



(12) 发明专利

(10) 授权公告号 CN 1894658 B

(45) 授权公告日 2012.02.29

(21) 申请号 200480015737.5

G10L 21/00(2006.01)

(22) 申请日 2004.06.07

(56) 对比文件

(30) 优先权数据

60/476,815 2003.06.06 US

US 2003/0078973 A1, 2003.04.24, 全文.

WO 02/44887 A2, 2002.06.06, 说明书第7页
第17-31行、第8页第1-13行和第27页第1-27
行,附图1-3和5.

(85) PCT申请进入国家阶段日

2005.12.06

US 6313854 B1, 2001.11.06, 说明书第3栏

第17-34行、第4栏第56行至第5栏第6行,附图
6和7.

(86) PCT申请的申请数据

PCT/US2004/018109 2004.06.07

(87) PCT申请的公布数据

W02004/109471 EN 2004.12.16

审查员 解欣

(73) 专利权人 纽约市哥伦比亚大学托管会

地址 美国纽约

(72) 发明人 迈克尔·L·查尼

(74) 专利代理机构 北京集佳知识产权代理有限公司

公司 11227

代理人 徐谦 杨红梅

(51) Int. Cl.

G06F 3/16(2006.01)

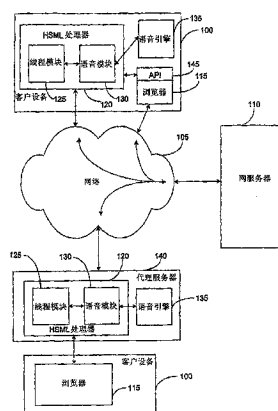
权利要求书 3 页 说明书 13 页 附图 2 页

(54) 发明名称

用于语音激活网页的系统和方法

(57) 摘要

一种用于处理对规定网页的数据的语音请求的方法,该语音请求包括基于规则的语法语句,所述方法包括:标识与所述基于规则的语法语句关联的语法;确定被标识为与所述语法关联的第一连接是否被规定在标识一个或多个连接的数据结构中;如果被标识为与所述语法关联的第一连接被规定在所述数据结构中,则在能够呈现网页并且与所述第一连接关联的窗口的至少部分中处理所述语音请求;以及如果被标识为与所述语法关联的第一连接未被规定在所述数据结构中,则:打开新连接;将所述新连接与能够呈现网页的新窗口关联;以及在能够呈现网页的所述新窗口的至少部分中处理所述语音请求。



1. 一种用于处理对规定网页的数据的话音请求的方法,该话音请求包括基于规则的语法语句,所述方法包括:

标识与所述基于规则的语法语句关联的语法;

确定被标识为与所述语法关联的第一连接是否被规定在标识一个或多个连接的数据结构中;

如果被标识为与所述语法关联的第一连接被规定在所述数据结构中,则在能够呈现网页并且与所述第一连接关联的窗口的至少部分中处理所述话音请求;以及

如果被标识为与所述语法关联的第一连接未被规定在所述数据结构中,则

打开新连接;

将所述新连接与能够呈现网页的新窗口关联;以及

在能够呈现网页的所述新窗口的至少部分中处理所述话音请求。

2. 根据权利要求1所述的方法,其中所述话音请求包括这样的请求,其包括具有短语部分、命令部分和标签部分的基于规则的语法语句。

3. 根据权利要求1所述的方法,该方法进一步包括:将所述新连接添加给所述数据结构并且将所述新连接与新语法关联。

4. 根据权利要求1所述的方法,该方法进一步包括:如果所述新连接未被规定在所述数据结构中,则所述数据结构被更新以将所述新连接与所述新窗口关联。

5. 根据权利要求4所述的方法,其中所述新窗口包括根据用户偏好选择的窗口。

6. 根据权利要求4所述的方法,其中所述新窗口包括与所述新连接关联的计算设备所加载的最后的窗口。

7. 根据权利要求4所述的方法,其中所述新窗口包括与所述新连接关联的计算设备处的活动窗口。

8. 根据权利要求1所述的方法,其中标识语法包括标识根据用户偏好的语法。

9. 根据权利要求1所述的方法,其中标识语法包括标识与由关联于一连接的计算设备所加载的最后窗口相关联的语法。

10. 根据权利要求1所述的方法,其中标识语法包括标识与关联于一连接的计算设备处的活动窗口相关联的语法。

11. 根据权利要求1所述的方法,其中在其中处理所述话音请求的所述窗口包括浏览器窗口。

12. 根据权利要求1所述的方法,其中在其中处理所述话音请求的所述窗口的至少部分包括框架。

13. 根据权利要求1所述的方法,其中所述话音请求包括对复杂数据的请求。

14. 根据权利要求13所述的方法,其中所述话音请求包括对在网页中嵌入的复杂数据的请求。

15. 根据权利要求13所述的方法,该方法进一步包括检查与所述话音请求关联的头部或脚部以确定所请求的数据的类型。

16. 根据权利要求15所述的方法,其中检查头部或脚部包括检查规定mime类型的头部或脚部以确定所请求的数据的类型。

17. 根据权利要求13所述的方法,该方法进一步包括检查与所述话音请求关联的文件

名或文件扩展以确定所请求的数据的类型。

18. 根据权利要求 1 所述的方法,该方法进一步包括:响应于窗口的关闭,从所述数据结构中去除与被关闭的所述窗口关联的连接标识。

19. 一种用于处理对规定网页的数据的话音请求的系统,该语音请求包括基于规则的语法语句,所述系统包括:

用于标识与所述基于规则的语法语句关联的语法的装置;

用于确定被标识为与所述语法关联的第一连接是否被规定在标识一个或多个连接的数据结构中的装置;

用于如果被标识为与所述语法关联的第一连接被规定在所述数据结构中,则在能够呈现网页并且与所述第一连接关联的窗口的至少部分中处理所述语音请求的装置;以及

用于如果被标识为与所述语法关联的第一连接未被规定在所述数据结构中,则进行如下处理的装置:

打开新连接;

将所述新连接与能够呈现网页的新窗口关联;以及

在能够呈现网页的所述新窗口的至少部分中处理所述语音请求。

20. 根据权利要求 19 所述的系统,其中所述语音请求包括这样的请求,其包括具有短语部分、命令部分和标签部分的基于规则的语法语句。

21. 根据权利要求 19 所述的系统,该系统进一步包括用于将所述新连接添加给所述数据结构并且将所述新连接与新语法关联的装置。

22. 根据权利要求 19 所述的系统,该系统进一步包括用于如果所述新连接未被规定在所述数据结构中,则所述数据结构被更新以将所述新连接与所述新窗口关联的装置。

23. 根据权利要求 19 所述的系统,其中所述对规定网页的数据的话音请求包括对复杂数据的请求。

24. 一种用于处理对规定网页的数据的话音请求的方法,该方法包括:

