

Adapting to Web Standards
CSS and Ajax for Big Sites

基于 Web 标准的网站 构建与经典案例分析

Christopher Schmitt
(美) Kimberly Blessing
Rob Cherny

等著 叶俊译

- ★★★★★ CSS+Ajax的Web建站标准
- ★★★★★ 以AOL.com为代表的大型网站分析
- ★★★★★ 资深的获奖Web设计师主笔

New
Riders



清华大学出版社

基于 Web 标准的网站构建 与经典案例分析

Christopher Schmitt
(美) Kimberly Blessing 等著
Rob Cherny
叶俊 译

清华大学出版社

北 京

Authorized translation from the English language edition, entitled *Adapting To Web Standards: CSS and Ajax for Big Sites*, 978-0-321-50182-0 by Christopher Schmitt, Kimberly Blessing, Rob Cherny, Meryl K. Evans, Kevin Lawver, and Mark Trammell, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2008.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by TSINGHUA UNIVERSITY PRESS, Copyright © 2008.

北京市版权局著作权合同登记号 图字: 01-2007-5038

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

基于 Web 标准的网站构建与经典案例分析/(美)施米特(Schmitt, C.)等著; 叶俊 译. —北京: 清华大学出版社, 2008.9

书名原文: *Adapting To Web Standards: CSS and Ajax for Big Sites*

ISBN 978-7-302-18446-1

I. 基… II. ①施… ②叶… III. ①主页制作—软件工具, CSS②计算机网络—程序设计 IV. TP393.09
中国版本图书馆 CIP 数据核字(2008)第 131018 号

责任编辑: 王 军 郑雪梅

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 16.25 字 数: 337 千字

版 次: 2008 年 9 月第 1 版 印 次: 2008 年 9 月第 1 次印刷

印 数: 1~4000

定 价: 38.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 026662-01

译者序

什么是 Web 标准？Web 标准并不是某一个标准，而是一系列标准的集合。我们知道，网页主要由三部分组成：结构(Structure)、表现(Presentation)和行为(Behavior)。因此 Web 标准也分三方面：其中结构化标准语言主要包括 XHTML 和 XML，表现标准语言主要包括 CSS，行为标准则主要包括对象模型和 ECMAScript 等。符合 Web 标准的网站可以做到在不同的浏览器中拥有基本一致的表现，网站的加载速度更快、而且更容易维护等等许多优点。

本书的作者 **Christopher Schmitt** 是一名一流的网站设计者，他曾带领着团队在 2000 年的“Cool Site in a Day”竞赛中在 8 小时内为一个非赢利组织制作了一个功能齐全、设计完美的网站，并因此一举夺魁；他撰写的 *CSS Cookbook* 被评为 2006 年最佳 Web 设计类图书。作者丰富的开发经验和写作经验是本书质量的保证。本书的第一部分详细阐述了 Web 标准的具体内容，第二部分则以 EverythingTori 和 AOL 为例介绍了基于 Web 标准的大型网站构建过程。相信本书对那些希望转向基于 Web 标准的设计和开发的网站设计师和开发人员会有很大的帮助，能够帮他们极大地缩短痛苦的转变过程，迅速适应 Web 标准，开发出(或将现有网站重构为)基于 Web 标准的网站。

本书的英文原版并不好读，作为英语母语作者撰写的英文技术书籍，原书中存在太多非常随意的语句，让人十分费解。作为中文译者，我的确在竭尽所能地试图拉近作者与中文读者之间的距离，可惜开发经验和表达能力有限，我想翻译过程中肯定存在一些错误，希望读者发现后不吝赐教，我将在我的博客(blog.csdn.net/yejun8214)中维护一张本书的翻译勘误表。

