测试员体力和精力上的双重消耗^[4].

近年来, 软件自动化测试技术的出现, 为传统的笔记本测试提供了很大帮助, 节省了更多的人力和财力,

① 收稿时间: 2018-12-18; 修改时间: 2019-01-10; 采用时间: 2019-01-28; csa 在线出版时间: 2019-05-25

同时也提高了测试的质量,缩短了研发和上市的周期^[5,6]。但完整的自动化测试工具除了应该包含自动化测试功能外,还应具有适应笔记本在多种状态下测试的任务自动分配功能。任务自动分配是自动化测试中非常重要的一环,可以进一步减少资源开销,使得自动化测试更加有意义^[7]。

因此,本文以笔记本测试的实际需求为切入点,设计了适应笔记本不同状态和随机性测试的任务模型并实现了自动分配的算法,使得每次测试任务与每组测试条件间形成一种映射。这种任务分配方式既能满足测试过程中各种条件的约束,真实的覆盖所有测试情况^[8],又能使测试员从复杂的任务分配中解放出来,并且能更好的避免任务分配重复或不足的现象,提高测试效率^[9]。

1 测试任务的模型构建

笔记本测试的目的是通过较真实地模拟用户的使用场景,以发现笔记本电脑中比较隐藏的问题和其发生的概率。通过分析测试要求,首先提出了用于构建模型的基础概念和条件,然后再根据各种条件的叠加关系计算实际的任务数据,最后再以表格的形式较直观的展示模型中数据的分布情况^[10]。

1.1 任务模型概念与条件

由于测试任务的复杂性,了解相关概念是进行模型构建的前提。本文基于实际的测试场景,提出了以下概念和条件:

(1) 检查点:笔记本电脑每个测试周期中的测试点,如相机、麦克风、触摸板、触摸屏等。由于检查点具有扩展性,因此本文将十个检查点为例构建任务模型。

(2) 测试周期:每个测试周期由多个仅测试一次的检查点构成。测试周期是测试任务的重要组成部分,也是笔记本完成一次测试的基本单位。

(3) 检查点测试百分比:每个测试周期中,某个检查点被测试的概率。

(4) 机器类型:不同机器型号的笔记本,测试的检查点不同。机器类型为同一类型的机器所测试检查点的集合。

(5) 测试计划:现阶段笔记本一般都有多种使用模式,如正常笔记本模式和平板电脑模式等,每种模式的测试可作为一个测试计划。本文将正常笔记本模式为例构建任务模型。

(6) 测试状态:测试状态是指在执行每个测试周期前笔记本的状态,常见的有睡眠、休眠、重启等。一种测试计划是由多种测试状态组成。

(7) 状态属性:状态属性描述的是笔记本以哪种方式进入测试状态,例如笔记本可以通过电源按钮、开始菜单和关闭盖子等方式进行睡眠状态。

(8) 电量区间:测试时该笔记本所处的电量区间。电量区间可根据不同的测试需求进行动态调节,本文将电量区间分为接通电源和断开电源下1%~50%,51%~100%这4种情况。

(9) 测试任务:一次测试任务由相应电量区间下的测试状态,状态属性和测试周期3部分组成。每部分的具体参数值可通过算法计算得出。

在笔记本测试过程中,除了包含以上9种概念外,任务分配还需要满足一定的条件,即根据测试对象制定相应的策略^[11],具体如下:

(1) 在同一电量区间下,测试状态共有集中测试,随机测试和轮换测试3种测试策略。集中测试是指连续多个测试周期均在一种测试状态下测试,直至此种状态规定的次数测试完毕才进入下一种状态测试;随机测试是指每个测试周期随机选择一种测试状态;轮换测试是指每种测试状态根据测试周期依次选择。

(2) 在一种测试状态下进行测试时,状态属性共有3种测试策略,即集中测试,随机测试和轮换测试,形式与测试状态的选择类似。

1.2 测试项计算方法

通过上述的概念可知,测试周期虽然为一次测试的基本单位,但由于检查点测试百分比和机器类型这两个因素的存在,每个周期内实际测试的检查点也具有差异性。在本节中,暂且抛开测试周期之间的异同之处,具体分析测试状态、测试属性和电量区间这3个因素在测试周期上的交互关系,并将满足相同因素的所有测试周期的集合称为一个“测试项”。文献^[12]采用“成对测试算法”来降低测试条件组合数,但笔记本测试是多条件组合且多次重复的测试,因此该方法并不适用,仍需考虑完整的条件覆盖。