从第一连接接收规定网页的数据,该数据包括具有短语部分、命令部分和标签部分的基于规则的语法语句;

确定所述第一连接是否被规定在标识一个或多个连接的数据结构中;

从标识一个或多个语法的数据结构中选择与所述第一连接关联的语法;

如果所述第一连接被规定在标识一个或多个连接的数据结构中,则在能够呈现网页并与所述第一连接关联的第一窗口的至少部分中使用所选语法来处理所述基于规则的语法语句;以及

如果所述第一连接未被规定在标识一个或多个连接的数据结构中,则从规定一个或多个语法的数据结构中选择与所述第一连接关联的新语法,并在与所述第一连接关联的新窗口的至少部分中使用所选语法来处理所述基于规则的语法语句。

25. 根据权利要求 24 所述的方法,该方法进一步包括:将所述第一连接添加到规定一个或多个连接的数据结构并且将所述第一连接与规定一个或多个语法的数据结构中的新语法相关联。

26. 一种用于处理对规定网页的数据的话音请求的方法,该方法包括:

接收对规定网页的数据的话音请求,该语音请求包括具有短语部分、命令部分和标签

部分的基于规则的语法语句；

确定一窗口是否与所述话音请求关联；

如果一窗口与所述话音请求关联，则在与所述话音请求关联的窗口的至少部分中处理所述话音请求；以及

如果没有窗口与所述话音请求关联，则在新窗口的至少部分中处理所述基于规则的语法语句。

27. 根据权利要求 26 所述的方法，该方法进一步包括：标识与所述话音请求关联的连接并且将该连接与所述新窗口相关联。

用于语音激活网页的系统和方法

[0001] 版权通知

[0002] 本专利文档的公开的一部分包含受到版权保护的材料。版权所有者并不反对对专利文档或专利公开的任何一个的影印复制,因为它出现在专利和商标局专利文件或记录中,然而无论如何,保留全部版权权利。

[0003] 相关申请

[0004] 本申请要求 2003 年 6 月 6 日提交的题为 SYSTEM AND METHOD FOR VOICE ACTIVATING WEB PAGES 的美国临时申请 No. 60/476,815 的优先权,该申请在此以整体引入作为参考。

[0005] 本申请亦涉及以下未决专利申请;其每个在此以整体引入作为参考:

[0006] 专利合作条约国际申请 No. PCT/US01/45223, 题为 A METHOD AND SYSTEM FOR VOICE ACTIVATING WEB PAGES, 2001 年 11 月 30 日提交, 2002 年 6 月 6 日公开, 国际公开 No. WO/02/44887A2; 以及

[0007] 每个临时专利 No. 60/250,809, 题为 A METHOD AND SYSTEM FOR VOICE ACTIVATING WEB PAGES, 2000 年 12 月 1 日提交。

背景技术

[0008] 在此公开的发明总体上涉及语音激活网页。更具体而言,本发明提供用于语音激活包含网页和复杂网页的多个窗口的系统和方法。

[0009] 在过去的十年中,自动语音识别(“ASR”)系统已经前进到可通过安装在中等价格个人计算机和工作站上的 ASR 系统获得高程度的识别精度的点。这已经导致可用于消费者和工业应用的 ASR 系统的数量的增加。

[0010] ASR 系统依赖于语音语法来识别经由麦克风输入的声音命令并作用于那些命令。语音语法分为两种类别:基于规则的语法和自由语音语法。基于规则的语法允许对有限预定短语集的识别。如果话语(utterance)导致一事件或事件集发生,则每个基于规则的语法被调用。如果经由麦克风输入的话语匹配对应于存储在所述预定短语集内的短语的语音模板,则基于规则的语法被调用。例如,用户可以在以字处理程序编辑文档的同时说“保存文件”以调用保存命令。

[0011] 另一方面,自由语音语法识别诸如商务英语的给定域中的大的字集。这些语法通常用于口述应用。这些系统的一些例子是 Dragon Naturally Speaking 和 IBM ViaVoice 7Millennium。ASR 系统亦已经引入了文本到语音(“TTS”)能力,其使 ASR 系统能通过使用合成语音来说出图形再现的文本。例如,ASR 系统可通过扬声器朗读字处理器内的加亮段落。

[0012] ASR 系统已经与网浏览器结合以创建语音使能的网浏览器。语音使能的网浏览器允许用户通过使用调用基于规则的语法的话音命令来导航于(navigate)因特网。这些浏览器所使用的一些话音命令包括使软件执行网浏览器所使用的传统命令的话语。例如,如果用户将“首页(home)”说到麦克风,则语音使能的浏览器将执行与用户点击语音使能的

网浏览器的“首页”按钮时话音使能的网浏览器将执行的例行程序相同的例行程序。

[0013] 另外,一些话音使能的网浏览器基于网页内容来创建基于规则的语法。在网页被下载和显示时,一些语音使能的网浏览器基于包含在网页内的链接来创建基于规则的语法。例如,如果网页显示链接“公司首页(company home)”,则这样的话音使能的网浏览器将创建在该网页被显示时有效的基于规则的语法,使得如果用户将短语“公司首页”发声到麦克风,则话音使能的网浏览器将显示与所述链接关联的网页。这种途径的一个缺点是从网页内容产生的规则在长时间段内是固定的,这是因为网页并不经常重新设计。另外,基于规则的语法是从主要想用于可视显示的网页内容产生的。实际上,这些系统限制了用户说出在屏幕上所显示的。

[0014] 网页亦可引入使声音输出的音频元素。目前,网页可以以两种方式将音频元素引入其网页中。引入音频元素的第一种方式是使用音频波文件内容来将人发出的话音提供给网页。使用音频波文件允许网页设计者独立设计网页的可视和音频部分,但该自由度和增加的功能性导致高价格。在因特网上将二进制声音文件传递给末端用户所需的带宽极大。

[0015] 引入音频元素的第二种方式是平衡 ASR 系统的功能性。话音使能的网浏览器可以以如此方式来利用 ASR 系统的 TTS 功能性,以使计算机“说出”网页的内容。使用这种途径使察看有或没有音频元素的网页所需的带宽近似相同,但将网浏览器可以讲述的内容的主题限制于网页的内容。

[0016] 话音 XML(VXML) 为网页设计者提供了另一个选择。VXML 允许用户通过使用典型地在电话上使用的音频命令来单独导航于网站。VXML 所需要的是, TTS 翻译器通过将可视网页翻译成网页的音频表达来将网页读给用户。用户通过说出用户想要跟随的链接来导航于该网。利用这种途径,用户可通过仅使用用户的话音来导航于因特网,但音频内容典型地从主要设计用于可视解释的网页内容产生;并且可视接口被从用户的经验中去除。