考虑到术语的中文翻译目前基本没有标准可以依循，以及由此可能造成的术语翻译的不一致性，本文采用了两种方式处理：第一是当某术语在本书中第一次出现时用括号给出英文原词，第二种方式则是直接保留原文(针对那些翻译成中文可能造成词不达意或者词性混淆的情况)。如果对读者造成阅读困难，敬请谅解，但译者实在主张所有术语都尽量保留原文。另外，考虑到中文有时在词性上非常模糊(比如第 5 章 Cycle of Standards，被翻译为“标准循环”，但其中“标准”一词的词性就很模糊)，或者不同的英文术语却对应相同中文的情况(如 term 和 project 都对应“项目”，但意义相差极大)，这都可能导致读者的费解，对于这两种情况，译者也尽量做到每次出现都用括号给出原英文术语。

阅读本书需要读者有较为丰富的 Web 开发相关经验，因为书中给出的往往是教条性的指导原则，非常抽象，没有开发经验的读者可能无法体会作者的高度和妙处，反而打击了学习的自信心和积极性。

最后，祝大家阅读愉快！

关于作者

Christopher Schmitt 是 Heatvision.com 有限公司的创始人，该公司位于俄亥俄州的辛辛那提，是一家新成立的小型媒体出版和设计公司。

作为一名一流的网站设计者，Christopher 从 1993 年开始就一直从事 Web 设计方面的工作。20 世纪 90 年代中期，当他还在佛罗里达州州立大学攻读图形设计方向的美术学位时，就曾为 David Siegel 和 Lynda Weinman 做实习生。他在大学毕业之后继续深造，获得了佛罗里达州州立大学通信学院的交互通信和新通信技术硕士学位。

2000 年，他带领着一个团队在 Cool Site in a Day 竞赛中一举夺魁。在该比赛中他与其他 5 名杰出的队员在 8 小时内为一个非赢利组织制作了一个功能齐全、设计完美的网站。

他是 *CSS Cookbook* 的作者，该书被评为 2006 年最佳 Web 设计类图书，他还是第一部关注 CSS 设计的书——*Designing CSS Web Pages* (New Riders) 的作者。另外，他还是 *Professional CSS* (Wrox)、*Photoshop in 10 Steps or Less* (Wiley) 及 *Dreamweaver Design Projects* (glasshaus) 等几部书的合著者。他还参与编写了 *XML, HTML, XHTML Magic* (New Riders) 的其中 4 章。此外，Christopher 还为 New Architect Magazine、A List Apart、Digital Web 和 Web Reference 撰稿。

在很多会议上，例如 Train the Trainer、Web Visions 和 SXSW 等，Christopher 都充分展示了可访问的、基于标准的设计的用途和好处。他也是 Babble(www.babblelist.com) 网站列表版的版主，Babble 是一个致力于高级 Web 设计和开发主题的邮件列表社区。

Christopher 在其个人网站上(www.christopher.org)展示了真实的自己和他最近的一些活动。他高 6 尺 7 英寸，尽管篮球打得并不是很专业，但却下得一手好国际象棋。

Kimberly Blessing 是一名计算机科学家、技术领袖，同时还是 Web 标准的传播者。在 PayPal 公司，她领导了一个 Web 开发平台小组，该小组致力于通过培训来促进 Web 标准的制定和推广。她还和其他人共同领导了一个倡导浏览器厂商以及开发者遵循并使用 Web 标准的草根机构——Web 标准项目(The Web Standards Project)。Kimberly 毕业于布林茅尔学院(Bryn Mawr College)的计算机科学系，她一直热衷于提高科技领域中女性工作者的人数。她的个人网站是 www.kimberlyblessing.com。

IV 基于 Web 标准的网站构建与经典案例分析

Rob Cherny 是华盛顿特区一家 Web 用户体验咨询公司——NavigationArts 公司的首席 Web 设计师。他拥有 11 年从事网站设计、内容管理以及基于 Web 的应用等方面工作的经验，尤其在协助设计小组和技术小组沟通方面经验丰富。

Rob 为很多领域的客户提供了基于 Web 的解决方案，包括 Sallie Mae、Sunrise Senior Living、美国红十字会、Discovery、Weatherbug、Marriott、Freddie Mac、GEICO、美国卫生和福利部以及美国国务院等。

