跨平台移动应用中间适配层设计与实现

施 伟[1,3],王硕苹[2],郭 鸣[2],吴明晖[2],梁 鹏[1,4]
SHI Wei[1,3], WANG Shuoping[2], GUO Ming[2], WU Minghui[2], LIANG Peng[1,4]

1.浙江大学 计算机科学与技术学院,杭州 310027
2.浙江大学城市学院 计算机与计算科学学院,杭州 310015
3.中国人民解放军 91199部队
4.中国人民解放军 94936部队
1.College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
2.College of Computer and Computation Science, Zhejiang University City College, Hangzhou 310015, China
3.Unit 91199 of PLA, China
4.Unit 94936 of PLA, China

SHI Wei, WANG Shuoping, GUO Ming, et al. Design and implementation of cross-platform mobile application middle adaptation layer. Computer Engineering and Applications, 2014, 50(16):39-44.

Abstract: Due to the incompatibility between the current popular mobile developments platforms, cause all kinds of waste of application development resources. In order to resolve the incompatibilities of the various platform application development, this paper proposes a solution that is to add a middle adaptation layer between mobile platform operating system layer and application layer. Adaptation layer encapsulates through a browser with Webkit as the core and extensions, support cross-platform mobile application development, mobile terminal access to local resources on a different platform and has a good support. The middle adaptation layer has good versatility and scalability, and has carried out simulation experiments on multiple platforms to verify the feasibility and practicability of the solution.
Key words: cross-platform; mobile application; middle layer; HTML5

摘 要: 由于当前主流的移动开发平台之间互不兼容,造成应用开发各种资源的浪费。为了解决各个平台应用开发的不兼容问题,提出在移动平台操作系统层和应用层之间添加中间适配层的方案。中间适配层通过对以Webkit为核心的浏览器进行封装和扩展,支持跨平台的移动应用开发,对不同平台移动终端的本地资源访问也有较好的支持。该中间适配层具有良好的通用性和扩展性,并已在多个平台进行仿真实验验证了方案的可行性和实用性。
关键词: 跨平台; 移动应用; 中间层; HTML5
文献标志码: A **中图分类号:** TP311.52 doi:10.3778/j.issn.1002-8331.1208-0481

1 引言

随着3G网络技术和移动互联网的快速发展,移动终端已经由功能性向智能性转变。Canalys 2012年2月数据显示,全球50.1%的智能终端搭载了Android系统,下面依次是iOS和BlackBerry,分别占据了较大的市场份额,如表1所示。因此要想获得更多的用户,选择单一平台来开发和发布的终端应用不再是一个可行的选择。

每个平台通常具有其自己的软件开发工具包和语言或支持的语言,见表2所示。由于当前主流的移动平台之间互不兼容,针对不同的移动平台系统,当前并没

基金项目: 国家科技重大专项(No.2011ZX0302-004-002);浙江省重点科技创新团队项目(No.2010R50009);浙江省科技厅公益技术研究项目(No.2011C33015)。
作者简介: 施伟(1980—),男,硕士研究生,研究方向为移动互联网应用;王硕苹(1972—),女,副教授,研究领域为信息系统设计、软件架构;郭鸣(1972—),男,博士,副教授,研究领域为知识表示、语义Web;吴明晖(1976—),男,通讯作者,博士,教授,研究领域为软件工程、人工智能;梁鹏(1982—),男,硕士研究生,研究方向为数据库安全。E-mail:mhwu@zucc.edu.cn
收稿日期: 2012-09-05 **修回日期:** 2012-11-20 **文章编号:** 1002-8331(2014)16-0039-06
CNKI **网络优先出版:** 2012-12-03, http://www.cnki.net/kcms/detail/11.2127.TP.20121203.1559.005.html

表1 2012年2月各平台市场份额[1]

平台	市场份额/(%)
Google	50.1
Apple	30.2
RIM	13.4
Microsoft	3.9
Symbian	1.5

表2 平台开发需要的语言[2]

系统平台	所需语言
Apple iOS	Objective-C
Google Android	Java
RIM BlackBerry	Java
Symbian	C,C++,Python,HTML/CSS/JS
Windows Mobile,7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C,C++,HTML/CSS/JS
Samsung bada	C++

有可以兼容的应用开发接口和语言。一个平台开发的应用程序不会轻易转化到另一个平台。

原生应用程序通过访问设备的API和框架,从而使设备的功能得到最佳发挥。但需要使用该设备的硬件和软件的开发人员更加专业化,以获得最大的用户体验,因此为每个平台开发原生应用的代价更为昂贵。