[0017] 这样,通过发明在专利合作条约国际申请 No. PCT/US01/45223 中进一步描述的系统,发明人解决了独立创建不需要大量传输带宽并结合网页可视组件而存在的网页音频组件的需要,该申请在此以整体引入作为参考。’45223 申请公开了尤其用于激活单个、简单可视网页中的话音内容的系统和方法。

[0018] ’45223 申请的系统经由一代理服务器来控制网页内的语音内容,该代理服务器具有对与浏览器相同的计算机设备(或该设备的声音输出通道)的访问。该代理服务器检查用于语音内容的数据,与此同时将所有其它所请求的数据馈送给浏览器。在用户点击链接的情况下,由 URL 规定的浏览器所请求的数据通过代理服务器传递到规定的网服务器。来自该网服务器的所请求的材料传递回到浏览器。

[0019] 然而,在用户经由语音事件请求新的 URL 的情况下,代理服务器经由浏览器请求来自规定网服务器的该数据(例如,执行网服务器上的规定程序或其它命令)。这样,所得到的数据需要被“推”回到浏览器。这是通过使用在’45223 申请中进一步描述的 multipart/x-mixed-replacemime 类型来完成的。这种类型使浏览器保持浏览器和代理服务器之间的连接打开,并且继续接受数据,直到给定信号或其它令牌被发送或连接被关闭。例如,终止可由于从浏览器请求的新“点击”或由于新页中没有语音内容而发生。终止的情形进一步描述在如’45223 申请的图 3B 中所示并且进一步在其中描述的真值表中。

[0020] 发明人已经确定了在此进一步描述的附加改进,包括如何扩展系统以对包含复

杂、聚集内容或来自同时工作的多个页的内容的网页起作用,例如在给定浏览器的多个实例或窗口中,或者在给定窗口内的多个框架内。

发明内容

[0021] 本发明尤其解决了以上讨论的有关话音激活网页的问题。

[0022] 在一个实施例中,所述系统包括一种用于处理对规定网页的数据的话音请求的方法,该话音请求包括基于规则的语法语句,所述方法包括:标识与所述基于规则的语法语句关联的语法;确定被标识为与所述语法关联的第一连接是否被规定在标识一个或多个连接的数据结构中;如果被标识为与所述语法关联的第一连接被规定在所述数据结构中,则在能够呈现网页并且与所述第一连接关联的窗口的至少部分中处理所述话音请求;以及如果被标识为与所述语法关联的第一连接未被规定在所述数据结构中,则:打开新连接;将所述新连接与能够呈现网页的新窗口关联;以及在能够呈现网页的所述新窗口的至少部分中处理所述话音请求。在一些实施例中,所述话音请求包括这样的请求,其包括具有短语部分、命令部分和标签部分的基于规则的语法语句。

[0023] 在一些实施例中,所述新连接被添加给所述数据结构并且被与新语法关联。在一些实施例中,所述数据结构被更新以将所述新连接与所述新窗口关联。所述系统亦包括响应于窗口的关闭,从所述数据结构中去除与被关闭的所述窗口关联的连接的标识。

[0024] 在一些实施例中,所述新窗口包括与新连接关联的计算设备所加载的最后的窗口,与新连接关联的计算设备处的活动窗口,或者是根据用户偏好来选择的。在一些实施例中,所述语法是与由关联于一连接的计算设备所加载的最后窗口相关联的语法,与关联于一连接的计算设备处的活动窗口相关联的语法,或者是根据用户偏好来选择的。在其中处理所述话音请求的所述窗口可包括浏览器窗口或窗口的部分,如框架。

[0025] 在一些实施例中,所述话音请求包括对复杂数据的请求,如对在网页中嵌入的复杂的数据的请求。在一些实施例中,所述系统检查与所述请求关联的头部或脚部(header/footer)以确定所请求的数据的类型,例如规定 mime 类型的头部或脚部。在一些实施例中,所述系统检查与所述请求关联的文件名或文件扩展以确定所请求的数据的类型。

[0026] 在一些实施例中,所述系统包括一种用于处理对规定网页的数据的话音请求的方法,该方法包括:从第一连接接收规定网页的数据,该数据包括具有短语部分、命令部分和标签部分的基于规则的语法语句;确定所述第一连接是否被规定在标识一个或多个连接的数据结构中;从标识一个或多个语法的数据结构中选择与所述第一连接关联的语法;如果所述第一连接被规定在标识一个或多个连接的数据结构中,则在能够呈现网页并与所述第一连接关联的第一窗口的至少部分中使用所选语法来处理所述基于规则的语法语句;以及如果所述第一连接未被规定在标识一个或多个连接的数据结构中,则从规定一个或多个语法的数据结构中选择与所述第一连接关联的新语法,并在与所述第一连接关联的新窗口的至少部分中使用所选语法来处理所述基于规则的语法语句。

[0027] 在一些实施例中,所述系统亦将所述第一连接添加到规定一个或多个连接的数据结构并且将所述第一连接与规定一个或多个语法的数据结构中的新语法相关联。

[0028] 在一些实施例中,所述系统包括一种用于处理对规定网页的数据的话音请求方法,该方法包括:接收对规定网页的数据的话音请求,该话音请求包括具有短语部分、命令

部分和标签部分的基于规则的语法语句;确定一窗口是否与所述话音请求关联;以及如果一窗口与所述话音请求关联,则在与所述请求关联的窗口的至少部分中处理所述话音请求。

[0029] 在一些实施例中,如果没有窗口与所述话音请求关联,则所述系统在新窗口的至少部分中处理所述基于规则的语法语句。在一些实施例中,所述系统标识与所述话音请求关联的连接并且将该连接与所述新窗口相关联。

附图说明

[0030] 本发明在附图的图中图示,所述附图旨在是示例性的而非限制性的,其中相同的参考旨在表示相同或对应的部分,并且其中:

[0031] 图 1 是根据本发明一个实施例的执行对网页的话音激活的系统的网络体系结构的块图;并且

[0032] 图 2 是根据本发明一个实施例的执行对网页的话音激活的方法的流程图。

具体实施方式

[0033] 参考图 1,所呈现的是本发明的实施例。图 1 呈现根据本发明一个实施例的执行对网页的话音激活的系统的网络体系结构的块图。如所示,所述系统包括连接到网络 105 的一个或多个客户设备 100、网服务器 110、浏览器 115、超语音置标语言(hyper-speech markup language) (“HSML”) 处理器 120、线程模块(thread module) 125、语音模块 130、语音引擎 135、代理服务器 140 和浏览器应用程序接口 (“API”) 145。

