

Web 应用浏览器兼容性的自动规范化方案^①

屈建勤¹, 陈耀森²

¹(汕头大学 工学院计算机系, 汕头 515063)

²(广州龙芯中科电子科技有限公司, 广州 510006)

摘要: 近年来万维网应用在中国正在飞速增长. 但是在我国很多 Web 应用要求用户只能用 IE 浏览器. 符合 W3C 组织制定的规范就能保证 Web 应用跨浏览器的移植性. 文中着眼于 Web 应用的特点, 从遗留系统的角度来看待这个问题, 提出了基于模型的自动规范化重构方法来解决这个问题. 作为一个例子, 构建了一个能够跨浏览器上访问本来只能用 IE 的中国银行的网上银行服务的实例, 证明此方法能够有效解决国内现存 Web 应用的规范化问题.

关键词: DOM; W3C; 自动规范化; 浏览器兼容性

Automatic Standardization Method of Web Application for Cross-Browser Portability

QU Jian-Qin¹, CHEN Yao-Sen²

¹(College of Engineering, Shantou University, Shantou 515063, China)

²(Loongson(Guangzhou)Technology Corporation Ltd., Guangzhou 510006, China)

Abstract: Web applications in China are growing rapidly in recent years. However many web applications in China require the exclusive use of IE browser. Conformance to W3C standards ensures cross-browser portability of web applications. In this paper, a model-based automatic refactoring method for standardization of web applications is proposed, which employs the characteristics of web applications from the perspective of legacy systems. As an example, a cross-browser access to the original IE-only on-line banking for Bank of China is shown, which demonstrates that the proposed method can handle the existing problems of the standardization of web applications effectively.

Key words: DOM; W3C; standardization; cross-browser portability

1 引言

随着互联网深入社会生活的每一个角落, 更多的企业、组织以及个人意识到万维网可以帮助他们宣传自己, 更可以在网络上实现同世界各地的客户开展商务活动. 作为用户了解企业的窗口, 浏览器兼容性问题是客户最先遇到的和最迫切需要解决的问题.

在当前的 B/S 架构中, 用户通过使用浏览器获取 Web 服务器传来的内容. 在我国, 由于垄断、盗版和商业竞争等原因, 很多网站开发不遵守 W3C 规范, 导致网页代码在不同的浏览器间的兼容性很差, 例如百度、淘宝的很多网页都是只能在基于微软的 IE (Internet Explorer)内核的浏览器下使用.

据 w3schools 网站统计, 如图 1 可知.

2011	Internet Explorer	Firefox	Chrome	Safari	Opera
March	25.8 %	42.2%	25.0%	4.0%	2.5%
February	26.5 %	42.4%	24.1%	4.1%	2.5%
January	26.6 %	42.8%	23.8%	4.0%	2.5%

图 1 各浏览器市场占有率

在浏览器市场上, IE 内核的浏览器所占用的比例并不高. 与此相对应, 仅能在 IE 浏览器下正常浏览的网页必然不能满足差异性的用户群体. 其必然的结果是削弱了企业的自我竞争力, 在国际化的道路上难以打开局面. 而像国外的花旗、渣打等银行则在代码级别上遵守了 W3C 的规范, 没有跨浏览器使用的问题.

^① 收稿时间:2013-05-15;收到修改稿时间:2013-07-22

长期以来,对于 DOM^[1-3]部分的兼容性问题,传统的解决方案有如下 2 种:

① 完全按照 W3C 的 DOM 规范里的 API 来编写程序.此方法的局限是,很多标准的 API 对于国内习惯于某一浏览器开发的人员来说,不熟悉并且不愿学习.