为了方便描述,不妨将测试状态的个数用 N_s 表示,每种测试状态设为 S_i ;同理测试属性的个数为 N_a ,每种测试属性为 A_j ;四种电量区间为 B_1 ~ B_4 ,每个电量区间对应的电量数值大小分别为 N_{b1} ~ N_{b4} 。再将一个测试计划的总测试周期数设为 T ,满足这3种条件下的测

试项包含的周期数设为 C_{ijk} , 则 C_{ijk} 与 S_i 、 A_j 、 B_k 和 T 的关系为:

$$C_{ijk} = T \times f(S_i, A_j, B_k)$$

其中, $1 \leq i \leq N_s, 1 \leq j \leq N_a, 1 \leq k \leq 4$ 且 i, j, k 为整数

$$f(S_i, A_j, B_k) = 1/N_s \times 1/N_a \times N_{bk} / (2 \times 100) \quad (1)$$

若因特殊的测试需求, 测试状态、测试属性和测试电量区间不再根据数值平均分配, 而是根据测试方提供的相应比率进行分配, 那么只需改变 $f(S_i, A_j, B_k)$ 的函数式即可. 不妨将不同测试状态下测试周期数所占比率的命名为 P_{si} , 不同测试属性所占的比率为 P_{aj} , 4 种电量区间所占的比率为 $P_{b1}-P_{b4}$. 则 $f(S_i, A_j, B_k)$ 的函数式为:

$$f(S_i, A_j, B_k) = P_{si} \times P_{aj} \times P_{bk}$$

其中, $1 \leq i \leq N_s, 1 \leq j \leq N_a, 1 \leq k \leq 4$ 且 i, j, k 为整数

通过以上的计算方法, 可以在满足不同的测

试条件下将一个测试计划的总测试周期数分为 $N_s \times N_a \times 4$ 种测试项, 每个测试项的测试周期数 C_{ijk} 与测试条件 S_i 、 A_j 、 B_k 相关.

1.3 生成模型数据表格

为了方便数据的展示, 采用二维表格形式显示模型数据. 可将十个检查点设置成为二维表格的列, 测试项为表格的行. 并且将测试状态与状态属性两个条件进行合并, 然后再与电量区间条件进行组合. 通过以上分析, 将测试项和检查点测试数表示成一个 $10 \times (N_s \times 4)$ 的二维测试数据表格 (10 表示为十个检查点), 如表 1 所示 (注意: 数据可根据具体测试需求进行更改; “触控笔”检查点表示只有部分笔记本支持此项测试, 在分配任务时, 需要根据包含此检查点的机器类型, 将此检测点的测试次数 100% 的分配给特定的笔记本测试).

表 1 正常笔记本测试模式测试数据表

测试状态	状态属性	电源	电量区间	检查点百分比										
				键盘	鼠标	触摸板	触摸屏	指纹	USB 接口	相机	麦克风	扬声器	触控笔	
				100%	100%	100%	100%	100%	50%	20%	20%	20%	100%	
睡眠	总次数: 1200	接通	1%-50%	300	300	300	300	300	300	150	60	60	60	15
	电源按钮: 30%		51%-100%	300	300	300	300	300	300	150	60	60	60	15
	开始菜单: 30%	断开	1%-50%	300	300	300	300	300	300	150	60	60	60	15
	关闭盖子: 40%		51%-100%	300	300	300	300	300	300	150	60	60	60	15
休眠	总次数: 800	接通	1%-50%	200	200	200	200	200	200	100	40	40	40	10
	电源按钮: 30%		51%-100%	200	200	200	200	200	200	100	40	40	40	10
	开始菜单: 30%	断开	1%-50%	200	200	200	200	200	200	100	40	40	40	10
	关闭盖子: 40%		51%-100%	200	200	200	200	200	200	100	40	40	40	10
重启	总次数: 400	接通	1%-50%	100	100	100	100	100	100	50	20	20	20	5
	电源按钮: 40%		51%-100%	100	100	100	100	100	100	50	20	20	20	5
	开始菜单: 60%	断开	1%-50%	100	100	100	100	100	100	50	20	20	20	5
			51%-100%	100	100	100	100	100	100	50	20	20	20	5

2 测试任务生成方法

在根据测试特点建立了测试任务模型之后, 则需要生成待测笔记本可以执行的测试任务. 依照测试条件, 电量区间由测试机决定, 不可改变, 因此不作计算; 测试状态和状态属性根据选择的测试策略动态调节; 测试周期将根据机器类型和检查点测试百分比计算生成. 经过上述过程, 生成一份完整的测试任务, 流程如图 1.