[0034] 客户设备 100 可包括例如客户计算机、蜂窝电话、个人数字助理 (“PDA”)、输入板(tablet) PC 以及其它配备有处理器并能够支持如在此和在’ 45223 申请中进一步描述的浏览器 115 软件模块的类似电子设备。客户设备 100 经由网络 105 在通信上耦合到网服务器 110 或其它类似服务器。例如,在一个实施例中,客户设备 100 可能经由因特网、局域网、广域网、无线网络或其它类似网络或者其组合连接到网服务器 110。

[0035] 客户设备 100 执行浏览器 115,如网浏览器,其通常被配置成接收来自网服务器 110 的超文本置标语言 (“HTML”) 页和其它数据文件,以及关联的视听文件和本领域公知的其它类型文件。例如,在一些实施例中,客户设备执行 Internet Explorer 浏览器、Netscape 浏览器、Opera 浏览器和其它类似的浏览器。

[0036] 浏览器 115 亦在通信上耦合到超语音置标语言 (“HSML”) 处理器 120,如在此和在’ 45223 申请中进一步描述的。HSML 处理器 120 包含软件模块、数据结构和其它程序代码,其通常被配置成处理和呈现包含 HSML 数据、HTML 数据和其它关联数据,如浏览器 115 中的视听数据,如在此和在’ 45223 申请中进一步描述的。

[0037] 在一些实施例中,HSML 处理器 120 包括线程模块 125 和语音模块 130。线程模块 125 包含附加软件模块、数据结构和其它程序代码,其通常被指向处置线程和检查在浏览器 115 和网服务器 110 以及连接到网络的其它设备之间传递的数据内容。线程模块 125 在通信上与语音模块 130 耦合。

[0038] 语音模块 130 包含附加软件模块、数据结构和其它程序代码,其通常被指向协调 HSML 处理器 120 和语音引擎 135 之间的通信。例如,语音模块 130 可能协调 HSML 处理器

120 和语音引擎 135, 如 DragonNaturally Speaking、IBM ViaVoice 或本领域公知的另一种语音引擎之间的通信。在一些实施例中, 语音模块 130 在通信上耦合到语音引擎 135 并将 HSML 数据和其它数据传递到语音引擎 135 以便于处理。在一些实施例中, 语音引擎 135 传递 HSML 标签数据或其它数据, 其指示语音模块 130 应当激活新页处理功能性或例行程序以处理新页。语音模块 130 然后将新页的地址从语音引擎 135 返回到线程模块 125 以传送到浏览器 115, 网服务器 110, 或连接到网络的其它计算设备。在一些实施例中, 语音模块 130 处理将得自 HSML 数据或其它数据的新语音数据加载到语音引擎 135 中的页并将新语法和其它信息的加载指引到语音引擎 135 中。

[0039] 语音模块 130 亦协调语音引擎 135 和线程模块 125 之间的通信。例如, 语音引擎 135 可能传递 HSML 数据或其它数据, 其指示新页引导被传递到语音模块 130 以便于处理。语音模块 130 然后将把新页的地址从语音引擎 135 返回到线程模块 125 以传送到浏览器 115, 网服务器 110, 或连接到网络的一些其它设备。

[0040] 在一些实施例中, HSML 处理器 120 驻留并执行在如图 1 中所示的代理服务器 140 上。在其它实施例中, HSML 处理器 120 驻留并执行在与浏览器协调的设备上, 或不然在通信上耦合到浏览器, 并且仅象如在'45223 申请中进一步描述的代理服务器那样起作用。在这两个前述实施例的任何一个中, 网服务器 110 和浏览器 115 之间的通信由此通过 HSML 处理器 120 而不是直接的。在一些实施例中, HSML 处理器 120 使用 multipart/x-mixed-replace mime 类型来维持浏览器 115 和 HSML 处理器 120 或代理服务器 140 之间的打开的连接 (open connection), 并且将数据推到浏览器, 如在此和在'45223 申请中进一步描述的。

[0041] 例如, 在一些实施例中, 所述系统以无连接方式来实施并且维持诸如按窗口地址 (per-window address) 的地址, 以跟踪与窗口关联的请求。窗口中的服务, 如小应用程序, 接受请求 (例如以更新其本身等)。例如代理服务器 140 的过程或在客户侧的其它过程确定内容应当指向的窗口并且处置与语音引擎 135 的通信。

[0042] 在其它实施例中, HSML 处理器 120 驻留在与浏览器 115 相同的设备上 (或不然在通信上与浏览器 115 耦合), 其中 HSML 处理器 120 经由 API 145 与浏览器通信, 并且 HSML 处理器 120 和浏览器 115 两者均直接与网服务器 110 通信。例如, 浏览器可能包含插件或其它类似 API, 其指向与 HSML 处理器 120 通信并且能够传递或执行命令、程序指令和其它数据于 HSML 处理器 120 和浏览器 115 之间。这样, 在一些实施例中, 浏览器 115 可能接收来自网服务器 110 的具有 HSML 数据的网页并且使用 API 145 将 HSML 数据和关联指令传递到 HSML 处理器 120 以便于如在此进一步描述的处理。

[0043] 在一些实施例中, 所述系统包括编程, 例如浏览器 115 中的小应用程序, 其无需单独的代理服务器 140 而起作用。例如, 小应用程序可与语音引擎 135、与网服务器、与浏览器、与其它小应用程序或与其它网络元素直接通信以协调页 HSML 数据和其它类型数据加载。这样, 分布式模型被呈现, 其中小应用程序针对哪个小应用程序具有焦点或活动语法、如何解决规则冲突等来彼此协商。在一些实施例中, 支持该功能性的数据结构被维持在浏览器存储器中或经由本领域公知的通信方法传递于小应用程序之间。

[0044] 图 2 呈现根据本发明一个实施例的执行对网页的话音激活的方法的流程图。所述系统检查浏览器 115 和网服务器 110 或其它服务器之间的数据传递, 步骤 150。例如, 在一些实施例中, 浏览器 115 处理 HTML 文档或包括复杂窗口的其它文档, 如具有嵌入数据的窗

口。例如,浏览器 115 可处理包含文本、图形和本领域公知的其它页资产 (page asset) 的文档。

[0045] 用于诸如图形的资产的二进制文件通常不随文档传递到浏览器 115,而相反是经由包含链接,如统一资源定位器 (“URL”) 链接的图像标签或其它类似置标标签引用到包含所述资产的二进制文件。浏览器 115 然后解析包含所述链接的图像标签,并使用该链接从网服务器 110 请求用于所述资产的二进制文件。

