

# 基于HTML5的移动Web App开发\*

黄悦深 (五邑大学图书馆)

**摘要** 分析了当前图书馆移动应用开发的两种模式基于WAP的B/S模式以及基于移动操作系统的原生应用模式存在的缺点,探讨了图书馆移动应用开发的发展趋势,提出了基于HTML5的移动Web App开发模式。总结了HTML5中适用于移动应用开发的技术标准。最后,详细分析了一款基于HTML5的图书馆用户教育类的Web App游戏开发过程。

**关键词** HTML5 Web App 移动应用 移动服务

DOI: 10.13663/j.cnki.lj.2014.07.012

## Mobile Web App Development Based on HTML5

Huang Yueshen (WuYi University Library)

**Abstract** The article analyzes the shortcomings of two kinds of library mobile application development models: B/S mobile based on WAP and native application model, discusses the development trend of mobile application development in library, and proposes the mobile Web application development model based on the HTML5. It also summarizes the HTML5 technical standards for mobile application development, as well as detailing how to develop a Web App game for educating library users based on HTML5.

**Key words** HTML5, Web App, Mobile application, Mobile service

从早期的手机短信平台发展至今,移动图书馆已经从辅助性的服务平台发展为承载图书馆泛在化、智能化服务的主要平台。2012年,第四届国际移动图书馆会议上,Ellyssa Kroski 概况了当前及今后移动图书馆建设的十项内容:(1)图书馆网站与移动OPAC;(2)移动馆藏;(3)图书馆指引;(4)移动数据库;(5)移动导引;(6)短信提醒;(7)短信参考咨询;(8)二维码;(9)移动社会网络;(10)其他(如增强现实)<sup>[1]</sup>。从实现形式来划分,这些内容可分为两大类:第一类是已有的图书馆服务向移动终端迁移,包括移动OPAC、移动阅读、移动资讯、移动咨询。第二类是以移动互联技术为特色的新应用。包括二维码应用、App应用、GPS应用、增强现实技术应用等。随着移动应用技术的不断发展以及移动终端功能的日益强大,图书馆的移动应用将朝着更加智能化、趣味化、多元化的方向发展。2012年12月,HTML5规范正式定稿,

成为开放的Web网络平台的奠基石<sup>[1]</sup>。这意味着HTML5不仅是面向传统PC的网页设计,而且是适应所有智能终端的使用需求,为移动互联网时代的网页应用提供更加丰富、强大的开发标准。

### 1 移动应用开发技术的变迁与走向

移动应用开发技术的演变过程,可分为三个阶段:从早期的WAP(包括WAP1.0和WAP2.0)网站到目前流行的原生应用(Native Application),再逐步转向以HTML5标准为基础的Web App。

1998年,遵循WAP1.0协议的无线标记

\* 本文系广东省江门市哲学社会科学项目“江门城市智慧图书馆的设计与构建”(项目编号:JM2013C35)研究成果之一。本文系广东省图书馆学会项目“高校图书馆新生入馆教育互动学习与考核平台构建研究”(项目编号:GDTK1267)研究成果之一。

语言 WML 诞生，成为低功耗手机访问网站的标准设计语言。随着移动终端和互联网的发展，WAP2.0 协议诞生，并出现了 XHTML 标记语言，较之 WML，XHTML 支持所有移动平台，适用于大部分智能移动终端，而且支持 JavaScript、CSS 等交互技术，用户体验良好。随着越来越多的网站推出适合移动设备访问布局及功能的页面，移动设备成为与 PC 同等重要的互联网访问工具。

如果说在此之前，移动应用的发展是在模仿或追随 PC 时代的互联网技术与应用，那么，由苹果公司掀起的原生应用（Native App）开发热潮，标志着移动应用开始超越 PC 时代的互联网应用辉煌。原生应用能够充分发挥移动设备的功能特性，除了呈现丰富的全媒体内容外，还能通过操控硬件带给用户人机互动、地理感知等立体化体验。但是，原生应用与移动设备的操作系统绑定，不能跨平台使用，而且开发难度大、周期长。这些缺点使得原生应用开发模式面临着很大的发展瓶颈。

