(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102662966 B
(45)授权公告日 2014.01.01

(21)申请号 201210060335.3

(22)申请日 2012.03.08

(73)专利权人 中国科学院计算机网络信息中心
地址 100190 北京市海淀区中关村南四街 4 号

(72)发明人 归文胜 黎建辉 杨风雷

(74)专利代理机构 北京君尚知识产权代理事务所（普通合伙） 11200
代理人 余长江

(51)Int.Cl.
G06F 17/30(2006.01)

审查员 唐楹琰

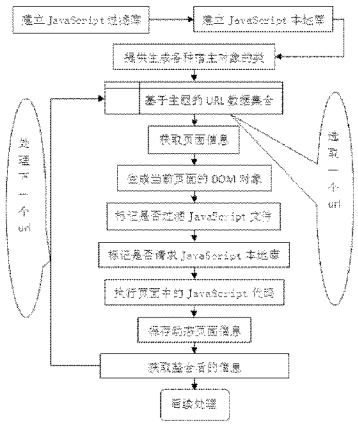
权利要求书2页 说明书8页 附图3页

(54)发明名称

一种面向主题的获取动态页面内容的方法及系统

(57)摘要

本发明公开了一种面向主题的获取动态页面内容的方法及系统，属于网络数据采集技术领域。本方法为：1）在抓取服务器端建立一 JavaScript 过滤库和一 JavaScript 本地库；2）获取每一抓取页面的页面信息，生成当前页面的 DOM 对象；3）根据过滤库检验当前页面中请求的外部文件，如果与主题无关，则在当前页面的 DOM 对象相应位置设置无需加载标记，否则设置正常加载标记；4）如果标记为正常加载的外部文件存在于本地库，则设置本地加载标记，否则设置正常加载标记；5）执行当前页面中的 JavaScript，获取动态页面信息；6）获取整合后的页面信息。与现有技术相比，本方法具有更高的时效性、且动态页面信息完整。



CN 102662966 B

1. 一种面向主题的获取动态页面内容的方法，其步骤为：

1）在抓取服务器端建立一 JavaScript 过滤库和一 JavaScript 本地库；

2）获取每一抓取页面的页面信息，生成当前页面的 DOM 对象；如果当前页面中使用宿主对象，则该抓取服务器将其实例化为相应对象；

3）根据所述 JavaScript 过滤库检验当前页面中请求的外部 JavaScript 文件，如果与主题无关，则在当前页面的 DOM 对象相应位置设置无需加载标记，否则设置正常加载标记；

4）对于标记为正常加载的外部 JavaScript 文件，如果当前处理的 JavaScript 文件存在于所述 JavaScript 本地库，则设置本地加载标记，否则设置正常加载标记；

5）执行当前页面中的 JavaScript，获取动态页面信息；其中，根据加载标记加载外部 JavaScript 文件；

6）检验获取的每一动态页面是否丢失了原有页面中的部分信息，如果丢失，则重新将丢失部分添加到动态页面中，得到整合后的页面信息。

2. 如权利要求 1 所述的方法，其特征在于所述 JavaScript 过滤库存储与抓取主题无关的 JavaScript 文件；所述 JavaScript 本地库存储外部 JavaScript 文件。

3. 如权利要求 2 所述的方法，其特征在于所述 JavaScript 过滤库还包括在线统计客户满意度，插入第三方广告推广代码功能的 JavaScript 文件。

4. 如权利要求 2 或 3 所述的方法，其特征在于根据所述 JavaScript 过滤库检验当前页面中请求的外部 JavaScript 文件，如果该文件在所述 JavaScript 过滤库中存在，则在当前页面对应的 DOM 对象相应位置设置无需加载标记，否则设置正常加载标记。

5. 如权利要求 1 所述的方法，其特征在于所述得到整合后的页面信息的方法为：遍历当前动态页面的动态页面库，针对每个动态页面，初始化一个空栈并对它的根标签执行：

a）取根标签下的第一个元素；

b）如果该元素不存在，弹出栈顶元素，然后获取栈顶元素；此时如果栈顶元素为空，则取下一个动态页面的根标签，否则取出该元素中的下一个元素；此时如果该元素的下一个元素为空，则弹出栈顶元素；此时如果栈为空，则取下一个动态页面的根标签，如果该元素为文本内容，则从当前页面的 DOM 中查询该文本内容；

c）如果标签内部包含标签，则将当前标签压入栈中，取出该标签下第一个元素，执行步骤 b）的处理；否则，取出该标签的文本内容，从当前页面的 DOM 中查询该文本内容；

d）如果从当前页面的 DOM 中找到查询的文本内容，则获取栈顶元素，如果栈顶元素为空，则取下一个动态页面的根标签；否则取出该元素中下一个元素，执行步骤 b）的处理；

e）将该文本内容放在根标签内并插入到当前页面主体标签 </body> 前面的位置，获取栈顶元素，并取其中的下一个元素，执行步骤 b）的处理；

f）如果下一个动态页面不存在，则结束处理。

6. 如权利要求 1 所述的方法，其特征在于该抓取服务器端包括一生成宿主对象的类，用于实例化相应宿主对象；所述宿主对象包括：HTML DOM 对象、CSS 对象、EVENT 对象、BOM 对象、XMLHttpRequest 对象。

