基于 WebKit 浏览器引擎的动态页面数据采集方案

陈飞
（北京邮电大学信息与通信工程学院，北京 100876）

摘要：随着 Web2.0 技术的快速发展，互联网上应用 AJAX 技术的网站越来越多，大量动态网页的出现为数据采集工作带来了很大的挑战，针对静态网页的爬虫等采集工具已无法满足互联网行业的需要，严重影响了网络内容监控、数据挖掘等研究工作的进展。基于此，本文在原有论坛数据采集系统基础上进行了扩展，提出了一种动态页面数据的采集方案，该方案采用 WebKit 浏览器引擎做为核心，实现了高效跨平台的动态页面采集，并且针对可能出现的复杂网络状况设计并实现了超时等待机制，满足了健壮性的需求，而且采用配置文件的方式使新添加的数据采集工作只需较少的编码及配置即可完成。

关键词：Web 数据采集； WebKit； AJAX
中图分类号：TP391

Dynamic Page Collection Method Based on WebKit

Chen Fei
(School of Information and Communication Engieering, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: With the development of the WEB 2.0, the webs on the internet designed by Ajax are increasing. Lots of dynamic pages bring a huge challenge to web data acquisition. The web crawler designed for static pages cannot satisfy the need of development of Internet and affect the progress of research work – such as web monitor, data mining. This paper introduced a new web improved crawler based on dynamic pages. This crawler uses WebKit browser engines as core, achieve efficient cross-platform acquisiton of dynamic pages, and design timeout waiting mechanisms for the complex network to satisfy the need of robustness of system. This improved clawer uses the configuration file to reduce the requirement of coding and configuration for new add data.

Key words: Web data acquisition; WebKit; AJAX

0 引言

随着 Web2.0 的兴起，AJAX 技术风靡一时，客户端与服务器端异步交互的方式既减小了服务器端的压力，而且带来了更好的用户体验。然而，使用该技术产生的大量动态网页给网络数据获取造成了新的难题，传统的用于采集静态网页的 Web 数据采集工具如网络爬虫抓取的内容远少于页面呈现的内容，大量动态网页中的有用信息无法获取使得以网络数据为主要处理对象的工作无法顺利进行，严重影响了网络内容监控，网络数据挖掘等学科的发展。

因此，如何改进传统的 Web 数据采集系统，使之支持动态页面解析，已经成为了当前信息采集技术的一个研究热点。互联网领域的专家学者们对此课题做了不少有益的研究尝试，提出了有建设性的思路和解决方案。当前动态页面采集的主要方法大体有两种[1,2,3]：一是利用开源浏览器接口（如 Firefox），以编写插件的形式对浏览器输出结果进行采集；二是利用现有的脚本解释引擎（如 SpiderMonkey、Rhino 等）根据信息采集的需要对相关 DOM 对象进行绑定，对输出结果进行采集。然而，目前的研究也存在一些问题：一是现在的研究主要面向设计大规模网络爬虫爬取动态网页的通用方法，对于一些有针对性的定向的数据采

作者简介：陈飞（1984-），男，硕士，主要研究方向：信息检索及自然语言处理. E-mail: cfhamlet@gmail.com

集（如特定论坛或商业网站商品信息的采集）支持效果不够理想；二是大部分方案实现较为复杂，并不适用于小规模即时的数据采集需求。

基于以上原因，本文在一个简单的爬取类论坛结构数据的爬虫基础上进行了扩展，提出了一种基于 WebKit 浏览器引擎的采集动态页面数据方案。通过采用 Qt 框架，使得程序有较好的可靠性及跨平台性；通过将接口与配置文件分离的方式，使程序具有很好的可扩展性；针对复杂的网络环境设计了超时等待机制，程序的鲁棒性有了很大的提高。

1 AJAX 技术及其特点

1.1 AJAX 的定义及关键技术

AJAX 是 Asynchronous JavaScript and XML（异步 JavaScript 和 XML）的缩写，由著名用户体验专家 Jesse-James Garrett 于 2005 年首先提出。

AJAX 不是一种新技术，而是一系列已经被广泛应用的 Web 相关技术的组合，如 XML、CSS、DOM、XMLHttpRequest、JavaScript 等。其成功之处就在于其构建了更为动态和响应更为灵敏的 Web 应用，实现了浏览器和服务器之间的异步并行处理，既减轻了服务器端的负担又带来了独特的用户体验。