2.1 生成测试状态

测试状态有集中测试, 随机测试和轮换测试 3 种测试策略, 具体选择哪一种方式可根据测试需求而定. 可以采用一个十进制数来表示每次任务选择的测试状

态. 可继续使用 N_s , N_a , P_{si} 等变量的含义, 另外用 S 表示该十进制 (S 可取的值为 $1-N_s$), 用 S_{Bik} 表示每种测试状态下每个电量区间内已测的周期数 (表 1 中 S_{Bik} 的个数为数据行数, 即 3×4 , 共 12 种). 则 3 种策略下 S 的计算方式为:

(1) 集中测试: 测完一个测试状态的所以周期数再切换到下一个测试状态.

$$S = \{i | num(i) > 0, 1 \leq i \leq N_s\}$$

(2) 随机测试: 随机测试策略下状态的选择是根据 P_{si} 的值来选择的.

$$S = \text{Min}\{i | \text{Random}(1) < \sum_{i=1}^{Ns} Psi\}$$

其中, S 等于满足 $\text{Random}(1) < \sum_{i=1}^{Ns} Psi$ 的最小 i 值.

(3) 轮换测试: 根据当前已测的周期数对测试状态个数取模得到结果.

$$S = S B i k \% N s$$

其中, $1 \leq i \leq N s, 1 \leq k \leq 4$.

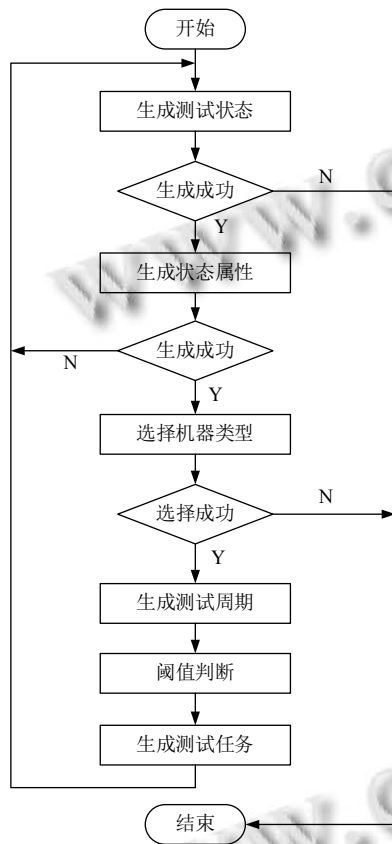


图1 生成测试任务流程图

2.2 生成状态属性

测试属性也有集中测试, 随机测试和轮换测试3种策略. 可用一个十进制变量 A 表示每次任务所选择的属性, $A B j k$ 表示每种属性下每个电量区间内已测的周期数. 3种策略下 A 的计算方式如下:

(1) 集中测试:

$$A = \{j | num(j) > 0, 1 \leq j \leq Na\}$$

(2) 随机测试:

$$A = \text{Min}\{j | \text{Random}(1) < \sum_{j=1}^{Na} P a j\}$$

其中, A 等于满足 $\text{Random}(1) < \sum_{j=1}^{Na} P a j$ 的最小 j 值.

(3) 轮换测试:

$$A = A B j k \% N a \tag{2}$$

其中, $1 \leq j \leq N a, 1 \leq k \leq 4$.

2.3 选择机器类型

机器类型是依据待测试的笔记本电脑的型号和功能, 选择所需测试的检查点集合, 一种集合即为一种机器类型. 该集合可用十位二进制数来描述, 每个检测点对应一位, 若对应二进制位的检查点属于此集合, 则该位为1, 反之则为0. 通过上述描述不难发现, 机器类型与待测机之间为一种一对多的关系. 在所有测试开始之前, 可以根据待测笔记本的特点, 设置多个机器类型, 然后在任务分配时, 根据待测笔记本特征选择一种机器类型进行计算.

2.4 生成测试周期

生成测试周期首先根据当前测试机所属机器类型的二进制数, 选择测试的检查点. 若某个检查点对应位置为0, 则直接不测; 若为1, 则进一步考虑此检查点的测试概率. 可将当前分配任务的测试机所选择的机器类型中每个检查点的值用 $M T m$ 表示, 相应地, 所生成的测试周期中每个检测点的值用 $C P m$ 表示, 每个检测点测试百分比用 $P c p m$ 表示, 则 $C P m$ 的计算方法为:

$$C P m = M T m \times (\text{Random}(1) < P c p m ? 1 : 0)$$

其中, $1 \leq m \leq 10$.