7. 如权利要求 1 所述的方法，其特征在于利用 HTML 解析器生成当前页面的 DOM 对象；所述 HTML 解析器为 DOMParser 或 HTMLParser。

8. 如权利要求 1 或 2 所述的方法，其特征在于所述 JavaScript 本地库包括 jQuery 文

件集合、Ext 文件集合、Dojo 文件集合、Google Web ToolKit 文件集合、ProtoType 文件集合、YUI 文件集合，并检验文件的逻辑完整性；然后根据关键字为各个文件集合建立关键字与各个文件集合的一对一映射关系。

9. 一种面向主题的获取动态页面内容的系统，其特征在于包括 JavaScript 过滤库，JavaScript 本地库，宿主对象生成模块，页面爬行器，页面 DOM 对象生成模块，JavaScript 过滤器，JavaScript 解析器，信息整合模块；其中，

所述 JavaScript 过滤库，用于存储与抓取主题无关的 JavaScript 文件；

所述 JavaScript 本地库，用于存储外部 JavaScript 文件；

所述页面爬行器，用来获取目标页面初始源码；

所述 JavaScript 过滤器，用来维护需要解析的 JavaScript 文件以及确定当前页面中是否有无需分析的 JavaScript 文件；

所述宿主对象生成模块，用于提供各种宿主对象的定义和生成方法；

所述信息整合模块，用于检验获取的每一动态页面是否丢失了原有页面中的部分信息，如果丢失，则重新将丢失部分添加到动态页面中，得到整合后的页面信息。

10. 如权利要求 9 所述的系统，其特征在于还包括 JavaScript 本地库管理器，动态页面库管理器；其中，所述 JavaScript 本地库管理器包括若干 JavaScript 文件，并检验文件的逻辑完整性；所述动态页面库管理器，用来存储当前页面中通过 Ajax 请求获取的动态页面内容，为信息整合模块提供资源。

一种面向主题的获取动态页面内容的方法及系统

技术领域

[0001] 本发明属于网络数据采集技术领域，尤其涉及一种面向主题的获取动态页面内容的方法及系统。

背景技术

[0002] 当今是信息技术高速发展的时代，各种新事物层出不穷，网络信息呈爆炸趋势，如何从海量数据信息中获取有用的信息，在很多领域和行业中成为一种必需的支撑手段，能够最快最全地掌握行业领域相关的信息往往有利于做出恰当的抉择。与此同时，互联网信息爬取技术已经有了相当发展，尤其是在通用搜索领域，提供给用户的信息量比以前更大，处理用户请求的响应时间也大大提高；特定于客户需求的服务也越来越人性化，搜索内容也从文本、图片覆盖到当今的热门领域－视频。然而随着信息多元化发展，对于特定领域和特定主题的搜索需求越来越多，但通用搜索技术在这些领域的召回率和准确率通常无法满足特定的需求。于是垂直搜索技术应运而生，由于其提供的信息相关度高、同主题信息更多更深入、目标群体更为明确等特点，当前该领域的新技术与新应用越来越广泛。

[0003] 尽管垂直搜索与通用搜索一样面临着在互联网抓取过程中如何爬取动态页面信息的问题，但由于垂直搜索面向的用户更为具体，需求更为明确，因此如何提供更为全面的基于主题的动态页面信息是垂直搜索引擎的生存基础。目前在垂直搜索领域，如何获取动态页面方面已经取得了一些进展，例如在获取动态页面信息上多采用在抓取客户端中嵌入浏览器内核的方式来获取动态内容，然而该方式虽能获取到一定层次的动态内容，但由于浏览器解析过程中加载了页面布局模块、大量的兼容性代码、与主题无关的动态图片或 Flash 以及各种广告联盟的广告推广代码等与抓取主题无关的内容，因此时空效率比较低下。为此有人提出将 JavaScript 解析器嵌入到抓取过程中来实现动态抓取的方式，通常的做法是获取页面、构造 DOM、标记 DOM 中的 JavaScript、构造宿主对象、执行 JavaScript、返回动态页面。通过这种方式虽然减少了与抓取主题无关的页面布局代码、兼容性代码和图片操作代码等内容的加载解析，但仍然存在着一些缺点：1. 加载与主题无关的 JavaScript；2. 从远程主机请求外部 JavaScript 文件的时间效率低；3. 在执行 JavaScript 获取的动态页面内容丢失了原页面中存在的部分信息。鉴于这种状况，本申请在这里提供一种新的面向主题的获取动态页面内容的方法。

