Web前端 / 数据库 / 综合技术　　（日）技术评论社 编

For All Web Application Developers　ウェブDBプレス

WEB+DB PRESS

WEB+DB PRESS 中文版 02

智能手机测试最前沿

应用、浏览器、服务器端完全自动化

Amazon Web Services 最新技巧

EC2、VPC、RDS、CloudFormation

Sass/Compass实战

用简洁的代码描述现代化的CSS

使用Grunt实现
前端开发的自动化

通过Doctrine Annotations实现的
声明式编程

用程序性能分析来分析
性能问题

移动设备环境下的调试技术

模拟移动功能以及iOS/Android中
远程调试功能的使用方法

中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

TURING

WEB+DB PRESS

Web前端 / 数据库 / 综合技术

（日）技术评论社 编

For All Web Application Developers ウェブDBプレス

WEB+DB PRESS 中文版 02

人民邮电出版社
北京

图书在版编目（CIP）数据

WEB+DB PRESS中文版. 2 / 日本技术评论社编 ; 匿名译. -- 北京 : 人民邮电出版社, 2015.6
ISBN 978-7-115-28305-4

Ⅰ. ①W… Ⅱ. ①日… ②匿… Ⅲ. ①网页制作工具－程序设计 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2015)第107715号

内 容 提 要

*WEB+DB PRESS*是日本主流的计算机技术杂志，旨在帮助程序员更实时、深入地了解前沿技术，扩大视野，提升技能。内容侧重于Web开发的相关技术。

本期的主题分为3个特辑：智能手机测试最前沿、Amazon Web Services最新技巧和Sass/Compass实战。特辑1从客户端到服务器端讲解智能手机应用程序自动化测试的相关内容。特辑2介绍了主流的云计算平台Amazon Web Services的最新功能，并对其主要服务的一些使用方法进行解说。特辑3则介绍Sass以及构建在其基础之上、能够扩展其功能的Compass框架，内容涵盖了基础知识以及在生产系统中的实战技巧。

本书适合各行业Web前端和数据库开发者阅读。

◆ 编　　[日] 技术评论社
责任编辑　乐　馨
执行编辑　高宇涵
责任印制　杨林杰
装帧设计　王　玲
◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷
◆ 开本：787×1092　1/16
印张：11
字数：296千字　　2015年6月第1版
印数：1-3 500册　　2015年6月北京第1次印刷
著作权合同登记号　图字：01-2015-1435号

定价：20.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

目 录

CONTENTS

分栏目录
CONTENTS BY TOPICS

特辑1　智能手机测试最前沿　应用、浏览器、服务器端完全自动化！

特辑2　Amazon Web Services最新技巧　EC2、VPC、RDS、CloudFormation

特辑3　Sass/Compass实战　用简洁的代码描述现代化的CSS

专　栏

连　载

专家视点

图灵访谈

UI/UX未来志向

预测未来之走向，知晓当下之所需

第4回

明治大学　综合数理学部　先进媒体科学专业专职讲师
渡边惠太（WATANABE Keita）　译/苏祎
URL http://persistent.org/　mail watanabe@gmail.com　Twitter @100kw

众多亮点的游戏设计世界

此次我们的主题是从电子游戏中学习UI/UX。近年来，电子游戏机在硬件接口上发生了很大的变化，例如任天堂3DS的双屏画面、Wii的手柄以及XBOX的Kinect等。这些变化值得我们去关注，但实际上，从20世纪80年代的所谓红白机（FC）时代开始，游戏领域里就已经对UI/UX有着非常深入的思考了。

电子游戏是以娱乐为目的的，虽然乍一看会觉得展现的都是实质内容，并不用涉及UI/UX，但实际上没有哪一个领域能比它更适合UI/UX的主题了。若要提及和其他应用开发的不同，那就是目的是在娱乐，还是在解决问题上了。