2.5 阈值判断

通过上述算法, 可以发现无论是测试状态和状态属性在随机测试策略下的计算, 还是测试周期中每个检查点的计算, 都用到了随机数 $\text{Random}(1)$ 作为参数. 由于随机概率具有不确定性的特点, 很难保证所有任务都能严格按照相应的百分比分布, 因此, 还需考虑设置一个阈值, 并且在每次生成任务时进行阈值判断. 由于测试状态和状态属性在计算测试任务时都会选择其中的一种, 且当某一种测试周期数测试完毕后, 之后的测试任务就只会剩余的选项中选择, 故不会存在漏测的情况. 对于检查点的计算, 当一个检查点的测试百分比小于100%时, 每次任务中此检查点是否测试完全依照随机概率和测试百分比计算得出, 因此会出现

漏测的情况. 可为每个测试项设置一个阈值 $TH=5$, 在计算该测试项内某个检查点是否测试时, 若满足条件:

$$Remainder(C_{ijk}) - Remainder(CP_m) \leq TH \quad (3)$$

其中, $Remainder(C_{ijk})$ 为测试项剩余周期数, $Remainder(CP_m)$ 为检查点剩余周期数

则在接下来的任务分配中, 该检查点的测试百分比会默认升为 100%, 即每次任务都会测试. 可重新计算 CP_m 的值:

$$CP_m = MT_m$$

2.6 生成测试任务

通过上述算法, 可以在每一次任务分配时, 依据选择的测试策略计算出这次任务的测试状态和状态属性, 再根据机器类型和检查点测试百分比生成每个检测点的测试情况并组合成了一个测试周期. 为了表示方便, 测试周期也用十位十进制数替代二进制数来表示, 每位数只有 0 或 1 两种数值. 则每次测试任务的组成结构为:

$$S | A | CP_1 CP_2 \dots CP_{10}$$

算法伪代码如下所示:

```
GetTask(battery) { //battery 为电池电量区间
s, a=getPara(battery); //获取测试状态和状态属性参数
//生成测试状态 state 和状态属性 attr
switch(strategy) { //选择的测试策略
case 1://集中测试
state=s;
attr=a;
case 2://随机测试
if(sum_p(s-1)<random(1)<=sum_p(s))
state=s;
```

```
if(sum_p(a-1)<random(1)<=sum_p(a))
attr=a;
case 3://轮换测试
state=(s++)%num(s);
attr=(a++)%num(a);
}
while(c<num(c)) { //生成测试周期 checkPoint[]
checkPoint[c] = MT[c]*(random(1)<p(c)?1:0);
if(getThreshold(c))//阈值判断
checkPoint[c]= MT[c];
}
task=state+attr+checkPoint[]; //生成测试任务
}
```

3 实例应用与评估

3.1 系统应用

目前, 采用该模型和算法的系统已应用于上海企顺信息系统有限公司的实际项目中, 该企业为联想、戴尔、英特尔等笔记本生产商提供测试服务. 以某型号笔记本的测试为例, 在测试中共有 1 台笔记本用作任务分配的服务器, 18 台用作领取测试任务的测试机. 测试状态为集中测试策略, 状态属性为轮换测试策略, 测试检查点个数为 15 个. 所采集的睡眠测试状态下测试任务总计数据如表 2 所示 (该次测试有五种测试状态, 大量数据无法全部展示, 且数据格式也具有重复性), 其中某 1 台测试机的测试任务如表 3 所示 (该台测试机选择的机器类型中未包含检查点“触控笔”). 从表中可以看出, 测试任务数据分布能很好的满足实际测试的条件和需求. 除此之外, 该算法可以实现向多台测试机分配任务的同时, 保证一个测试计划中周期数的一致性约束.

表 2 测试任务总计

任务总计	USB 设备	键盘	设备管理器	指示灯	托盘	触摸板	触摸屏	指纹	TypeC	相机	麦克风	扬声器	外显	蓝牙	触控笔
	100%	100%	100%	100%	100%	100%	100%	100%	50%	20%	20%	20%	15%	10%	100%
1%-50%	500	500	500	500	500	500	500	500	250	100	100	100	75	50	25
51%-100%	500	500	500	500	500	500	500	500	250	100	100	100	75	50	25
1%-50%	500	500	500	500	500	500	500	500	250	100	100	100	75	50	25
51%-100%	500	500	500	500	500	500	500	500	250	100	100	100	75	50	25

3.2 模型与算法评估

任务自动分配的优点是使测试员从复杂的任务分配过程中解脱出来. 为了综合评估该算法的实用性, 将实现的任务自动分配功能与传统的人工分配做比较, 从一次任务分配时间、人力资源、用户体验、符合测试要求程度四个方面对任务分配进行评估. 表 4 为任

务分配的比较.

