嵌入式移动终端软件自动化测试技术研究

张洪春
(湖北职业技术学院信息技术学院 湖北 432000)

[摘 要] 在深入分析嵌入式软件测试特点和测试现状、总结对比移动终端软件测试技术和方法后,提出了移动终端软件的自动化测试原理和测试模型。以此为基础,通过手机终端软件测试实例研究,构建出手机软件自动化测试体系结构,讨论了手机软件自动化测试的关键技术,展望了未来该领域新技术发展趋势,为进一步研究移动终端软件自动化测试机制及运行机理提供参考和理论依据。

[关键词] 软件测试 自动化测试 嵌入式软件 移动终端
中图分类号:TP311 文献标识码:A 文章编号:1008-1739(2012)19-68-4

Research on Automatic Test Technology of Embedded Software in Mobile Terminal

ZHANG Hong-chun
(School of Information Technology, Hubei Polytechnic Institute, Wuhan Hubei, 432000, China)

Abstract: The automatic test principles and test models of mobile terminal software are proposed based on analysis of test characteristics and test status of embedded software, and summary and comparison of mobile terminal software test techniques and methods. Based on the test principles and test models, the handset software automatic test architecture is established through handset terminal software test example research, the key technique of handset software automatic test is discussed in details, and future development trends of new techniques in the field are expected. The research provides reference and theoretical basis for further studying mobile terminal software automatic test mechanism and operation mechanisms.

Key words: software test; automation test; embedded software; mobile terminal

1 引言

现在有很多移动终端软件测试工具,这些工具要么覆盖测试范围很窄,要么实现困难或成本昂贵。很多测试方法也受特定的软件结构和程序语言的限制,很多的建模技术和方法也只能只针对整个测试过程的某个阶段,而并不能指导整个测试流程,贯穿软件测试的始终[1]。

手机是典型的移动终端产品。近年来,手机终端发展迅猛,随着用户对其功能、性能、稳定性的要求越来越高,其测试的复杂繁琐性也愈来愈突出,手机软件测试面临极大的挑战,如下:① 手机行业市场响应在加快,出货周期迅速,生命周期缩短,市场竞争压力大,要快速使产品稳定必须要有极多的人力和较长的时间来执行测试;② 传统的手工测试难以满足手机软件系统快速稳定的要求。手机软件测试比常规的桌面软件手工测试疲倦更为严重,极大的影响了测试效果和效率;③ 手机软件系统复杂,手机平台及应用的多样化,使得很多项测试是无法用手工测试或者手工测试根本就不可靠、不精准。因为手工测试无法保证测试的一致性、可重复性等。

目前,嵌入式软件测试方式可分为纯软件测试方式、软硬件结合测试方式和硬件仿真测试方式3大类[3],并主要采用有以下几种测试技术方法:① 目标环境测试:直接对真实终端进行测试,这是适宜于黑盒的测试。该方法主要是基于手工操

定稿日期:2012-09-12

作，很难实施自动化测试，需要消耗较多的经费和时间；② 宿主环境测试：将嵌入式软件与终端完全剥离，利用模拟环境进行的测试。该方建立模拟环境难度较大，并且模拟环境很难真实地反映软件实际运行环境；③ 基于脚本驱动的非侵入式自动化测试方法：运用交叉测试方式实时地将测试信息通过网线/串口传到宿主机上，以克服移动终端的实时性、内存有限、环境开发工具昂贵等。目标环境测试适宜于黑盒测试，嵌入式软件的白盒测试一般要在宿主环境下完成[4]。另外，除了集成测试是在宿主/目标交叉环境或目标环境下执行外，单元测试、确认测试和系统测试均需在目标环境下进行[5]。

2 手机软件自动化测试体系结构

2.1 手机软件自动化测试的问题

手机测试主要强调时间、次数、并发、极限和负载测试，因此手机自动化测试至少要能解决以下的问题。① 压力和容量测试：反复进行一套或长时间进行一个动作操作或进行快速操作，比如反复切换歌曲播放、联网、System Memory 和 User Memory 测试等；② 极限临界测试：事件发生的开始和结束瞬间、内存处于满和空的状态应是关注的重点，比如进行一些极限条件的构造（创建多个列表）及输入字符个数等；③ 兼容及中断测试：人为中断、新任务中断、意外中断以及多任务测试，比如在播放或下载歌曲时候来电话或者信息；④ 基本功能回归测试：能大大地节约时间和人力成本，提高效率。

