

自适应网页设计中的关键技术

冯兴利 洪丹丹^{*} 罗军锋 锁志海

(西安交通大学 数据与信息中心, 西安 710049)

(* 通信作者电子邮箱 ddhong@xjtu.edu.cn)

摘 要: 针对移动设备上浏览网页时产生的可读性差、人机交互性差、加载速度慢、页面布局不合理、用户体验差等缺点, 提出了基于媒体查询技术和流动布局技术的自适应网页设计方法。媒体查询技术是层叠样式表(CSS3)的关键技术, 能够将页面内容不经修改而直接显示在特定尺寸的设备屏幕上, 而流动布局技术则能够使页面普遍适应各种分辨率不同的设备, 做到网页的普遍自适应。该技术使移动客户端网页布局更加合理, 页面内容具备更好的可读性。经 WebPagetest 测试表明, 自适应网页与非自适应网页相比, 页面加载时间能够明显缩短, 加载速度大幅提高, 更好地提升了用户体验。

关键词: 媒体查询; 流动布局; HTML5; CSS3; 自适应网页

中图分类号: TP393.092 **文献标志码:** A

Key technologies of responsive Web design

FENG Xingli, HONG Dandan^{*}, LUO Junfeng, SUO Zhihai

(Data and Information Center, Xi'an Jiaotong University, Xi'an Shaanxi 710049, China)

Abstract: In order to resolve the disadvantages of low readability, poor human-computer interaction, slow loading, unreasonable page layout and bad user experience when browsing the webpages on mobile devices, the research on key technologies of responsive Web design based on HTML5 (Hypertext Markup Language 5) and CSS3 (Cascading Style Sheet 3): media query and fluid layout was carried out. The results show that the technology make mobile client page layout more reasonable, the page content has better readability, and the page loading speed and the user experience are significantly improved.

Key words: media query; fluid layout; HTML5; CSS3; responsive Web design

0 引言

随着移动通信技术的普及, 越来越多的人使用移动设备上网, 截至 2015 年 6 月, 手机网民规模达 5.94 亿^[1]。而移动设备种类繁多, 屏幕分辨率大小迥异, 比如 iPhone 5S 的分辨率为 1136 × 640 px(像素)、iPad 的分辨率为 1024 × 768 px、MacBook 的分辨率为 2304 × 1440 px 等。因此网络设计师不得不面对一个问题: 如何将同一个网页在不同分辨率的设备上完美地呈现出来。

对于这个问题, 最早的解决办法是为不同的设备提供不同的网站版本, 比如为 iPad 专门提供一个 iPad 版本(分辨率为 1024 × 768), 为 MacBook 专门提供一个 MacBook 版本(分辨率为 2304 × 1440 px)等, 这样做虽然能够保证页面效果, 但却要同时维护多个网站版本, 大大增加了维护成本。于是, 美国网页设计师 Ethan Marcotte 在 2010 年提出了自适应网页设计(Responsive Web Design, RWD)理念, 即可以通过自动识别屏幕分辨率而做出相应布局调整的网页设计方式^[2]。自适应网页设计不仅能够响应多个有限视窗, 也能使网页在尽可能短的时间内快速加载。

媒体查询(Media Query)、流动布局(Fluid Layout)是自适应网站设计的两个关键技术。本文使用超文本标记语言(Hypertext Markup Language 5, HTML5)和层叠样式表

(Cascading Style Sheet 3, CSS3)对这两个关键技术作了相应研究。相比旧版本 HTML4, HTML5 的优势在于精简标记语言, 更节省时间、语义性更强、代码量更少^[3]; 而 CSS3 更灵活、更丰富, 可以更快地得到以往用很多插件才能得到的效果, 用元素来替代图片, 能够大幅度提升网页加载速度, 减少开发与维护成本, 提高页面性能^[4]。

本文中使用的 Google Chrome 携带的扩展程序 Window Resizer 模拟不同尺寸的屏幕分辨率。Window Resizer 能够模拟的尺寸有 10 种, 分别是: 480 × 800 px、640 × 960 px、768 × 1280 px、1024 × 768 px、1366 × 768 px、1280 × 800 px、1280 × 1024 px 以及 1680 × 1050 px^[5]。