他和他的妻子、两条狗以及三只猫住在马萨诸塞州波士顿的郊区。他通常不会过多地呆在电脑前工作，而是喜欢和他的妻子以及爱犬一起欣赏电影或者远足。他定期在他的个人网站 www.cherny.com 上发表有关 Web 设计和开发的博客。

Rob 拥有位于马里兰的陶森州立大学的美术史学士学位。

Meryl K. Evans 是一位内容专家，并且是 *Brilliant Outlook Pocketbook* 一书的作者。她写过很多文章并且编辑过许多关于 Web 设计、商业以及写作方面的书籍。Meryl 还为很多出版社和网站等撰稿以及编辑，例如 The Dallas Morning News、Digital Web Magazine、MarketingProfs.com、PC Today、O'Reilly、Pearson、Penguin、Sams、Wiley 及 WROX 等。您可以通过她的个人网站 www.meryl.net 了解到一个纯正的德克萨斯人。

Kevin Lawver 已经为 AOL 工作了 12 年，主要创建各种 Web 应用。他热衷于倡导基于标准的开发，目前是 AOL 在 W3C 咨询委员会的代表，并且是 CSS Working Group 的成员。他大部分时间用来编程、写博客、推广 Web 标准，以及在会议上发表有关 Web 标准、mashups、最佳实践以及 Ruby on Rails 等的演讲。您可以在 Kevin 的个人网站上查看他的相关资料：<http://lawver.net>。

Mark Trammell 从 1995 年开始就一直致力于将功能和可用性作为 Web 设计的标准。Mark 是 PayPal 公司的 Web 标准推广员，并且设计了佛罗里达大学的网站。他在佛罗里达大学任职期间领导了一个广受赞誉的项目：基于 Web 标准的 www.ufl.edu 网站的重建。要下载 Mark 采访 Jimmy Byrum——负责设计雅虎网站 (Yahoo! 具体网址为 www.yahoo.com) 主页的 Web 设计师的视频，只要在 www.webstandardsbook.com 网站上在线注册本书即可。

目 录

第一部分 构建基于 Web 标准的网站

第 1 章	编写前端代码	9
1.1	从哪里开始	10
1.2	文档结构：选择合适的标记语言	11
1.2.1	HTML 与 XHTML	11
1.2.2	DOCTYPE 的切换以及浏览器的渲染模式	17
1.2.3	是否验证标记	27
1.2.4	内容和结构：设计执行	29
第 2 章	表现层叠样式表	39
2.1	究竟需要多少 CSS 文件	40
2.2	微格式及其引入的命名规范、语义和工具	47
2.3	太多类了	52
2.3.1	典型的 classitis	52
2.3.2	治愈 classitis	54
2.4	CSS 文件中内容的结构	57
2.5	可选的媒体 CSS	60
2.6	表现的自由	65
第 3 章	集成表现层	67
3.1	现代 Ajax 方法	68
3.2	JavaScript 需要的文件和功能列表	71
3.2.1	糟糕的脚本	72
3.2.2	非侵入性的改进	76
3.2.3	弹出窗口	79
3.2.4	动态元素以及 innerHTML	82
3.3	在 CSS 和表现层中 JavaScript 的行为	84
3.4	自定义 JavaScript 与框架	89
3.4.1	jQuery 框架代码示例	90
3.4.2	框架使得 Ajax 非常简单	94

