

FIG. 4

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 2008172488 A [0004]
- US 20050060410 A1 [0004]

(19)



(11)

EP 2 638 683 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
15.03.2017 Bulletin 2017/11

(51) Int Cl.:
H04L 29/06 (2006.01) **G06F 15/16** (2006.01)
H04L 29/08 (2006.01)

(21) Application number: **11840003.5**

(86) International application number:
PCT/US2011/059994

(22) Date of filing: **09.11.2011**

(87) International publication number:
WO 2012/064857 (18.05.2012 Gazette 2012/20)

(54) **METHODS FOR REDUCING LATENCY IN NETWORK CONNECTIONS USING AUTOMATIC REDIRECTS AND SYSTEMS THEREOF**

VERFAHREN ZUR LATENZVERRINGERUNG IN NETZWERKVERBINDUNGEN MIT AUTOMATISCHEN UMLEITUNGEN UND SYSTEME DAFÜR

PROCÉDÉS DE RÉDUCTION DE LATENCE DANS DES CONNEXIONS DE RÉSEAU À L'AIDE DE REDIRECTIONS AUTOMATIQUES, ET SYSTÈMES CORRESPONDANTS

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(72) Inventor: **SCODA, Enrico Martignacco(UD) (IT)**

(30) Priority: **09.11.2010 US 927169**
13.07.2011 US 201113135707

(74) Representative: **Cinquantini, Bruno et al Notarbartolo & Gervasi S.p.A. Corso di Porta Vittoria, 9 20122 Milano (IT)**

(43) Date of publication of application:
18.09.2013 Bulletin 2013/38

(56) References cited:
US-A1- 2004 215 717 US-A1- 2005 060 410
US-A1- 2008 172 488 US-A1- 2008 195 696
US-A1- 2010 198 911 US-A1- 2010 223 355

(73) Proprietor: **Usablenet Inc. New York, NY 10019 (US)**

EP 2 638 683 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description**FIELD**

[0001] This technology generally relates to web content optimization apparatuses and, more particularly, to methods and systems for reducing latency in network connections using automatic redirects in web content optimization apparatuses.

BACKGROUND

[0002] When a client device connects to a server (e.g., a web server, or a content server) to get a network resource using a network protocol, e.g., the Hyper-text Transfer protocol (HTTP), the server responds by sending the network resource or by sending a redirect message back to the client device over a communication channel. If the client device receives a redirect message, it will need to send a new request to the server based upon the redirect message, and the server will again respond with a redirect or a real resource. This communication process between the client device and the server repeats until the client device is able to get the resource, if available.

[0003] However, when the client device, e.g., a cell phone using a radio network, or a computer having a slow Internet connection, requests a network resource and has to perform more than one redirects to obtain the network resource, the client device will experience substantial delay and will spend a considerable amount of time to execute the whole process before finally being provided with the network resource. The delay can occur, for example, because of a large time to establish a connection and send the HTTP request, also referred to as latency time of radio networks or other slow network connections (e.g., Internet via a dial-up connection). Unfortunately, this delay can often lead to the client device not being able to obtain the network resource at all, or the client device giving up or relinquishing attempts to obtain the network resource under time constraints.

[0004] One conventional solution built to obtain faster HTTP responses for slow connection networks utilizes one or more proxy server devices (e.g., web proxy servers). Another conventional solution utilizes telephone carrier data centers that handle the network traffic for each client device by handling one or more requests when the client device is a mobile telephone, or a mobile personal digital assistant (PDA) device, for example. Unfortunately, the above-noted conventional solutions do not resolve the redirection problem associated with network resources stored on the servers since the redirect messages are forwarded on to the client devices for handling resulting in multiple back and forth communication between the client devices and the servers. US patent application n. 2008/0195696 discloses a redirection request interception method includes: a step of receiving an HTTP response sent to a user station; a step of ob-

taining a redirection address contained in that response; a step of creating and a step of sending a substitution request to that redirection address; a step of receiving the response to that substitution request; and a step of transmitting that response to the user station. US patent application n. 2004/0215717 discloses a method for prefetching of structured data between a client device and a server device adapted to achieve more efficient data transmission, in particular transmission of structured data, such as web pages, in a system comprising a server means and a client means which are interconnected via a plurality of proxy means, including a server proxy means and a client proxy means.

SUMMARY

[0005] A method for reducing latency in network connections includes receiving at an optimization apparatus a request from one of one or more client devices for network content hosted at one of one or more server devices. The optimization apparatus determines whether an automatic redirect parameter is enabled for the request. The request from one proxy processing device acting as the requesting one of the client devices within the optimization apparatus is sent to another proxy processing device within the optimization apparatus when the automatic redirect parameter is determined to be enabled. One or more redirect messages with one or more optimized cookies associated with the requested network resource are obtained from the one or more server devices at the other proxy processing device within the web content optimization apparatus. The last of the one or more redirects messages with the one or more optimized cookies when the requested network resource is located is sent from the other proxy processing device within the optimization apparatus to the one proxy processing device within the optimization apparatus. The last of the one or more redirect messages with the one or more optimized cookies associated with the requested network resource is sent from the optimization apparatus to the requesting one of the one or more of client devices.

[0006] A non-transitory computer readable medium having stored thereon instructions for reducing latency in network connections comprises machine executable code which when executed by at least one processor, causes the processor to perform steps including receiving a request from one of one or more client devices for network content hosted at one of one or more server devices. A determination is made whether an automatic redirect parameter is enabled for the request. The request from one proxy processing device acting as the requesting one of the client devices within an optimization apparatus is sent to another proxy processing device within the optimization apparatus when the automatic redirect parameter is determined to be enabled. One or more redirect messages with one or more optimized cookies associated with the requested network resource is obtained from the one or more server devices at the

other proxy processing device within the optimization apparatus. The last of the one or more redirects messages with the one or more optimized cookies when the requested network resource is located is sent from the other proxy processing device within the optimization apparatus to the one proxy processing device within the optimization apparatus. The last of the one or more redirect messages with the one or more optimized cookies associated with the requested network resource is sent to the requesting one of the one or more of client devices.

[0007] An apparatus configured to reduce latency in network connections comprising one or more processors and a memory coupled to the one or more processors configured to execute programmed instructions stored in the memory including receiving a request from one of one or more client devices for network content hosted at one of one or more server devices. A determination is made whether an automatic redirect parameter is enabled for the request. The request from one proxy processing device acting as the requesting one of the client devices within an optimization apparatus is sent to another proxy processing device within the optimization apparatus when the automatic redirect parameter is determined to be enabled. One or more redirect messages with one or more optimized cookies associated with the requested network resource is obtained from the one or more server devices at the other proxy processing device within the optimization apparatus. The last of the one or more redirects messages with the one or more optimized cookies when the requested network resource is located is sent from the other proxy processing device within the optimization apparatus to the one proxy processing device within the optimization apparatus. The last of the one or more redirect messages with the one or more optimized cookies associated with the requested network resource is sent to the requesting one of the one or more of client devices.

[0008] This technology provides a number of advantages including providing a method, computer readable medium and apparatus that efficiently manages the whole redirect chain on behalf of the client device by optimizing at least a portion of redirect messages automatically within a web content optimization apparatus. Since at least a portion of the redirect chain associated with locating and obtaining the network resource is handled internally by separate processing devices within the web content optimization apparatus, network communications between the client devices and the servers for network resources are sped up. This technology exploits the faster internal processing of redirect messages in one or more redirect chains by the web content optimization apparatus, as compared to slower communication speeds between client device and the servers.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]

FIG. 1 is a block diagram of an exemplary network environment with a web content optimization apparatus interposed between client devices and server devices;

FIG. 2 is a flow chart of a method for processing an exemplary sequence of HTTP requests and responses between one of the client devices and the server devices hosting example.com and sample.com;

FIG. 3 is a flow chart of a method for processing an exemplary sequence of HTTP requests and responses between a client device and the web content optimization apparatus; and

FIG. 4 is a flow chart of a method for processing an exemplary sequence of HTTP requests and responses between a client device and the web content optimization apparatus when auto redirect is enabled; and

FIG. 5 is a flowchart of an exemplary method for reducing latency in network connections by optimizing network resource using automatic redirection of redirect response messages at the web content optimization apparatus.

DETAILED DESCRIPTION

[0010] An exemplary environment 10 in which a web content optimization apparatus 12 is optimized for reducing latency in network connections is illustrated in FIG. 1. The exemplary environment 10 includes the web content optimization apparatus 12, client devices 14(1)-14(n), server devices 16(1)-16(n), and communication networks 18(1)-18(2), although other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can be used. This technology provides a number of advantages including providing a method, computer readable medium, and an apparatus that reduces latency in network connections, for example, HTTP connections.

[0011] Referring more specifically to FIG. 1, the web content optimization apparatus 12 manages handling of redirect messages or redirection responses from the server devices 16(1)-16(n) for and/or on behalf of requesting client devices 14(1)-14(n) and provides updated cookie information to the client devices 14(1)-14(n) for future requests for network resources from the client devices 14(1)-14(n), although the web content optimization apparatus 12 can provide other numbers and types of functions and other types of processing devices can be used. Although one web content optimization apparatus 12 is shown, other numbers and types of optimization devices and systems can be used.

[0012] The web content optimization apparatus 12 in-

cludes central processing units (CPU) or processors 13(1) and 13(2), a memory 15, and an interface system 17 which are coupled together by a bus 19 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processors 13(1) and 13(2) in the web content optimization apparatus 12 executes a program of stored instructions to carry out or perform one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although other numbers of processors can be used and one or more of the processors could execute other numbers and types of programmed instructions.

[0013] The memory 15 in the web content optimization apparatus 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processors 13(1) and 13(2), can be used for the memory 15 in the web content optimization apparatus 12. In these embodiments, the memory 15 includes a core module 21 and a redirect module 23 which store programmed instructions for one or more aspects of the present invention as described and illustrated herein, although the memory can comprise other types and numbers of systems, devices, and elements in other configurations which store other data.

[0014] The interface system 17 in the web content optimization apparatus 12 is used to operatively couple and communicate between the web content optimization apparatus 12 and the client devices 14(1)-14(n) and the server devices 16(1)-16(n) via the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. By way of example only, the communication networks 18(1) and 18(2) can use TCP/IP over Ethernet and industry-standard protocols, including HTTP, HTTPS, WAP, and SOAP, although other types and numbers of communication networks, such as a direct connection, a local area network, a wide area network, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used. In one exemplary embodiment, one of communication networks 18(1) and 18(2) can be operating over one or more low-speed connections (e.g., a dial-up connection) while the other one of the communication networks 18(1) and 18(2) can be operating over a high speed, high bandwidth connection (e.g., optical fiber based communication network). In yet another exemplary embodiment, one or more of communication networks 18(1) and 18(2) can be a radio network, a satellite

network, an Internet connection, a wired cable network, or combinations thereof, well known to one of ordinary skill in the art reading this disclosure.

[0015] Each of the client devices 14(1)-14(n) enables a user to request, obtain, and interact with one or more network resources, e.g., web pages from one or more web sites, hosted by server devices 16(1)-16(n) through the web content optimization apparatus 12 via one or more communication networks (e.g., communication network 18(1)), although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. Although multiple client devices 14(1)-14(n) are shown, other numbers and types of user computing systems could be used. In one example, the client devices 14(1)-14(n) comprise mobile devices with Internet access that permit a website form page or other retrieved data that is a part of a requested network resource to be displayed, although each of the client devices 14(1)-14(n) can connect to server devices 16(1)-16(n) via other types of network connections directly or indirectly, depending upon specific scenarios, as can be contemplated by one of ordinary skill in the art, after reading this disclosure. By way of example only, one or more of the client devices 14(1)-14(n) can comprise smart phones, personal digital assistants, computers, or other computing devices.

[0016] Each of client devices 14(1)-14(n) in this example is a computing device that includes a central processing unit (CPU) or processor 20, a memory 22, user input device 24, a display 26, and an interface system 28, and which are coupled together by a bus 30 or other link, although one or more of client devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 20 in each of client devices 14(1)-14(n) executes a program of stored instructions for aiding one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0017] The memory 22 in each of the client devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein as well as other data, such as updated cookies associated with a network resource and received as part of one or more redirect chains forwarded by web content optimization apparatus 12, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 20 can be used for the memory 22 in each of the client devices 14(1)-14(n).

[0018] The user input device 24 in each of the client devices 14(1)-14(n) is used to input selections, such as requests for a network resource, e.g., a particular website form page or to enter data in fields of a form page, although the user input device could be used to input other types of data and interact with other elements of exemplary environment 10. The user input device 24 can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of user input devices can be used.

[0019] The display 26 in each of the client devices 14(1)-14(n) is used to show data and information to the user, such as requested website page by way of example only. The display in each of the client devices 14(1)-14(n) is a mobile phone screen display, although other types and numbers of displays could be used depending on the particular type of client device, as can be contemplated by one of ordinary skill in the art, after reading this disclosure.

[0020] The interface system 28 in each of the client devices 14(1)-14(n) is used to operatively couple and communicate between the client devices 14(1)-14(n) and the web content optimization apparatus 12 and server devices 16(1)-16(n) over the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0021] The server devices 16(1)-16(n) provide one or more pages from one or more web sites for use by one or more of the client devices 14(1)-14(n) via the web content optimization apparatus 12, although the server devices 16(1)-16(n) can provide other numbers and types of applications and/or content and can have provide other numbers and types of functions. Although server devices 16(1)-16(n) are shown for ease of illustration and discussion, other numbers and types of server systems, for example, web servers, and devices can be used. In one example, server devices 16(1)-16(n) can be web servers having dedicated hardware with software executing on the dedicated hardware to facilitate the web content optimization apparatus 12 and client devices 14(1)-14(n) in their functioning. In another example, server devices 16(1)-16(n) can be content servers that are configured to deliver network resources stored thereupon using the HTTP protocol, or other network protocols for example. Content stored on server devices 16(1)-16(n) that can be part of the network resources requested by client devices 14(1)-14(n) can be web pages, electronic files and documents, configuration data, metadata, or other network data and files, by way of example only and not by way of limitation.

[0022] Each of the server devices 16(1)-16(n) include a central processing unit (CPU) or processor, a memory, and an interface system which are coupled together by a bus or other link, although each of the server devices 16(1)-16(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor

in each of the server devices 16(1)-16(n) executes a program of stored instructions one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions. When one of the server devices 16(1)-16(n) does not store the requested content, the server device may respond by sending a redirect message to the web content optimization apparatus 12.

[0023] The memory in each of the server devices 16(1)-16(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated by way of the embodiments, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor, can be used for the memory in each of the server devices 16(1)-16(n).

[0024] The interface system in each of the server devices 16(1)-16(n) is used to operatively couple and communicate between the server devices 16(1)-16(n) and the web content optimization apparatus 12 and the client devices 14(1)-14(n) via communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0025] Although embodiments of the web content optimization apparatus 12, the client devices 14(1)-14(n), and the server devices 16(1)-16(n), are described and illustrated herein, each of the client devices 14(1)-14(n), the web content optimization apparatus 12, and the server devices 16(1)-16(n), can be implemented on any suitable computer system or computing device. It is to be understood that the devices and systems of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0026] Furthermore, each of the systems of the embodiments may be conveniently implemented using one or more general purpose computer systems having non-transitory computer readable medium, microprocessors, digital signal processors, and microcontrollers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0027] In addition, two or more computing systems or devices can be substituted for any one of the systems in any embodiment of the embodiments. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the embodiments. The em-

bodiments may also be implemented on computer system or systems that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0028] The embodiments may also be embodied as non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0029] Referring to FIG. 2, an exemplary method 200 for processing an exemplary sequence of HTTP requests and responses between one of the client devices 14(1)-14(n) and the server devices 16(1)-16(n) hosting example.com and sample.com without a the web content optimization apparatus is illustrated. It is to be noted that the sequence of steps for the method 200 is only exemplary and one of ordinary skill in the art, after reading this disclosure, can contemplate alternative sequences of steps that achieve substantially the same result.

[0030] More specifically, at step 202 one of the client devices 14(1)-14(n) (e.g., a mobile device) sends an HTTP request: GET/A.html HTTP/1.1 for a network resource at a Uniform Resource Locator (URL) <http://www.example.com/A.html>, although other types of requests for other types of network resources may be sent. Although in this example one of the client devices 14(1)-14(n) via a web browser requests a page A.html at the URL, "www.example.com" as shown, by way of example only, client devices 14(1)-14(n) may send a request for a network shared data file using a file transfer protocol instead of a URL using the HTTP protocol.

[0031] In step 204, the requesting one of the client devices 14(1)-14(n) gets a response from the one of the server devices 16(1)-16(n) that was previously hosting the network resource, in this example the requested page A.html. The response includes a temporary redirect message (shown as an exemplary status code 302 with a message "Moved Temporarily") to a different URL <http://sample.com/B.html> along with a "Set-Cookie" including a cookie named "SESSION" set to value "1234".

[0032] Generally, the cookie is a string formed by the pair "name=value" (e.g., "SESSION=1234", followed by optional attributes. Although one illustrative example is described herein, this technology can be used with specifications for all cookies.

[0033] In step 206, the requesting one of the client devices 14(1)-14(n) processes the response and sends a new HTTP request to the one of the server devices 16(1)-

16(n) hosting the URL <http://www.sample.com/B.html>.

[0034] In step 208, the requesting one of the client devices 14(1)-14(n) gets a response from the one of the server devices 16(1)-16(n) that was previously hosting the network resource, in this example the requested page B.html. The response includes another temporary redirect message, "Moved Temporarily" to a different URL <http://example.com/C.html> along with a "Set-Cookie" including a cookie named "LANG" set to value "EN".

[0035] In step 210, the requesting one of the client devices 14(1)-14(n) processes the response and sends another new HTTP request to the one of the server devices 16(1)-16(n) hosting the URL <http://www.example.com/C.html>.

[0036] In step 212, the requesting one of the client devices 14(1)-14(n) gets a response from the one of the server devices 16(1)-16(n) hosting the network resource with a status identifier message shown as status code 200 set to "OK" and the network resource can now be retrieved. Although exemplary steps 202-212 are shown, a higher or a lower number of redirections may be required by a particular request.

[0037] Referring to FIG. 3, a flow chart of a method for processing an exemplary sequence of HTTP requests and responses between one of the client devices 14(1)-14(n) and the web content optimization apparatus 12 is illustrated. In step 302, one of the client devices 14(1)-14(n) makes a request for <http://m.acme.com/m/example.com/A.html> which is received by the web content optimization apparatus 12.

[0038] In step 304, the web content optimization apparatus 12 generates a request for A.html at example.com which is transmitted to the one of the server devices 16(1)-16(n) thought to host the network resource.

[0039] In step 306, the web content optimization apparatus 12 receives a redirect response from the one of the server devices 16(1)-16(n) to <http://sample.com/B.html> and setting the cookie SESSION=1234 which is provided to the requesting one of the client devices 14(1)-14(n).

[0040] In step 308, this redirect response is optimized by the web content optimization apparatus 12 into a redirect to <http://m.acme.com/m/sample.com/B.html> with optimized version cookie SESSION+example.com+%2F=1234 in this example and transmitted to the requesting one of the client devices 14(1)-14(n).

[0041] In step 310, the requesting one of the client devices 14(1)-14(n) makes the next request for <http://m.acme.com/m/sample.com/B.html> based on the received optimized redirect response. This request is received by the web content optimization apparatus 12.

[0042] In step 312 the web content optimization apparatus 12 generates a request for document B.html on sample.com which is transmitted to the one of the server devices 16(1)-16(n) thought to currently host the network resource.

[0043] In step 314, the web content optimization apparatus 12 receives a redirect response from the one of

the server devices 16(1)-16(n) to `http://example.com/C.html` setting the cookie `LANG=EN`.

[0044] In step 316, this redirect response is optimized by the web content optimization apparatus 12 into a redirect to `http://m.acme.com/m/example.com/C.html` with optimized version cookie `LANG+sample.com+%2F=EN` which is transmitted to the requesting one of the client devices 14(1)-14(n).

[0045] In step 318, the one of the client devices 14(1)-14(n) makes the next request for `http://m.acme.com/m/example.com/C.html` which is received by the web content optimization apparatus 12.

[0046] In step 320 the web content optimization apparatus 12 generates a request for document `C.html` on `example.com` passing the matching cookie `SESSION=1234` to the one of the server devices 16(1)-16(n) thought to currently host the network resource.

[0047] In step 322, gets a response from the one of the server devices 16(1)-16(n) with a status identifier message shown as status code 200 set to "OK" indicating the network resource has been found. In step 324, the web content optimization apparatus 12 provides this response to the requesting one of the client devices 14(1)-14(n) to retrieve the located network resource.

[0048] Referring to FIG. 4, an exemplary method 400 for processing an exemplary sequence of HTTP requests and responses between one of the client devices 14(1)-14(n) and the web content optimization apparatus 12 when auto redirect is enabled. In step 402, one of the client devices 14(1)-14(n), which in this example is a mobile device, makes a new request for the URL: `m.acme.com/m/example.com/A.html` which is received by the web content optimization apparatus 12.

[0049] In step 404, the web content optimization apparatus 12 processes the request utilizing processors 13(1) and 13(2) which act as both the requesting client device and the proxy server device as explained in greater detail below.

More specifically, in step 404a one of the processors 13(1) in the web content optimization apparatus 12 acting as the client device sends a request for `/m/example.com/A.html` to another one of the processors 13(2) in the web content optimization apparatus 12. The auto redirect between the processors 13(1) and 13(2) in the web content optimization apparatus 12 is turned off.

[0050] In step 404b, the other one of the processors 13(2) in the web content optimization apparatus 12 sends the request to one of the server devices 16(1)-16(n) currently identified as hosting the requested resource. The processor 13(2) in the web content optimization apparatus 12 gets a redirect response from one of the server devices 16(1)-16(n) thought to store the network resource. The processor 13(2) in the web content optimization apparatus 12 optimizes the redirect response to `http://m.acme.com/sample.com/B.html`, and cookie to `SESSION+example.com+%2F=1234`, which is provided to the processor 13(1) in the web content optimization apparatus 12 acting as the requesting client device.

[0051] In step 404c, based on the redirect response the processor 13(1) in the web content optimization apparatus 12 acting as the requesting client device sends an updated HTTP request to the processor 13(2) in the web content optimization apparatus 12 acting as the proxy server device. The processor 13(2) in the web content optimization apparatus 12 sends the updated request to the one of the server devices 16(1)-16(n) identified as hosting the network resource.

[0052] In step 404d, the processor 13(2) in the web content optimization apparatus 12 gets a redirect response from the one of the server devices 16(1)-16(n) identified as hosting the network resource based on the updated request. The processor 13(2) in the web content optimization apparatus 12 optimizes the redirect response to `http://m.acme.com/sample.com/C.html`, and the cookie `LANG+sample.com+%2F=EN`, which is provided to the processor 13(1) in the web content optimization apparatus 12 acting as the requesting client device.

[0053] In step 404e, based on the last redirect response the processor 13(1) in the web content optimization apparatus 12 acting as the requesting client device sends another updated HTTP request to the processor 13(2) in the web content optimization apparatus 12 acting as the proxy server device. The processor 13(2) in the web content optimization apparatus 12 sends the updated request to the one of the server devices 16(1)-16(n) identified as currently hosting the network resource.

[0054] In step 404f, the processor 13(2) in the web content optimization apparatus 12 gets a response from the one of the server devices 16(1)-16(n) with a status identifier message shown as status code 200 set to "OK" indicating the network resource has been found.

[0055] In step 406, when the processor 13(1) in the web content optimization apparatus 12 gets the `http://m.acme.com/m/example.com/C.html` response that is a 200 (i.e. a real resource), then it creates the http response for the requesting one of the client devices 14(1)-14(n) that is a redirect to `http://m.acme.com/m/example.com/C.html` with the two collected cookies from steps 404b and 404d.

[0056] Referring to FIG. 5, an exemplary method for reducing latency in network connections between requesting client devices 14(1)-14(n) and content hosting server devices 16(1)-16(n) is described using flowchart 500 with reference back to FIGS. 1 and 3-4. More specifically, details of operation of web content optimization apparatus 12 with respect to optimization of redirect chain 300 and 400 using core module 21 and redirect module 23 are described. Redirect module 23 is part of a web content optimization system (referred to as "WCOS," not shown) implemented in web content optimization apparatus 12 that is responsible for handling multiple content server redirect response messages from server devices 16(1)-16(n) and sending them to client devices 14(1)-14(n) that originally made the HTTP request (e.g., as shown in steps 202 and 302). In one example, only the last URL of the redirect chain 300 is han-

dled by the redirect module 23. Core module 21 and redirect module 23 reside as part of WCOS on hardware of memory 15. Alternatively, core module 21 and 23 may be implemented at least partially in software residing on a hardware portion of memory 15. The embodiment described in FIG. 5 adds an additional layer of optimization to handling redirect chains 300 and 400 by operating upon prior already optimized requests.

[0057] The method of flowchart 500 begins in step 502 where redirect module 23 inside web content optimization apparatus 12 receives a request from core module 21, which received request was a part of a request originally received by the web content optimization apparatus 12 from the requesting one of the client devices 14(1)-14(n) (e.g., from a web browser on one of the client devices 14(1)-14(n)).

[0058] In step 504, redirect module 23 determines whether an automatic-redirect identifier or flag is enabled and was received as part of the original request for the network resource by the requesting one of the client devices 14(1)-14(n). The automatic-redirect, also referred to as "auto-redirect" herein, is a programming instruction that is used to decide which HTTP requests from client devices 16(1)-16(n) should be optimized for handling by the web content optimization apparatus 12, in accordance with the redirect chain 400 described by FIG. 4, and which requests should be handled directly by the requesting client devices 14(1)-14(n) directly, as shown by FIG. 2 with minimum processing by the web content optimization apparatus 12, although other options could be provided such as the exemplary redirect illustrated and described with reference to FIG. 3. The auto-redirect flag/identifier adds an additional level of optimization because automatic redirection by web content optimization apparatus 12 is used only when needed and when chosen by the programmer or administrator of the web content optimization apparatus 12. If the auto-redirect flag/identifier is not enabled or set to a value "false" set programmatically by a WCOS programmer, the No branch is taken in step 504, where the request is returned to core module 21 by redirect module 23, as shown in step 518. In step 518, the web content optimization apparatus 12 can process the request from the requesting one of the client devices 14(1)-14(n), send a request to one of the server devices 16(1)-16(n), and then optimize the response which is sent back to the requesting one of the client devices 14(1)-14(n).

[0059] However, if the auto-redirect flag/parameter/identifier is set to a value "true," the Yes branch is taken and the flow proceeds to step 506. In step 506, the same HTTP request is sent back to the core module 21 in WCOS with its "auto-redirect" flag set to a value "false." By way of example only, m.acme.com is the host name associated with the web content optimization system residing in memory 15 of the web content optimization apparatus 12. When the auto-redirect parameter is disabled and m.acme.com receives an exemplary request <http://m.acme.com/example.com/A.html>, it sends a re-

quest to <http://example.com/A.html> getting a redirect to <http://sample.com/B.html> with `SESSION=1234` cookie (as shown, for example, in FIG. 2). Accordingly, m.acme.com will send response to the client device with a redirect to <http://m.acme.com/sample.com/B.html> with cookie modified to `SESSION+example.com+%2F=1234`. Modification of the cookie is explained in more detail, by way of example only, in U.S. Patent Application Serial No. 12/660,637, filed March 2, 2010, entitled "Method for Optimizing a Web Content Proxy Server and Devices Thereof". When core module 21 and redirect module 23 receive a request for <http://m.acme.com/sample.com/B.html> from the client device (e.g., client device 14(1)), a request to <http://sample.com/B.html> is sent and then a redirect to <http://example.com/C.html> with `LANG=EN` cookie is received that will be transformed in a redirect response to <http://m.acme.com/example.com/C.html> with `LANG+sample.com+%2F=EN` cookie, to be sent to the client web browser of the client device.

[0060] In another exemplary scenario, automatic-redirect is enabled for first request for <http://m.acme.com/example.com/A.html>. Since automatic-redirect is enabled, the redirect module 23 will decide to follow the redirect chain 300 making the HTTP requests internally to itself within the web content optimization apparatus 12, including all the intermediate redirection requests. Accordingly, redirect module 23 will make HTTP request to <http://m.acme.com/example.com/A.html> with auto redirect disabled. This way, correct optimized cookies will be collected by redirect module 23 and WCOS redirects will be followed. When all the redirect chain 400 is collected as illustrated and described with reference to FIG. 4, a redirect response to <http://m.acme.com/example.com/C.html> will be sent to the client browser having new cookies set to `SESSION+example.com+%2F=1234` and `LANG+sample.com+%2F=EN` that are valid cookies for m.acme.com domain.

[0061] In step 508, the HTTP response is received by core module 21. Sending HTTP requests back to core module 21 of WCOS instead of to hosting one of the server devices 16(1)-16(n) enables the WCOS to optimize the redirect chain 400 of already optimized HTTP responses, although again other optimized redirect chains can be used as illustrated and described with reference to FIG. 3 by way of example. In some examples, core module 21 of WCOS can create longer redirect chains than original content site. By way of example only, the web content optimization system adds new redirect chains to the original server request-response flow. For example, when the web content optimization system has to simulate a complex JavaScript behavior that is available on the original web site page for JavaScript enabled browsers, on its server side JavaScript emulator. Using the auto redirect module 23 over already optimized WCOS pages can advantageously improve the WCOS performance in the web content optimization apparatus 12.

[0062] In step 510, if the response from the redirect module 23 in WCOS optimization system is a redirect response message, then the Yes branch is taken to step 512.

[0063] In step 512, a new HTTP request is created for the redirect URL and the auto-redirect flag is set to a value "false." Further, by way of example only, cookies in the header field are the result of merging the cookies set by the redirect responses with the cookies passed to the first HTTP request.

[0064] In step 514, the core module 21 will then create the appropriate HTTP response to be sent to the redirect module 23. If the HTTP response from the redirect module 23 is a redirect response, the previous steps 510-514 will be repeated one or more times.

[0065] However, in step 510, if the response is a real resource, then the No branch is taken to step 516 where the last redirect response coupled with collected response cookies in the header fields are sent back to the core module 21. By way of example only, the method of flowchart 500 is used when the requesting one of the client devices 14(1)-14(n) is a mobile client device, for example, a cellular telephone.

[0066] Accordingly, as illustrated and described herein this technology provides a number of advantages including providing a method, computer readable medium and an apparatus that reduces latency in network connections, for example, by managing handling of redirect messages from server devices 16(1)-16(n) substantially at web content optimization apparatus 12 interposed between the client devices 14(1)-14(n) and server devices 16(1)-16(n) on which various network resources are stored, which network resources are requested by client devices 14(1)-14(n). With this technology, the web content optimization apparatus 12 optimizes network resource provisioning by returning only the last redirect message along with all cookies collected from intermediate redirect messages from a chain of redirect messages from server devices 16(1)-16(n) to client devices 14(1)-14(n) for a particular network resource, thereby reducing latency in network connections between client devices and server devices.

[0067] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby as well.

[0068] Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims.

Claims

1. A method for reducing latency in network connections, the method comprising:

receiving, at an optimization apparatus (12), a request, from one of one or more client devices (14(1)-14(n)), for a network resource hosted at one of one or more server devices (16(1)-16(n)); determining, at the optimization apparatus (12), whether an automatic redirect parameter is enabled for the request;

sending the request from one proxy processing device acting as the requesting one of the one or more client devices (14(1)-14(n)) within the optimization apparatus (12) to another proxy processing device within the optimization apparatus (12) when the automatic redirect parameter is determined to be enabled;

repeatedly obtaining, at the another proxy processing device within the optimization apparatus (12), a redirect message with one or more optimized cookies associated with the requested network resource from the one or more server devices (16(1)-16(n)), sending the redirect message and the one or more optimized cookies from the another proxy processing device within the optimization apparatus (12) to the one proxy processing device within the optimization apparatus (12), sending from the one proxy processing device back to the another proxy processing device a new request for the requested network resource using the redirect message until the requested network resource is located; and

forwarding, when the requested network resource is located, from the one proxy processing device to the requesting one of the one or more client devices (14(1)-14(n)), only a last of the obtained redirect messages with the one or more optimized cookies associated with the requested network resource.

2. The method as set forth in claim 1 wherein the determining at the optimization apparatus (12) whether the automatic redirect parameter is enabled for the request is based on information contained in the request.
3. The method as set forth in any of claims 1 to 2 further comprising disabling the automatic redirect parameter when the request is sent from the one proxy processing device acting as the requesting one of the one or more client devices (14(1)-14(n)) within the optimization apparatus (12) to the another proxy processing device within the optimization apparatus (12).
4. The method as set forth in any of claims 1 to 3 further

comprising:

processing at the optimization apparatus (12) a revised request for the requested network resource from the requesting one of the one or more client devices (14(1)-14(n)) based on the last of the obtained redirect messages with the one or more optimized cookies;
 obtaining network content at the optimization apparatus (12) from one of the one or more server devices (16(1)-16(n)) based on the revised request;
 optimizing at the optimization apparatus (12) the obtained network content at the optimization apparatus (12) for the requesting one of the one or more client devices (14(1)-14(n)); and
 forwarding from the optimization apparatus (12) the optimized obtained network content to the requesting one of the one or more client devices (14(1)-14(n)).

5. The method as set forth in any of claims 1 to 4, wherein the receiving further comprises receiving at the optimization apparatus (12) the request from the one of the one or more client devices (14(1)-14(n)) comprising a mobile device configured to display the obtained network content on a browser of the mobile device.
6. The method as set forth in any of claims 2 to 5, wherein the information contained in the request comprises a Uniform Resource Identifier (URI).
7. The method as set forth in any of claims 1 to 6, wherein the network resource comprises one or more web pages.
8. A computer readable medium having stored thereon instructions for reducing latency in network connections comprises machine executable code which when executed by at least one processor, causes the processor to perform the steps of the method according to claims 1 to 7.
9. An optimization apparatus (12) configured to reduce latency in network connections comprising one or more processors (13) acting as proxy processing devices and a memory (15) coupled to the one or more processors (13) configured to execute programmed instructions stored in the memory (15) comprising:

receiving a request from one of one or more client devices (14(1)-14(n)) for a network resource hosted at one of one or more server devices (16(1)-16(n));
 determining whether an automatic redirect parameter is enabled for the request;
 sending the request from one proxy processing

device acting as the requesting one of the client devices (14(1)-14(n)) to another proxy processing device within the optimization apparatus (12) when the automatic redirect parameter is determined to be enabled;
 repeatedly obtaining at the another proxy processing device a redirect message with one or more optimized cookies associated with the requested network resource from the one or more server devices (16(1)-16(n)), sending the redirect message and the one or more optimized cookies from the another proxy processing device to the one proxy processing device, and sending from the one proxy processing device back to the another proxy processing device a new request for the requested network resource using the redirect message until the requested network resource is located ; and
 forwarding, when the requested network resource is located, from the one proxy processing device to the requesting one of the one or more client devices (14(1)-14(n)), only a last of the obtained redirect messages with the one or more optimized cookies associated with the requested network resource.

10. The apparatus as set forth in claim 9 wherein the one or more processors (13) is further configured to execute programmed instructions stored in the memory (15) for the determining further comprising determining at the optimization apparatus (12) whether the automatic redirect parameter is enabled for the request based on information contained in the request.
11. The apparatus as set forth in any of claims 9 to 10 wherein the one or more processors (13) is further configured to execute programmed instructions stored in the memory (15) further comprising disabling the automatic redirect parameter when the request is sent from the one proxy processing device acting as the requesting one of the one or more client devices (14(1)-14(n)) within the optimization apparatus (12) to the another proxy processing device within the optimization apparatus (12).
12. The apparatus as set forth in any of claims 9 to 11 wherein the one or more processors (13) is further configured to execute programmed instructions stored in the memory (15) further comprising:

processing a revised request for the requested network resource from the requesting one of the one or more client devices (14(1)-14(n)) based on the last of the obtained redirect messages with the one or more optimized cookies;
 obtaining network content from one of the one or more server devices (16(1)-16(n)) based on

the revised request;
 optimizing the obtained network content at the optimization apparatus (12) for the requesting one of the one or more client devices (14(1)-14(n)); and
 forwarding the optimized obtained network content to the requesting one of the one or more client devices (14(1)-14(n)).

13. The apparatus as set forth in any of claims 9 to 12, wherein the one or more processors (13) is further configured to execute programmed instructions stored in the memory (15) for the receiving further comprising receiving the request from the one of the one or more client devices (14(1)-14(n)) comprising a mobile device configured to display the obtained network content on a browser of the mobile device.
14. The apparatus as set forth in any of claims 10 to 13, wherein the information contained in the request comprises a Uniform Resource Identifier (URI).
15. The apparatus as set forth in any of claims 9 to 14, wherein the network resource comprises one or more web pages.

Patentansprüche

1. Verfahren zum Reduzieren von Verzögerungen in Netzwerkverbindungen, wobei das Verfahren aufweist:
- Erhalten einer Anfrage an einer Optimierungsvorrichtung (12) von einer von einer oder mehreren Client-Vorrichtungen (14(1)-14(n)) für eine Netzwerkressource, die auf einer von einer oder mehreren Server-Vorrichtungen (16(1)-16(n)) gehostet ist;
 - Erkennen an der Optimierungsvorrichtung (12), ob ein automatischer Redirect-Parameter für die Anfrage frei geschaltet ist;
 - Versenden der Anfrage von einer Proxy-Verarbeitungseinheit, die als die Anfragende der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)) innerhalb der Optimierungsvorrichtung (12) agiert, zu einer anderen Proxy-Verarbeitungseinrichtung innerhalb der Optimierungsvorrichtung (12), wenn der automatische Redirect-Parameter als freigeschalten erkannt wurde;
 - Wiederholtes Erhalten an der anderen Proxy-Verarbeitungseinrichtung in der Optimierungsvorrichtung (12) einer Redirect-Nachricht mit einem oder mehreren optimierten Cookies, die mit der angefragten Netzwerkressource der einen oder mehreren Server-Vorrichtungen (16(1)-16(n)) verbunden sind, Versenden der Redirect-

Nachricht und des einen oder der mehreren optimierten Cookies der anderen Proxy-Verarbeitungsvorrichtung innerhalb der Optimierungsvorrichtung (12) zu der einen Proxy-Verarbeitungsvorrichtung innerhalb der Optimierungsvorrichtung (12), Zurücksenden einer neuen Anfrage für die angefragte Netzwerk-Ressource von der einen Proxy-Verarbeitungsvorrichtung an die andere Proxy-Verarbeitungseinrichtung unter Verwendung der Redirect-Nachricht, solange bis die angefragte Netzwerkressource lokalisiert wurde; und
 - Weiterleiten, wenn die angefragte Netzwerkressource lokalisiert ist, nur einer letzten der erhaltenen Redirect-Nachrichten mit einem oder mehreren optimierten Cookies, die mit der nachgefragten Netzwerkressource verbunden sind, von der einen Proxy-Verarbeitungsvorrichtung zu der Anfragenden der einen oder mehreren Client-Vorrichtungen ((14(1)-14(n))).

2. Verfahren nach Anspruch 1, wobei das Erkennen an der Optimierungsvorrichtung (12), ob der automatische Redirect-Parameter freigeschaltet ist für die Anfrage basierend auf Informationen, die in der Anfrage enthalten sind.
3. Verfahren nach einem der Ansprüche 1 bis 2, weiter aufweisend, Deaktivieren des automatischen Redirect-Parameters, wenn die Anfrage von der einen Proxy-Verarbeitungsvorrichtung, die als die Anfragende der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)) innerhalb der Optimierungsvorrichtung (12) agiert, zu der anderen Proxy-Verarbeitungsvorrichtung innerhalb der Optimierungsvorrichtung (12) versendet wird.
4. Verfahren nach einem der Ansprüche 1 bis 3 weiter aufweisend:
- Verarbeiten in der Optimierungsvorrichtung (12) einer überarbeiteten Anfrage für die von der Anfragenden der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)) angefragten Netzwerkressource basierend auf der letzten erhaltenen Redirect-Nachricht mit einem oder mehreren optimierten Cookies;
 - Erhalten von Netzwerkinhalt an der Optimierungsvorrichtung (12) von einer der einen oder mehreren Server-Vorrichtungen (16(1)-16(n)), basierend auf der überarbeiteten Anfrage;
 - Optimieren des an der Optimierungsvorrichtung (12) erhaltenen Netzwerkinhalts in der Optimierungsvorrichtung (12) für die Anfragende der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)); und
 - Weiterleiten durch die Optimierungsvorrichtung (12) des erhaltenen optimierten Netz-

werksinhalts an die Anfragende der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)).

5. Verfahren nach einem der Ansprüche 1 bis 4, wobei das Erhalten weiter das Erhalten der Anfrage der einen der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)) an der Optimierungsvorrichtung (12) aufweist, welche eine Mobilvorrichtung aufweist, die ausgebildet ist, den erhaltenen Netzwerkinhalt in einem Browser der Mobilvorrichtung anzuzeigen. 5 10
6. Verfahren nach einem der Ansprüche 2 bis 5, wobei die in der Anfrage enthaltene Information einen Uniform Resource Identifier (URI) aufweist. 15
7. Verfahren nach einem der Ansprüche 1 bis 6, wobei die Netzwerkressource eine oder mehrere Webseiten aufweist.
8. Computerlesbares Medium auf dem Instruktionen zum Reduzieren von Verzögerungen in Netzwerkverbindungen gespeichert sind, aufweisend Maschinenausführungscode, der, wenn durch zumindest einen Prozessor ausgeführt, den Prozessor veranlasst, die Schritte des Verfahrens gemäß den Ansprüchen 1 bis 7 auszuführen. 20 25
9. Optimierungsvorrichtung (12), ausgebildet zum Reduzieren von Verzögerungen in Netzwerkverbindungen, aufweisend einen oder mehrere Prozessoren (13), die als Proxy-Verarbeitungsvorrichtungen agieren sowie einen Speicher (15), der mit einem oder mehreren Prozessoren (13) verbunden ist, die ausgebildet sind, um programmierte Instruktionen, die in dem Speicher (15) gespeichert sind, auszuführen, aufweisend: 30 35
 - Erhalten einer Anfrage für eine Netzwerkressource, die auf einem von einer oder mehreren Servervorrichtungen (16(1)-16(n)) gehostet ist, von einer von einer oder mehreren Client-Vorrichtungen (14(1)-14(n)); 40
 - Erkennen, ob ein automatischer Redirect-Parameter für die Anfrage freigeschaltet ist;
 - Versenden der Anfrage der einen Proxy-Verarbeitungsvorrichtung, die als die Anfragende der Client-Vorrichtungen (14(1)-14(n)) agiert, zu einer weiteren Proxy-Verarbeitungsvorrichtung in der Optimierungsvorrichtung (12), wenn der automatische Redirect-Parameter als freigeschaltet erkannt ist; 50
 - Wiederholtes Erhalten an der weiteren Proxy-Verarbeitungsvorrichtung einer Redirect-Nachricht mit einem oder mehreren optimierten Cookies, die mit der von der einen oder mehreren Server-Vorrichtungen (16(1)-16(n)) angefragten Netzwerkressource verbunden sind, Senden der Redirect-Nachricht und des einen 55

oder mehreren optimierten Cookies der anderen Proxy-Verarbeitungsvorrichtung zu der einen Proxy-Verarbeitungsvorrichtung und Zurücksenden von der einen Proxy-Verarbeitungsvorrichtung zu der anderen Proxy-Verarbeitungsvorrichtung einer neuen Anfrage für die angefragte Netzwerkressource unter Verwendung der Redirect-Nachricht, solange bis die angefragte Netzwerkressource lokalisiert ist; und
 - Weiterleiten, wenn die angefragte Netzwerkressource lokalisiert ist, nur einer letzten der erhaltenen Redirect-Nachrichten mit einem oder mehreren optimierten Cookies, die mit der nachgefragten Netzwerkressource verbunden sind, von der einen Proxy-Verarbeitungsvorrichtung zu der Anfragenden der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)).

10. Vorrichtung nach Anspruch 9, wobei der eine oder mehrere Prozessor (13) weiter ausgebildet ist, um programmierte Instruktionen, die in dem Speicher (15) zum Erkennen gespeichert sind, aufweisend weiter ein Erkennen an der Optimierungsvorrichtung (12), ob der automatische Redirect-Parameter freigeschaltet ist für die Anfrage, die auf Information basiert, die in der Anfrage enthalten sind.
11. Vorrichtung nach einem der Ansprüche 9 bis 10, wobei der eine oder mehrere Prozessor (13) weiter ausgebildet sind, um programmierte Instruktionen, die in dem Speicher (15) gespeichert sind, auszuführen, aufweisend weiter ein Abschalten des automatischen Redirect-Parameters, wenn die Anfrage von der einen Proxy-Verarbeitungsvorrichtung, die als die Anfragende der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)) innerhalb der Optimierungsvorrichtung (12) agiert, zu der anderen Proxy-Verarbeitungsvorrichtung innerhalb der Optimierungsvorrichtung (12) gesendet wird.
12. Vorrichtung nach einen der Ansprüche 9 bis 11, wobei der eine oder die mehreren Prozessoren (13) weiter ausgebildet sind, um programmierte Instruktionen, die in dem Speicher (15) gespeichert sind, auszuführen weiter aufweisend:
 - Verarbeiten einer überarbeiteten Anfrage für die angefragte Netzwerkressource der Anfragenden der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)), basierend auf der letzten der erhaltenen Redirect-Nachrichten mit einem oder mehreren optimierten Cookies;
 - Erhalten von Netzwerkinhalt von einer der einen oder mehreren Server-Vorrichtungen (16(1)-16(n)) basierend auf der überarbeiteten Anfrage;
 - Optimierung des erhaltenen Netzwerkinhalts in der Optimierungsvorrichtung (12) für die An-

fragende der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)); und
- Weiterleiten des optimierten erhaltenen Netzwerkinhalts zu der Anfragenden der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)).