为了解决各个平台应用开发的不兼容问题,一种替代方案就是尝试在不同设备的应用层之间的抽象共性。例如,所有的智能终端有一个Web浏览器。移动Web应用程序可以是一种方法。另一种方法是使用一个框架,可以在应用程序中嵌入设备的浏览器,并提供应用程序编程接口(API),允许Web代码和设备硬件交互的一种混合方法。

移动Web应用程序,特别是那些利用HTML5的特性来编写移动应用程序是很有潜力的。例如,移动Web应用程序易安装、分布性良好,开发人员的支持[3]。HTML5 API包括联机和脱机模式下与应用程序进行交互的能力,开发人员可以使用智能终端上的音频、视频和有限的设备传感器比如GPS等数据。但是,移动Web应用也存在劣势。比如对没有定位传感器装置终端的支持非常有限。对内容捕获的摄像头和麦克风的支持也是很有限的。在一些本地资源的使用方面,Web应用的用户体验不及原生应用程序那么良好。

本文结合国家科技重大专项课题(移动互联网智能终端应用中间件开发)的研究,将原生应用和Web应用开发的优势结合起来,提出了基于浏览器作为中间层的跨平台智能终端应用设计方案。本文分析其设计原理和实现技术,给出符合W3C标准的、统一的API。然后使用HTML5、CSS和JavaScript开发应用程序并在不同平台进行仿真实验来验证方案的可行性和实际效果。

2 相关工作

随着人们对跨平台应用开发研究的不断深入,目前主要有以下相关研究。文献[4]指出对于移动开发者来说很难找到最合适的开发平台,分析Android、iPhone、Qt 的关键开发技术,重要的共同点和差异,但没有解决跨平台的问题。解决跨平台的一种方案就是尝试在不同设备的应用层之间的抽象共性。比如文献[5]提出了一种通用的平台,此平台需要一台互联网服务器通过一个特定的XML文档保持与智能手机的连接。每个智能手机的用户所做的更改会影响服务器,也会影响用户各自的操作系统中的XML文件中的数据,这样使所有其他用户得到最新的状态和数据连续更新。但是目前只是在Android和Blackberry平台上实验成功,而且特定的XML文件的传输问题很大程度上决定方案的可行性。另一种方法是使用一个框架,文献[6]提出了HTML5开发移动应用实现跨平台,介绍了一些可用框架和移动开发工具。国内的主要有AppCan和ExMobi。AppCan免费但不是开源的,ExMobi是商业性质的。国外的比如PhoneGap、jQuery Mobile、Sencha和Titanium,但是PhoneGap不支持UI设计,jQuery Mobile不支持访问本地资源,Sencha和Titanium性能和用户体验没有原生应用的好。相对于以上相关工作,本方案与之相同之处是由HTML、JavaScript编写的应用,易发生代码篡改的问题,存在一定的安全问题。本方案与之不同的是提供符合W3C标准的统一的API,并且具有较高的灵活性和良好的可扩展性。

3 智能终端应用中间层设计与实现

本文将原生应用和Web应用开发的优势结合起来,提出一种基于浏览器作为中间层的跨平台智能终端应用设计方案。

3.1 跨平台智能终端应用设计方案原理

Webkit是当前最新的、速度最快的开源浏览器引擎。Webkit支持多种移动应用所需要的HTML5特性。目前在Android和iOS等主流浏览器中,都对这些特性提供了本地支持。本方案主要设计原理是针对不同移动平台的操作系统层之上添加一层中间适配层,此中间适配层对上层(Mobile Application)提供统一的服务和接口,对下屏蔽各移动智能终端操作系统的差异。其在移动应用和设备之间搭建一个通信的桥梁(Middleware Layer),封装移动设备的平台差异,统一使用JavaScript接口实现JavaScript和本地API之间的调用和通信,从而提供跨平台解决方案。中间适配层利用基于Webkit为核心的浏览器的插件扩展机制可以提供对智能终端设备的本地资源的访问和支持。本设计方案主要有以下优点:

(1)跨平台,屏蔽移动智能终端操作系统的差异,从

而实现“一次编码，多处运行”。

(2)直接访问移动智能终端本地资源，通过统一的API可以直接访问联系人、短信、摄像头、GPS、WIFI、蓝牙、多媒体、数据库和文件系统等本地资源。

(3)本方案提供的API完全兼容W3C标准，而且提供统一标准和丰富的API。

(4)易于使用，本方案完全采用HTML5+CSS+JavaScript技术开发移动智能终端应用，丰富的互联网应用程序可以稍做修改即可成为移动智能终端应用程序。

(5)具有较强的灵活性和扩展性，开发者可以利用现有成熟的JavaScript库和UI框架开发跨平台的移动应用。

跨平台移动应用中间层设计架构如图1。

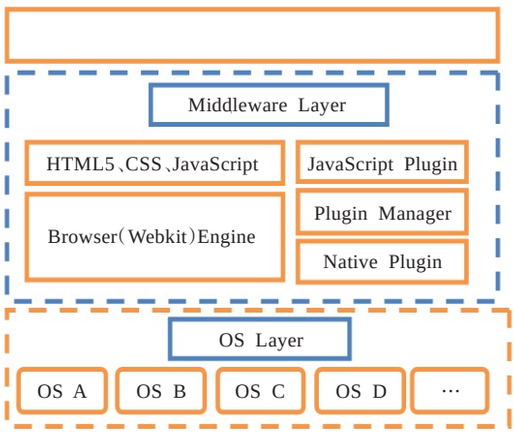

图1　跨平台移动应用中间层设计架构图

3.2　跨平台智能终端应用设计方案的实现

众所周知，不同的移动平台已内置浏览器功能组件。浏览器具有一个本地API和移动设备双向通信的基本能力，可以通过调用本地 JavaScript访问设备的API[7]。JavaScript在浏览器组件中的通信有两种方式，即异步通信（Ajax)和同步通信。Ajax称为"异步JavaScript和XML"，是一种创建交互式Web应用程序的通信技术。使用 Ajax 的最大优点是维护数据时在无需更新整个页面的前提下更新局部数据，大大减轻了页面服务端的负担，使用户的感觉更加直观，使浏览器的交互能力大大加强。Ajax技术可以用于在后台，实现与服务端的Web应用程序进行通信(http://en.wikipedia.org/wiki/Ajax_(programming))。然而Ajax是不可以跨域的，也就是说如果Web端的html不是本地的文件而是从远端服务器下载下来的，那么它就不能向本地的server发起Ajax请求(因为不同域)，所以本方案选择XMLHttpRequest(http://www.w3.org/TR/XMLHttpRequest/)和JSONP同用，JSONP是一个标准的解决Ajax跨域的方案。

在开发移动智能终端应用过程中各平台之间最大的不兼容主要表现在各平台的API上，比如在处理事件、错误、请求使用元数据和访问本地系统资源上API表现各不相同[8]。为此，需要开发符合W3C标准的统一的API(http://www.w3.org/2012/05/mobile-web-app-state/)，包括Geolocation、WebGL、Device、Media、Connection、Notification、Storage、Contacts、Sensors和File API等。而且要考虑每一个跨平台的开发方案都要面临满足开发者需求和满足用户体验的挑战[9]。本方案采用一个具有基本浏览器功能的组件来渲染HTML，使用一个插件模型来封装本地API，它涵盖了浏览器原来的基本功能和方法来实现一个Web端口上的移动设备的本地调用和移动设备服务端端口到本地Web端口返回异步或同步调用的结果，并在各个移动平台上封装API。这样通过HTML5、JavaScript、CSS等Web技术实现的本地应用的表现层，直接由Webkit引擎来渲染呈现，同时也能提供更丰富，且与原生应用相同的用户体验。图2是插件模型的架构图。中间适配层包括Browser(Webkit) Engine、JavaScript Plugin、Plugin Manager和Native Plugin。具体流程是由Browser(Webkit)Engine渲染HTML来呈现Web内容，移动应用(HTML、JavaScript、CSS)通过JavaScript API调用基于Plugin模式的封装Native API，以XHR或JSONP的方式来实现Native端向Web端返回异步调用的结果。通过持久性的XHR连接，JavaScript可以不断轮询内部XHR服务器存储的信息，从而实现了从Native端到Web方向的通信。从Native端返回的结果进而由Browser(Webkit)Engine渲染并显示。