2.2 手机软件自动化测试思路

手机软件的自动化测试技术才刚刚起步，研究的首要问题是要分析手机系统的自身特性及其面临的挑战。现结合手机软件系统实现机制，提出自动化测试思路如下。① 与常规的桌面软件自动化测试原理相同，软件关键图标与文字信息抓取识别，以及动作录制回放，实现快速回归机制；② 由于桌面自动化测试工具捕获或录制机制已经成熟，对于手机软件来说需要手机本身的软件系统（OS 层）提供底层 API 的支持，比如底层图片处理方面的 API 等；③ 利用手机与 PC 通讯设备（串口、USB 等）连接的桥梁，在手机端驻留一个测试代理。其思路是将更多的测试转移到宿主环境中进行，使得绝大部分对手机的操作转移到 PC 软件的测试操作，而把宿主环境测试无法实现的复杂和独特功能放在目标环境中测试。

2.3 手机软件自动化测试体系结构

2.3.1 移动终端软件测试原理和模型

基于上述脚本驱动的自动化交叉测试方法，在参考有关文献的基础上，提出了移动终端软件自动化测试原理模型，如图 1 所示。

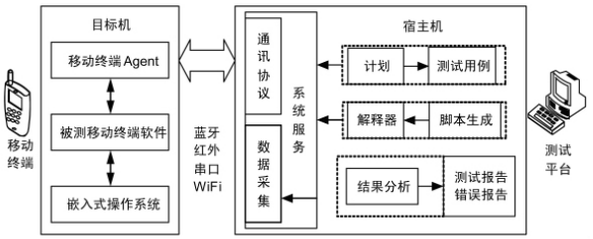

图 1 移动终端交叉测试原理示意图

基本原理：测试工具运行在软硬件资源丰富宿主机上，测试所需要的信息在目标机上产生，并能传输到宿主机上，由测试工具接收。被测软件运行在软硬件资源相对匮乏的目标机上，目标机上的 agent 是具有推理判断能力和智能行为的软件实体，可在特定环境下实现过程持续和对象自治运行，并能将监听被测软件产生的响应结果信息回传至宿主机。测试用例必须写成脚本的形式，通过对应的脚本解释器转换成相应的命令流，对平台进行自动化操作，并能间接对移动终端进行自动的功能和结果对比。

(1) 测试平台组成

在图 1 中，测试平台由 3 部分组成：

① 脚本编辑部分：利用测试平台制订测试计划，设计测试用例，自动或手工生成测试脚本；

② 脚本执行部分：测试脚本由脚本解释器解释执行，将测试指令与测试数据传输到目标机上，通过脚本的自动运行来对移动终端发送各种命令以及获取移动终端的对应屏幕信息；

③ 测试报告部分：测试平台可以自动采集结果数据，分析结果信息，产生测试报告。

(2)测试要求

在 PC 机上实现移动设备的仿真环境，在对 PC 仿真界面操作的同时，实时地发送相关的命令至移动终端，使移动终端产生各种对应的按键动作和屏幕信息，从而自动控制移动终端的键盘、旋钮和触摸屏，以模拟测试工程师的双手操作，能自动抓取 LCD 显示内存中的位图文件，使用相关技术模拟测试工程师的双眼进行内容识别和逻辑判断。

3 手机终端自动化测试体系结构

手机终端自动化测试体系结构图 2 所示。Agent TestAgent 为嵌入在手机软件系统中的一个测试代理模块，通过串口或蓝牙设备与 PC 端中的 TestTool 建立通讯，解决 PC 端与手机

端交互处理及互联消息通讯问题，这是区别于其他桌面软件自动化测试的关键点。

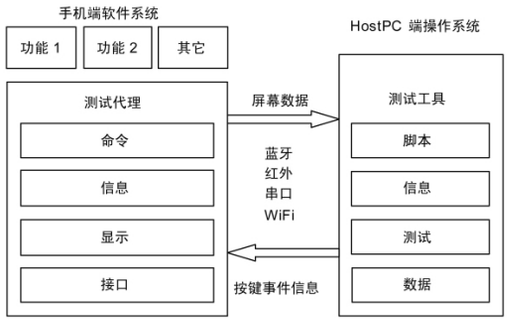

图 2 手机自动化测试体系结构示意图

TestAgent 具备的主要功能如下：① 定义各种与 PC 对应的取屏的指令；② 识别和接收 TestTool 发送的指令和消息，向手机端软件系统分发消息及任务；③ 根据指令要求收集屏幕的信息、压缩屏幕信息并对压缩后的屏幕信息进行编码；④ 监控手机端软件运行情况并根据相应的约束反馈给 PC 端的 TestTool；⑤ 被测软件的功能（接口）封装及消息响应。