13. Vorrichtung nach einem der Ansprüche 9 bis 12, wobei der eine oder die mehreren Prozessoren (13) weiter ausgebildet sind, um programmierte Instruktionen, die in dem Speicher (15) für das Erhalten gespeichert sind, auszuführen, aufweisend weiter ein Erhalten der Anfrage von der einen der einen oder mehreren Client-Vorrichtungen (14(1)-14(n)), die eine mobile Vorrichtung aufweist, die ausgebildet ist, um den erhaltenen Netzwerkinhalt auf einem Browser der mobilen Vorrichtung anzuzeigen.
14. Vorrichtung nach einem der Ansprüche 10 bis 13, wobei die in der Anfrage enthaltene Information einen Uniform Resource Identifier(URI) aufweist.
15. Vorrichtung nach einem der Ansprüche 9 bis 14, wobei die Netzwerkressource eine oder mehrere Webseiten aufweist.

Revendications

1. Procédé de réduction de la latence dans des connexions de réseau, le procédé comprenant :

la réception, au niveau d'un appareil d'optimisation (12), d'une demande, en provenance d'un ou de plusieurs dispositifs clients (14(1)-14(n)), pour une ressource de réseau hébergée au niveau d'un parmi un ou plusieurs dispositifs serveurs (16(1)-16(n)) ;
le fait de déterminer, au niveau de l'appareil d'optimisation (12), si un paramètre de redirection automatique est activé pour la demande ;
l'envoi de la demande depuis un dispositif de traitement de serveur mandataire agissant comme celui demandant l'un parmi les un ou plusieurs dispositifs clients (14(1)-14(n)) au sein de l'appareil d'optimisation (12) à un autre dispositif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12) lorsque le paramètre de redirection automatique est déterminé comme étant activé ;
l'obtention de manière répétée, au niveau de l'autre dispositif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12), d'un message de redirection avec un ou plusieurs témoins optimisés associés à la ressource de réseau demandée à partir des un ou plusieurs dispositifs serveurs (16(1)-16(n)), l'envoi du message de redirection et des un ou plusieurs témoins optimisés à partir de l'autre dis-

positif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12) au un dispositif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12), l'envoi en retour depuis le un dispositif de traitement de serveur mandataire vers l'autre dispositif de traitement de serveur mandataire d'une nouvelle demande pour la ressource de réseau demandée à l'aide du message de redirection jusqu'à ce que la ressource de réseau demandée soit localisée ; et
le réacheminement, lorsque la ressource de réseau demandée est localisée, depuis le un dispositif de traitement de serveur mandataire vers le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)), uniquement d'un dernier des messages de redirection obtenus avec les un ou plusieurs témoins optimisés associés à la ressource de réseau demandée.

2. Procédé selon la revendication 1, dans lequel le fait de déterminer au niveau de l'appareil d'optimisation (12) si le paramètre de redirection automatique est activé pour la demande est basé sur des informations contenues dans la demande.

3. Procédé selon l'une quelconque des revendications 1 à 2, comprenant en outre la désactivation du paramètre de redirection automatique lorsque la demande est envoyée du un dispositif de traitement de serveur mandataire agissant comme le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)) au sein de l'appareil d'optimisation (12) vers l'autre dispositif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12).

4. Procédé selon l'une quelconque des revendications 1 à 3, comprenant en outre :

le traitement au niveau de l'appareil d'optimisation (12) d'une demande révisée pour la ressource de réseau demandée à partir du demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)) en se basant sur le dernier des messages de redirection obtenus avec les un ou plusieurs témoins optimisés ;
l'obtention d'un contenu de réseau au niveau de l'appareil d'optimisation (12) à partir de l'un des un ou plusieurs dispositifs serveurs (16(1)-16(n)) en se basant sur la demande révisée ;
l'optimisation au niveau de l'appareil d'optimisation (12) du contenu de réseau obtenu au niveau de l'appareil d'optimisation (12) pour le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)) ; et
le réacheminement depuis l'appareil d'optimisation (12) du contenu de réseau obtenu optimisé vers le demandeur des un ou plusieurs disposi-

tifs clients (14(1)-14(n)).

5. Procédé selon l'une quelconque des revendications 1 à 4, dans lequel la réception comprend en outre la réception au niveau de l'appareil d'optimisation (12) de la demande depuis le un des un ou plusieurs dispositifs clients (14(1)-14(n)) comprenant un dispositif mobile configuré pour afficher le contenu de réseau obtenu sur un navigateur du dispositif mobile.
6. Procédé selon l'une quelconque des revendications 2 à 5, dans lequel les informations contenues dans la demande comprennent un identifiant de ressource uniforme (URI).
7. Procédé selon l'une quelconque des revendications 1 à 6, dans lequel la ressource de réseau comprend une ou plusieurs pages Web.
8. Support lisible par ordinateur sur lequel sont stockées des instructions permettant de réduire la latence dans des connexions de réseau comprenant un code exécutable par machine qui, lorsqu'il est exécuté par au moins un processeur, amène le processeur à réaliser les étapes du procédé selon les revendications 1 à 7.
9. Appareil d'optimisation (12) configuré pour réduire la latence dans des connexions de réseau comprenant un ou plusieurs processeurs (13) agissant comme des dispositifs de traitement de serveur mandataire et une mémoire (15) couplée aux un ou plusieurs processeurs (13) configurés pour exécuter des instructions programmées stockées dans la mémoire (15) comprenant :

la réception d'une demande provenant d'un parmi un ou plusieurs dispositifs clients (14(1)-14(n)) pour une ressource de réseau hébergée au niveau de l'un des un ou plusieurs dispositifs serveurs (16(1)-16(n)) ;

le fait de déterminer si un paramètre de redirection automatique est activé pour la demande ; l'envoi de la demande depuis un dispositif de traitement de serveur mandataire agissant comme le demandeur d'un des dispositifs clients (14(1)-14(n)) à un autre dispositif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12) lorsque le paramètre de redirection automatique est déterminé comme étant activé ;

l'obtention de manière répétée au niveau de l'autre dispositif de traitement de serveur mandataire d'un message de redirection avec un ou plusieurs témoins optimisés associés à la ressource de réseau demandée à partir des un ou plusieurs dispositifs serveurs (16(1)-16(n)), l'envoi du message de redirection et des un ou plu-

sieurs témoins optimisés depuis l'autre dispositif de traitement de serveur mandataire vers un dispositif de traitement de serveur mandataire, et l'envoi en retour depuis le un dispositif de traitement de serveur mandataire vers l'autre dispositif de traitement de serveur mandataire d'une nouvelle demande pour la ressource de réseau demandée à l'aide du message de redirection jusqu'à ce que la ressource de réseau demandée soit localisée ; et

le réacheminement, lorsque la ressource de réseau demandée est localisée, depuis le un dispositif de traitement de serveur mandataire vers le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)), uniquement d'un dernier des messages de redirection obtenus avec les un ou plusieurs témoins optimisés associés à la ressource de réseau demandée.

10. Appareil selon la revendication 9, dans lequel les un ou plusieurs processeurs (13) sont en outre configurés pour exécuter des instructions programmées stockées dans la mémoire (15) pour la détermination comprenant en outre le fait de déterminer au niveau de l'appareil d'optimisation (12) si le paramètre de redirection automatique est activé pour la demande en se basant sur des informations contenues dans la demande.

11. Appareil selon l'une quelconque des revendications 9 à 10, dans lequel les un ou plusieurs processeurs (13) sont en outre configurés pour exécuter des instructions programmées stockées dans la mémoire (15) comprenant en outre la désactivation du paramètre de redirection automatique lorsque la demande est envoyée depuis le un dispositif de traitement de serveur mandataire agissant comme le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)) au sein de l'appareil d'optimisation (12) vers l'autre dispositif de traitement de serveur mandataire au sein de l'appareil d'optimisation (12).

12. Appareil selon l'une quelconque des revendications 9 à 11, dans lequel les un ou plusieurs processeurs (13) sont en outre configurés pour exécuter des instructions programmées stockées dans la mémoire (15) comprenant en outre :

le traitement d'une demande révisée pour la ressource de réseau demandée à partir du demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)) en se basant sur le dernier des messages de redirection obtenus avec les un ou plusieurs témoins optimisés ;

l'obtention d'un contenu de réseau à partir de l'un des un ou plusieurs dispositifs serveurs (16(1)-16(n)) en se basant sur la demande révisée ;

l'optimisation du contenu de réseau obtenu au niveau de l'appareil d'optimisation (12) pour le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)) ; et
 le réacheminement du contenu de réseau obtenu optimisé vers le demandeur des un ou plusieurs dispositifs clients (14(1)-14(n)).

5

13. Appareil selon l'une quelconque des revendications 9 à 12, dans lequel les un ou plusieurs processeurs (13) sont en outre configurés pour exécuter des instructions programmées stockées dans la mémoire (15) pour la réception comprenant en outre la réception de la demande provenant de l'un des un ou plusieurs dispositifs clients (14(1)-14(n)) comprenant un dispositif mobile configuré pour afficher le contenu de réseau obtenu sur un navigateur du dispositif mobile.

10

15

14. Appareil selon l'une quelconque des revendications 10 à 13, dans lequel les informations contenues dans la demande comprennent un identifiant de ressource uniforme (URI).

20

15. Appareil selon l'une quelconque des revendications 9 à 14, dans lequel la ressource de réseau comprend une ou plusieurs pages Web.

25

30

35

40

45

50

55

15

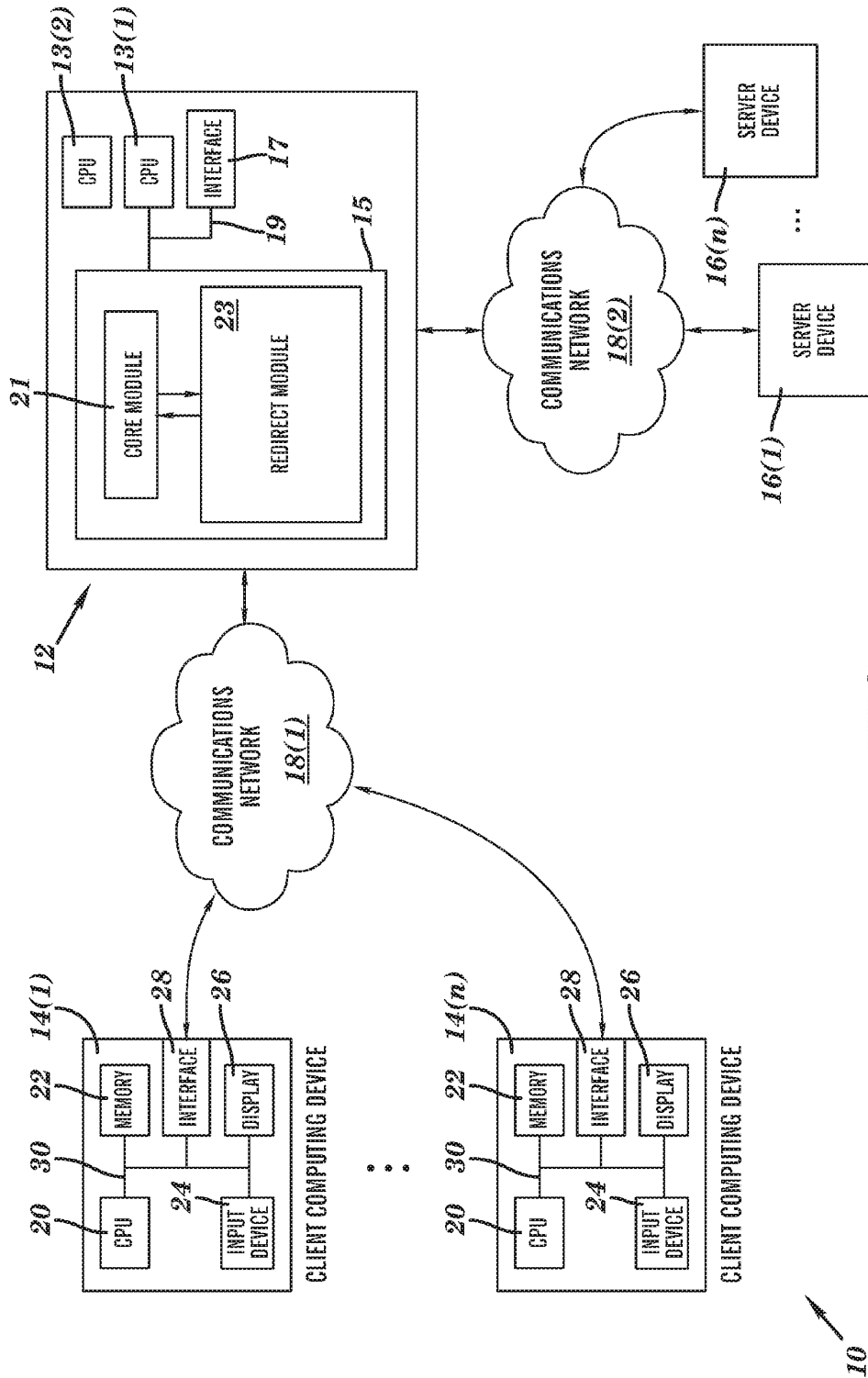


FIG. 1

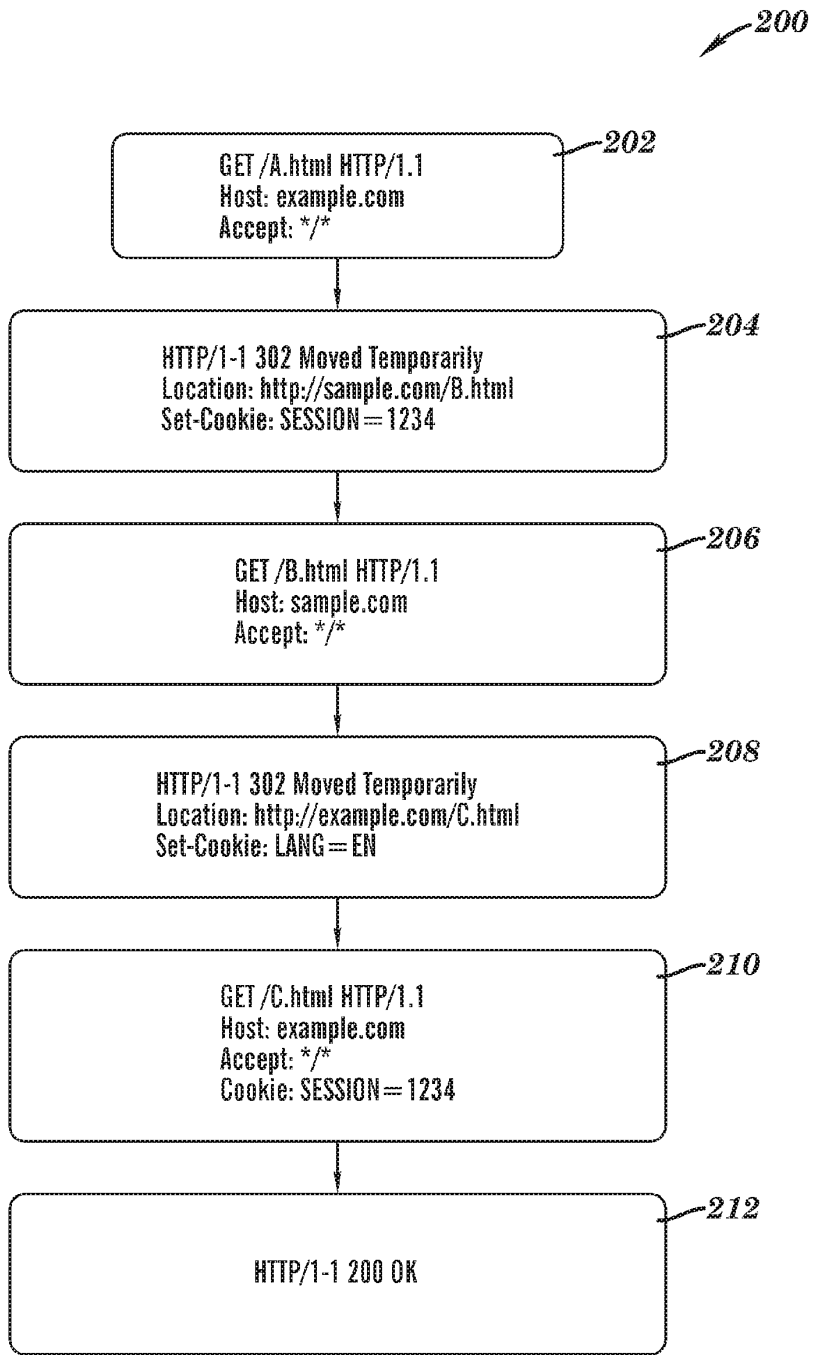


FIG. 2

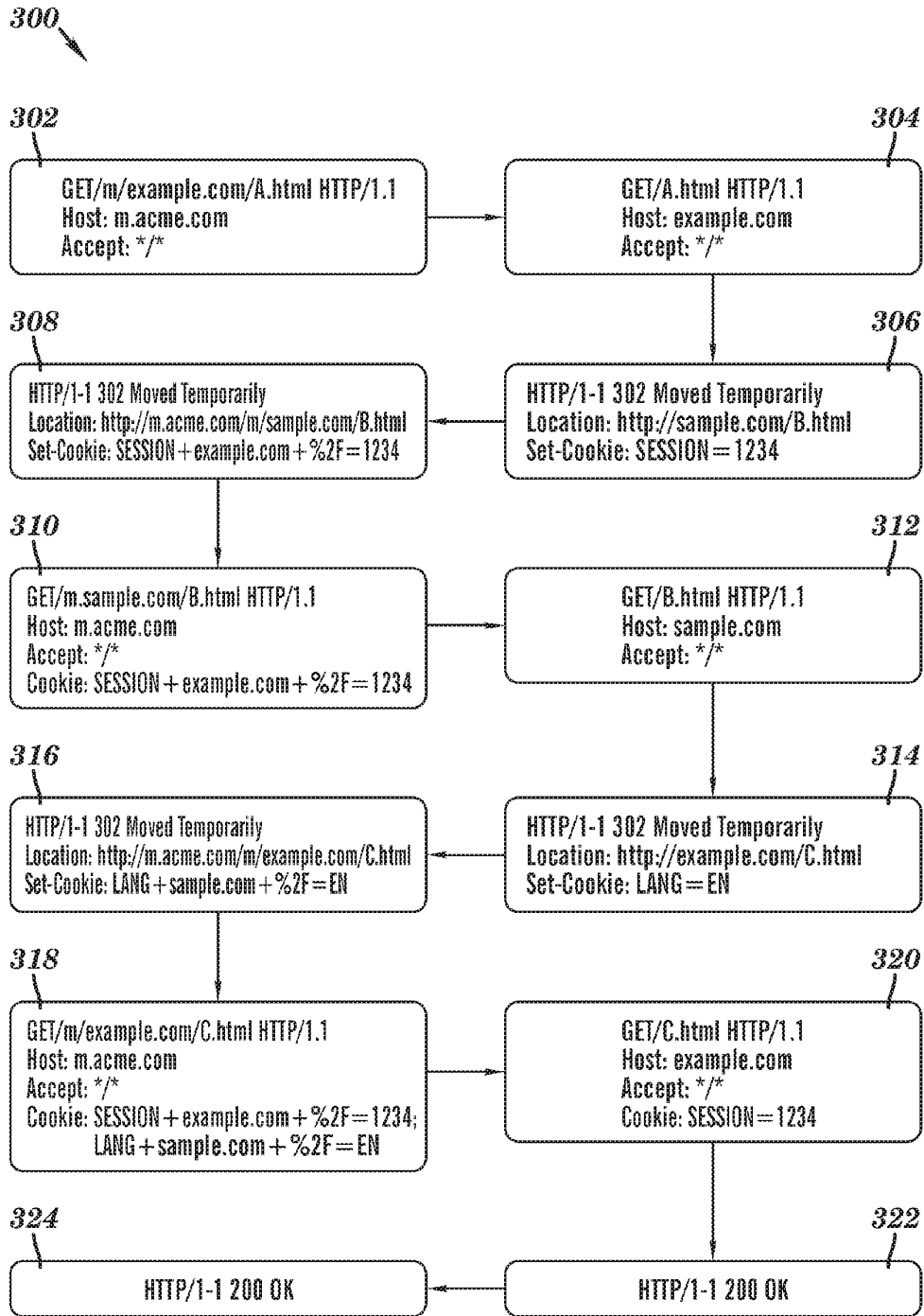


FIG. 3

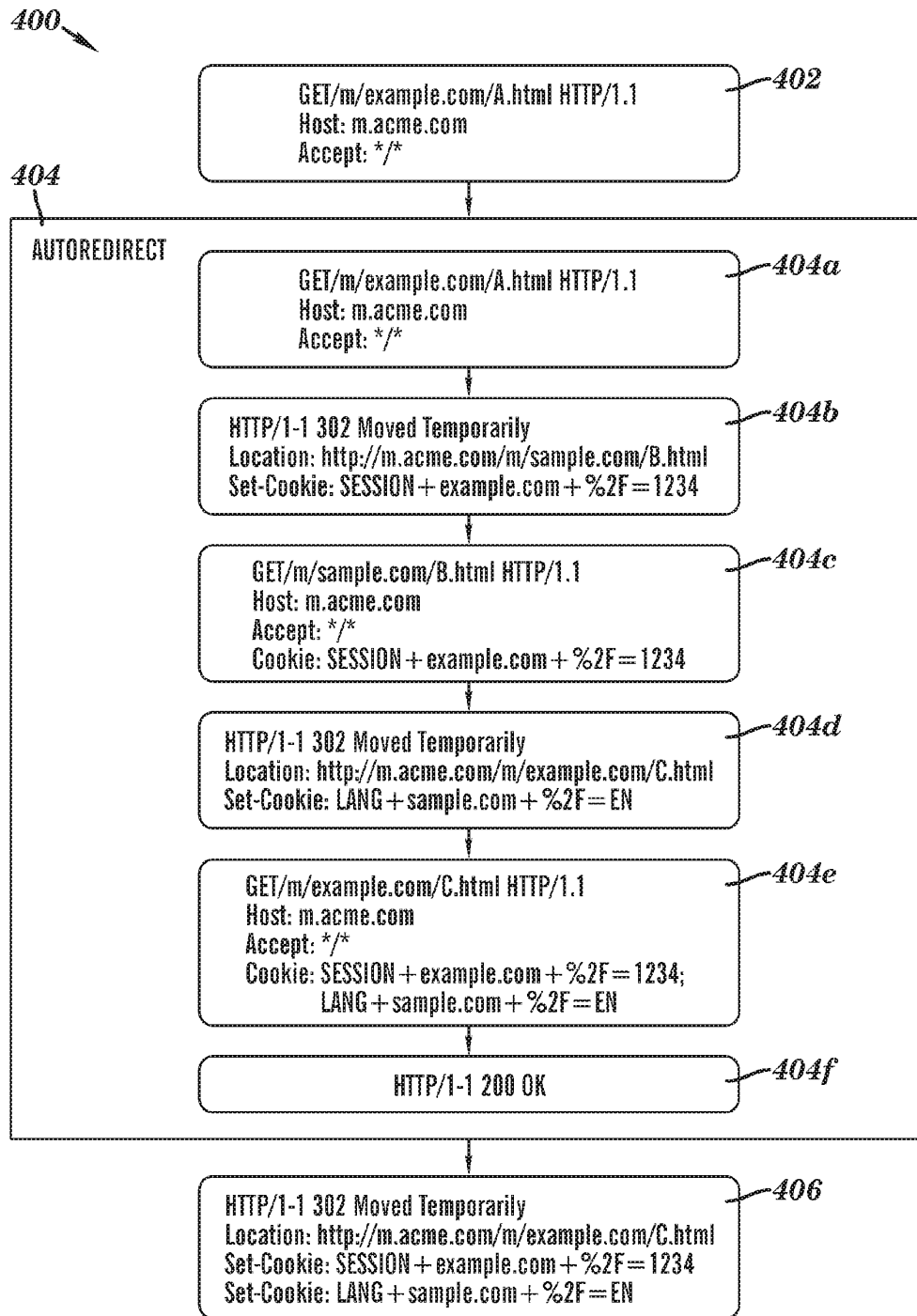


FIG. 4

500

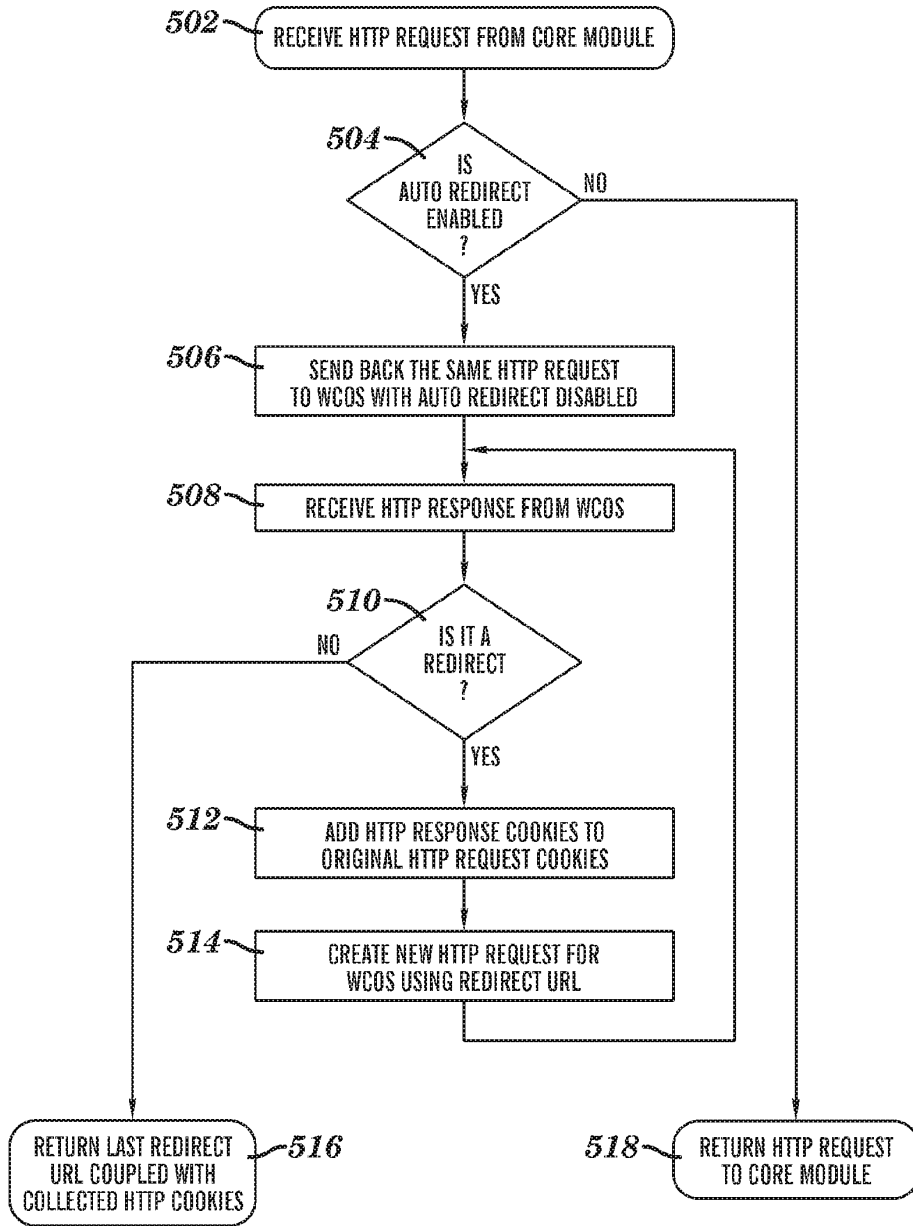


FIG. 5

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20080195696 A [0004]
- US 20040215717 A [0004]
- US 66063710 A [0059]

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 2 656 303 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2012/088326 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2012/088326 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2012/088326 (art. 153(3) CBE).

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(10) International Publication Number
WO 2012/088326 A1

(43) International Publication Date
28 June 2012 (28.06.2012)

- (51) International Patent Classification:
G06Q 50/32 (2012.01)
- (21) International Application Number:
PCT/US2011/066611
- (22) International Filing Date:
21 December 2011 (21.12.2011)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
12/975,990 22 December 2010 (22.12.2010) US
- (71) Applicant (for all designated States except US): **US-ABLENET INC.** [US/US]; 28 W. 23rd Street, 6th Floor, New York, NY 10010 (US).
- (72) Inventors; and
(75) Inventors/Applicants (for US only): **TAYLOR, Jason** [GB/GB]; 93 North Street, Caine, Wiltshire SN11 0EN (GB), **SPERETTA, Mirco** [IT/US]; 367 Woodridge Avenue, Fairfield, CT 06825 (US).
- (74) Agents: **LEINBERG, Gunnar, G.** et al.; LeClairRyan, 70 Linden Oaks, Rochester, NY 14625 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:
— with international search report (Art. 21(3))

[Continued on next page]

(54) Title: METHODS FOR EMAILING LABELS AS PORTABLE DATA FILES AND DEVICES THEREOF

(57) Abstract: A method, computer readable medium and device that enables emailing a label as a portable data file to destination information associated with a mobile computing device includes obtaining at a proxy server device a web page with a label from a web server device in response to a request from a mobile computing device. The proxy server device obtains destination information for the label associated with the requesting mobile computing device. The proxy server device converts the label into a portable data file and transmits the portable data file using the destination information for the label associated with the requesting mobile computing device.

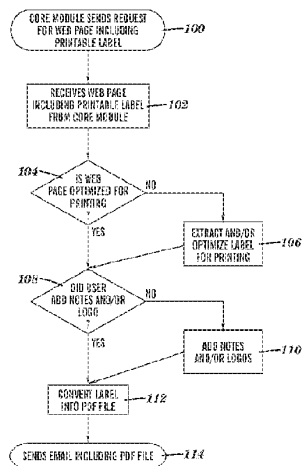


FIG. 3

WO 2012/088326 A1

--- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

- 1 -

METHODS FOR EMAILING LABELS AS PORTABLE DATA FILES AND DEVICES THEREOF

FIELD

[0001] This technology relates to methods for emailing labels as portable
5 data files and devices thereof.

BACKGROUND

[0002] Some Web based applications, usually developed for desktop
browsers, provide the functionality of printing labels, such as bar codes, proof of
insurance, medical records, or bank statements. All these documents have a well-
10 defined printing format, can include personal information and usually are
generated from password protected Web content.

[0003] A desktop browser can easily exploit this type of printing
functionalities. Unfortunately, mobile browsers do not have the same capability
since they usually lack of a direct connection with a printing device.

15

SUMMARY

[0004] A method for emailing a label as a portable data file to destination
information associated with a mobile computing device includes obtaining at a
proxy server device a web page with a label from a web server device in response
to a request from a mobile computing device. The proxy server device obtains
20 destination information for the label associated with the requesting mobile
computing device. The proxy server device converts the label into a portable data
file and transmits the portable data file using the destination information for the
label associated with the requesting mobile computing device.

[0005] A non-transitory computer readable medium having stored thereon
25 instructions for emailing a label as a portable data file to destination information
associated with a mobile computing device comprising machine executable code
which when executed by at least one processor, causes the processor to perform
steps including obtaining a web page with a label from a web server device in

- 2 -

response to a request from a mobile computing device. Destination information for the label associated with the requesting mobile computing device is obtained. The label is converted into a portable data file and then is transmitted using the destination information for the label associated with the requesting mobile
5 computing device.

[0006] A proxy server device includes one or more processors and a memory coupled to the one or more processors which are configured to execute programmed instructions stored in the memory including obtaining a web page with a label from a web server device in response to a request from a mobile
10 computing device. Destination information for the label associated with the requesting mobile computing device is obtained. The label is converted into a portable data file and then is transmitted using the destination information for the label associated with the requesting mobile computing device.

[0007] This technology provides a number of advantages including
15 providing a method, computer readable medium and an apparatus that enables a mobile computing device to create a portable data file of one or more labels and send the portable data file, along with custom information, such as descriptions and logos, to a designated email address associated with the mobile computing device. Accordingly, with this technology data files that are not viewable on
20 mobile computing devices can be downloaded and printed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 is a block diagram of an exemplary environment with an exemplary proxy server device;

[0009] FIG. 2 is a screen shot of an exemplary confirmation page with
25 entry fields for data to generate and send a portable data file; and

[00010] FIG. 3 is a flow chart of an exemplary method for generating and emailing labels as portable data files and devices thereof.

DETAILED DESCRIPTION

[00011] An exemplary environment 10 with a proxy server device 12 configured to enable one of the mobile computing devices 14(1)-14(n) to download a printable version of a label that was converted into PDF format is illustrated in FIG. 1. The exemplary environment 10 includes the proxy server device 12, mobile computing devices 14(1)-14(n), web content server devices 16(1)-16(n), and communication networks 18(1)-18(2), although other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can be used. This technology provides a number of advantages including providing a method, computer readable medium and an apparatus that enables a mobile computing device to create a portable data file of one or more labels and send the portable data file, along with custom information, such as descriptions and logos, to a designated email address associated with a mobile computing device.

[00012] Referring more specifically to FIG. 1, the proxy server device 12 is a web content optimization device configured to execute the exemplary methods and other technology illustrated and described herein, although the proxy server can comprise other types of computing devices configured to execute the exemplary methods illustrated and described herein. This exemplary proxy server device 12 includes a central processing unit (CPU) or processor 13, a memory 15, and an interface system 17 which are coupled together by a bus 19 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor 13 in the proxy server device 12 executes a program of stored instructions one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[00013] The memory 15 in the proxy server device 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different

- 4 -

types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 13, can be used for the memory 15 in the proxy server device 12.

[00014] The interface system 17 in the proxy server device 12 is used to operatively couple and communicate between the proxy server device 12 and the mobile computing devices 14(1)-14(n), and the web content server devices 16(1)-16(n) via the communication networks 18(1)-18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. By way of example only, the communication networks 18(1)-18(2) can use TCP/IP over Ethernet and industry-standard protocols, including HTTP, HTTPS, WAP, and SOAP, although other types and numbers of communication networks, such as a direct connection, a local area network, a wide area network, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[00015] Each of the mobile computing devices 14(1)-14(n) enables a user to request, get and interact with documents and other files from one or more web sites hosted by the web content server devices 16(1)-16(n) through the proxy server device 12 via one or more communication networks, although one or more of the mobile computing devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. Although multiple mobile computing devices 14(1)-14(n) are shown, other numbers and types of user computing systems could be used.

[00016] Each of mobile computing devices 14(1)-14(n) in this example is a computing device that includes a central processing unit (CPU) or processor 20, a memory 22, user input device 24, a display 26, and an interface system 28, and which are coupled together by a bus 30 or other link, although one or more of

- 5 -

mobile computing devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 20 in each of mobile computing devices 14(1)-14(n) can execute a program of stored instructions for one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[00017] The memory 22 in each of the mobile computing devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 20 can be used for the memory 22 in each of the mobile computing devices 14(1)-14(n).

[00018] The user input device 24 in each of the mobile computing devices 14(1)-14(n) is used to input request, selections and other data, although the user input device could provide other functions and interact with other elements. The user input device can include keypads, touch screens, and/or vocal input processing systems although other types and numbers of user input devices can be used.

[00019] The display 26 in each of the mobile computing devices 14(1)-14(n) is used to show data and information to the user, such as a website page optimized for viewing on a mobile computing device by way of example only. The display in each of the mobile computing devices 14(1)-14(n) is a computer screen display, although other types and numbers of displays could be used depending on the particular type of mobile device.

[00020] The interface system 28 in each of the mobile computing devices 14(1)-14(n) is used to operatively couple and communicate between the mobile

- 6 -

computing devices 14(1)-14(n) and the proxy server device 12, and the web content server devices 16(1)-16(n) over the communication networks 18(1)-18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

5 [00021] The web content server devices 16(1)-16(n) provide one or more pages from one or more web sites, although the web content server devices 16(1)-16(n) can provide other numbers and types of applications and/or content and can have provide other numbers and types of functions. Although web content server devices 16(1)-16(n) are shown for ease of illustration and discussion, other
10 numbers and types of web server systems and devices can be used.

[00022] Each of the web content server devices 16(1)-16(n) include a central processing unit (CPU) or processor, a memory, and an interface system which are coupled together by a bus or other link, although each of the web content server devices 16(1)-16(n) could have other numbers and types of
15 components, parts, devices, systems, and elements in other configurations and locations can be used. The processor in each of the web content server devices 16(1)-16(n) executes a program of stored instructions one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed
20 instructions.

[00023] The memory in each of the web content server devices 16(1)-16(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated by way of the embodiments, although some or all of the programmed instructions could be stored and/or executed elsewhere.
25 A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor, can be used for the memory in each of the
30 web content server devices 16(1)-16(n).

- 7 -

[00024] The interface system in each of the web content server devices 16(1)-16(n) is used to operatively couple and communicate between the web content server devices 16(1)-16(n) and the proxy server device 12, the mobile computing devices 14(1)-14(n), and the client computing devices 15(1)-15(n) via communication networks 18(1)-18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[00025] Although embodiments of the proxy server device 12, the mobile computing devices 14(1)-14(n), and the web content server devices 16(1)-16(n), are described and illustrated herein, each can be implemented on any suitable computer system or computing device. It is to be understood that the devices and systems of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[00026] Furthermore, each of the systems of the embodiments may be conveniently implemented using one or more general purpose computer systems, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[00027] In addition, two or more computing systems or devices can be substituted for any one of the systems in any embodiment of the embodiments. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the embodiments. The embodiments may also be implemented on computer system or systems that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched

- 8 -

Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[00028] The embodiments may also be embodied as non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the 5 embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[00029] An exemplary method for emailing labels as portable data files and 10 devices thereof will now be described with reference to FIGS. 1-3. Referring more specifically to FIG. 2, in step 100 the proxy server device 12 receives an HTTP request for a web page from one of the mobile computing devices 14(1)-14(n). The proxy server device 12 transmits the request to one of the web server devices 16(1)-16(n) hosting the website with the requested content.

15 [00030] In step 102, the proxy server device 12 receives the requested content, such as a web page including a printable label from the one of the web server devices 16(1)-16(n) hosting the website with the requested content. The printable label can comprise a variety of different printable content which is not viewable on the requesting one of the mobile computing devices 14(1)-14(n), such 20 as a bar code, proof of insurance, medical record, or bank statement by way of example.

[00031] In step 104, the proxy server device 12 determines whether the web page with the printable label from the one of the web server devices 16(1)-16(n) is optimized for printing. If in step 104, the proxy server device 12 25 determines the web page with the printable label is not optimized for printing, then the No branch is taken to step 106. In step 106, the proxy server device 12 extracts and/or optimizes the printable label for printing and then proceeds to step 108. If in step 104, the proxy server device 12 determines the web page with the printable label is optimized for printing, then the Yes branch is taken to step 108.

- 9 -

[00032] In step 108, the proxy server device 12 determines whether a user at the requesting one of the mobile computing devices 14(1)-14(n) has provided notes and/or logos as well as an email address to receive the generate portable data file, although other types of data and destination information could be used. If in
5 step 108 the proxy server device 12 determines the user at the requesting one of the mobile computing devices 14(1)-14(n) has not provided any notes and/or logos as well as an email address associated with the requesting one of the mobile computing devices 14(1)-14(n), then the No branch is taken to step 110. In step
10 FIG. 2, to the requesting one of the mobile computing devices 14(1)-14(n) to obtain notes and/or logos as well as an email address associated with the requesting one of the mobile computing devices 14(1)-14(n) to receive the label, although other manners of obtaining customizing and addressing information relating to the label could be used. The proxy server device 12 adds the obtained
15 notes and/or logos or other data from entries in the fields shown in FIG. 2 to customize the label and then proceeds to step 112.

[00033] If in step 108 the proxy server device 12 determines the user at the requesting one of the mobile computing devices 14(1)-14(n) has not added notes and/or logos for the label and already has provided an email address, then
20 the Yes branch is taken to step 112. In step 112, the proxy server device 12 converts the customized label into a PDF file, although the label can be converted in other numbers and types of portable data files for transmission.

[00034] In step 114, the proxy server device 12 transmits the generated label to the email address associated with the requesting one of the mobile
25 computing devices 14(1)-14(n). The proxy server device 12 may optionally send a confirmation to the requesting one of the mobile computing devices 14(1)-14(n) that the PDF or other portable data file has been sent. The user associated with the requesting one of the mobile computing devices 14(1)-14(n) can later access the email account through for example a desktop computing device connected to a
30 printer to print the customized label which was not viewable on the requesting one of the mobile computing devices 14(1)-14(n).

- 10 -

[00035] Accordingly, as illustrated and described herein this technology provides a number of advantages including providing a method, computer readable medium and an apparatus that enables a mobile computing device to create a portable data file of one or more labels and send the portable data file, along with custom information, such as descriptions and logos, to a designated email address associated with the mobile computing device. One of the advantages of this technology is that mobile computing devices can now obtain labels that were not viewable on the mobile computing device.

[00036] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

- 11 -

CLAIMS

What is claimed is:

1. A method for emailing a label as a portable data file to destination information associated with a mobile computing device, the method
5 comprising:
 - obtaining at a proxy server device a web page with a label from a web server device in response to a request from a mobile computing device;
 - 10 obtaining with the proxy server device destination information for the label associated with the requesting mobile computing device;
 - converting with the proxy server device the label into a portable data file; and
 - 15 transmitting with the proxy server device the portable data file using the destination information for the label associated with the requesting mobile computing device.
2. The method as set forth in claim 1 further comprising:
 - 20 determining with the proxy server device whether the web page with the label is optimized for printing; and
 - extracting and optimizing with the proxy server device the label for printing when the determining indicates the web page and label are not optimized for printing.
3. The method as set forth in claim 1 further comprising:
 - 25 determining with the proxy server device whether the requesting mobile computing device has provided at least one of notes and a logo to customize the label; and
 - 30 customizing with the proxy server device the label when the requesting mobile computing device has provided at least one of the notes and the logo.

- 12 -

4. The method as set forth in claim 1 wherein the portable data file is a PDF file.

5. The method as set forth in claim 1 wherein the obtained destination information comprises an email address.

6. A non-transitory computer readable medium having stored thereon instructions for emailing a label as a portable data file to destination information associated with a mobile computing device comprising machine executable code which when executed by at least one processor, causes the processor to perform steps comprising:

obtaining a web page with a label from a web server device in response to a request from a mobile computing device;

obtaining destination information for the label associated with the requesting mobile computing device;

converting the label into a portable data file; and

transmitting the portable data file using the destination information for the label associated with the requesting mobile computing device.

7. The medium as set forth in claim 6 further comprising: determining whether the web page with the label is optimized for printing; and

extracting and optimizing the label for printing when the determining indicates the web page and label are not optimized for printing.

8. The medium as set forth in claim 6 further comprising: determining whether the requesting mobile computing device has provided at least one of notes and a logo to customize the label; and customizing the label when the requesting mobile computing device has provided at least one of the notes and the logo.

9. The medium as set forth in claim 6 wherein the portable data file is a PDF file.

- 13 -

10. The medium as set forth in claim 6 wherein the obtained destination information comprises an email address.

5 11. A proxy server device comprising:
one or more processors;
a memory coupled to the one or more processors which are configured to execute programmed instructions stored in the memory comprising:
obtaining a web page with a label from a web server device in response to a request from a mobile computing device;
10 obtaining destination information for the label associated with the requesting mobile computing device;
converting the label into a portable data file; and
transmitting the portable data file using the destination information for the label associated with the requesting mobile
15 computing device.

12. The device as set forth in claim 11 wherein the one or more processors is further configured to execute programmed instructions stored in the memory further comprising:
20 determining whether the web page with the label is optimized for printing; and
extracting and optimizing the label for printing when the determining indicates the web page and label are not optimized for printing.

25 13. The device as set forth in claim 11 wherein the one or more processors is further configured to execute programmed instructions stored in the memory further comprising:
determining whether the requesting mobile computing device has provided at least one of notes and a logo to customize the label; and
30 customizing the label when the requesting mobile computing device has provided at least one of the notes and the logo.

- 14 -

14. The device as set forth in claim 11 wherein the portable data file is a PDF file.

15. The device as set forth in claim 11 wherein the obtained destination information comprises an email address.

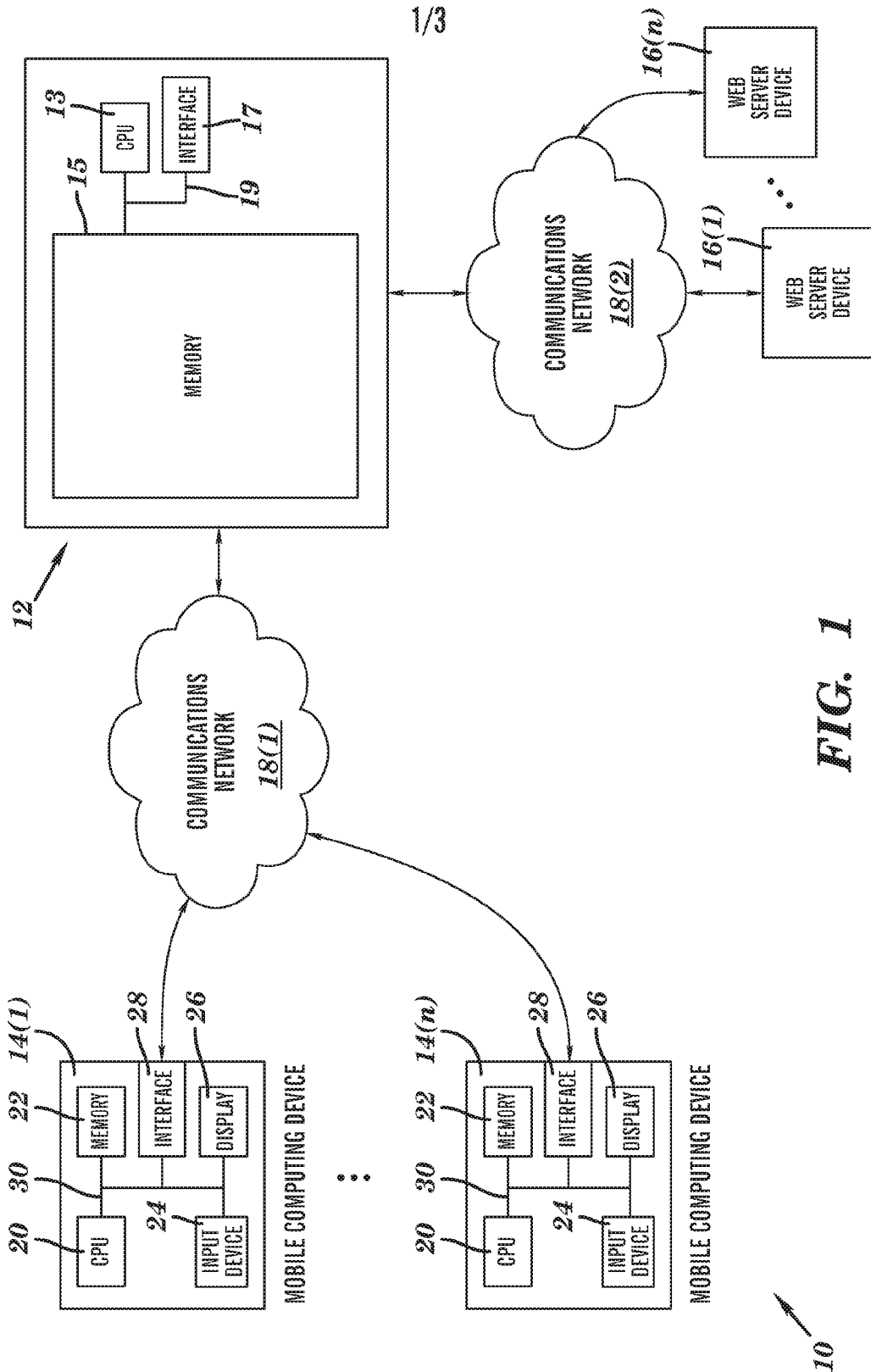


FIG. 1

Confirmation

Dear John Doe,
thank you for...