标准的 AJAX 包括[4]：

（1）采用 XHTML 和 CSS 标准化显示

（2）采用 DOM 实现动态显示和交互

（3）采用 XML 和 XSLT 进行数据交互和处理

（4）采用 XMLHttpRequest 进行异步数据获取

（5）采用 JavaScript 进行绑定和处理数据

1.2 Ajax 模型与传统 Web 应用模型的区别

与传统的 Web 应用不同，AJAX 并不是基于静态页面的方式来构建应用，它推行更少量的页面组成，其中每个页面是一个更小型的使用 JavaScript 开发的 AJAX 组件，这些组件使用 XMLHttpRequest 对象以异步的方式与服务器通信，从服务器端获取需要的数据后对使用 DOM API 更新页面内容。

在传统的 Web 应用模型中，典型的交互的方式是由客户端浏览器向 Web 服务器发送 HTTP 请求。Web 服务器对用户的请求进行处理，将处理结果以 HTML 页面的形式返回给客户端浏览器。用户必须在发送 HTTP 请求或 Web 服务器处理过程中进行等待，而且即便是仅仅改变页面中的一小部分内容，Web 服务器都要返回一个完整的 Web 页面，浪费了大量的时间和带宽。传统的 Web 应用模型如图 1 所示。

AJAX 的工作原理与传统 Web 应用模型不同之处在于浏览器与 Web 服务器之间采用的是异步通信方式，在客户端和服务器之间增加了一个中间层——AJAX 引擎，用以处理客户端的请求，实现了交互的异步化。用户的操作并不都提交给服务器，一些数据验证和处理由 AJAX 引擎完成，只有需要从服务器读取新数据时才由 AJAX 引擎向服务器提交请求。AJAX 引擎以后台运行方式获取所需数据，不需要重载整个页面，只需更新所需部分的内容，大大减少了数据传输量，缩短了响应时间，既减轻了服务器端的负担又提升了用户体验。AJAX 应用模型如图 2 所示[5]。