3.4.3 框架总结	96
第 4 章 开发 Web 软件应用	99
4.1 Web 应用在过去遇到了困难	100
4.2 指导方针、规则和 Web 标准	102
4.2.1 编写代码的规则	102
4.2.2 现代标记可以更好地生成表单	103
4.2.3 服务器端框架和模板工具	106
4.3 微软的 ASP.NET 框架	110
4.3.1 ASP.NET 数据输出	112
4.3.2 ASP.NET HTML 控件、Web 控件以及其他	119
4.4 内容管理	123
4.4.1 基线内容管理	124
4.4.2 内容管理和简洁的内容	124
4.4.3 内容管理输出以及模块	125
4.4.4 内容管理模板	126
4.4.5 让内容作者所见即所得	129
4.4.6 第三方	131
4.5 如何开发 Web 应用	132
第 5 章 标准循环	133
5.1 团队中的陋习	134
5.2 引入标准循环	135
5.2.1 标准经理	136
5.2.2 标准的制定和文档化	137
5.2.3 培训和交流	140
5.2.4 质量评审流程	141
5.3 行动起来	143
5.3.1 保持动力	143
5.3.2 小结	144

第二部分 案例分析

第 6 章 EverythingTori.com	149
6.1 后台	151
6.1.1 深入到 Tori Amos 的世界中	151
6.1.2 开始设计过程	152

6.1.3	构造线框图	152
6.1.4	设计网站	158
6.1.5	CSS 幕后	162
6.2	启动该网站	171
6.3	与设计师 Philip Fierlinger 面对面	172
6.4	终曲	175
第 7 章	AOL.com	177
7.1	如何使您的团队避免失败、获得成功	179
7.2	为性能而设计	198
7.2.1	在写任何代码之前就估计性能	199
7.2.2	性能关注点	202
7.2.3	采访 David Artz	207
7.2.4	可重复的步骤	209
7.3	系统设计和架构	209
7.3.1	二人同行制	210
7.3.2	创建代理	210
7.3.3	考虑一下工作流	211
7.4	前端向导	213
7.4.1	使您的标记符合 DOCTYPE	213
7.4.2	CSS 最佳实践	216
7.4.3	可访问的 CSS	219
7.4.4	在现实世界中的性能	225
7.5	小结	227
附录 A	定位 Web 浏览器	229
附录 B	可访问性	235
附录 C	提高 Web 站点性能的技巧	237
附录 D	CSS 选择器的参考资料	245
后 记		249

第一部分

构建基于 Web 标准的网站

引言

网站建设已经发生了根本性的变革，然而却没有人把这一点告诉架构师。

Web 浏览器对现代技术的支持给网站前端(front end)的设计带来了全新的规范和控制。这些新的最佳实践也就是通常所说的“基于 Web 标准的”设计或者开发。

采用基于 Web 标准的方法益处多多。Web 发展得越大或者越复杂，Web 标准就会变得越重要。尤其是对于那些需要考虑所有权、渠道以及品牌等许多不同问题的大公司而言，Web 标准显得尤为重要。此外，严格基于 Web 标准的应用和内容管理具有非常良好的前景，并且它已经成为确保一个网站的每个层次都具有高品质的必要条件。

了解标准还仅仅只是开始。要制定一个可以经受住时间考验的标准策略，并且使它能被优雅地应用，同时能够在机构、团队或者公司内部进行调整，还必须进行一定的设计。必须理解基本原理，经过深思熟虑之后作出选择(而不是随意的偶然的决定)，只有这样才能奠定坚实的基础。

本书将帮助 Web 开发团队重新思考为什么要建设基于 Web 标准的网站，以及怎样最好地完成它。它将帮助我们衡量现在已经有有哪些可用的技术，以及 Web 标准对于开发团队或者整个网站的影响。本书还将告诉我们如何长期保持有效的组织，以及如何提高 Web 应用的稳定性并降低风险。它将帮助我们创造一些技术，以便充分利用 Web 标准在内容管理系统(Content Management System, CMS)中的独特优势。本书的最后部分将阐述有关 Web 标准的过程和人员配备问题。

什么是 Web 标准

“Web 标准”这个术语是指使用万维网联盟(World Wide Web Consortium, W3C)或者其他标准组织提供的开放且兼容的(而不是封闭、专有的，或者是属于某个公司所有的)推荐规范(recommendation)来设计网页。这些推荐规范和现代最佳实践组合起来，可以充分挖掘出在市场上占统治地位的现代 Web 浏览器的潜能，这恰好同那些过时的浏览器形成鲜明的对比，它们虽然表现力丰富，但是却总是不一致，甚至不兼容。在一个网站的主页上设置一个图标，说明“该网站是为 Netscape 浏览器设计的”，这种做法应该成为历史。