To receive the label as a pdf file please provide your email address.

*Email:

Notes:

Logo:

*Required

FIG. 2

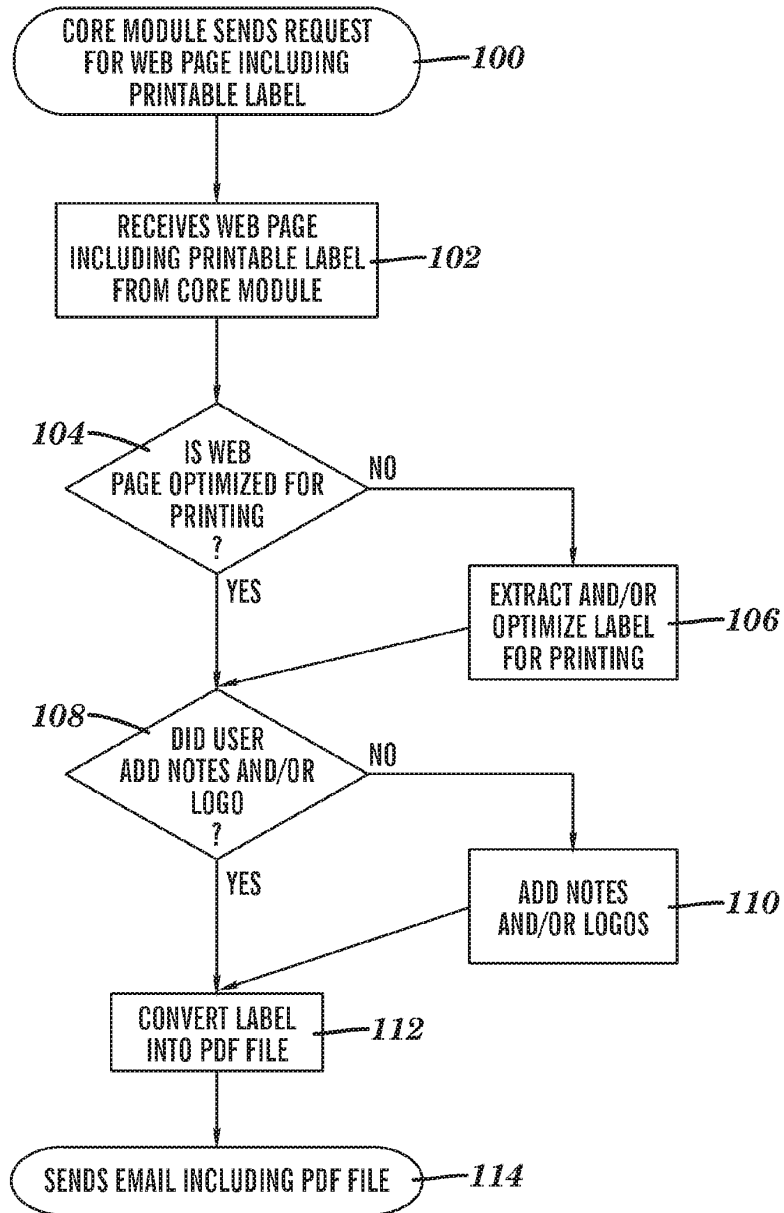




FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/066611

A. CLASSIFICATION OF SUBJECT MATTER		
G06Q 50/32(2012.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06Q 50/32; G06F 15/16; G06F 12/00; H04W 4/00; G06F 3/12; H04W 8/24		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: mobile, print, proxy server, label, convert, data format, portable data file, pdf		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2004-0029438 A (PEERLESS SYSTEMS CORPORATION) 6 APRIL 2004 See the abstract, page 2, line 9 - page 40, line 64, claims 1-72 and figures 1-60.	1-15
A	KR 10-2008-0095834 A (RESEARCH IN MOTION LIMITED) 29 OCTOBER 2008 See the abstract, page 4, paragraph [0002] - page 14, paragraph [0094], claims 1-22 and figures 1-9.	1-15
A	US 2002/0049777 A1 (TERAYAMA, YASUHIRO et al.) 25 APRIL 2002 See the abstract, page 1, paragraph [0002] - page 10, paragraph [0133], claims 1-28 and figures 1-25.	1-15
A	US 7,080,181 B2 (PALEYICH, JOHN H. et al.) 18 JULY 2006 See the abstract, column 1, line 14 - column 14, line 67, claims 1-25 and figures 1-7.	1-15
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 31 MAY 2012 (31.05.2012)		Date of mailing of the international search report 01 JUNE 2012 (01.06.2012)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer SON, HEE SOO Telephone No. 82-42-481-5960 

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2011/066611

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR 10-2004-0029438 A	06.04.2004	AU 2002-247362 A8	08.10.2002
		AU 2002-308678 A1	08.10.2003
		CN 1307565 C	28.03.2007
		CN 1537298 A	13.10.2004
		CN 1575458 A	02.02.2005
		CN 1575460 A	02.02.2005
		EP 1380194 A2	14.01.2004
		EP 1428129 A1	16.06.2004
		EP 1428134 A2	16.06.2004
		EP 1490829 A1	29.12.2004
		JP 04202272 B2	24.12.2008
		JP 2004-535618 A	25.11.2004
		JP 2005-501341 A	13.01.2005
		JP 2005-521166 A	14.07.2005
		JP 2005-523489 A	04.08.2005
		KR 10-2004-0039304 A	10.05.2004
		KR 10-2004-0058105 A	03.07.2004
		US 2002-0138564 A1	26.09.2002
		US 2003-0078965 A1	24.04.2003
		US 2003-0079030 A1	24.04.2003
		US 2003-0182378 A1	25.09.2003
		US 2006-0294251 A1	28.12.2006
		US 2007-0022180 A1	25.01.2007
		US 2007-0168514 A1	19.07.2007
		US 6993562 B2	31.01.2006
		US 8019829 B2	13.09.2011
		US 8024398 B2	20.09.2011
		US 8065357 B2	22.11.2011
		WO 2002-076175 A2	03.10.2002
		WO 2003-019389 A1	06.03.2003
		WO 2003-019403 A2	06.03.2003
		WO 2003-081524 A1	02.10.2003
KR 10-2008-0095834 A	29.10.2008	CA 2621343 A1	12.04.2007
		CA 2621343 C	20.04.2010
		EP 1929725 A1	11.06.2008
		EP 1929725 B1	30.06.2010
		WO 2007-040514 A1	12.04.2007
US 2002/0049777 A1	25.04.2002	CN 1314634 A	26.09.2001
		JP 2001-331362 A	30.11.2001
		KR 10-2001-0096592 A	07.11.2001
		TW 539942 A	01.07.2003
		TW 539942 B	01.07.2003
		US 7010551 B2	07.03.2006
US 7,080,131 B2	18.07.2006	US 2005-0144325 A1	30.06.2005
		US 2006-0212539 A1	21.09.2006
		US 6889256 B1	03.05.2005

Form PCT/ISA/210 (patent family annex) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2011/066611

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 7441012 B2	21. 10. 2008

Form PCT/ISA/210 (patent family annex) (July 2009)

(19)



(11)

EP 2 724 251 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
08.08.2018 Bulletin 2018/32

(51) Int Cl.:
G06F 17/00 (2006.01) **G06F 17/20** (2006.01)
G06F 15/16 (2006.01)

(21) Application number: **12802874.3**

(86) International application number:
PCT/US2012/044012

(22) Date of filing: **25.06.2012**

(87) International publication number:
WO 2012/178167 (27.12.2012 Gazette 2012/52)

(54) **METHODS FOR MAKING AJAXWEB APPLICATIONS BOOKMARKABLE AND CRAWLABLE AND DEVICES THEREOF**

VERFAHREN ZUR LESEZEICHENFÄHIGKEIT UND CRAWLFÄHIGKEIT VON AJAX-WEBANWENDUNGEN UND VORRICHTUNGEN DAFÜR

PROCÉDÉS PERMETTANT À DES APPLICATIONS WEB AJAX D'ÊTRE MISES EN SIGNETS ET D'ÊTRE PARCOURUES, ET DISPOSITIFS ASSOCIÉS

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(72) Inventors:
 • **SCODA, Enrico**
I-33100 Martignacco (UD) (IT)
 • **PEZZANO, Simone**
I-33100 Udine (IT)

(30) Priority: **24.06.2011 US 201113135112**

(43) Date of publication of application:
30.04.2014 Bulletin 2014/18

(74) Representative: **Gervasi, Gemma et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(73) Proprietor: **Usablenet Inc.**
New York, NY 10019 (US)

(56) References cited:
WO-A1-2011/035944 WO-A2-2009/076187
US-A1- 2009 282 052 US-A1- 2011 066 609

EP 2 724 251 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description**FIELD**

[0001] This technology generally relates to methods for processing web application content and, more particularly, to methods for making an asynchronous JavaScript and extensible mark-up language (AJAX) application content state bookmarkable and crawlable and devices thereof.

BACKGROUND

[0002] An asynchronous JavaScript and extensible mark-up language (AJAX) web application uses AJAX to communicate with a server to load data or page fragments. Programming with AJAX offers benefits in the context of web application development. For example, AJAX offers site developers an additional level of innovation that may make a site easier to use for all visitors. AJAX may also be utilized to give users the sense that a web application is more responsive than a traditional web site.

[0003] However, with the use of AJAX to manipulate web page content, an issue arises with regard to providing bookmarkability of content that may be dynamic rather than static during a user session. For example, while a web page may have a single non-changing uniform resource locator (URL) stored in the web browser, the content in many AJAX web sites is asynchronously changed and replaced. Accordingly, when a user sets a bookmark, he may only be able to capture the starting URL of the web site instead of the exact content state he is currently viewing. Users therefore will want a way to bookmark a dynamic web page in a way that allows them to restore a specific content state.

[0004] One common solution is to change the hash part of the URL, since doing so does not activate any actions, such as loading a new page on the browser side. However, this approach yields yet another issue as to how a web crawler or web robot can access that URL. Even if the hash part of the URL contains all of the information needed to fetch correct data using AJAX technology, the web crawler will not be able to access it since web crawlers are not able to execute JavaScript code inside the web pages they download. Examples of prior art methods are described in the following documents: PCT application WO2009076187A discloses a method performed by a server for maintaining state in a stateless server environment, comprising receiving a request for performance of an operation by an application from a client wherein the request includes a first universal resource locator (URL). A first set of state information may be extracted from the first URL and evaluated, a second set of state information associated with the operation is assembled, said second set of state information containing at least the client identification information, the application information, and a counter to keep track of how

many times the client has interacted with the server for the operation. PCT application WO2011035944 discloses a method for managing application state information, which is represented by a substring of a Uniform Resource Identifier (URI), which identifies a resource in a network. The substring comprises a plurality of nested data arrays. A URI parser and a URI generator convert the application state information between an object representation and a URI string representation. Preferably, the URI is a Uniform Resource Locator (URL), the application state information is provided by a web application, and the URI string representation uses a JavaScript Object Notation (JSON). An object of the present invention is how to bookmark dynamic web pages more effectively with respect to the methods known in the art, in order to allow users to restore the dynamic web pages to a specific content state.

SUMMARY

[0005] An exemplary method for making an asynchronous JavaScript and extensible mark-up language (AJAX) application content state bookmarkable and crawlable includes receiving at a web application processing server device a HyperText Transfer Protocol (HTTP) request including an HTTP request uniform resource locator (URL) representing a current application content state. When the HTTP request URL contains a first reference substring, a second substring based upon the first reference substring is extracted from the HTTP request URL with the web application processing server device. Based at least in part on the extracted second substring, a modified URL representing the current application content state is generated by the web application processing server device.

[0006] An exemplary computer-readable storage medium having stored thereon instructions for making an AJAX application content state bookmarkable and crawlable. The instructions include machine executable code which, when executed by at least one processor, causes the processor to perform steps including receiving a HTTP request including an HTTP request uniform resource locator (URL) representing a current application content state. When the HTTP request URL contains a first reference substring, a second substring based upon the first reference substring is extracted from the HTTP request URL. Based at least in part on the extracted second substring, a modified URL representing the current application content state is generated.

[0007] An exemplary web application processing server device including one or more processors and a memory coupled to the one or more processors which are configured to execute programmed instructions stored in the memory, the programmed instructions including receiving a HTTP request including an HTTP request uniform resource locator (URL) representing a current application content state. When the HTTP request URL contains a first reference substring, a second substring

based upon the first reference substring is extracted from the HTTP request URL. Based at least in part on the extracted second substring, a modified URL representing the current application content state is generated.

[0008] This technology provides effective and efficient methods and devices for creating a concise URL representing a specific AJAX web application page or content state, which thereby allows generic web crawlers or robots to index the page or content state thus represented.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]

FIG. 1 is a block diagram of a network environment with an exemplary web application processing server device; and

FIG. 2 is a flowchart of an exemplary method for making an AJAX application content state bookmarkable and crawlable.

DETAILED DESCRIPTION

[0010] A network environment 10 with an exemplary web application processing server device 12 is illustrated in FIG. 1. The environment 10 may include the web application processing server device 12, client computing devices 14(1)-14(n), web crawling devices 40(1)-40(n), a web server device 16, and communication networks 18(1)-18(2), although other numbers and types of systems, devices, servers, and/or elements in other configurations can be used. This technology provides effective and efficient methods and devices for creating a concise URL representing a specific AJAX web application page or content state, which thereby allows generic web crawlers or robots to index the page or content state thus represented.

[0011] This technology is directed to methods, devices, and computer-readable storage media for making an asynchronous JavaScript and extensible mark-up language (AJAX) application content state bookmarkable and crawlable. In this example, the web application processing server device 12 receives a HyperText Transfer Protocol (HTTP) request including an HTTP request uniform resource locator (URL) representing a current application content state; extracts, from the HTTP request URL, a second substring based upon a first reference substring, when the HTTP request URL contains the first reference substring; and generates a modified URL based at least in part on the extracted second substring, wherein the modified URL represents the current application content state. Although one web application processing server device 12 is shown, other numbers and types of web application processing server systems can be used.

[0012] In some examples, the web application processing server device 12 may be a web content proxy

server, or any other type of proxy server. For example, the web application processing server device 12 may manage the handling of redirect messages or redirection responses from the one or more web server devices 16 for and/or on behalf of requesting client devices 14(1)-14(n) and/or requesting web crawling devices 40(1)-40(n), although the web application processing server device 12 may provide other numbers and types of functions.

[0013] In other examples, the web application processing server device 12 may itself be a web server device. For example, the web application processing server device 12 may be contained in or identical to web server device 16. In such examples, the web application processing server device 12 may host one or more AJAX web applications and store associated web content in memory 15.

[0014] Referring more specifically to FIG. 1, the web application processing server device 12 may include a central processing unit (CPU) or processor 13, a memory 15, and an interface system 17 which are coupled together by a bus 19 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used.

The processor 13 in the web application processing server device 12 may execute a program of stored instructions one or more aspects of the present technology as described and illustrated by way of the examples herein, including methods for making an asynchronous AJAX application content state bookmarkable and crawlable, although the processor could execute other numbers and types of programmed instructions.

[0015] The memory 15 in the web application processing server device 12 may store these programmed instructions for one or more aspects of the present technology as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 13, may be used for the memory 15 in the web application processing server device 12. In some of these examples, the memory 15 may include a core module 21, a URL construction module 23, and a memory cache 25 which may store programmed instructions for one or more aspects of the present technology as described and illustrated herein, although the memory may comprise other types and numbers of systems, devices, and elements in other configurations which may store other data.

[0016] The interface device 17 in the web application processing server device 12 may be used to operatively couple and communicate between the web application processing server device 12 and the client computing devices 14(1)-14(n), web crawling devices 40(1)-40(n),

and the web server device 16 via the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. By way of example only, the communication networks 18(1) and 18(2) may use TCP/IP over Ethernet and industry-standard protocols, including HTTP, HTTPS, WAP, and SOAP, although other types and numbers of communication networks, such as a direct connection, a local area network, a wide area network, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, may be used.

[0017] Each of the client computing devices 14(1)-14(n) may enable a user to access content and utilize one or more applications from the web server device 16 through the web application processing server device 12 through one or more communication networks, although one or more of the client computing devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. Although multiple client computing devices 14(1)-14(n) are shown, other numbers and types of user computing systems could be used. In this example, the client computing devices 14(1)-14(n) comprise devices with Internet access that permit a website page or other retrieved data to be displayed. By way of example only, one or more of the client computing devices 14(1)-14(n) may comprise smart phones, personal digital assistants, or desktop computers using visual or voice browsers.

[0018] Each of client computing devices 14(1)-14(n) may include a central processing unit (CPU) or processor 20, a memory 22, user input device 24, a display 26, and an interface system 28, and which are coupled together by a bus 30 or other link, although one or more of client computing devices 14(1)-14(n) may include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 20 in each of client computing devices 14(1)-14(n) may execute a program of stored instructions for one or more aspects of the present technology as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0019] The memory 22 in each of the client computing devices 14(1)-14(n) may store these programmed instructions for one or more aspects of the present technology as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 20 may be used for the memory 22 in each of the client computing devices 14(1)-14(n).

[0020] The user input device 24 in each of the client computing devices 14(1)-14(n) may be used to input selections, such as requests for a particular website page, although the user input device could be used to input other types of data and interact with other elements. The user input device may include keypads, touch screens, and/or vocal input processing systems although other types and numbers of user input devices can be used.

[0021] The display 26 in each of the client computing devices 14(1)-14(n) may be used to show data and information to the user, such as a website page by way of example only. The display in each of the client computing devices 14(1)-14(n) may be a phone screen display, although other types and numbers of displays could be used.

[0022] The interface system 28 in each of the client computing devices 14(1)-14(n) may be used to operatively couple and communicate between the client computing devices 14(1)-14(n) and the web application processing server device 12 and web server device 16 over the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations may be used.

[0023] The web server device 16 may provide one or more web software applications for use by one or more of the client computing devices 14(1)-14(n), although the web server device 16 can provide other numbers and types of applications and/or content and can have provide other numbers and types of functions. Although one web server device 16 is shown for ease of illustration and discussion, other numbers and types of web server systems and devices can be used.

[0024] The web server device 16 may include a central processing unit (CPU) or processor, a memory, and an interface system which are coupled together by a bus or other link, although the web server device 16 may have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations may be used. The processor in the web server device 16 may execute a program of stored instructions for one or more aspects of the present technology as described and illustrated by way of the examples herein, including managing application functionality, although the processor could execute other numbers and types of programmed instructions.

[0025] The memory in the web server device 16 may store these programmed instructions for one or more aspects of the present technology as described and illustrated by way of the examples herein including managing application functionality, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system

that is coupled to the processor, may be used for the memory in each of the web server device 16.

[0026] The interface system in the web server device 16 may be used to operatively couple and communicate between the web server device 16 and the web application processing server device 12, the client computing devices 14(1)-14(n), and web crawling devices 40(1)-40(n) via communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0027] Although examples of the web application processing server device 12, the client computing devices 14(1)-14(n), web crawling devices 40(1)-40(n), and the web server device 16, are described and illustrated herein, each of the client computing devices 14(1)-14(n), web crawling devices 40(1)-40(n), the web application processing server device 12, and the web server device 16, may be implemented on any suitable computer system or computing device. It is to be understood that the devices and systems of the examples described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the examples are possible, as will be appreciated by those skilled in the relevant art(s).

[0028] Furthermore, each of the systems of the examples may be conveniently implemented using one or more general purpose computer systems, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the examples, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0029] In addition, two or more computing systems or devices can be substituted for any one of the systems in any embodiment of the examples. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the examples. The examples may also be implemented on computer system or systems that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0030] The examples may also be embodied as a computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0031] Referring to FIGS. 1-2, an exemplary method

200 for making an AJAX application content state bookmarkable and crawlable will now be described. In step 210, the web application processing server device 12 receives an HTTP request which includes an HTTP request URL representing a current application content state. In other examples, such as where the HTTP request originated from a client computing device 14, the HTTP request URL may correspond to a bookmark.

[0032] In step 220, the web application processing server device 12 determines whether the HTTP request contains a first reference substring. If in step 220 the HTTP request URL contains the first reference substring in step 220, then the Yes branch is taken to step 230.

[0033] In step 230, the web application processing server device 12 extracts a second substring from the HTTP request URL based upon the first reference substring. In some examples, the second substring may follow the first reference substring. In some of these examples, the second substring may include all of the text immediately following the first reference substring.

[0034] In step 240, the web application processing server device 12 generates a modified URL representing the current application content state based at least in part on the extracted second substring. The modified URL may, for example, be generated by the URL construction module 23 of the web application processing server device 12.

[0035] In step 242, the web application processing server device 12 determines if the source of the HTTP request is one of the web crawling device 40(1)-40(n) based upon a header in the HTTP request, although other manners for determining an identity of the source can be used. If in step 242 the web application processing server device 12 determines the source of the HTTP request is not one of the web crawling device 40(1)-40(n), then the No branch is taken to step 244. In step 244, the web application processing server device 12 provides the modified URL to the appropriate web server device 16. Based on the modified URL, the web server device 16 generates a response to the HTTP request which is then communicated to the source of the request (e.g., one of the client computing devices 14(1)-14(n)), and then the method ends. In some embodiments, the response generated by the web server device 16 may include the current application content state corresponding to the modified URL.

[0036] If in step 242 the web application processing server device 12 determines the source of the HTTP request is one of the web crawling devices 40(1)-40(n), then the Yes branch is taken to step 246. In step 246, the web application processing server device 12 returns the modified URL as an HTTP redirect response to the identified one of the one of the web crawling device 40(1)-40(n). The web application processing server device 12 also may store the default URL at a location (e.g., in memory 15) associated with the HTTP redirect response. After step 246, this method ends.

[0037] If back in step 220, the received HTTP request

URL does not contain the first reference substring in step 220, then the No branch is taken to step 222.

[0038] In step 222, the web application processing server device 12 determines based upon a header in the HTTP request if the source of the HTTP request is one of the web crawling devices 40(1)-40(n), although other manners for determining the source can be used. If in step 222 the web application processing server device 12 determines the source of the HTTP request is not one of the web crawling devices 40(1)-40(n), then the No branch is taken to step 224. In step 224, the web application processing server device 12 provides a default URL to the appropriate web server device 16. Based on the default URL, the web server device 16 generates a response to the HTTP request which is then communicated to the requesting one of the client computing devices 14(1)-14(n), and then the method ends, although other types of devices and systems could be the source of the request. In some embodiments, the response generated by the web server device 16 may include the current application content state corresponding to the default URL.

[0039] If in step 222 the web application processing server device 12 determines the source of the HTTP request is one of the web crawling devices 40(1)-40(n), then the Yes branch is taken to step 226. In step 226, the web application processing server device 12 returns the default URL as an HTTP redirect response to the identified one of the web crawling devices 40(1)-40(n) that was the source of the HTTP request. The web application processing server device 12 also may store the default URL at a location (e.g., in memory 15) associated with the HTTP redirect response. After step 226 this method ends.

[0040] In an embodiment utilizing the exemplary method 200, a web application located at www.example.com/index.html may load the fragment [welcome.html](#) when starting up. This web application may also load product description pages identified by such links as [/products/pdp.jsp?id=2234](#). The exemplary method 200 also may be utilized to address the problem of creating a URL representing an AJAX web application page or content state. One concept associated with exemplary method 200 is that relevant web pages (e.g., pages whose URLs will likely be saved or shared) may be a small portion of the entire web application. Moreover, such relevant web pages may often be identified by a small set of identifiers corresponding to keys in catalog databases or specific events (e.g., dates, flight numbers, etc.). For example, the URL [/products/pdp.jsp?id=2234](#) may be represented by the identifier "2234". Therefore, in some examples, the AJAX web application JavaScript code may contain a function that creates a mapping between such an identifier and its corresponding URL. In some of these examples, a standard filename may be established for the portion of the JavaScript code containing this mapping function. The resulting JavaScript file containing this function may be limited to this mapping

function and its dependencies (e.g., data structures or utility functions). Thus, a simple implementation of the JavaScript interpreter may be hosted on the web application processing server device 12, and the mapping function may be executed to obtain the URL being used by the web application AJAX engine. For example, in step 240, URL construction module 23 of web application processing server device 12 may, upon receiving such an identifier or second substring as an input (e.g., "2234"), generate the corresponding modified or processed URL associated with that identifier.

[0041] In another embodiment utilizing the exemplary method 200, an HTTP request URL may contain the first reference substring ["/shareCode-"](#) in step 220. In some of these examples, the substring following ["/shareCode-"](#) may be extracted in step 230 and passed along to the URL construction module 23 of the web application processing server device 12. In step 240, URL construction module 23 may then map this extracted second substring or identifier to a corresponding modified URL representing the current application content state. If, on the other hand, the HTTP request URL does not contain the first reference substring ["/shareCode-"](#) in step 220, then a default URL (e.g., a default startup URL) may be provided to the source of the HTTP request, as described in step 224 or step 226.

[0042] For example, consider an embodiment in which the HTTP request URL contains the path ["/shareCode-2234"](#). In step 230, the web application processing server device 12 extracts the identifier or second substring "2234" and passes it to the URL construction module 23, which returns the URL ["/products/pdp.jsp?id=2234"](#) in step 240. If, on the other hand, the URL path is equal to ["/index.html"](#), a default URL (e.g., ["/welcome.html"](#)) may be returned in step 224 or step 226.

[0043] Beneficially, this approach results in short and concise URLs representing web application pages. Moreover, relevant data is in the URL path component such that it is universally recognized as a URL portion to keep. (Some web crawler devices, for example, will remove the hash portion of the URL when indexing.) Additionally, since only key information from the URL used by the AJAX call is used in the URL representing the web application content state, any marginal changes to the AJAX call URL will not invalidate the web application URL being saved, shared, or indexed.

[0044] In another embodiment utilizing exemplary method 200, one of the web crawling devices 40(1)-40(n) may send an HTTP request to the web application processing server device 12. The web application processing server device 12 may recognize that the HTTP request originated from one of the web crawling devices 40(1)-40(n) by analyzing a header (e.g., the User-Agent HTTP request header) in the HTTP request. In some examples, if the header value is included in a list of supported or recognized web crawling devices, exemplary method 200 will be executed.

[0045] In some of these examples, if, in step 220, the

HTTP request URL does not contain the first reference substring (e.g., "/shareCode-"), the web application processing server device 12 may provide to the one of the web crawling devices 40(1)-40(n) an HTTP redirect response containing the web application main HTML file (or any other suitable default URL) as content in step 226. [0046] Otherwise, if the HTTP request URL does contain the first reference substring (e.g., "/shareCode-") in step 220, then, in some examples, the substring following "/shareCode-" may be extracted in step 230 and passed to the URL construction module 23. In steps 240, 242, and 246, the URL construction module 23 may generate and return the corresponding modified URL as an HTTP redirect response to one of the web crawling devices 40(1)-40(n). In this way, the one of the web crawling devices 40(1)-40(n) may access the URL of the HTML page containing the relevant information to be indexed.

[0047] Beneficially, this approach allows any of the web crawling devices 40(1)-40(n) to access AJAX web application pages or content states without a need to execute JavaScript code to emulate the AJAX web application behavior. Additionally, there is no need to add new standards or protocols to instruct web server devices and/or web crawling devices how to share information about AJAX web application pages.

[0048] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby as well. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for making an asynchronous JavaScript and extensible mark-up language, AJAX, application content state bookmarkable and crawlable, the method comprising:

receiving by a web application processing server device (12) a HyperText Transfer Protocol, HTTP, request comprising an HTTP request uniform resource locator, URL, representing a current application content state;
determining, by the web application processing server device (12), when the HTTP request URL comprises a first reference substring and when the determining indicates that the HTTP request

URL comprises the first reference substring:

extracting, by the web application processing server device (12), a second substring from the HTTP request URL based upon the first reference substring; mapping by the web application processing server device the extracted second substring to a modified URL representing the current application content state; and
returning, by the web application processing server device (12), the modified URL and determining when the HTTP request is from a web crawler, wherein the modified URL is returned as an HTTP redirect response when the determining indicates that the HTTP request is from the web crawler.

2. The method according to claim 1 further comprising: providing by the web application processing server device (12) a default URL when the HTTP request URL does not contain the first reference substring.
3. The method according to claim 1, wherein the second substring follows the first reference substring.
4. The method according to claim 1, wherein the HTTP request URL corresponds to a bookmark.
5. The method according to claim 1 further comprising: identifying, by the web application processing server device, when the HTTP request is from a web crawler based upon a header in the HTTP request.
6. The method according to claim 1 further comprising: storing by the web application processing server device (12) the modified URL at a location associated with the HTTP redirect response.
7. A computer readable medium having stored thereon instructions for making an asynchronous JavaScript and extensible mark-up language, AJAX, application content state bookmarkable or crawlable comprising machine executable code which when executed by at least one processor, causes the processor to perform the steps of the method according to any of claims from 1 to 6.
8. A web application processing server device (12) for making an asynchronous JavaScript and extensible mark-up language, AJAX, application content state bookmarkable and crawlable, the device comprising:
 - one or more processors (13); and
 - a memory (15) coupled to the one or more processors (13) which are configured to execute programmed instructions stored in the memory (15), the programmed instructions comprising

the steps of the method according to any of claims from 1 to 6.

Patentansprüche

1. Verfahren, um einen Content-Zustand einer Anwendung mit einem asynchronen JavaScript und mit einer erweiterbaren Markierungssprache, AJAX-Anwendung, lesezeichenfähig und für Suchmaschinen durchsuchbar zu machen, wobei das Verfahren umfasst, dass:

von einer Webanwendungsverarbeitungs-Servervorrichtung (12) eine Hypertexttransferprotokoll-Anfrage, HTTP-Anfrage, empfangen wird, welche einen Uniform Resource Locator, URL, der HTTP-Anfrage umfasst, der einen aktuellen Content-Zustand der Anwendung repräsentiert;

von der Webanwendungsverarbeitungs-Servervorrichtung (12) ermittelt wird, ob der URL der HTTP-Anfrage einen ersten Referenz-Teilstring umfasst, und wenn das Ermitteln anzeigt, dass der URL der HTTP-Anfrage den ersten Referenz-Teilstring umfasst:

von der Webanwendungsverarbeitungs-Servervorrichtung (12) ein zweiter Teilstring aus dem URL der HTTP-Anfrage auf der Grundlage des ersten Referenz-Teilstrings extrahiert wird;

von der Webanwendungsverarbeitungs-Servervorrichtung der extrahierte zweite Teilstring einem modifizierten URL zugeordnet wird, der den aktuellen Content-Zustand der Anwendung repräsentiert; und von der Webanwendungsverarbeitungs-Servervorrichtung (12) der modifizierte URL zurückgegeben wird und ermittelt wird, ob die HTTP-Anfrage von einem Web-Crawler stammt, wobei der modifizierte URL als HTTP-Umleitungsantwort zurückgegeben wird, wenn das Ermitteln anzeigt, dass die HTTP-Anfrage von dem Web-Crawler stammt.

2. Verfahren nach Anspruch 1, das ferner umfasst, dass:
von der Webanwendungsverarbeitungs-Servervorrichtung (12) ein Standard-URL bereitgestellt wird, wenn der URL der HTTP-Anfrage den ersten Referenz-Teilstring nicht enthält.
3. Verfahren nach Anspruch 1, wobei der zweite Teilstring auf den ersten Referenz-Teilstring folgt.

4. Verfahren nach Anspruch 1, wobei der URL der HTTP-Anfrage einem Lesezeichen entspricht.

5. Verfahren nach Anspruch 1, das ferner umfasst, dass:
von der Webanwendungsverarbeitungs-Servervorrichtung auf der Grundlage eines Headers in der HTTP-Anfrage erkannt wird, ob die HTTP-Anfrage von einem Web-Crawler stammt.

6. Verfahren nach Anspruch 1, das ferner umfasst, dass:
von der Webanwendungsverarbeitungs-Servervorrichtung (12) der modifizierte URL an einer Stelle gespeichert wird, die der HTTP-Umleitungsantwort zugeordnet ist.

7. Computerlesbares Medium, in dem Anweisungen gespeichert sind, um den Content-Zustand einer Anwendung mit einem asynchronen JavaScript und mit einer erweiterbaren Markierungssprache, AJAX-Anwendung, lesezeichenfähig und für Suchmaschinen durchsuchbar zu machen, das maschinenausführbaren Code umfasst, welcher, wenn er von mindestens einem Prozessor ausgeführt wird, veranlasst, dass der Prozessor die Schritte des Verfahrens nach einem der Ansprüche 1 bis 6 ausführt.

8. Webanwendungsverarbeitungs-Servervorrichtung (12), um den Content-Zustand einer Anwendung mit einem asynchronen JavaScript und mit einer erweiterbaren Markierungssprache, AJAX-Anwendung, lesezeichenfähig und für Suchmaschinen durchsuchbar zu machen, wobei die Vorrichtung umfasst:

einen oder mehrere Prozessoren (13); und einen Speicher (15), der mit dem einen oder den mehreren Prozessoren (13) gekoppelt ist, welche ausgestaltet sind, um programmierte Anweisungen auszuführen, die in dem Speicher (15) gespeichert sind, wobei die programmierten Anweisungen die Schritte des Verfahrens nach einem der Ansprüche 1 bis 6 umfassen.

Revendications

1. Procédé de réalisation d'un état de contenu d'application de langage JavaScript et de balisage extensible asynchrones, AJAX, pouvant être mis en signets et pouvant être parcouru, le procédé comprenant les étapes consistant à :

recevoir, via un dispositif serveur de traitement d'application Web, une demande de protocole de transfert hypertexte, HTTP, comprenant un localisateur uniforme de ressources, URL, de

demande HTTP, représentant un état de contenu d'application en cours ;
détecter, via le dispositif serveur de traitement d'application web (12), lorsque l'URL de demande HTTP comprend une première sous-chaîne de référence, et lorsque la détermination indique que l'URL de demande HTTP comprend la première sous-chaîne de référence :

extraire, via le dispositif serveur de traitement d'application web (12), une seconde sous-chaîne à partir de l'URL de requête HTTP sur la base de la première sous-chaîne de référence ;

mapper, via le dispositif serveur de traitement d'application web, la seconde sous-chaîne extraite à un URL modifié représentant l'état de contenu d'application en cours ; et

renvoyer, via le dispositif serveur de traitement d'application web (12), l'URL modifié et déterminer quand la demande HTTP provient d'un robot d'exploration Web, dans lequel l'URL modifié est renvoyé comme réponse de redirection HTTP lorsque la détermination indique que la demande HTTP provient du robot d'exploration Web.

2. Procédé selon la revendication 1, comprenant en outre :

La fourniture, via le dispositif serveur de traitement d'application web (12), d'un URL par défaut lorsque l'URL de demande HTTP ne contient pas la première sous-chaîne de référence.

3. Procédé selon la revendication 1, dans lequel la seconde sous-chaîne suit la première sous-chaîne de référence.

4. Procédé selon la revendication 1, dans lequel l'URL de demande HTTP correspond à un signet.

5. Procédé selon la revendication 1, comprenant en outre l'étape consistant à :

identifier, via le dispositif serveur de traitement d'application Web, lorsque la demande HTTP provient d'un robot d'exploration Web sur la base d'un en-tête dans la requête HTTP.

6. Procédé selon la revendication 1, comprenant en outre :

le stockage, via le dispositif serveur de traitement d'application web (12), de l'URL modifié à un emplacement associé à la réponse de redirection HTTP.

7. Support lisible par ordinateur sur lequel sont stockées des instructions pour réaliser un état de contenu d'application d'un langage JavaScript et de bai-

sage extensible asynchrones, AJAX, pouvant être mis en signet ou rami fiable comprenant un code exécutable qui, exécuté par au moins un processeur, provoque les étapes du procédé selon l'une quelconque des revendications 1 à 6.

8. Dispositif serveur de traitement d'application web (12) pour faire un langage JavaScript asynchrone et de balisage extensible, AJAX, état de contenu d'application pouvant être mis en signet et pouvant être parcouru, le dispositif comprenant :

un ou plusieurs processeurs (13) ; et
une mémoire (15) couplée aux un ou plusieurs processeurs (13) qui sont configurés pour exécuter des instructions programmées stockées dans la mémoire (15), les instructions programmées comprenant les étapes du procédé selon l'une quelconque des revendications 1 à 6.

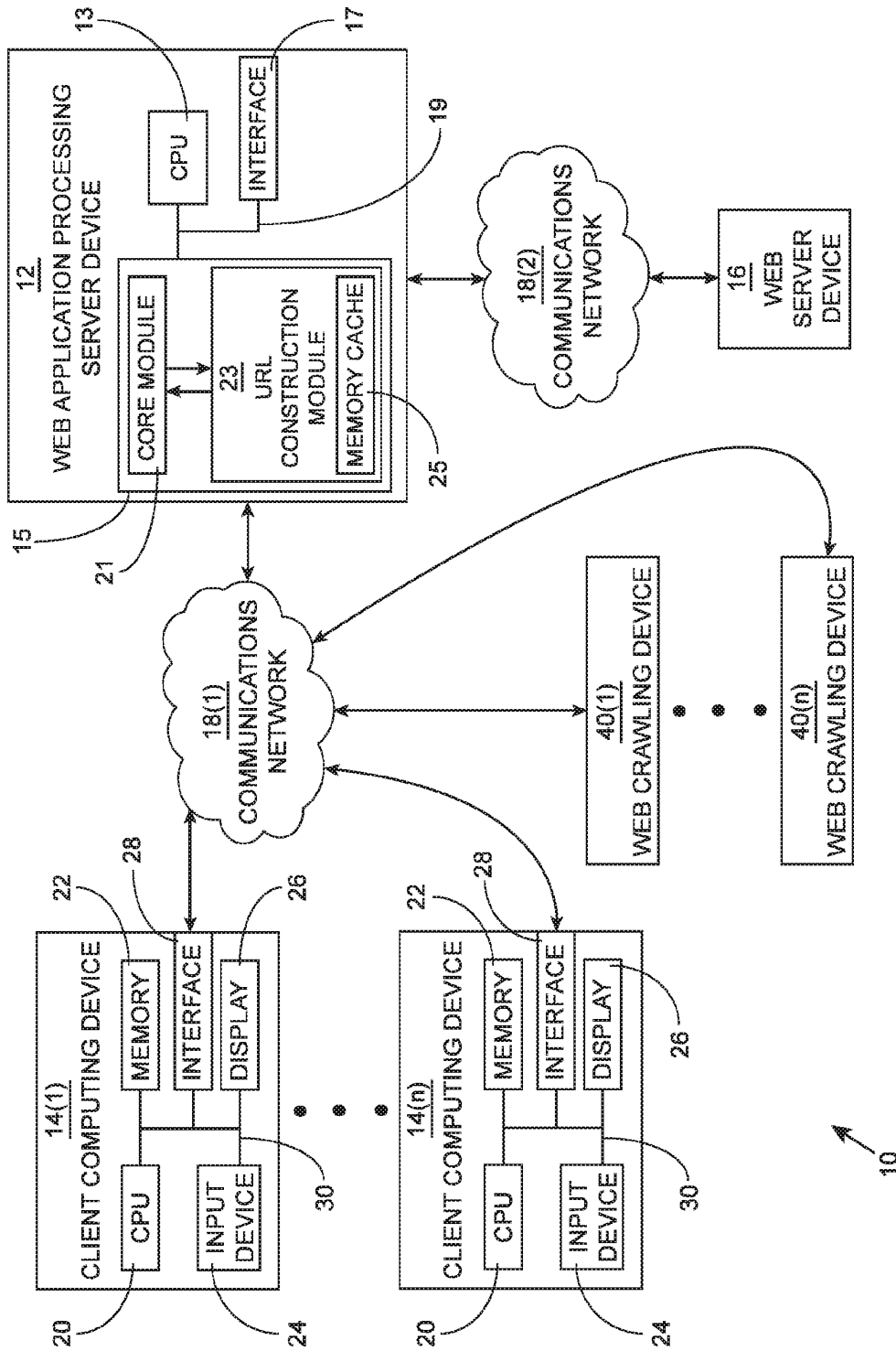


FIG 1

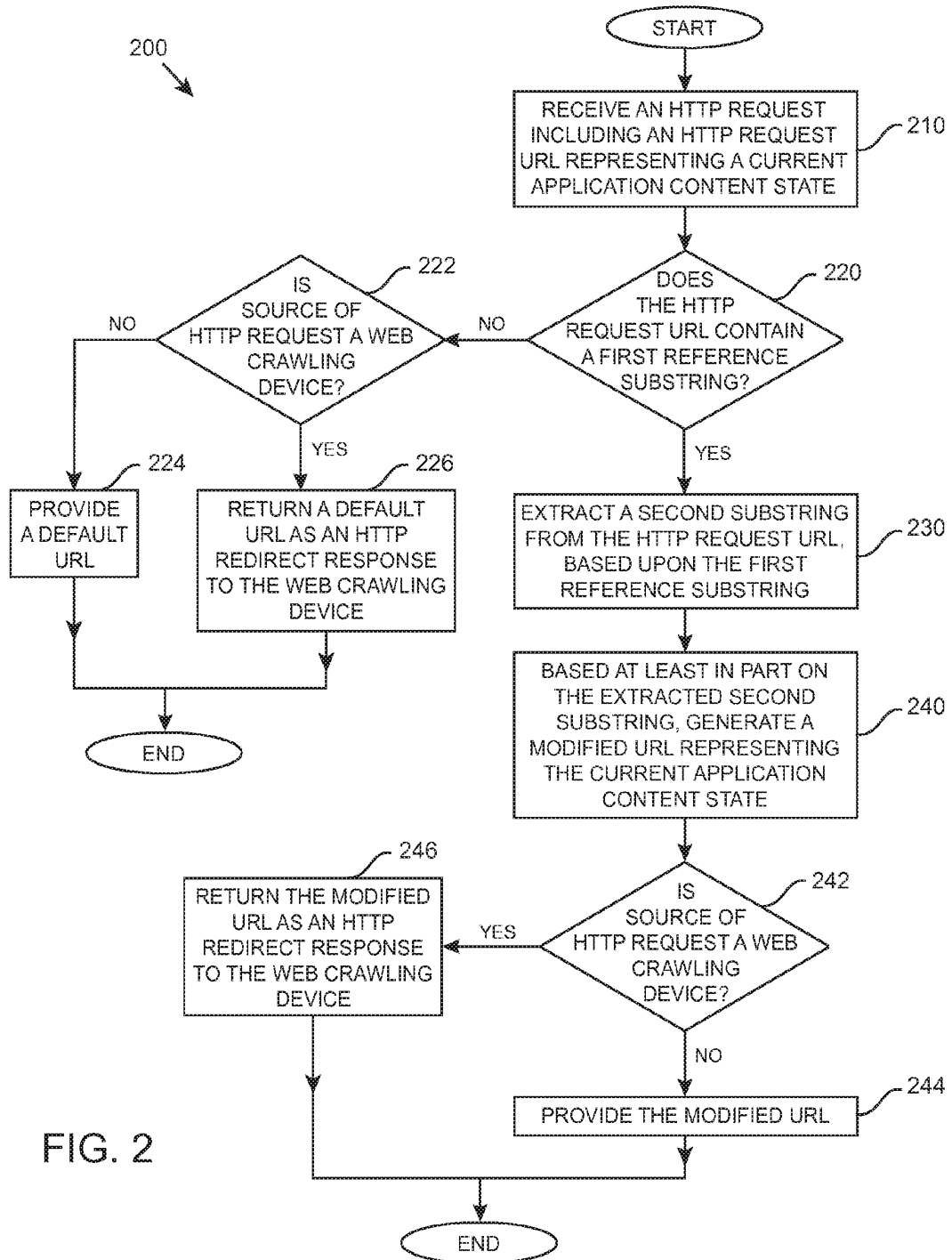


FIG. 2

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 2009076187 A [0004]
- WO 2011035944 A [0004]

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 2 724 253 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2013/003455 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2013/003455 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2013/003455 (art. 153(3) CBE).