2008 年，HTML5 标准草案颁布，为移动应用开发提供了另外一种技术方案。2010 年，苹果宣布其移动设备不支持 Flash 插件后，HTML5 标准更是迅速成为移动网站打造丰富的网页应用的首选。为了适应移动互联应用“富内容、富交互、富体验、跨平台”的要求，HTML5 较之 HTML4 做出了大幅的改良。突出表现在：支持离线访问、支持全媒体内容展示、支持交互式的 2D/3D 图像渲染、提供丰富的 API 以调用移动设备的硬件。这使得 HTML5 可以实现堪比 Native App 效果的 Web App，而且，其具有更优越的安全性、跨平台访问性。更重要的是，Web App 开发难度较之 Native App 大幅降低，易用性更高。

## 2 图书馆的移动应用开发之路

技术与服务相融合是移动图书馆的建设之道。随着移动应用开发技术的发展，移动图书馆服务也日趋丰富。中国国家图书馆的“掌上国图”、上海图书馆、浙江大学图书馆、厦门大学图书馆、汇文掌上图书馆等智能手机 App 产品，实现了图书馆服务移动化、泛在化，代

表了当前图书馆移动应用开发的高度。

### 2.1 图书馆移动应用的发展趋势

图书馆移动应用呈现出以下发展趋势：

（1）服务内容从单一向综合演变。从最早的“短信提醒”服务起步，图书馆的移动应用经历了多次技术革新，在不断丰富界面表现形式，提高用户可用性的同时，逐步实现与图书馆集成管理系统、图书馆网站平台、参考咨询系统的集成融合。目前，已提供包括移动 OPAC、移动参考咨询、移动资讯推送、移动借阅管理、移动资源导航在内的综合服务。

（2）应用类型从传统向创新演变。早期的乃至当前的大部分图书馆移动应用，都是对实体图书馆或者数字图书馆业务的复制，即实现传统业务向移动终端迁移。随着以苹果、三星等智能手机的发展与普及，移动设备所带来的全新人机交互与用户体验，以 HTML5、二维码技术、GPS 技术为基础的移动应用正成为图书馆移动应用开发的新方向。

（3）建设主体从单一向多元演变。无论是图书馆移动应用的实施体系还是系统体系，都经历着由简单到复杂的变化。就实施体系而言，早期的移动应用从方案策划到组织实施再到使用维护，一般是由图书馆一手包办。当前的移动应用则涉及包括图书馆、电子资源商、软件开发商在内的多个建设主体的分工与协调，图书馆在其中担当何种角色，采取何种方案参与应用开发则是值得探讨的。

（4）系统结构从简单向复杂演变。早期的图书馆移动应用功能单一、集成度低，系统结构相对简单。当前的图书馆移动应用功能丰富、跨系统集成调用多，系统结构复杂，大致可分为数据资源层、集成接口层、应用实现层。数据资源层是以适合移动设备访问和阅读的文件格式来存储的数据和文档。集成接口层是沟通应用层与数据层的桥梁，通过构建与调用图书馆集成系统、电子资源平台等外部系统的访问接口，实现移动应用对数据资源的访问。应用层是利用 WAP、APP、HTML5 等移动技术构建功能丰富、内容展示美观的用户使用平台。

### 2.2 HTML5：图书馆移动应用开发的新契机

从 HTML4 到 HTML5 的提出，经历了整

整十年的演变, 当中的变化不是小幅的版本升级, 而是质变的飞跃。HTML5 彻底颠覆了 HTML4 以呈现网页文本为主的功能定位, 适应当前移动智能终端的强大功能, 定位于提供一个功能完整、开放易用的网页应用开发平台。基于 HTML5 开发的 Web App, 在不借助第三方插件的情况下, 能够实现丰富的交互操作以及多线程处理, 这使得 Web App 能够完成复杂的操作任务。相比于以往的开发模式, 基于 HTML5 的 Web App 模式是更加理想的移动应用开发方案。

(1) 相比于早期的 WAP2.0 标准以及近年新兴的 Native App 构建方案, HTML5 标准具有明显的优势。WAP2.0 是对 WAP1.0 的改进, 其初衷是针对低能耗、低流量手机提供移动应用开发标准, 虽然 WAP2.0 在功能及表现力上有所改善, 但是, 对于当前主流的智能移动终端的强大性能仍然欠缺支持。Native App 是适应移动智能终端的开发方案, 能够充分发挥设备的性能, 为用户提供诸多新颖有趣的使用体验。但是, Native App 高度绑定于移动操作系统, 存在开发难度大、部署应用不便的缺陷。相对而言, HTML5 是基于 B/S 架构的标准, 不存在 Native App 的固有缺点, 具有良好的跨平台统一性, 而且, HTML5 功能强全面, 为开发者提供了丰富的移动应用开发标准, 弥补了 WAP 标准功能弱小的不足。总体而言, 基于 HTML5 的移动应用开发方案具有开发难度低、功能丰富、部署应用简易、跨平台访问便捷的优势。