发明内容

[0004] 针对当前普遍采用的以嵌入 JavaScript 解析器的方式实现动态页面内容获取的方法存在的问题，本发明的目的在于提供一种面向主题的获取动态页面内容的方法及系统。

[0005] 本发明提出以下解决方案，通过建立 JavaScript 过滤库以过滤与主题无关的 JavaScript 文件，从而减少加载与主题无关的外部 JavaScript 文件；通过建立 JavaScript 本地库以便从本地加载原本需要从远程主机加载的 JavaScript 文件，从而减少与远程主

机之间的交互，进而减少加载外部 JavaScript 文件所需的时间；通过将原页面中存在而 JavaScript 解析器解析后的动态页面中缺少的信息加入到动态页面中来提高动态页面的完整性。

[0006] 本申请提供一种面向主题的获取页面动态内容的方法及系统，用以解决垂直搜索领域如何爬取动态页面信息的问题，技术方案如下：

[0007] 本申请提供一种面向主题的获取页面动态内容的方法，具体步骤如下：

[0008] 1. 建立 JavaScript 过滤库

[0009] 分析每一个抓取页面内部的 JavaScript 文件，根据其是否与抓取主题相关来决定是否将其相关信息存入到 JavaScript 过滤库，并增加库维护模块。

[0010] 2. 建立 JavaScript 本地库

[0011] 初始存入常见的 JavaScript 文件，然后根据每一次远程请求的外部 JavaScript 文件来完善该 JavaScript 本地库，并增加库维护模块。

[0012] 3. 提供生成各种宿主对象的类

[0013] 主要包括 JavaScript 语言本身不存在但在执行 JavaScript 代码时可能需要访问的对象。

[0014] 4. 获取页面信息

[0015] 基于正确的页面编码获取页面信息。

[0016] 5. 生成当前页面的 DOM 对象

[0017] 利用 DOMParser，HTMLParser 等 HTML 解析器生成该页面的 DOM 对象，当前页面中如果使用到相关宿主对象，则从步骤 3 提供的宿主类中实例化相应对象。

[0018] 6. 标记是否需过滤 JavaScript 文件

[0019] 根据步骤 1 中提供的 JavaScript 过滤库信息来检验页面中请求的外部 JavaScript 文件是否需要继续加载处理，如果该文件在 JavaScript 过滤库中存在，则在该页面对应的 DOM 对象中相应位置设置无需加载的过滤标记，否则设置正常加载标记。

[0020] 7. 标记是否需请求 JavaScript 本地库

[0021] 如果当前处理的 JavaScript 文件不是当前站点内部文件并且该文件在 JavaScript 本地库中存在，则设置从 JavaScript 本地库加载的本地标记，否则设置正常加载标记。

[0022] 8. 执行页面中的 JavaScript

[0023] 利用 JavaScript 解析器执行页面中的 JavaScript 获取动态页面信息，其中需要加载外部 JavaScript 文件时应根据步骤 6 和 7 中设置的过滤标记和本地标记来判断是否需加载该文件以及是否从 JavaScript 本地库加载该文件。

[0024] 9. 保存动态页面信息

[0025] 主要包含通过 Asynchronous JavaScript and XML（简称 Ajax）请求动态获取的页面信息。

[0026] 10. 获取整合后的页面信息

[0027] 通过执行 JavaScript 获取的动态页面可能会丢失一些原页面中存在的信息，将这部分丢失的信息重新加入到动态页面中，从而提高页面信息的完整性。

[0028] 本申请还提供一种面向主题的获取页面动态内容的系统，具体步骤如下：

[0029] 1. JavaScript 过滤库

[0030] 用页面中与抓取主题无关的 JavaScript 文件建立 JavaScrip 过滤库，并提供用于判别是否需过滤的模块。

[0031] 2. JavaScript 本地库

[0032] 初始库中存入常见的 javaScript 文件，然后基于每一次远程请求的 JavaScript 文件去完善该 JavaScript 本地库，并提供库维护模块。