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau



(10) International Publication Number
WO 2013/003455 A2

(43) International Publication Date
3 January 2013 (03.01.2013)

- (51) International Patent Classification: Not classified
- (21) International Application Number: PCT/US2012/044395
- (22) International Filing Date: 27 June 2012 (27.06.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 61/501,391 27 June 2011 (27.06.2011) US
- (71) Applicant (for all designated States except US): US-ABLENET INC. [US/US]; 28 W. 23rd St., 6th Floor, New York, NY 10010 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): SCODA, Enrico [IT/IT]; Via Cividina 416/3, I-33035 Martignacco (ud) (IT). BRONDANI, Marco [IT/IT]; Via Paniae 24, 33010 Colloredo Di Monte Aibano, Udine (IT).
- (74) Agents: GOLDMAN, Michael, L. et al.; Leclairryan, A Professional Corporation, 70 Linden Oaks, Suite 210, Rochester, NY 14625 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:
— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

WO 2013/003455 A2

(54) Title: METHODS FOR IMPLEMENTING WEB SERVICES AND DEVICES THEREOF

(57) Abstract: A method, non-transitory computer readable medium, and apparatus that determines when a JavaScript file is associated with a received web service request. A JavaScript environment is emulated when the determining indicates the JavaScript file is associated with the received web service request. A JavaScript object including one or more parameters of the received web service request is created. The JavaScript file associated with the received web service request is executed in the emulated JavaScript environment using the the JavaScript object. A web service response is constructed based on the JavaScript object as populated based on the executing. The constructed web service response is provided to a source of the received web service request.

- 1 -

METHODS FOR IMPLEMENTING WEB SERVICES AND DEVICES THEREOF

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent
5 Application Serial No. 61/501,391, filed June 27, 2011, which is hereby
incorporated by reference in its entirety.

BACKGROUND

[0002] Web services provide a standardized way of integrating web-based
applications using eXtensible Markup Language (XML), SOAP, Web Services
10 Description Language (WSDL), and/or Universal Description Discovery and
Integration (UDDI) standards over an Internet Protocol (IP) backbone. XML can
be used to tag data used by a web service, SOAP can be used to transfer the data,
WSDL can be used for describing the web services available and UDDI can be
used for listing the available web services. Web services allow different
15 applications located at different sources to communicate with each other
efficiently and without custom coding which can require a significant amount of
resources. Additionally, because communications are in XML, web services are
not tied to any operating system or programming language.

[0003] Unlike traditional client/server models, web services do not
20 provide an end user with a graphical user interface (GUI). Instead, web services
share data and processes through an application interface across a network. These
application interfaces are invoked and used to interpret any resulting data.

[0004] With the growing market for applications, web services are
increasingly popular since it is easy to integrate them into applications to extend
25 the features offered to the end user. Unfortunately, there are limitations on the
ability of mobile computing devices to effectively utilize web services integrated
into such applications including, for example, with respect to the execution of
functionality implemented using JavaScript code.

- 2 -

SUMMARY

[0005] A method for implementing a web service includes determining with a web content proxy server when a JavaScript file is associated with a received web service request. A JavaScript environment is emulated with the web content proxy server when the determining indicates the JavaScript file is associated with the received web service request. A JavaScript object including one or more parameters of the received web service request is created with the web content proxy server. The JavaScript file associated with the received web service request is executed with the web content proxy server in the emulated JavaScript environment using the the JavaScript object. The emulated JavaScript environment is capable of facilitating execution of web transactions including one or more pages. A web service response is constructed with the web content proxy server based on the JavaScript object as populated based on the executing. The constructed web service response is provided with the web content proxy server to a source of the received web service request.

[0006] A non-transitory computer readable medium having stored thereon instructions for implementing a web service comprising machine executable code which when executed by at least one processor, causes the processor to perform steps including determining whether a JavaScript file is associated with a received web service request. A JavaScript environment is emulated when it is determined that there is a JavaScript file associated with the received web service request. A JavaScript object including one or more parameters of the received web service request is created. The JavaScript file associated with the received web service request is executed in the emulated JavaScript environment using the the JavaScript object. The emulated JavaScript environment is capable of facilitating execution of web transactions including one or more pages. A web service response is constructed based on the JavaScript object as populated based on the executing. The constructed web service response is provided to a source of the received web service request.

[0007] A web content proxy server includes one or more processors and a memory coupled to the one or more processors which are configured to execute

- 3 -

programmed instructions stored in the memory including determining whether a JavaScript file is associated with a received web service request. A JavaScript environment is emulated when it is determined that there is a JavaScript file associated with the received web service request. A JavaScript object including one or more parameters of the received web service request is created. The JavaScript file associated with the received web service request is executed in the emulated JavaScript environment using the the JavaScript object. The emulated JavaScript environment is capable of facilitating execution of web transactions including one or more pages. A web service response is constructed based on the JavaScript object as populated based on the executing. The constructed web service response is provided to a source of the received web service request.

[0008] This technology provides a number of advantages including providing methods and devices for more easily implementing web services with a web content proxy server executing a web service automator module and JavaScript environment emulator. With this technology, web services can be more effectively extended and utilized by mobile computing devices that may not otherwise be capable of exploiting such functionality.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 is a block diagram of an environment with an exemplary web content proxy server with a web content optimization module, web service automator module, and a JavaScript environment emulator;

[0010] FIG. 2 is an exemplary login page, such as may be located at a *.login.html web address, and its hypertext markup language (HTML) source code;

25 [0011] FIG. 3 is an exemplary successful login page including a welcome message and a list of links to access different functionalities and logout and its HTML source code;

[0012] FIG. 4 is a flow chart of an exemplary method for implementing web services; and

- 4 -

[0013] FIG. 5 is exemplary JavaScript code for implementing a function facilitating generation of a web service response.

DETAILED DESCRIPTION

[0014] An exemplary environment 10 with client devices 12(1)-12(n),
5 server devices 14(1)-14(n), and web content proxy server 16, including a web
service automator module 18, a JavaScript environment emulator 20, and a web
content optimization module 22, coupled to communication networks 24(1)-24(2)
is illustrated in FIG. 1. Other numbers and types of systems, devices, and/or
elements in other configurations and environments with other communication
10 network topologies can also be used. By way of example only, the web service
automator module 18, JavaScript environment emulator 20, and/or web content
optimization module 22 can be implemented as an executable module of
programmed instructions for one or more of the methods described herein which
is stored in a memory 26 and executed by one or more processors or CPUs 28 in
15 the web content proxy server 16. This technology provides a number of
advantages including providing methods, non-transitory computer readable
medium, and devices for more easily and effectively implementing web services
on behalf of mobile computing devices.

[0015] Referring more specifically to FIG. 1, the web content proxy server
20 16 includes a central processing unit (CPU) 28 or processor, a memory 26, and an
interface system 30 which are coupled together by a bus 32 or other link, although
other numbers and types of components, parts, devices, systems, and elements in
other configurations and locations can be used. The processor 28 in the web
content proxy server 16 executes a program of stored instructions one or more
25 aspects of the present invention as described and illustrated by way of the
embodiments herein, although the processor 28 could execute other numbers and
types of programmed instructions.

[0016] The memory 26 in the web content proxy server 16 stores these
programmed instructions for one or more aspects of the present invention as
30 described and illustrated herein, although some or all of the programmed

- 5 -

instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or
5 written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 28, can be used for the memory 26 in the web content proxy server 16. In some embodiments, the memory 26 includes a web service automator module 18, a JavaScript environment emulator 20, and a web content optimization module 22 comprising programmed instructions and/or configurable
10 hardware logic for one or more aspects of the present invention as described and illustrated herein, although the memory 26 can comprise other types and numbers of systems, devices, and elements in other configurations which store other data.

[0017] The web service automator module 18 can include programmed instructions and/or logic configured to facilitate one or more of the methods
15 described and illustrated in detail below including receiving a web service request from one of the client devices 12(1)-12(n) and providing a web service response. The JavaScript environment emulator 20 can include programmed instructions and/or logic configured to simulate a JavaScript environment for executing one or more JavaScript files, as described and illustrated in more detail below. The web
20 content optimization module can include programmed instructions and/or logic configured to extract content relevant to a requesting client device 12(1)-12(n) and adapt, transform, or otherwise modify this extracted content to fit the specifications of the requesting client device 12(1)-12(n). One or more of the web service automator module 18, the JavaScript environment emulator 20, or the web
25 content optimization module 22 can also have other types and numbers of functions as described and illustrated herein. Additionally, one or more of the web service automator module 18, the JavaScript environment emulator 20, or the web content optimization module 22 can be stored at and/or implemented by a separate device coupled to the web content proxy server 16 by one or more of the
30 communication networks 24(1)-24(2), for example.

[0018] The interface system 30 in the web content proxy server 16 is used to operatively couple and communicate between the web content proxy server 16,

- 6 -

the client devices 12(1)-12(n) and the server devices 14(1)-14(n) via the communication networks 24(1)-24(2), although other types and numbers of communication networks 24(1)-24(2) with other types and numbers of connections and configurations can be used. Additionally, one or more of the communication networks 24(1)-24(2) can include one or more networks, such as one or more local area networks (LANs) and/or wide area networks (WANs). By way of example only, the communication networks can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardware communication technology, each having their own communications protocols, can be used.

[0019] The client devices 12(1)-12(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the server devices 14(1)-14(n) through the web content proxy server 16 via one or more communication networks 24(1), although one or more of the client devices 12(1)-12(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. In some examples, the client devices 12(1)-12(n) comprise mobile computing devices with Internet access that enable one or more web services to be accessed. By way of example only, the client devices 12(1)-12(n) can be smart phones, personal digital assistants, or computers.

[0020] In some examples, the client devices 12(1)-12(n) include a central processing unit (CPU) or processor, a memory, a user input device, a display, and an interface system, which are coupled together by a bus or other link, although one or more of client devices 12(1)-12(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor in the client devices 12(1)-12(n) can execute a program of instructions stored in the memory of the client device 12(1)-12(n) for one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

- 7 -

[0021] The user input device in the client devices 12(1)-12(n) can be used to input selections, such as a request for a particular web service, although the user input device could be used to input other types of requests and data and interact with other elements. The user input device can include keypads, touch screens, and/or vocal input processing systems although other types and numbers of user input devices can be used.

[0022] The display the client device can be used to show data and information to the user, such as a response to the web service request by way of example only. The display in the client device is a phone screen display, although other types and numbers of displays could be used depending on the particular type of client device.

[0023] The interface system in the client devices 12(1)-12(n) can be used to operatively couple and communicate between the client devices 12(1)-12(n), the web content proxy server 16, and the server devices 14(1)-14(n) over the communication networks 24(1)-24(2).

[0024] The server devices 14(1)-14(n) provide content including web pages for use by one or more of the client devices 12(1)-12(n) via the web content proxy server 16, although the server devices 14(1)-14(n) can provide other numbers and types of functions.

[0025] Each of the server devices 14(1)-14(n) can include a central processing unit (CPU) or processor, a memory, and an interface system which are coupled together by a bus or other link, although each of the web server devices 14(1)-14(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations. The processor in each of the server devices 14(1)-14(n) executes a program of instructions stored in the memory of the server devices 14(1)-14(n) for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

- 8 -

[0026] The interface system in each of the server devices 14(1)-14(n) is used to operatively couple and communicate between the server devices 14(1)-14(n), the web content proxy server 16, and the client devices 12(1)-12(n) via communication networks 24(1)-24(2).

5 [0027] Although embodiments web content proxy server 16, the client devices 12(1)-12(n), and the server devices 14(1)-14(n), are described and illustrated herein, each of the web content proxy server 16, the client devices 12(1)-12(n), and the server devices 14(1)-14(n), can be implemented on any suitable computer apparatus or computing device. It is to be understood that the
10 apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0028] Furthermore, each of the devices of the embodiments may be
15 conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0029] In addition, two or more computing apparatuses or devices can be
20 substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that
25 extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched
30 Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

- 9 -

[0030] The embodiments may also be embodied as one or more non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor,
5 cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0031] An exemplary method for implementing web services will now be described below. This technology defines methods, non-transitory computer readable medium, and devices to implement web services using web site(s) and/or
10 web page(s), and associated content and/or source code as the source data, to produce results including one or more web service responses. Additionally, as many web sites depend on web cookies as a way to store and keep the state of a transaction or session, web services implemented as described with this technology can extend the standard web service definition to include web cookies
15 to keep the mapping between web services and a web site data source.

[0032] This technology also utilizes a JavaScript environment emulator 20 configured to make asynchronous JavaScript and XML (Ajax) calls and to execute web site transactions corresponding to the functionality implemented by web services. By way of example only, a JavaScript environment emulator 20 in a
20 web content proxy server 16 is illustrated and described in U.S. Patent Application Serial No. 12/802,670 entitled, "Methods For Utilizing A JavaScript Emulator In A Web content proxy Server And Devices Thereof" which is herein incorporated by reference in its entirety.

[0033] As illustrated in FIG. 1, the web service automator module 18 can
25 be an application comprising programmed instructions as described and illustrated herein and executed by the web content proxy server 16 that interacts with one of the client devices 12(1)-12(n) making a web service request and server device(s) 14(1)-14(n) acting as the source data for the web service, although other configurations can be used. By way of example, the web service automator
30 module 18 can be a module in memory 26 executed by the web content proxy server 16. Optionally, the web content proxy server 16 can further execute the

- 10 -

web content optimization module 22 which can be used as a “preprocessor” that modifies web content before the content is passed to the web service automator module 18.

[0034] Referring to FIG. 2, a diagram of an exemplary login page 200,
5 such as may be located at a *login.html uniform resource locator (URL) or web address, where “*” is a wild card referring to a host name and/or one or more higher level domains, for example, and its HTML source code 202, is shown. The HTML source code 202 includes a form pointing to a /login.do script and having a “sessionid” hidden field, “user” and “passwd” text fields for receiving inserted
10 login credentials, and a submit button labeled “Login”.

[0035] When a user of one of the client devices 12(1)-12(n) submits valid login credentials and engages the “Login” submit button on the web page 200 with a user input device, the user can receive a web page, such as the exemplary web page 300 shown in FIG. 3. The web page 300 includes a welcome message
15 and a list of links to access different functionalities and logout.

[0036] Further illustrated in FIG. 3 is the HTML source code 302 associated with the successful login web page 300. Upon submission of the form, the submitted values for the “user” and “passwd” fields will be communicated along with the value for the hidden “sessionid” field. Optionally, the responding
20 device, such as one of the server devices 14(1)-14(n), can use a set-cookie header populated with the “sessionid” field value (e.g. sessionid=dhfumsn3942b) as a cookie to store identification information for the session. Accordingly, in this example, an expected web service response for a submission including a valid username and password is XML data that can be rendered in a browser of the
25 client device 12(1)-12(n) and which includes a welcome message for the validated user such as “<message>welcome message...</message>”.

[0037] Referring to FIG. 4, an exemplary method for implementing a web service is illustrated. In step 400, the web content proxy server 16 executing the web service automator module 18 receives a web service request from one of the
30 client devices 12(1)-12(n). Referring to a web service implementation of the login

- 11 -

process illustrated in FIGS. 2 and 3, the received web service request can be associated with a URL such as

“https://processor.com/ws/samplesite/login?username=jake&password=123”,

where processor.com is the web site hosting the web service, /ws/ is a web

5 application running a web service engine, samplesite/login is a URL substring identifying the web service, and the username and password are the two parameters of the web service. While the username and password are shown in this exemplary URL, the user name and password parameter values can be hidden using techniques that are well-known in the art.

10 [0038] In step 402, the web content proxy server 16 executing the web service automator module 18 determines whether there is a JavaScript file corresponding to the web service. Accordingly, in the above example, the web content proxy server 16 executing the web service automator module 18 determines whether there is a project in the memory called “samplesite” which
15 contains a JavaScript file called “login.js”. If the web content proxy server 16 executing the web service automator module 18 determines that the identified JavaScript file does not exist, then a No branch is taken to step 404. In step 404, the web content proxy server 16 executing the web service automator module 18 sends an error page web service response to the requesting one of the client
20 devices 12(1)-12(n) indicating the web service is not available.

[0039] If the web content proxy server 16 executing the web service automator module 18 determines that the identified JavaScript file does exist, then a Yes branch is taken to step 406. In step 406, the web content proxy server 16 executing the web service automator module 18 communicates with the JavaScript
25 environment emulator 20 in order to generate a JavaScript environment, such as illustrated and described by way of example only in U.S. Patent Application Serial No. 12/802,670 which is herein incorporated by reference in its entirety.

[0040] In step 408, the web content proxy server 16 executing the web service automator module 18 creates a JavaScript object representing the web
30 service, although the object can be generated in other manners. With reference to FIG. 5, in which JavaScript code 500 for implementing a function facilitating

- 12 -

generation of a web service response is shown, the JavaScript object can be named “webService”, although other names can also be used. In this example, the created JavaScript object contains the parameters of the web service request (e.g. user name and password values), as retrieved from the URL and/or through HTTP
5 GET and/or POST commands, for example.

[0041] In step 410, the web content proxy server 16 executing the JavaScript environment emulator 20 executes a function in the matching JavaScript file identified in step 402. In the example shown in FIG. 5, the function is named “service”, although other functions and/or names, the identity
10 of which is known by the web content proxy server 16, can also be used. The function is executed using the environment generated in step 406 and the created web service object as an input parameter. In this example, the “service” function uses a library, such as jQuery, that simplifies the use of Ajax inside the JavaScript environment in the JavaScript environment emulator 20 of the web content proxy
15 server 16, although any other functions or libraries can also be used. The executed function is responsible for populating the created web service object (e.g. the “webService” object), with the necessary data to construct a web service response.

[0042] The exemplary “service” function shown in the JavaScript code of
20 FIG. 5 retrieves the login.html page from the responding one of the server devices 14(1)-14(n). Next, the “service” function fills the login.html form text fields with the parameters passed to the function as contained by the web service object input parameter (e.g. “webService”). Once filled, the “service” function submits the form data to the “login.do” script indicated by the form action in the HTML
25 source code 202, simulating submission of the form by the requesting one of the client devices 12(1)-12(n). The “login.do” script can, for example, validate the user name and password parameter values, although any other functionality can also be provided.

[0043] In step 412, the web content proxy server 16 executing the web
30 service automator module 18 generates a web service response based on the input web service object, as modified by the executed function. Accordingly, in the

- 13 -

example shown in FIG. 5, the “service” function further sets a “response” property of the “webService” web service object to include “<message>” and “</message>” XML tags with a text message (e.g. “Hi Jake, welcome back!”) inserted between the tags. Any other XML tags, or any other code capable of
5 being interpreted by the web service automator module 18, and/or the browser of the requesting one of the client devices 12(1)-12(n), can be used to populate the web service object. Additionally, any other web service functionality can be provided by the JavaScript code 500. Accordingly, the web content proxy server 16 executing the web service automator module 18 inspects the web service object
10 returned from, and as modified by, the JavaScript environment emulator 20, and generates the web service response.

[0044] In step 414, the web content proxy server 16 executing the web service automator module 18 provides the generated web service response to the requesting one of the client devices 12(1)-12(n). The response is in a form that
15 can be rendered by the requesting one of the client devices 12(1)-12(n). Optionally, the web content proxy server 16 executing the web service automator module 18 includes a cookie, such as a cookie including the sessionid information in this example, in order to maintain the state of the session between the requesting one of client devices 12(1)-12(n) and the responding one of the server
20 devices 14(1)-14(n).

[0045] Accordingly, as illustrated and described herein this technology provides a number of advantages including providing methods, non-transitory computer readable medium, and devices for more easily implementing web services with a web content proxy server. With this technology, a web content
25 proxy server can implement a web service thereby allowing mobile computing devices to send and receive web service requests and responses even though the devices may not be capable of performing such functionality. Thereby, web services can be more effectively extended and utilized by mobile computing devices.

30 [0046] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is

- 14 -

intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are
5 within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

- 15 -

CLAIMS

What is claimed is:

1. A method for implementing a web service, the method comprising:
5 determining with a web content proxy server when a JavaScript file is associated with a received web service request;
emulating with the web content proxy server a JavaScript environment when the determining indicates the JavaScript file is associated with the received web service request;
10 creating with the web content proxy server a JavaScript object including one or more parameters of the received web service request;
executing with the web content proxy server the JavaScript file associated with the received web service request in the emulated JavaScript environment using the the JavaScript object;
15 constructing with the web content proxy server a web service response based on the JavaScript object as populated based on the executing; and
providing with the web content proxy server the constructed web service response to a source of the received web service request.
20
2. The method as set forth in claim 1 further comprising setting with the web content proxy server one or more cookies based on the constructed web service response.
- 25 3. The method as set forth in claim 1 further comprising providing with the web content proxy server an error message to the source of the received web service request as a web service response when there is no determined JavaScript file associated with the received web service request.
- 30 4. The method as set forth in claim 1 wherein the web content proxy server comprises at least one of a web service automator module, a web content optimization module, or a JavaScript environment emulator.

- 16 -

5 5. The method as set forth in claim 4 wherein the JavaScript environment emulator is configured to make asynchronous JavaScript and XML (Ajax) calls and execute web site transactions corresponding to functionality implemented by one or more web services.

10 6. The method as set forth in claim 1 wherein the received web service request comprises at least one of a GET HTTP request, a POST HTTP request, a PUT HTTP request, or a DELETE HTTP request.

 7. The method as set forth in claim 1 wherein the received web service request and the constructed web service response contain cookie and set-cookie headers, respectively.

15 8. The method as set forth in claim 1 wherein the constructed web service response comprises one of a hypertext markup language (HTML), JavaScript Object Notation (JSON), or eXtensible Markup Language (XML) response.

20 9. A non-transitory computer readable medium having stored thereon instructions for implementing a web service comprising machine executable code which when executed by at least one processor, causes the processor to perform steps comprising:

25 determining when a JavaScript file is associated with a received web service request;

 emulating a JavaScript environment when the determining indicates the JavaScript file is associated with the received web service request;

30 creating a JavaScript object including one or more parameters of the received web service request;

 executing the JavaScript file associated with the received web service request in the emulated JavaScript environment using the the JavaScript object;

- 17 -

constructing a web service response based on the JavaScript object as populated based on the executing; and
providing the constructed web service response to a source of the received web service request.

5

10. The medium as set forth in claim 9 further having stored thereon machine executable code that when executed by the at least one processor cause the processor to perform steps further comprising setting one or more cookies based on the constructed web service response.

10

11. The medium as set forth in claim 9 further having stored thereon machine executable code that when executed by the at least one processor cause the processor to perform steps further comprising providing an error message to the source of the received web service request as a web service response when there is no determined JavaScript file associated with the received web service request.

15

12. The medium as set forth in claim 9 wherein the received web service request comprises at least one of a GET HTTP request, a POST HTTP request, a PUT HTTP request, or a DELETE HTTP request.

20

13. The medium as set forth in claim 9 wherein the received web service request and the constructed web service response contain cookie and set-cookie headers, respectively.

25

14. The medium as set forth in claim 9 wherein the constructed web service response comprises one of a hypertext markup language (HTML), JavaScript Object Notation (JSON), or eXtensible Markup Language (XML) response.

30

15. A web content proxy server, comprising:
one or more processors; and

- 19 -

19. The apparatus as set forth in claim 18 wherein the JavaScript environment emulator is configured to make asynchronous JavaScript and XML (Ajax) calls and execute web site transactions corresponding to functionality implemented by one or more web services.

5

20. The apparatus as set forth in claim 15 wherein the received web service request comprises at least one of a GET HTTP request, a POST HTTP request, a PUT HTTP request, or a DELETE HTTP request.

10

21. The apparatus as set forth in claim 15 wherein the received web service request and the constructed web service response contain cookie and set-cookie headers, respectively.

15

22. The apparatus as set forth in claim 15 wherein the constructed web service response comprises one of a hypertext markup language (HTML), JavaScript Object Notation (JSON), or eXtensible Markup Language (XML) response.

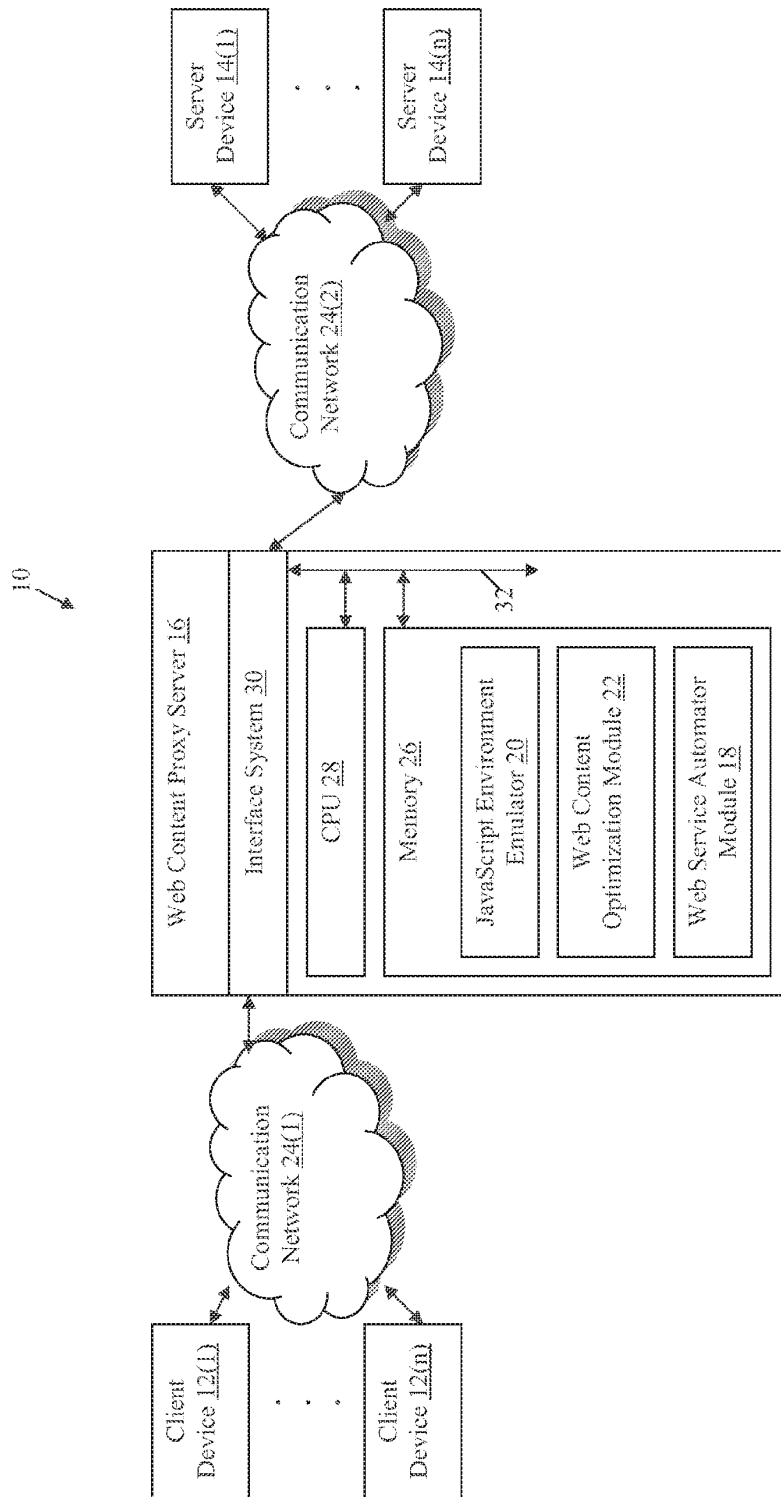


FIG. 1

200

Please Login

User ID:

Password:

Login

202

```
<html>
  <head>
    <title>Login Page</title>
  </head>
  <body>
    <form id="login" method="post" action="/login.do">
      <fieldset>
        <legend>Please Login</legend>
        <input type="hidden" name="sessionId" value="d9fumsn3942b"/>
        <div>
          User ID:<br/>
          <input type="text" id="user" name="user"/>
        </div>
        <div>
          Password:<br/>
          <input type="text" id="passwd" name="passwd"/>
        </div>
        <input type="submit" value="Login"/>
      </fieldset>
    </form>
  </body>
</html>
```

FIG. 2



FIG. 3

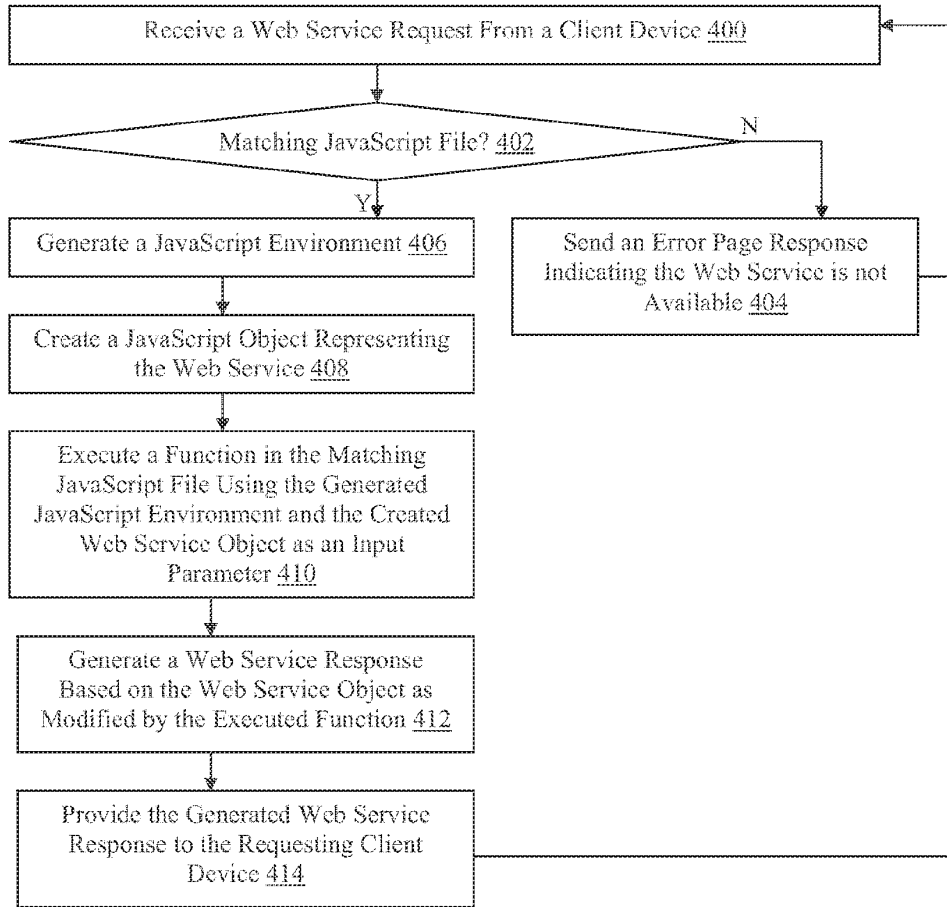


FIG. 4

```
function service(webService) {  
    $( 'body' ).load( 'https://samples.ashx.com/login.html', null, function( responseText, status, xhr ) {  
        $( 'input' ).val( webService.params[ 'username' ] );  
        $( 'input' ).val( webService.params[ 'password' ] );  
        var form = $( '#login' );  
        $.post( form.attr( 'action' ), form.serialize(), function( responseText, status, xhr ) {  
            $( 'body' ).html( responseText );  
            webService.contentType = 'text/html';  
            webService.response = '<message>' + $( 'body div' ).text() + '</message>';  
        });  
    });  
}
```

500



FIG. 5

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 2 732 385 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2014/040080 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2014/040080 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2014/040080 (art. 153(3) CBE).

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(10) International Publication Number
WO 2014/040080 A1

(43) International Publication Date
13 March 2014 (13.03.2014)

- (51) International Patent Classification:
G06F 17/00 (2006.01) G06F 3/14 (2006.01)
G06F 9/45 (2006.01)
- (21) International Application Number:
PCT/US2013/059087
- (22) International Filing Date:
10 September 2013 (10.09.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/698,948 10 September 2012 (10.09.2012) US
13/796,857 12 March 2013 (12.03.2013) US
- (71) Applicant: USABLENET INC. [US/US]; 28 W. 23rd St., 6th Floor, New York, NY 10010 (US).
- (72) Inventor: SCODA, Enrico; Via Cividina 416/3, I-33035 Martignacco, UD (IT).
- (74) Agents: LEINBERG, Gunnar, G. et al.; Leclairryan, A Professional Corporation, 70 Linden Oaks, Suite 210, Rochester, NY 14625 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

(54) Title: METHODS FOR FACILITATING WEB PAGE IMAGE HOTSPOTS AND DEVICES THEREOF

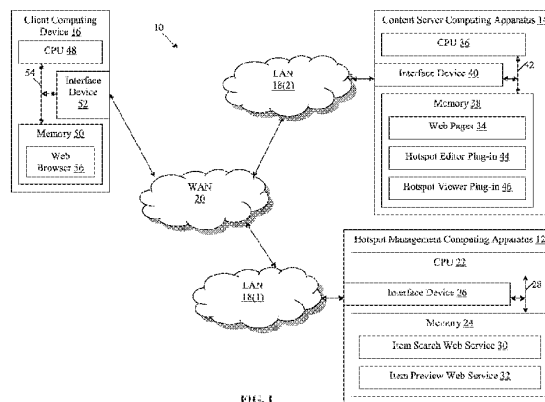


FIG. 1

(57) Abstract: A method, non-transitory computer readable medium, and hotspot management computing apparatus that receives a JavaScript Object Notation with Padding (JSONP) request from a client computing device, wherein the JSONP request comprises an identifier associated with a selected hotspot associated with an image of a first web page. A HyperText Markup Language (HTML) fragment is generated based on the identifier, wherein the HTML fragment defines a first description associated with the hotspot and an input element associated with a Uniform Resource Locator (URL) of a second web page including a second description associated with the hotspot. The HTML fragment is sent to the requesting client computing device.

WO 2014/040080 A1

- 1 -

METHODS FOR FACILITATING WEB PAGE IMAGE HOTSPOTS AND DEVICES THEREOF

[0001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 61/698,948, filed September 10, 2012, which is hereby incorporated by reference
5 in its entirety.

FIELD

[0002] This technology generally relates to web page image hotspots and, more particularly, to methods, non-transitory computer readable medium, apparatuses that
10 facilitate establishing and interacting with web page image hotspots.

BACKGROUND

[0003] Images have become increasingly pervasive across the Internet, particularly with respect to social media and retail web sites for example, although many different types of web sites host images. Hotspots, or identified regions of images, are often established in
15 order for users to associate metadata or other information with the images. For example, a host of an image on a social media web site can establish a hotspot (often referred to as “tagging”) to indicate another user present in the image. Upon selection of the hotspot, the other user’s name and a link to the other user’s profile can be displayed, for example. On a retail web site, a host may identify a region of an image including an item for sale as a
20 hotspot. Upon selection of the hotspot by a user, content regarding the item can be displayed, often with a link to a more detailed description of the item.

[0004] However, web browsers generally prohibit communication across domains based on a same origin policy. Accordingly, the available content for a user to associate with a web page image hotspot is generally limited to content hosted by a web server in the same
25 domain as the web page that includes the image. Additionally, web services hosted by web servers, and configured to identify content to be associated with a hotspot, are often developed for a single domain. Accordingly, such web services do not provide an interface

- 2 -

or structure common to any other web service configured to provide similar functionality but associated with a different domain.

SUMMARY

5 [0005] A method for facilitating interaction with a web page image hotspot includes receiving, by a hotspot management computing apparatus, a JavaScript Object Notation with Padding (JSONP) request from a client computing device, wherein the JSONP request comprises an identifier associated with a selected hotspot associated with an image of a first web page. A HyperText Markup Language (HTML) fragment is generated, by the hotspot
10 management computing apparatus, based on the identifier, wherein the HTML fragment defines a first description associated with the hotspot and an input element associated with a Uniform Resource Locator (URL) of a second web page including a second description associated with the hotspot. The HTML fragment is sent, by the hotspot management computing apparatus, to the requesting client computing device.

15 [0006] A computer readable medium having stored thereon instructions for facilitating interaction with a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps including receiving a JSONP request from a client computing device, wherein the JSONP request comprises an identifier associated with a selected hotspot associated with an image of a first
20 web page. An HTML fragment is generated based on the identifier, wherein the HTML fragment defines a first description associated with the hotspot and an input element associated with a URL of a second web page including a second description associated with the hotspot. The HTML fragment is sent to the requesting client computing device.

[0007] A hotspot management computing apparatus includes a memory coupled to a
25 processor which is configured to execute programmed instructions stored in the memory comprising receiving a JSONP request from a client computing device, wherein the JSONP request comprises an identifier associated with a selected hotspot associated with an image of a first web page. An HTML fragment is generated based on the identifier, wherein the HTML fragment defines a first description associated with the hotspot and an input element

- 3 -

associated with a URL of a second web page including a second description associated with the hotspot. The HTML fragment is sent to the requesting client computing device.

[0008] A method for generating a web page image hotspot includes receiving, with a hotspot management computing apparatus, a JSONP request comprising search criteria.

5 Content satisfying the search criteria is identified by the hotspot management computing apparatus. An HTML fragment defining an input element associated with each of one or more search results and an identifier is generated by the hotspot management computing apparatus, wherein each of the one or more search results includes at least a portion of the identified content. The HTML fragment is sent to the requesting client computing device by
10 the hotspot management computing apparatus.

[0009] A computer readable medium having stored thereon instructions for facilitating generation of a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps including receiving a JSONP request comprising search criteria. Content satisfying the search criteria
15 is identified. An HTML fragment defining an input element associated with each of one or more search results and an identifier is generated, wherein each of the one or more search results includes at least a portion of the identified content. The HTML fragment is sent to the requesting client computing device.

[0010] A hotspot management computing apparatus includes a memory coupled to
20 one or more processors which are configured to execute programmed instructions stored in the memory including receiving a JSONP request comprising search criteria. Content satisfying the search criteria is identified. An HTML fragment defining an input element associated with each of one or more search results and an identifier is generated, wherein each of the one or more search results includes at least a portion of the identified content.
25 The HTML fragment is sent to the requesting client computing device.

[0011] A method for generating a web page image hotspot includes generating, by a client computing device, a search panel in response to a user identifying a region of an image rendered on a web page as a hotspot, the search panel comprising one or more input fields. A

- 4 -

JSONP request is sent, by the client computing device, based on interaction with the input element of the search panel, to a hotspot management computing apparatus, wherein the JSONP request comprises search criteria entered in the one or more input fields. An HTML fragment received from the hotspot management computing apparatus is rendered, by the client computing device, in a results panel, the HTML fragment defining an input element associated with each of one or more search results, wherein each input element is associated with an identifier of one of the search results. The identifier associated with a selected one of the input elements is sent, by the client computing device, to the hotspot management computing apparatus.

10 [0012] A non-transitory computer readable medium having stored thereon instructions for generating a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps including generating a search panel in response to a user identifying a region of an image rendered on a web page as a hotspot, the search panel comprising one or more input fields. A JSONP request is sent based on interaction with the input element of the search panel, to a hotspot management computing apparatus, wherein the JSONP request comprises search criteria entered in the one or more input fields. An HTML fragment received from the hotspot management computing apparatus is rendered in a results panel, the HTML fragment defining an input element associated with each of one or more search results, wherein each input element is associated with an identifier of one of the search results. The identifier associated with a selected one of the input elements is sent to the hotspot management computing apparatus.

25 [0013] A client computing device includes a memory coupled to a processor which is configured to execute programmed instructions stored in the memory including generating a search panel in response to a user identifying a region of an image rendered on a web page as a hotspot, the search panel comprising one or more input fields. A JSONP request is sent based on interaction with the input element of the search panel, to a hotspot management computing apparatus, wherein the JSONP request comprises search criteria entered in the one or more input fields. An HTML fragment received from the hotspot management computing

- 5 -

apparatus is rendered in a results panel, the HTML fragment defining an input element associated with each of one or more search results, wherein each input element is associated with an identifier of one of the search results. The identifier associated with a selected one of the input elements is sent to the hotspot management computing apparatus.

5 [0014] A method for interacting with a web page image hotspot includes displaying, with a client computing device, one or more hotspots of an image included on a first web page. A JSONP request is sent, with the client computing device, to a hotspot management computing apparatus in response to user interaction with one of the one or more hotspots, wherein the JSONP request comprises an identifier associated with the one hotspot. An
10 HTML fragment received from the hotspot management computing apparatus is rendered, with the client computing device, in a preview panel, wherein the HTML fragment is based on the identifier and defines a first description associated with the one hotspot and an input element associated with a URL of a second web page including a second description associated with the one hotspot. The second web page is displayed, with the client
15 computing device, in response to user interaction with the input element of the preview panel.

[0015] A non-transitory computer readable medium having stored thereon instructions for interacting with a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps including
20 displaying one or more hotspots of an image included on a first web page. A JSONP request is sent, with the client computing device, to a hotspot management computing apparatus in response to user interaction with one of the one or more hotspots, wherein the JSONP request comprises an identifier associated with the one hotspot. An HTML fragment received from the hotspot management computing apparatus is rendered, with the client computing device,
25 in a preview panel, wherein the HTML fragment is based on the identifier and defines a first description associated with the one hotspot and an input element associated with a URL of a second web page including a second description associated with the one hotspot. The second web page is displayed, with the client computing device, in response to user interaction with the input element of the preview panel.

- 6 -

[0016] A client computing device including a memory coupled to a processor which is configured to execute programmed instructions stored in the memory including displaying one or more hotspots of an image included on a first web page. A JSONP request is sent, with the client computing device, to a hotspot management computing apparatus in response to user interaction with one of the one or more hotspots, wherein the JSONP request comprises an identifier associated with the one hotspot. An HTML fragment received from the hotspot management computing apparatus is rendered, with the client computing device, in a preview panel, wherein the HTML fragment is based on the identifier and defines a first description associated with the one hotspot and an input element associated with a URL of a second web page including a second description associated with the one hotspot. The second web page is displayed, with the client computing device, in response to user interaction with the input element of the preview panel.

[0017] This technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatuses that facilitate establishing web page image hotspots using a plug-in sent along with a web page. The plug-in can obtain data to be associated with the hotspot from a web service hosted by a different domain using JSONP. This technology also facilitates user interaction with web page image hotspots using a plug-in configured to communicate using JSONP with a web service, which can also be hosted by a different domain.

20 BRIEF DESCRIPTION OF THE DRAWINGS

[0018] FIG. 1 is a block diagram of a network environment which incorporates an exemplary hotspot management computing apparatus;

[0019] FIG. 2 is a flowchart of an exemplary method of inserting a web page image hotspot;

25 [0020] FIG. 3 is a flowchart of ;an exemplary method of facilitating the association of a web page image hotspot with content;

- 7 -

[0021] FIG. 4 is a flowchart of an exemplary method of displaying a web page image hotspot;

[0022] FIG. 5 is a flowchart of an exemplary method of facilitating the display of a web page image hotspot;

5 [0023] FIG. 6 is an exemplary web page including an image and a plurality of hotspots associated with the image;

[0024] FIG. 7 is an exemplary search panel for obtaining search criteria for content to be associated with a web page image hotspot;

10 [0025] FIG. 8 is an exemplary search results panel including identified content to be associated with a web page image hotspot; and

[0026] FIG. 9 is an exemplary web page including a plurality of hotspots and an exemplary preview panel associated with one of the hotspots.

DETAILED DESCRIPTION

[0027] An exemplary network environment 10 is illustrated in FIG. 1 as including an
15 exemplary hotspot management computing apparatus 12. In this example, the hotspot management computing apparatus 12 is coupled to a content server computing apparatus 14 and a client computing device 16 by local area networks (LANs) 18(1) and 18(2) and a wide area network (WAN) 20, although other types and numbers of devices, components, and elements in other topologies could be used. This technology provides a number of
20 advantages including methods, non-transitory computer readable medium, and apparatuses for more easily and effectively generating and viewing web page image hotspots across domains and sources of content associated with the hotspots.

[0028] Referring more specifically to FIG. 1, the hotspot management computing apparatus 12 includes at least one processor or CPU 22, a memory 24, and an interface
25 device 26, which are coupled together by a bus 28 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and

- 8 -

locations can be used. The processor 22 of the hotspot management computing apparatus 12 may execute one or more stored programmed instructions for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor 22 could execute other numbers and types of programmed instructions.

5 [0029] The memory 24 of the hotspot management computing apparatus 12 stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. The memory 24 of the hotspot management computing apparatus 12 may include one or more tangible storage media and/or devices, such as RAM, ROM,
10 flash memory, CD-ROM, floppy disk, hard disk drive(s), solid state memory, DVD, or any other memory storage types or devices, including combinations thereof, which are known to those of ordinary skill in the art.

[0030] In this example, the memory 24 of the hotspot management computing apparatus 12 includes an item search web service 30 and an item preview web service 32
15 including JavaScript instructions, for example, for one or more aspects of the present invention as described and illustrated herein, although the memory 24 can include other types and numbers of systems, devices, and elements in other configurations.

[0031] The interface device 26 in the hotspot management computing apparatus 12 is used to operatively couple and communicate between the hotspot management computing
20 apparatus 12 and the client computing device 16 via LAN 18(1) and WAN 20, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. The LANs 18(1) and 18(2) and WAN 20 can use TCP/IP over Ethernet and industry-standard protocols, including NFS, CIFS, SOAP, XML, LDAP, and SNMP, for example, although other types
25 and numbers of communication networks can also be used.

[0032] Generally, the content server computing apparatus 14 processes requests for web pages 34 received from the client computing device 16 via LAN 18(2) and WAN 20 according to the HTTP-based protocol, for example. The content server computing apparatus

- 9 -

14 includes at least one processor or CPU 36, a memory 38, and an interface device 40, which are coupled together by a bus 42 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can also be used. The processor 36 in the content server computing apparatus 14 executes a
5 program of stored instructions for one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 36 could execute other numbers and types of programmed instructions.

[0033] The memory 38 in the content server computing apparatus 14 stores these programmed instructions for one or more aspects of the present invention, as described and
10 illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing
15 system that is coupled to the processor 36, can be used for the memory 38 in the content server computing apparatus 14.

[0034] In this example, the memory 38 includes a hotspot editor plug-in 44 and a hotspot viewer plug-in 46 including JavaScript instructions, for example, configured to be included in any of the web pages stored by the memory and provided in response to a client
20 computing device request. The memory 38 can further include other types and numbers of systems, devices, and elements in other configurations which store other data.

