

基于HTML5的移动Web应用浅析

张成岩 吴 静 仇剑书 严斌峰
中国联通研究院 北京 100032

摘 要 随着移动互联网的快速发展和智能终端的普及, 基于HTML5的移动Web应用已经成为移动互联网应用新的发展趋势。文章从HTML5和移动Web应用结合的角度出发, 从HTML5应用技术、应用开发平台、终端侧运行环境等多个方面进行论述, 并对基于HTML5的移动Web应用发展进行展望。

关键词 移动互联网; HTML5; 智能终端; 应用

引言

目前互联网的发展正处在历史性的转折点, 移动平台和应用正在逐渐超越固定的客户端/服务器模式。随着3G甚至4G/LTE技术的商用化和智能移动终端的大规模应用, 移动互联网正处于迅猛发展当中。而移动应用作为移动互联网的主角, 对最终用户、网络运营商、应用开发商、平台提供商、投资商都有巨大影响, 是推动未来移动互联网发展的主要力量。

为支持丰富多样的互联网应用, 国际标准组织W3C于2007年成立新的HTML工作团队, 2008年发布了HTML5的第一份草案。基于HTML5系列标准的应用凭借其良好的跨平台性, 能很好地解决目前Native操作系统割裂的问题, 将为移动互联网产业链带来全新的商业模式, 为产业链各方带来全新的平等竞争机会, 引导移动互联网产业链的新模式发展。将HTML5的技术优势广泛地运用于移动互联网中, 基于HTML5的跨平台特性, 面向越来越丰富的智能终端提供移动Web应用, 是加速HTML5应用成熟、引导互联网应用产业良性发展的捷径。

1 HTML5应用技术

HTML5应用技术指面向HTML5应用开发的主要关键技术, 包括Offline Application技术、Sever-sent Events技术、Websocket技术、Web Workers技术、HTML5视频和音频技术以及Device API技术。下文将

对上述关键技术进行简单的介绍与分析。

1) Offline Application, 即离线应用。在开发支持离线的Web应用程序时, 要使用以下三个方面的功能。

①离线资源缓存。需要一种方式来指明应用程序离线工作时所需的资源文件。这样, 浏览器才能在在线状态时, 把这些文件缓存到本地。此后, 当用户离线访问应用程序时, 这些资源文件会自动加载, 从而让用户正常使用。HTML5中, 通过Cache Manifest文件指明需要缓存的资源, 并支持自动和手动两种缓存更新方式。

②在线状态检测。开发者需要知道浏览器是否在线, 这样才能够针对在线或离线的状态, 做出对应的处理。在HTML5中, 提供了两种检测当前网络是否在线的方式。

③本地数据存储。离线时需要把数据存储到本地, 以便在线时同步到服务器上。为了满足不同的存储需求, HTML5提供了DOM Storage和Web SQL Database两种存储机制。在应用中可以考虑将用户的一些数据存储在本地, 供离线模式下应用。

2) Sever-sent Events, 即服务器推送技术。客户端使用EventSource对象实现。服务端也有相应的要求, 表现在如下两方面: 一是要求网页的MIME类型为text/event-stream, 这个MIME类型定义了事件框架格式, 二是text/event-stream有一定的格式。主要内容是根据需要在应用中向用户推送通知等信息。

3) Websocket, 也叫套接字技术^[1]。当存在和服务频繁数据交流的时候, 就有了如何进行通信的问题。

正常情况下，浏览器访问Web页面时，一般会向页面所在的Web服务器发送一个HTTP请求；Web服务器识别请求，然后返回响应。大多数情况下，当内容呈现在浏览器页面上时，可能已经没有时效性。如果用户想要获得最新的实时信息，就需要不断地手动刷新页面，这显然不是一个明智的做法。目前实时Web应用的实现方式，大部分是围绕轮询和其他服务器端推送技术展开的，包括轮询、长连接、流解决等方案。

4) Web Workers技术^[2]。可以让一部分JavaScript程序代码运行在客户端操作系统的线程中，而非浏览器的线程中，从而可以实现多线程。Web Workers在JavaScript代码执行复杂运算的过程中起到优化效果。每一个运行在客户端操作系统线程中(非浏览器线程)的JavaScript程序代码被称为一个Worker，分为专职Worker和共享Worker两种。

5) HTML5视频和音频。在HTML5网页中不再需要插件播放音频或视频多媒体文件，只需要加入video和audio标签并使用JavaScript代码开发操作面板，即可实现在网页灵活播放多媒体，同时可以结合Canvas技术增加对多媒体文件的效果与控制操作。

6) Device API^[3]。HTML5异军突起，在Web应用中表现突出，通过调用终端本地上的软硬件(比如摄像头、话筒、GPS、各种传感器等)在移动设备上的Web App能与Native App相媲美。但是这需要一套统一完善的API，Device API工作组就针对HTML5制定了一套API，主要包括以下几点。