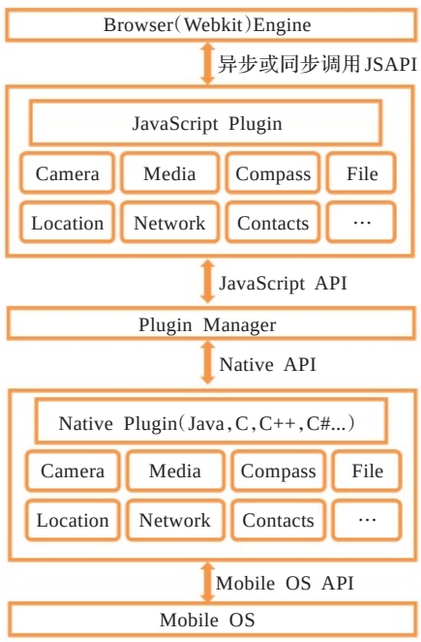

图2　插件模型架构图

3.2.1　插件管理模块的设计与实现

插件的核心方法为execute方法，将负责实际处理接口调用请求。插件管理模块分为接口，接口父类，服务(例：Contacts)接口子类，三者关系如图3所示。

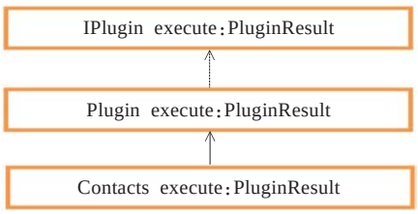

图3　插件管理类图

IPlugin接口为模块的接口,由Plugin抽象类实现。在Plugin中,execute方法为抽象方法,必须由各个继承Plugin的服务接口类来实现,负责处理实际的口调用请求。以下是Web客户端通过JavaScript调用移动智能终端的Native API的流程,见图4。

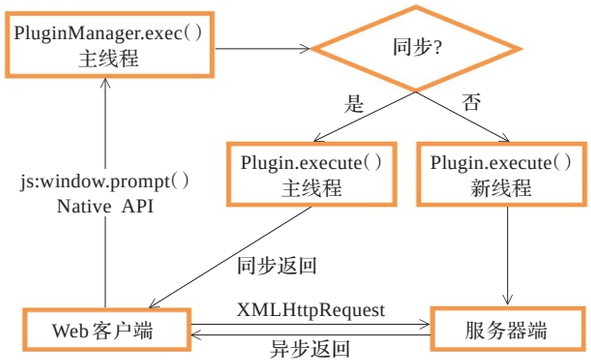

图4　中间层执行流程图

如图4所示,中间层将Web客户端调用Native API请求包装为prompt()事件,因此,中间层通过监听JSPrompt()事件,获取适配层的接口调用请求。以Android平台为例,平台本身提供了监听应用层事件的机制,通过继承Activity类,并重载其onJsPrompt()方法,可以将应用程序层的接口调用请求事件捕获,onJsPrompt()方法通过调用PluginManager.exec()方法,将所接收的调用请求进行分发并处理。如果是同步请求,则直接由主线程的插件的Plugin.execute()方法执行,然后就执行结果PluginResult返回给Web客户端即移动应用程序;如果是异步请求,则将启动新的线程来处理,处理完后,将结果通过服务器端写到客户端。服务器端相当于XMLHttpResponse,负责将数据异步写到客户端。它在内部会有一个socket监听,不停的接收来自于客户端的请求,如果发现变量(JavaScript)中有数据,就写到客户端,如果没有,则休眠片刻,休眠后,如果有数据,则写到客户端,否则写一个404异常到客户端,然后此次连接中断,重新接收新的客户端请求。

3.2.2　Native API模块的设计与实现

上面已经提到服务接口子类,Native Plugin必须由各个继承Plugin的服务接口类来实现。以SMS为例给出服务子类的Java[10]实现原型。所有服务子类的实现严格按照W3C标准执行。按照相应需求设计服务子类的属性和方法。

```
public class SMS extends Plugin{
public PluginResult execute(
String action,JSONArray args,StringcallbackId){
//execute the action
}
//implement other utilities
}
```

其中execute方法的参数分别为:

action:被调用的行为

args:被调用行为的相关参数

callbackID:用于Web客户端回调的ID

Native Plugin类在执行来自Web客户端的调用请求之后,返回的对象为PluginResult。PluginResult根据调用请求的callbackID,返回onSuccess与onError结果,其实现原型如下:

```
public class PluginResult{
private final int status;
private final String message;
//Constructor
public PluginResult (Status status){
this.status=status.ordinal();
this.message=(""+StatusMessages[this.status]+"");}
public int getStatus(){return this.status;}
public String getMessage(){return
this.message;}
public String toSuccessCallback(String callbackId){
  //return on Success with callbackID
  }
public String toErrorCallback(String callbackId){
  //return on Error with callbackID
  }
}
```