② 使用能力检测.也叫做对象检测,指的是在执行代码之前检测某个脚本对象或者方法是否存在,而不是依赖于你对哪个浏览器具有哪些特性的了解.如果必需的对象或方法存在,那么说明浏览器能够使用它,并且代码也可以按照预期执行.例子如下:

```
if (document.all) {
//IE 兼容的部分代码
} else {
//Firefox 版本的代码
}
```

对于现有的代码来说,以上两种方案有一个共同的缺点就是按照上面的方法重新开发程序工作量太大成本太高,难以实现自动化.此类现有代码的问题在软件工程学科上属于遗留系统^[4]的改造问题.对网页代码的改造的目的是要将不能够跨浏览器的网页代码改造为能够跨浏览器的,同时要保持原有的功能不变.在遗留系统的改造中,重构是能够达到将已有的代码进行改造,同时又不改变已有功能性的目的.故我们提出了基于重构的方式对非兼容网页代码进行改造,同时又不人工改动现有代码.由于网页本身是以 W3C DOM 模型规范作为指导,本文提出的针对非规范网页代码的改造,以 DOM 模型规范作为指导,以符合 W3C 规范为目标,即基于模型的网页代码自动规范化的重构.

2 基于 DOM 模型规范的重构

在 W3C 组织制定针对网页代码 DOM 规范前,各浏览器通过使用一种叫做 DHTML^[5](Dynamic HTML)技术来操纵文档、样式等特性.DHTML 为嵌入到网页中的脚本语言提供了一个编程接口,但由于各种浏览器提供的接口各不相同,其应用和发展受到了很大的制约.DOM(Document Object Model)起源于 DHTML,虽然 W3C 组织早在 1998 年 10 月就提出了 DOM 的正式建议(Recommendation),但到目前为止,这个建议被一些主流浏览器实施的程度差异很大.

DOM 是一套平台和语言独立的接口,允许通过程序或者脚本动态地访问和更新文档的内容、结构和

样式.DOM 规范分成了不同的级别,以当前常用的 DOM Level 2 为例,DOM2 推荐标准分成了以下 6 个不同的规范:DOM2 Core、DOM2 HTML、DOM2 Events、DOM2 Style、DOM2 Traversal and Range、DOM2 Views.其中 DOM2 HTML 是专门针对 HTML 网页文档设计的,通过将整个 HTML 页面文档规划成由多个相互连接的节点构成的文档,文档中的每个部分都可以看成是一个节点(Node)的衍生物.这样的一个节点集合看成是一棵节点树(Tree),也就是浏览器载入网页文档后得到的 HTMLDocument 对象.通过这个对象,开发者可以对文档的内容和结构拥有强大的控制力,并且使用 DOM 提供的 API 在文档树中遍历、删除、添加、修改和替换节点,由此生成丰富的应用形式.浏览器厂家由于为了提高竞争力等原因,在对 DOM 规范的实现上,并没有完全遵守 W3C 制定的标准.通过对 DOM 解析文档后得到 Document 对象统计发现,对 Document 对象的方法的部分兼容性测试如表 1 所示.

表 1 各浏览器的 DOM 兼容性支持

方法名	IE 6.0	FF	W3C DOM
addEventListener	N	Y	Level 3 Core
adoptNode	N	Y	Level 3 Core
attachEvent	Y	N	
clear	N	Y	Level 1
close	Y	Y	Level 1
compareDocumentPosition	N	Y	Level 3 Core
createAttribute	Y	Y	Level 1
createAttributeNS	N	Y	Level 2 Core
createCDATASection	N	Y	Level 2 Core
createComment	Y	Y	Level 1
createDocumentFragment	Y	Y	Level 1
createElement	Y	Y	Level 1

对于已有的不符合 DOM 模型规范的代码,使用重构^[6]的方式能够使不符合规范的代码规范化,实现代码的跨浏览器兼容性.若是使用手工的方式对代码进行重构的话不仅费时费力,而且容易出错.以遵守 DOM 模型规范为目标的自动化重构工具能够提供很好的重构效果和用户体验.

要构造的自动重构工具面向的是存在兼容性问题的网页代码,其目标是遵守 DOM 模型规范.浏览器是通过嵌入的脚本语言实现对 DOM 模型的操纵,发现不兼容的代码则首先要实现对嵌入脚本进行解析.想要从进行解析后的嵌入脚本中发现不兼容的部分,则需要有已知的不兼容列表.重构工具所需的程序数据库在生成符合 DOM 规范的代码时经常用到.以上这

些是构造自动化重构工具的必要条件. 则重构工具的整个工作过程如下图 2 所示.