[0035] The interface device 40 in the content server computing apparatus 14 is used to operatively couple and communicate between the content server computing apparatus 14 and the client computing device 16 via LAN 18(2) and WAN 20, although other types and
25 numbers of communication networks with other types and numbers of connections and configurations can be used. The content server computing apparatus 14 may be hardware or software or may represent a system with multiple content server computing apparatuses in a pool, which may include internal or external networks. In this example the content server

- 10 -

computing apparatus 14 may be any version of Microsoft[®] IIS server or Apache[®] server, although other types of content server computing apparatus 14 may be used.

[0036] The client computing device 16 enables a user to request, receive and interact with services and content hosted by the content server computing apparatus 14 and hotspot management computing apparatus 12 via the LANs 18(1) and 18(2) and WAN 20, although
5 the client computing device 16 could access content and utilize other types and numbers of content or applications from other sources and could provide a wide variety of other functions for a user. By way of example only, the client computing device 16 can be a mobile computing device, smart phone, personal digital assistant, or computer, for example.

10 [0037] In this example, the client computing device 16 includes at least one processor or CPU 48, a memory 50, and an interface device 52, and which are coupled together by a bus 54 or other link, although the client computing device 16 can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 48 in the client computing device 16 executes a program of stored instructions for
15 one or more aspects of the present invention as described and illustrated herein, although the processor 48 could execute other numbers and types of programmed instructions.

[0038] The memory 50 in the client computing device 16 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed
20 elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 48 can be used for the memory 50 in the client computing device 16. In this example, the client computing device 16 is
25 configured to access web services and web content through a web browser 56 stored in the memory 50.

[0039] The interface device 52 in the client computing device 16 is used to operatively couple and communicate between the client computing device 16 and the content

- 11 -

server computing apparatus 14 and the hotspot management computing apparatus 12 via the LANs 18(1) and 18(2) and the WAN 20, although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

5 [0040] Although embodiments of the hotspot management computing apparatus 12, content server computing apparatus 14, and client computing device 16 are described and illustrated herein, each of these devices can be implemented on any suitable computer system or computing device. It is to be understood that the devices and systems of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and
10 software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s). Furthermore, each of the systems of the embodiments may be conveniently implemented using one or more general purpose computer systems, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be
15 appreciated by those ordinary skill in the art.

[0041] In addition, two or more computing systems or devices can be substituted for any one of the devices in any embodiment. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the
20 embodiments. The embodiments may also be implemented on computer system(s) that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications
25 networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0042] The examples may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when

- 12 -

executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0043] An exemplary method for facilitating web page image hotspots will now be described with reference to FIGS. 1-9. Referring specifically to FIG. 2, an exemplary
5 method of inserting a web page image hotspot will now be described. In this example in step 200, a user of the client computing device 16 requests and obtains, using the web browser 56, one of the web pages 34 hosted by the content server computing apparatus 14 and including one or more images. An exemplary one of the web pages 34 stored in the memory 24 of the content server computing apparatus 14 is illustrated in FIG. 6 as web page 600, which
10 includes an image 602.

[0044] In this example, along with sending the requested web page 600 to the client computing device 16, the content server computing apparatus 14 sends the hotspot editor plug-in 44. The content server computing apparatus 14 can send the hotspot editor plug-in 44 by inserting the JavaScript code of the plug-in 44 into the source code of the requested
15 web page 600. In another example, the content server computing apparatus 14 provides the hotspot editor plug-in 44 along with the requested web page 600 as a bookmarklet. In yet another example, the hotspot editor plug-in 44 can be integrated within a platform, such as a blog engine or a social network platform, for example. Other methods of providing the hotspot editor plug-in 44 can also be used. Upon receiving the requested web page 600, the
20 web browser 52 of the client computing device 16 attempts to render the requested web page on a display of the client computing device 16.

[0045] In step 202, while rendering the web page 600, and upon encountering the JavaScript code of the hotspot editor plug-in 44, the web browser 52 of the client computing device 16 executes the code to generate a hotspot identification interface. The interface is
25 configured to facilitate identifying a region of the image 602 of the requested web page 600 as a hotspot. The interface can be presented visually by the web browser 52, such as in an overlay, new panel, or new window, for example, which allows selection of the hotspot features (e.g. size or representation style) by the user. Alternatively, the interface can be run

- 13 -

in the background without a graphical interface while still allowing the user to select regions of the image 602 of the web page 600 to be identified as hotspots.

[0046] In step 204, the web browser 52 of the client computing device 16 executing the hotspot editor plug-in 44 generates a search panel. In this example, the web browser 52
5 generates the search panel in response to a user of the client computing device 16 identifying a region of the image 602 of the requested web page 600 as a hotspot using the interface generated in step 202. An exemplary identified hotspot of the image 602 of the web page 600 is illustrated as hotspot 604 in FIG. 6. Referring to FIG. 7, an exemplary search panel 700 generated by the web browser 52 of the client computing device 16 in response to the
10 user identifying a region of the image 602 as hotspot 604 is illustrated. The search panel 700 includes one or more input fields, such as the search text field 702, a source selector element 704, and an input element, such as the search button 706, although other types and number of input fields and elements can also be used.

[0047] In this example, the source selector element 704 is a drop-down menu
15 identifying a plurality of optional sources selectable by the user. Each source is associated by the hotspot editor plug-in 44 with a Uniform Resource Locator (URL) of an item search web service, such as the item search web service 30 stored in the memory 24 of the hotspot management computing apparatus 12, as described and illustrated in more detail below. The source selector element 704 can be populated with sources, and associated URLs, based on
20 an identifier associated with the user sent to the content server computing apparatus 12 during an initialization or registration process. In another example, the source selector element 704 can be populated with sources, and associated URLs, based on a hardcoding by an administrator or developer of the hotspot editor plug-in 44. Other methods for populating the source selector element can also be used.

[0048] In step 206, the web browser 52 of the client computing device 16 executing
25 the hotspot editor plug-in 44 sends, upon user interaction with the input element 706 of the search panel 700, a JavaScript Object Notation with Padding (JSONP) request to the item search web service 30 of the hotspot management computing apparatus 12. In this example, the JSONP request includes search criteria input using the input field 702 of the search panel

- 14 -

700 and is sent to the item search web service 30 located at the URL associated with the source selected by the user in the source selector element 704. Exemplary sources identified using the source selector element can be a product catalog, a blog post, or a social network feed, although any other types of sources can also be used.

5 [0049] Each source and/or item search web service 30 can be located on the same or different hotspot management computing apparatus 12. The search criteria input by the user is used by the item search web service 30 to locate content that can be presented to and identified by the user as associated with the hotspot 604. Optionally, the JSONP request further includes an attribute indicating a layout to be used as described and illustrated in
10 more detail below. The layout can be a default layout included in the JavaScript code of the hotspot editor plug-in 44, for example, although other methods of indicating a layout can also be used.

[0050] In this example, the JSONP request is sent to a URL endpoint associated with the selected source. Additionally, the JSONP request identifies the item search web
15 service 30 hosted by a hotspot management computing apparatus 12. By using the JSONP protocol, the hotspot editor plug-in 44 can advantageously communicate with an item search web service 30 that is cross-domain or located at a URL including a different domain, host, port, and/or application layer protocol than a URL of the content server computing apparatus 14 storing the web page 600 and the hotspot editor plug-in 44.

20 [0051] Accordingly, with this example of the technology, the hotspot editor plug-in 44 and item search web service 30 are decoupled such that the plug-in 44 can be included with or inserted into the source code of any of the web pages 34 and the item search web service 30 can be located on any server computing device despite web browser enforcement of a same origin policy. The URL of the item search web service 30 can be included in the
25 hotspot editor plug-in 44 such that user interaction with the input element 706 of the search panel 700 initiates the JSONP request to the item search web service 30 located at the URL.

[0052] In step 208, the web browser 52 of the client computing device 16 executing the hotspot editor plug-in 44 renders a HyperText Markup Language (HTML) fragment. The

- 15 -

HTML fragment is received by the web browser 52 of the client computing device from, and is generated by, the item search web service 30, as described in more detail below. The HTML fragment, when rendered, is configured to display a results panel 800, such as the results panel 800 illustrated in FIG. 8, for example. The content of the results panel 800 can also be output to a display of the client computing device 16 in the form of a new window or overlay, for example, although any other form of display can also be used.

[0053] Optionally, the item search web service 30 can generate the HTML fragment according to a layout attribute value received in the JSONP request sent by the client computing device 16 in step 206. Referring more specifically to FIG. 8, in this example, the HTML fragment defines results panel input elements 802(1) and 802(2), each of which is associated with one of the search results 804(1) and 804(2). Additionally, each of the results panel input elements 802(1) and 802(2) is associated with an attribute having a value of an identifier of an associated one of the search results 804(1) and 804(2). Accordingly, in this example, the item search web service 30 processes the search criteria sent in the JSONP request in step 206 and sends a response with search results 804(1) and 804(2) and including an HTML fragment defining the display of those search results 804(1) and 804(2).

[0054] Referring back to FIG. 2, in step 210, the client computing device 16 executing the hotspot editor plug-in 44 stores the identifier associated with one of the results panel input elements 802(1) and 802(2) selected by a user to be associated with the identified hotspot 604. For example, the hotspot editor plug-in 400 can be configured to send the association information to the hotspot management computing apparatus 12 for storage in the memory 24. Thereby, the hotspot 604 of the image 602 of the web page 600 will be associated with the content of one of the search results 804(1) and 804(2) corresponding to the selected one of the results panel input elements 802(1) and 802(2). Accordingly, the content of the one of the search results 804(1) and 804(2) can be displayed by a web browser of another client computing device upon subsequent retrieval of the web page 600 and selection of the hotspot 604 by a user of the other client computing device, as described and illustrated in more detail below.

- 16 -

[0055] Referring more specifically to FIG. 3, an exemplary method of facilitating the association of a web page image hotspot with content is illustrated. In this example, in step 300, the hotspot management computing apparatus 12 implementing the item search web service 30 receives a JSONP request from a hotspot editor plug-in 44, such as the JSONP request sent in step 206. As discussed above, the item search web service 30 of the hotspot management computing apparatus 12 can be located at a different domain, host, port, and/or application layer protocol than a URL of the content server computing apparatus 14 that provided the hotspot editor plug-in 44 that originated the JSONP request. The JSONP request includes search criteria submitted by a user of the client computing device 16 using the search results panel 700, such as a source of the content to be associated with the hotspot and text search terms for the content.

[0056] In step 302, the hotspot management computing apparatus 12 implementing the item search web service 30 identifies content satisfying the search criteria to generate search result(s). Accordingly, the source and text inputs, for example, can be used by the item search web service 30 to locate possible content that the user of the client computing device 16 may want to associate with an identified hotspot, such as hotspot 604 of the image 602 on the web page 600, for example, as illustrated in FIG. 6.

[0057] In step 304, the hotspot management computing apparatus 12 implementing the item search web service 30 generates an HTML fragment defining an input element associated with each search result, such as input elements 802(1) and 802(2) associated with search result 804(1) and 804(2), respectively, illustrated in FIG. 8. Additionally, an identifier of each search result 804(1) and 804(2) is associated by the item search web service 30 with each input element 802(1) and 802(2) by the HTML fragment. Optionally, the JSONP request further includes an indication of a layout to be used for presenting the search results 804(1) and 804(2) and the HTML fragment can be generated according to the layout.

[0058] In step 306, the hotspot management computing apparatus 12 implementing the item search web service 30 sends the HTML fragment to the requesting hotspot editor plug-in 44 executed by the web browser 52 of the client computing device 16. The results

- 17 -

panel 800 shown in FIG. 8 is an example of a rendering by the web browser 52 of the client computing device 16 of the HTML fragment generated in step 306.

[0059] Referring more specifically to FIG. 4, an exemplary method of displaying a web page image hotspot is shown. In this example, in step 400, a user of the client
5 computing device 16 requests and obtains, using the web browser 52, one of the web pages 34 hosted by the content server computing apparatus 14 and including one or more images, such as the exemplary web page 600 illustrated in FIG. 6 which includes the image 602. Along with sending the requested web page 600 to the client computing device 16, the content server computing apparatus 14 sends the hotspot viewer plug-in 46, such as by
10 inserting the JavaScript code of the hotspot viewer plug-in 46 into the source code of the requested web page 600, although other methods of providing the hotspot viewer plug-in can also be used. The web browser 52 of the client computing device 16 then attempts to render the requested web page 600 on a display of the client computing device 16.

[0060] While rendering the web page 600, and upon encountering the JavaScript code
15 of the hotspot viewer plug-in 46, in step 402, the web browser 52 of the client computing device 16 executes the code to display the hotspot(s) of the image(s) included in the requested web page, such as the hotspot 604 of the image 602 of the web page 600 illustrated in FIG. 6. The hotspot 604 can be inserted into the image 602 of the web page 600 as described and illustrated above with respect to FIGS. 2-3, for example, although other
20 methods of inserting the hotspot 604 can be used.

[0061] In step 404, the web browser 52 of the client computing device 16 executing the hotspot viewer plug-in 46 sends a JSONP request to the item preview web service 32 of the hotspot management computing apparatus 12 in response to a user interaction with one of the hotspots, such as hotspot 604. In this example, the JSONP request includes an identifier
25 associated with the hotspot 604. The identifier can be the identifier stored as described with respect to the step 210, for example. Accordingly, the web page 600 obtained from the content server computing apparatus 14 in step 400 can include one of the stored identifiers as associated with each hotspot 604. Upon user interaction with the hotspot 604 of the image

- 18 -

602 of the web page 600, the JSONP request including the identifier is generated and sent to the item preview web service 32.

[0062] Optionally, the JSONP request further includes an attribute indicating a layout to be used, as described and illustrated in more detail below. The layout can be a default
5 layout included in JavaScript code of the hotspot viewer plug-in 46, for example, although other methods of indicating a layout can also be used. Also optionally, the JSONP request can further include tracking information associated with the client computing device 16
executing the hotspot viewer plug-in 46.

[0063] The item preview web service 32 can be located at a URL including a
10 different domain, host, port, or application layer protocol than the URL of the content server computing apparatus 14 from which the web page 600 was obtained in the step 400. The URL of the item preview web service 32 can be included in the hotspot viewer plug-in 44 such that user interaction with the hotspot 604 initiates the JSONP request to the item
preview web service 46 located at the URL.

[0064] In step 406, the web browser 52 of the client computing device 16 executing
15 the hotspot viewer plug-in 46 renders an HTML fragment received from and generated by the item preview web service 32, as described and illustrated in more detail below. The HTML fragment defines a preview panel, such as the exemplary preview panel 900 illustrated in
FIG. 9. The content of the preview panel 900 can also be output to a display of the client
20 computing device 16 in the form of a new window or overlay, for example, although any other form of display can also be used.

[0065] The HTML fragment is generated based on the identifier and defines at least a
first, generally shorter, description associated with the hotspot and an input element, such as
the input element 904 illustrated in FIG. 9. Accordingly, based on the identifier, the item
25 preview web service 32 can obtain the content associated with the hotspot 604, such as the content associated with the selected search result 804(1). Upon obtaining the content, the item preview web service 32 can generate the HTML fragment according to the layout
attribute value received in the JSONP request or a default layout, for example. In some

- 19 -

example, the input element 902 of the preview panel 900 is associated with a URL of a second web page including a second, generally longer, description associated with the hotspot 604, such as the remaining portion(s) of the content not displayed in the preview panel 900.

5 [0066] In step 408, the web browser 52 of the client computing device 16 executing the hotspot viewer plug-in 46 displays the second web page in response to a user interaction with the input element 904 of the preview panel 900. Optionally, the input element of the preview panel 900 is further associated with a viewer attribute. The viewer attribute can indicate the behavior to be used to generate the second web page (e.g. in a new panel or in a
10 new window). Accordingly, in this example, the second web page is displayed on a display of the client computing device 16 according to the viewer attribute.

[0067] Referring to FIG. 5, an exemplary method of facilitating the display of a web page image hotspot is illustrated. In this example, in step 500, the hotspot management computing apparatus 12 implementing the item preview web service 32 receives a JSONP
15 request from the hotspot viewer plug-in 46, such as the JSONP request sent in the step 404, for example. The JSONP request includes at least an identifier associated with a selected web page image hotspot, such as the hotspot 604 of the image 602 of the web page 600, for example. The item preview web service 32 of the hotspot management computing apparatus 12 can be located at a different domain, host, port, and/or application layer protocol than a
20 URL of the content server computing apparatus 12 that provided the hotspot viewer plug-in 46 that originated the JSONP request.

[0068] In step 502, the hotspot management computing apparatus 12 implementing the item preview web service 32 generates an HTML fragment based on the identifier associated with the hotspot 604 sent along with the JSONP request. The HTML fragment
25 defines a description associated with the hotspot 604, such the description associated with the content of the search result 804(1), as well as an input element 902, as described above. Additionally, in this example, the input element 902 is associated with a URL of a second web page including a second description associated with the selected hotspot 604 of the image 602 of the web page 600.

- 20 -

[0069] Optionally, the JSONP request further includes an indication of a layout to be used for presenting the preview content and the HTML fragment is generated according to the layout. Also optionally, the JSONP request includes tracking information associated with the client computing device 16 executing the hotspot viewer plug-in 46. The tracking
5 information can be stored in the memory 24 of the hotspot management computing apparatus 12 for subsequent retrieval by an administrator, for example, although the tracking information can also be sent and/or stored elsewhere.

[0070] In step 504, the hotspot management computing apparatus 12 implementing the item preview web service 32 sends the HTML fragment to the requesting hotspot viewer
10 plug-in 46 executed by the web browser 52 of the client computing device 16. The preview panel 900 shown in FIG. 9 is an example of a rendering by the web browser 52 of the client computing device 16 of the HTML fragment generated in step 504.

[0071] Accordingly, as illustrated and described herein this technology provides a number of advantages including improved methods, non-transitory computer readable
15 medium, and devices for establishing and generating web page image hotspots. With this technology, inserting or editing an image hotspot of a web page can be facilitated by a JavaScript plug-in sent along with the web page. The JavaScript plug-in communicates with a JavaScript web service configured to locate content that can be associated with the hotspot. The JavaScript web service can be hosted by a hotspot management computing apparatus
20 outside the domain of the content server apparatus hosting the web page.

[0072] Additionally, viewing image hotspots of a web page can be facilitated by a JavaScript plug-in sent along with the web page. The JavaScript plug-in communicates with a JavaScript web service configured to generate a preview panel to be displayed upon
25 selection of one of the hotspots. The JavaScript web service can also be hosted by a hotspot management computing apparatus outside the domain of the content server apparatus hosting the web page.

[0073] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be

- 21 -

presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the
5 recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

- 22 -

CLAIMS

What is claimed is:

1. A method for facilitating interaction with a web page image hotspot,
5 the method comprising:
receiving, by a hotspot management computing apparatus, a JavaScript
Object Notation with Padding (JSONP) request from a client computing device, wherein the
JSONP request comprises an identifier associated with a selected hotspot associated with an
image of a first web page;
10 generating, by the hotspot management computing apparatus, a
HyperText Markup Language (HTML) fragment based on the identifier, wherein the HTML
fragment defines a first description associated with the hotspot and an input element
associated with a Uniform Resource Locator (URL) of a second web page including a second
description associated with the hotspot; and
15 sending, by the hotspot management computing apparatus, the HTML
fragment to the requesting client computing device.
2. The method as set forth in claim 1, wherein the client computing
device is a mobile computing device, the JSONP request further comprises an indication of a
20 layout, and the HTML fragment is generated according to the layout.
3. The method as set forth in claim 1, wherein the JSONP request further
comprises tracking information associated with the client computing device and the method
further comprises storing, with the hotspot management computing apparatus, the tracking
25 information as associated with the client computing device.
4. The method as set forth in claim 1, wherein one or more of the steps
are performed by an item preview web service.

30

- 23 -

5. A computer readable medium having stored thereon instructions for facilitating interaction with a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:
- receiving a JavaScript Object Notation with Padding (JSONP) request
5 from a client computing device, wherein the JSONP request comprises an identifier associated with a selected hotspot associated with an image of a first web page;
- generating a HyperText Markup Language (HTML) fragment based on the identifier, wherein the HTML fragment defines a first description associated with the hotspot and an input element associated with a Uniform Resource Locator (URL) of a second
10 web page including a second description associated with the hotspot; and
- sending the HTML fragment to the requesting client computing device.
6. The medium as set forth in claim 5, wherein the client computing
15 device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is generated according to the layout.
7. The medium as set forth in claim 5, wherein the JSONP request further comprises tracking information associated with the client computing device and the medium
20 further has stored thereon instructions comprising machine executable code which when executed by the processor causes the processor to perform steps further comprising storing the tracking information as associated with the client computing device.
8. The medium as set forth in claim 5, wherein one or more of the steps
25 are performed by an item preview web service.
9. A hotspot management computing apparatus, comprising:
- a memory coupled to a processor which is configured to execute programmed instructions stored in the memory comprising:
30 receiving a JavaScript Object Notation with Padding (JSONP)

- 24 -

request from a client computing device, wherein the JSONP request comprises an identifier associated with a selected hotspot associated with an image of a first web page;

generating a HyperText Markup Language (HTML) fragment based on the identifier, wherein the HTML fragment defines a first description associated with the hotspot and an input element associated with a Uniform Resource Locator (URL) of a second web page including a second description associated with the hotspot; and
5 sending the HTML fragment to the requesting client computing device.

10 10. The apparatus as set forth in claim 9, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is generated according to the layout.

15 11. The apparatus as set forth in claim 9, wherein the JSONP request further comprises tracking information associated with the client computing device and the processor is further configured to execute programmed instructions stored in the memory further comprising storing the tracking information as associated with the client computing device.

20 12. The apparatus as set forth in claim 9, wherein one or more of the steps are performed by an item preview web service.

25 13. A method for interacting with a web page image hotspot, the method comprising:
displaying, with a client computing device, one or more hotspots of an image included on a first web page;
30 sending, with the client computing device, a JavaScript Object Notation with Padding (JSONP) request to a hotspot management computing apparatus in response to user interaction with one of the one or more hotspots, wherein the JSONP request comprises an identifier associated with the one hotspot;

- 25 -

rendering, with the client computing device, a HyperText Markup Language (HTML) fragment received from the hotspot management computing apparatus in a preview panel, wherein the HTML fragment is based on the identifier and defines a first description associated with the one hotspot and an input element associated with a Uniform Resource Locator (URL) of a second web page including a second description associated with the one hotspot; and

displaying, with the client computing device, the second web page in response to user interaction with the input element of the preview panel.

10 14. The method as set forth in claim 13, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is rendered according to the layout.

15 15. The method as set forth in claim 13, wherein the JSONP request further comprises tracking information associated with the client computing device.

20 16. The method as set forth in claim 13, wherein the client computing device is a mobile computing device, the input element of the preview panel is further associated with a viewer attribute, and the second web page is displayed according to the viewer attribute.

25 17. The method as set forth in claim 13, wherein the JSONP request is sent to or the HTML fragment is received from an item preview web service of the hotspot management computing apparatus.

30 18. A non-transitory computer readable medium having stored thereon instructions for interacting with a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:
displaying one or more hotspots of an image included on a first web page;

- 26 -

sending a JavaScript Object Notation with Padding (JSONP) request to a hotspot management computing apparatus in response to user interaction with one of the one or more hotspots, wherein the JSONP request comprises an identifier associated with the one hotspot;

5 rendering a HyperText Markup Language (HTML) fragment received from the hotspot management computing apparatus in a preview panel, wherein the HTML fragment is based on the identifier and defines a first description associated with the one hotspot and an input element associated with a Uniform Resource Locator (URL) of a second web page including a second description associated with the one hotspot; and
10 displaying the second web page in response to user interaction with the input element of the preview panel.

19. The medium as set forth in claim 18, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is rendered according to the layout.
15

20. The medium as set forth in claim 18, wherein the JSONP request further comprises tracking information associated with the client computing device.

20 21. The medium as set forth in claim 18, wherein the client computing device is a mobile computing device, the input element of the preview panel is further associated with a viewer attribute, and the second web page is displayed according to the viewer attribute.

25 22. The medium as set forth in claim 18, wherein the JSONP request is sent to or the HTML fragment is received from an item preview web service of the hotspot management computing apparatus.

30

- 27 -

23. A client computing device, comprising:
a memory coupled to a processor which is configured to execute
programmed instructions stored in the memory comprising:
displaying one or more hotspots of an image included on a first
5 web page;
sending a JavaScript Object Notation with Padding (JSONP)
request to a hotspot management computing apparatus in response to user interaction with
one of the one or more hotspots, wherein the JSONP request comprises an identifier
associated with the one hotspot;
10 rendering a HyperText Markup Language (HTML) fragment
received from the hotspot management computing apparatus in a preview panel, wherein the
HTML fragment is based on the identifier and defines a first description associated with the
one hotspot and an input element associated with a Uniform Resource Locator (URL) of a
second web page including a second description associated with the one hotspot; and
15 displaying the second web page in response to user interaction
with the input element of the preview panel.

24. The device as set forth in claim 23, wherein the client computing
device is a mobile computing device, the JSONP request further comprises an indication of a
20 layout, and the HTML fragment is rendered according to the layout.

25. The device as set forth in claim 23, wherein the JSONP request further
comprises tracking information associated with the client computing device.

25 26. The device as set forth in claim 23, wherein the client computing
device is a mobile computing device, the input element of the preview panel is further
associated with a viewer attribute, and the second web page is displayed according to the
viewer attribute.

30

- 28 -

27. The device as set forth in claim 23, wherein the JSONP request is sent to or the HTML fragment is received from an item preview web service of the hotspot management computing apparatus.

5 28. A method for generating a web page image hotspot, the method comprising:
receiving, with a hotspot management computing apparatus, a
JavaScript Object Notation with Padding (JSONP) request from a client computing device,
the JSONP request comprising search criteria;
10 identifying, by the hotspot management computing apparatus, one or
more search results satisfying the search criteria;
generating, by the hotspot management computing apparatus, a
HyperText Markup Language (HTML) fragment defining an input element associated with
each of the one or more search results and an identifier, wherein each of the one or more
15 search results includes content to be associated with a web page image hotspot when an
associated one of the input elements is selected by a user of the client computing device; and
sending, by the hotspot management computing apparatus, the HTML
fragment to the client computing device.

20 29. The method as set forth in claim 28, wherein the client computing
device is a mobile computing device, the JSONP request further comprises an indication of a
layout, and the HTML fragment is generated according to the layout.

25 30. The method as set forth in claim 28, wherein one or more of the steps
are performed by an item search web service and the JSONP request is received from a
hotspot editor plug-in retrieved by the client computing device from a content server
computing apparatus located in a different domain than the item search web service.

30 31. A computer readable medium having stored thereon instructions for
facilitating generation of a web page image hotspot comprising machine executable code

- 29 -

which when executed by a processor, causes the processor to perform steps comprising:

receiving a JavaScript Object Notation with Padding (JSONP) request from a client computing device, the JSONP request comprising search criteria;

identifying one or more search results satisfying the search criteria;

5 generating a HyperText Markup Language (HTML) fragment defining an input element associated with each of the one or more search results and an identifier, wherein each of the one or more search results includes content to be associated with a web page image hotspot when an associated one of the input elements is selected by a user of the client computing device; and

10 sending the HTML fragment to the requesting client computing device.

32. The medium as set forth in claim 31, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is generated according to the layout.

33. The medium as set forth in claim 31, wherein one or more of the steps are performed by an item search web service and the JSONP request is received from a hotspot editor plug-in retrieved by the client computing device from a content server computing apparatus located in a different domain than the item search web service..

34. A hotspot management computing apparatus, comprising:

a memory coupled to one or more processors which are configured to execute programmed instructions stored in the memory comprising:

25 receiving a JavaScript Object Notation with Padding (JSONP) request from a client computing device, the JSONP request comprising search criteria;

identifying one or more search results satisfying the search criteria;

30 generating a HyperText Markup Language (HTML) fragment defining an input element associated with each of the one or more search results and an

- 31 -

search results; and

sending, by the client computing device, the identifier associated with a selected one of the input elements to the hotspot management computing apparatus.

5 38. The method as set forth in claim 37, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is rendered according to the layout.

10 39. The method as set forth in claim 37, wherein the JSONP request is sent to an item search web service of the hotspot management computing apparatus by a hotspot editor plug-in retrieved from a content server computing apparatus located in a different domain than the item search web service.

15 40. A non-transitory computer readable medium having stored thereon instructions for generating a web page image hotspot comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:
generating a search panel in response to a user identifying a region of an image rendered on a web page as a hotspot, the search panel comprising one or more input fields;

20 sending based on interaction with the input element of the search panel a JavaScript Object Notation with Padding (JSONP) request to a hotspot management computing apparatus, wherein the JSONP request comprises search criteria entered in the one or more input fields;

25 receiving, by the client computing device, a HyperText Markup Language (HTML) fragment from the hotspot management computing apparatus in response to the JSONP request;

rendering the HTML fragment in a results panel, the HTML fragment defining an input element associated with each of one or more search results, wherein each input element is associated with an identifier of one of the search results; and

30 sending the identifier associated with a selected one of the input

- 32 -

elements to the hotspot management computing apparatus.

41. The medium as set forth in claim 40, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is rendered according to the layout.

42. The medium as set forth in claim 40, wherein the JSONP request is sent to an item search web service of the hotspot management computing apparatus by a hotspot editor plug-in retrieved from a content server computing apparatus located in a different domain than the item search web service.

43. A client computing device, comprising:
a memory coupled to a processor which is configured to execute programmed instructions stored in the memory comprising:
generating a search panel in response to a user identifying a region of an image rendered on a web page as a hotspot, the search panel comprising one or more input fields;
sending based on interaction with the input element of the search panel a JavaScript Object Notation with Padding (JSONP) request to a hotspot management computing apparatus, wherein the JSONP request comprises search criteria entered in the one or more input fields;
receiving, by the client computing device, a HyperText Markup Language (HTML) fragment from the hotspot management computing apparatus in response to the JSONP request;
rendering the HTML fragment in a results panel, the HTML fragment defining an input element associated with each of one or more search results, wherein each input element is associated with an identifier of one of the search results; and
sending the identifier associated with a selected one of the input elements to the hotspot management computing apparatus.

30

- 33 -

44. The device as set forth in claim 43, wherein the client computing device is a mobile computing device, the JSONP request further comprises an indication of a layout, and the HTML fragment is rendered according to the layout.

5 45. The device as set forth in claim 43, wherein the JSONP request is sent to an item search web service of the hotspot management computing apparatus by a hotspot editor plug-in retrieved from a content server computing apparatus located in a different domain than the item search web service.

10

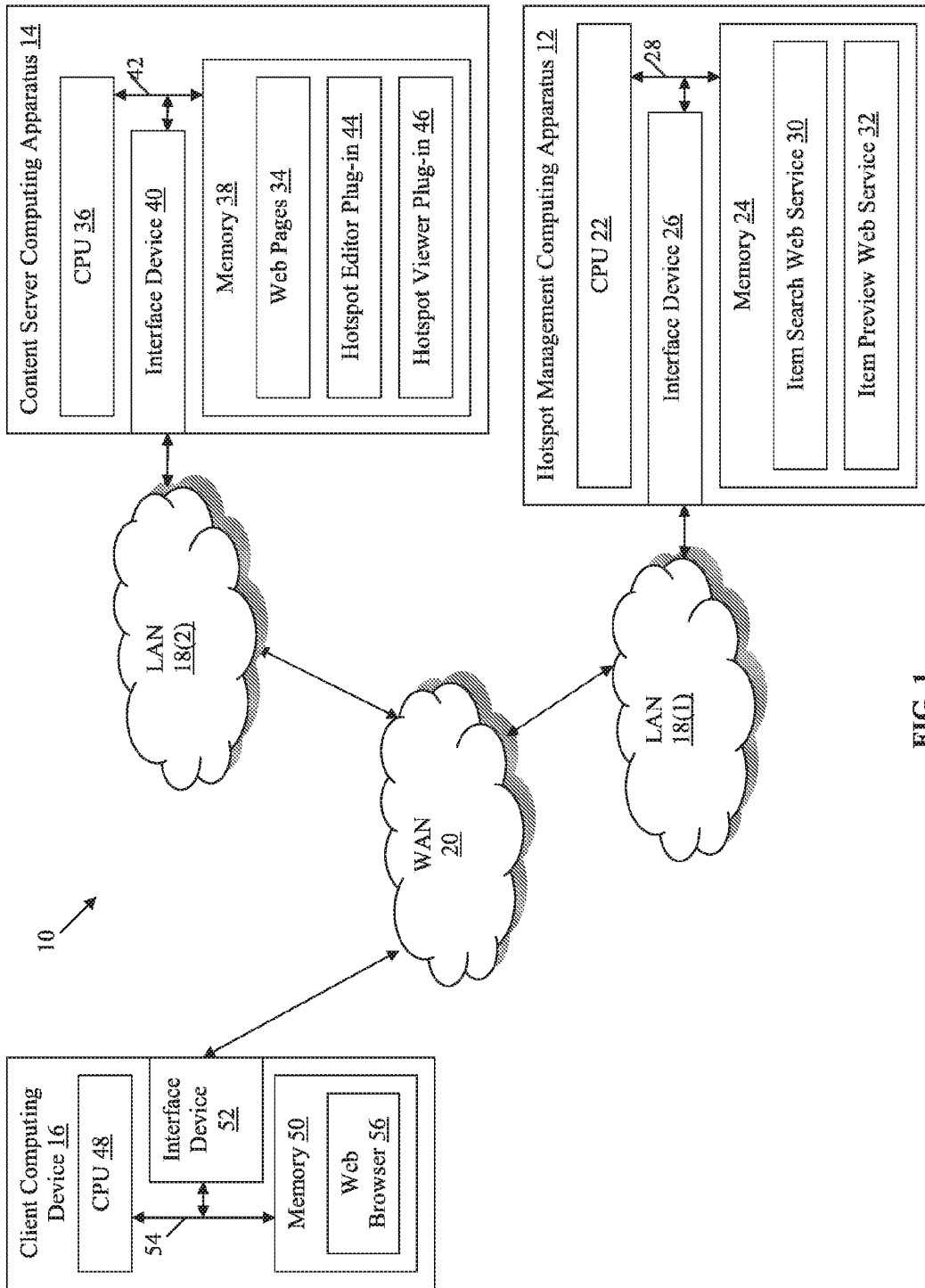


FIG. 1

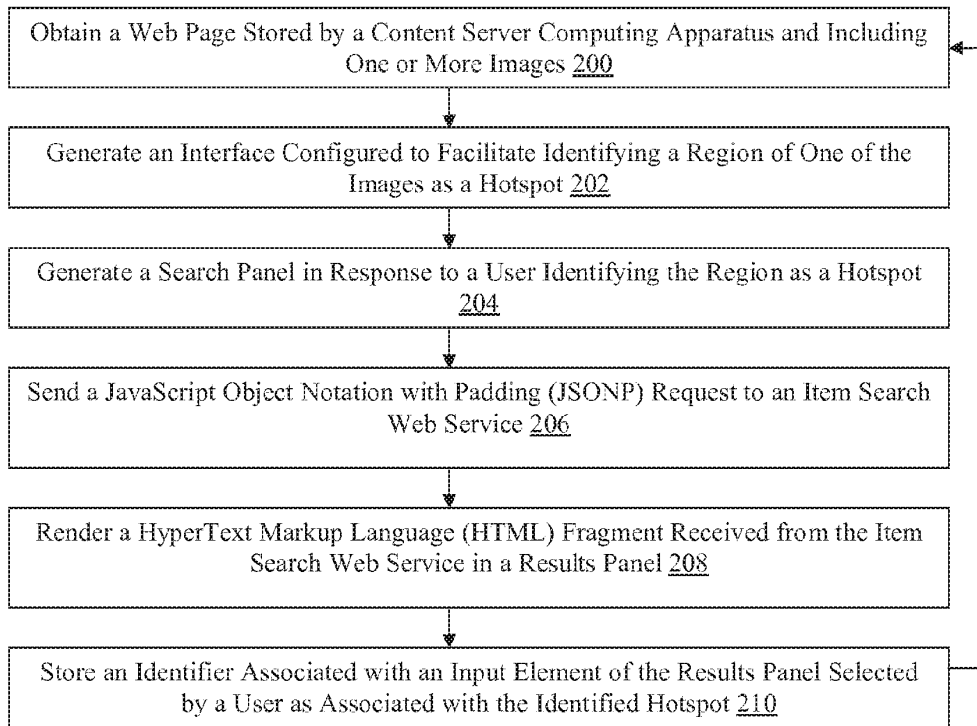


FIG. 2

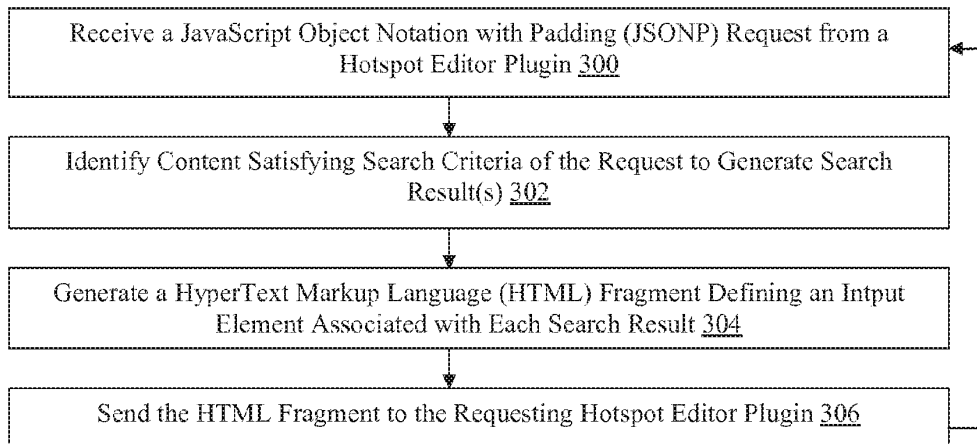


FIG. 3

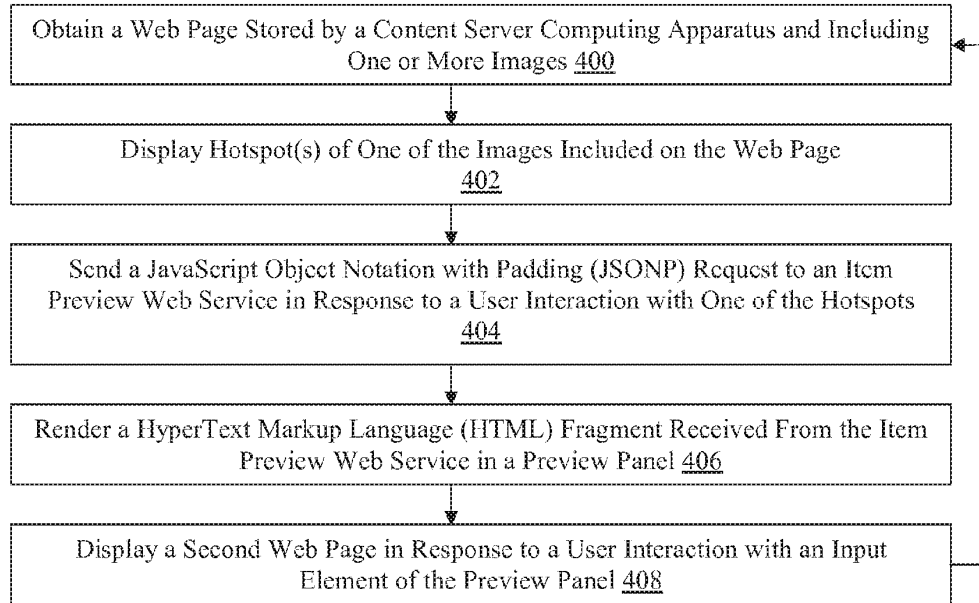


FIG. 4

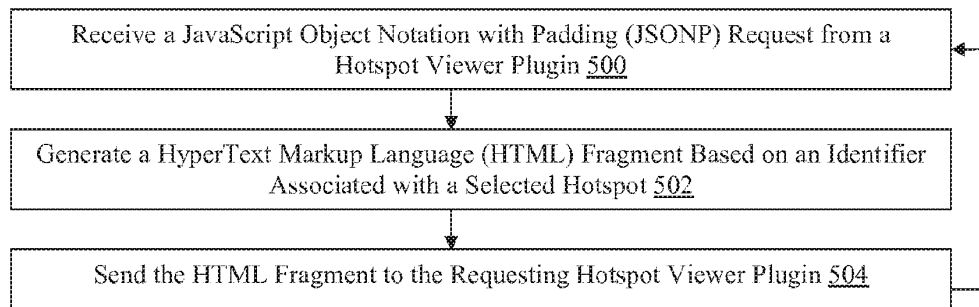


FIG. 5

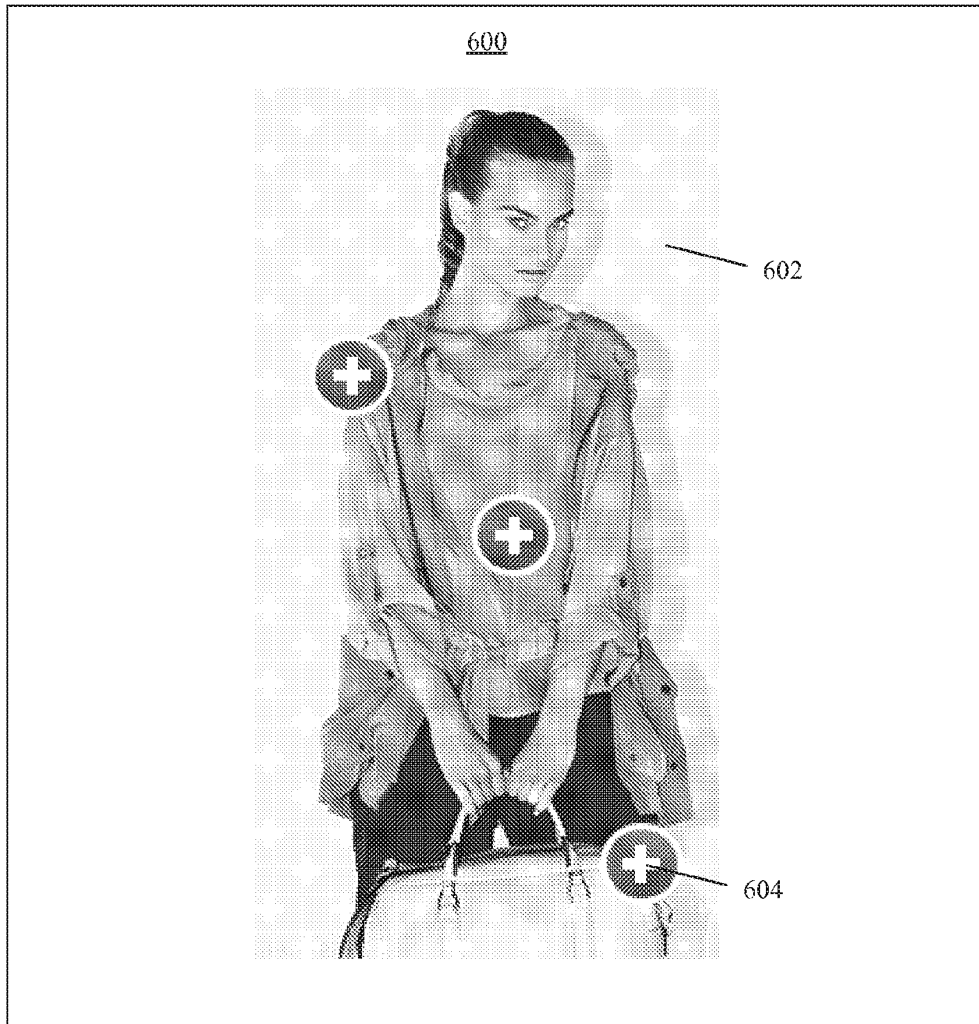


FIG. 6

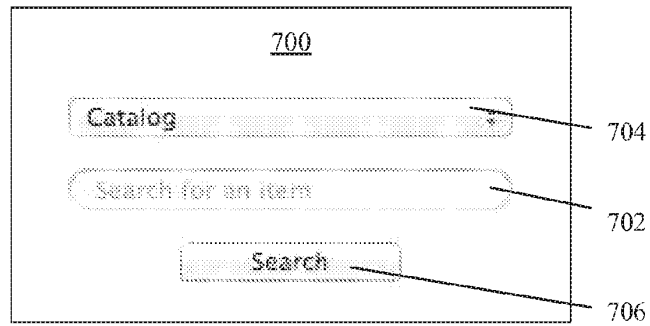


FIG. 7



FIG. 8

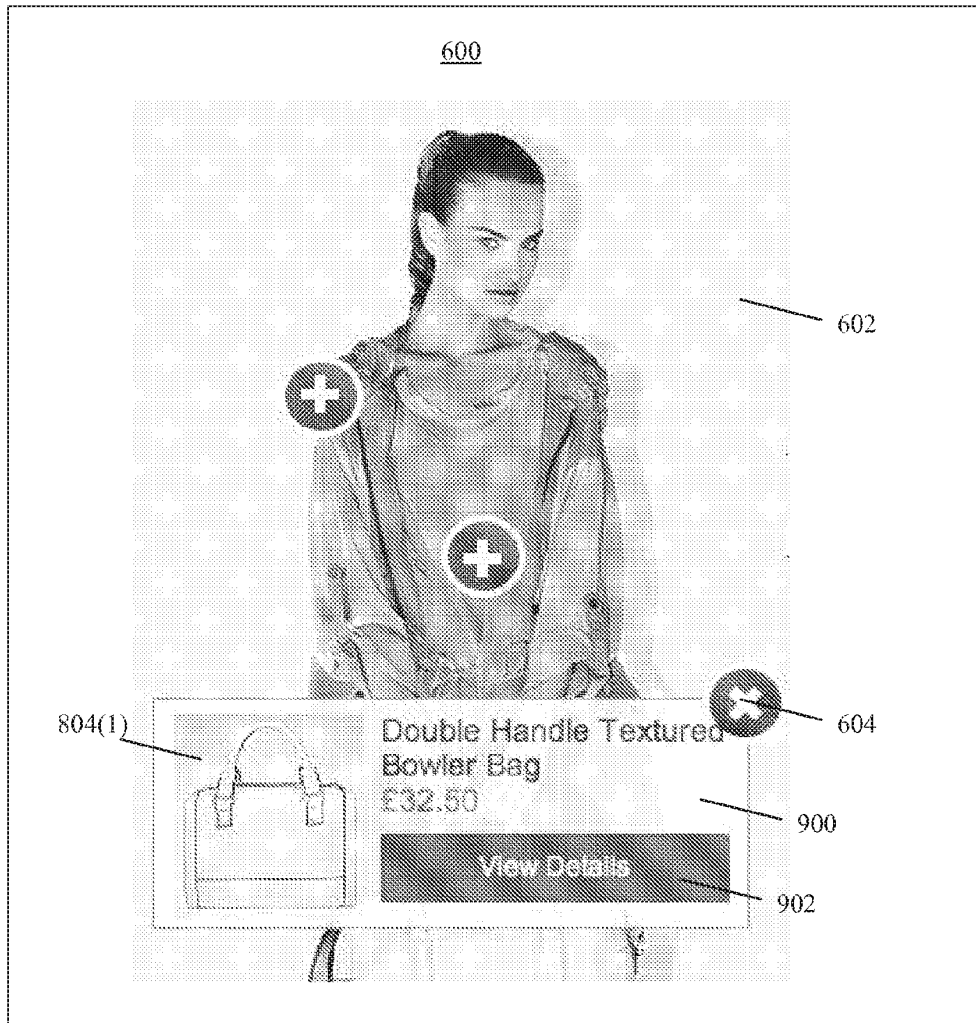




FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/059087

A. CLASSIFICATION OF SUBJECT MATTER G06F 17/00(2006.01)i, G06F 9/45(2006.01)i, G06F 3/14(2006.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F 17/00; G06F 15/16; G06F 15/00; G06F 17/30; G06F 13/00; G06F 9/45; G06F 3/14		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: facebook, image, tag, identified region, hotspot, JSONP, identifier, and similar terms		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011-0264736 A1 (MARK E. ZUCKERBERG et al.) 27 October 2011 See paragraphs [0006], [0021], [0041]; and figure 3.	1-45
A	US 2010 0125623 A1 (TIMOTHY S. RICE et al.) 20 May 2010 See paragraph [0006]; figure 7; and claims 1, 4.	1 45
A	WO 2012-011151 A1 (EMPIRE TECHNOLOGY DEVELOPMENT LLC) 26 January 2012 See paragraphs [0008], [0036]; and claim 1.	1-45
A	WO 2012-030550 A1 (CBS INTERACTIVE INC. et al.) 08 March 2012 See paragraphs [0023]-[0026], [0039]; figures 1-2, 5 ; and claim 25.	1-45
A	US 2011-0202531 A1 (ZUCKERBERG MARK et al.) 18 August 2011 See paragraphs [0055]-[0058]; and figures 1, 5, 8.	1-45
<input type="checkbox"/> Further documents are listed in the continuation of Box C.		<input checked="" type="checkbox"/> See patent family annex.
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search 23 December 2013 (23.12.2013)		Date of mailing of the international search report 26 December 2013 (26.12.2013)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. +82-42-472-7140		Authorized officer HWANG, Yun Koo  Telephone No. +82-42-481-5715

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/059087

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0264736 A1	27/10/2011	AU 2011-243009 A1 CA 2796944 A1 CN 102812432 A EP 2561436 A1 JP 2013-525883 A US 2012-239745 A1 US 2012-278394 A1 US 2012-284614 A1 US 2012-284615 A1 US 8250145 B2 US 8280959 B1 WO 2011-133403 A1	13/09/2012 27/10/2011 05/12/2012 27/02/2013 20/06/2013 20/09/2012 01/11/2012 08/11/2012 08/11/2012 21/08/2012 02/10/2012 27/10/2011
US 2010-0125623 A1	20/05/2010	US 7941546 B2	10/05/2011
WO 2012-011151 A1	26/01/2012	CN 103026350 A EP 2596431 A1 KR 10-2013-0040255 A US 2012-0023156 A1	03/04/2013 29/05/2013 23/04/2013 26/01/2012
WO 2012-030550 A1	08/03/2012	US 2012-0054596 A1	01/03/2012
US 2011-0202531 A1	18/08/2011	US 2008-091723 A1 US 2011-202822 A1 US 2011-225481 A1 US 2011-231747 A1 US 7945653 B2	17/04/2008 18/08/2011 15/09/2011 22/09/2011 17/05/2011



(11) EP 2 738 696 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
04.06.2014 Bulletin 2014/23

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

(21) Application number: 13194794.7

(22) Date of filing: 28.11.2013

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Usablenet Inc.**
New York, NY 10010 (US)

(72) Inventor: **Scoda, Enrico**
33035 Martignacco (IT)

(74) Representative: **Gervasi, Gemma et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(30) Priority: 29.11.2012 US 201261731205 P

(54) **Methods for providing web search suggestions and devices thereof**

(57) A method, non-transitory computer readable medium, and programmed device that use local storage in a browser to provide one or more web search suggestions includes receiving at least a partial entry in field of a web page. Any responsive web search suggestion en-

tries to the received at least a partial entry are provided in the field of the web page from a web search suggestion file stored in a web browser of the client computing device.

