

基于 API 和 Postscript 的图形自动化验证方法

张磊磊, 李心科

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘要: 当今, 软件测试工程师越来越倾向于自动化测试的研究工作。自动化测试主要由测试用例的设计、执行测试和验证三个模块组成。而验证模块实现上的困难使得自动化研究陷入了举步维艰的境地。特别是现有的自动测试工具都不能对图形对象进行有效的对比验证。因此文中提出一种自动验证的技术, 用于解决屏幕图像、纸质输出图像的验证问题, 并借助 PowerPoint 以实例加以论述。

关键词: 软件测试; 自动验证; API; Postscript

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2008)05-0094-04

Research on Graphical Automated Verification Method Based on API and Postscript

ZHANG Lei-lei, LI Xin-ke

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: Recently, software testers have relied more on automated testing to test software. Automated testing method consists of three modules: test case design, execution and verification. Yet, to accomplish these three phases, people are always in a dilemma due to a lack of a verification function. Nearly all the commercial automated testing tools cannot efficiently compare graphic objects. This research develops a technique that aids automatic verification for a difficult problem: determining the correction of screen and paper output. Methodology to capture and compare the output is presented and a case study using PowerPoint is described.

Key words: software testing; automated verification; API; Postscript

0 引言

当前存在着这样的问题, 大量的图形图像应用于各式各样的桌面软件和网页中, 而自动测试工具却很难捕捉和对比这些图形对象。此外, 有关图像打印测试方面的研究依然很少, 几乎没有现成的自动工具可以处理打印测试。因此, 无论是对于测试者还是开发者来说, 都不可避免地会遇到图形测试带来的种种问题, 他们不得不手工运行测试用例, 密切关注着计算机屏幕或打印机输出的图像。而随着图像技术的发展, 依靠手工运行图形测试用例将变得不现实。

在传统的图形验证工作中, 测试者将图像文件作为二进制信息进行逐位对比验证^[1]。但是实际工作中的种种制约, 如储存空间的约束(图像的规模越来越大)、处理时间的约束(逐位对比非常耗时), 这使得测试者对逐位对比方法的适用性产生了质疑。为了解决

图形测试所面临的这些问题, 提出一种自动验证方法 APIC(API comparison) 和 PSC(postscript comparison), 下面将分别介绍。

1 图形的验证方法

图形验证的关键在于如何对比图形对象。对于屏幕显示的图像来说, 使用较多的是“逐位对比”的验证方法, 而针对打印测试目前却没有切实可行的方法。

1.1 屏幕显示图像的验证

先简要地说明一下逐位对比法, 再提出另一种更行之有效的验证方法 APIC。

1.1.1 逐位对比

大多数商业工具都采用逐位对比的方法, 将图像转化为二进制文件, 对比特位进行逐一对比。这种方法简单清晰, 便于工具的开发。但是它存在一个严重的不足——对于图形的变化过于敏感。例如在 Windows 环境下绘制一幅图像, 以“A”命名保存, 同样的图形再以“B”命名另存; 使用逐位对比工具比较两幅图, 比较程序将会判断图像是不一致的。

收稿日期: 2007-08-27

作者简介: 张磊磊(1982-), 男, 安徽六安人, 硕士研究生, 研究方向为软件工程、软件测试; 李心科, 副教授, 博士, 研究方向为软件工程、神经网络等。

1.1.2 APIC 方法

在 Windows 环境中,操作系统在处理多任务时,避免因争夺资源而产生的冲突,硬件资源是被设置成共享的。因此应用程序是不可以直接访问图形设备的,它是通过系统的 API(应用程序接口)实现对图形设备的访问^[2]。

绘图软件就是使用图形 API 指令渲染屏幕中的图像,利用这点,中断 API 指令的调用,捕捉并存储其中的绘制信息,使用捕捉到的信息进行图形对比,称为 APIC 方法(见图 1)。例如,在 PowerPoint 中画一条直线,就是使用指令 LineTo 将“绘制一条直线”的信息传递到操作系统的。这种情况下,测试工具只需要中断“LineTo”获取其信息。这个方法的优点在于中断 API 指令、获取信息并不会受到系统环境和专用软件的制约。