[0046] 可替换地,浏览器 115 可将用于数据的 HTML 请求或其它请求,例如规定新页的请求传递到网服务器 110。这样,从浏览器 115 到网服务器 110 的请求通常包含 mime 类型,其指示所述请求是否是用于当前页的资产,如图像文件、音频文件或其它类似资产,或者所述请求是否是 HTML 请求。所述系统检查所述数据,因为仅 HTML 请求和其它类型请求可属于系统的 HSML 功能性,如在’ 45223 申请中进一步描述的。

[0047] 在 HSML 处理器 120 是代理服务器 140 的组件或通常作为代理服务器 140 来起作用的实施例,所述系统使用 multipart/x-mixed-replace mime 类型来保持浏览器 115 和 HSML 处理器 120 之间的连接打开,而同时接受对嵌入在当前页内的聚集数据的请求并对其进行检查。在其它实施例中,HSML 处理器 120 与浏览器 115 的 API 145 交互以对网页进行话音使能。例如,API 145 检查数据以确定数据是否包括应当传递给 HSML 处理器 120 以便于处理的语音事件 (连同有关与该数据关联的窗口或框架的信息),或者数据是否包括待由浏览器处置的复杂数据,如在此和在’ 45223 申请中进一步描述的。HSML 处理器 120 亦处理语音事件并经由 API 145 将结果传送给浏览器 115。

[0048] 这样,所述系统检查从浏览器 115 到网服务器 110 的请求以确定所述请求是对嵌入于当前页的聚集或复杂数据的请求还是对新数据的 HTML 请求,步骤 155。在一些实施例中,所述系统检查规定 mime 类型或其它信息的与请求关联的头部、脚部或其它数据以确定请求的类型。例如,请求头部 mime 类型可以指示文件是 JPEG 文件,并且系统由此确定请求是对复杂数据的请求而不是新 HTML 请求。在一些实施例中,请求头部、脚部或与请求关联的其它数据未规定 mime 类型或对应于所请求的数据的其它信息。在这些情况下,系统通过检查与请求关联的文件名和文件扩展来确定所请求的数据内容。这样,如果用于文件 “picture.gif” 的请求头部未规定 mime 类型,则通过检查文件名 “picture.gif” 以将文件扩展 “.gif” 相关于有可能包含复杂数据而不是新 HTML 请求的已知文件扩展的列表或其它数据结构,系统将试图猜测所述请求是针对复杂数据还是新 HTML 数据。

[0049] 如果所述请求不是对新数据的 HTML 请求或其它请求,例如对新页或新窗口的请求,则数据从浏览器 (在存在 API 145 的情况下) 或类似于本领域公知的正常代理服务器而起作用的 HSML 处理器 120 直接传递直至服务器,步骤 160。在 HSML 处理器类似于代理服务器而起作用的实施例中,在请求用于给定页的所有数据时,“主” 连接,将 multipart/x-mixed-replace mime 类型规定到浏览器的连接,被维持于整个该过程。

[0050] 然而,如果请求是针对新 HTML (或文本) 数据,则数据被认为是来自浏览器 115 的新请求,并且系统确定所述请求是否由语音事件发起,步骤 162。如果事件不是由语音事件发起,例如,如果事件由用户鼠标点击或由用户将 URL 直接敲入浏览器 115 而发起,则系统将把该事件作为本领域公知的对新数据的普通 HTML 请求来处理,并且控制传递到块 175。

[0051] 在一些实施例中,系统检测与连接关联的活动,例如对不包含 HSML 数据的新页的

请求或页关闭,并且执行清除操作,如在'45223 申请中进一步描述的。例如,系统执行终止与浏览器的旧连接的清除操作并且通常亦删除与那个连接关联的先前语法。

[0052] 如果对新 HTML(或文本)数据的请求由语音数据发起,则系统确定所述请求是来自现有连接还是来自新浏览器连接,步骤 165。例如,在一个实施例中,系统接收来自一连接的请求,如规定网页的数据,该数据包括具有短语部分、命令部分和标签部分的基于规则的语法语句。在一些实施例中,HSML 处理器 120 和浏览器 115 经由 API 145 来通信以评价对新 HTML 数据的请求。在一些实施例中,所述系统使用在“JavaSpeech Grammar Format Specification”中进一步描述的数据结构、功能、过程和其它方法,其可获得于

[0053] <http://java.sun.com/products/java-mcdia/speech/forDevelopers/JSGF/JSGF.html#11803>

[0054] 并且在此以整体引入作为参考。在其它实施例中,所述系统包含连接管理器或其它软件模块或软件对象,其指向管理打开的或现有的连接或插槽(socket)与其关联语法之间的关联。例如,用户可能使几个浏览器 115 窗口打开或具有多个框架的一个浏览器 115 窗口打开。这些窗口或框架的每个与一打开的连接(经由 multipart/x-mixed-replace mime 类型服务器推方法或其它类型方法)和关联语音语法相关联,如在'45223 申请中进一步描述的。这样,连接管理器尤其检查对新 HTML 数据的请求以确定所述请求是来自现有连接还是新连接。

[0055] 在一些实施例中,连接管理器通常包括这样的软件对象,其包括两组列表或其它数据结构,用于从列表中添加或删除项目,并用于将本身和对应列表重新计算为散列(hash)。所述两个列表包括:第一列表,其跟踪到窗口或框架的打开的或现有的连接或插槽和与窗口或框架关联的语法之间的关联;以及第二列表,其跟踪语音规则和与语音规则关联的语法之间的关联。在一些实施例中,连接可包括线程,例如其中处理发生在客户处的多线程环境。根据本发明一个实施例的用于这些列表的示例性数据结构作为附录 A 附加于此。在一些实施例中,连接管理器包括与线程模块 125 或 API 145 关联的软件模块或软件对象。在其它实施例中,连接管理器包括独立的软件模块或软件对象。

[0056] 如果请求来自打开的或现有的连接,则连接管理器检查包含在该请求中或与之关联的语音规则,并且针对其列表来执行查询以确定与该请求关联的语法和连接,步骤 170。在一些实施例中,新连接被建立于浏览器 115 和网服务器 110 或代理服务器 140 之间,步骤 175。在一些实施例中,与代理服务器 140 的新连接被添加给连接管理器所维持的连接列表或其它类似数据结构。在一些实施例中,所述系统亦跟踪代理服务器 140 和网服务器 110 或浏览器 115 之间的连接,以解决线程同时请求或用于其它目的。