面对过时的浏览器时，基于 Web 标准的网站可以优雅地降级工作。基于 Web 标准设计的网页具有更好的可访问性(accessibility)，同时可以方便地移植到其他类型的设备上。这些技术还可以为用户、公司以及负责建设网站的团队带来更多其他的好处。本书的主题就是阐述这些技术。

Web 标准的基本优势

基于 Web 标准创建的 Web 站点有许多优势，它们是立即可用的，几乎不需要任何复杂的技术或经验。其优势包括以下几方面：

- 样式和脚本具有可重用性和一致性
- 带宽的使用及样式和脚本文件的缓存(caching)都减少了
- 网页的渲染更加快速
- 更简洁更易于维护的代码
- 对辅助技术而言它更容易访问
- 易于优化搜索引擎

- 提高不同浏览器厂商的产品之间的兼容性
- 使得提高下一代浏览器文本清晰度的可能性增大
- 提高您的网页的读者人数

Web 用户界面

在软件术语中, Web 站点的前端被称为用户界面(User Interface, UI)层。Web 站点的 UI 层包括用户在浏览器中所看到的所有插图、文本、格式命令、交互指令, 以及通过 Internet 从 Web 服务器发送过来的控制。用户可以通过单击某个对象或者输入文字来同 Web 页面的 UI 进行交互或者“接触”, 通过这种方式输入一个新的请求, 然后该请求通过 Internet 传送回服务器, 于是开始了一个新的循环(如图 IN-1 所示)。

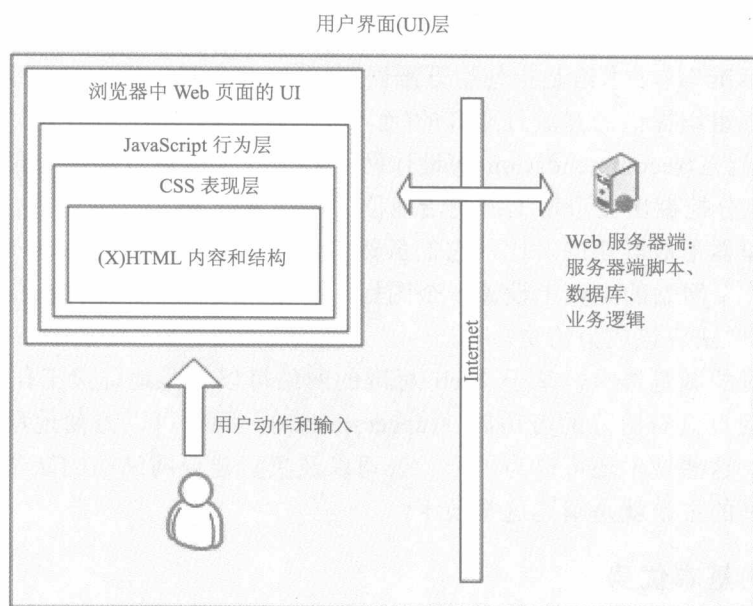


图 IN-1 Web 页面的用户界面由若干个技术层组成

比较前端和服务端程序, 可以发现它们有很大的不同, 服务器端程序包括业务逻辑以及与数据库或者其他数据仓库的直接交互。通常, 服务器端程序必须负责对 UI 层的渲染。同样, UI 层也可以向服务器端程序发送指令或者输入, 而且也可以包含一些业务逻辑。这说明 UI 层的作用是多么广泛, 它几乎涉及了 Web 站点的各个方面, 从最简单的静态标记网页到复杂的业务逻辑。

