Digital Library Forum 数字图书馆论坛
http: // www.dlf.net.cn

基于智能手机平台的OPAC系统设计和实现——以Android平台为例*

施千卫 黄志强 楼向英 刘翔 / 浙江理工大学图书馆 杭州 310018

摘要：随着信息通信技术与智能手机的不断发展和普及，基于智能手机平台的移动应用在各个领域的发展也越来越快。文章着重讨论了图书馆OPAC系统作为移动应用在Android平台智能手机上的设计和实现，从而使图书馆服务更加符合时代需求。

关键词：智能手机，OPAC，Android
DOI：10.3772/j.issn.1673—2286.2011.11.005

1 引言

近年来，随着信息通信技术的不断发展，智能手机越来越受到关注和普及，各种智能手机平台也随之应运而生。目前智能手机平台主要有Symbian（诺基亚）、iOS（苹果公司）、Android（Google）、Windows Mobile（微软）、BlackBerry（RIM）等。据国际市场分析公司尼尔森的分析报告显示[1]，截至2011年第一季度，全球智能手机市场份额中，Android和iPhone手机占总份额的62%左右，其中Android手机占33%。与此同时，基于智能手机平台的移动应用（Mobile Application）发展也越来越快，各智能手机厂商针对各自的平台纷纷开发了移动应用商店，如苹果公司的App Store，Google的Android Market，以及国内的移动运营商如中国移动的Mobile Market等。由于移动应用能给用户的生活工作带来极大的便利，而且还有丰富的用户体验，所以其呈现出了爆炸式的增长，截至目前，Android Market上的应用数量已经达32万[2]。

另一方面，通过调研各类图书馆（尤其是高校图书馆）网站可以看出，目前提供智能手机客户端应用的图书馆可以说是凤毛麟角。截至2011年4月，除国家图书馆、上海图书馆提供手机客户端外，笔者尚未发现国内高校图书馆提供手机客户端应用[3]。而在西方国家，特别是美国高校图书馆、公共图书馆对手机客户端的应用相对比较多，可详见Library Success: A Best Practices Wiki中“Mobile applications”汇集的典型案例[4]。

在此背景下，提出了图书馆书目检索系统（OPAC）在Android平台智能手机上的设计和实现。由于OPAC是图书馆业务中读者使用最为频繁的服务，所以如果把OPAC作为移动应用移植到智能手机平台上，不但能使读者可以摆脱计算机的约束，而且能随时随地进行图书查阅，同时也能享受到更人性化的用户体验，从而令图书馆服务内容和服务模式扩展到更广、更深的层次，更好地服务读者[5]。

2 基于Android平台的OPAC系统设计

2.1 设计原理

纵观当前各图书馆OPAC系统，可以看出基本都是商业化产品，所以基本不提供统一的对外编程接口。但是笔者通过对各OPAC系统的深入分析得出，各OPAC都是基于B/S结构，并且都有相似的查询请求格式，即：

http://<host>:<port>/<path>?<search-part>&<search-

* 文章系国家社会科学基金项目“移动阅读与图书馆延伸服务研究”（项目号：10CTQ004）研究成果之一。

param>

如要查书名含有android的书可以用如下格式：

http://some.opac.server/searchbook?q=android&type=name

同时查询请求所返回的数据都是基于HTML或者是XML（如RSS）等国际标准数据格式，因此如果在Android客户端发出相应的查询请求，然后根据请求取得返回数据并对数据进行抽取、封装，最后通过Android提供的丰富的UI组件（如TextView、ListView等）来进行展现，就可以达到我们所需的系统要求。

2.2 Web Service的引入

虽然Android智能手机可以看作缩小版的个人电脑，具有CPU、内存、独立的操作系统等个人电脑所必须具备的软硬件条件，但是其各方面性能，如数据处理、数据存储、图形处理等方面仍无法与个人电脑比拟；而通过设计原理可以看出，所有的操作流程、数据处理都在Android客户端完成，这无疑会增加Android系统的性能负担（如计算能力、存储能力、电池消耗等），甚至会影响到用户体验。这是Android性能设计法则[6]所不允许的，因此如何优化性能成为系统设计的关键。为此如果将影响性能的数据处理部分从系统中抽离出来，使客户端仅负责数据展现，将大大减轻客户端的压力，成为解决问题的重要因素。鉴于此，笔者构建了一个SOAP（简单对象访问协议）Web Service并部署到OPAC服务器上。首先Web Service接收来自Android客户端所发的OPAC查询请求，然后对查询所返回的数据进行解析、抽取并通过SOAP将所需的数据传回给Android客户端。SOAP是一种轻量的、简单的、基于XML的对象访问协议，所以非常适合本系统的要求。由此，Android客户端性能将大大优化。

2.3 系统框架

具体系统框架图如图1所示。

从图1中，可以看出本系统由Web服务端和Android客户端组成。其中，Web服务端主要部署了一个SOAP Web Service，用来接收Android客户端的SOAP请求，Android客户端主要由UI Component组成，用来封装和展现返回的SOAP数据。