[0057] 所述系统亦评价属于新连接的输入数据以确定是否存在与该新连接关联的新语法。如果新语法与该连接关联,则与其关联的该语法和任何语音规则被加载并且亦添加给适当的连接管理器列表,如在此先前所描述的。如在'45223 申请中进一步描述的,所述系统通常执行清除操作并终止浏览器和 HSML 处理器 120 之间的旧连接,并且通常亦删除与那个连接关联的先前语法,步骤 180。请求或其它动作被处理于对应于新连接的窗口或框架中或不然被指向它,步骤 180。这样,数据被馈送回到与语音请求或其它请求关联的窗口。

[0058] 在一些实施例中,如果在步骤 165 中所述的对新 HTML 数据(或文本)的请求未与打开的或现有的连接或插槽关联,则连接管理器将新连接添加到其连接列表,步骤 190。连

接管理器亦将与那个连接关联的任何语法和语音规则添加到其列表,如先前在此所述,步骤 195。系统在对应于所述连接的窗口上处理所述动作,步骤 200,控制返回到块 150,并且系统继续处理数据,如在此和在' 45223 申请中进一步描述的。

[0059] 在一些实施例中,多个打开的窗口或框架的每个可具有活动的关联语法。在一些实施例中,系统必须决定如何处理包含在多个活动语法中的语音事件的冲突,这是通过确定哪个打开的窗口或框架对应于该语音事件来进行的。例如,两个打开的窗口或框架可能包含与语音事件“返回到主页”关联的代码。在处理指示“返回到主页”的语音事件或指令时,系统必须确定哪个打开的窗口或框架应当被返回到其主页。

[0060] 在一些实施例中,当检测到这样的冲突时,系统显示请求用户输入或其它输入的对话或其它通知以选择对应于所述语音事件或指令或其它事件的适当语法或窗口。

[0061] 在其它实施例中,系统检测一个语法来用作处理语音事件和其它指令或事件所针对的焦点语法,并且停用其它语法。这样,没有了处理包含在多个活动语法中的语音事件和其它指令或事件的混乱。与打开的窗口或框架关联的所有语法通常仍继续存在于存储器中,但在任何时间只有一个语法具有焦点。

[0062] 例如,在一些实施例中,系统选择与浏览器 115 所加载的最后窗口或框架关联的语法。在其它实施例中,系统选择与活动浏览器 115 窗口关联的语法。例如,在 Microsoft Windows 操作系统或具有图形用户接口的其它操作系统中,多个窗口可同时打开,但这些窗口通常被呈现为层,或不然与作为活动窗口的顶部窗口级联。在其它实施例中,系统可能需要的是,框架不包含重复的语音事件或指令或其它事件。

[0063] 在此所描述的系统 and 模块可包括适合于在此所述的目的的软件、固件、硬件或软件、固件或硬件的任何(多个)组合。软件和其它模块可驻留在服务器、工作站、个人计算机、计算机化输入板、PDA 和适合于在此所述的目的的其它设备。软件和其它模块可经由本地存储器、经由网络、经由浏览器或 ASP 情况下的其它应用、经由适合于在此所述的目的的其它方式来接入。在此所述的数据结构可包括适合于在此所述的目的的计算机文件、变量、编程阵列、编程结构、或任何电子信息存储方案或方法或其任何组合。在此所述的用户接口元素可包括来自图形用户接口、命令行接口和适合于在此所述的目的的其它接口的元素。在此所呈现和描述的屏幕截图可如本领域所公知的不同地显示以输入、访问、改变、操纵、修改、更改和作用于信息。

[0064] 尽管已结合优选实施例描述和图示了本发明,可在本发明的精神和范围内进行如对本领域的技术人员将显而易见的许多变化和修改,并且本发明由此并不局限于以上所提出的方法学或构造的确切细节,因为这样的变化和修改旨在被包括在本发明的范围内。

[0065] 附录 A

[0066] 连接数据结构

[0067]

>From SMLApp.java: The main module that handles thread management

```

    /** ConnectionListManager.
    /** Collections provide binary search. This provides a hashtable lookup which believe will be
    faster.
    /** Add/rem are a bit slower, but only one add/rem per set of voice pages. One lookup per page
    /** Hashtables also makes explicit what our searches will be on (what our keys are).
    static class ConnectionListManager {
    the impl is explicitly provided in the declr.
        static class Connection {
            Socket sockS;
            OutputStream osO;
            InputStream isI; /** Necessary ?
            String sGrammar;
            boolean bPrevMsgGram = false;
        }

        static Hashtable hGrammar = new Hashtable(); /** Grammar is the key
        static Hashtable hSock = new Hashtable(); /** Socket is the key.
        static LinkedList llConnections = new LinkedList();
        synchronized static void Add(Connection C) throws ConnectionException {
            Dbg.println("SMLApp", Dbg.CONN, "Add", "Connection: " + C + ". C.sGrammar: " +
            C.sGrammar + ". C.sockS: " + C.sockS);
            llConnections.add(C);
            if ( (C.sGrammar == null) || (C.sockS == null) ) throw new ConnectionException("Null
            Grammar or Socket");
            hGrammar.put(C.sGrammar, new Integer(llConnections.indexOf(C)));
            hSock.put(C.sockS, new Integer(llConnections.indexOf(C)));
        }

        synchronized static void Update(Connection C, String sGrammar) throws ConnectionException
        {
            Dbg.println("SMLApp", Dbg.CONN, "Update", "Connection: " + C + ". C.sGrammar: "
            + C.sGrammar + ". sGrammar: " + sGrammar + ". C.sockS: " + C.sockS);
            if ( (sGrammar == null) || (C.sGrammar == null) ) throw new
            ConnectionException("Null Grammar");
            if ( llConnections.indexOf(C) < 0 ) throw new ConnectionException("Invalid
            Connection");
            hGrammar.remove(C.sGrammar);
            C.sGrammar = sGrammar;
            hGrammar.put(C.sGrammar, new Integer(llConnections.indexOf(C)));
        }

        synchronized static void Rem(Socket sockS) {
            Connection C = Lookup(sockS);
            Rem(C);
        }