当 Web 作者创建一个现代的 UI 层的时候, 它们可能包含复杂的指令, 或者在

网页和服务器端程序之间共享某些代码，以使得程序更加高效。因此，UI 的重新设计或者修改可能会十分复杂。

如何才能高效地管理这些代码，并且在大型开发团队中共享这些代码，同时还要从生产力的角度使代码保持长期有效呢？

用户界面设计

上个世纪 90 年代的“dot-com”热潮带来了许多恐怖的 UI 实践，导致了許多 Web 站点臃肿不堪，结构不好且存在风险，并且创建网站的效率十分低。简单的 Web 页面的结构却非常糟糕，也就是人们常说的“标签汤(tag soup)”——由一堆胡乱嵌套的 HTML 表格以及单像素透明分隔符 GIF(single-pixel transparent spacer GIF) 组成(必须设计好它们之后才能处理页面内容或者应用)，这简直就是一场灾难(如图 IN-2 所示)。

```
<td align=center width=107>
<a href="#272"
onMouseOver="define('etwo5b'); window.status='Electronic Reporting';return true;"
onMouseOut="define('prblnk4a');">
</a><br>
<td align=left width=246>
<a href="#cash"
onMouseOver="define('cash6b'); window.status='Cash Information';return true;"
onMouseOut="define('prblnk4a');">
</a><br>
<tr>
<td colspan=3 align=center>
<br>
</table>
<center>
<table width=380 cellpadding=2 border=0>
<td valign=bottom width=125>
<center>
<font face="Arial, Helvetica" size="-2">
To utilize or learn <br>
about a specific <br>
payment / reporting<br>
<br><br>
</font>
</center>
<td rowspan=4 align=center width=2>

<td width=50%>
<font face="Arial, Helvetica" size="-1">
<font face="Times New Roman" size=+2>T</font>he Management is
<p><font face="Times New Roman" size=+2>I</font>n keeping with tradition,
```

图 IN-2 一个老式 HTML 页面的例子，代码中满是内联表现属性、内联事件处理器及标签

当时，大部分的 HTML 页面都是这种形式的，完全不重视用户的使用体验，且

对页面要做哪怕是最简单的修改也非常困难。要通过 JavaScript 来启用用户交互则更是一场灾难,它将导致大量内嵌在 HTML 代码中的事件处理器和基于某某浏览器专有技术的代码分支语句等。最终,要控制您的 Web 站点或应用的任何输出,您都需要查找所有层中的代码,修改内容、表现及应用逻辑,这将造成难以估量的业务风险和管理困难。

现代网站规划

当前,在大规模 Web 项目中,绝大部分的工作和项目计划都不重视 UI 层,并且认为只要事后再补充即可,实际上 UI 层会深刻地影响内容管理、Web 应用、搜索引擎优化(Search Engine Optimization, SEO)、带宽开销、站点性能及维护的难易程度等。通常的项目规划都从后端软件开始,并且只在设计层面上涉及到一点点 UI。

幸运的是,现在已经有办法可以减小传统 Web 编码方式的长期风险并摆脱其束缚。拥抱现代技术要从 W3C 及其推荐规范开始,也就是通常所说的 Web 标准。

不仅仅要从设计角度来考虑 Web 标准,还要从内容管理、应用以及其他动态系统来考虑。如果一个 Web 站点想要获得基于 Web 标准的 UI 所带来的好处,它就需要从各个层次来考虑,并且引入能够让该站点智能成长的规划。

1. Web 标准的关键

当使用基于 Web 标准的方法来创建网站的时候,究竟有哪些不同的地方呢?