游戏是非常注重体验的，所以一旦玩家觉得玩法太复杂，游戏就会被嫌弃。因此，和应用程序一样，游戏也需要花费非常大的精力来提升UI和用户体验的品质。在此次的文章中，我们将考察为了重视用户的体验，游戏都在UI上下了哪些工夫。

剧情声手法

在电影中，有一种惯用的手法称为剧情声（Diegetic）。一般在荒野打斗的场景中，都会播放具有紧张感的音乐，但主角们并不是一边听着这种音乐，一边在打斗的。在电影的世界里，可能会有风声，也可能会有空气的声音，但是具有紧张感的音乐则是为了电影这一类媒体的演出效果而在后期加上去的。说起来可能会觉得太夸张了，但其实电影里的音乐大部分都是“演出效果”。

而剧情声手法则与此相反。也就是说，是电影里的主角们也会实际听到的音乐效果，并且音源是处于明确显示的状态。例如，如果是拍摄有人在酒吧等场所演奏钢琴的话，这个场所中响起的音乐就会如实地在电影中呈现出来。剧情声和演出效果音的区别在于是否会在画面中出现成为音源的那部分。也就是说，是充满了真实体验的表现手法。

游戏中的剧情声手法

这个剧情声手法其实在电子游戏的UI中也有使用。UI上的剧情声是怎么一回事呢？一起来看一下具体的例子吧。

格斗游戏和射击游戏中，画面上会显示代表自己剩余能量的血槽。这是给游戏玩家的提示，与游戏的真实场景并无关联。从这点看来，与前面的音乐类似，这也不是场景的真实内容，而只是一种演出效果，并不属于剧情声手法。

但是在最近的游戏里，也开始不在画面上向玩家显示上述的血槽UI了，或者说在UI方面也开始使用剧情声手法了。那么是如何显示这些信息的呢？以能量槽为例的话，就是让能量归属于游戏角色。例如在《死亡空间3》（Dead Space3）[①]中，游戏角色被设定为穿着在背后显示其生命状态的制服，用来向玩家显示现在的能量（图1）。这样，能量槽就成为了这个游戏世界中真实存在的东西。同时，去除了向玩家提示的能量槽，画面结构也更加简洁清爽。从结果来看，可以改善是通过画面来玩游戏的感觉，具有提高游戏世界沉浸感的效果。

特别是在最近的大制作游戏中，CG也采用了如同电影般极

① Electronic Arts（2013年），Windows / PlayStation 3 / Xbox 360。

度精美的画面，如果在UI画面上显示文字的话会对场景的美观有一定损害，因此尽量不表示这类提示信息才是大势所趋。

另外，在游戏中还有拿着自己拥有的武器、替换武器、查看地图等设置方面的操作。在世界观设定时，将角色装备的护目镜设定为AR（虚拟现实）的透视型显示器，将这些操作的设置画面显示在该装备上，就会减弱这些画面是特意为了玩家显示出来的感觉。

教　学

游戏虽然是娱乐活动，但是最近有些游戏很复杂，用户需要学习“怎么玩”“有哪些功能”。但是，又很难向玩家说出“请看完说明书后再来玩”这样的话。

因此，复杂的游戏经常会在初始阶段设置很多教学（Instruction），让玩家自然而然地进行学习。而有些教学做得几乎让人无法察觉这是教学，能够自然地让玩家进行练习。

例如在《超级马里奥》[2]中，第一关有两个阶梯状的由砖块组成的小山，每个小山的中心是悬崖。在第一个悬崖掉落的话由于有地面不会有危险。但是第二个小山的悬崖下没有地面，掉下去的话就会少一条命。