1 媒体查询技术

1.1 技术简介

媒体查询是 CSS3 的一个关键技术, 用来为同一个页面的不同视窗提供支持, 同时允许用户针对设备的显示性能选择特定的 CSS 样式。例如, 通过简短几行 CSS 代码, 用户可以根据视窗宽度、屏幕高宽比或屏幕方向(横向或纵向)来改变页面内容的显示方式。通过使用媒体查询技术, 页面内容可以不经过修改而“定制”显示在输出设备的特定范围内。如果不使用媒体查询技术, 开发人员将不能针对设备的特定性能(如视窗宽度)选择特定的 CSS 样式, 所以媒体查询技术使

收稿日期: 2015-11-23; 修回日期: 2016-01-07。 基金项目: 教育部科技发展中心专项(2011jybki01)。

作者简介: 冯兴利(1985—), 女, 山东枣庄人, 工程师, 硕士, 主要研究方向: 应用系统、高校信息化; 洪丹丹(1981—), 女(满族), 辽宁丹东人, 工程师, 硕士, 主要研究方向: 移动互联网、高校信息化; 罗军锋(1976—), 男, 陕西澄城人, 工程师, 硕士, 主要研究方向: 数据挖掘、高校信息化; 锁志海(1971—), 男(回族), 上海人, 研究员, 硕士, 主要研究方向: 数据挖掘、高校信息化。

自适应网页设计成为可能^[6]。

1.2 技术探讨: 使用 HTML5 充分利用可用空间

对于同一个页面内容,当设备宽度较小时会显得网页拥挤;而设备宽度较大时会显得页面空旷,网页剩余空间多,不能得到充分利用,用户体验不好。如果能够在浏览器宽度变化时根据需要显示部分或全部内容,即页面宽度较小时仅显示其中一部分(如文字),页面宽度比较大时可将所有信息(如文字、注释和图片等)一起显示,可大大增强用户体验。具体代码及实例如下:

```
<ul id="nav">
  <li><a data-email="commentsxhw@126.com" href="mailto:commentsxhw@126.com">Xinhua</a></li>
  <li><a data-email="webmaster@chinaedunet.com" href="mailto:webmaster@chinaedunet.com">China Education</a></li>
  <li><a data-email="kf@peopledaily.com.cn" href="mailto:kf@peopledaily.com.cn">People Daily</a></li>
</ul>
```

在 HTML5 中,带有 data-前缀的属性可以用来存储信息^[7],例子中 data-email 属性用来存储 Email 账户,该属性在后续会用到。该列表相应的 CSS 样式如下:

```
#sidebar ul li a {
  color: #F00;
  text-decoration: none;
  padding: 3px 0;
  display: block;
}
```

当页面宽度小于或等于 640 px 时,页面效果如图 1 所示。



图 1 浏览器宽度为 640 px 时的页面一效果

当浏览器宽度更大(641 ~ 768 px)时,页面可利用空间更大,可以在每个邮箱账户前加上 Email 图标。具体代码如下:

```
@media all and (max-width: 768px) and (min-width: 641px) {
  #sidebar ul li a: before {
    content: "Email: ";
    font-style: italic;
    color: #00f;
  }
}
```

页面效果如图 2 所示。



图 2 浏览器宽度为 768 px 时的页面一效果

当浏览器宽度更大(769 ~ 1024 px)时,可直接将邮箱地址加到每个账户后面,此处就用到了上述定义的 data-email 属性,代码如下:

```
@media all and (max-width: 1024px) and (min-width: 769px) {
  #sidebar ul li a: after {
    content: " (" attr(data-email) ") ";
    font-size: 11px;
    font-style: italic;
    color: #0c6a15;
  }
}
```

页面效果如图 3 所示。



图 3 浏览器宽度为 1024 px 时的页面一效果

当浏览器宽度达到 1025 px 以上时,可将 email 图标加到每列之前,将所有信息都显示出来。具体代码如下:

```
@media all and (min-width: 1025px) {
  #sidebar ul li a {
    padding-left: 21px;
    background: url(email.png) left center no-repeat;
  }
}
```

页面效果如图 4 所示。



图 4 浏览器宽度为 1366 px 时的页面一效果

此外,屏幕分辨率调整前后,页面运行效果从一个状态瞬间转换到另一种状态,速度之快通常令人无法通过肉眼分辨页面的细微变化,可通过使用 CSS3 transition 技术使页面在某个时间段内从一种效果平滑过渡到另一种效果,同时将页面内容的缩放变化逐渐显示出来,提高了用户体验^[8]。具体代码如下:

```
body, img {
  transition: all 3s linear;
  -o-transition: all 3s linear;
  -moz-transition: all 3s linear;
  -webkit-transition: all 3s linear;
}
```