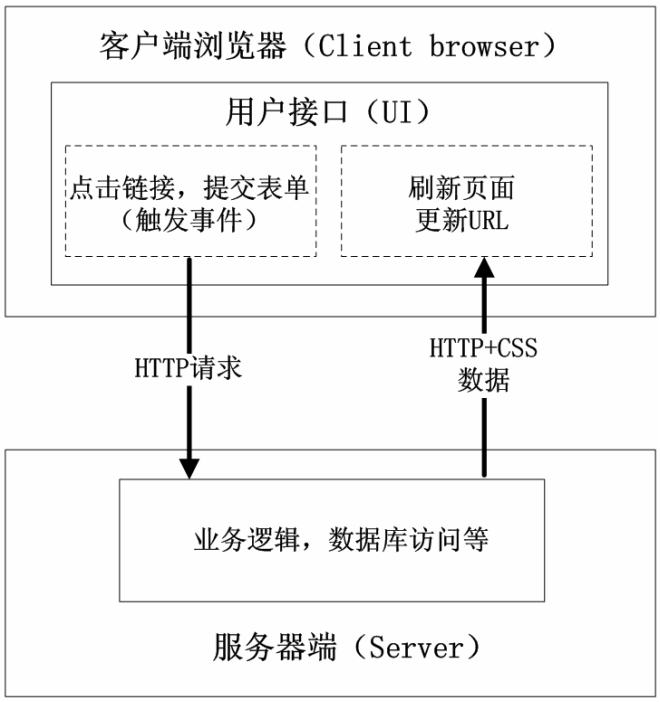

图 1　传统 Web 应用模型
Fig. 1　Traditional Web Application Model

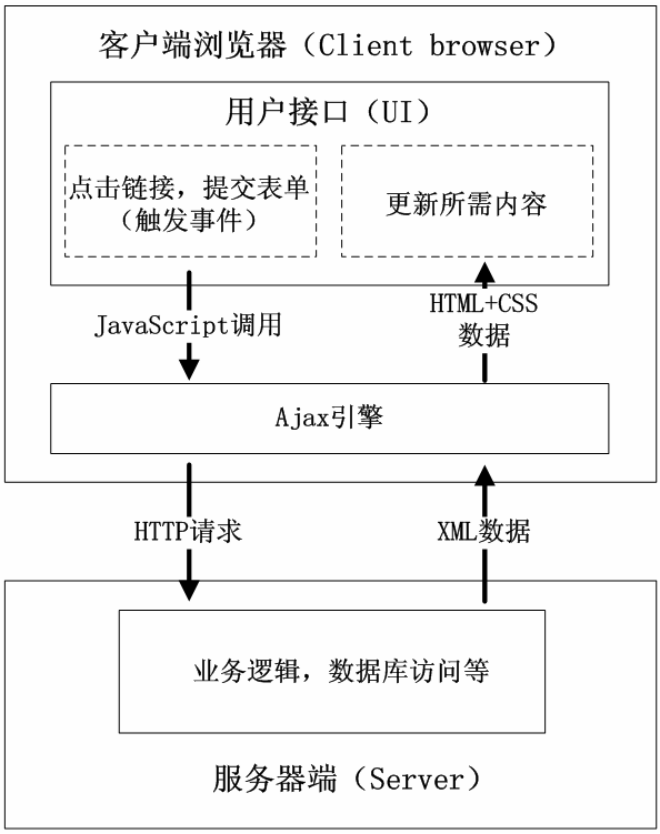

图 2　AJAX 应用模型
Fig. 2　Web Application Model Based on AJAX

2 主要开发工具简介

2.1 WebKit 浏览器内核

WebKit 的前身是 KDE 小组的 KHTML，简单来说是一个开源的 Web 浏览器引擎，也就是浏览器的内核。Apple 的 Safari, Google 的 Chrome, Nokia S60, Android 手机的默认浏览器均采用的 Webkit 作为内核，是同 Gecko、Trident 并称的当今主流的三大浏览器内核之一。其引擎高效稳定，兼容性好，源码结构清晰，易于维护。

WebKit 从代码结构上来说主要包含三大部分，如图 3 所示[6]。

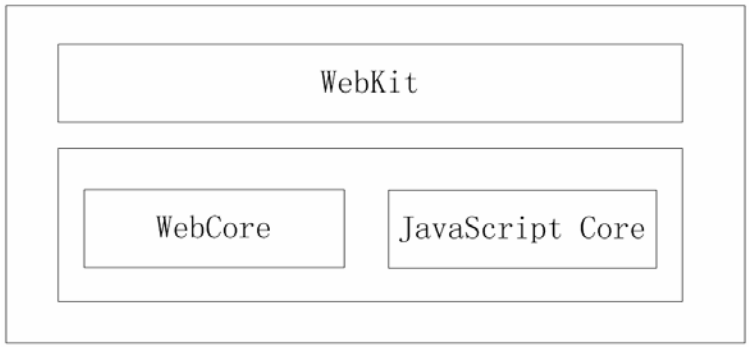

图 3 WebKit 结构
Fig. 3 The Structure of WebKit

其中核心部分为 WebCore，它实现了对文档的模型化，包括 CSS，DOM，Render 等，JavaScript Core 是对 JavaScript 的支持。而 Webkit 部分抽象出了与浏览器直接对应的一些概念的实现，如 WebView，WebPage，WebFrame 等。应用程序不需要直接操作 WebCore 及 JavaScript Core，而是同 WebKit 模块提供的 API 进行交互。

2.2 Qt 开发框架及对 WebKit 的支持

Qt 是著名的跨平台的 C++应用程序开发框架，自 Qt4.5 集成了 WebKit 以来，其丰富便利的通用接口模糊了应用程序与网络内容的差别。其对 WebKit 的支持主要包括表 1 所示的几个类[7]。

表 1 Qt 中对 WebKit 支持的类
Tab. 1 The Classes of Qt for WebKit

Class name	*function*
QWebDatabase	*Access to HTML 5 databases created with JavaScript*
QWebFrame	*Represents a frame in a web page*
QWebHistory	*Represents the history of a QWebPage*
QWebHistoryInterface	*Interface to implement link history*
QWebHistoryItem	*Represents one item in the history of a QWebPage*
QWebHitTestResult	*Information about the web page content after a hit test*
QWebPage	*Object to view and edit web documents*
QWebPluginFactory	*Creates plugins to be embedded into web pages*
QWebSecurityOrigin	*Defines a security boundary for web sites*
QWebSettings	*Object to store the settings used by QWebPage and QWebFrame*
QWebView	*Widget that is used to view and edit web documents*