也就是说，第一个小山是用来练习的。但是多数玩家会认为关卡就是这么设计的，自然地就学会了跳跃的技巧。

类似这样的自然教学形式在游戏关卡设计中也很重要，很多游戏都设计为通过不断地提高难度和学习来推动游戏。

在应用程序中的应用

在UI的设计上，这样的教学也相当重要。不特意为了操作单写一份说明书，而是让用户在使用的过程中不知不觉地掌握应用程序和服务的功能。一些Web服务还导入了奖励机制，也就是促使用户进行实际操作，当所有操作都完成后，作为奖励用户能获得积分，或者增加可以使用的容量，或者可以使用其他一些高级功能。这也是通过让用户尝试所有的功能，让他们对产品的价值能有更好的理解。

游戏与UI/UX

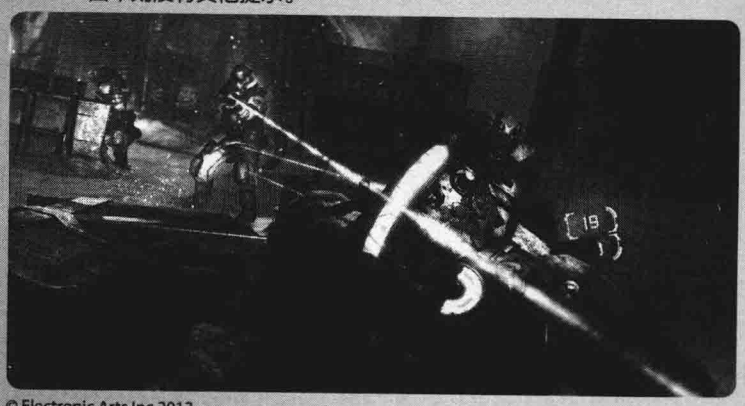
刚才提过，游戏的核心是体验。为了UX，我们要绞尽脑汁将UI提升到极致。

经常会有人认为UX就是通过调查，进行人物角色和剧情的设定，从而发掘新的价值，带来体验。但是，即便发掘了新的价值，在落实到画面设计的时候，很容易就陷入某个按钮、某个菜单、某些文字放在哪里这样非常表面的配置上的改善，多数情况下只是在品牌建立的层面上改善了体验设计。

当然，也不是说“新的体验＝奇特的互动”才是最佳的解决方案，但是操作中的舒适感是在互动中产生的。而这点我们在游戏中可以学到很多。在进行UX设计的时候，流程方面的宏观角度虽然也很重要，但作为人类个体的知觉和身体会如何感知也是非常重要的。体验并不会发生在会议室，而是产生在使用它的用户身上。

在撰写本文章时，笔者与游戏设计研究者筑濑洋平先生做了一定的沟通交流。在此感谢他对本文的帮助。

▼图1 《死亡空间3》的剧情声手法在背脊上有代表能量槽的部分，而在画面中则没有其他提示。



② 任天堂（1985年），Family computer（FC）。

特辑 1

智能手机测试最前沿

应用、浏览器、服务器端完全自动化!!!

为智能手机开发的应用程序，不仅要对应Android、iOS等多种操作系统，提供服务的方式也分为本机应用程序和Web应用程序等。此外，应用在多数情况下还会与服务器端协同工作。像这样在开发时所涉及的目标会有很多，因此在测试的时候要考虑的事情也更多更复杂。基于这些情况，自动化测试势在必行。本特辑将从客户端到服务器端彻底讲解智能手机应用程序自动化测试的相关内容。

第1章 智能手机测试的基本知识

首先要建立测试策略

中川胜树　NAKAGAWA Masaki
DeNA股份有限公司
GitHub masaki
Twitter @ikasam_a
译/刘卓

特辑简介

近些年，智能手机市场迅速发展，开发智能手机的本机应用程序和Web应用程序的需求也随之不断增加（在下文中，二者统称为智能手机应用程序）。

在针对智能手机开发应用程序的时候，由于现在市面上有Android、iOS等多种操作系统，因此要为每个操作系统都开发不同的版本。此外，由于应用程序类型也分为本机应用程序和Web应用程序等，在开发时也要根据需求来选择开发的种类。