该段代码的加入,使用户在缩小浏览器宽度时,能在 0.3 s 时间内看到页面上的元素逐渐缩放,而不是瞬间缩放。

2 流动布局技术

2.1 技术简介

对某个特定的移动设备(如 iPhone 4S),仅仅使用媒体查询技术能做到自适应网页设计。但是这种方式有一个严重的缺点:不能普遍适应各种分辨率不同的设备。此时,仅有媒体

查询是不够的,需要一种适应性更强的设计方式。要做到这一点,就必须摒弃刚性的、固定的布局,使用流动布局^[9]。

2.2 技术探讨: 通过上下文感知调整图片尺寸

对于移动客户端来说,如果加载的图片尺寸(如宽度 1 000 px)过大,无法在屏幕范围内显示,若将其按照屏幕宽度缩小(如宽度缩小到 200 px),通常会导致页面运行速度慢,用户体验差。所以,为不同尺寸的浏览器提供不同尺寸的图片很有必要,即对于屏幕较小的设备,显示尺寸较小的图片;而对于屏幕尺寸较大的设备,相应地显示大尺寸的图片^[10]。

此处我们使用的解决办法是:在 php 开发环境下,采用 js 脚本提供基于屏幕尺寸的图片。代码如下:

```
$images[flower][100] = 'img_1280x960.jpg';
$images[flower][75] = 'img_960x720.jpg';
$images[flower][50] = 'img_640x480.jpg';
$images[flower][25] = 'img_320x240.jpg';
```

上述代码定义了一个二维数组,用于存放图片文件名及相对缩放比例。将 1 280 × 960 的尺寸定义为 100%。相应的 640 × 480 尺寸为 50%。在 getimage.php 文件中,将图片及相对比例作为需求参数传给函数 getImageScale(\$img, \$scale),通过与数组元素比较,返回第一个比该缩放比例大的图片文件并使用函数 displayImage(\$fname) 将其显示在指定范围内,即 getimage.php?id=flower&scale=47 将返回相对比例为 50 的图片(此处为 img_640x480.jpg)。在.html 文件中, img 标签的 src 属性值设置为 src="getimage.php?id=flower",同时在 CSS 中设置 img { max-width: 100%; } 保证图片大小不会超出浏览器范围。此外,使用 js 代码计算图片缩放比例对应的图片尺寸,计算结果附加在 img 的 src 属性后作为参数 scale 的值,如 getimage.php?id=puppy&scale=84。相关 js 代码如下^[11]:

```
var scale = Math.floor((( $(container).width() ) / containerMaxWidth) * 100);
$( '.scaleable-image' ).each(function() {
    var src = $(this).attr('src').concat('&scale=', scale);
    $(this).attr('src', src);
});
```

这样一来,服务器端返回的是与 image 缩放比例相应的图片,所以移动设备的浏览器显示的图片要比桌面浏览器的小。页面效果如图 5、图 6 所示。

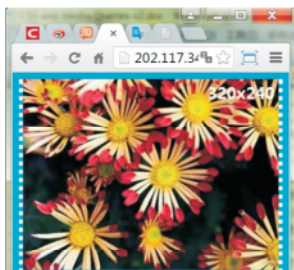


图 5 浏览器宽度为 480 px 时的页面二效果

2.3 页面加载速度测试

为了进一步测试网页性能,本文对自适应网页及相应的非自适应网页的页面响应时间进行对比分析。

页面响应时间决定着用户对网站的体验、在网页上的停留时间以及对页面的满意程度。8 秒原则是互联网的一个著名的原则。用户在访问 Web 网页时,如果时间超过 8 秒就会感到不耐烦,如果加载需要太长时间,他们就会放弃访问^[12]。

2006 年,Amazon 曾做过一个调查,页面响应时间每减少 0.1 秒,他们便会增加 1% 的收入^[13]。据统计,网页反应延迟 1 秒会导致访问量下降 7%,如果一家电子商务网站日收益为 10 万美金,那么延迟一秒可能会带来每年 250 万美金的损失^[14]。这些足以表明缩短页面加载时间的重要性。

为了测试页面响应时间,文中使用网站性能测试软件 WebPagetest 对页面首次被访问时的响应时间进行测试,并选取 iPhone 6 (ios9 操作系统) 模拟移动客户端浏览器,对实例中自适应网页和相应的非自适应网页进行测试并对比分析。WebPagetest 原本是由 AOL 开发的供内部使用的工具,后来在 Google Code 上开源,是一款非常优秀的网页前端性能测试工具^[15]。