2.4 系统实现

根据系统框架，实现也主要包含了两部分，一部分是Web Service实现，另一部分是Android客户端的实现。由于笔者所在单位所使用的是江苏汇文的OPAC，所以本系统将以此为研究对象进行实现。同理，针对其他OPAC系统可以方便移植。

（1）Web Service实现

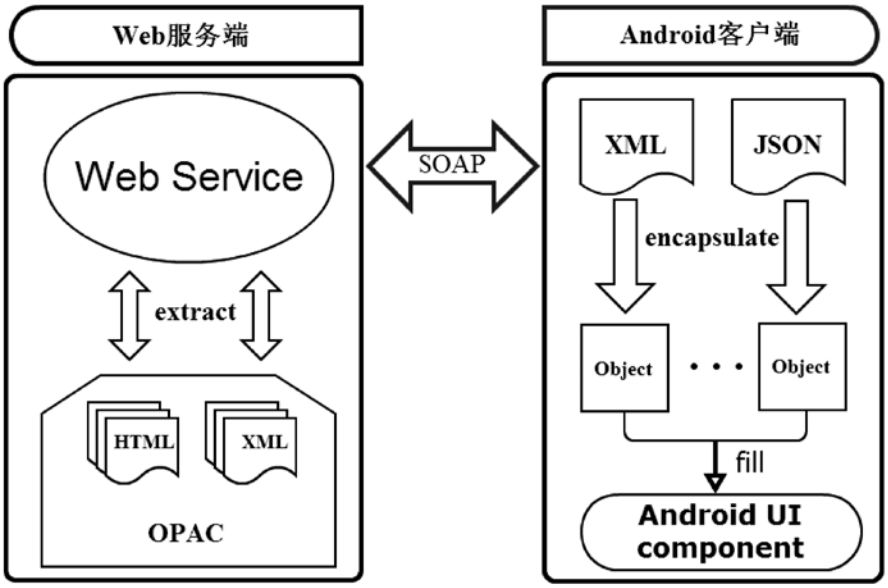

图1 系统框架

当前主流的编程语言（JAVA、.NET、PHP等）都可以搭建Web Service。由于ASP.NET具有易用、灵活、功能强大等特点，是目前快速Web解决方案的首选，所以笔者选用了ASP.NET作为开发工具。通过可视化.NET编程工具Visual Studio，创建了一个名为OPAC Web Service的服务。该服务包含了一个WebMethod：Search，其相应的SOAP格式如下：

SOAP请求：

```
<soap12:Envelope>
 <soap12:Body>
  <Search xmlns="http://tempuri.org/">
   <keyword>string</keyword>
   <searchType>string</searchType>
   <page>string</page>
   <pageSize>string</pageSize>
  </Search>
 </soap12:Body>
</soap12:Envelope>
```

SOAP响应：

```
<soap12:Envelope>
 <soap12:Body>
  <SearchResponse xmlns="http://tempuri.org/">
   <SearchResult>string</SearchResult>
  </SearchResponse>
 </soap12:Body>
</soap12:Envelope>
```

其中SOAP请求部分定义了Search方法所带的参数以及参数类型，分别为keyword、searchType、page、pageSize。keyword代表了查询的关键字；searchType表示查询类型，如书名、作者、ISBN等；page是页码；pageSize是每页显示的结果数。SOAP响应定义了Search方法返回的数据类型和值，如字符串类型、XML等。当Android客户端发出SOAP请求时，Search方法将把请求中带的参数分别写入keyword、searchType、page、pageSize，然后构建一个OPAC查询语句提交给OPAC服务器，如：

http://server/opac/search?strText=<keyword>&strSearchType=<searchType>&page=<page>displaypg=<pageSize>，服务器会返回相应的HTML或XML结果，最后对HTML或XML进行解析、抽取所有需要的数据，并将其返回。

（2）Android客户端实现

Android的客户端开发由著名的IDE工具Eclipse完成，由于Android开发是基于JAVA语言的，所以Eclipse是目前最好的选择。具体开发环境搭建可参考Android SDK手册。客户端部分主要包含了UI展示界面。Android中每个屏幕界面都称为一个Activity，通过Activity用户可以进行人机交互，Activity之间可以相互跳转；并且每个Activity都有自己的生命周期（lifecycle）：Create—Start—Resume—Pause—Stop—Destory。本系统主要包含了四个Activity：启动界面（SplashActivity）、检索界面（SearchActivity）、检索结果界面（ShowResultActivity）以及展示条目界面（ShowItemActivity）。为了增加用户体验，在SplashActivity中加入了网络连接性的检测和程序自动更新检测；当用户未开启手机网络连接（3G或WiFi）时，程序会友好提示用户进行网络设置；如果程序有新版本时，会提示用户下载安装。之后会进入到检索界面SearchActivity，用户可以输入相应的检索条件。当检索开始时，会发出一个以检索条件为参数的SOAP请求到Web服务端，服务端作出响应并返回馆藏书目结果。这些书目会在ShowResultActivity中以列表的形式进行展示，并且用户可以进行上下页翻阅。同时用户可以点击某条结果记录以获得更多的详细书目信息和借阅信息，如馆藏地、可借状态等，即显示ShowItemActivity界面。部分核心代码如下：