如果开发时针对的是多操作系统，在测试时也必须将多操作系统的问题考虑在内。更要针对不同的应用程序类型，变换相应的测试方法和框架。总之，需要考虑的事情有很多，而且很复杂。

此外，虽然有些智能手机的应用程序是在本机中单独运行的，但是在大多数的设计中本机应用都会和服务器端协作，此时当然就要考虑到服务器端的测试了，也需要考虑在这种情况下（应用程序需要客户端和服务器端协作时）系统级别的测试。

一旦应用程序、系统的结构变得更加复杂，测试的对象和数量也就随之增加了，因此测试的自动化已经成为必然的趋势。

在本特辑中，我们将围绕智能手机应用程序开发时进行的测试，解说应用程序客户端和服务器端上的自动化测试。

本特辑的组成

本特辑的组成部分如下所示。

在本章中，我们会将智能手机应用程序和测试分类，并针对应用程序的种类大体说明其测试的相关内容。

第2章将讲解本机应用程序UI（User Interface）的自动化测试方法。

第3章将重点说明在开发智能手机的Web应用程序时，使用浏览器实现的自动化测试。

第4章将针对智能手机Web应用程序开发时必不可少的JavaScript，讲解如何实现自动化单元测试和集成测试。

第5章将针对与智能手机应用程序相互协作的服务器端系统，通过列举Web API实例来讲解自动化测试的方法。

最后的第6章将针对智能手机应用程序的构建和发布，通过实例讲解实现自动化的方法。

智能手机应用程序的分类

智能手机应用程序的形式多种多样，既有与设备框架紧密结合的本机应用程序，也有在HTML5的基础上制作的Web应用程序，还有两者结合使用的混合型应用程序。

下面我们来介绍它们各自的特征。

本机应用程序

本机应用程序是指从如App Store和Google Play等应用商店中下载下来，并安装在iPhone、Android等智能手机，或是iPad等平板终端上使用的应用程序。

因为本机应用程序已经提前安装在设备上了，所以可以根据设备不同的状态控制是否要发生通信等动作，它的启动等所有操作也往往比Web应用程序更快。另外，由于可以根据应用的需求定制UI，所以应用程序的可操作性也提高了。

并且，它也可直接使用设备内置的相机、麦克风、传感器等固有的功能，以及电话薄等设备内置的数据，这点可以说是个很大的优势。

◉本机应用程序的开发

在开发本机应用程序的时候，要使用各目标设备中的操作系统所提供的框架。因此，在开发iOS应用时要使用Objective-C，在开发Android时要使用Java。使用设备自带的框架，就可以集成前面说到的相机、麦克风或是电话簿等设备中固有的功能或数据来进行开发。

在开发上最大的问题是需要根据目标设备中的操作系统使用不同的开发语言，整体的开发成本很高。因此，如果要开发iOS和Android都能使用的应用程序，就必须要用Objective-C和Java分别制作两个不同的应用程序。

为了解决这样的问题，出现了例如Titanium Mobile[①]这样的框架，可以用同一份代码生成多种操作系统的本机应用程序。

Web应用程序

智能手机的Web应用程序与传统的Web应用程序基本一样，是一个运行在浏览器上的应用程序。由于使用了浏览器，就不必像本机应用程序一样安装于各种不同的终端，只要访问特定的URL就立即可以使用了。智能手机的Web应用程序与传统的Web应用程序之间的区别在于要针对不同的智能手机固化屏幕大小，以及需要考虑页面布局以适应小尺寸屏幕。

对于应用程序的发布，由于应用程序的代码并没有在终端上，而是在服务器上，所以更新或发布都很简单，即使没有通过应用商店的审核也可以立刻发布。另外，只要是安装了浏览器的终端都可以使用Web应用程序，因此也不必为了对应不同的设备而制作多个应用程序。