这样通过返回PluginResult给Web客户端完成对Native API的调用。

3.2.3　JavaScript 插件库的设计与实现

JavaScript面向对象与传统的基于类的面向对象不同,方案基于Prototype模式的接口构造,通过对象中的Prototype属性,返回对象的原型引用。

Prototype模式是一种对象创建型模式,它跟工厂模式,Builder模式等一样,都用来创建类的实例对象。它通过拷贝这些原型创建新的对象,其UML类图结构如图5所示。它适用于以下几种情况[11]。

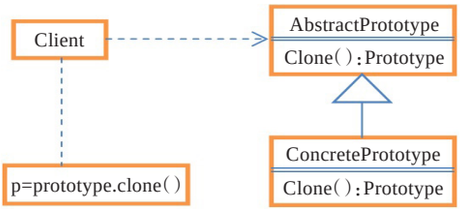

图5　Prototype模式UML类结构图

(1)当一个系统应该独立于它的产品创建、构成和表示时；

(2)当要实例化的类是在运行时刻指定时；

(3)为了避免创建一个与产品类层次平行的工厂类层次时；

(4)当一个类的实例只能有几个不同状态组合中的一种时。

AbstractPrototype：声明一个克隆自身的接口。

ConcretePrototype：实现一个克隆自身的操作。

Client：原型克隆自身从而创建一个新的对象。

JavaScript为每一个类型都提供了一个Prototype属性[12]，将这个属性指向一个对象，这个对象就成为了这个类型的“原型”，这意味着由这个类型所创建的所有对象都具有这个原型的特性。

对于JavaScript来说，每个具体的JavaScript类型有且仅有一个原型(Prototype)，即原型继承不能用于多继承。每个类型的实例的所有类型，必须是满足原型关系的类型链。以SMS为例，SMS接口有send方法的访问。SMS接口下，send方法的构造实现如下：

```
var SMS=function (){};
SMS.prototype.send=function (phone, message, successCallback, failureCallback) { return Plugin.exec (successCallback,failureCallback,SMS,"send",[phone,message]);
};
```

然后在插件中注册，方法如下：

```
Plugin.addConstructor(function(){
Plugin.addPlugin("sms",new SMS());
});
```

注册后就可以在应用中通过JavaScript调用SMS的send方法发送短信了。

各平台封装对应的API，具体如表3。

表3　JavaScript API

API	属性	方法	参数
File	File.name File.type File.size …	FileEntry.moveTo() FileEntry.copyTo() FileEntry.remove() …	parentEntry, newName, success,fail
Device	name platform version	vibrate() beep() getInfo()	inMilliseconds inMilliseconds inInfoID
Compass		getCurrentHeading() watchHeading() clearWatch()	onSuccess, onError,options watchID
Contacts		find() create()	fields,onSuccess, onError,options properties
SMS		send() read()	onSuccess, onError,options
…	…	…	…

限于篇幅有限，API没有完全列出。

4　仿真实验

本文提出的移动应用中间层已在多个平台进行了应用开发验证。

此处以发送短信为例，以相同的应用程序(含HTML、JavaScript和CSS文件)分别在Win Phone7平台、Android平台和palm webOS平台上进行仿真实验。

```
<label for="phone">电话号码:</label>
<input type="tel" id="phone" name="phone"
placeholder="请输入电话号码"/>
<label for="message">短信内容:</label>
<textarea id="message" name="message"
 placeholder="请输入内容"></textarea>
<a href="#" id="send" onclick="sendsms(
$('#phone').val(),$('#message').val());"
        data-role="button">直接发送</a>
```

图6　HTML代码

```
53 //2012.05.26 added by shiwei
54 function sendsms(phonenum,msg){
55     navigator.sms.send(phonenum,msg,
56             function(){
57         alert('短信发送成功!');
58     },function (e) {
59             alert('短信发送失败:' + e);
60         });
61
62 }
```