这些类中，最主要的为 QWebView，QWebPage，QWebFrame 三个，他们的关系如图 4 所示。

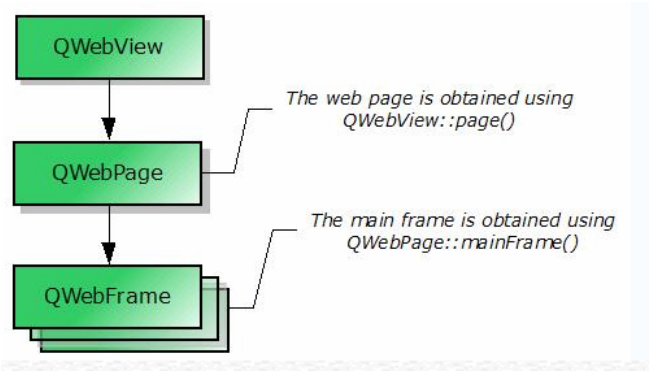

图 4　QWebView、QWebPage 及 QWebFrame 类之间的关系
Fig. 4　The Relationship Between QWebView QWebPage and QwebFrame

QWebView 类可以包含 QWebPage 和 QWebFrame 的对象。QWebView 通过创建 QwebPage 对象进而创建可视可编辑的网页。QWebFrame 是 QWebPage 的元对象，每个 QWebPage 对象至少有一个 QWebFrame，称为 mainframe，可通过 QWebPage::mainframe() 方法得到。通过调用 QWebFramed 的 page()方法可返回它所在的 QWebPage 对象。

3　类论坛结构数据采集总体框架介绍

论坛，商业网站一般以两层结构组织数据，这种组织形式称为类论坛结构，典型的类论坛结构中第一层为列表页，第二层页面为主要采集的数据体。此类数据结构相对固定，数据量大且较为集中，具有较高的研究价值。目前抓取此类数据的采集系统基本采用如图 5 所示的体系。

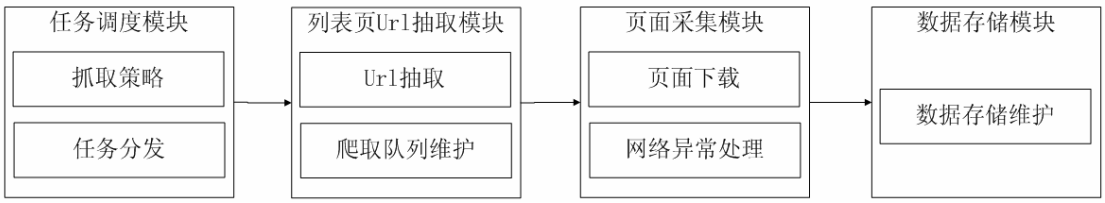

图 5　类论坛结构数据采集体系
Fig. 5　Framework of BBS-Data Acquisition

任务调度模块主要负责维护爬取策略，进行任务分发；列表页 Url 抽取模块根据不同的模板抽取列表页中的 url，并且维护爬取队列。页面采集模块根据抓取策略，通过 http 协议访问服务器，采集页面。此模块需要访问网络，因此要处理复杂的网络异常状况，如超时、网络中断等；数据存储模块负责存储维护，需要对数据库或者文件系统进行大量的 IO 操作。