与本机应用程序相反，Web应用程序由于是运行在浏览器上的，所以可能会限制使用相机等设备中的固有功能。

◉Web应用程序的开发

智能手机的Web应用程序大多使用HTML5、CSS3、JavaScript开发。我想对于那些开发过传统Web应用程序的工程师来说，也可以在一定程度上沿用他们以往的经验和既有的代码来开发智能手机的Web应用程序，不会出现什么大的问题。

由于使用了如CSS3和HTML5等Web开发的先进技术，相比传统的Web应用程序，智能手机Web应用程序的表现力也更加丰富。另外，以jQuery Mobile[②]为代表的专门针对智能设备的程序库，也使得基于Web的应用程序可以具有与本机应用程序相似的操作性。

在页面的外观和布局方面，由于智能手机与PC上的浏览器不同，可以通过PC浏览器的设置或扩展功能将浏览器的用户代理（User Agent）设置为智能手机，再通过浏览器上的iOS和Android模拟器实施测试。

在开发环境方面，可以使用与开发传统Web应用程序相同的环境。编码时也可以使用自己喜欢的编辑器或者IDE（Intergrated Development Environment，集成开发环境）。如果选择Ruby

① https://www.appcelerator.com/platform/titanium-platform/

② http://jquerymobile.com/

on Rails等Web应用程序框架的话，还可以使用这个框架中提供的调试环境和测试环境。

混合应用程序

所谓混合应用程序，顾名思义，就是本机应用程序和Web应用程序的结合。应用程序本身由本机应用程序组成，不过在本机应用程序中嵌入了名为WebView的浏览器组件。此时使用本机应用程序查看网站时显示的页面，与Web应用程序的页面是一模一样的。

由于混合应用程序本身是作为本机应用程序开发的，因此不仅可以发布到应用商店中，也可以使用设备上的固有功能。另一方面，因为混合应用程序的页面和页面跳转又是一个Web应用程序，因此开发和修改都可以在服务器端快速地完成。

在混合应用程序中，可以充分利用本机应用程序和Web应用程序中的各种优势。既可以拥有本机应用程序的精准控制能力和性能方面的优势，也可以拥有Web应用程序的快速开发能力，从而担当起开发大部分内容的任务。总之，充分地利用不同类型的应用程序的优势十分重要。

◉混合应用程序的开发

当然，在开发混合应用程序时，无论是本机应用程序还是Web应用程序都需要开发。

Web应用程序的部分前面已经说过，在开发时需要注意智能手机的限制。而本机应用程序的部分一定会根据目标设备的环境而不同，因此就开发前的准备工作和开发本身来说，也必须要花费一定的时间和精力。

混合应用程序开发的支持工具中有一个名为PhoneGap[③]的框架工具，使用它可以将HTML5+CSS3+JavaScript开发的Web应用程序直接转换为本机应用程序，进而成为一个混合应用程序。在此基础上，由于PhoneGap自身也提供了使用本机功能的程序库，因此在基于Web应用程序进行开发时，也可以使用设备中的固有功能。

此外，Adobe PhoneGap Build[④]云服务提供了将Web应用程序上传，就可以将其构建为本机应用程序的功能。

应用程序测试的分类

本特辑将会提及各种测试角度，在这里我们先从这些角度出发对测试进行整理和分类。

根据开发阶段分类

根据开发的阶段或测试对象的粒度，测试可以作如下分类。

- 单元测试
- 集成测试
- 验收测试

单元测试是指针对模块等最小单元进行的测试。

集成测试是指在存在多个模块或组件、应用程序或服务的情况下进行的测试。像模块集成测试与组件集成测试这样，要通过明确集成对象来显示系统的粒度[⑤]。集成测试一般由开发小组实施。

验收测试并不是由开发小组执行的测试，而是由接受产品的公司或组织来测试。测试对象的规模与系统级别的集成测试相当，但在整个开发过程中，验收测试和集成测试实施的阶段并不同。