① The Media Capture API和HTML Media Capture API都是针对摄像头和话筒硬件，前者是对摄像头和话筒的可编程访问，后者是通过HTML表单交互来调用摄像头和话筒。

② Device Orientation Event提供了对设备物理方向和运动信息的事件处理，主要关注运动和方位传感器，它并不提供对基本传感器数据的访问或者对传感器的控制，而是高层次的封装。

③ Sensor API提供所有已知传感器的访问，并且提

供的是原始的传感器数据。

④ HTML5 Geolocation API^[4]用于将用户当前地理位置信息共享给信任的站点，但这会涉及用户的隐私安全问题，所以当站点需要获取用户的当前地理位置，浏览器会提示用户是“允许”还是“拒绝”。获取地理位置在应用中将会得到非常广泛的使用。

综上所述，HTML5是近十年来Web开发标准中最大的飞跃，和以前的版本不同，HTML5并非仅仅用来表示Web内容，它的新使命是将Web带入一个成熟的应用平台。在基于HTML5的移动应用开发中广泛使用上述关键技术，才能实现移动Web应用达到和Native应用相似的用户体验。

2 HTML5应用开发平台

对于移动互联网应用而言，产业生态环境的主要关注点是开发环境和应用发布；前者是为开发者提供API，后者则使得开发者通过应用商店发布应用。生态环境的控制本质就是平台的控制，平台决定API。在Native APP中，平台由智能终端操作系统厂商牢牢控制着，苹果和谷歌这两个公司都借助自身的平台已经形成了自己强大的生态系统。同时，微软在Windows Phone投资巨大，并已围绕平台创建庞大的生态系统参与竞争，Windows Marketplace在短短10个月内达到了3万应用。

相对于已经成熟的原生态应用平台，基于HTML5 Web App的生态系统正处于萌芽期。由于HTML5的规范还在制定和讨论中，运用Web App开发的过程中仍然会遇到对事件的响应速度及页面渲染等性能问题、不同机型及Web浏览器内核的兼容性问题、对移动设备硬件资源的访问等众多问题；因此，搭建HTML5应用开发平台需要总结目前各大互联网商提供的HTML5应用开发的优劣势，结合对今后移动互联网HTML5的重点业务和应用预测，满足基于HTML5的移动Web应用对HTML5关键特性的需求，以及对移动终端设备能力的调用需求。

另一方面，目前开发者的一大困难就是推出应用的技术障碍高、推出困难用户接受难度大，或者由于技术能力不足而无法提供一些具有优势的服务。搭建HTML5应用开发平台，需要帮助开发者方便、灵活地推出个人应用。例如开发者没有LBS，应用平台可以提供LBS的位置；开发者在页面上想获得页面的数据，原来由于没有插件无法实现，而应用平台可以用插件的模式做深度合作，帮助其解决面临的短板。

综上所述，搭建HTML5应用开发平台的关键就在于实现其技术标准化、接口统一化、流程规范化、提高终端对基于HTML5的移动Web应用的兼容性，更好地为开发者服务。

3 HTML5终端侧运行环境

基于HTML5的移动Web应用期望获得大发展和广大用户的认可离不开智能终端对它的支持。为支持基于HTML5的移动Web应用，需要在终端侧部署Web运行环境。Web运行环境是Web应用在终端上展现及执行的环境，可以是操作系统Web运行环境、浏览器、Widget引擎等。浏览器内核是Web运行环境的核心，可分为渲染引擎和JavaScript引擎两部分：渲染引擎负责HTML、CSS、DOM等网页标记语言的渲染，JavaScript引擎负责JavaScript脚本的逻辑执行。HTML5终端侧运行环境按照实现方式的不同可以分为基于HTML5的终端浏览器和基于HTML5的WebOS两种技术实现模式，下文将分别对两种方式进行介绍与分析。

1) 基于HTML5的终端浏览器技术实现。基于浏览器的方式在目前主流的Native智能终端操作系统上即可实现，只要升级浏览器支持HTML5的相关特性即可。基于智能终端的操作系统浏览器可以是操作系统原生的浏览器(例如Safari、Android原生浏览器等)或者第三方浏览器(例如Ucweb、360浏览器等)。这种技术实现方式通常是在操作系统层嵌入浏览器内核(例如Webkit)，在其基础上扩展Web渲染引擎和JavaScript引擎的功能，并将HTML5特性在引擎中实现，在上层封