4　动态页面采集模块

4.1　总体设计方案

动态页面采集模块是对传统静态页面采集模块进行的扩展与改进。关键步骤如图 6 所示。

整个流程分为三大步骤：

一、向服务器端发送 http 请求，接收原始页面数据，构建 DOM 树。此步骤中发送请求，接收数据，解析 js 及构建 DOM 树由 WebKit 底层实现。

二、针对不同的网站，维护相应的配置文件，配置文件中包含触发相应事件的 js 代码，

以字符串的形式传递给 WebKit 提供的 js 执行接口。由 WebKit 根据事件相应，更新 DOM 树。

三、调用 WebKit 的 I/O 接口，将 DOM 树转化成 html 格式，以字符串的形式输出。

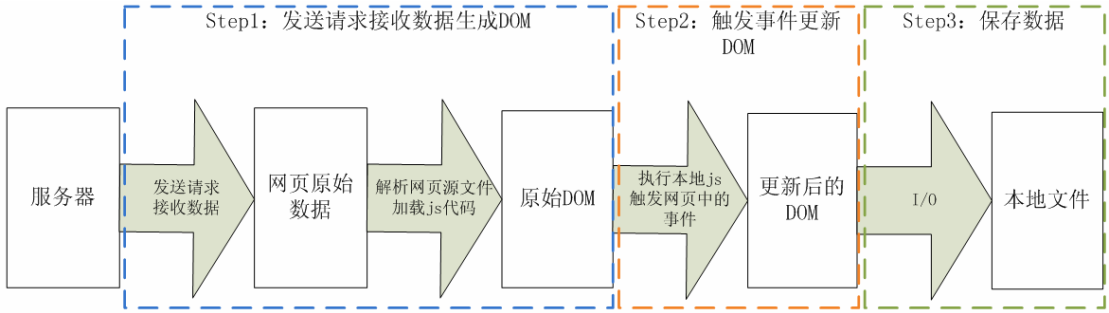

图 6　动态页面采集模块

Fig. 6　Dynamic Page Acquisition Model

整个模块主要使用 QWebPage 提供的各种接口，QWebPage 类可视为如图 7 所示的黑盒。

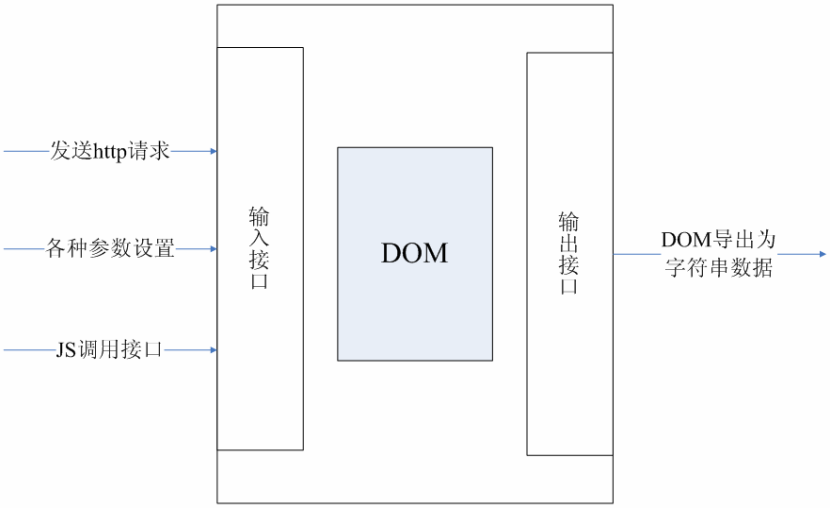

图 7　QWebPage 类接口示意

Fig. 7　API for QwebPage

http 请求接口负责发送请求，QWebPage 底层实现接收数据等操作；参数设置接口设置代理、是否自动加载图片、是否可执行 js 等；JS 调用接口读入本地 js 代码，QWebPage 底层解析执行；DOM 导出接口将 DOM 导出为 html 格式字符串。

4.2　数据接收数流程

由于网络状况相对来说较为复杂，服务器超时，网络中断等情况会对正常的数据采集造成影响，在接收数据阶段需要对异常状况进行处理和应对，然而 WebKit 提供接口无法满足需求。为此，采用三线程方式实现数据接收。具体流程如图 8 所示。