[0033] 3. 宿主对象生成模块

[0034] 提供各种宿主对象的定义和生成方法。

[0035] 4. 页面爬行器

[0036] 以正确的页面编码获取页面源代码的模块。

[0037] 5. 页面 DOM 对象生成模块

[0038] 利用页面解析器生成当前页面的 HTML DOM 对象。

[0039] 6. JavaScript 过滤器

[0040] 基于 JavaScript 过滤库来判断当前页面中的 JavaScript 文件是否需要解析的模块。

[0041] 7. JavaScript 本地库管理器

[0042] 提供维护 JavaScript 本地库的模块以及本地库自动加载模块。

[0043] 8. JavaScript 解析器

[0044] 嵌入外部的 JavaScript 引擎为执行页面中的 JavaScript 脚本提供支持的模块。

[0045] 9. 动态页面管理器

[0046] 保存并维护通过异步 JavaScript 与 XML(Ajax) 请求的页面信息。

[0047] 10. 信息整合模块

[0048] 该模块主要用来检验动态生成的页面是否丢失了原有页面中的一些内容，如果丢失则重新将其添加到动态页面中。

[0049] 以上技术方案，总体过程是首先建立 JavaScript 本地库，JavaScript 过滤库和生成各种宿主对象，接着通过页面爬行器获取正确的页面信息，随后通过 HTML DOM 解析器生成该页面的 DOM 对象，在该过程中由 JavaScript 过滤器标记当前页面中的 JavaScript 文件是否需要解析，由 JavaScript 本地库管理器标记当前页面中的 JavaScript 文件是否需从 JavaScript 本地库加载，之后由 JavaScript 引擎执行当前页面中存在 JavaScript 文件，与此同时，将解析过程中由 Ajax 请求获取的页面保存在动态页面库中，接着将 JavaScript 解析器解析后的页面信息经由信息整合模块处理，最后返回更为完整的页面内容，其中 JavaScript 解析器中的数据流如图 6 所示。

[0050] 与现有技术相比，本发明的积极效果为：

[0051] 应用如上技术方案，垂直搜索引擎在爬取过程中可以获取更为完整的动态页面信息，而且与现有的调用浏览器内核或者调用 JavaScript 解析器的方案相比具有更高的时效性。

附图说明

[0052] 图 1 是本申请方法实现获取动态内容的方法流程；

[0053] 图 2 是本申请方法中页面爬行器的结构示意图；

[0054] 图 3 是本申请方法中 JavaScript 过滤库模块的结构示意图；

[0055] 图 4 是本申请方法中宿主对象生成模块的结构示意图；

[0056] 图 5 是本申请方法中 JavaScript 本地库模块的结构示意图；

[0057] 图 6 是本申请方法中 JavaScript 解析器模块的数据流程图。

具体实施方式

[0058] 如图 1 所示由本方法所构建的网页动态内容获取方法主要包括：1. 建立 JavaScript 过滤库；2. 建立 JavaScript 本地库；3. 提供生成各种宿主对象的类；4. 获取页面信息；5. 生成当前页面的 DOM 对象；6. 标记是否需过滤 JavaScript 文件；7. 标记是否请求 JavaScript 本地库；8. 执行页面中的 JavaScript 代码；9. 保存动态页面信息；10. 获取整合后的页面信息。

[0059] 该方法对应的系统为：

[0060] 1. JavaScript 过滤库，如图 3；2. JavaScript 本地库，如图 5；3. 宿主对象生成模块，如图 4；4. 页面爬行器，如图 2；5. 页面 DOM 对象生成模块；6. JavaScript 过滤器；7. JavaScript 本地库管理器；8. JavaScript 解析器；9. 动态页面库管理器；10. 信息整合模块。其中，页面爬行器用来获取目标页面初始源码，是整个系统持续运行的基础；JavaScript 过滤器用来维护需要解析的 JavaScript 文件以及确定当前的页面中是否有无需分析的文件，是减少与目标无关的资源加载的关键一步；宿主对象生成模块实现了 HTML DOM、EVENT、CSS、Browser Object Model、XMLHttprequest 规范定义的接口，是 JavaScript 解析器正确解析、执行页面中 JavaScript 代码的关键；JavaScript 本地库管理器提供了大多数常用的 JavaScript 文件，尽可能将远程请求转换为本地请求，以便减少与远程主机的交互，是减少加载外部 JavaScript 文件的关键所在；动态页面库管理器主要用来存储当前页面中通过 Ajax 请求获取的动态页面内容，为信息整合模块提供必须的资源；信息整合模块主要用于提高目标内容的完整性。

[0061] 为了便于本领域工程技术人员实施，下面详细说明该方法的实施步骤：

[0062] 步骤一：建立 JavaScript 过滤库

[0063] 基于主题建立与目标内容无关的 JavaScript 过滤库，该库中主要包含两类可执行的文件：1. 与主题明显无关的 JavaScript 文件，例如用来改变页面布局的 JavaScript 文件；2. 用于在线统计客户满意度，插入第三方广告推广代码等功能的文件，例如在线统计客户满意度代码 ForSee Results Survey Code，百度联盟、淘宝联盟以及 Google AdSense 提供的以 JavaScript 形式实现的广告服务。

[0064] 该库中内容的选择，主要使用逐一分析加入到抓取 URL 集中的页面，通过判断其中所包含的外部 JavaScript 文件是否与抓取主题相关来决定是否将其存入 JavaScript 过滤库。