TestTool 是自动化测试工具，在 PC 端用于测试控制及测试操作实体，与 TestAgent 对应，该工具与常规的自动化测试软件是一样的。TestTool 具备的主要功能如下：① 向手机端 TestAgent 发送可识别的消息及任务；② 接收来自手机端 TestAgent 的反馈结果；③ 对来自手机端 TestAgent 的反馈进行测试业务的处理；④ 将测试业务的处理结果呈现给测试人员。

4 测试服务

测试服务主要有以下 2 种。

(1) 主动式测试

TestTool 主动式测试是通过运行脚本向 TestAgent 发送消息及任务，TestAgent 再向被测软件分发消息及任务，并将结果原路返回给 TestTool，TestTool 再通过数据处理分析得出测试结果。主动测试要根据测试需求比如（压力、性能、极限）在 TestTool 中编写测试脚本，然后控制手机端软件进行测试，或者构造一些手工很难实现的测试场景。主动式测试对脚本的依赖和目标机上的信息收集与回传是问题的关键。测试关键点：发送和分发消息、接收及处理反馈结果（结果判断）。

(2) 回归式测试

基本功能的回归测试最为简单的方法就是录制和回放机制。通过运行录制的测试脚本完成先前的操作顺序、步骤和输入数据等，以再次测试被测软件，达到回归测试的目的。与主动式测试方式不同的是回归式测试是事先要录制脚本，通过录制脚本来代替人工编写脚本。

录制：在执行手工测试时将手工测试的任何操作及返回结果（预期正确的结果）通过 TestAgent 在 TestTool 中保存下来，并进行分析处理形成一个可执行的脚本。录制的关键点：按键或触屏消息、坐标、响应结果（GUI 界面）。

回放：与录制相对应，运行录制时产生的、在 TestTool 中保存下来脚本。回放关键点：发送和分发消息、接收及处理反馈结果（结果判断）。

5 手机软件自动化测试关键技术

5.1 消息传送机制

消息传送机制：① 可使用 ATC 进行通信。目前手机厂商提供了常用的 AT Command，利用手机 Modem 中提供的 AT Command 并通过串口向手机端建立命令消息通讯，能基本满足普通的自动化测试需求；② 可使用用户自定义 ATC 功能。当标准的 AT Command 不能满足自动化测试需求时，用手机自己专门的命令可支持远程终端操控手机，比如键盘控制等，以实现自动化测试中所需要的消息通讯。例如在 MTK 平台上实现自定义 AT Command 的关键样例代码如下。

```
view plaincopy to clipboardprint?
#define CUSTOM_SYMBOL  ´^´   // ´+´ and ´/´ and ´ \ ´is NOT allowed
#define MAX_UART_LEN  128
#define PLAY "play"
#define STOP "stop"
kal_uint8 custom_get_atcmd_symbol(void);
void custom_command_hdlr(char *full_cmd_string);
extern void rmmi_write_to_uart (kal_uint8 * buffer, kal_uint16 length, kal_bool stuff);
……
void custom_command_hdlr(char *full_cmd_string)
{ char buffer[MAX_UART_LEN];
char cmd_name[15];
kal_uint8 index = 3;     // we start parsing index after the CUSTOM_SYMBOL
kal_uint8 tmp_idx = 0;
while  ((full_cmd_string [index] ! = ´=´ ) && //might be TEST command or EXE command
  (full_cmd_string [index] ! = ´?´ ) && // might be READ command
 (full_cmd_string[index] ! = 13 ))  //carriage return
{ cmd_name[tmp_idx] = full_cmd_string[index] ;
```

```
tmp_idx ++;   index ++;  }
cmd_name[tmp_idx] = '\0';
if (strcmp(cmd_name, PLAY) == 0)
{ //Todo 实现消息分发或功能调用
sprintf(buffer, "Hello Play");
printf("%s\n", "Hello Play");
rmmi_write_to_uart ((kal_uint8*)buffer,  (kal_uint16)strlen (buffer),
KAL_TRUE); }
else if (strcmp(cmd_name, STOP) == 0)
{ //Todo 实现消息分发或功能调用
sprintf(buffer, "Hello Stop");
printf("%s\n", "Hello Stop");
rmmi_write_to_uart ((kal_uint8*)buffer,  (kal_uint16)strlen (buffer),
KAL_TRUE); }
else  {  sprintf(buffer, "ERROR");
printf("%s\n", "ERROR");
rmmi_write_to_uart ((kal_uint8*)buffer,  (kal_uint16)strlen (buffer),
KAL_TRUE); }
return;  }
kal_uint8 custom_get_atcmd_symbol(void)
{ return (CUSTOM_SYMBOL); }
```