测试方法如下:在浏览器地址栏输入 WebPagetest 官方网站: <http://www.webpagetest.org/>。通过在打开页面中输入待测试网页地址、选择测试地点和浏览器类别即可开始进行测试。本文选择的测试地点为移动设备专用地点: Dulles, VA USA (Android, iOS 9),选取的浏览器类别为 iPhone 6 iOS 9。通过分别输入非自适应网页和相应的自适应网页的地址,得到如图 7 和图 8 所示的测试结果。

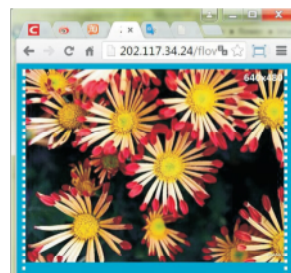


图 6 浏览器宽度为 640 px 时的页面二效果

Document Complete			Fully Loaded		
Time	Requests	Bytes In	Time	Requests	Bytes In
3.324s	2	299 KB	3.324s	2	299 KB

图 7 非自适应页面的加载时间

Document Complete			Fully Loaded		
Time	Requests	Bytes In	Time	Requests	Bytes In
2.304s	4	78 KB	2.302s	4	78 KB

图 8 自适应页面的加载时间

从图中可以发现,用户在 iPhone 6 手机端访问非自适应页面时,加载整个页面耗费的时间为 3.324 s,访问自适应页面所用的时间为 2.302 s,前者耗费的时长比后者长约 1.02 s,依据统计结果,与自适应页面相比,非自适应页面将可能丢失 7% 的用户访问量。

3 结语

本文使用 CSS3 和 HTML5 结合实例对自适应网页中的媒体查询技术和流动布局技术进行了探讨。首先,对于 CSS3 的媒体查询技术,利用 HTML5 的 data-属性设计开发了含有文字和图像的网页,实例表明该技术能够灵活加载网页内容,充分利用网页空间,增强页面内容的可读性;使用 CSS3 的 transition 属性对该网页内容过渡效果进行设置,使得用户能够清晰地看到屏幕分辨率变化时网页内容的细微变化过程,增强了用户体验;其次,研究并探讨流动布局技术,通过上下

(下转第 256 页)

同时,任务产生模块除了产生正常任务实现实时数据的采集功能,还产生校验任务实现回溯机制以保证历史数据的完整性和精准性。

任务处理子系统架构设计体现了可重用性及扩展灵活性。预析取模块的三种预析取方法和数据析取模块的六种析取方法的分别统一接口设计,有利于模块重用。正因为模块可重用性高,各个模块之间的独立性强,当进行数据抓取方法扩展时,只需在已有模块中设计并添加新的抓取方法,即达到“即插即用”的效果。

5 结语

本文提出的信牌驱动式 Web 数据采集模型有效地解决了数据源复杂、实时性强、准确性高和数据类型多样的 Web 空间环境数据采集任务。模型的有效应用——SEDGSS 设计与实现不仅满足了我国空间环境数据采集的功能性需求,而且具有易配置性、可重用性和扩展灵活性,同时实现了自动化运行机制和回溯校验机制。基于该模型设计而实现的 SEDGSS 运行良好,并成功服务于我国空间环境预报,为航天领域从事科研与应用工作的科研工作者以及从事空间技术研究的科研工作者提供了很好的帮助。我们将进一步完善此模型的应用研究,并重点探索分布式架构的扩展实现负载均衡以优化系统性能。

参考文献:

- [1] 叶宗海,都亨,龚建村. 中国的空间环境研究与空间环境预报[J]. 地球物理学进展, 1999, 14(S1): 20-29.
- [2] 齐鹏,李隐峰,宋玉伟. 基于 Python 的 Web 数据采集技术[J]. 电子科技, 2012, 25(11): 118-120.
- [3] CALIFF M E, MOONEY R J. Relational learning of pattern-match rules for information extraction[C]// AAAI99/IAAI99: Proceed-

ings of the Sixteenth National Conference on Artificial Intelligence. Menlo Park: American Association for Artificial Intelligence, 1999: 328-334.