在本特辑中，第2章、第3章、第5章将专注于集成测试的介绍，而在第4章中，单元测试和集成测试都会重点介绍。

③ http://phonegap.com/

④ https://build.phonegap.com/

⑤ 有时候根据测试对象的集成程度使用的名词也会不同，如果比较小的时候被称作集成测试、较大的集成度被称为系统测试。

根据测试的执行方式分类

按照其执行的方法，测试可以作如下分类。

- 手动测试
- 自动测试
- 半自动测试

自动测试是无需人工干预，通过程序自动运行的测试。与之相对，手动测试就是人通过自己的手和眼睛通一边操作应用程序，一边执行的测试。

在这里还有一个概念，那就是介于自动测试和手动测试之间的执行状态——半自动测试。具体来说，就是测试是通过程序自动执行的，而测试的结果则是手动检查的。反之，手动执行测试，自动确认结果的测试方法也归类为半自动测试。

在本特辑中，整体将以自动测试为主，当然也会接触一些半自动测试和手动测试的内容。

在下文中，我们将介绍自动测试和手动测试的特征。关于半自动测试将在第2章中介绍。

◉ 自动测试的特征

自动测试中，一次性编写测试代码后可以多次自动执行，因此在需要反复执行测试的时候就能够看到它的威力了。

然而，由于需要编写测试代码，会增加初期开发成本，在应对需求变更的时候也需要花费相应的成本。因此，如果只是执行一次的测试或是在需求变化频繁的阶段，自动化测试的效率很低，并不是最好的选择。尤其是在UI自动化测试的时候，比起非UI测试，编写代码的难度更大、付出的成本也更高。

如果要执行的测试依赖于和设备关联的物理条件，例如通过全球定位系统（GPS）获取的位置信息等，那么该测试的自动化可能会变得很困难甚至无法实现。有些时候也会出现很难验证测试结果有效性的情况。例如，为了自动确认页面显示和布局，需要进行屏幕截图等工作，但对于视频和动画等动态的输出结果，想要确认的话在技术上是非常困难的。

◉ 手动测试的特征

因为不需要开发测试代码，所以在手动测试中需要专注的是测试用例的设计、执行以及资源计划。如果需要重复执行测试，那么就需要耗费与执行次数成正比的时间和人力资源，因此测试执行的效率很低。

在测试执行速度和同质化方面，手动测试多依赖于测试执行者的专业知识和技能熟练程度，测试的结果容易出现偏差。

在确认测试结果的时候，页面输出的手动确认灵活性较高，可以应对所有测试用例。然而，用眼睛确认时，只能大概地确认页面的显示结果，如果要精确到每个像素的正确性就十分困难了。这个问题在确认视频和动画的测试结果时也同样存在。

根据测试的目的分类

从测试目的或质量的角度来看，测试可以作如下分类。

- 功能测试
- 性能测试
- 安全性测试

功能测试是验证应用程序是否满足功能需求的测试；性能测试是检查能否实现预期性能的测试；安全性测试是为了确认安全性需求和漏洞的测试。总之，测试可以按照执行测试是为了达成什么目的来进行分类。这样的话，测试可以分成很多类，前面列出来的三种只是众多分类中的一部分。

在本特辑中，我们将通过一个实例介绍功能测试。

根据测试技术分类

测试技术指的就是如何创建测试用例，从这个角度来看，测试可以作如下分类。

- 白盒测试
- 黑盒测试
- 灰盒测试

白盒测试着眼于测试对象的内部结构来创建测试用例。例如，验证程序处理的执行顺序是否正确，或是测试程序内部某个值的变化。

黑盒测试着眼于测试对象的规格来创建测试用例。测试时并不在意测试对象内部执行了哪些处理，而是测试程序在输入了特定的内容后能否得到正确的结果。

顾名思义，灰盒测试介于上面两者之间，创建设计用例时既着眼于程序功能，也关注内部结构。例如，灰盒测试可以从黑盒测试的角度创建测试用例，也可以从白盒测试的角度编写测试代码。这种情况下，可以通过测试代码掌握程序的内部结构并加以利用，因此就能在黑盒测试中创建某些重现困难的特定条件来执行测试。