5.2 图像识别

图像识别主要通过抓取 LCD 屏幕显示图像进行智能识别来模拟测试工程师的双眼辨识文字或图像信息，以此判断测试结果。主要涉及图像的获取和对比分析，智能识别是一个比较专业的研究领域，目前可以考虑通过第三方工具来实现，比如借助目前已经成熟的测试工具 QTP 等，通过截屏，再调用第三方软件来和标准图像比较。

5.3 录制回放

保存手机的关键屏幕内容以及屏幕之间的路径信息，所有跟手机有关的细节如手机主题、屏幕尺寸、语言以及其它主观信息都被自动封装。无需编写任何代码就可完成测试用例的开发、调试和运行，且测试用例无需改动或者稍微改动，即可移植应用到其他类型的手机上。录制的信息及相应的实现方式如下：

① 按键消息：由 TestAgent 捕获该消息并同步给 PC 端的 TestTool；

② 笔点消息：由 TestAgent 捕获该消息并同步给 PC 端的 TestTool；

③ 坐标：由 TestAgent 捕获该坐标信息并同步给 PC 端的 TestTool；

④ 响应结果：通过图像抓取接口抓取图像并同步给 PC 端的 TestTool；

⑤ 时钟同步：操作步骤的时间点、操作的先后顺序、输出结果响应时间。

⑥ 录制脚本组装：TestTool 将所有的录制信息进行处理并组装成一套可运行的测试脚本，要求运行该脚本后能够与录制时的操作完全一样，并能将回放时的实际结果与预期结果进行比较从而得出执行结果。

⑦ 回放：主要是运行组装好的测试脚本，将回放时的实际结果与预期结果进行比较从而得出执行结果，关键点还是图像识别。

6 测试环境与管理

脚本语言和环境：可以选用 VBScript、Per、Python、Java 等作为脚本语言，用户根据脚本语言规范编写测试用例。比如 Nokia 的一些工具就是使用 python 脚本。测试数据需要建立一个数据仓库管理数据，比如录制时产生的消息、坐标、GUI 图像信息等。比如测试用例库、手机函数库等。测试过程管理与结果管理方面主要涉及测试规划、测试用例、测试报告、缺陷报告、改进建议的呈现及保存。

7 结束语

文章探索了移动终端软件自动化测试原理，构建了手机软件测试体系结构，实现了部分关键技术，以实现在手机软件测试中，用自动化测试工具替代传统的手工测试，提高测试效率，降低测试成本。因此这项研究具有一定的现实意义和应用前景，并为最终实现有效而全面的自动化测试解决方案奠定了理论基础。进一步研究工作主要包括完善智能移动终端库，实现自动化、可视化的测试脚本代码自动生成，以及支持分布式对象扩展等。

参考文献

[1] 陆永忠,宋骏礼,谷希谦.基于行为的软件测试过程模型及研究[J].计算机应用,2007,27(5):1238- 1239.
[2] 宋文,于林宇,刘军.通用嵌入式软件测试环境在武器装备软件测试的应用[J].测控技术,2005,24(10):66- 67.
[3] 滕克难. 基于多 Agent 舰空导弹协同反导作战体系结构研究[J].火力与指挥控制,2008,33(3):117- 119.
[4] 王荣. 嵌入式软件测试方法[J].航空兵器,2003,(5):12—14.
[5] 朱燕芬,佘检求,王健.分析仪器中嵌入式软件黑盒测试的研究[J].化工自动化及仪表,2008,35(1):57- 60.