- [4] KUSHMERICK N. Wrapper induction: Efficiency and expressiveness[J]. Artificial Intelligence, 2000, 118(1): 15-68.
- [5] BAUMGARTNER R, FLESCA S, GOTTLOB G. Visual Web information extraction with lixto[C]// VLDB01: Proceedings of the 27th International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers, 2001: 119-128.
- [6] BASAK O, ALBAYRAK Y E. Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems[J]. Computers & Industrial Engineering, 2014, 86: 116-126.
- [7] DENARO G, PEZZE M. Petri nets and software engineering[C]// Lectures on Concurrency and Petri Nets, LNCS 3098. Heidelberg: Springer Berlin, 2004: 439-466.
- [8] TOSIC M, MANIC M. A RESTful technique for collaborative learning content transclusion by Wiki-style mashups[C]// Proceedings of the 2011 5th IEEE International Conference on E-Learning in Industrial Electronics. Piscataway: IEEE, 2011: 38-43.
- [9] GLEZ-PEÑA D, LOURENÇO A, LÓPEZ-FERNÁNDEZ H, et al. Web scraping technologies in an API world[J]. Briefings in bioinformatics, 2014, 15(5): 788-797.
- [10] HE W. System and method for synchronized Web scraping: U. S. Patent 20140351091[P]. 2014-11-27.
- [11] BONIFACIO C, BARCHYN T E, HUGENHOLTZ C H, et al. CCDST: a free Canadian climate data scraping tool[J]. Computers & Geosciences, 2015, 75: 13-16.
- [12] 岳晓丽,杨斌,郝克刚. 信牌驱动式 workflow 计算模型[J]. 计算机研究与发展, 2000, 37(12): 1513-1519.

(上接第251页)

文感知屏幕分辨率来加载相应尺寸的图片,而不是将仅有的一个图形缩放到适合屏幕的尺寸,实例表明该技术极大增强了用户上网体验。通过使用 WebPagetest 进行性能测试,结果表明自适应网页与非自适应网页相比,页面加载时间明显缩短,页面加载速度大大提高。

参考文献:

- [1] CNNIC. 第36次中国互联网络发展状况统计报告[EB/OL]. [2015-07-23]. http://news.xinhuanet.com/zgjx/2015-07/23/c_134438782.htm.
- [2] MARCOTTE E. Responsive Web design [EB/OL]. (2010-05-25) [2014-07-14]. <http://alistapart.com/article/responsive-web-design/>.
- [3] KNIGHT K. Responsive Web Design: What it is and how to use it [EB/OL]. (2011-01-12) [2014-07-12]. <http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>.
- [4] HTML5 & CSS3 的新交互特性 [EB/OL]. [2011-11-29]. <http://www.csdn.net/article/2011-11-29/308243>.
- [5] Window Resizer [EB/OL]. [2015-11-01]. <http://www.winresizer.com/>.
- [6] Responsive Webdesign from the future [R/OL]. [2011-06-12]. <http://warpspire.com/talks/responsive/responsive.pdf>.

- [7] BEN F. Responsive Web design with HTML5 and CSS3 [M]. Birmingham: Packt Publishing Ltd, 2012: 50-68.
- [8] BENJAMIN L. HTML5 and CSS3 responsive Web design cookbook [M]. Birmingham: Packt Publishing Ltd, 2013: 89-96.
- [9] 大漠. 图解 CSS3 核心技术与案例实战[M]. 北京: 机械工业出版社, 2014: 49-70.
- [10] 毕剑,刘晓艳,张禹. 使用响应式网页设计构建图书馆门户网站——以云南大学图书馆为例[J]. 现代图书情报技术, 2015(2): 97-101.
- [11] 张树明. 基于响应式 Web 设计的网页模板的设计与实现[J]. 计算机与现代化, 2013(6): 125-127.
- [12] 8 秒原则 [EB/OL]. [2015-11-01]. http://baike.baidu.com/link?url=fqTb00Byf9ghd-zaR30Bet_dxD8HpcNgyell7kpUqJBKQ0LY0LsMPMc2B9xOXxxfMzFH_GRfgU43I9MbX0pawo7YF5LffmnsDOP3R_fne.
- [13] 度量 Web 性能的关键指标 [EB/OL]. [2015-11-01]. <http://www.csdn.net/article/2013-11-09/2817556>.
- [14] 图说: 网页加载时间如何影响网站收益? [EB/OL]. [2015-11-01]. <http://www.itongji.cn/article/09142K92013.html>.
- [15] GUBE J. 开发者必备: 测试网站速度的五个免费在线工具 [EB/OL]. [2015-11-01]. <http://www.csdn.net/article/2013-11-04/2817380-website-speed-testing-tools>.