图7　JavaScript代码

仿真结果如图8~10所示。

图8　Win Phone7平台仿真实验

图9　Android平台仿真实验

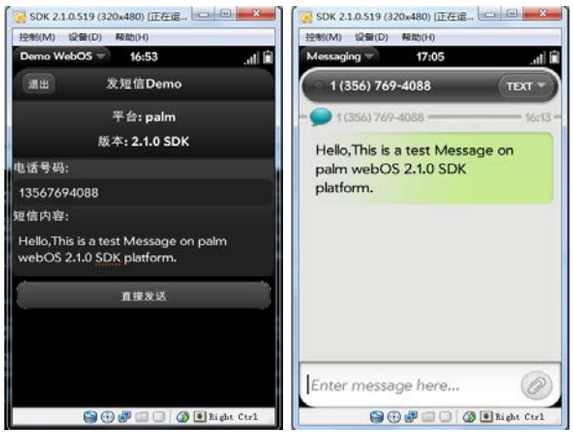

图10 palm webOS平台仿真实验

在图8,图9和图10中,分别调用中间适配层的API函数,这里是调用sendms(phonenum,msg)方法,包含phonenum和msg两个参数,phonenum表示要发送的电话号码,msg表示要发送的短信内容。图8,图9和图10分别展示了在win phone7、Android和webOS平台上的效果。中间适配层可以很好地支持移动应用开发。安装并配置相关平台的开发环境,在HTML中调用中间适配层的API库,比如<script type="text/javascript"charset="gb2312"src="main.js"></script>,其中main.js是中间适配层API库。开发者根据需要可以调用中间适配层提供的各种函数访问本地资源和网络资源,以开发各种移动应用。

5 结束语

本文提出了基于中间层的跨平台移动智能终端应用方案设计并实现。通过理论设计和在不同平台的仿真实验,可以肯定本方案有很多优势:(1)是跨平台。(2)是可直接访问智能终端的本地资源。(3)是提供符合W3C标准的统一的API。(4)是降低移动智能终端应用开发的难度。(5)是具有较高的灵活性和良好的可扩展性。但本方案也有一些不足之处:(1)是开发的移动应用对HTML5的支持程度受制于Webkit浏览器内核。(2)是由HTML、JavaScript编写的应用,易发生代码篡改的问题,存在一定的安全问题。(3)是它不支持所有的平台,因为有一些特殊的API,例如日志记录的API和WRT平台的传感器API。

参考文献:

[1] comScore.comScore Reports February 2012 U.S.Mobile Subscriber Market Share[EB/OL](2012-04-07).http://www.comscore.com/Press_Events/Press_Releases/2012/4/comScore_Reports_February_2012_U.S._Mobile_Subscriber_Market_Share.

[2] Charland A,Leroux B.Mobile application development:web vs.native[J].Communications,2011,54(5):49-53.

[3] Melamed T,Clayton B.A comparative evaluation of HTML5 as a pervasive Media platform[J].Social-Informatics and Telecommunications Engineering,2010:307-325.

[4] Lettner M,Tschernuth M,Mayrhofe R.Mobile platform architecture review:Android,iPhone,Qt[R].Lecture Notes in Computer Science,2012.

[5] Iyer A,Jadhav A,Dhangare N.Common platform for mobile application[J].Advances in Computer Science and its Applications,2012,1(2):174-184.

[6] Pavel S.Mobile development tools and cross-platform solutions[C]//2012 13th International Carpathian Control Conference (ICCC),2012:653-656.

[7] Shi Wei,Wu Minghui.Local resource accessing mechanism on multiple mobile platform[C]//High Performance Computing and Communications,2012:1716-1721.

[8] Mendes P,Caceres M,Dwolatzky B.A review of the widget landscape and incompatibilities between widget engines[C]//IEEEAFRICON,2009:23-25.

[9] Ohrt J,Turau V.Cross-platform development tools for smartphone applications[J].IEEE Computer Society,10.1109/MC.2012.121.

[10] Skrien D.Object-oriented design using Java[M].腾灵灵,仲婷,译.北京:清华大学出版社,2009:173-192.

[11] Taivalsaari A.Kevo-a prototype-based object-oriented language based onconcatenation and module operations[R].Canada,University of Victoria,B C,1992.

[12] 阎宏.Java与模式[M].北京:电子工业出版社,2002:317-343.

跨平台移动应用中间适配层设计与实现

作者： 施伟， 王硕苹， 郭鸣， 吴明晖， 梁鹏， SHI Wei， WANG Shuoping， GUO Ming， WU Minghui， LIANG Peng

作者单位： 施伟,SHI Wei(浙江大学 计算机科学与技术学院，杭州 310027; 中国人民解放军 91199部队)， 王硕苹,郭鸣,吴明晖,WANG Shuoping,GUO Ming,WU Minghui(浙江大学城市学院 计算机与计算科学学院,杭州,310015)， 梁鹏,LIANG Peng(浙江大学 计算机科学与技术学院，杭州 310027; 中国人民解放军 94936部队)

刊名： 计算机工程与应用 ISTIC PKU

英文刊名： Computer Engineering and Applications

年，卷(期)： 2014(16)