(2) HTML5 大幅降低了图书馆移动应用开发的技术门槛。经过 PC 时代的互联网应用开发, 图书馆对 HTML 标准下的应用开发技术与流程已经十分熟悉。HTML5 继续沿用 HTML4 时代的开发部署方案, 相对于全新的 Native App 构建方案, 基于 HTML5 的移动应用开发更容易被图书馆的技术人员接纳。

(3) 基于 HTML5 的 Web App 开发为移动图书馆建设注入活力。2008 年以来, 随着智能移动设备的迅速普及, 国外图书馆逐渐将移动服务的重点转移到移动应用程序开发和移动网站用户体验优化上面<sup>[3]</sup>。但是, 由于此前的移

动应用开发难度大, 图书馆真正参与移动应用开发的程度不高, 移动应用服务基本来自于软件公司的产品。HTML5 标准的颁布有力地改变了这种局面, 图书馆可以多元化地选择移动应用的建设策略。笔者认为, 对于功能多样、体系复杂的移动应用适宜购买, 而对于开发难度较低、个性化程度高的移动应用适宜自建。虽然, 目前自建移动应用的实践不多, 但是, 自建移动应用却是移动图书馆建设的活力所在, 正如 PC 时代各个图书馆在门户网站建设、数字化服务上各出奇招、争奇斗艳, 催生了数字图书馆建设大繁荣一样, 只有大众的、草根的、参与度高的事物, 其生命力才能强大而持久。笔者预期, HTML5 标准将以其易用、开放、强大的特点, 成为图书馆自建或共建 Web App 的通用标准。

### 3 HTML5 的核心功能分析

#### 3.1 2D/3D 图像绘制技术

Canvas 是最强大的 API 之一, 过去通过 FLASH 等插件实现的图书及动画, 现在只需使用 JavaScript 就能够实现。Canvas 对于 Web App 开发意义重大, 也是 HTML5 中最受关注的功能之一。

Canvas 是用于在网页中绘制位图的技术。目前, 它对绘制各种 2D 图形、图表、动画效果提供了良好的技术支持。目前创建 2D 图像的 Canvas API 有: Canvas 状态 API、简单形状的 API, 填充, 画笔, 擦除等等, 这些 API 定义了基本的绘画方法。

在 HTML5 上创建 3D 图像有多种方案。目前, 大致使用三种模型来创建 3D 图像<sup>[4]</sup>: 使用标准低级 API (诸如, OpenGL ES), 这种方法被称之为“WebGL”。使用 DOM tree, 并且在 X3D 场景树之间进行 DOM tree 映射, 这种模式被称之为 X3DOM 模式。O3D 模型。O3D 是 3D 应用程序的一个接口, 由 GOOGLE 研发。

#### 3.2 离线应用技术

离线应用是指在脱离网络连接的环境下, Web App 依然能够进行数据存储与交换业务, 从而保证程序的正常运行。这项技术使得 Web

App 克服了过去离线即瘫痪的局限,应用适应性更强。在开发中,该技术涉及以下三种功能实现: 离线资源缓存。HTML5 通过 Cache Manifest 缓存清单指明 Web App 在离线工作时所需的资源文件,并支持自动和手动两种缓存更新方式。 在线状态检测。HTML5 提供了 HTML5 提供了 Navigator.online 属性值判断和 Online/Offline 事件监控两种检测方式。 本地数据存储。HTML5 提供 Web Storage 和 Web SQL Database 两种存储机制: Web Storage 是在客户端以 key-value 形式保存数据的功能,类似于 HTML4 中的 Cookie,但安全性更好; Web SQL Database 是 HTML5 草案中追加的新功能,它允许用户在客户端通过 JavaScript 操作数据库<sup>[5]</sup>。