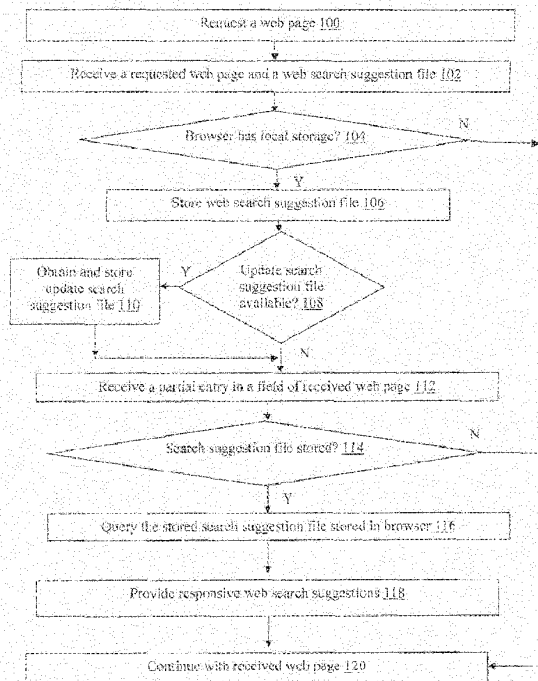


FIG. 2

EP 2 738 696 A1

Description

FIELD

[0001] This technology generally relates to methods and devices for assisting web searching and, more particularly, to methods for using local storage in a browser at a client computing device for providing one or more web search suggestions and devices thereof.

BACKGROUND

[0002] Web sites improve the user experience around "search item" functionalities by predicting a word or a phrase when a user at a client computing device starts typing. By way of example, when the user at the client computing device starts typing the beginning of the word, such as "New" for a city, an airport, or a rail station, then the Web content provider could provide suggestions, such as "New York" and "New Heaven". In another example, when the user at the client computing device starts typing "soc", then the Web content provider could provide suggestions, such as "socks 15" and "soccer 3" where the numbers indicate the number of items in the catalog including the suggested words.

[0003] This search suggestion functionality is a common user interface feature that is usually implemented with Ajax technology at the client computing device. When a user at the client computing device types in a text field, the JavaScript engine in the client computing devices sends a request to a Web content provider. This Web content provider returns a data set, such as JSON data set, that the JavaScript at the client computing device will use to display results as a list of suggestions.

[0004] When the internet connection for the client computing device is fast, i.e. low latency, this search suggestion functionality works pretty well. However when the internet connection for the client computing device is slow or intermittent, such as when browsing the Web from a mobile computing device, then the performance of this search suggestion service often deteriorates.

SUMMARY

[0005] A method using local storage in a browser to provide one or more web search suggestions includes receiving by a client computing device at least a partial entry in field of a web page. Any responsive web search suggestion entries to the received at least a partial entry are provided by the client computing device in the field of the web page from a web search suggestion file stored in a web browser of the client computing device.

[0006] A non-transitory computer readable medium having stored thereon instructions for providing one or more web search suggestions comprising machine executable code which when executed by at least one processor, causes the processor to perform steps comprising receiving at least a partial entry in field of a web page.

Any responsive web search suggestion entries to the received at least a partial entry are provided in the field of the web page from a web search suggestion file stored in a web browser of the client computing device.

5 [0007] A computing device includes a memory coupled to one or more processors which are configured to execute programmed instructions stored in the memory including receiving at least a partial entry in field of a web page. Any responsive web search suggestion entries to the received at least a partial entry are provided in the field of the web page from a web search suggestion file stored in a web browser of a client computing device.

10 [0008] This technology provides a number of advantages including providing methods, non-transitory computer readable medium, and devices that more quickly and effectively provide web search suggestions. In particular, with for example HTML5, this technology can utilize local storage within a browser to store a web search suggestion file for use in providing web search suggestions. As a result, in environments where the internet connection is slower, i.e. has higher latency, this technology is able to more quickly provide web search suggestions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]

FIG. 1 is a block diagram of an environment with a client computing device which uses local storage in a browser to facilitate providing web search suggestions

FIG. 2 is a flow chart of an example of a method for using local storage in a browser to provide one or more web search suggestions;

FIG. 3A is a screen shot of an example of a web page for ACME airlines with two entry fields for a flight reservation; and

FIG. 3B is a screen shot of an example of web search suggestions provided from the web search suggestion file in local storage in the browser of the client computing device.

DETAILED DESCRIPTION

[0010] An exemplary environment 10 with a client computing device 12 which uses local storage in a browser to facilitate providing web search suggestions is illustrated in FIG. 1. In this example, the client computing device 12 is coupled to one or more server devices 14(1)-14(n) by a communication network 16, although other types and numbers of systems, devices, components, and/or elements in other topologies could be used. This technology provides a number of advantages including providing methods, non-transitory computer readable medium, and devices that more quickly and effectively provide

web search suggestions.

[0011] Referring more specifically to FIG. 1, the client computing device 12 is a mobile phone, although other types and numbers of client computing devices may be used, such as computer tablets, laptop computers and desktop computers by way of example only. In this example, the client computing device 12 includes at least one processor or central processing unit (CPU) 18, a memory 20 with a browser 21, a network interface 22, a user input device 24, and a display device 26 which are coupled together by a bus 28 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can also be used. Generally, the client computing device 12 may for example request and receive web pages and other web content from one or more of the server devices 14(1)-14(n) via the communication network 16 according to the HTTP-based protocol, for example, although the client computing device 12 can have and provide other types and numbers of functions and other operations. The processor 18 in the client computing device 12 may execute a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 18 could execute other numbers and types of programmed instructions.

[0012] The memory 20 in the client computing device 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 18, can be used for the memory 20 in the client computing device 12. In this example, the memory includes the browser 21 which may comprise any application configured to for example retrieve, present and/or traverse information resources and other content on the world wide web. With HTML 5, data, such as a web search suggestion file, can be locally stored within the browser 21, although other manners of obtaining local storage with the browser could be used.

[0013] The network interface 22 in the client computing device 12 is used to operatively couple and communicate between the client computing device 12 and the server devices 14(1)-14(n) via the communication network 16, although other types and numbers of networks with other types and numbers of connections and configurations can also be used.

[0014] The user input device 24 in the client computing device 12 can be used to input selections, such as a request for a particular web page or to enter data into a field of a web page, although the user input device could be used to input other types of requests and data and

interact with other elements. The user input device in the client computing device 12 can include a keypad, touch screen, and/or vocal input processing system, although other types and numbers of user input devices can also be used.

[0015] The display device 26 in the client computing device 12 can be used to show data and other information to the user, such as a requested web page by way of example only. The display device 26 in the client computing device 12 can be an LCD, LED, or OLED display, for example, although other types and numbers of displays could be used depending on the particular type of client computing device 12.

[0016] The communication network 16 can include one or more networks, such as one or more wide area networks (WANs), for example the Internet, and/or one or more local area networks (LANs). By way of example only, the communication network 16 can use TCP/IP over Ethernet and industry-standard protocols, including Hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks having their own communications protocols, can also be used.

[0017] The server devices 14(1)-14(n) may each one or more host applications, web services, and/or other data and content which may be requested and retrieved by the client computing device 12 via the communication network 16, although the server devices could each provide a wide variety of other types of functions and other operations.

[0018] In this example, each of the server devices 14(1)-14(n) includes at least one processor or a CPU, a memory, and a network interface, which are coupled together by a bus or other link, although one or more of server devices 14(1)-14(n) can include other numbers and types of systems, devices, components, or other elements in other configurations. The processor in each of the server devices 14(1)-14(n) can execute a program of instructions stored in the memory of each of the server devices 14(1)-14(n) for one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0019] The memory in each of the server devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor can be used for the memory in each of the server devices 14(1)-14(n).

[0020] The network interface in each of the server devices 14(1)-14(n) is used to operatively couple and com-

municate between each of the server devices 14(1)-14(n) and the client computing device 12 via the communication network 16, although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0021] Although embodiments of the client computing device 12 and server devices 14(1)-14(n) are described and illustrated herein, each of the client computing device 12 and the server devices 14(1)-14(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s). Furthermore, each of the devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0022] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0023] The examples may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0024] An exemplary method for using local storage in a client computing device 12 for one or more web search suggestions will now be described with reference to FIGS. 1-3B. In step 100, the client computing device 12 requests a web page of a web site from one of the server devices 14(1)-14(n), although other types of content could be request. The one of the web content server devices 14(1)-14(n) which hosts the web site responds with the requested web page as well as a current version of web search suggestion file for the web site which is received by the client computing device 12 in step 102.

[0025] In step 104, the requesting client computing device 12 determines whether the browser has local storage for the received version of the web search suggestion file. In this example, with HTML 5 the browser 21 in the requesting computing device 12 can provide local storage so the Yes branch is taken to step 106. In step 106, the received version of the web search suggestion file is stored locally within the browser 21 of the client computing device 12 and in this example without the use of a cookie, although other manners of storing locally within the browser 21 can be used. More specifically and by way of example only, once the web search suggestion file is downloaded to the client computing device 12, a set of records can be inserted in to one or more tables in a database in browser 21 to assist with future queries, although other manners for organizing and locally storing the web search suggestion file can be used. If in step 104, the requesting computing device 12 can not provide local storage, then the No branch is taken to step 120 where the user of the client computing device 12 can continue with any viewing or other interaction with the received web page.

[0026] In step 108, the requesting client computing device 12 may optionally determine whether to update any stored version of a web search suggestion file. By way of example only, the requesting client computing device 12 may adhere to a predefined and stored synchronization protocol for updating any stored version of a web search suggestion file, although other manners for maintaining updates of any locally stored web search suggestion file at the requesting one of the client computing devices can be used.

[0027] If in step 108 the requesting client computing device 12 determines to update any stored version of a web search suggestion file, then the Yes branch is taken to step 110. In step 110, the client computing device 12 will request and receive from the one of the server devices 14(1)-14(n) the updated web search suggestion file which is stored locally within the browser 21 in the client computing device 12. If in step 108 the requesting client computing device 12 determines not to update any stored version of a web search suggestion file, then the No branch is taken to step 112.

[0028] In addition to any optional periodic synchronization, anytime the client computing device 12 submits another request for a web page of the web site, the client computing device 12 may optionally perform a version check to compare the current locally stored version of the web search suggestion file for the web site against the version of the web search suggestion file for the web site at the one of the web content server devices 14(1)-14(n) which hosts the web site. If the versions do not match, then the new web search suggestion file may be requested and downloaded from the one of the web content server devices 14(1)-14(n) to the client computing device 12 updating the locally stored database instance in browser 21.

[0029] In step 112, text may be typed into a text field

of a received web page using user input device 24, although other manners for making an entry in other locations of a web page could be used. By way of example only, the text of "New" is typed into a "From" text field 202(1) with a currently empty "To" text field 202(2) of a received web page 200 for "ACME Airlines" for making flight reservations as illustrated in FIG. 3A.

[0030] In step 114, when text is entered into a field of a received web page on display device 26 of the client computing device 12, such as the text of "New" in the "From" text field 202(1), then in this example a JavaScript code executes in the client computing device 12 to determine if a database instance of the web search suggestion file is in the local storage within browser 21, although other types of processes could be used. If in step 114, the client computing device 12 determines a database instance of the web search suggestion file is not stored in the local storage in browser 21, then the No branch is taken to step 120 as described earlier. If in step 114, the client computing device 12 determines a database instance of the web search suggestion file is stored in the local storage in browser 21, then the Yes branch is taken to step 116.

[0031] In step 116, in this example the client computing device 12 executes an SQL query to retrieve all the rows from the stored a database instance of the web search suggestion file within browser 21 containing the text that was entered, although other manners for querying the locally stored web search suggestion file could be used. More specifically, in this particular illustrative example two types of queries could be performed by the client computing device 12, although other types and numbers of queries could be utilized. One type of query retrieves all rows that include all words entered by the user at the client computing device 12 in the version of the web search suggestion file stored by utilizing HTML 5 functionalities. This is a simple query and can be used when the number of results is not high. For example, given the entry of the word "New" entered in the text field 202(1) in FIG. 3A, the client computing device 12 would retrieve or suggest all text containing this string, which in this particular example might suggest for the text "New" in the text field 202(1) in the web page "New York", "New Haven", or "New Plymouth" as illustrated in FIG. 3B by way of example only

[0032] Another type of optional query retrieves all rows that include the string entered by the user at the client computing device 12. Accordingly, this type of query retrieves records that include the entered string entered in the version of the web search suggestion file stored at the client computing device 12 by utilizing HTML 5 functionalities. For example, given a string "soc" entered in a text field which is searched, the client computing device 12 might retrieve, group and suggest items, such as "soccer" (2) or "socks" (8), where the number in parenthesis represents the number of records including the previous string. This technique is more CPU intensive and time consuming, however it provides a better user experience

at the client computing device 12 especially when applied to large catalogs or other data.

[0033] In step 118, the client computing device 12 provides the suggestions identified from the locally stored web search suggestion file for the text field 202(1) as illustrated in FIG. 3B in this example, although other manners for providing the suggestions could be used. In step 120, the client computing device 12 can continue to view and interact with the web page on the display device 26.

[0034] Accordingly, as illustrated and described with the examples herein this technology provides methods, non-transitory computer readable medium, and devices that more quickly and effectively provide web search suggestions. In these examples this technology uses the local storage capabilities provided by HTML5 at the client computing device. With this technology, the process of predicting words or phrases can be applied directly and efficiently in the browser of the client computing device.

[0035] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for providing one or more web search suggestions, the method comprising:

receiving, by a client computing device, at least a partial entry in field of a web page;
providing, by the client computing device, in the field of the web page any responsive web search suggestion entries to the received at least a partial entry from a web search suggestion file stored in a web browser of the client computing device.

2. The method of claim 1 further comprising determining, by the client computing device, whether the web search suggestion file is stored in the web browser of the client computing device, wherein the providing in the field of the web page any responsive web search suggestion entries occurs when the determining indicates the web search suggestion file is stored in the web browser of the client computing device.

3. The method as set forth in claim 2 further comprising:

determining, by the client computing device, whether the web search suggestion file stored in the web browser has an updated version; and
5 obtaining and storing, by the client computing device, the updated version in the web browser at the client computing device when the determining indicates the update version is available.

4. The method as set forth in claim 1 further comprising utilizing, by the client computing device, HTML5 to store the web search suggestion file in the web browser of the client computing device.

5. The method as set forth in claim 1 wherein the providing further comprises:

obtaining, by the client computing device, any responsive web search suggestion entries to the at least a partial entry from the web search suggestion file;
20 grouping, by the client computing device, any of the obtained responsive web search suggestion entries based on at least one common characteristic; and
25 providing, by the client computing device, the grouped responsive web search suggestion entries with a number in each group corresponding to a number of the responsive web search suggestion entries.

6. A non-transitory computer readable medium having stored thereon instructions for providing one or more web search suggestions comprising machine executable code which when executed by at least one processor, causes the processor to perform steps comprising:

receiving at least a partial entry in field of a web page; and
40 providing in the field of the web page any responsive web search suggestion entries to the received at least a partial entry from a web search suggestion file stored in a web browser of the client computing device.

7. The medium of claim 6 further comprising determining whether the web search suggestion file is stored in the web browser of the client computing device, wherein the providing in the field of the web page any responsive web search suggestion entries occurs when the determining indicates the web search suggestion file is stored in the web browser of the client computing device.

8. The medium as set forth in claim 7 further comprising:

determining whether the web search suggestion file stored in the web browser has an updated version; and
obtaining and storing the updated version in the web browser at the client computing device when the determining indicates the update version is available.

9. The medium as set forth in claim 6 further comprising utilizing HTML5 to store the web search suggestion file in the web browser of the client computing device.

10. The medium as set forth in claim 6 wherein the providing further comprises:

obtaining any responsive web search suggestion entries to the at least a partial entry from the web search suggestion file;
grouping any of the obtained responsive web search suggestion entries based on at least one common characteristic; and
providing the grouped responsive web search suggestion entries with a number in each group corresponding to a number of the responsive web search suggestion entries.

11. A computing device comprising:

one or more processors;
a memory coupled to the one or more processors which are configured to execute programmed instructions stored in the memory comprising:

receiving at least a partial entry in field of a web page; and
providing in the field of the web page any responsive web search suggestion entries to the received at least a partial entry from a web search suggestion file stored in a web browser of the client computing device.

12. The device as set forth in claim 11 wherein the memory coupled to the one or more processors is further configured to execute programmed instructions stored in the memory further comprising determining whether the web search suggestion file is stored in the web browser of the client computing device, wherein the providing in the field of the web page any responsive web search suggestion entries occurs when the determining indicates the web search suggestion file is stored in the web browser of the client computing device.

13. The device as set forth in claim 12 wherein the memory coupled to the one or more processors is further configured to execute programmed instructions stored in the memory further comprising:

determining whether the web search suggestion file stored in the web browser has an updated version; and
 obtaining and storing the updated version in the web browser at the client computing device when the determining indicates the update version is available.

5

14. The device as set forth in claim 11 wherein the memory coupled to the one or more processors is further configured to execute programmed instructions stored in the memory further comprising utilizing HTML5 to store the web search suggestion file in the web browser of the client computing device.

10

15

15. The device as set forth in claim 11 wherein the memory coupled to the one or more processors is further configured to execute programmed instructions stored in the memory for the providing further comprises:

20

obtaining any responsive web search suggestion entries to the at least a partial entry from the web search suggestion file;
 grouping any of the obtained responsive web search suggestion entries based on at least one common characteristic; and
 providing the grouped responsive web search suggestion entries with a number in each group corresponding to a number of the responsive web search suggestion entries.

25

30

35

40

45

50

55

7

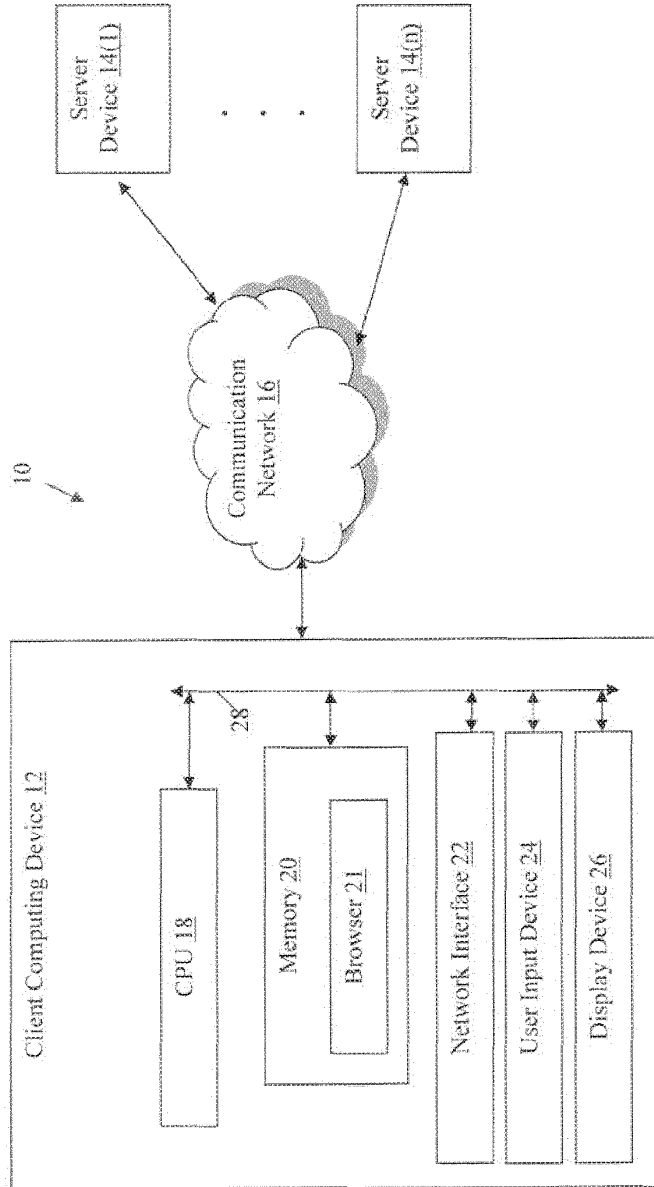


FIG. 1

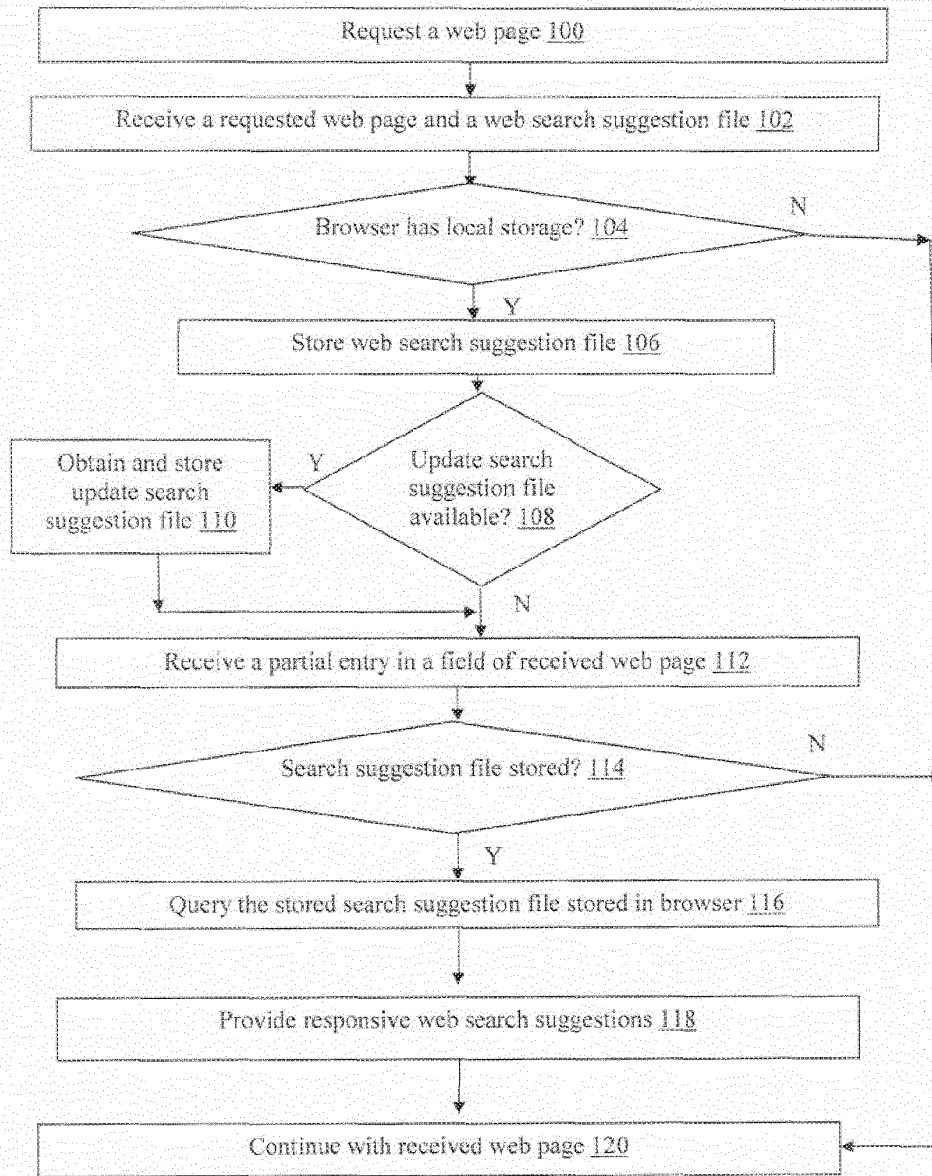


FIG. 2

200

ACME AIRLINES

Flight Reservations:

From: 202(1)

To: 202(2)

FIG. 3A

204

ACME AIRLINES

New York
New Haven
New Plymouth

FIG. 3B



EUROPEAN SEARCH REPORT

Application Number
EP 13 19 4794

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	WO 01/80079 A2 (AMAZON COM INC [US]) 25 October 2001 (2001-10-25) * abstract * * page 4, line 15 - page 6, line 30; figures 1,2a,2b * -----	1-15	INV. G06F17/30
X	US 7 685 144 B1 (KATRAGADDA LALITESH [US]) 23 March 2010 (2010-03-23) * column 1, line 6 - line 11 * * column 5, line 20 - line 37 * * column 7, line 56 - column 8, line 22 * * column 11, line 32 - column 12, line 33 * -----	1-15	
X	US 2012/023120 A1 (KANEFSKY STEVEN T [US]) 26 January 2012 (2012-01-26) * abstract * * paragraphs [0005], [0025] - [0038] * -----	1-15	
			TECHNICAL FIELDS SEARCHED (IPC)
			G06F
The present search report has been drawn up for all claims			
1	Place of search Berlin	Date of completion of the search 29 January 2014	Examiner Siódmok, Wojciech
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document			

1
EPC FORM 1501 (03.03.2011)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 13 19 4794

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29-01-2014

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0180079 A2	25-10-2001	AU 4980001 A	30-10-2001
		EP 1389322 A2	18-02-2004
		US 6564213 B1	13-05-2003
		WO 0180079 A2	25-10-2001

US 7685144 B1	23-03-2010	US 7685144 B1	23-03-2010
		US 7908287 B1	15-03-2011

US 2012023120 A1	26-01-2012	AU 2011282258 A1	31-01-2013
		CN 103221951 A	24-07-2013
		EP 2596441 A2	29-05-2013
		JP 2013535729 A	12-09-2013
		US 2012023120 A1	26-01-2012
		WO 2012012396 A2	26-01-2012

EPC FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82



(11) **EP 2 778 975 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
17.09.2014 Bulletin 2014/38

(51) Int Cl.:
G06F 17/30 (2006.01) H04L 29/08 (2006.01)

(21) Application number: **14159441.6**

(22) Date of filing: **13.03.2014**

(84) Designated Contracting States:
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR**
Designated Extension States:
BA ME

(72) Inventors:
• **Scoda, Enrico**
33035 Martignacco (IT)
• **Brondani, Marco**
33035 Colloredo di Monte Albano (IT)

(30) Priority: **13.03.2013 US 201313801821**

(74) Representative: **Cinquantini, Bruno et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria, 9
20122 Milano (IT)

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(54) **Methods for compressing web page menus and devices thereof**

(57) A method, non-transitory computer readable medium, and apparatus that obtains an original menu associated with a web page requested by a client computing device. The original menu comprises at least a plurality of URLs, each comprising one or more fragments, and a plurality of titles, each comprising one or more words. A unique index for one occurrence of each of at least a subset of the fragments and the words is generated. At least one dictionary comprising the gener-

ated indices associated with a corresponding one of the fragments or the words is generated. A modified menu is generated by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding one of the unique indices. The modified menu and the at least one dictionary are sent to the client computing device.

EP 2 778 975 A1

Description**FIELD OF THE INVENTION**

[0001] This technology generally relates to methods, non-transitory computer readable medium, and apparatuses for compressing source code defining web page menus and, more particularly, for reducing the amount of time required to send, and the amount of memory required to store, source code defining web page menus.

BACKGROUND

[0002] Many web sites, including particularly retail web sites, have a large number of web pages organized based of categories and subcategories of content. In order to present a user with information regarding the organization of the web site, these web pages often include a hierarchical menu. The hierarchical menu allows a user to browse the categories and subcategories without leaving the current web page.

[0003] While a hierarchical menu enhances the user experience, including a hierarchical menu increases the size of the web pages. The increased size is due to the information, including category and subcategory titles and hyperlink addresses, required to present the hierarchical menu. Due to the increased size, web pages with menus require an increased amount of time to retrieve from a server.

[0004] To reduce the time required to retrieve each web page, some web site hosts store the menu source code in an external file separate from the web pages. The external file is downloaded one time at a client device and is referenced by the web pages when the menu is rendered by a web browser of the client device. While the web pages might be retrieved relatively quickly, the initial download of this external file can require a significant amount of time. Additionally, this external file stored at the client device can require a significant amount of memory. In many client devices, particularly mobile devices, such as smart phones, memory is a limited resource and utilizing a significant amount of memory to store this external file defining a menu is not desirable.

SUMMARY OF THE INVENTION

[0005] A method for compressing menus includes obtaining, with a menu management computing apparatus, an original menu associated with a web page requested by a client computing device. The original menu comprises at least a plurality of uniform resource locators (URLs), each comprising one or more fragments, and a plurality of titles, each comprising one or more words. A unique index for one occurrence of each of at least a subset of the one or more fragments and the one or more words is generated with the menu management computing apparatus. At least one dictionary comprising the generated indices associated with a corresponding one of the sub-

set of the one or more fragments or the one or more words is generated with the menu management computing apparatus. A modified menu is generated by the menu management computing apparatus by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding one of the unique indices. The modified menu and the at least one dictionary are sent with the menu management computing apparatus to the client computing device.

[0006] A non-transitory computer readable medium having stored thereon instructions for compressing menus includes machine executable code which when executed by a processor, causes the processor to perform steps including obtaining an original menu associated with a web page requested by a client computing device. The original menu comprises at least a plurality of URLs, each comprising one or more fragments, and a plurality of titles, each comprising one or more words. A unique index for one occurrence of each of at least a subset of the one or more fragments and the one or more words is generated. At least one dictionary comprising the generated indices associated with a corresponding one of the subset of the one or more fragments or the one or more words is generated. A modified menu is generated by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding one of the unique indices. The modified menu and the at least one dictionary are sent to the client computing device.

[0007] A menu management computing apparatus includes a memory and a processor coupled to the memory. The processor is configured to execute programmed instructions stored in the memory including obtaining an original menu associated with a web page requested by a client computing device. The original menu comprises at least a plurality of URLs, each comprising one or more fragments, and a plurality of titles, each comprising one or more words. A unique index for one occurrence of each of at least a subset of the one or more fragments and the one or more words is generated. At least one dictionary comprising the generated indices associated with a corresponding one of the subset of the one or more fragments or the one or more words is generated. A modified menu is generated by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding one of the unique indices. The modified menu and the at least one dictionary are sent to the client computing device.

[0008] This technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatuses that reduce the amount of resources required to send and store web page menus. With this technology, a web page menu is parsed at a menu management computing apparatus and modified to replace URL fragments and words in category and subcategory titles with unique indices. A dictionary is

generated to store each of the fragments and words and corresponding indices. The modified original menu and the dictionary, which together require less time to send and less memory to store than the original menu, are then sent to a client computing device that requested the web page having the associated original menu.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]

FIG. 1 is a block diagram of a network environment which incorporates an exemplary menu management computing apparatus;

FIG. 2 is a flowchart of an exemplary method for compressing a web page menu;

FIG. 3 is an exemplary original web page menu; and

FIG. 4 is the exemplary original web page menu of FIG. 3 modified according to the exemplary method of FIG. 2.

DETAILED DESCRIPTION OF THE INVENTION

[0010] An exemplary network environment 10 is illustrated in FIG. 1 as including an exemplary menu management computing apparatus 12. In this example, the menu management computing apparatus 12 is coupled to a plurality of client computing devices 14(1)-14(n) by communication network 16, although other types and numbers of devices, components, and elements in other topologies could be used. This technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatuses that compress web page menus and reduce the amount of resources required to send and store web page menus.

[0011] Referring more specifically to FIG. 1, the menu management computing apparatus 12 includes at least one processor or central processing unit (CPU) 18, a memory 20, and a network interface 22, which are coupled together by a bus 24 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can also be used. Generally, the menu management computing apparatus 12 processes requests for web pages and other web content received from the client computing devices 14(1)-14(n) via the communication network 16 according to the HTTP-based protocol, for example, although the menu management computing apparatus 12 can also provide other numbers and types of functions. The processor 18 in the menu management computing apparatus 12 may execute a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 18 could execute other numbers and types of programmed instruc-

tions.

[0012] The memory 20 in the menu management computing apparatus 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 18, can be used for the memory 20 in the menu management computing apparatus 12.

[0013] The network interface 22 in the menu management computing apparatus 12 is used to operatively couple and communicate between the menu management computing apparatus 12 and the client devices 14(1)-14(n) via the communication network 16, although other types and numbers of networks with other types and numbers of connections and configurations can also be used. The menu management computing apparatus 12 may be a server computing device, such as any version of Microsoft® IIS server or Apache® server, although other types of servers may be used, or a web content proxy server, for example.

[0014] The communication network 16 can include one or more networks, such as one or more local area networks (LANs) and/or wide area networks (WANs) such as the Internet. By way of example only, the communication network 16 can use TCP/IP over Ethernet and industry-standard protocols, including Hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks having their own communications protocols, can also be used.

[0015] The client computing devices 14(1)-14(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the menu management computing apparatus 12 via the communication network 16, although one or more of the client computing devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. By way of example only, the client computing devices 14(1)-14(n) can be mobile computing devices, smart phones, personal digital assistants, or computers.

[0016] In this example, each of the client computing devices 14(1)-14(n) includes at least one processor or a CPU, a memory, a network interface, a user input device, and a display, which are coupled together by a bus or other link, although one or more of client computing devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor in each of the client computing devices 14(1)-14(n) can execute a program of instructions stored in the memory of each of the client

computing devices 14(1)-14(n) for one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0017] The memory in each of the client computing devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor can be used for the memory in each of the client computing devices 14(1)-14(n). Each of the client computing devices 14(1)-14(n) can be configured to access web services and content through a web browser stored in the memory.

[0018] The network interface in each of the client computing devices 14(1)-14(n) is used to operatively couple and communicate between each of the client computing devices 14(1)-14(n) and the menu management computing apparatus 12 via the communication network 16, although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0019] The user input device in each of the client computing devices 14(1)-14(n) can be used to input selections, such as a request for a particular web page, although the user input device could be used to input other types of requests and data and interact with other elements. The user input device in each of the client computing devices 14(1)-14(n) can include a keypad, touch screen, and/or vocal input processing system, although other types and numbers of user input devices can also be used.

[0020] The display in each of the client computing devices 14(1)-14(n) can be used to show data and information to the user, such as a requested web page by way of example only. The display in each of the client computing devices 14(1)-14(n) can be an LCD, LED, or OLED display, for example, although other types and numbers of displays could be used depending on the particular type of client computing device 14(1)-14(n).

[0021] Although embodiments of the menu management computing apparatus 12 and client computing devices 14(1)-14(n) are described and illustrated herein, each of the menu management computing apparatus 12, and client computing devices 14(1)-14(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s). Furthermore, each of the devices of the embodiments may be conveniently imple-

mented using one or more general purpose computers, microprocessors, digital signal processors, and microcontrollers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0022] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0023] The examples may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0024] An exemplary method for compressing an original menu will now be described with reference to FIGS. 1-4. Referring more specifically to FIG. 2, in step 200 the menu management computing apparatus 12 receives a request for a web page, such as a hypertext transfer protocol (HTTP) request, from one of the client computing devices 14(1)-14(n). In step 202, the menu management computing apparatus 12 obtains the requested web page from one of the menu management computing apparatus 12.

[0025] In step 204, the menu management computing apparatus 12 determines whether the web page has an associated original menu. The associated original menu can be any menu, although hierarchical menus with categories and subcategories may have repeated content allowing for relatively significant compression, as described and illustrated in more detail later. In some examples, the original menu is included in the source code for the web page and, in other examples, the web page source code references a separate external file, such as a JavaScript file, which includes the original menu.

[0026] In examples in which the original menu is defined in a separate file, the initially-requested web page source code may cause a web browser of the requesting one of the client computing device 14(1)-14(n) to submit a request for the file. The menu management computing apparatus 12 can parse the web page or file obtained from the one of the menu management computing app-

paratus 12 to identify any key words or patterns representing a menu (e.g. alternating plain text titles and URL hyperlinks), for example, although other methods of identifying an original menu associated with the requested web page can also be used.

[0027] If the menu management computing apparatus 12 determines in step 204 that the requested web page does not have an associated original menu, then the No branch is taken to step 200 and another request is received from one of the client computing devices 14(1)-14(n). If the menu management computing apparatus 12 determines in step 204 that the requested web page does have an associated original menu, then the Yes branch is taken to step 206.

[0028] Optionally, in step 206, the menu management computing apparatus 12 determines whether the requesting one of the client computing devices 14(1)-14(n) is a device having relatively limited storage capabilities, referred to herein as a mobile computing device. The menu management computing apparatus 12 can determine whether the requesting one of the client computing devices 14(1)-14(n) is a mobile computing device based on a user agent header included in one or more of the packets associated with the HTTP request received in step 200, for example, although other methods of identifying mobile computing devices can also be used.

[0029] If the menu management computing apparatus 12 determines that the requesting one of the client computing devices 14(1)-14(n) is not a mobile computing device, then the No branch is taken to step 200 and another request is received from one of the client computing devices 14(1)-14(n). Accordingly, in some examples, steps 208-220 are only performed for mobile computing devices which have relatively limited storage capabilities and can therefore benefit most from the compression method described and illustrated herein.

[0030] If the menu management computing apparatus 12 determines in step 206 that the requesting one of the client computing devices 14(1)-14(n) is a mobile computing device, then the Yes branch is taken to step 208. In step 208, the menu management computing apparatus 12 parses the original menu and generates a unique index for at least a subset of the fragments and/or words of the original menu. The original menu can include a plurality of titles, each including one or more words, and a plurality of URLs, each including one or more fragments. Exemplary URL fragments include an entire URL, a prefix, a path, one or more words of a path, a query, a key/value pair of a query, a key of a query, a value of a query, or combinations thereof, for example, although other URL fragments can also be used.

[0031] Referring to FIG. 3, an exemplary original menu 300 associated with a web page is illustrated. In this example, the original menu 300 defines a hierarchical menu with categories having titles of "Shoes," "Accessories," and "Dresses." The "Shoes" and "Dresses" categories each have two sub-categories with titles of "Men Shoes" and "Women Shoes" and "Long Dresses" and "Ceremo-

nies," respectively. In this example, each of the "Accessories" category and the four subcategories has an associated URL of a web page having content associated with the respective category or subcategory title.

[0032] Referring back to FIG. 2, in step 208, only one index is generated for multiple occurrences of the same fragment and/or word in the original menu 300. In this example, only one index is generated for the title word "Shoes" in the source code 300 even though the word appears in more than one title. Additionally, only one index is generated for the key/value pair fragment "sort=price" even though the fragment appears in all of the URLs. The menu management computing apparatus 12 can be configured to parse the original menu 300 to identify title words and URL fragments based on an established configuration in order to generate the indices.

[0033] The menu management computing apparatus 12 parses some or all of the titles of the original menu 300 and generates an index for some or all words or combination of words in each of the titles. Words can be any sequence of characters and numbers separated by a space (e.g. "Men" and "Shoes"). In one exemplary configuration, the menu management computing apparatus 12 generates an index for every encountered word in a title. Words included in titles may be repeated as the title in a lower category in a hierarchical menu is likely to share word(s) with the title in a higher category, while adding word(s) to be more specific.

[0034] The menu management computing apparatus 12 also parses some or all of the URLs of the original menu 300 and generates an index for some or all fragments or combination of fragments in each of the URLs. In one exemplary configuration, the menu management computing apparatus 12 only generates an index upon encountering a URL fragment more than once or a predetermined number of times, although indices can also be generated by the menu management computing apparatus 12 upon initially encountering a fragment.

[0035] In the exemplary original menu 300 illustrated in FIG. 3, in this example, the menu management computing apparatus 12 will only generate a unique index for the multiple occurrences of the "sort=price" key/value pair fragment upon encountering it a second time in the URL associated with the "Women Shoes" subcategory. The menu management computing apparatus 12 can parse the original menu 300 and maintain a list of previously encountered fragments and/or words which is used to determine whether an associated index should be, or has previously been, generated.

[0036] In another example, the menu management computing apparatus 12 can be configured to parse each URL to identify the prefix (e.g. "http://www.acme.com/catalog"), path (e.g. "Men_Shoes/products"), and/or query fragments (e.g. "plp_i=2354&sort=price") that are likely to be repeated in other URLs of the source code 300. Although one example is illustrated and described here, other methods and configurations can be used to parse the URLs and titles of the original menu

300.

[0037] Optionally, the generated indices include one or more digits in base 62 format with each digit being an ASCII lower case letter, upper case letter, or number from 0-9. In one example, the menu management computing apparatus 12 uses two base 62 digits which allows the menu management computing apparatus 12 to represent 620 or 3844 numbers, although other numbers of digits and bases can also be used.

[0038] In step 210, the menu management computing apparatus 12 generates at least one dictionary including the generated indices associated with a corresponding one of the fragments or words or combinations of fragments or words. In this example, the menu management computing apparatus 12 generates a words dictionary for storing the generated unique indices for occurrences of a word, URL, path, or word of a path, and a parameters dictionary for storing the generated unique indices for occurrences of a query, key/value pair of a query, key of a query, or value of a query for example. Other numbers of dictionaries storing other fragments and/or words can also be used.