通过上述比较可见, 自动任务分配效率远高于人工分配. 该算法的实现充分利用了计算机运算速度快, 处理能力强的特点, 有效的降低了任务分配时间和人力资源开销. 除此之外, 该算法相较于人工而言更能满足测试要求, 可以较真实的模拟测试的随机性和交替

性. 总而言之, 自动化测试任务分配模型和算法的实现 在自动化测试方面具有更高的优越性和实用性.

表3 某1台测试机的测试任务

任务总计	USB 设备	键盘	设备管理器	指示灯	托盘	触摸板	触摸屏	指纹	TypeC	相机	麦克风	扬声器	外显	蓝牙	触控笔
	100%	100%	100%	100%	100%	100%	100%	100%	50%	20%	20%	20%	15%	10%	100%
1%~50%	27	27	27	27	27	27	27	27	13	6	5	6	6	3	0
51%~100%	30	30	30	30	30	30	30	30	16	6	6	5	8	3	0
1%~50%	26	26	26	26	26	26	26	26	13	4	7	4	5	2	0
51%~100%	28	28	28	28	28	28	28	28	15	5	5	5	5	3	0

表4 任务分配的比较

分配方式	一次任务分配时间 (s)	人力资源	用户体验	符合测试要求程度
自动分配	<0.5	1人测试3台	轻松	较符合
人工分配	>8.0	1人测试1台	疲劳	符合

4 总结与展望

笔记本测试自动化任务分配是笔记本自动化测试的一个重要组成部分, 实现此部分功能也是迈向全自动化测试的一个重要阶段. 本文从笔记本电脑自动化测试出发, 设计了用于测试任务自动化分配的模型和算法, 从而进一步提升自动化测试的百分比. 从实际测试的效果可以看出, 该算法既能满足复杂条件下的测试需求, 也能进一步解放测试过程的劳动力, 提高测试效率. 当然, 本文设计的算法刚处于初步实用阶段, 在多客户端并发任务分配效率上还有很大的提升空间, 将在以后的研究中进一步改进和完善.

参考文献

- 陈江勇, 许力, 张辉. Web 表单自动化测试工具的设计与实现. 计算机系统应用, 2011, 20(9): 241-244, 177. [doi: 10.3969/j.issn.1003-3254.2011.09.057]
- 邓青华. 软件自动化测试工具研究. 软件导刊, 2011, 10(1): 57-59.
- 董娜. 自动化测试方法在笔记本代工企业软件测试中的应用. 科协论坛, 2012, (6): 89-90. [doi: 10.3969/j.issn.1007-3973.2012.06.053]
- 李玉, 尉双梅, 汪添生, 等. 基于 QTP 的企业级应用软件自动化测试方法. 计算机系统应用, 2016, 25(6): 219-224. [doi: 10.15888/j.cnki.csa.005200]
- 王兴野. 基于 QTP 功能自动化测试工具及框架研究. 电脑编程技巧与维护, 2018, (4): 52-53, 57. [doi: 10.3969/j.issn.1006-4052.2018.04.019]
- 姜文, 刘立康. 基于 Selenium 的 Web 软件自动化测试. 计算机技术与发展, 2018, 28(9): 47-52, 58. [doi: 10.3969/j.issn.1673-629X.2018.09.011]
- 刘金波, 黄海于. 耦合分布式系统多任务动态调度算法. 计算机技术与发展, 2017, 27(12): 16-19. [doi: 10.3969/j.issn.1673-629X.2017.12.004]
- Sattar H, Bajwa IS, Shafi UF, et al. Automated DD-path testing: A challenging task in software testing. Proceedings of the 9th International Conference on Digital Information Management. Phitsanulok, Thailand, 2014: 230-236.
- 张晓航, 李国良, 冯建华. 大数据群体计算中用户主题感知的任务分配. 计算机研究与发展, 2015, 52(2): 309-317.
- Türküzan M, Yildirim Y, Atakan HC, et al. A modular, extendible, and reusable test configuration for design verification testing of mission computers. Proceedings of 2016 IEEE Autotestcon. Anaheim, CA, USA, 2016: 1-4.
- 干晓鸣. 软件自动化测试的合理应用. 计算机应用与软件, 2010, 27(8): 172-174, 214. [doi: 10.3969/j.issn.1000-386X.2010.08.052]
- Monteiro CBAL, Dias LAV, da Cunha AM. A case study on pairwise testing application. Proceedings of the 11th International Conference on Information Technology: New Generations. Las Vegas, NV, USA, 2014: 639-640.