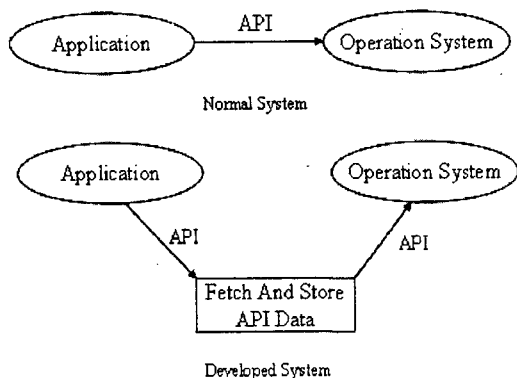


图 1 APIC 模型

1.2 打印图像的验证

一般的,打印在纸质上的图像,自动测试工具是无法获取其打印信息的。虽然图像扫描仪可以通过扫描获取信息,但是那种信息过于模糊。针对打印测试,下面利用 Postscript 指令来获取打印图像的信息。Postscript^[3]语言是由 Adobe 公司开发,并广泛运用于各类桌面排版软件。一个图形对象是可以通过 Postscript 指令生成的。如果应用程序提出打印请求,操作系统将接收来自应用程序的 API 指令,为响应请求,将图像信息传递给打印驱动,最后打印输出(见图 2)。

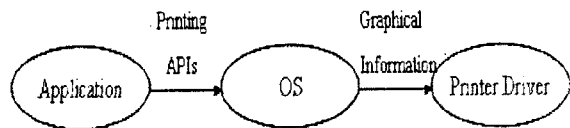


图 2 PSC 模型

从中捕捉图形 Postscript 指令,并用于验证,这种方法就被称为 PSC。例如,在打印纸上绘制一条从坐标(0,0)到(100,100)的直线,则可生成下面的 Postscript 指令:

```

0      0      moveto
100    100    lineto

```

指令 moveto 设置了坐标点(x,y),lineto 表示在两点之间绘制一条直线。

2 APIC 和 PSC 的优点

采用 APIC 和 PSC 方法可以给测试工作带来很多益处,下面介绍几个主要的优点。

2.1 APIC 系统

(1)系统配置:逐位对比验证对变化非常的敏感。很多不可控制的因素常常影响结果。举例来说,一幅完全相同的图像,测试者 A 使用较先进的图形处理系统,它支持 1600 * 1200 的分辨率;测试者 B 使用 1024 * 768 分辨率的图形系统,而这种处理环境的微小差别就可以造成屏幕显示图像的验证失败。但是,使用 APIC 就可以避免这种系统差别所带来的影响。

(2)图形放大和缩小的测试:大多数的绘图软件都支持图形的放大和缩小。如果采用逐位对比的方法测试这种缩放功能,测试者需要保存每一种可能的放大率下的图像。当测试图像放大率从 1% 开始,并以 1% 递增至 100% 时,就需要 100 个测试用例,即必须存储 100 幅图像用于验证。如果采用 APIC,测试者只要保存一幅图像,就可以测试任何放大率下的图像了。

(3)基于模型的测试:采用逐位对比法执行基于模型的测试时,最大问题在于需要手工完成验证工作,效率太低^[4]。而 APIC 就是为实现自动化验证而提出的,摆脱手工工作,极大地提高测试效率。

2.2 PSC 系统

由于有关打印测试的研究很少,事实上,大多数测试人员在对打印的图像进行验证时就是采用视觉上的比较。这样的工作不仅造成时间、人力资源上的浪费,而且验证的准确率也难以保证。在图像复杂的情况下,更是难以通过视觉直观地进行比较。PSC 方法则可以很好地处理打印测试。

3 工具的开发

3.1 APIC 工具

笔者参与开发了一套可以在 PowerPoint 上获取图形 API 接口信息的工具。PowerPoint 在绘制图形时,调用了软件 Office 中的 graph9.exe 组件。为获取信息,需要对 graph9.exe 文件进行修改。将修改的部分文件代码以“gdi23.dll”重新命名保存,使其具有两个功能:传递 API 接口信息和储存信息,而修改前的代码文件“gdi32.dll”也依然留用。

完成修改后,当 PowerPoint 使用 LineTo()和 Poly-

gon() 这样的图形 API 指令时, graph9.exe 就利用 gdi23.dll 获取其中的图形信息。如图 3 所示, API 接口信息先通过 gdi23.dll, 再传递到 gdi32.dll, 最后传至硬件驱动执行打印工作。其中通过 gdi23.dll 捕捉、保存的信息就可用来验证。