线程一负责正常的数据接收，监听 loadFinished 信号，若正常接收则终止线程二，设置接收标志位为成功状态 。

线程二为定时器线程，该线程监听接收时间，若超过预定接收时间即认为超时，终止正常接收线程，并且设置接收标志位为失败状态。

线程三对外提供接口，提供接收标志状态。后续步骤可通过此标志位进行相应的处理。

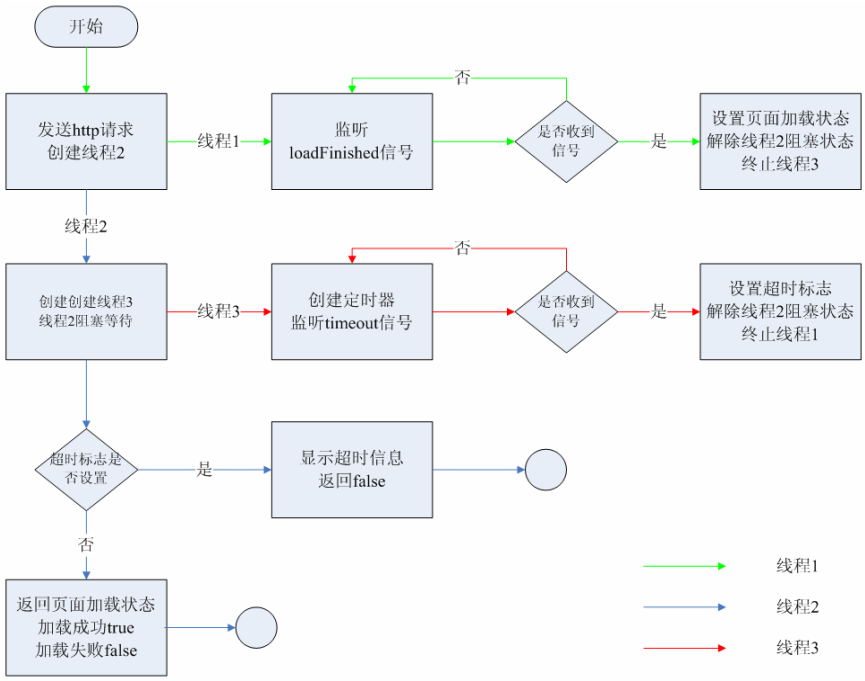

图 8　数据接收流程

Fig. 8　Procedure of Data Receiving

4.3　触发网页事件更新 DOM 流程

对于某些网页，需要触发网页上的某些事件与服务器进行交互才能获取所需数据。例如浏览器用户浏览网页时经常需要某些点击网页上的按钮，才能在页面上看到数据。为实现自动化的数据采集需要程序应能模拟真实用户的行为，如点击鼠标，滚动页面等。类似操作可以通过针对不同网站定制配置文件，编写 js 代码模拟触发事件的动作。程序提供了本地 js 代码调用接口，js 代码将以字符串的形式传给 WebKit，由内核低层模块实现事件触发更新 DOM 的操作。为保证与服务器交互的健壮性，设计了双线程的方式实现上述功能。具体流程如图 9 所示。

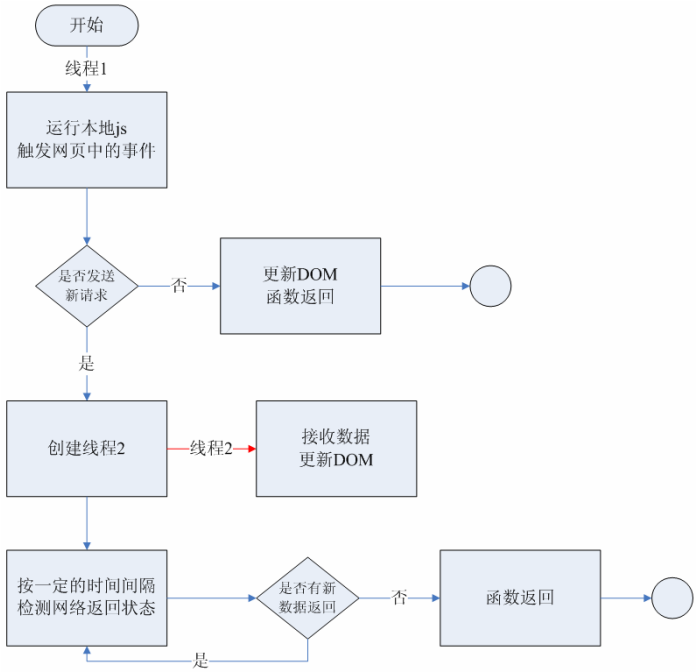

图 9　触发网页事件流程

Fig. 9　Procedure of Invoking Event

线程一运行配置文件中 js 代码，模拟触发事件的操作，并且循环等待服务器数据。若接收数据完成则唤醒线程二，对外提供接收状态指示接口。线程二负责等待线程一数据完成更新 DOM。

5 结论

本文给出了一个基于 WebKit 浏览器引擎采集动态页面数据的方案。对整体的结构及关键流程做了详细的说明。经过在多个论坛及商品销售网站上进行测试，验证了该方法可行高效，并且通过设计实现超时等待机制加强了程序的健壮性，可以应对较为复杂的网络环境。通过配置文件的方式实现了可扩展性的需求。对于即时的中小规模的动态页面数据采集有很好的实用借鉴价值。

[参考文献] (References)

[1] 闫谦时. 一种动态页面采集技术研究[J]. 计算机光盘软件与应用，2010(8)：116~117.
[2] 郭浩，陆余良，刘金红. 一种基于状态转换图的 Ajax 爬行算法[J]. 计算机应用研究，2009，Vol.26，No.11：4266~4269.
[3] 罗兵. 支持 AJAX 的互联网搜索引擎爬虫设计与实现[D]. 浙江省杭州市：浙江大学，2007.
[4] 怀艾芹. AJAX 技术在 Web 系统开发中的研究及应用[J]. 计算机时代，2010(9)：55~57.
[5] 李健，黄晗文，刘芳，等. Ajax 在 Web 中的应用研究[J]. 计算机与现代化，2009(7)：84~91.
[6] 李嘉昱. WebKit 内核研究[OL]. http://www.cnblogs.com/jyli/archive/2010/01/31/1660355.html
[7] Qt Port of WebKit[OL]. http://trac.webkit.org/wiki/QtWebKit