装后形成浏览器Web运行引擎，并以Web浏览器形式向应用层开放，如图1所示。

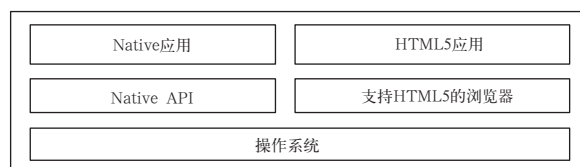


图1 基于HTML5的终端浏览器技术实现架构

2) 基于HTML5的Web OS技术实现。随着HTML5应用的兴起，在目前的Native智能终端操作系统外，也出现了一种新型的、基于HTML5的操作系统——Web OS，其技术架构如图2所示。Web OS完全基于HTML5技术开发，以统一的Web API形式对上层应用开放终端能力，能够避免目前不同Native操作系统提供不同API开发接口的缺陷，能够很好地支持HTML5应用的跨平台运行。采用这种技术架构实现的典型操作系统包括Firefox OS、Tizen及Ubuntu。



图2 基于HTML5的Web OS技术实现架构

综上所述，上述两种方式从技术实现角度来看是类似的，技术实现机制的核心是依托Web应用运行环境为中间层，对下实现对操作系统的适配和终端能力的封装，对上提供Web应用 API以支撑Web应用开发。而这两种实现方式的主要区别为，在为上层应用开发提供Web API的同时，是否还保留原生的应用开发接口和机制，而对于纯粹的Web OS，基于操作系统的应用应该全部是Web应用，不会存在其他类型的应用。

4 展望

从移动互联网技术与产业的发展趋势来看，云计算的迅猛发展，尤其是“平台及服务”(PaaS)的加速部署，开启了低成本、大规模、易运营的云服务时代，而HTML5作为Web云服务天生的前端技术顺势而起，极大增强了Web云服务在移动终端上的展现与交互能力。

随着Web应用在功能、性能及业务体验上开始接近原生应用，Web应用生态必然会在长尾化、轻量级的应用领域挤占属于原生应用的部分份额。

从互联网发展历程来看，Web与原生应用一直是两种并行不悖的互联网服务提供方式，Web应用(B-S模式)侧重提供基本的信息服务，原生应用(C-S模式)侧重提供复杂的功能应用。无论Web应用还是原生应用，都是应用服务前端构建方式的一种，并非完全对立的关系，两者适用不同的应用场景，满足不同的用户需求^[5]。原生应用与Web应用各有优势，原生应用执行效率高、用户体验好，适合实现复杂功能和酷炫效果的用户常用重量级应用；而Web应用顺应了云计算大规模、低成本、易运营的发展趋势，适合长尾化的轻量级服务。两者可以优势互补、并行发展，共同构成未来移动互联网的应用生态系统。

参考文献

- [1] Ian Hickson. The Web Sockets API, W3C Working Draft[EB/OL]. (2009-12-22)[2013-07-14]. <http://www.w3.org/TR/2009/WD-websockets-20091222/>
- [2] Ian Hickson. Web Workers, W3C Working Draft[EB/OL]. (2009-10-29)[2013-07-14]. <http://www.w3.org/TR/2009/WD-workers-20091029/>
- [3] Frederick Hirsch. Recommendation Deliverables[EB/OL]. [2013-07-14]. <http://www.w3.org/2009/dap/>
- [4] Andrei Popescu. Geolocation API Specification W3C Candidate Recommendation 07[EB/OL]. (2010-09)[2013-07-14]. <http://www.w3.org/TR/2010/CR-geolocation-API-20100907/>
- [5] 工信部电信研究院. 新一代移动Web应用技术白皮书(2012年)[R]. 2013

作者简介

张成岩

硕士，主要从事移动业务平台、移动终端业务的研究、设计和开发工作，曾主持十几项有关移动业务、终端方面的研发项目。



吴 静

长期从事移动业务及终端相关技术的研发工作，作为骨干成员参与完成联通研究院3G增值业务试验平台建设、智能手机嵌入式软件平台研发及产业化项目等多个重要项目，在国内外期刊上发表论文10余篇，拥有专利授权7项。



仇剑书

硕士，高级工程师。长期从事电信设备、数据通信设备和软件、电信增值业务产品方面的研究和开发工作，多次成功组织商用产品项目的研发工作，具有丰富的一线开发工作经验，尤其在终端研究和嵌入式软件开发方面具有较深的研究。



严斌峰

博士后，中国联通研究院终端研究与支撑中心主任、长期从事移动增值业务和终端相关的标准工作，现担任中国通信标准化协会无线通信委员会移动业务和应用组副组长。曾参与制定移动多媒体广播/手机电视国家标准的编制工作，积极参加移动增值业务行业标准化工作，向3GPP2、OMA、IETF等国际标准组织提交多篇文稿，出版专著1部，在国内外期刊上发表10余篇论文，并申请了10余项技术发明专利。

(下转80页)