本特辑将重点关注黑盒测试和灰盒测试，不过在第4章中也会接触到一些白盒测试的内容。

智能手机应用程序的测试策略

至此，我们已经了解到了智能手机应用程序以及应用程序测试的分类。现在，测试已经成为了应用程序开发中不可或缺的一个部分。根据应用程序的不同，测试的方法和难易度也会产生差异。

单就测试方法来说，可以先大致分为本机应用程序特有的测试方法和Web应用程序的测试方法两种。而在测试混合应用程序时，要根据测试对象区别对待，将本机应用程序和Web应用程序中的各种测试方法适当地组合起来。

本机应用程序的测试

在测试本机应用程序时，因为不同的目标设备使用不同的开发语言，所以基本上测试也要依赖于特定的开发语言。例如，iOS应用程序的测试代码要用Objective-C来实现，Android应用程序的测试代码则要用Java来实现。

这里先简要介绍一下本机应用程序的单元测试和集成测试。第2章将会更加详细地介绍本机应用程序的集成测试。

◉ 本机应用程序的单元测试

在执行本机应用程序的单元测试时，首先要像上面说的那样，根据目标设备编写相应的测试代码。简而言之就是使用某个测试框架来编写测试代码。测试框架中具有代表性的有Objective-C语言的GHUnit[6]和Java语言的JUnit[7]，等等。

针对智能手机的单元测试与普通的单元测试一样，执行测试时要以应用程序的逻辑部分为中心，并且要尽量涵盖应用程序的控制器和处理器等部分。

◉ 本机应用程序的集成测试

在执行本机应用程序的集成测试时，要使用模拟器和设备本身实施UI测试。在Web应用程序的浏览器中，即使执行测试的平台不同，只要浏览器相同操作就能统一响应。但是在本机应用程序中，执行测试的平台不同，操作响应的方式和开发语言就也都不同。因此，目标OS之间的差异还是成为了多设备测试的阻碍。

目前现有的面向集成测试的UI自动化测试工具，都是实现了自动操作本机应用程序的框架。有关本机应用程序的UI测试和框架，请参考第2章。

Web应用程序的测试

在测试Web应用程序时，可以直接使用在传统Web应用程序测试中总结出来的测试方法。其中一个最常用的模式就是在单元测试中测试应用程序的逻辑，在集成测试中使用浏览器进

⑥ https://github.com/gabriel/gh-unit
⑦ http://junit.org/

行端到端测试（End To End Testing）。在Web应用程序测试中，这一模式也可直接使用。

在这里，我们简要地介绍了智能手机Web应用程序的单元测试和集成测试。关于浏览器测试的内容将在第3章中详细介绍。第4章则会详细介绍使用JavaScript开发的Web应用程序该如何测试。

◉Web应用程序的单元测试

因为可以把Web应用程序看作是一个传统的Web应用程序，所以也不需要特意介绍它的单元测试了。只需要根据所使用的WAF（Web Application Framework，Web应用程序框架）的结构以及特性，针对应用程序的业务逻辑、控制器等方面实施测试即可。

◉Web应用程序的集成测试

集成测试其实并不是很难，沿用传统Web应用程序的浏览器测试也没有问题。可以利用Selenium等框架，根据目前所积累的技术和实践经验来实施测试。

要说智能手机Web应用程序与PC机Web应用程序之间的差别的话，那就是前者的页面技术采用了HTML5和CSS3等最新的技术、存在触摸和滑动等针对智能设备的特殊操作，还有页面大小是固定的，等等。