[0039] An exemplary words dictionary generated in step 210 for the source code 300 is illustrated in Table 1.

Table 1

Index	Fragment/Word
00	products
01	02/ 00
02	Accessories
03	Shoes
04	Men
05	Women
06	price
07	Dresses
08	Long
09	Ceremonies

[0040] An exemplary parameters dictionary for the source code 300 is illustrated in Table 2.

Table 2

Index	Fragment/Word
00	sort=price
01	plp_i
02	sort

[0041] Optionally, in step 212, the menu management computing apparatus 12 compresses at least a portion of one or more of the dictionaries. Also optionally, the

menu management computing apparatus 12 can determine whether the requesting one of the client computing devices 14(1)-14(n) is a mobile computing device, as described and illustrated earlier in step 206, and only compress at least a portion of one or more of the dictionaries when it is determined that the requesting one of the client computing devices 14(1)-14(n) is a mobile computing device. Accordingly, in some examples, step 212 is only performed for mobile computing devices which have relatively limited storage capabilities and can therefore benefit most from a second level of compression as described and illustrated herein.

[0042] In the exemplary original menu 300 illustrated in FIG. 3, in one example, the words dictionary includes an entry of "01:Accessories/Products" which establishes an association of the base 62 number "01" with the "Accessories/Products" path fragment. In this example, the menu management computing apparatus 12 can compress the dictionary by replacing "Accessories/Products" in the dictionary entry associated with the "01" index with indices generated in step 208 for portions of the fragment (e.g. "Accessories" and "Products"). In this example, "Accessories" is associated with the 02 index and "Products" is associated with the 00 index in the dictionary. Accordingly, the menu management computing apparatus 12 can replace the "01:Accessories/Products" entry using the indices generated in step 208 (e.g. "00:02/ 00").

[0043] Alternatively, in examples in which indices were not generated for portions of a fragment in step 208, the menu management computing apparatus 12 can generate indices in step 212, store the generated indices and associations in the dictionary, and replace the fragment as described and illustrated earlier. In some examples, the menu management computing apparatus 12 can compress the dictionary for fragments and/or words having a threshold number of characters or a threshold number of portions shared with other fragments and/or words in the dictionary. By establishing a threshold, the increased time required to compress the dictionary can be balanced with the reduced size of the compressed dictionary. Other methods of compressing the dictionary can also be used.

[0044] In step 214, the menu management computing apparatus 12 generates a modified menu by replacing occurrence(s) of the fragments and/or words in the original menu 300 with a corresponding one of the indices generated in step 208. In one example, the modifications to the original menu 300 occur during an initial parsing of the original menu 300 and while the dictionary is being generated by the menu management computing apparatus 12. In another example, the menu management computing apparatus 12 parses the original menu 300 a second time, subsequent to generating the dictionary, and uses the dictionary to determine which fragments and/or words are to be replaced.

[0045] Optionally, the fragments and/or words in the original menu 300 are replaced by the longest (e.g. largest number of successive characters) corresponding

fragment and/or word in the dictionary. For example, the path fragment (e.g. "Men_Shoes/products") of a URL associated with a subcategory is repeated several times in the original menu 300. Based on the configuration of the menu management computing apparatus 12, the path fragment may be included in the dictionary associated with an index. In this example, occurrences of the path of the URL are replaced by the corresponding index even when indices were generated, in step 208, and associated in the dictionary, in step 210, with one or more words of the path (e.g. "Men," "Shoes," or "products"). In this example, the dictionary entry for the path fragment can optionally be compressed using the entries of the words of the path, as described and illustrated earlier with reference to step 212, and path fragment in the original menu 300 will be replaced with a reduced number of indices.

[0046] Referring to FIG. 4, an exemplary modified menu 400 generated in step 214 is illustrated. In this example, the modified menu 400 includes "0403" in place of the "Men Shoes" subcategory title included in the original menu 300. As indicated in the exemplary words dictionary shown in Table 1, the base 62 index "04" was generated by the menu management computing apparatus 12, in step 208, and associated with the word "Men" in the dictionary, in step 210. Similarly, the base 62 index "03" was generated by the menu management computing apparatus 12, in step 208, and associated with the word "Shoes" in the dictionary, in step 210. Other indices and dictionary entries can also be used.

[0047] The source code 400 further includes "0403/00?01&2354&00" in place of the "http://www.acme.com/catalog/Men_Shoes/products?plp_i=2354&sort=price" URL associated with the "Men Shoes" subcategory. The "Men_Shoes" portion of the path in the URL associated with the "Men Shoes" subcategory is replaced with "0403" which are the indices associated with the "Men" and "Shoes" words, respectively, in the words dictionary. Additionally, the "products" portion of the path and "plp_i" key fragment in the URL associated with the "Men Shoes" subcategory are replaced with "00" and "01," respectively, which are the indices associated with the "products" portion of the path and the "plp_i" key fragment in the words and parameters dictionaries, respectively. Finally, the "sort=price" key/value pair fragment in the URL associated with the "Men Shoes" subcategory is replaced with "00," which is the index associated with the "sort=price" key/value pair fragment in the parameters dictionary. The other titles and URLs in the source code 300 are similarly modified as illustrated in the modified menu 400.

[0048] In step 216, the menu management computing apparatus 12 optionally applies one or more rules to remove and/or insert spaces and/or special characters in the modified menu 400, for example, so that the titles and URLs can be properly interpreted and decompressed at the requesting one of the client computing device 14(1)-14(n). In one example in which all words in

titles are replaced by a corresponding two digit base 62 index generated in step 208, the spaces or other standard separator in the titles can be removed. In this example, every two digits in a title represents a word of the title so the separators do not need to be included and can be removed to reduce the size of the modified menu 300. In the modified menu 400, the space between indices that replaced the words in the "Men Shoes," "Women Shoes," and "Long Dresses" titles is removed.

[0049] In examples in which some fragments of a URL (e.g. words of a path) are not be replaced by an index, the menu management computing apparatus 12 can insert a space before and/or after any set of two or more sequential indices in the URL. By inserting a space, which is not a valid character inside a URL, any valid characters and/or fragments not replaced can be differentiated from any indices that replaced fragments included in the original menu 300. Other invalid characters can also be used to differentiate indices that replaced fragments in the original menu 300.

[0050] Additionally, in examples in which all key/value pair fragments are converted, a "&" or other special character separating sequential key/value pairs can be removed by the menu management computing apparatus 12. Instead, in these examples, the "&" character can be inserted by the menu management computing apparatus 12 before and/or after any value fragment of a query fragment for any of the URLs. By inserting the "&" character(s), the value can be differentiated from any indices that replaced fragments included in the original menu 300.

[0051] In the modified menu 400, the value fragment "2354" of the URL associated with the "Men Shoes" subcategory is preceded and followed by a "&" character, which differentiates the "2354" value fragment from the "01" and "00" indices that precede and follow, respectively, the "2354" value fragment. Other special characters can also be used to differentiate indices that replaced fragments in the original menu 300.

[0052] Because all of the URLs have the same prefix fragment of "http://www.acme.com/catalog" in the original menu 300, the prefix fragment is removed and replaced by a space in the modified menu 400. The prefix fragment could also be replaced by an index value or a special character, for example. Additionally, the "_" special character can be removed from any of the URLs since two indices (e.g. "0403") representing two words of a path fragment are always separated by the "_" character as a space is an invalid character in a URL and the words would be represented by only one index if there were no "_" character between them.

[0053] Other characters (e.g. "/" and "?") are not removed from the URLs in this example since they are valid characters and are not digits in the base 62 format used for the indices. Other rules for modifying titles and/or URL fragments with respect to spaces and special characters can also be used.

[0054] Referring back to FIG. 2, in step 218, the menu

management computing apparatus 12 sends the modified menu 400 and the dictionary to the requesting one of the client computing devices 14(1)-14(n). In step 220, the menu management computing apparatus 12 sends decompression source code to the requesting one of the client computing devices 14(1)-14(n). The decompression source code, when executed by the requesting one of the client computing devices 14(1)-14(n), converts the modified menu 400 into the original menu 300 using the dictionaries generated in step 210, and sent to the requesting one of the client computing devices 14(1)-14(n) in step 218, and based on the rules applied in step 216.

[0055] Accordingly, the decompression source code, sent by the menu management computing apparatus 12 with the dictionaries and the modified menu 400, is configured to parse the modified menu 400, replace encountered indices with corresponding URL fragments and/or title words in the dictionaries, and reverse the results of the rules applied in step 216. For example, the decompression source code can be configured to inset a space following each word in a title that is converted. In another example, a "_" special character in a path fragment of a URL is inserted by the menu management computing apparatus 12 following any each word of the path that is converted.

[0056] The decompression source code can be configured to reverse the result of applying one or more other rules and/or perform other functions in order to convert the modified menu 400 into the original menu 300. Additionally, the decompression source code can be stored in the memory 20 of the menu management computing apparatus 12 and sent to a requesting one of the client computing devices 14(1)-14(n) following, or along with, the modified menu 400 and the dictionaries.

[0057] Accordingly, as illustrated and described herein this technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatuses that reduce the storage requirements for, and time required to retrieve, web page menus. With this technology, a web page menu is compressed using at least one dictionary to store indices corresponding to word(s) in menu titles and/or fragment(s) in URLs of the menu. The compressed menu is sent to a requesting client computing device along with the dictionaries and decompression source code configured to convert the modified menu into the original menu. Advantageously, the modified menu, dictionaries, and decompression source code can be sent in less time and are smaller in size than the original menu.

[0058] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the inven-

tion. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for compressing menus, the method comprising:

obtaining, with a menu management computing apparatus, an original menu associated with a web page requested by a client computing device, the original menu comprising at least a plurality of uniform resource locators (URLs), each comprising one or more fragments, and a plurality of titles, each comprising one or more words;

generating, with the menu management computing apparatus, a unique index for one occurrence of each of at least a subset of the one or more fragments and the one or more words;

generating, with the menu management computing apparatus, at least one dictionary comprising the generated indices associated with a corresponding one of the subset of the one or more fragments or the one or more words;

generating, with the menu management computing apparatus, a modified menu by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding one of the unique indices; and
sending, with the menu management computing apparatus, the modified menu and the at least one dictionary to the client computing device.

2. The method of claim 1, wherein the one or more fragments comprise one or more of a URL, prefix, path, word of a path, query, key/value pair of a query, key of a query, or value of a query.

3. The method of claim 1, wherein the at least one dictionary comprises a words dictionary for storing the generated unique indices for occurrences of a word, URL, prefix, path, or word of a path and a parameters dictionary for storing the generated unique indices for occurrences of a query, key/value pair of a query, key of a query, or value of a query.

4. The method of claim 1, further comprising:

compressing, with the menu management computing apparatus, at least a portion of the at least one dictionary; and

sending, with the menu management computing apparatus, to the client computing device, source code configured to convert the modified source code into the original source code using the at least one dictionary.

- 5. The method of claim 1, wherein the subset of the one or more fragments includes only one or more of the fragments occurring more than once in the plurality of URLs.
- 6. The method of claim 1, wherein the unique indices comprise one or more digits in base 62 format.
- 7. The method of claim 1, wherein the modifying further comprises:

- removing any separator between words for any of the plurality of titles comprising two or more words;
- inserting a space before or after any two or more sequential indices for any of the URLs; and
- inserting a special character before or after any value of a query for any of the URLs.

- 8. The method of claim 1, further comprising:

- determining, with the menu management computing apparatus, whether the client computing device is a mobile computing device; and
- only performing the generating and sending steps when it is determined that the client computing device is a mobile computing device.

- 9. A non-transitory computer readable medium having stored thereon instructions for compressing menus comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:

- obtaining an original menu associated with a web page requested by a client computing device, the original menu comprising at least a plurality of uniform resource locators (URLs), each comprising one or more fragments, and a plurality of titles, each comprising one or more words;
- generating a unique index for one occurrence of each of at least a subset of the one or more fragments and the one or more words;
- generating at least one dictionary comprising the generated indices associated with a corresponding one of the subset of the one or more fragments or the one or more words;
- generating a modified menu by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding

- one of the unique indices; and
- sending the modified menu and the at least one dictionary to the client computing device.

- 10. The medium of claim 9, wherein the one or more fragments comprise one or more of a URL, prefix, path, word of a path, query, key/value pair of a query, key of a query, or value of a query.

- 11. The medium of claim 9, wherein the at least one dictionary comprises a words dictionary for storing the generated unique indices for occurrences of a word, URL, prefix, path, or word of a path and a parameters dictionary for storing the generated unique indices for occurrences of a query, key/value pair of a query, key of a query, or value of a query.

- 12. The medium of claim 9, further comprising machine executable code which, when executed by the processor, causes the processor to perform steps further comprising:

- compressing at least a portion of the at least one dictionary; and
- sending to the client computing device source code configured to convert the modified source code into the original source code using the at least one dictionary.

- 13. The medium of claim 9, wherein the subset of the one or more fragments includes only one or more of the fragments occurring more than once in the plurality of URLs.

- 14. The medium of claim 9, wherein the unique indices comprise one or more digits in base 62 format.

- 15. The medium of claim 9, wherein the modifying further comprises:

- removing any separator between words for any of the plurality of titles comprising two or more words;
- inserting a space before or after any two or more sequential indices for any of the URLs; and
- inserting a special character before or after any value of a query for any of the URLs.

- 16. The medium of claim 9, further comprising machine executable code which, when executed by the processor, causes the processor to perform steps further comprising:

- determining whether the client computing device is a mobile computing device; and
- only performing the generating and sending steps when it is determined that the client computing device is a mobile computing device.

17. A menu management computing apparatus, comprising:

a memory; and
a processor coupled to the memory and configured to execute programmed instructions stored in the memory, comprising:

obtaining an original menu associated with a web page requested by a client computing device, the original menu comprising at least a plurality of uniform resource locators (URLs), each comprising one or more fragments, and a plurality of titles, each comprising one or more words;
generating a unique index for one occurrence of each of at least a subset of the one or more fragments and the one or more words;
generating at least one dictionary comprising the generated indices associated with a corresponding one of the subset of the one or more fragments or the one or more words;
generating a modified menu by replacing each occurrence of each of the at least a subset of the one or more fragments and the one or more words of the original menu with a corresponding one of the unique indices; and
sending the modified menu and the at least one dictionary to the client computing device.

18. The apparatus of claim 17, wherein the one or more fragments comprise one or more of a URL, prefix, path, word of a path, query, key/value pair of a query, key of a query, or value of a query.

19. The apparatus of claim 17 wherein the at least one dictionary comprises a words dictionary for storing the generated unique indices for occurrences of a word, URL, prefix, path, or word of a path and a parameters dictionary for storing the generated unique indices for occurrences of a query, key/value pair of a query, key of a query, or value of a query.

20. The apparatus of claim 17, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising:

compressing at least a portion of the at least one dictionary; and
sending to the client computing device source code configured to convert the modified source code into the original source code using the at least one dictionary.

21. The apparatus of claim 17, wherein the subset of the one or more fragments includes only one or more of the fragments occurring more than once in the plurality of URLs.

22. The apparatus of claim 17 wherein the unique indices comprise one or more digits in base 62 format.

23. The apparatus of claim 17, wherein the modifying further comprises:

removing any separator between words for any of the plurality of titles comprising two or more words;
inserting a space before or after any two or more sequential indices for any of the URLs; and
inserting a special character before or after any value of a query for any of the URLs.

24. The apparatus of claim 17, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising:

determining whether the client computing device is a mobile computing device; and
only performing the generating and sending steps when it is determined that the client computing device is a mobile computing device.

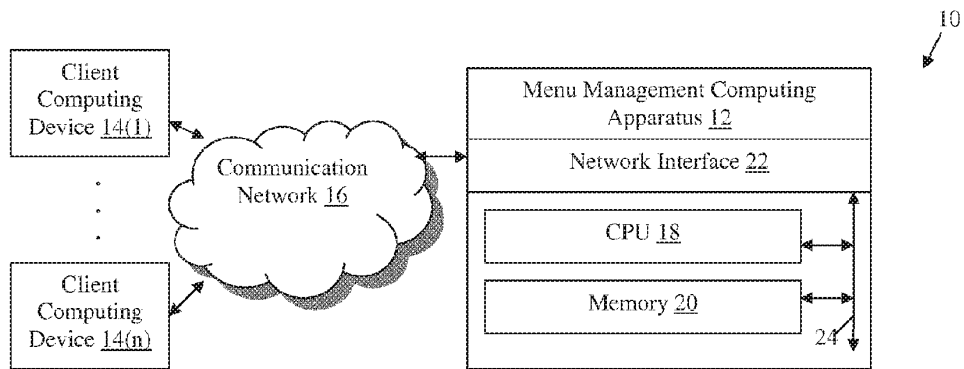


FIG. 1

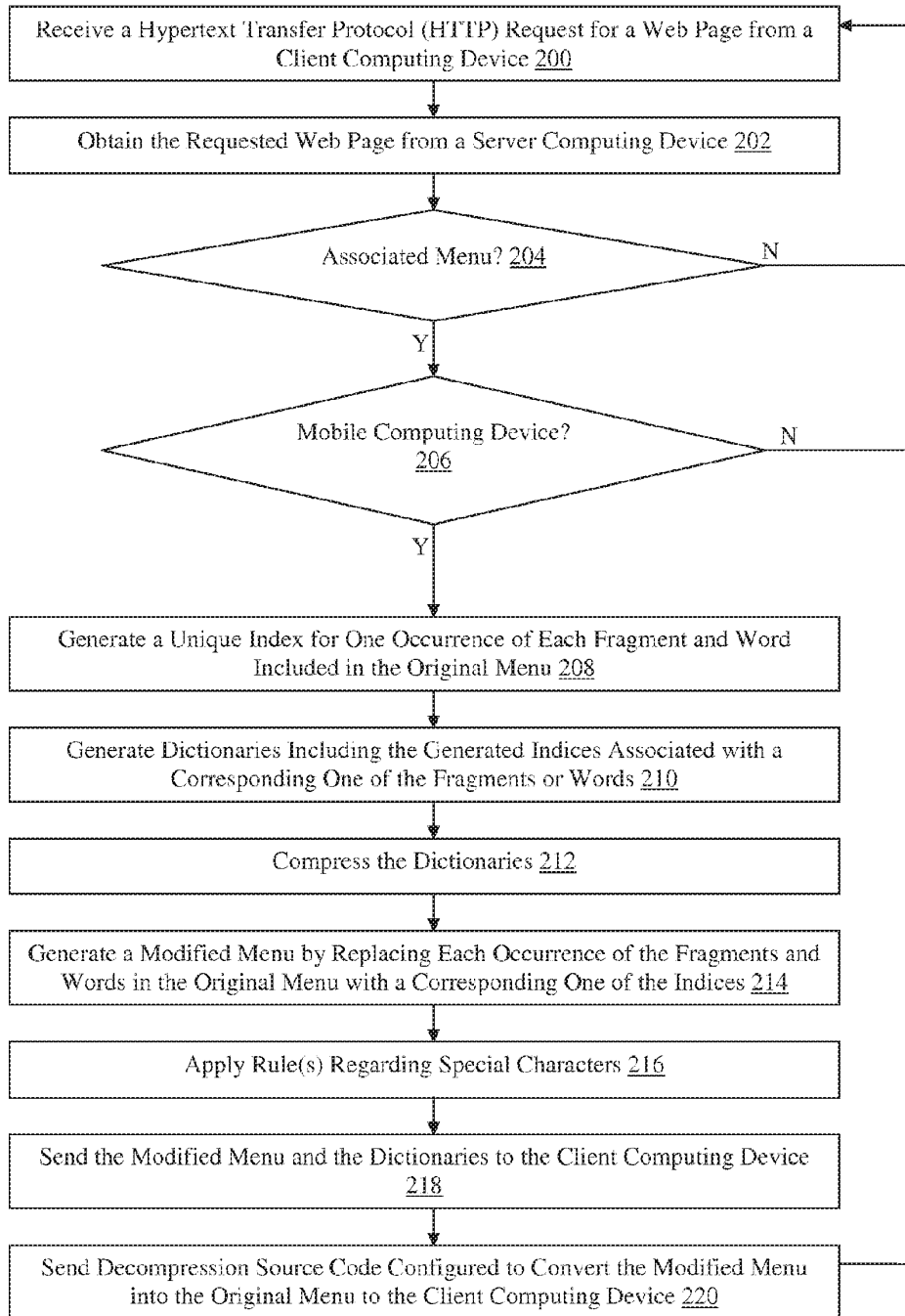


FIG. 2

```
[
  "Shoes",
  [
    "Men Shoes",
    "http://www.acme.com/catalog/Men_Shoes/products?plp_i=2354&sort=price",
    "Women Shoes",
    "http://www.acme.com/catalog/Women_Shoes/products?plp_i=2358&sort=price"
  ],
  "Accessories",
  "http://www.acme.com/catalog/Accessories/products?plp_i=1348&sort=price",
  "Dresses",
  [
    "Long Dresses",
    "http://www.acme.com/catalog/Accessories/products?plp_i=4242&sort=price",
    "Ceremonies",
    "http://www.acme.com/catalog/Ceremonies/products?plp_i=7718&sort=price"
  ]
]
```

FIG. 3

```

{
  "longWords":3,
  "longParams":1,
  "ch":"_",
  "baseUrl":"http://www.acme.com/catalog/",
  "params":["02& 06&","plp_i","sort"],
  "wl":2,
  "words":["products"," 02/ 00","Accessories","Shoes","Men",
    "Women","price","Dresses","Long","Ceremonies"],
  "tree":[
    "03",
    [
      "0403",
      " 0403/ 00701&2354&00",
      "0503",
      " 0503/ 00701&2358&00"
    ],
    "02",
    " 02/ 00701&1348&00",
    "07",
    [
      "0807",
      " 02/ 00701&4242&00",
      "09",
      " 09/ 00701&7718&00"
    ]
  ]
}

```

↙400

FIG. 4



EUROPEAN SEARCH REPORT

Application Number
EP 14 15 9441

5

10

15

20

25

30

35

40

45

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	Paul Sandoz, Alessandro Triglia, Santiago Pericas-Geertsen, : "Fast Infoset", 30 June 2004 (2004-06-30), XP055123030, Retrieved from the Internet: URL:http://www.oracle.com/technetwork/arti cles/javase/fastinfoset-139262.html [retrieved on 2014-06-12] * Introduction, "The Fast Infoset Standard and ASN.1", "Advantages and Properties of teh Fast Infoset", Table 1, "Tables and Indexing", "Mandatory and Optional Constrained Indexing", "Examples of Indexing" * -----	1-24	INV. G06F17/30 H04L29/08
			TECHNICAL FIELDS SEARCHED (IPC)
			G06F H04L H03M
The present search report has been drawn up for all claims			
Place of search		Date of completion of the search	Examiner
The Hague		27 June 2014	Targon, Valerio
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EPO FORM 1503 (03.02) (Pst001)

55



(11) **EP 2 778 976 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
27.12.2017 Bulletin 2017/52

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

(21) Application number: **14159461.4**

(22) Date of filing: **13.03.2014**

(54) **Methods for processing cascading style sheets and devices thereof**

Verfahren zur Verarbeitung von Kaskadenstilentwürfen und Vorrichtungen damit

Procédés de traitement de feuilles de style en cascade et dispositifs associés

(84) Designated Contracting States:
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR**

(30) Priority: **13.03.2013 US 201313798844**

(43) Date of publication of application:
17.09.2014 Bulletin 2014/38

(73) Proprietor: **Usablenet Inc.**
New York, NY 10019 (US)

(72) Inventor: **Scoda, Enrico**
33035 Martignacco (IT)

(74) Representative: **Gervasi, Gemma et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(56) References cited:
WO-A1-2005/003998 US-B1- 6 589 291
US-B1- 7 134 073 US-B1- 7 574 486

EP 2 778 976 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description**FIELD**

[0001] This technology generally relates to methods, non-transitory computer readable medium, and apparatuses for processing cascading style sheets (CSSs) included in web pages and, more particularly, for reducing the amount of code required to define CSSs.

BACKGROUND

[0002] Cascading style sheets (CSSs) are used by software developers to define the format, appearance, and/or layout of web pages or to define media associated with web pages, such as animations. The interpretation of the properties of CSS code is browser-specific. For compatibility, developers often include multiples of each property and the rendering web browser utilizes only the properties it recognizes. An exemplary CSS code fragment 300 is illustrated in FIG. 3 as including six CSS properties 302(1)-302(6). The CSS properties 302(1)-302(4) are used by WebKit-based web browsers, the CSS property 302(5) is used by Mozilla Firefox™ web browsers, and the CSS property 302(6) is used by standards-compliant web browsers to define the "rounded_border" class attribute. US patent number 6589291 discloses a method, system, and computer-readable code for dynamically determining the most appropriate location for applying style sheets. US patent number 7574486 discloses a system, a method, and a computer readable medium for reformatting web content into a format readable on one or more mobile devices. US patent number 7134073 discloses an apparatus and a method for enabling composite style sheet application to multi-part electronic documents.

[0003] Due to the multiple properties, defining CSSs can require a large amount of code, which may be complex and/or difficult to maintain. Communicating and rendering web pages with CSSs can also require a relatively large amount of resources and time. Additionally, the implementations required for compatibility are exposed to the client computing devices, which may not be desirable.

SUMMARY

[0004] A method for processing cascading style sheets includes receiving at a web content proxy server a Hypertext Transfer Protocol (HTTP) request for a web page from a client computing device. A type of browser from which the HTTP request originated is identified with the web content proxy server based on a user agent header included in the HTTP request. At least one cascading style sheet (CSS) associated with the requested web page is obtained with the web content proxy server. The at least one CSS including a CSS property comprising at least a prefix followed by a name. A determination is made whether the prefix of the CSS property matches

an established value with the web content proxy server. A transformation function for the CSS property is applied with the web content proxy server, when it is determined that the prefix of the CSS property matches the established value. The transformation function is configured to modify the web page to replace the CSS property based on the identified type of browser and the name of the CSS property. The modified web page is sent with the web content proxy server to the requesting client computing device.

[0005] A non-transitory computer readable medium having stored thereon instructions for processing cascading style sheets including machine executable code which when executed by a processor, causes the processor to perform steps including receiving an HTTP request for a web page from a client computing device. A type of browser from which the HTTP request originated is identified based on a user agent header included in the HTTP request. At least one cascading style sheet (CSS) associated with the requested web page is obtained. The at least one CSS including a CSS property comprising at least a prefix followed by a name. A determination is made whether the prefix of the CSS property matches an established value. A transformation function for the CSS property is applied, when it is determined that the prefix of the CSS property matches the established value. The transformation function is configured to modify the web page to replace the CSS property based on the identified type of browser and the name of the CSS property. The modified web page is sent to the requesting client computing device.

[0006] A web content proxy server apparatus includes a memory coupled to a processor which is configured to execute programmed instructions stored in the memory comprising receiving an HTTP request for a web page from a client computing device. A type of browser from which the HTTP request originated is identified based on a user agent header included in the HTTP request. At least one cascading style sheet (CSS) associated with the requested web page is obtained. The at least one CSS including a CSS property comprising at least a prefix followed by a name. A determination is made whether the prefix of the CSS property matches an established value. A transformation function for the CSS property is applied, when it is determined that the prefix of the CSS property matches the established value. The transformation function is configured to modify the web page to replace the CSS property based on the identified type of browser and the name of the CSS property. The modified web page is sent to the requesting client computing device.

[0007] This technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatuses that reduce the amount and complexity of code required to define web page CSSs. With this technology, developers can include a prefix to indicate that a CSS property should be transformed server-side to be compatible with a web browser

of a client computing device that originated the request for the web page with the CSS. Accordingly, CSS code can be optimized and web pages with CSSs can be communicated relatively quickly to requesting client computing devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008]

FIG. 1 is a block diagram of a network environment which incorporates an exemplary web content proxy server;

FIG. 2 is a flowchart of an exemplary method for processing a cascading style sheet (CSS) included in a web page;

FIG. 3 is an exemplary cascading style sheet (CSS) code fragment defining a class attribute for a plurality of web browsers;

FIG. 4 is an exemplary CSS code fragment defining the class attribute of FIG. 3 and including a CSS property with an exemplary prefix indicating that the property requires transformation;

FIG. 5 is an exemplary CSS code fragment defining the class attribute of FIG. 3 and including the CSS property of FIG. 4 transformed for a standards-compliant web browser;

FIG. 6 is an exemplary CSS code fragment defining the class attribute of FIG. 3 and including the CSS property of FIG. 4 transformed for a Mozilla Firefox™ web browser; and

FIG. 7 is an exemplary CSS code fragment defining the class attribute of FIG. 3 and including the CSS property of FIG. 4 transformed for a WebKit-based web browser.

DETAILED DESCRIPTION

[0009] An exemplary network environment 10 is illustrated in FIG. 1 as including an exemplary web content proxy server 12. In this example, the web content proxy server 12 is coupled to a plurality of server computing devices 14(1)-14(n) and a plurality of client computing devices 16(1)-16(n) by communication networks 18(1) and 18(2), although other types and numbers of devices, components, and elements in other topologies could be used. This technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatuses for more efficiently processing cascading style sheets (CSSs) of a web page based on the web browser originating the request for the web page and without requiring that the web page include multiple

properties of the CSS for compatibility.

[0010] Referring more specifically to FIG. 1, the web content proxy server 12 includes at least one processor or central processing unit (CPU) 20, a memory 22, and a network interface 24, which are coupled together by a bus 26 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can also be used. The processor 20 in the web content proxy server 12 may execute a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 20 could execute other numbers and types of programmed instructions.

[0011] The memory 22 in the web content proxy server 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 20, can be used for the memory 22 in the web content proxy server 12.

[0012] The network interface 24 in the web content proxy server 12 is used to operatively couple and communicate between the web content proxy server 12 and the client devices 16(1)-16(n) and server devices 14(1)-14(n) via the communication networks 24(1) and 24(2), although other types and numbers of networks with other types and numbers of connections and configurations can also be used. For example, one or more of the communication networks 24(1) and 24(2) can include one or more networks, such as one or more local area networks (LANs) and/or wide area networks (WANs) such as the Internet. By way of example only, the communication networks can use TCP/IP over Ethernet and industry-standard protocols, including Hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks having their own communications protocols, can also be used.

[0013] Generally, the server computing devices 14(1)-14(n) process requests for web pages and other web content received from the client computing devices 16(1)-16(n) via the communication networks 18(1) and 18(2) according to the HTTP-based protocol, for example, although the server computing devices 14(1)-14(n) can also provide other numbers and types of functions. Each of the server computing devices 14(1)-14(n) can include a at least one processor or CPU, a memory, and a network interface, which are coupled together by a bus or other link, although each of the server computing devices 14(1)-14(n) could have other numbers and types of components, parts, devices, systems, and elements in other

configurations. The processor in each of the server computing devices 14(1)-14(n) executes a program of stored instructions for one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[0014] The memory in each of the server computing devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor, can be used for the memory in each of the server computing devices 14(1)-14(n).

[0015] The network interface in each of the server computing devices 14(1)-14(n) is used to operatively couple and communicate between the server computing devices 14(1)-14(n) and the web content proxy server 12 and client computing devices 16(1)-16(n) via the communication networks 24(1) and 24(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. Each of the server computing devices 16(1)-16(n) may be hardware or software or may represent a system with multiple servers in a pool, which may include internal or external networks. In this example the each of the server computing devices 16(1)-16(n) may be any version of Microsoft® IIS server or Apache® server, although other types of servers may be used.

[0016] The client computing devices 12(1)-12(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the server computing devices 14(1)-14(n) through the web content proxy server 12 via the communication networks 24(1) and 24(2), although one or more of the client computing devices 12(1)-12(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. By way of example only, the client computing devices 12(1)-12(n) can be smart phones, personal digital assistants, or computers.

[0017] In this example, each of the client computing devices 16(1)-16(n) includes at least one processor or a CPU, a memory, a network interface, a user input device, and a display, which are coupled together by a bus or other link, although one or more of client computing devices 16(1)-16(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor in each of the client computing devices 16(1)-16(n) can execute a program of instructions stored in the memory of each of the client computing devices 16(1)-16(n) for one or more aspects of the present invention as described and illustrated here-

in, although the processor could execute other numbers and types of programmed instructions.

[0018] The memory in each of the client computing devices 16(1)-16(n) stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM or a ROM in the system or a floppy disk, hard disk, CD ROM, or other non-transitory computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor can be used for the memory in each of the client computing devices 16(1)-16(n). Each of the client computing devices 16(1)-16(n) can be configured to access web services and content through a web browser stored in the memory.

[0019] The network interface in each of the client computing devices 16(1)-16(n) is used to operatively couple and communicate between each of the client computing devices 16(1)-16(n) and the web content proxy server 12 and server computing devices 12(1)-12(n) via the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0020] The user input device in each of the client computing devices 16(1)-16(n) can be used to input selections, such as a request for a particular web page, although the user input device could be used to input other types of requests and data and interact with other elements. The user input device in each of the client computing devices 16(1)-16(n) can include a keypad, touch screen, and/or vocal input processing system, although other types and numbers of user input devices can also be used.

[0021] The display in each of the client computing devices 16(1)-16(n) can be used to show data and information to the user, such as a requested web page by way of example only. The display in each of the client computing devices 16(1)-16(n) can be an LCD, LED, or OLED display, for example, although other types and numbers of displays could be used depending on the particular type of client computing device 16(1)-16(n).

[0022] Although embodiments of the web content proxy server 12, client computing devices 16(1)-16(n), and server computing devices 14(1)-14(n) are described and illustrated herein, each of the web content proxy server 12, client computing devices 16(1)-16(n), and server computing devices 14(1)-14(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s). Furthermore, each of the devices of the embodiments may be conveniently implemented using

one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0023] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0024] The examples may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0025] An exemplary method for processing cascading style sheets (CSSs) included in a web page will now be described with reference to FIGS. 1-7. Referring specifically to FIG. 2, in step 200, the web content proxy server 12 receives a Hypertext Transfer Protocol (HTTP) request for a web page from one of the client computing devices 16(1)-16(n) via the communication network 18(1).

[0026] In step 202, the web content proxy server 12 identifies a type of web browser from which the HTTP request received in step 200 originated. In this example, the HTTP request can include a user agent header which includes information regarding the originating web browser including at least the type of the web browser. In step 204, the web content proxy server 12 retrieves the requested web page from one of the server computing devices 14(1)-14(n), although the web page could be retrieved from other types of computing devices.

[0027] In step 206, the web content proxy server 12 determines whether the retrieved web page has an associated CSS. In this example the CSS is included in the source code for the web page. However, in other examples, the web page source code references a separate external file which includes a CSS. In examples in which a CSS is defined in a separate file, the initially-requested web page source code may cause a web browser of the requesting one of the client computing device 16(1)-16(n) to submit a request for the file. The web content proxy

server 12 can parse the web page or the file retrieved from the one of the server computing devices 14(1)-14(n) to identify character strings associated with a CSS included therein, for example, although other methods of determining whether the web page has an associated CSS can also be used.

[0028] If the web content proxy server 12 determines that the retrieved web page does not have an associated CSS, then the No branch is taken to step 208. In step 208, the retrieved web page is sent to the requesting one of the client computing devices 16(1)-16(n). Optionally, the web content proxy server 12 can perform other optimizations on the web page prior to forwarding the web page to the requesting one of the client computing devices 16(1)-16(n), such as optimizing the web page for display on a mobile computing device, for example.

[0029] Referring back to step 206, if the web content proxy server 12 determines that the retrieved web page does have an associated CSS, then the Yes branch is taken to step 210. In step 210, the web content proxy server 12 determines whether a prefix of one of the CSS properties of the CSS matches an established value. The established value can be set by an administrator and stored in the memory 22 of the web content proxy server 12, for example. The established value can also be shared with web page developers. If the web content proxy server 12 determines that the prefix of the CSS property does match the established value, then the Yes branch is taken to step 212.

[0030] Referring to FIG. 4, an exemplary CSS code fragment 400 defining the "rounded_border" class attribute of FIG. 3 is illustrated. In this example, the CSS code fragment 400 includes a CSS property 402 with a "-u-" prefix. In this example, the "-u-" prefix matches the established value, although the established value can be any value that does not match a prefix used by a web browser (e.g. "-webkit-" or "-moz-").

[0031] Referring back to FIG. 2, in step 212, the web content proxy server 12 determines whether a transformation function is stored in the memory 22 for the CSS property based on the type of web browser identified in step 202 and a name of the CSS property (e.g. "border-radius" of the CSS property 402). In this example, the web content proxy server 12 can store transformation functions for a plurality of CSS properties and a plurality of web browsers. The transformation functions are configured to modify the web page to replace a CSS property with one or more CSS properties that are recognized by the web browser of the requesting one of the client computing devices 16(1)-16(n) and that provide equivalent functionality. If the web content proxy server 12 determines that there is a transformation function for the CSS property and the identified type of web browser, then the Yes branch is taken to step 214.

[0032] In step 214, the web content proxy server 12 applies the transformation function to the CSS property of the web page. Referring to FIGS. 5-7, exemplary CSS code fragments 500, 600, and 700 defining the

"rounded_border" class attribute of FIG. 3 are illustrated. The CSS code fragments 500, 600, and 700 include the CSS property 402 of FIG. 4 transformed for a standards-compliant, Mozilla Firefox™, and WebKit-based type of web browser, respectively, into the CSS properties 502, 602, and 702. Although one example of a transformation and web browser is illustrated and described here, other types of transformations and other types web browsers can also be used.

[0033] In the examples shown in FIGS. 5-7, the applied transformations replace the prefix of the CSS property, but in another example one or more of the CSS property prefix, name, or value (e.g. "5px 10px 15px 20px" of the CSS property 402) can be removed, replace, or modified. Additionally, the CSS property 402 is an example of a CSS rule, but the CSS property can also be a CSS extension function or any other type of CSS statement or declaration.

[0034] In step 216, the web content proxy server 12 determines whether there is an additional CSS property included in the CSS. For example, the CSS can include multiple class attributes having at least one CSS property and/or one or more class attributes with multiple CSS properties. If the web content proxy server 12 determines that the CSS includes an additional CSS property, then the Yes branch is taken to step 210. Optionally, in this example steps 210-214 would be repeated for each CSS property included in the CSS.

[0035] If in step 216 the web content proxy server 12 determines that there is no additional CSS property, then the No branch is taken to step 218. In step 218, the web content proxy server 12 determines whether the retrieved web page includes an additional CSS. If the web content proxy server 12 determines that there is an additional CSS, then the Yes branch is taken to step 210. Optionally, in this example steps 210-216 would again be repeated for each additional CSS included in the retrieved web page.

[0036] If, in step 218, the web content proxy server 12 determines there is no additional CSS, then the No branch is taken to step 208. In step 208, the web page, which may have come from step 206 or may have been modified based on any transformation functions applied in step 214, is sent to the requesting one of the client computing devices 16(1)-16(n).

[0037] Referring back to step 210, if the web content proxy server 12 determines that the prefix of the CSS property does not match the established value, then the No branch is taken to step 216. In step 216, the web content proxy server 12 determines whether the CSS includes an additional CSS property, as described and illustrated earlier.

[0038] Additionally, referring back to step 212, if the web content proxy server 12 determines that there is not a transformation function for the CSS property and the identified type of web browser, then the No branch is optionally taken to step 220. In step 220, the web content proxy server 12 removes the CSS property from the re-

trieved web page. Since the CSS property includes a prefix matching an established value and that the web browser of the requesting one of the client computing devices 16(1)-16(n) will not recognize, the CSS property will not be interpreted by the web browser and can be removed. Once removed, the web content proxy server 12 determines, in step 216, whether the CSS includes an additional CSS property, as described and illustrated earlier.

[0039] Accordingly, as illustrated and described herein this technology provides a number of advantages including improved methods, non-transitory computer readable medium, and apparatuses for reducing the amount of code required to define web page CSSs. With this technology, developers can indicate, using an established prefix value, CSS properties that are to be transformed to be compliant with a web browser from which a request for the web page included the CSS originated. Thereby, developers can define web page CSSs using less and simpler code, which can advantageously be optimized server-side for each requesting web browser.

[0040] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the scope of the invention as claimed. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for processing cascading style sheets, comprising:

receiving by a web content proxy server a Hypertext Transfer Protocol (HTTP) request for a web page from a client computing device;
 identifying by the web content proxy server a type of a browser executing on the client computing device from which the HTTP request originated based on a user agent header included in the HTTP request;
 obtaining by the web content proxy server at least one cascading style sheet (CSS) associated with the requested web page, the at least one CSS including a CSS property, the CSS property comprising at least a prefix followed by a name;
 determining by the web content proxy server and

subsequent to obtaining the CSS when the prefix of the CSS property included in the CSS matches an established value;
 applying by the web content proxy server a transformation function for the CSS property, the transformation function configured to modify the previously obtained CSS to replace the CSS property based on the identified type of browser and the name of the CSS property, when the prefix of the CSS property is determined to match the established value; and
 sending by the web content proxy server the modified CSS to the requesting client computing device to be rendered by the browser executing on the client computing device.

- 2. The method as set forth in claim 1, further comprising:

determining by the web content proxy server when there is a transformation function for the CSS property based on the identified type of browser and the name of the CSS property, when the prefix of the CSS property is determined to match the established value; and
 removing by the web content proxy server the CSS property, when the determining indicates there is not a transformation function for the CSS property.

- 3. The method as set forth in claim 1, wherein the CSS property further comprises at least one value following the name of the CSS property and the transformation function is further configured to replace the prefix or the value of the CSS property.

- 4. The method as set forth in claim 1, further comprising:

determining by the web content proxy server when the at least one CSS includes one or more other CSS properties; and
 repeating by the web content proxy server the determining when the prefix of the one or more other CSS properties matches the established value and the applying the transformation function for the CSS property for each of the one or more other CSS properties prior to sending the modified CSS to the requesting client computing device, when the at least one CSS is determined to include the one or more other CSS properties.

- 5. The method as set forth in claim 1, wherein the CSS property is a CSS rule or a CSS extension function.

- 6. A non-transitory computer readable medium having stored thereon instructions for processing cascading style sheets comprising machine executable code

which when executed by a processor, causes the processor to perform steps comprising:

receiving a Hypertext Transfer Protocol (HTTP) request for a web page from a client computing device;
 identifying a type of a browser executing on the client computing device from which the HTTP request originated based on a user agent header included in the HTTP request;
 obtaining at least one cascading style sheet (CSS) associated with the requested web page, the at least one CSS including a CSS property, the CSS property comprising at least a prefix followed by a name;
 determining subsequent to obtaining the CSS when the prefix of the CSS property included in the CSS matches an established value;
 applying a transformation function for the CSS property, the transformation function configured to modify the previously obtained CSS to replace the CSS property based on the identified type of browser and the name of the CSS property, when the prefix of the CSS property is determined to match the established value; and
 sending the modified CSS to the requesting client computing device to be rendered by the browser executing on the client computing device.

- 7. The medium as set forth in claim 6, further having stored thereon instructions comprising machine executable code which when executed by the processor causes the processor to perform steps further comprising:

determining when there is a transformation function for the CSS property based on the identified type of browser and the name of the CSS property, when the prefix of the CSS property is determined to match the established value; and
 removing the CSS property, when the determining indicates there is not a transformation function for the CSS property.

- 8. The medium as set forth in claim 6, wherein the CSS property further comprises at least one value following the name of the CSS property and the transformation function is further configured to replace the prefix or the value of the CSS property.

- 9. The medium as set forth in claim 6, further having stored thereon instructions comprising machine executable code which when executed by the processor causes the processor to perform steps further comprising:

determining when the at least one CSS includes

one or more other CSS properties; and repeating the determining when the prefix of the one or more other CSS properties matches the established value and the applying the transformation function for the CSS property for each of the one or more other CSS properties prior to sending the modified CSS to the requesting client computing device, when the at least one CSS is determined to include the one or more other CSS properties.

10. The medium as set forth in claim 6, wherein the CSS property is a CSS rule or a CSS extension function.

11. A web content proxy server apparatus, comprising:

a memory coupled to a processor which is configured to execute programmed instructions stored in the memory comprising:

receiving a Hypertext Transfer Protocol (HTTP) request for a web page from a client computing device;
 identifying a type of a browser executing on the client computing device from which the HTTP request originated based on a user agent header included in the HTTP request;
 obtaining at least one cascading style sheet (CSS) associated with the requested web page, the at least one CSS including a CSS property, the CSS property comprising at least a prefix followed by a name;
 determining subsequent to obtaining the CSS when the prefix of the CSS property included in the CSS matches an established value;
 applying a transformation function for the CSS property, the transformation function configured to modify the previously obtained CSS to replace the CSS property based on the identified type of browser and the name of the CSS property, when the prefix of the CSS property is determined to match the established value; and
 sending the modified CSS to the requesting client computing device to be rendered by the browser executing on the client computing device.

12. The apparatus as set forth in claim 11, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising:

determining when there is a transformation function for the CSS property based on the identified type of browser and the name of the CSS property, when the prefix of the CSS property is de-

termined to match the established value; and removing the CSS property, when the determining indicates there is not a transformation function for the CSS property.

13. The apparatus as set forth in claim 11, wherein the CSS property further comprises at least one value following the name of the CSS property and the transformation function is further configured to replace the prefix or the value of the CSS property.

14. The apparatus as set forth in claim 11, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising:

determining when the at least one CSS includes one or more other CSS properties; and repeating the determining when the prefix of the one or more other CSS properties matches the established value and the applying the transformation function for the CSS property for each of the one or more other CSS properties prior to sending the modified CSS to the requesting client computing device, when the at least one CSS is determined to include the one or more other CSS properties.