检测网络是否连通：

```
……
State mobile = conMan.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState();
State wifi = conMan.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState();
if(mobile == State.CONNECTED || mobile==State.CONNECTING)
 return true;
if(wifi == State.CONNECTED || wifi == State.CONNECTING)
 return true;
……
```

自动安装更新程序：

```
……
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setDataAndType(Uri.fromFile(new
```

```
File(Environment
                                .getExternalStorageDir
ectory(), "opac.apk")),
                                "application/vnd.
android.package-archive");
                startActivity(intent);
                        ……
```

查询模块：

```
                        ……
                request = new SoapObject(WSDL_TA
RGET_NAMESPACE, METHOD_NAME);
                request.addProperty("keyword", input);
                request.addProperty("searchType", type);
                request.addProperty("page", currentPage);
                envelope = new SoapSerializationEnvelop
e(SoapEnvelope.VER11);
                envelope.dotNet = true;
                envelope.setOutputSoapObject(request);
                httpTransport = new HttpTransportSE(use
dConnection);
    httpTransport.call(SOAP_ACTION, envelope);
                SoapPrimitive response = (SoapPrimitive)
envelope.getResponse();
                        ……
```

（3）Cache机制的引入

众所周知，目前手机上网的流量费用相对来说还比较高，当用户不在WiFi可用范围内，就必须开启移动网络（如GPRS、3G等）来查询书目，这就会产生一定的流量。为了节省流量，使用户花最少的钱享受到全面的服务，本系统创建了一个存放书目历史记录的cache。每次浏览查询结果时，如果返回的结果是用户第一次浏览的，则存入cache中作为历史记录。当用户想浏览以前的记录，可以直接从cache中取出相应的记录，而不需要再次对OPAC服务器发出查询请求。因此cache的引入能很好地为用户节省费用。

3 系统测试

由于目前市面上的Andorid智能手机屏幕大小不一，所以为了使系统可以在不同大小屏幕的手机上运行，Android提供了各种屏幕的支持参数设置[7]：

```
<manifest xmlns:android="http://schemas.android.com/
apk/res/android">
  <supports-screens
    android:smallScreens="true"
    android:normalScreens="true"
    android:largeScreens="true"
    android:xlargeScreens="true"
    android:anyDensity="true" />
  ...
</manifest>
```

所有的Android智能手机都被分为smallScreens、normalScreens、largeScreens和xlargeScreens四种屏幕尺寸。通过设置参数，可以使程序自适应相应的屏幕来显示内容。图2给出了系统的运行界面。

图2 系统运行图

4 结束语

文章分别从两个背景，即Android平台智能手机的发展趋势和移动应用在图书馆的实践现状，提出了图书馆OPAC系统在Android平台智能手机上的应用，并对系统设计和实现进行了论述，最后完成了系统的测试运行（读者可以从http://lib.zstu.edu.cn/opac.apk下载）。在此基础上可以进一步完善，比如增加条码识别和语音识别等功能，使读者使用起来更加方便。随着智能移动终端和移动应用的快速发展，如何将图书馆的各种服务（如读者借阅信息、图书到期提醒、图书预约、信息咨询、手机阅读等）以移动应用的方式呈现给读者，使读者能享受到更人性化的服务，必将成为移动图书馆发展的一个重要方向。

参考文献

[1] Android Leads in U.S. Smartphone Market Share and Data Usage [OL]. [2011-05-16]. http://blog.nielsen.com/nielsenwire/consumer/android-leads-u-s-in-smartphone-market-share-and-data-usage/.
[2] AndroLib [OL]. [2011-05-06]. http://www.androlib.com.
[3] 马爱芳,杨国美.我国高校图书馆手机服务现状的调查与思考——以“211工程”院校为例[J].图书馆工作与研究,2009(12):87-90.
[4] M-Libraries [OL]. [2011-05-06]. http://www.libsuccess.org/index.php?title=M-Libraries.
[5] 姜海峰.移动图书馆的兴起和解决方案[J].大学图书馆学报,2010(6).
[6] Designing for Performance [OL]. [2011-05-06]. http://developer.android.com/guide/practices/design/performance.html.
[7] Supporting Multiple Screens [OL]. [2011-05-06]. http://developer.android.com/guide/practices/screens_support.html.

作者简介

施干卫（1977-），男，计算机硕士，馆员，研究方向：数字图书馆、移动图书馆，浙江理工大学图书馆信息与数字化建设部。Email：gwshi@yeah.net

Design and Implementation of OPAC Based on Smartphone Platform – Taking Android as an Example

Shi Ganwei, Huang Zhiqiang, Lou Xiangying, Liu Xiang / Library of Zhejiang Sci-Tech University, Hangzhou, 310018

Abstract: With the development of information communication technology and smartphone, the mobile application is developing in various territories greatly as well. The paper mainly discusses how to design and implement the Online Public Access Catalog in smartphone based on Android platform as a mobile application.

Keywords: Smartphone, OPAC, Android

（收稿日期：2011-08-26）