不过话说回来，如果能在一定程度上更新测试时使用的PC机浏览器并利用扩展功能的话，PC机的Web应用程序也能够支持最新的Web开发技术，虽然不是完全支持。还有，如果使用模拟器或真实设备来进行测试的话，在测试时就可以考虑加入触摸和滑动等操作，也可以正确重现页面使用的实际环境。

本章小结

在本章中，我们围绕着智能手机开发的现状进行了说明，也为智能手机的应用程序分了类。另外，还在结合应用程序分类的基础上大致说明了各种应用程序的测试策略。

在后面的章节中，我们将接触到一些智能手机开发中的具体示例，并通过这些示例中的场景说明测试的思路以及自动化测试的方法。

专栏

测试代码的可维护性

对于测试代码来说，能够使操作自动化固然重要，但是可读性和扩展性等也十分重要。如果编写出来的测试代码虽然可以执行自动化测试，但是却毫无可维护性，这样的代码不仅无法继续使用下去，而且人们也会逐渐忘记它是测什么的以及怎么测的。这样的话，测试代码就完全失去意义了。

最近，出现了一些与浏览器测试中经常用到的程序库类似的，面向本机应用程序的框架。例如，在本机应用程序测试时，使用通过导入代理层来弱化不同操作系统间差异的UI测试框架，以及使用DSL（Domain Spacific Language，特定领域语言）以更容易理解的方式描述本机应用程序操作的框架等。

第2章 本机应用程序的UI自动化测试

使用Calabash编写跨Android/iOS平台的测试

山内沙瑛　YAMAUCHI Sae
贾成锴　JIA Chengkai
小俣裕一　OMATA Yuichi
DeNA股份有限公司
译/刘卓

在对智能手机的本机应用程序实施UI自动化测试的时候，会出现测试框架或使用的语言因手机系统的平台而异、开发资源受到限制等多个问题。在本章中，我们将要介绍如何使用Calabash测试框架解决这些问题，实现跨平台的UI自动化测试。

Calabash简介

Calabash的特点

首先来简单介绍一下Calabash的特点。

◉同时支持Android/iOS

Calabash可以同时支持Android/iOS两个平台的真机和模拟器，将两个平台的测试开发语言和测试代码等通用化，实现高效的测试开发。

◉支持最新的操作系统

由于目前框架也处在积极开发的阶段，所以用Calabash编写的测试代码不论是现在还是将来，都可以运行在最新版本的操作系统上。

◉支持Web视图

由于通过组合Calabash API的查询和CSS选择器可以指定Web视图（WebView，分为iOS的UIWebView和Android的WebView）中的元素，因此可以完成点击链接或向文本域内输入字符等操作。

```
query("webView css:'#header'")
touch("webView css:'a'")
set_text "webView css:'login'", "hoge"
```

◉支持定制视图组件

支持单独实现的定制视图组件。

◉使用Ruby编写测试代码

如果在测试时，既要使用非主流的开发语言，又要使用特殊的框架，那恐怕要花费很多学习成本。Calabash的默认开发语言是Ruby，因此使用时完全不需要有这样的担心[①]。

◉可以分别编写测试场景和测试代码

Calabash框架的主要目的是验收测试，因此使用了在Rails开发中常用的BDD（Behavior Driven Development，行为驱动开发）工具Cucumber[②]。使用Cucumber可以让测试人员等非开发人员编写测试场景，来分担开发人员的工作，而开发人员只需编写测试代码即可。

◉提供实现测试的支持功能

开发测试的方法有两种，一是从头开始编写测试代码，二是按照测试场景对用户操作进行录制和回放。Calabash可以利用录制和回放功能作为测试开发过程的一部分[③④]。

① 预计在将来也会支持Ruby之外的语言。
② 也可以和Cucumber之外的工具配合使用。
③ 在本文发布时只有calabash-iOS中包含此功能。
④ 这句话的意思是Calabash可以先用录制和回放功能录制测试场景的代码，然后再通过编程的方式修改前面录制好的代码，以此完成测试的开发。——译者注