### 3.3 设备适配技术

不同类型、系列的移动设备在屏幕尺寸、分辨率、键盘类型、媒体摄录能力等方面存在巨大的差异。为了兼容不同设备的浏览访问,输出合适的页面布局,HTML5 提供了两种设备适配方案。 设备信息: Device Description Repository API, 一个统一的服务器 API, 允许 Web 开发者从众多的设备信息数据库中,获取正在访问他们网站的设备的配置信息数据。 基于 CSS 的适配: CSSMediaQueries, 提供了一套机制允许 Web 页面根据一些设备特性(包括屏幕分辨率)适配布局和行为。 CSSDeviceAdaptation 定义了一套 CSS 指令,参照所持设备的尺寸,指定每个布局所适用的尺寸。

### 3.4 提供丰富的支持Web应用扩展的API

HTML5 提供了操作移动设备硬件的 API, 包括: 地理位置感知 API、运动感应事件、通用感应器 API、触控交互事件、WebSocket API、Messaging API 等。它们为 Web App 调用电话、短信、邮件、摄像头、地理定位、文件存储等终端功能提供了简化的操作接口,这是 HTML4 和 XHTML 无法企及的。

## 4 图书馆移动 Web App 构建实例

用户教育是图书馆的核心服务之一,对于宣传图书馆服务、普及图书馆知识发挥着积极的作用。然而,由于教育方式僵化、内容乏

味,用户参与程度不高,教育效果不佳。为此,需要根据用户的使用喜好,有针对性地设计用户教育内容。针对用户喜欢“富内容、富体验、富交互”的使用需求,笔者设计了一款模拟图书排架的 Web App——“图书排架王”,旨在通过趣味游戏的方式传授图书索书号的相关知识。

### 4.1 设计思路

为增强趣味性,游戏中加入了难度设置、限时通关的元素。读者必须在限定的时间内,使用鼠标移动操作,把“桌上”的全部图书放到“书架”的正确位置上。限时内完成任务者可以获得相应级别的徽章,并可以通过社交网即时与好友分享。

### 4.2 实现过程

“图书排架王”Web App 是一款面向移动终端的互动游戏。移动设备操控简易、屏显面积小的特点,决定了游戏的设计必须遵循简单易用的原则,同时具备娱乐性,不能过于简单而失去了游戏的趣味性和互动性。游戏实现过程分为场景设计、动作设计和情节设计三部分。

(1) 场景是游戏的载体,它的作用是清晰地向读者传达出游戏设计的意图和玩法,以及通过生动有趣的画面吸引读者进入到游戏中。为了生动而真实地模拟图书上架的过程,笔者采用卡通描绘的手法设计了游戏场景,如下图所示。读者需要在倒计时结束前,将书桌上的图书排放到书架的正确位置上。

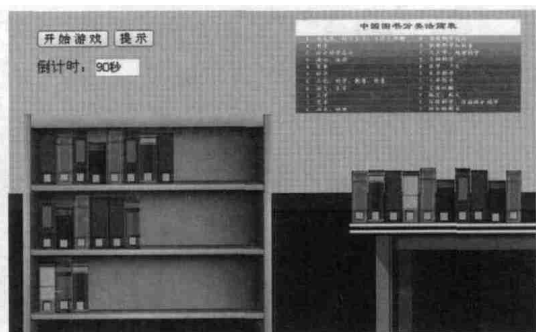


图1 “图书排架王”游戏场景图

(2) 动作是游戏的精髓,它的作用是真实地模拟图书上架的操作过程,给予读者真实的操作体验。笔者将整个图书上架过程细分为选

择、抽取、移动、排序、插入 5 个动作, 每个动作的操控及效果见下表。

表 1 “图书排架王”游戏操控说明

动作名称	动作操控	动作效果
选择	光标移动至图书上	在图书下方显示索取号
抽取	点击图书	图书被抽离书桌
移动	拖动图书	图书随光标移动到任意位置
排序	放置图书到书架上	显示插入位置相邻图书的索取号
插入	释放图书	图书进入书架, 所有后续图书向后移动一位

(3) 情节是游戏的渲染, 它的作用是增强游戏的趣味性和可玩性。笔者设计了三个游戏情节。游戏前, 读者可进行游戏难度设置, 难度越高, 待上架图书的数量和种类越多。游戏中, 设加入了计时控制, 读者必须在倒计时结束前完成图书上架。游戏后, 挑战成功的读者可以获取对应级别的荣誉勋章, 并可以通过社交网站与好友分享。下图是“初哥”级和“达人”级勋章的设计示例。