[0065] 在选择过程中需要注意以下几点：

[0066] (1) 对于一个主题而言无用的第三方 JavaScript 文件可能对于另一个主题来说是必须的。

[0067] (2) 如果第三方 JavaScript 文件包含多个 JavaScript 文件，需要确保包含的文件

具有完整的逻辑依赖性。

[0068] (3) 存储第三方 JavaScript 文件时以该 JavaScript 文件名称中的关键词部分命名。关键词部分，可以以此种方式来实现，比如 ForSee 集合，通常以 *_ForSee_* 形式来命名，那么此时的关键词部分就是指 ForSee。

[0069] 另外，针对该 JavaScript 过滤库增加一些辅助功能：

[0070] A. 增加增删改查方法，以便于提供人工更新库中信息的接口。

[0071] B. 过滤外部 JavaScript 文件

[0072] 在加载外部 JavaScript 文件时，通过文件名称中的关键词部分去搜索 JavaScript 过滤库，如果能在该库中找到匹配的 JavaScript 文件，则在 DOM 树相应位置设置过滤标记 1（即无需加载该文件），否则设置正常加载标记 0。

[0073] 步骤二：建立 JavaScript 本地库

[0074] 该库初始应该加入 jQuery，Ext，Dojo，Google Web ToolKit，ProtoType，YUI 等文件集合，根据关键字为各个文件集合建立关键字与文件集合的一对一映射关系，以便于加载时准确快捷。随后根据每一次去当前站点外请求的 JavaScript 文件名称的关键字部分来检验是否在 JavaScript 本地库中存在，如果不存在则发出 Ajax 请求获取该文件并保存到 JavaScript 本地库；如果存在则不用发送请求，直接进行本地下载。为方便后面的环节使用该 JavaScript 本地库，需为该库增加增删查的维护功能和标记是否从该库加载的功能。

[0075] A. 增删查的功能具体如下：

[0076] (1) 增加 JavaScript 文件到 JavaScript 本地库，需要检验该文件的逻辑完整性，比如增加 jQuery UI 子文件，应保证增加 jQuery-Core 文件。

[0077] (2) 从 JavaScript 本地库中删除 JavaScript 文件，需要保证文件的逻辑依赖性，比如首先删除 jQuery UI，而不能首先删除 jQuery-Core，当且仅当没有其他文件对 jQuery-Core 有依赖后才可以删除 jQuery-Core。

[0078] (3) 每个文件集合包含的子文件，以及子文件之间的依赖关系通过 XML 文件进行配置。

[0079] B. 检验是否从 JavaScript 本地库加载文件

[0080] 依次顺序检验当前 DOM 树中每一个 JavaScript 文件的对应结点处设置的过滤标识来验证该文件是否需要被加载，此时分两种情况：

[0081] (1) 如果是过滤标识为 1，即无需加载，则不予处理本请求，直接处理当前 DOM 树中下一个 JavaScript 文件对应的结点，如果当前结点是 DOM 树中最后一个结点，则终止整个检验过程；

[0082] (2) 如果是正常加载标识 0，则根据该 JavaScript 文件的关键字部分在 JavaScript 本地库中查询其中是否存在该文件，此时分两种情况：

[0083] i. 如果存在，则设置该文件在对应 DOM 树结点的标识属性为 2，即从 JavaScript 本地库加载；

[0084] ii 否则继续处理当前 DOM 树中下一个 JavaScript 文件对应的结点，如果当前节点是 DOM 树中最后一个结点，则终止整个检验过程。

[0085] JavaScript 文件的关键字部分根据步骤二 B 项中的方式来确定。该部分加载文件

的方式与步骤一中页面爬行器采用的方式大体一样，不同之处在于加载 JavaScript 文件时，直接采用 UTF-8 编码方式，不进行编码识别。

[0086] 步骤三：提供生成各种宿主对象的类

[0087] A. 实现 HTML DOM 对象

[0088] 基于 DOM 规范实现相应等级的 DOM 对象，该模块应该包含相应等级对应规范中定义的所有 DOM 对象，其中要保证常见对象的所有属性以及所有对象的常用属性必须实现，以确保在 JavaScript 解析器在执行时能够找到相应的 DOM 对象。

[0089] B. 实现 CSS 对象

[0090] 由于本申请方法以及系统中只关注页面中的目标内容，因此涉及页面布局、展示方式的 CSS 只提供基本的实现即可，该实现中至少应该包括如下部分：CSS 样式转文本 property 的操作方法、CSS 样式与文本转换的方法、CSS 支持的样式集合、CSS 选择器规则以及 CSS 样式表的解析模块。注意，如果不提供该实现会造成 JavaScript 解析中找不到相应属性或方法等异常。

[0091] C. 实现 EVENT 对象

[0092] 该模块部分，应该实现 Event 注册、Event 派发功能，主要用于触发 onload、onunload、onerror 等以 on 开头的事件。