```

[0068]

```

synchronized static void Rem(Connection C) {
    Dbg.println("SMLApp", Dbg.CONN, "Rem", "Connection: " + C + ". C.sGrammar: " +
C.sGrammar + ". C.sockS: " + C.sockS);
    llConnections.remove(C);
    RecomputeHashTables();
}

static void RecomputeHashTables() {
    hGrammar.clear();
    hSock.clear();
    Connection C = (Connection) llConnections.getFirst();
    if (C == null) return;
    int iIndex = 0;
    hGrammar.put(C.sGrammar, new Integer(iIndex));
    hSock.put(C.sockS, new Integer(iIndex)); iIndex++;
    while (iIndex < llConnections.size()){
        C = (Connection) llConnections.get(iIndex);
        hGrammar.put(C.sGrammar, new Integer(iIndex));
        hSock.put(C.sockS, new Integer(iIndex));
        iIndex++;
    }
}

synchronized static Connection Lookup(String sGrammar) {
    int iIndex = -1;
    try {
        Dbg.println("SMLApp", Dbg.CONN, "Lookup", "sGrammar: " + sGrammar);
        Object oTmp = hGrammar.get(sGrammar);
        if (oTmp == null) return null;

        iIndex = ((Integer) oTmp).intValue();
        return (Connection) llConnections.get(iIndex);
    }
    catch(IndexOutOfBoundsException e) {
        Dbg.println("SMLApp", Dbg.ERR, "Lookup(sGrammar)",
        "IndexOutOfBoundsException: iIndex" + iIndex + ".
sGrammar: " + sGrammar);
        return null;
    }
}

synchronized static Connection Lookup(Socket sockS) {
    int iIndex = -1;
    try {
        Dbg.println("SMLApp", Dbg.CONN, "Lookup", "sockS" + sockS);
        Object oTmp = hSock.get(sockS);
        if (oTmp == null) return null;
        iIndex = ((Integer) oTmp).intValue();
        return (Connection) llConnections.get(iIndex);
    }
    catch(IndexOutOfBoundsException e) {
        Dbg.println("SMLApp", Dbg.ERR, "ConnectionListManager.Lookup(sockS)",
        "IndexOutOfBoundsException: iIndex" + iIndex + ". sockS: "
+ sockS);
        return null;
    }
}

```

[0069]

```

    }
}

public static class ConnectionException extends Exception {
    public ConnectionException(String msg) {
        super(msg);
        Dbg.println("SMLApp", Dbg.ERR, "ConnectionException", msg);
    }
}

static ConnectionListManager CLM;
static ConnectionListManager.Connection connGDisplay = null;           /* Connection
based on gDisplay grammar lookup.

```

[0070] 语法数据结构

[0071]

>From SpeechIO.java:

```

public class Rule {
    String sTag = null;
    String sJSGF = null ;
    String sJSMML = null ;
    String sURL = null;      /* URL to call when this rule triggered
    String sURLSep = null ;
    String sURLExt = null ;
    int iSpeakJSGF = 0; /* Create speakable phrase from (depth - (iSpeakJSGF-1)) level JSGF.
}

public class EGBlock { /* Embedded Grammar JSGF rules
int iNumRules = 0;      /* Number of rules in this block
Rule arRule[] = new Rule[iMaxNumRule]; /* Linked list later...
String sJSMML = null ;
String sGrammar;
String sEntryPhrase = null;
boolean bDefaultSpeak = true;      /* Speak the JSML in this block w/o prompting.
/* false if jsgf present in this block.

public class EGBlockSet {
    int iIncReal;
    String sGrammar;
    EGBlock[] aEGBlock;
}

static Hashtable hGrammar = new Hashtable();      /* Grammar is the key
static LinkedList lIEGBlockSets = new LinkedList();

synchronized static void Add(EGBlockSet EGBS) throws SpeechIOException {
    Dbg.println("SpeechIO", Dbg.GRAMMAR, "Add", "EGBlockSet: " + EGBS + ".
    EGBS.sGrammar: " + EGBS.sGrammar);
    lIEGBlockSets.add(EGBS);
    if (EGBS.sGrammar == null) throw new SpeechIOException("Null Grammar");
    hGrammar.put(EGBS.sGrammar, new Integer(lIEGBlockSets.indexOf(EGBS)));
}

```

[0072]

```

synchronized static void Rem(String sGrammar) {
    EGBlockSet EGBS = Lookup(sGrammar);
    Rem(EGBS);
}

synchronized static void Rem(EGBlockSet EGBS) {
    Dbg.println("SpeechIO", Dbg.GRAMMAR, "Rem", "EGBlockSet: " + EGBS + ".
EGBS.sGrammar: " + EGBS.sGrammar);
    IIEGBlockSets.remove(EGBS);
    RecomputeHashTables();
}

static void RecomputeHashTables() {
    try {
        hGrammar.clear();
        EGBlockSet EGBS = (EGBlockSet) IIEGBlockSets.getFirst();
        if (EGBS == null) return;
        int iIndex = 0;
        hGrammar.put(EGBS.sGrammar, new Integer(iIndex));
        iIndex++;
        while (iIndex < IIEGBlockSets.size()){
            EGBS = (EGBlockSet) IIEGBlockSets.get(iIndex);
            hGrammar.put(EGBS.sGrammar, new Integer(iIndex));
            iIndex++;
        }
    } catch (java.util.NoSuchElementException E) {}
}

synchronized static EGBlockSet[] toArray() {
    EGBlockSet[] aEGBS = new EGBlockSet[1];
    aEGBS = (EGBlockSet[]) IIEGBlockSets.toArray(aEGBS);
    return aEGBS;
}

synchronized static EGBlockSet Lookup(String sGrammar) {
    int iIndex = -1;
    try {
        Dbg.println("SpeechIO", Dbg.GRAMMAR, "Lookup", "sGrammar: " + sGrammar);
        Object oTmp = hGrammar.get(sGrammar);
        if (oTmp == null) return null;

        iIndex = ((Integer) oTmp).intValue();
        return (EGBlockSet) IIEGBlockSets.get(iIndex);
    }
    catch (IndexOutOfBoundsException e) {
        Dbg.println("SpeechIO", Dbg.ERR, "Lookup(sGrammar)",
            "IndexOutOfBoundsException: iIndex" + iIndex + ". sGrammar: " +
sGrammar);
        return null;
    }
}

public static class SpeechIOException extends Exception {

```

[0073]

```
public SpeechIOException(String msg) {  
    super(msg);  
    Dbg.println("SpeechIO", Dbg.ERR, "SpeechIOException", msg);  
}}
```