不同点首先是在 UI 层上,符合 Web 标准意味着要将表现同内容、结构以及 UI 元素的脚本行为百分百地分离。其次是在后端,即限制 Web 应用中的 UI 代码和那些需要进行定期更新的 CMS 代码的混合,并且将这种严格的分离应用到其他所有的静态页面上。

需要注意的完全不同之处有:

- 内容和结构——即标记层,基于 Web 标准的标记层通常由超文本标记语言(HyperText Markup Language, HTML)或者可扩展超文本标记语言(eXtensible HyperText Markup Language, XHTML)组成
- 表现层——由层叠样式表(Cascading Style Sheets, CSS)组成,为标记和站点的脚本所引用
- 行为层——那些使得用户事件和交互成为可能的 JavaScript 元素
- 软件和 CMS 层——该层有自己的 UI,并且通常可以创建以上的 UI 层
- 帮助创建以上各层的开发团队和开发过程

在没有软件或者大型开发团队的时候,要在静态的设置中达到 UI 层的分离,也并不困难。关键是 Web 软件需要足够重视以上这些不同点,并且在项目规划中要把 UI 层看做是“一等公民”,UI 层需要同所有系统进行智能的、有意义的交互,而不能把它视为仅需要做事后补充的“二等公民”。

2. 软件架构模式

不同的代码层应该有不同的作用，这对于软件产业来说并不是什么新的概念。事实上，教科书中有许多架构设计模式的例子，这些例子软件学生都已经学习了很多年了。有关架构设计模式的例子，维基百科上给出了一个很好的列表，具体链接为：http://en.wikipedia.org/wiki/Architectural_pattern_%28computer_science%29。

一个很流行的架构模式就是“模型-视图-控制器(model-view-controller)”，简单地说，该模式就是：

- 模型：代表那些用于各种商业用途的原始数据的逻辑意义。可以把模型层看做是应用程序接口(Application Program Interface, API)，程序的其他部分可以通过此接口同它进行交互。模型层主要负责计算性的工作，也就是我们通常依赖电脑为我们所做的，例如计算购物车中所有物品的总价格，或者确定今天是否是我们的结婚纪念日。
- 视图：视图就是指当模型被渲染到 UI 层或者到部分 UI 层上的时候，用户所看到的视觉享受。可以把 Web 应用中的 HTML+CSS 的 Web 页面看做是视图。
- 控制器：控制器通常是事件驱动的，它负责解释和响应用户的动作，同时它还可能推动模型的修改。可以把控制器对应于 Web 应用中负责处理用户动作(包括单击鼠标以及提交基于 Web 的表格等)的那一层。

有人将此模型扩展到了 Web 软件和 Web 标准中，将 UI 层内容、表现和行为的分离对应于该模式中的模型(内容和结构)、视图(表现)和控制器(行为)。有经验的软件架构师通常都会很渴望采用这种层次分明的前端架构，无论他们是否很熟悉 Web 设计。

一种新方法：UI 架构方案

传统的规划一般先思考后端要求，然后再创建 UI 层代码。现在，使用现代的基于 Web 标准的方法之后，开发团队应该不断地问自己以下的问题：

- UI 层的创建和结构是否易于维护？
- UI 层如何影响 SEO？
- UI 层如何同网站的内容管理系统(Content Management System, CMS)交互？
- 有没有可能在不太影响 CMS 或者不需要 CMS 工作人员的前提下，重新设计或者做一些简单的设计修改？
- 当需要修改或者增强 UI 的时候，会发生什么事？
- 如何将 UI 和 Web 应用结合起来？
- 当应用逻辑发生改变时，会发生什么事？
- 设计的改变会给一个应用带来多大的风险？

- 关键任务(mission-critical)应用应该屈服于不必要的风险设计、简单内容或者脚本变化的压力么？

一个规划良好的基于 Web 标准的方法可以在以下两个层次上减轻上述风险：首先，前端代码；其次，在后端代码和前端代码相遇的地方。

随着时间的推移，这些问题对于任何 Web 站点来说都会成为大问题。大型公司通常都有自己的 Web 站点，要想在一夜之间对这些站点进行大量的修改是不可能的，或者是非常困难的。因此就可能要求逐步地修改。究竟如何操作，应该要根据具体情况具体分析。

当我们规划修改方案时，首先要想清楚需要设计什么，究竟是商业内容还是一个应用，以及它需要如何在浏览器中被渲染。其次，根据每个选项的优缺点做出合理的选择。最后，思考如何使得 Web 站点达到符合标准这个目标，以及如何使它保持标准。