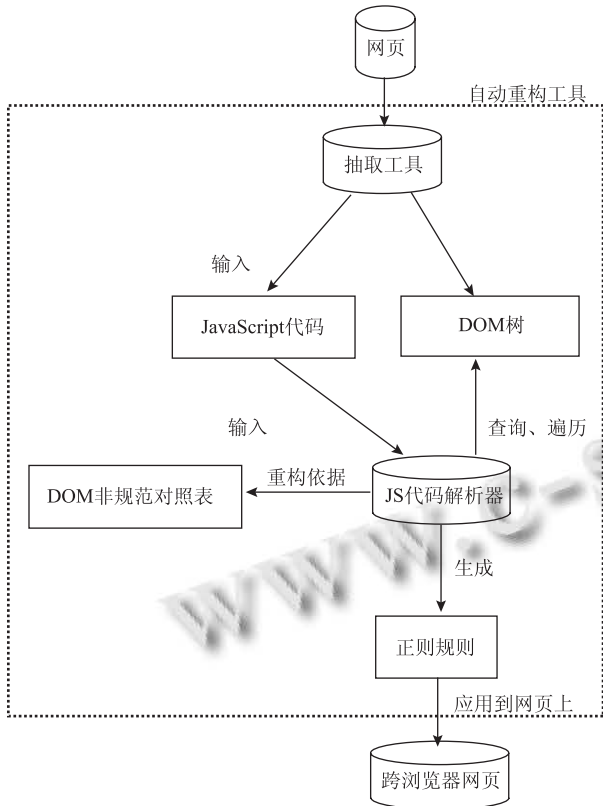


图 2 重构工具工作过程

3 构造基于DOM模型规范的自动重构工具

在我们的具体实现中, 采用的是对嵌入到网页中的 JavaScript^[7]脚本代码进行解析并发现和纠正不兼容部分的. 针对具体的 JavaScript 脚本, 构造一个语言解析器^[8], 采用了 ANTLR^[9](Another Tool for Language Recognition)实现. ANTLR 能够接受文法语言描述, 能够依据给出的语法规则生成相应的基于 Java、C++或 C#的词法分析器或语法分析器代码, 极大地提高了自行编写分析器的效率. ANTLR 接受 LL(k)文法. LL(k)文法书写的限制是: 1: 不能用左递归. 2: 向前看的字符数最多是 k 个, 当编译程序递归 k 次仍无法找到匹配的的产生式时, 则句子识别失败. 重构工具的程序数据库在以网页文档作为输入时, 就是需要从网页文档解析出 DOM 模型出来, 使用解析后得到的 Document 对象, 使用 DOM API 可以轻松地对整个网页内容进行遍历、查询等. 使用开源的 HTML 解析器 NekoHTML 可以把网页解析出一棵 DOM 树. NekoHTML 是一个简单地 HTML 扫描器和标签补偿器^[10](tag balancer), 使得程序

能解析 HTML 文档并用标准的 XML 接口来访问其中的信息. 这个解析器能够扫描 HTML 文件并“修正”许多作者(人或机器)在编写 HTML 文档过程中常犯的错误. NekoHTML 能增补缺失的父元素、自动用结束标签关闭相应的元素, 以及不匹配的内嵌元素标签. NekoHTML 的开发使用了 Xerces Native Interface (XNI), 后者是 Xerces2 的实现基础.

使用 NekoHTML 作为预处理器对输入的网页文档预处理得到 DOM 树, 使用 ANTLR 作为 JavaScript 代码解析器解析 JS 代码, 使用解析过程中获取的上下文信息和浏览器兼容性对照表则可以识别并对非兼容代码予以纠正, 起到了重构的作用. 非 DOM 兼容的代码重构后完全符合 W3C 的 DOM 模型规范, 且易维护、易扩充、代码更加健壮.

4 实验结果

为测试对网页代码的重构效果, 特别发起了一个针对中国银行网上银行进行重构的项目, 通过构造自动重构工具, 对含有较多浏览器兼容问题的中国银行网银的网页代码进行测试. 为了方便用户测试, 该项目将重构工具部署在一台中间服务器上, 当用户设置代理服务器为中间服务器即可. 实验结果为, 当使用重构工具后, 在多种系统包括特别是龙芯上的 Linux 系统下用多种浏览器如 Firefox 等都可以十分顺利登陆并且使用中国银行网上银行的几乎全部页面, 详情可参照本项目的网站 <http://www.w3cbank.org/webank>.