[0093] D. 实现 BOM 对象

[0094] 该模块主要用于 JavaScript 解析器在执行 JavaScript 代码时访问浏览器对象的情况，必须实现窗口 (Window)，历史 (History)，导航器 (Navigator)，屏幕 (Screen)，文档 (Document)，位置 (Location) 六个对象，其中窗口 (Window)，历史 (History)，文档 (Document)，位置 (Location) 这四个常用对象的属性和方法实现要全面，对于未实现部分要给出异常信息提示。

[0095] E. 实现 XMLHttpRequest 对象

[0096] 该对象的实现中应该包括，open，send，setRequestHeader，getResponseHeader，getResponseHeaders 方法，其中 send 方法中需要对 Cookie 做相应处理。

[0097] 步骤四：获取页面信息

[0098] 需要抓取的 url 地址来源于特定于主题的 URL 集合，代码获取部分与传统互联网抓取客户端类似，主要包括两个部分：

[0099] A. 识别页面编码

[0100] 首先通过 HTTP 响应头获取 Content-Type 字段，如果该字段中不包含 charset 字符，则以 GBK，UTF-8 等字符集中任意一种作为当前字符集读取页面的一部分代码，然后查找其中的 charset 字符串从中截取 charset，如果仍然不能确定字符集，则默认当前字符集为 UTF-8。

[0101] B. 读取页面

[0102] 本步骤与传统互联网客户端采用的技术一样，通过 URL(Uniform Resource，统一资源定位符）地址读取该地址对应的页面代码内容。

[0103] 步骤五：生成当前页面的 DOM 对象

[0104] 利用 HTMLParser 等 HTML 解析器解析当前页面，该解析器具有在 DOM 树的每个 javascript 结点上增加一个标识 flag 属性的功能，其中 flag 意义为：0 代表正常；1 代表

需过滤；2 代表需向本地 JavaScript 框架库请求该文件。

[0105] 步骤六：标记是否需过滤 JavaScript 文件

[0106] 遍历当前页面的 DOM 树，对所遇到的 JavaScript 结点利用步骤一辅助功能 B 设置过滤标识。

[0107] 步骤七：标记是否请求 JavaScript 本地库

[0108] 遍历当前页面的 DOM 树，对所遇到的 JavaScript 结点利用步骤二 B 中的方法设置是否从 JavaScript 本地库加载标识。

[0109] 步骤八：执行页面中的 JavaScript 代码

[0110] 这里可以采用现有的 JavaScript 解析器，比如 SpiderMonkey，Rhino 或者 Google v8 等 JavaScript 引擎，采用其中任何一种引擎，都应该首先将步骤三中所描述的所有宿主对象加载进去。需要执行的 JavaScript 代码具体分为两部分：

[0111] (1) 在构造 DOM 对象过程中加载外部 JavaScript 文件，或执行页面中存在的代码段或存在于属性值内部的代码语句

[0112] (2) 在构造 DOM 结束后，触发 onload 等以 on 开头的注册事件所包含的代码，其中既包括页面中调用注册 onload 等事件，也包括加载的外部 JavaScript 文件中包含的 onload 等事件。

[0113] 其中，JavaScript 文件是指通过 HTML 标签的 src 属性加载的外部 JavaScript 文件，代码段是指存在于 HTML 标签 <script></script> 之间的代码，代码语句是指存在于 HTML 标签属性值中以 JavaScript：方式开头的语句。

[0114] 在执行代码过程中，根据每个需要从外部加载的 JavaScirpt 文件所对应的 DOM 结点上的 flag 属性来决定是否需要加载和是否需要从 JavaScipt 本地库加载该文件，如果需要则从 JavaScript 本地库加载，否则按照原有方式请求远程主机。

[0115] 步骤九：保存动态页面信息

[0116] 将当前页面中所涉及到的所有 JavaScript 代码中发出的 Ajax 请求获取到的页面保存为当前页面的动态页面库。对于库中的每个页面，只有 <body> 标签内部部分对于主题来说是有意义的，因此提取出每一个页面 <body> 标签内部的内容，嵌套在 标签内。

[0117] 步骤十：获取整合后的页面信息

[0118] 对于每个当前页面都建立一个动态页面库，遍历当前动态页面的动态页面库，针对每个动态页面，对它的根 标签执行以下算法，其中该算法根据深度优先的顺序来取下一个标签，另需要初始化一个空栈：

[0119] (1) 取根 div 标签下的第一个元素；

[0120] (2) 如果该元素不存在，弹出栈顶元素，然后获取栈顶元素，此时如果栈顶元素为空，转步骤 (7)，否则取出该元素中的下一个元素，此时如果该元素的下一个元素为空则弹出栈顶元素，此时如果栈为空，则转步骤 (7)；如果该元素为文本内容，则转向步骤 (4)；