下面列出了 gdi23.dll 用于获取 LineTo 指令的部分源代码。

```
KERNEL23_API BOOL WINAPI myLineTo(HDC hdc, int iX, int iY){
typedef BOOL(CALLBACK * LPFN)(HDC, int, int);
HINSTANCE bltH_Dll;
LPFN bltPtrFn_Function;
BOOL ReturnValue;
char ptrChr_DllPath[MAX_STR];
GetSystemDirectory(ptrChr_DllPath, MAX_STR);
//Load gdi32.dll
strcat(ptrChr_DllPath, "\\gdi32.Dll");
bltH_Dll = LoadLibrary(ptrChr_DllPath);
bltPtrFn_Function(LPFN)GetProcAddress(bltH_Dll, "LineTo");
//To store LineTo information into disk
LogFile("LineTo( * hdc, ", iX, iY, LINETO_LOG);
ReturnValue = bltPtrFn_Function(hdc, iX, iY);}
```

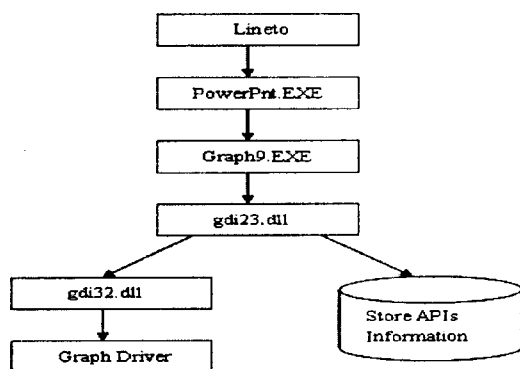


图 3 捕捉/存储信息

此工具主要应用于 PowerPoint, 运行测试用例(待测试的图形)时, 捕捉其中的图形信息, 并保存为测试图形信息。为了与已知的原始图形信息进行对比, 工具将原始图形信息和测试图形信息绑定在一起。这个绑定的文件被编译、连接, 最后通过一个 C++ 比较程序执行验证。

3.2 PSC 工具

类似的, 捕捉 Postscript 指令要比获取 API 指令简单一些。PowerPoint 可以自动地将 Postscript 信息以文件形式保存, 但是不能直接地利用此文件进行比较。因为保存的指令文件包括了许多的冗余信息, 如指令的定义、函数的定义等等, 这些都是不用于验证的。自动工具需要添加过滤功能, 以便选择出与图像渲染有关的指令和数据, 再对有效信息进行验证, 最后显示图

形的不同之处。

4 PowerPoint 的实例

PowerPoint 是目前最流行的图形演示软件, 可以清晰地用于演示 APIC 和 PSC 方法的实际运用和测试效果。

4.1 屏幕显示图形的验证实例

先使用 APIC 方法对简单的图形进行测试, 然后再进行扩展, 对现实中的复杂图像进行测试。

4.1.1 线和圆

线和圆是基本的图形物体, 常用来构造更复杂的图形图像。使用 APIC 对图 4 绘制的简单图形进行测试演示, 如下:

测试用例 1: 比较两幅简单的图形。

- (1) 运行 PowerPoint;
- (2) 绘制一个矩形和圆(见图 4);
- (3) 作为原始图形 Original 保存;
- (4) 改变矩形的长度;
- (5) 作为目标图形 Modified 保存。

预期结果: 显示两幅图形的区别。

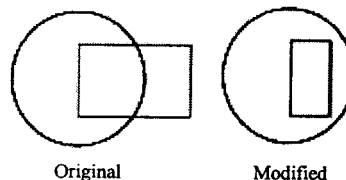


图 4 简单的图形

APIC 记录的原始图形 Original 的信息:

Ellipse(hdc, 87, 51, 204, 168);
Rectangle(hdc, 147, 80, **238**, 146);

APIC 记录的目标图形 Modified 的信息:

Ellipse(hdc, 87, 51, 204, 168);
Rectangle(hdc, 147, 80, **185**, 146); ← 黑体部分可以清楚地看出区别。

4.1.2 复杂的图形

测试对象是 PowerPoint 的两幅柱状图(见图 5), 图中的物体主要由几组多边形构成, 为了方便分析和论述, 工具仅测试多边形的有关信息。图中仅改变第一个柱状体的高度, 当两幅图形进行对比时, 测试结果

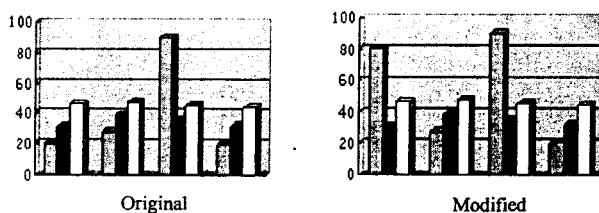


图 5 柱状图

应该显示出第一个柱状体的区别。

测试用例如下:

测试用例 2: 比较两幅图形的区别。

- (1) 运行 PowerPoint;
- (2) 插入图形对象;
- (3) 保存为原始图形(Original);
- (4) 改变第一个柱状体的大小;
- (5) 另存为目标图形(Modified)。

预期结果: 显示第一个柱状体的不同。

测试分析:

- * 发现 42 个多边形;
- * 2 个多边形具有不同的坐标点;
- * 9 条直线的长度被改变。

测试结果: Pass。

比较柱状图, 程序会发现有 2 个多边形发生改变, 包括 9 条直线。由此可知自动工具能够和预期的一样发现图形的区别。值得注意的是, 逐位对比法几乎很难发现此类图形的不同和造成其差别的原因, 而 APIC 可以显示出图形的差别源于接口指令的不同。因此, 在调试时, APIC 能给出清晰的错误信息, 可以更快辅助调试。

4.2 打印图像的验证实例

与 APIC 测试演示类似, 也分别用上文中两个测试用例对打印的图像进行 PSC 方法验证。

线和圆——测试用例 1 被执行, 对于原始图形 Original 生成如下的 Postscript 指令。

```
5296 1762 Movoto
3446 1762 Lineto
3446 3112 Lineto
5296 3112 Lineto
```

同样的, 对目标图形 Modified 也生成 Postscript 指令, 如下:

```
4196 1762 Movoto
3446 1762 Lineto
3446 3112 Lineto
4196 3112 Lineto
```

区别。

复杂图形——测试用例 2 应用在 PSC 方法上。测试分析如下:

- * 发现 265 条直线;
- * 8 条线段的端点值发生改变。

测试结果表明两幅柱状图是不同的, 差别在于线段的长度发生了改变。

5 结束语

简述了传统的逐位对比法, 再详细探讨了 APIC 和 PSC 的方法思想, 并与前者进行优劣比较; 最后通过实例论述 APIC 和 PSC 的实际运用, 从而说明了新方法在图形自动化验证方面取得的进步。然而任何技术方法都会面临工具实现的难易程度、系统环境的适应性和未来技术的挑战^[5], 而在这些方面, APIC 和 PSC 却表现出极高的适应性和扩展能力。

(1) 实现的难易——APIC 方法思想清晰、易理解, 关键在于捕捉、储存 API 接口的信息。目前, 很多商业软件工具都提供捕捉接口信息的功能。只需要根据具体的测试环境, 对它们稍加改动, 增添合适的比较程序, 即可实现 APIC 验证方法。PSC 思想源于 Postscript 语言, 很多的打印设备都支持 Postscript, 从打印设备本身即可获得所需的 Postscript 指令文件, 用于验证。

(2) 适应性——为了使用 PowerPoint 直观的演示图形验证过程, 选用了 Windows 操作系统。由于没有使用 Windows 的任何源代码, 因此要在 Windows 系统中开发出相应的代码提供支持。而在 Linux 和 Unix 系统上则可以获得直接的支持, 不需要编写额外的源代码。

(3) 新技术的扩展——目前, 诸如基于模型的测试、GUI 容错技术和测试用例的自动生成等等先进的测试技术得到了广泛的研究^[6]。但是, 由于缺少验证功能的支持, 这些新技术难于直接应用于工具的开发。APIC 和 PSC 可以方便地和这些新技术相融合, 提供验证支持, 为图形自动化测试的发展注入新的动力。

参考文献:

- [1] Fewster M, Graham D. Software Test Automation[M]. N. Y.: Addison Wesley, 1999.
- [2] Takahashi J. An Automated Oracle for Verifying GUI Objects[J]. ACM Soft. Eng. Note, 2001, 26(4): 83-88.
- [3] Adobe Systems, Inc. PostScript language Reference[M]. Third Edition. [s. l.]: Addison - Wesley Publishing Company, 1999.
- [4] Takahashi J, Kakuda Y. Extended Model - Based Testing toward High Code Coverage Rate[C]//7th European Conf. on Soft. Quality. Helsinki: [s. n.], 2002.
- [5] Hearn D. 计算机图形学[M]. 第 3 版. 北京: 电子工业出版社, 2005.
- [6] Li Kanglin, Wu Mengqi. 图形用户界面测试自动化[M]. 王铁昆译. 北京: 电子工业出版社, 2005.