5 总结

本文通过分析当前我国许多网站网页代码出现的浏览器兼容性问题的原因, 提出了针对非规范的网页代码进行自动规范化来对网页代码实现重构的做法. 自动重构后的网页代码, 遵照 DOM 模型, 符合 W3C 规范, 从而实现了网页跨浏览器的兼容性. 最后通过实现一个基于 DOM 模型的自动规范化重构工具对网页代码进行重构. 使用网页自动重构工具将会对现有网页文档的规范化有非常积极的意义, 以较小的代价实现了对现存的不规范 Web 应用遗留系统的规范化实现跨系统跨浏览器的兼容性. 因此本文提出的方案对于解决国内广泛存在的 Web 应用跨浏览器兼容性问题具有非常强的实用价值和推广价值.

(下转第 187 页)

时表示键按下, 值为 0 时表示键抬起, 值为 2 时表示按键按下之后并没抬起. 这样用户就可以通过得到的键值以及状态来编写相应的按键应用程序.

测试结果如图 3、图 4 所示.

```
type: EV_KEY = 1, event = 3, value = 1
type: EV_KEY = 1, event = 3, value = 0
type: EV_KEY = 1, event = 3, value = 1
type: EV_KEY = 1, event = 3, value = 0
type: EV_KEY = 1, event = 3, value = 1
type: EV_KEY = 1, event = 3, value = 0
```

图 3 按键状态 1

```
type: EV_KEY = 1, event = 3, value = 2
type: EV_KEY = 1, event = 3, value = 2
type: EV_KEY = 1, event = 3, value = 2
type: EV_KEY = 1, event = 3, value = 2
type: EV_KEY = 1, event = 3, value = 2
type: EV_KEY = 1, event = 3, value = 2
type: EV_KEY = 1, event = 3, value = 0
```

图 4 长按状态 2

4 结束语

本文介绍了嵌入式 Linux 下 Input 输入子系统设备驱动的结构, 向上为用户提供接口函数, 向下为驱动层程序提供统一接口函数, 以及以 4x4 矩阵按键为例说明驱动层程序调用了 Input 子系统的那些接口函数. 最后以 ARM9 系列的芯片 S3c2416 为核心板的硬件基础上进行外围硬件扩展, 编写了简单的测试程序实现了按键驱动和 Input 子系统为用户提供的接口函数的调用. 对诸如嵌入式中的 Input 子系统中输入设备的应用提供了一定的参考价值.

参考文献

- 1 李俊. 嵌入式 Linux 设备驱动开发详解. 北京: 人民邮电出版社. 2008.
- 2 郑强. 驱动开发入门与实战. 北京: 清华大学出版社. 2011.
- 3 Bovell DP, Cesati M. Understanding the Linux Kernel. 北京: 中国电力出版社. 2007.
- 6 Fowler M. 熊节译. 重构: 改善既有代码的设计. 北京: 人民邮电出版社. 2010.
- 7 David Flanagan. 李强等译. JavaScript 权威指南. 北京: 机械工业出版社. 2007.
- 8 Aho AV. Compilers: Principles, Techniques & Tools. 北京: 人民邮电出版社. 2008.
- 9 Parr T. The Definitive ANTLR Reference. Pragmatic Bookshelf, 2007.
- 10 Sambells J, Gustafson A. 李松峰, 李雅雯译. JavaScript DOM 高级程序设计. 北京: 人民邮电出版社. 2008.
- 1 Document Object Model(DOM)Level 1 Specification, Version 1.0. W3C Recommendation, Sep.1998.
- 2 Document Object Model(DOM)Level 2 Core Specification, Version 1.0. W3C Recommendation, Nov.2000.
- 3 Document Object Model(DOM)Level 3 Core Specification, Version 1.0. W3C Recommendation, Apr.2004.
- 4 Seacord RC, Plakosh D, Lewis GA. 梁海华译. 遗留系统的现代化改造. 北京: 清华大学出版社. 2004.
- 5 DHTML Reference, MSDN Library. Microsoft, 1996.

(上接第 198 页)

参考文献