[0121] (3) 如果标签内部包含标签，则将当前标签压入栈中，取出该标签（即当前标签）下第一个元素，转步骤 (2)，否则取出该标签的文本内容；

[0122] (4) 在前面已经构造好的 DOM（即当前页面的 DOM 树）中查询该本文内容；

[0123] (5) 如果找到，则获取栈顶元素，如果为空，则转向步骤 (7)，否则取出该元素中下一个元素，转向步骤 (2)；

[0124] (6) 将该文本内容放在根标签 内并插入到当前页面主体标签 </body> 前面的位置，获取栈顶元素，并取其中的下一个元素，转向步骤 2)；

[0125] (7) 取下一个动态页面的根 ；

[0126] (8) 如果下一个动态页面存在，则转步骤 (1)，否则结束处理。

[0127] 最终获取 JavaScript 解析器执行整个 DOM 操作后获取的动态页面源码与执行过程中未被插入进当前 DOM 的内容的结合页面信息，该页面信息与传统 JavaScript 解析后的页面相比，提供了更为完整的页面信息。

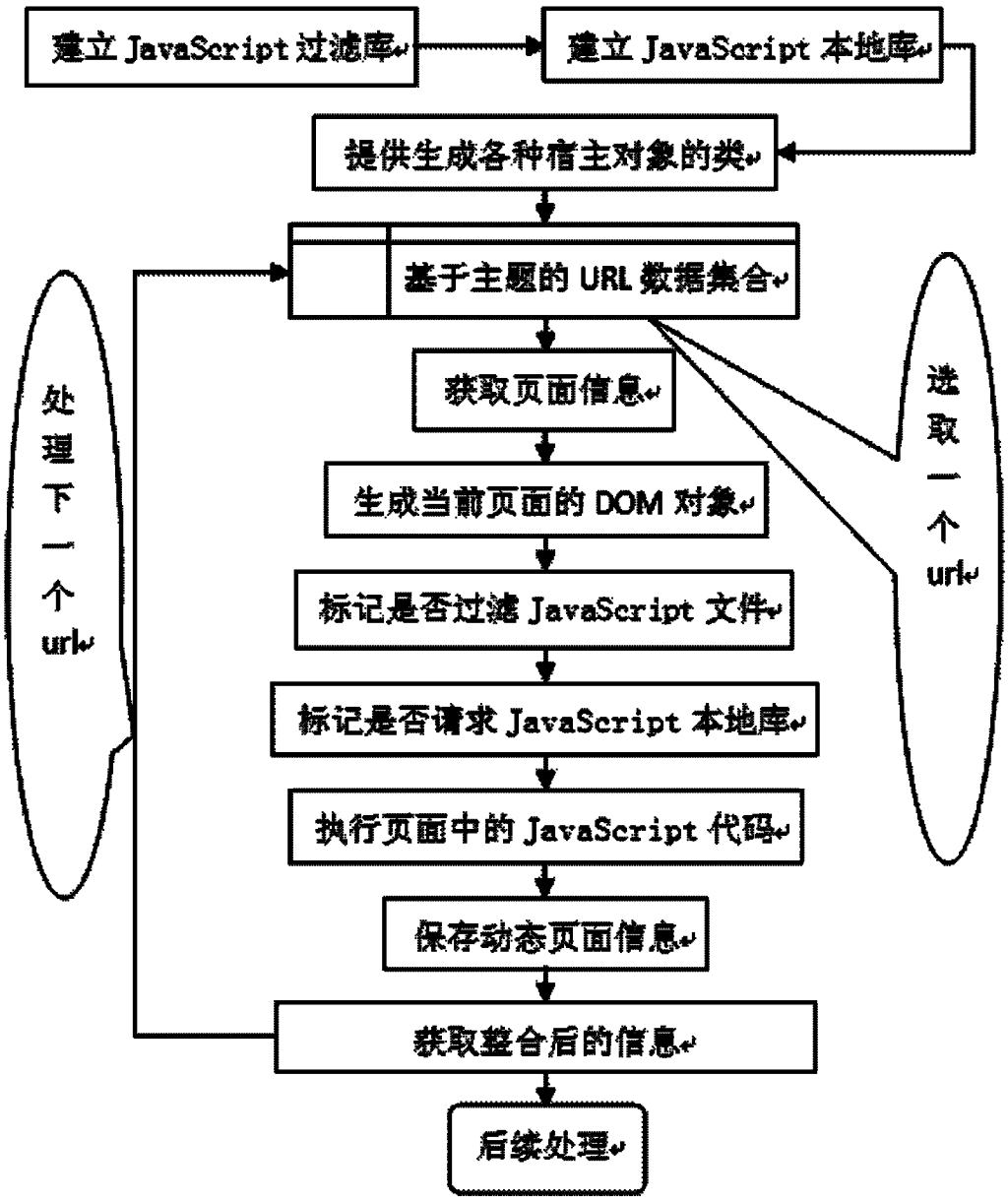

建立 JavaScript 过滤库
建立 JavaScript 本地库
提供生成各种宿主对象的类
基于主题的 URL 数据集合
获取页面信息
生成当前页面的 DOM 对象
标记是否过滤 JavaScript 文件
标记是否请求 JavaScript 本地库
执行页面中的 JavaScript 代码
保存动态页面信息
获取整合后的信息
后续处理
处理下一个 url
选取一个 url