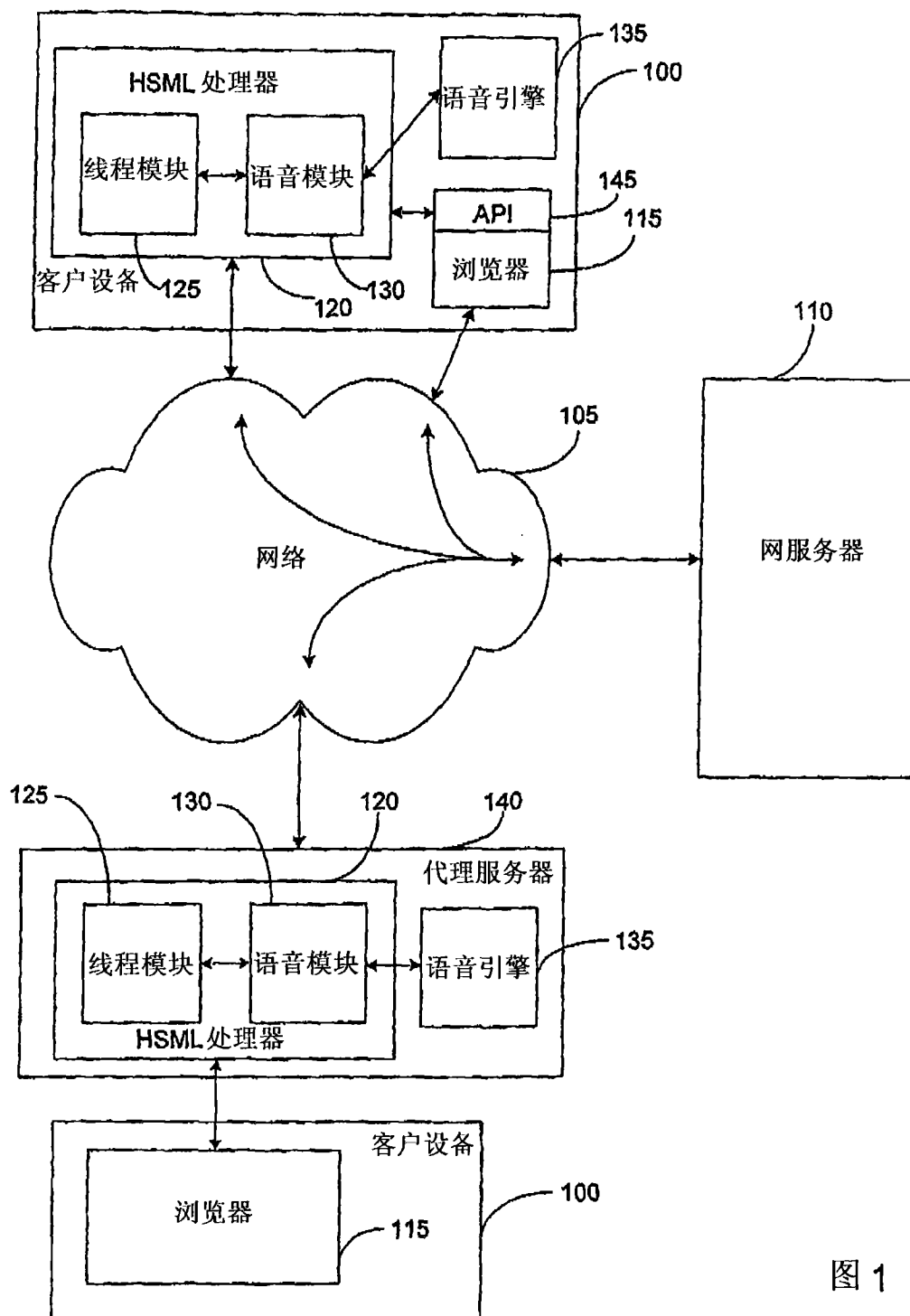


图 1

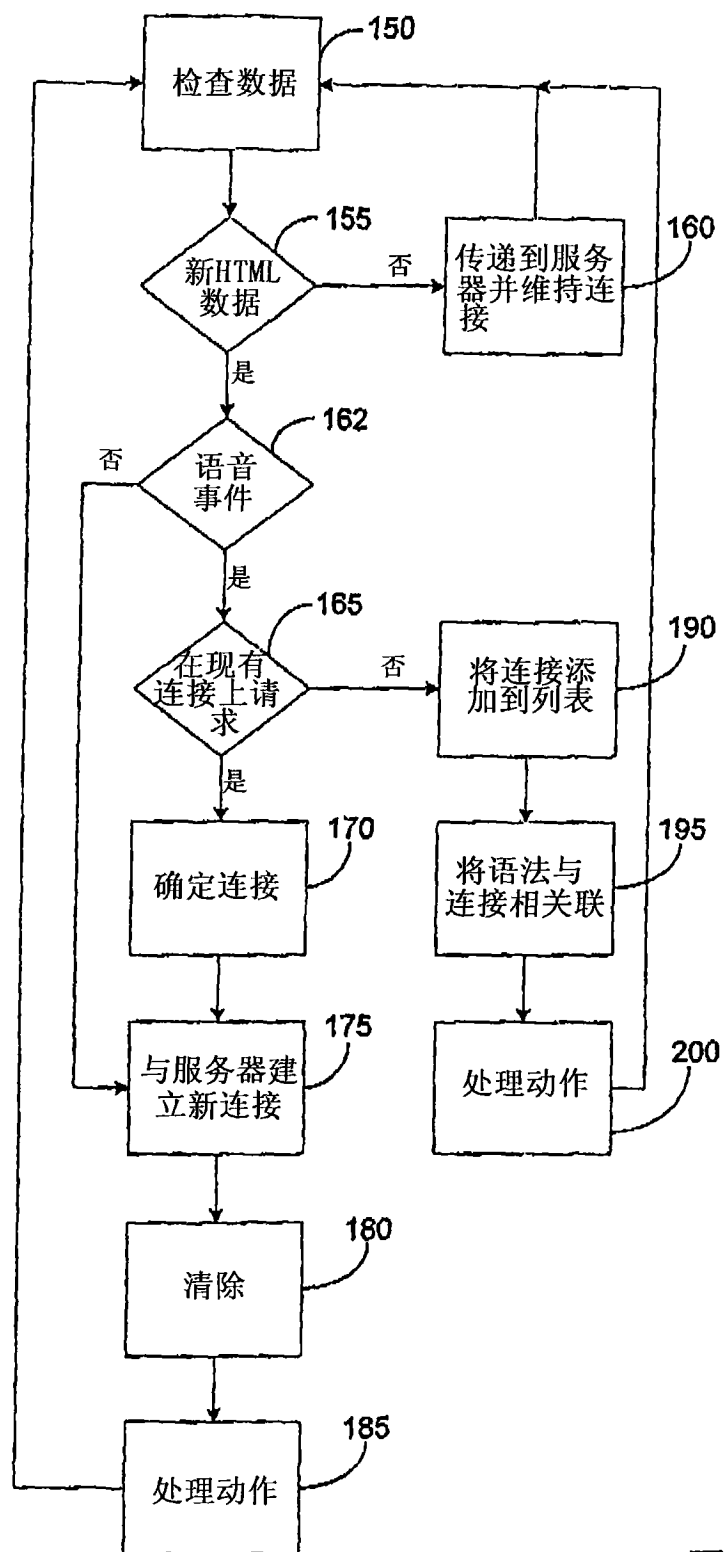


图2