图 2 “图书排架王”荣誉勋章

#### 4.3 核心功能

在面向 PC 浏览器的页游中, Flash 是实现技术。但是, 由于运行 Flash 需要加载插件, 既降低了通用性又延缓了加载速度, 不适用于开发通用性强、响应快速的移动应用。因此, 笔者采用了基于 HTML5 的 Web App 实现方案。

本游戏中, 核心操控是实现对图书的自由拖放与排序。在 HTML5 中, 拖放页面元素是一项常规功能, 有专用的拖放事件及 API。该功能通过定义拖放对象及一系列拖放事件来实现。拖放对象分为两类: 一是拖动元素对象, 在本例中是图书, 二是拖动元素的接收对象, 在本

例中是书架。拖动元素对象的事件包括 dragstart (拖动开始时) drag (拖动中) dragend (拖动结束时) 拖动元素的接收对象的事件包括 dragenter (拖动元素进入时) dragleave (拖动元素拖出时) dragover (拖动中) drop (拖动元素被放下时), 这些事件能够增强拖放过程的交互性和真实性。排序功能通过调用 JQuery UI 框架中的 Sortable 函数来实现。在本例中, 通过为图书和书架添加各种响应事件, 模拟出图书在选择、抽取、移动、排序、插入每个阶段的动作效果。主要代码如下:

HTML 页面代码:

```
<div id="desk">
    
    
    
</div>
<div id="shelf">
    
    
    
</div>
```

JavaScript 代码:

```
window.addEventListener("load",
function(){
    document.getElementById("shelf").
sortable();

    var books=document. getElementById
("desk").childNodes;
    for (var i=0;i<books.length;i++){
        books[i].addEventListener("dr
agstart",function(evt){
            var e=evt.target;
            evt.dataTransfer.setData
('Text',e.id);
```

```

        evt.stopPropagation();
    }, false);
}

var shelf= document.getElementById
("shelf");
shelf.addEventListener("dragen
ter",function(evt){
    evt.preventDefault();
}, false);
shelf. addEventListener("drago
ver",function(evt){
    evt.preventDefault();
}, false);
shelf. addEventListener("drop",
function(evt){
    var e=evt.target;
    var id=evt.dataTransfer.getData
('Text');
    var target=document.
getElementById(id);
    if(target){
        shelf.appendChild(target);
    }
    evt.preventDefault();
}

```

#### 参考文献

- [1] Ellyssa Kroski. Libraries to Go: Mobile Tech in Libraries[EB/OL]. [2013-05-02]. <http://www.slideshare.net/ellyssa/libraries-to-go-mobile-tech-in-libraries-presentation>.
- [2] W3C. Web Design and Applications[EB/OL]. [2013-05-02]. <http://www.w3.org/standards/>.
- [3] 谢强, 牛现云, 赵娜. 移动数字图书馆服务体系研究[J]. 2013(4): 6-10.
- [4] 彭涛, 杨炼. 基于移动浏览器的HTML5核心技

```

    }, false);
})();

```

HTML 代码的关键点是将需要拖动的图书元素的 draggable 属性为 true。JavaScript 代码分为三个实现功能。第一部分调用 JQuery 的 sortable ( ) 函数实现在 “书架” 中 “图书” 可自由排序。第二部分调用 “图书” 的 dragstart 事件函数, 目的是在开始拖动 “图书” 时记录下 id 号。第三部分调用 “书架” 的 drop 事件函数, 目的是获取放入 “图书” 的 id 号, 然后把对应 id 的 “图书” 元素从书桌上移除, 添加到 “书架” 上。

## 5 结语

技术的洪流滚滚向前, 只有顺势而为, 才能保持进步。在移动互联时代, 图书馆面临的任务更加艰巨了, 因为读者的使用需求更高了, 尤其是年轻用户, 对于富内容、富体验、富交互的全媒体应用需求强烈。所幸的是, 技术的进步也为图书馆带来了更开放、易用的技术支持方案。如何在新的技术环境下, 创造出读者喜闻乐见的服务内容, 让我们的服务在移动互联时代继续保持活力, 这是需要所有图书馆人发挥创意、共同实践的工作。

术的研究及其应用[J]. 广东通信技术, 2012(4): 47-50.

- [5] 王志刚, 王中元, 江友华. HTML5移动开发即学即用[M]. 北京: 电子工业出版社, 2012: 277.

黄悦深 五邑大学图书馆, 馆员。E-mail: huangyueshen@163.com 广东江门 529020

(收稿日期: 2013-12-02)