图 1

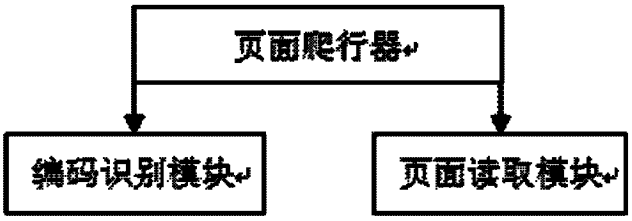

页面爬行器
编码识别模块
页面读取模块

图 2


JavaScript 过滤库模块
JavaScript 过滤库维护模块
JavaScript 文件过滤判定模块

图 3

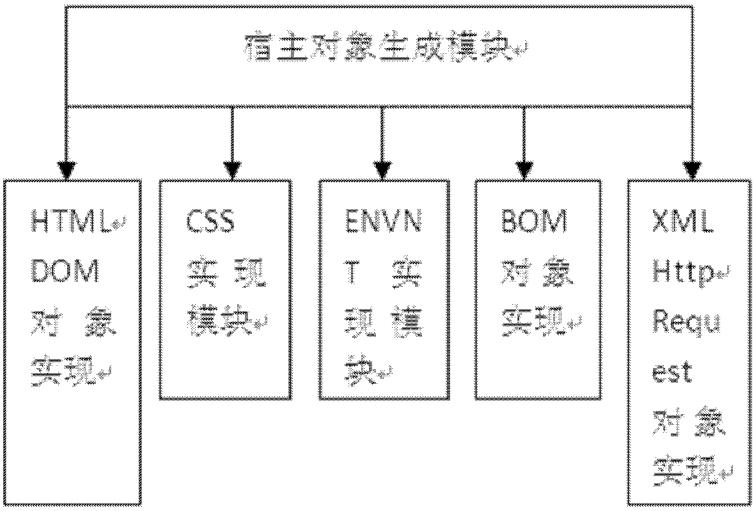

宿主对象生成模块
HTML DOM 对象实现
CSS 实现模块
ENVNT 实现模块
BOM 对象实现
XML Http Request 对象实现

图 4

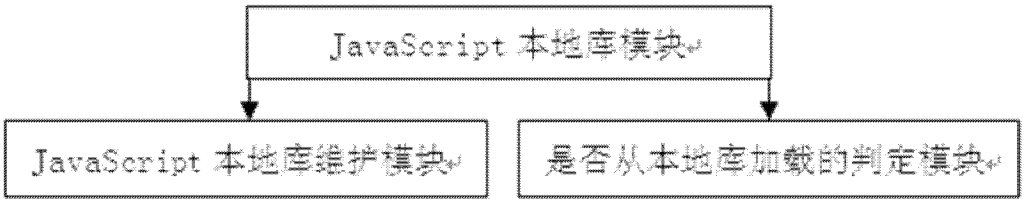

JavaScript 本地库模块
JavaScript 本地库维护模块
是否从本地库加载的判定模块

图 5

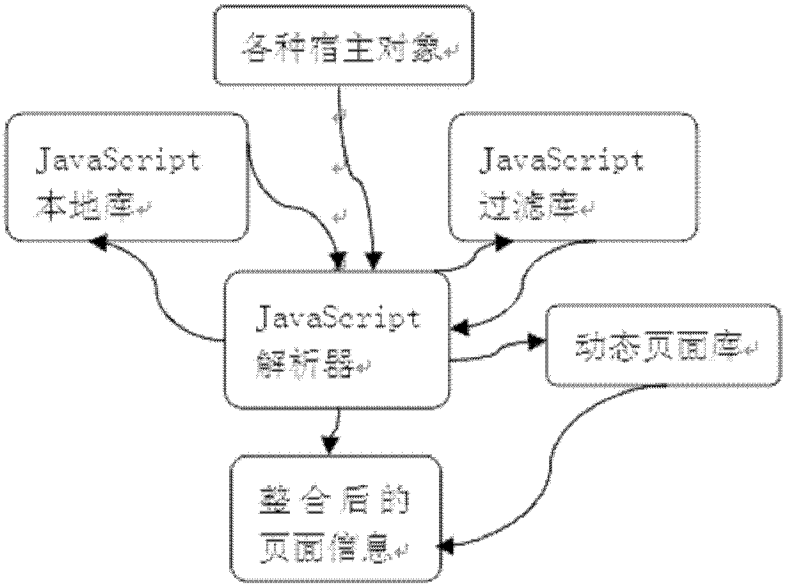

各种宿主对象
JavaScript
本地库
JavaScript
过滤库
JavaScript
解析器
动态页面库
整合后的
页面信息

图 6