15. The apparatus as set forth in claim 11, wherein the CSS property is a CSS rule or a CSS extension function.

Patentansprüche

1. Verfahren zum Verarbeiten von Cascading Style Sheets, wobei das Verfahren umfasst, dass:

durch einen Proxy-Server für Webinhalte eine Hypertext-Transfer-Protokoll-Anfrage (HTTP-Anfrage) nach einer Webseite von einer Client-Rechenvorrichtung empfangen wird;
 durch den Proxy-Server für Webinhalte ein Typ eines Browsers, der gerade auf der Client-Rechenvorrichtung ausgeführt wird, von der die HTTP-Anfrage stammte, auf der Grundlage eines User-Agent-Headers, der in der HTTP-Anfrage enthalten ist, identifiziert wird;
 durch den Proxy-Server für Webinhalte mindestens ein Cascading Style Sheet (CSS), das der angefragten Webseite zugeordnet ist, beschafft wird, wobei das mindestens eine CSS eine CSS-Eigenschaft enthält, wobei die CSS-Eigenschaft mindestens einen Vorspann gefolgt von einem Namen umfasst;
 durch den Proxy-Server für Webinhalte und im Anschluss an das Beschaffen des CSS festgestellt wird, ob der Vorspann der CSS-Eigen-

schaft, die in dem CSS enthalten ist, mit einem festgelegten Wert übereinstimmt; durch den Proxy-Server für Webinhalte eine Transformationsfunktion für die CSS-Eigenschaft angewendet wird, wobei die Transformationsfunktion ausgestaltet ist, um das zuvor beschaffte CSS zu modifizieren, indem sie die CSS-Eigenschaft auf der Grundlage des identifizierten Typs des Browsers und des Namens der CSS-Eigenschaft ersetzt, wenn festgestellt wird, dass der Vorspann der CSS-Eigenschaft mit dem festgelegten Wert übereinstimmt; und durch den Proxy-Server für Webinhalte das modifizierte CSS an die anfragende Client-Rechenvorrichtung zur Wiedergabe durch den Browser, der gerade auf der Client-Rechenvorrichtung ausgeführt wird, gesendet wird.

2. Verfahren nach Anspruch 1, das ferner umfasst, dass:

durch den Proxy-Server für Webinhalte auf der Grundlage des identifizierten Typs des Browsers und des Namens der CSS-Eigenschaft festgestellt wird, ob es eine Transformationsfunktion für die CSS-Eigenschaft gibt, wenn festgestellt wird, dass der Vorspann der CSS-Eigenschaft mit dem festgelegten Wert übereinstimmt; und durch den Proxy-Server für Webinhalte die CSS-Eigenschaft entfernt wird, wenn die Feststellung anzeigt, dass es keine Transformationsfunktion für die CSS-Eigenschaft gibt.

3. Verfahren nach Anspruch 1, wobei die CSS-Eigenschaft ferner mindestens einen Wert umfasst, der dem Namen der CSS-Eigenschaft folgt, und wobei die Transformationsfunktion ferner ausgestaltet ist, um den Vorspann oder den Wert der CSS-Eigenschaft zu ersetzen.

4. Verfahren nach Anspruch 1, das ferner umfasst, dass:

durch den Proxy-Server für Webinhalte festgestellt wird, ob das mindestens eine CSS eine oder mehrere andere CSS-Eigenschaften enthält; und

durch den Proxy-Server für Webinhalte das Feststellen, ob der Vorspann der einen oder der mehreren anderen CSS-Eigenschaften mit dem festgelegten Wert übereinstimmt, und das Anwenden der Transformationsfunktion für die CSS-Eigenschaft für jede der einen oder der mehreren anderen CSS-Eigenschaften vor dem Senden des modifizierten CSS an die anfragende Client-Rechenvorrichtung wiederholt werden, wenn festgestellt wird, dass das mindes-

tens eine CSS die eine oder die mehreren anderen CSS-Eigenschaften enthält.

5. Verfahren nach Anspruch 1, wobei die CSS-Eigenschaft eine CSS-Regel oder eine CSS-Erweiterungsfunktion ist.

6. Nicht vorübergehendes computerlesbares Medium mit darin gespeicherten Anweisungen zum Verarbeiten von Cascading Style Sheets, wobei die Anweisungen einen maschinenausführbaren Code umfassen, welcher, wenn er von einem Prozessor ausgeführt wird, veranlasst, dass der Prozessor Schritte ausführt, die umfassen, dass:

eine Hypertext-Transfer-Protokoll-Anfrage (HTTP-Anfrage) nach einer Webseite von einer Client-Rechenvorrichtung empfangen wird; ein Typ eines Browsers, der gerade auf der Client-Rechenvorrichtung ausgeführt wird, von welcher die HTTP-Anfrage stammt, auf der Grundlage eines User-Agent-Headers, der in der HTTP-Anfrage enthalten ist, identifiziert wird;

mindestens ein Cascading Style Sheet (CSS) beschafft wird, das der angefragten Webseite zugeordnet ist, wobei das mindestens eine CSS eine CSS-Eigenschaft enthält, wobei die CSS-Eigenschaft mindestens einen Vorspann gefolgt von einem Namen umfasst;

im Anschluss an das Beschaffen des CSS festgestellt wird, ob der Vorspann der CSS-Eigenschaft, die in dem CSS enthalten ist, mit einem festgelegten Wert übereinstimmt;

eine Transformationsfunktion für die CSS-Eigenschaft angewendet wird, wobei die Transformationsfunktion ausgestaltet ist, um das zuvor beschaffte CSS zu modifizieren, indem sie die CSS-Eigenschaft auf der Grundlage des identifizierten Typs des Browsers und des Namens der CSS-Eigenschaft ersetzt, wenn festgestellt wird, dass der Vorspann der CSS-Eigenschaft mit dem festgelegten Wert übereinstimmt; und

das modifizierte CSS an die anfragende Client-Rechenvorrichtung zur Wiedergabe durch den Browser, der gerade auf der Client-Rechenvorrichtung ausgeführt wird, gesendet wird.

7. Medium nach Anspruch 6, in dem ferner Anweisungen gespeichert sind, die einen maschinenausführbaren Code umfassen, welcher, wenn er von dem Prozessor ausgeführt wird, veranlasst, dass der Prozessor Schritte ausführt, die ferner umfassen, dass:

auf der Grundlage des identifizierten Typs des Browsers und des Namens der CSS-Eigen-

schaft festgestellt wird, ob es eine Transformationsfunktion für die CSS-Eigenschaft gibt, wenn festgestellt wird, dass der Vorspann der CSS-Eigenschaft mit dem festgelegten Wert übereinstimmt; und
 die CSS-Eigenschaft entfernt wird, wenn das Feststellen anzeigt, dass es keine Transformationsfunktion für die CSS-Eigenschaft gibt.

8. Medium nach Anspruch 6,
 wobei die CSS-Eigenschaft ferner mindestens einen Wert umfasst, der dem Namen der CSS-Eigenschaft folgt, und wobei die Transformationsfunktion ferner ausgestaltet ist, um den Vorspann oder den Wert der CSS-Eigenschaft zu ersetzen.

9. Medium nach Anspruch 6,
 in dem ferner Anweisungen gespeichert sind, die einen maschinenausführbaren Code umfassen, welcher, wenn er von dem Prozessor ausgeführt wird, veranlasst, dass der Prozessor Schritte ausführt, die ferner umfassen, dass:

festgestellt wird, ob das mindestens eine CSS eine oder mehrere andere CSS-Eigenschaften enthält; und
 das Feststellen, ob der Vorspann der einen oder der mehreren anderen CSS-Eigenschaften mit dem festgelegten Wert übereinstimmt, und das Anwenden der Transformationsfunktion für die CSS-Eigenschaft für jede der einen oder der mehreren anderen CSS-Eigenschaften wiederholt werden, bevor das modifizierte CSS an die anfragende Client-Rechenvorrichtung gesendet wird, wenn festgestellt wird, dass das mindestens eine CSS die eine oder die mehreren anderen CSS-Eigenschaften enthält.

10. Medium nach Anspruch 6,
 wobei die CSS-Eigenschaft eine CSS-Regel oder eine CSS-Erweiterungsfunktion ist.

11. Proxy-Servervorrichtung für Webinhalte, umfassend:

einen Speicher, der mit einem Prozessor gekoppelt ist, welcher ausgestaltet ist, um programmierte Anweisungen auszuführen, die in dem Speicher gespeichert sind, welche umfassen, dass:

eine Hypertext-Transfer-Protokoll-Anfrage (HTTP-Anfrage) nach einer Webseite von einer Client-Rechenvorrichtung empfangen wird;
 ein Typ eines Browsers, der gerade auf der Client-Rechenvorrichtung ausgeführt wird, von welcher die HTTP-Anfrage stammt, auf

der Grundlage eines User-Agent-Headers, der in der HTTP-Anfrage enthalten ist, identifiziert wird;

mindestens ein Cascading Style Sheet (CSS) beschafft wird, das der angefragten Webseite zugeordnet ist, wobei das mindestens eine CSS eine CSS-Eigenschaft enthält, wobei die CSS-Eigenschaft mindestens einen Vorspann gefolgt von einem Namen umfasst;

nach dem Beschaffen des CSS festgestellt wird, ob der Vorspann der CSS-Eigenschaft, die in dem CSS enthalten ist, mit einem festgelegten Wert übereinstimmt; eine Transformationsfunktion für die CSS-Eigenschaft angewendet wird, wobei die Transformationsfunktion ausgestaltet ist, um das zuvor beschaffte CSS zu modifizieren, indem sie die CSS-Eigenschaft auf der Grundlage des identifizierten Typs des Browsers und des Namens der CSS-Eigenschaft ersetzt, wenn festgestellt wird, dass der Vorspann der CSS-Eigenschaft mit dem festgelegten Wert übereinstimmt; und das modifizierte CSS an die anfragende Client-Rechenvorrichtung zur Wiedergabe durch den Browser, der gerade auf der Client-Rechenvorrichtung ausgeführt wird, gesendet wird.

12. Vorrichtung nach Anspruch 11,
 wobei der Prozessor ferner ausgestaltet ist, um programmierte Anweisungen auszuführen, die in dem Speicher gespeichert sind, die ferner umfassen, dass:

auf der Grundlage des identifizierten Typs des Browsers und des Namens der CSS-Eigenschaft festgestellt wird, ob es eine Transformationsfunktion für die CSS-Eigenschaft gibt, wenn festgestellt wird, dass der Vorspann der CSS-Eigenschaft mit dem festgelegten Wert übereinstimmt; und die CSS-Eigenschaft entfernt wird, wenn das Feststellen anzeigt, dass es keine Transformationsfunktion für die CSS-Eigenschaft gibt.

13. Vorrichtung nach Anspruch 11,
 wobei die CSS-Eigenschaft ferner mindestens einen Wert umfasst, der dem Namen der CSS-Eigenschaft folgt, und wobei die Transformationsfunktion ferner ausgestaltet ist, um den Vorspann oder den Wert der CSS-Eigenschaft zu ersetzen.

14. Vorrichtung nach Anspruch 11,
 wobei der Prozessor ferner ausgestaltet ist, um programmierte Anweisungen auszuführen, die in dem Speicher gespeichert sind, die ferner umfassen,

dass:

festgestellt wird, ob das mindestens eine CSS eine oder mehrere andere CSS-Eigenschaften enthält; und
das Feststellen, ob der Vorspann der einen oder der mehreren anderen CSS-Eigenschaften mit dem festgelegten Wert übereinstimmt, und das Anwenden der Transformationsfunktion für die CSS-Eigenschaft für jede der einen oder der mehreren anderen CSS-Eigenschaften wiederholt werden, bevor das modifizierte CSS an die anfragende Client-Rechenvorrichtung gesendet wird, wenn festgestellt wird, dass das mindestens eine CSS die eine oder die mehreren anderen CSS-Eigenschaften enthält.

15. Vorrichtung nach Anspruch 11, wobei die CSS-Eigenschaft eine CSS-Regel oder eine CSS-Erweiterungsfunktion ist.

Revendications

1. Procédé de traitement de feuilles de style en cascade, comprenant les étapes consistant à :

recevoir, par un serveur proxy de contenus Web, une demande de Protocole de Transfert Hypertexte (HTTP) pour une page Web à partir d'un dispositif informatique client ;
identifier, par le serveur proxy de contenus Web, un type de navigateur s'exécutant sur le dispositif informatique client d'où la demande de HTTP est originaire sur la base d'un en-tête d'agent utilisateur inclus dans la demande de HTTP ;
obtenir, par le serveur proxy de contenus Web, au moins une feuille de style en cascade (CSS) associée à la page Web demandée, la au moins une CSS incluant une propriété de CSS, la propriété de CSS comprenant au moins un préfixe suivi d'un nom ;
déterminer, par le serveur proxy de contenus Web et après l'obtention de la CSS, lorsque le préfixe de la propriété de CSS incluse dans la CSS correspond à une valeur établie ;
appliquer, par le serveur proxy de contenus Web, une fonction de transformation pour la propriété de CSS, la fonction de transformation étant configurée pour modifier la CSS précédemment obtenue pour remplacer la propriété de CSS sur la base du type de navigateur identifié et du nom de la propriété de CSS, lorsque le préfixe de la propriété de CSS est déterminé pour correspondre à la valeur établie ; et
envoyer, par le serveur proxy de contenus Web, la CSS modifiée au dispositif informatique client

demandeur pour un rendu par le navigateur s'exécutant sur le dispositif informatique client.

2. Procédé selon la revendication 1, comprenant en outre les étapes consistant à :

déterminer, par le serveur proxy de contenus Web, lorsqu'il existe une fonction de transformation pour la propriété de CSS sur la base du type de navigateur identifié et du nom de la propriété de CSS, lorsque le préfixe de la propriété de CSS est déterminé pour correspondre à la valeur établie ; et
supprimer, par le serveur proxy de contenus Web, la propriété de CSS, lorsque la détermination indique qu'il n'y a pas de fonction de transformation pour la propriété de CSS.

3. Procédé selon la revendication 1, dans lequel la propriété de CSS comprend en outre au moins une valeur suivant le nom de la propriété de CSS, et la fonction de transformation est en outre configurée pour remplacer le préfixe ou la valeur de la propriété de CSS.

4. Procédé selon la revendication 1, comprenant en outre les étapes consistant à :

déterminer, par le serveur proxy de contenus Web, lorsque la au moins une CSS comprend une ou plusieurs autres propriétés de CSS ; et
répéter, par le serveur proxy de contenus Web, la détermination lorsque le préfixe des une ou plusieurs autres propriétés de CSS correspond à la valeur établie, et appliquer la fonction de transformation pour la propriété de CSS pour chacune des une ou plusieurs autres propriétés de CSS avant d'envoyer la CSS modifiée au dispositif informatique client demandeur, lorsque la au moins une CSS est déterminée pour inclure les une ou plusieurs autres propriétés de CSS.

5. Procédé selon la revendication 1, dans lequel la propriété de CSS est une règle de CSS ou une fonction d'extension de CSS.

6. Support lisible par ordinateur non transitoire sur lequel sont stockées des instructions pour traiter des feuilles de style en cascade comprenant un code exécutable par une machine qui, lorsqu'il est exécuté par un processeur, amène le processeur à effectuer des étapes consistant à :

recevoir une demande de Protocole de Transfert hypertexte (HTTP) pour une page Web à partir d'un dispositif informatique client ;
identifier un type de navigateur s'exécutant sur

- le dispositif informatique client d'où la demande de HTTP est originaire sur la base d'un en-tête d'agent utilisateur inclus dans la demande de HTTP ;
- obtenir au moins une feuille de style en cascade associée à la page Web demandée, la au moins une CSS incluant une propriété de CSS, la propriété de CSS comprenant au moins un préfixe suivi d'un nom ;
- déterminer après l'obtention de la CSS lorsque le préfixe de la propriété de CSS incluse dans la CSS correspond à une valeur établie ;
- appliquer une fonction de transformation pour la propriété de CSS, la fonction de transformation étant configurée pour modifier la CSS précédemment obtenue pour remplacer la propriété de CSS en fonction du type de navigateur identifié et du nom de la propriété de CSS, lorsque le préfixe de la propriété de CSS est déterminé pour correspondre à la valeur établie ; et
- envoyer la CSS modifiée au dispositif informatique client demandeur pour un rendu par le navigateur s'exécutant sur le dispositif informatique client.
7. Support selon la revendication 6, comportant en outre des instructions stockées sur celui-ci comprenant un code exécutable par une machine qui, lorsqu'il est exécuté par le processeur, amène le processeur à effectuer des étapes suivantes consistant à :
- déterminer quand il existe une fonction de transformation pour la propriété de CSS sur la base du type de navigateur identifié et du nom de la propriété de CSS, lorsque le préfixe de la propriété de CSS est déterminé pour correspondre à la valeur établie ; et
- supprimer la propriété de CSS, lorsque la détermination indique qu'il n'y a pas de fonction de transformation pour la propriété de CSS.
8. Support selon la revendication 6, dans lequel la propriété de CSS comprend en outre au moins une valeur suivant le nom de la propriété de CSS, et la fonction de transformation est en outre configurée pour remplacer le préfixe ou la valeur de la propriété de CSS.
9. Support selon la revendication 6, comprenant en outre des instructions stockées sur celui-ci comprenant un code exécutable par une machine qui, lorsqu'il est exécuté par le processeur, amène le processeur à effectuer des étapes consistant en outre à :
- déterminer quand la au moins une CSS comprend une ou plusieurs autres propriétés de CSS ; et
- répéter la détermination lorsque le préfixe des une ou plusieurs autres propriétés de CSS correspond à la valeur établie, et appliquer la fonction de transformation pour la propriété de CSS pour chacune des une ou plusieurs autres propriétés de CSS avant d'envoyer la CSS modifiée au dispositif informatique client demandeur, lorsque la au moins une CSS est déterminée pour inclure les une ou plusieurs autres propriétés de CSS.
10. Support selon la revendication 6, dans lequel la propriété de CSS est une règle de CSS ou une fonction d'extension de CSS.
11. Appareil de serveur proxy de contenus Web, comprenant :
- une mémoire couplée à un processeur qui est configuré pour exécuter des instructions programmées stockées dans la mémoire, comprenant les étapes consistant à :
- recevoir une demande de Protocole de Transfert hypertexte (HTTP) pour une page Web à partir d'un dispositif informatique client ;
- identifier un type de navigateur s'exécutant sur le dispositif informatique client d'où la demande de HTTP est originaire sur la base d'un en-tête d'agent utilisateur inclus dans la demande de HTTP ;
- obtenir au moins une feuille de style en cascade (CSS) associée à la page Web demandée, la au moins une CSS incluant une propriété de CSS, la propriété de CSS comprenant au moins un préfixe suivi d'un nom ;
- déterminer, après l'obtention de la CSS, lorsque le préfixe de la propriété de CSS incluse dans la CSS correspond à une valeur établie ;
- appliquer une fonction de transformation pour la propriété de CSS, la fonction de transformation étant configurée pour modifier la CSS précédemment obtenue pour remplacer la propriété de CSS en fonction du type de navigateur identifié et du nom de la propriété de CSS, lorsque le préfixe de la propriété de CSS est déterminé pour correspondre à la valeur établie ; et
- envoyer la CSS modifiée au dispositif informatique client demandeur pour un rendu par le navigateur s'exécutant sur le dispositif informatique client.
12. Appareil selon la revendication 11, dans lequel le processeur est en outre configuré pour exécuter des

instructions programmées stockées dans la mémoire, comprenant en outre les étapes consistant à :

déterminer quand il existe une fonction de transformation pour la propriété de CSS sur la base du type de navigateur identifié et du nom de la propriété de CSS, lorsque le préfixe de la propriété de CSS est déterminé pour correspondre à la valeur établie ; et
 supprimer la propriété de CSS, lorsque la détermination indique qu'il n'y a pas de fonction de transformation pour la propriété de CSS.

13. Appareil selon la revendication 11, dans lequel la propriété de CSS comprend en outre au moins une valeur suivant le nom de la propriété de CSS, et la fonction de transformation est en outre configurée pour remplacer le préfixe ou la valeur de la propriété de CSS.

14. Appareil selon la revendication 11, dans lequel le processeur est en outre configuré pour exécuter des instructions programmées stockées dans la mémoire, comprenant en outre les étapes consistant à :

déterminer quand la au moins une CSS comprend une ou plusieurs autres propriétés de CSS ; et
 répéter la détermination lorsque le préfixe des une ou plusieurs autres propriétés de CSS correspond à la valeur établie, et appliquer la fonction de transformation pour la propriété de CSS pour chacune des une ou plusieurs autres propriétés de CSS avant d'envoyer la CSS modifiée au dispositif informatique client demandeur, lorsque la au moins un CSS est déterminée pour inclure les une ou plusieurs autres propriétés de CSS.

15. Appareil selon la revendication 11, dans lequel la propriété de CSS est une règle de CSS ou une fonction d'extension de CSS.

5

10

15

20

25

30

35

40

45

50

55

13

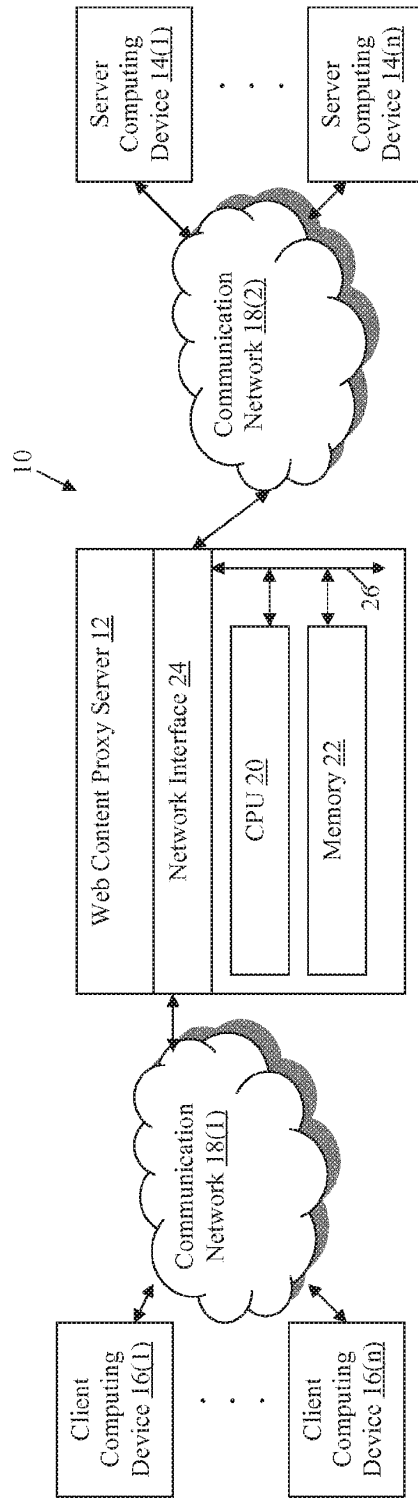


FIG. 1

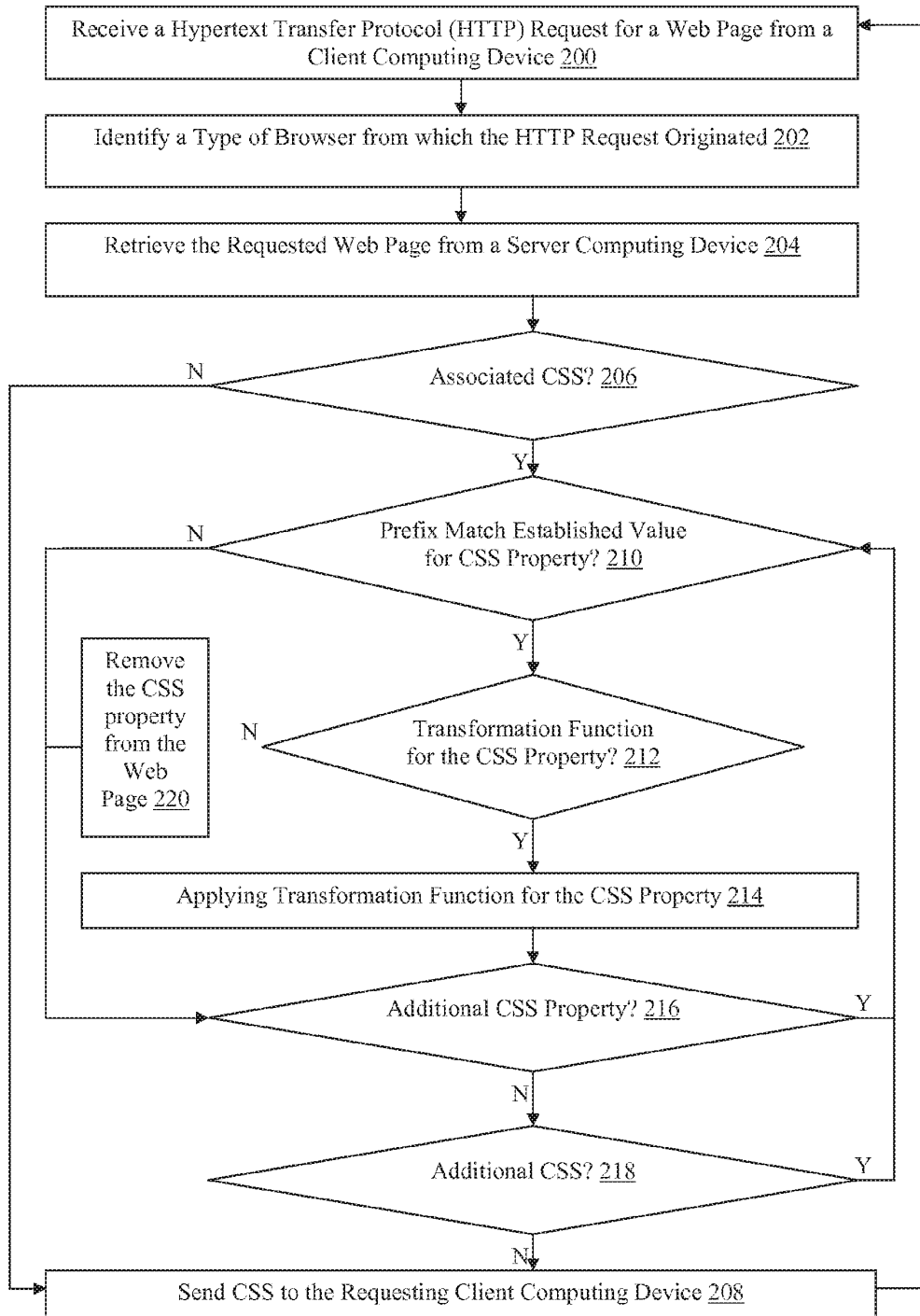


FIG. 2

```
300
.rounded_border {
  -webkit-border-top-left-radius: 5px; 302(1)
  -webkit-border-top-right-radius: 10px; 302(2)
  -webkit-border-bottom-right-radius: 15px; 302(3)
  -webkit-border-bottom-left-radius: 20px; 302(4)
  -moz-border-radius: 5px 10px 15px 20px; 302(5)
  border-radius: 5px 10px 15px 20px; 302(6)
}
```

FIG. 3

```
400
.rounded_border {
  -u-border-radius: 5px 10px 15px 20px; 402
}
```

FIG. 4

```
500
.rounded_border {
  border-radius: 5px 10px 15px 20px; 502
}
```

FIG. 5

```
600
.rounded_border {
  -moz-border-radius: 5px 10px 15px 20px; 602
}
```

FIG. 6

```
700
.rounded_border {
  -webkit-border-top-left-radius: 5px; 702(1)
  -webkit-border-top-right-radius: 10px; 702(2)
  -webkit-border-bottom-right-radius: 15px; 702(3)
  -webkit-border-bottom-left-radius: 20px; 702(4)
}
```

FIG. 7

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 6589291 B [0002]
- US 7574486 B [0002]
- US 7134073 B [0002]



(11) **EP 2 787 454 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
08.10.2014 Bulletin 2014/41

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

(21) Application number: **14163353.7**

(22) Date of filing: **03.04.2014**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(72) Inventor: **Scoda, Enrico**
33035 Martignacco (IT)

(30) Priority: **03.04.2013 US 201313856183**

(74) Representative: **De Ros, Alberto et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria, 9
IT-20122 Milano (IT)

(54) **Methods for optimizing a web content proxy server and devices thereof**

(57) A method, non-transitory computer readable medium, and apparatus that includes obtaining content with an original server cookie comprising a name and a value in response to a client request. Whether the value includes one or more of an established set of characters is determined. A new value is generated based on the value of the original server cookie and a URL encoding of the one or more of the established set of characters and any percent characters included in the value of the

original server cookie prefixed by a first indicator character, when it is determined that the value includes one or more of the established set of characters. A web optimized client cookie comprising the new value and the name of the original server cookie concatenated with a domain attribute and path attribute associated with the content is generated. The web optimized client cookie is provided to the client.

EP 2 787 454 A1

Description**FIELD OF THE INVENTION**

[0001] This invention generally relates to proxy servers and, more particularly, methods for optimizing web content proxy servers and apparatuses thereof.

BACKGROUND

[0002] A web content proxy server optimizes web pages obtained from remote web servers for client devices with special requirements, such as mobile phones, PDAs, and smartphones. Every time a client device requests a web page, the web content proxy server downloads the original page from a remote web server, applies some customized rules to extract relevant content, and adapts it to fit the needs of the requesting client device. By way of example, the web content proxy server may remove JavaScript, linearize content, and adapt the original page to a smaller screen layout for the requesting client device.

[0003] In computing, a cookie, such as a tracking cookie, browser cookie, and HTTP cookie, is a small piece of text stored by a web browser on the client device. A cookie includes one or more name-value pairs containing data, such as user preferences, shopping cart contents, the identifier for a server-based session, or other data used by websites.

[0004] Web content proxy servers need to save cookies to enable the client devices to interact with the original website at the remote web servers in the correct way. Accordingly, web content proxy servers store these cookies in an internal memory and associate them with the corresponding session from each client device so that when the same client device sends a request for a new page, the web content proxy server will load the matching cookies and send them to the remote web server to get the page to process. Unfortunately, storing the cookies for these client devices causes problems with scalability, security, and privacy of the web content proxy servers.

SUMMARY OF THE INVENTION

[0005] A method for optimizing a web content proxy server includes obtaining at a web content proxy server content with an original server cookie comprising at least a name and a value from a content server in response to a request from a client device for the content. A determination is made whether the value of the original server cookie includes one or more of an established set of characters at the web content proxy server. A new value is generated at the web content proxy server based on the value of the original server cookie and a uniform resource locator (URL) encoding of at least the one or more of the established set of characters and any percent characters included in the value of the original server cookie prefixed by a first indicator character, when the determination in-

dicates the value of the original server cookie includes one or more of the established set of characters. A web optimized client cookie including the new value and a new name with at least the name of the original server cookie concatenated with at least a portion of a domain attribute and a path attribute associated with the obtained content is generated at the web content proxy server. At least the web optimized client cookie is provided by the web content proxy server to the requesting client device.

[0006] A non-transitory computer readable medium having stored thereon instructions for optimizing a web content proxy server comprising machine executable code which when executed by a processor, causes the processor to perform steps including obtaining content with an original server cookie comprising at least a name and a value from a content server in response to a request from a client device for the content. A determination is made whether the value of the original server cookie includes one or more of an established set of characters. A new value is generated based on the value of the original server cookie and a uniform resource locator (URL) encoding of at least the one or more of the established set of characters and any percent characters included in the value of the original server cookie prefixed by a first indicator character, when the determination indicates the value of the original server cookie includes one or more of the established set of characters. A web optimized client cookie including the new value and a new name with at least the name of the original server cookie concatenated with at least a portion of a domain attribute and a path attribute associated with the obtained content is generated. At least the web optimized client cookie is provided to the requesting client device.

[0007] A web content proxy server apparatus, comprising a memory and a processor coupled to the memory and configured to execute programmed instructions stored in the memory including obtaining content with an original server cookie comprising at least a name and a value from a content server in response to a request from a client device for the content. A determination is made whether the value of the original server cookie includes one or more of an established set of characters. A new value is generated based on the value of the original server cookie and a uniform resource locator (URL) encoding of at least the one or more of the established set of characters and any percent characters included in the value of the original server cookie prefixed by a first indicator character, when the determination indicates the value of the original server cookie includes one or more of the established set of characters. A web optimized client cookie including the new value and a new name with at least the name of the original server cookie concatenated with at least a portion of a domain attribute and a path attribute associated with the obtained content is generated. At least the web optimized client cookie is provided to the requesting client device.

[0008] This technology provides a number of advantages including providing a method, non-transitory com-

puter readable medium and an apparatus that optimizes implementation of a web content proxy server for interactions involving cookies between client devices and remote web servers. With this technology, original server cookies are transformed by the web content proxy server to web optimized client cookies which are transmitted to the client devices requesting the web pages for storage and use with subsequent requests.

[0009] This technology provides greater scalability because the web optimized client cookies are stored in the web browser at the client device, not in memory at the web content proxy server. As a result, the web content proxy server does not face any issues with respect to memory storage capacity due to the number of sessions with cookies for client devices. The web content proxy server can use the same memory whether there are 100 or 1,000,000 or more client devices engaged in sessions with the remote web servers through the web content proxy server.

[0010] Additionally, this technology provides greater security and privacy because the web content proxy server does not contain a centralized database of original server cookies which contain session information from client devices browsing pages of web sites. Instead, these original server cookies are translated into web optimized client cookies which are then dispersed out among the client devices. As a result, the web content proxy server does not have any stored cookies from interactions between client devices and remote web servers that could be used to steal identity or other confidential information of these client devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011]

FIG. 1 is a block diagram of an exemplary environment with an optimized web content proxy server;

FIG. 2A is an example of a HTTP request for a web page from a remote web server;

FIG. 2B is an example of a HTTP response with an original server cookie from a remote web server to a HTTP request;

FIG. 2C is an example of a HTTP response containing the web optimized client cookie generated from the original server cookie received shown in FIG. 2B;

FIG. 2D is an example of another HTTP request with the web optimized client cookie shown in FIG. 2C for a web page from a remote web server.

FIG. 2E is an example of the another HTTP request with the web optimized client cookie shown in FIG. 2D translated into the original server cookie for transmission to the remote web server with the another

get request;

FIG. 3 is a flow chart of an example of a method for generating a web optimized client cookie from an original server cookie to optimize implementation of a web content proxy server; and

FIG. 4 is a flow chart of an example of a method for transforming a web optimized client cookie back to an original server cookie to optimize implementation of a web content proxy server.

DETAILED DESCRIPTION OF THE INVENTION

[0012] An exemplary environment 10 in which a web content proxy server 12 is optimized is illustrated in FIG. 1. The exemplary environment 10 includes a web content proxy server or apparatus 12, client devices 14(1)-14(n), web server devices 16(1)-16(n), and communication networks 18(1)-18(2), although other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can be used. This technology provides a number of advantages including providing a method, computer readable medium and an apparatus that optimizes implementation of a web content proxy server for interactions involving cookies between client devices and remote web servers.

[0013] Referring more specifically to FIG. 1, the web content proxy server 12 optimizes the handling of original server cookies from the web server devices 16(1)-16(n) for requesting client devices 14(1)-14(n) and the handling of web optimized client cookies, although the web content proxy server 12 can provide other numbers and types of functions. Although one web content proxy server 12 is shown, other numbers and types of web content proxy devices and systems can be used.

[0014] The web content proxy server 12 includes a central processing unit (CPU) or processor 13, a memory 15, and an interface system 17 which are coupled together by a bus 19 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor 13 in the web content proxy server 12 executes a program of stored instructions one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[0015] The memory 15 in the web content proxy server 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medi-

um which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 13, can be used for the memory 15 in the web content proxy server 12. In these embodiments, the memory 15 includes a core module 21 and a cookie module 23 which store programmed instructions for one or more aspects of the present invention as described and illustrated herein, although the memory can comprise other types and numbers of systems, devices, and elements in other configurations which store other data. The cookie module 23 includes programmed instructions and/or logic configured to translate an original server cookie into a web optimized client cookie and to extract the original server cookie when a web optimized client cookie is received, although the cookie module 23 can have other types and numbers of functions as described and illustrated herein.

[0016] The interface system 17 in the web content proxy server 12 is used to operatively couple and communicate between the web content proxy server 12 and the client devices 14(1)-14(n) and the web server devices 16(1)-16(n) via the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. By way of example only, the communication networks 18(1) and 18(2) can use TCP/IP over Ethernet and industry-standard protocols, including HTTP, HTTPS, WAP, and SOAP, although other types and numbers of communication networks, such as a direct connection, a local area network, a wide area network, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[0017] Each of the client devices 14(1)-14(n) enables a user to request, get and interact with web pages from one or more web sites hosted by the web server devices 16(1)-16(n) through the web content proxy server 12 via one or more communication networks, although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. Although multiple client devices 14(1)-14(n) are shown, other numbers and types of user computing systems could be used. In this example, the client devices 14(1)-14(n) comprise mobile devices with Internet access that permit a website form page or other retrieved data to be displayed, although each of the client devices 14(1)-14(n). By way of example only, one or more of the client devices 14(1)-14(n) can comprise smart phones, personal digital assistants, or computers.

[0018] Each of client devices 14(1)-14(n) in this example is a computing device that includes a central processing unit (CPU) or processor 20, a memory 22, user input device 24, a display 26, and an interface system 28, and which are coupled together by a bus 30 or other link, although one or more of client devices 14(1)-14(n) can

include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 20 in each of client devices 14(1)-14(n) executes a program of stored instructions for one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0019] The memory 22 in each of the client devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein as well as the web optimized client cookies, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 20 can be used for the memory 22 in each of the client devices 14(1)-14(n).

[0020] The user input device 24 in each of the client devices 14(1)-14(n) is used to input selections, such as requests for a particular website form page or to enter data in fields of a form page, although the user input device could be used to input other types of data and interact with other elements. The user input device can include keypads, touch screens, and/or vocal input processing systems although other types and numbers of user input devices can be used.

[0021] The display 26 in each of the client devices 14(1)-14(n) is used to show data and information to the user, such as website page by way of example only. The display in each of the client devices 14(1)-14(n) is a phone screen display, although other types and numbers of displays could be used depending on the particular type of client device.

[0022] The interface system 28 in each of the client devices 14(1)-14(n) is used to operatively couple and communicate between the client devices 14(1)-14(n) and the web content proxy server 12 and web server devices 16(1)-16(n) over the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0023] The web server devices 16(1)-16(n) provide one or more pages from one or more web sites for use by one or more of the client devices 14(1)-14(n) via the web content proxy server 12, although the web server devices 16(1)-16(n) can provide other numbers and types of applications and/or content and can have provide other numbers and types of functions. Although web server devices 16(1)-16(n) are shown for ease of illustration and discussion, other numbers and types of web server systems and devices can be used.

[0024] Each of the web server devices 16(1)-16(n) include a central processing unit (CPU) or processor, a memory, and an interface system which are coupled to-

gether by a bus or other link, although each of the web server devices 16(1)-16(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor in each of the web server devices 16(1)-16(n) executes a program of stored instructions one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[0025] The memory in each of the web server devices 16(1)-16(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated by way of the embodiments, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor, can be used for the memory in each of the web server devices 16(1)-16(n).

[0026] The interface system in each of the web server devices 16(1)-16(n) is used to operatively couple and communicate between the web server devices 16(1)-16(n) and the web content proxy server 12 and the client devices 14(1)-14(n) via communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0027] Although embodiments of the web content proxy server 12, the client devices 14(1)-14(n), and the web server devices 16(1)-16(n), are described and illustrated herein, each of the client devices 14(1)-14(n), the web content proxy server 12, and the web server devices 16(1)-16(n), can be implemented on any suitable computer system or computing device. It is to be understood that the devices and systems of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0028] Furthermore, each of the systems of the embodiments may be conveniently implemented using one or more general purpose computer systems, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0029] In addition, two or more computing systems or devices can be substituted for any one of the systems in any embodiment of the embodiments. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the embodiments. The em-

bodiments may also be implemented on computer system or systems that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0030] The embodiments may also be embodied as a computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0031] An exemplary method for generating a web optimized client cookie from an original server cookie to optimize implementation of the web content proxy server 12 will now be described with reference to FIGS. 1-2C and 3. In step 50, in this example one of the client devices 14(1)-14(n) via a web browser requests a page A.html at the website, "www.example.com" as shown in one example in FIG. 2A. This request is transmitted to the web content proxy server 12 which processes and transmits the request to the one of the web servers 16(1)-16(n) hosting the website "www.example.com."

[0032] The hosting one of the web servers 16(1)-16(n) provides a response in this example for the requested page A.html which also contains an original server cookie "SESSION" to the web content proxy server 12 as shown in FIG. 2B. In this example, SESSION has a value equal to "1234", the domain attribute is equal to ".example.com" and the path attribute is equal to "/". This response uses the HTTP header Field "Set-Cookie". The cookie is a string formed by the pair "name=value", followed by optional attributes, like those in this example indicating the server domain attribute and path attribute accepting this cookie. Although one illustrative example is described herein, this technology can be used with specifications for all cookies.

[0033] Next, in step 52 the web content proxy server 12 determines whether the original server cookie includes the domain attribute for the requested web page provided by the hosting one of the web servers 16(1)-16(n). If in step 52 the web content proxy server 12 determines the original server cookie does not include the domain attribute, then the No branch is taken to step 54. In step 54, the web content proxy server 12 extracts the domain attribute from the requested web page provided by the hosting one of the web servers 16(1)-16(n). If in step 52 the web content proxy server 12 determines the original server cookie does include the domain attribute, then the Yes branch is taken to step 56.

[0034] In step 56, the web content proxy server 12 de-

termines whether the original server cookie includes the path attribute for the requested web page provided by the hosting one of the web servers 16(1)-16(n). If in step 56 the web content proxy server 12 determines the original server cookie does not include the path attribute, then the No branch is taken to step 58. In step 58, the web content proxy server 12 extracts the path attribute from the requested web page provided by the hosting one of the web servers 16(1)-16(n). If in step 56 the web content proxy server 12 determines the original server cookie does include the path attribute, then the Yes branch is taken to step 60.

[0035] In step 60, the web content proxy server 12 determines whether the domain attribute or the path attribute contains any characters in an established set or a percent, a first, or a second character. In this example, the established set of characters includes a control character, a double quote character, a space character, a comma character, a semicolon character, a backslash character, or one or more reserved characters identified in Request for Comment No. 6265 ("RFC6265"). In this example, the first character is a plus sign character and the second character is an equal sign character, although other characters can also be used in the established set or as the first or second character.

[0036] If the web content proxy server 12 determines that the domain attribute or the path attribute associated with the content contains any characters in the established set or a percent, a first, or a second character, then the Yes branch is taken to step 62. In step 62, the web content proxy server 12 generates a new domain attribute and/or a new path attribute by performing a uniform resource locator (URL) encoding of any characters in the established set and any percent, first, or second characters in the domain attribute and/or path attribute. Since the domain attribute and path attribute in this example are incorporated into a name of a web optimized client cookie, as described and illustrated in more detail below, the domain attribute and path attribute must be converted to conform to cookie name specifications set forth in RFC6265.

[0037] Additionally, the plus sign character is used to separate the name of the original server cookie from a domain attribute and a path attribute in a new name for the web optimized client cookie, as described and illustrated in detail below. The equal sign character is used to separate the new name from a new value in the web optimized client cookie, also as described and illustrated in detail below. Other characters can also be used as separators and corresponding first and second characters. Since, in this example, some character(s) of the domain attribute and/or the path attribute may be URL encoded, instances of the percent character in the domain attribute and path attribute must also be URL encoded to distinguish instances of the percent character from a percent character introduced by a URL encoding of another character in the domain attribute or path attribute.

[0038] Subsequent to the encoding in step 62 or, if the web content proxy server 12 determines, in step 60, that the domain attribute and path attributes associated with the content do not contain any characters in the established set or any percent, first, or second characters, and the No branch is taken, the web content proxy server 12 generates a name for a new web optimized client cookie in step 64. The web content proxy server 12 generates the name for the new web optimized client cookie by concatenating the original name and the original domain attribute and/or the path attribute, or the new domain attribute and/or the new path attribute generated in step 62, each separated by the plus character, although other manners for generating the new name can be used.

[0039] The resulting new name is "universal resource locator encoded" to keep conformance to the cookie specification. Additionally, the resulting new name is unique even if different domain attributes contain cookies with the same name. This new name contains all the information necessary for the web content proxy server 12 to extract the original server cookie later, as described in greater detail by reference to FIG. 4.

[0040] In step 66, the web content proxy server 12 determines whether the value of the original server cookie includes one or more of an established set of characters. Optionally, the established set of characters is the same set of characters used in step 60, although a different set of characters can also be used. If the web content proxy server 12 determines that the value of the original server cookie includes one or more of the established set of characters, then the Yes branch is taken to step 68.

[0041] In step 68, the web content proxy server generates a new value based on the value of the original server cookie and a URL encoding of any characters included in the established set of characters, as well as any percent characters. Since the new value must comply with RFC 6265 but it is not guaranteed that original server cookie value is compliant, the original server cookie value must be converted to comply with cookie-protocol specifications

[0042] Additionally, instances of the percent character in the value must also be URL encoded to distinguish the instances with a percent character introduced by a URL encoding of another character in the value. In this example, the generated new value is further prefixed by a first indicator character such as "e" in this example, although other characters can be used for the first indicator character.

[0043] Referring back to step 66, if the web content proxy server 12 determines that the value of the original server cookie does not include any of the established set of characters, then the No branch is taken to step 70. In step 70, the web content proxy server 12 optionally generates a new value based on the value of the original server cookie prefixed by a second indicator character such as "n" in this example, although other characters can be used for the second indicator character. In other examples, the first indicator character is used to deter-

mine whether the value has been encoded, as described and illustrated in greater detail with reference to FIG. 4, and a second indicator character is not used.

[0044] In step 72, the web content proxy server 12 forms a new web optimized client cookie having the new name generated in step 64 and new value generated in step 68 or step 70. In this example, the domain attribute in the web optimized client cookie is not specified, and the path attribute is associated with a value "/". Other values can be used, such as one for the path attribute that corresponds to a prefix associated with this optimization method (by way of example only "/mt/").

[0045] By way of example only, when the web content proxy server 12 receives a response with the original server cookie as shown in FIG. 2B, the web content proxy server 12 generates a web optimized client cookie as shown in FIG. 2C. More specifically, the original server cookie: SESSION=1234; domain attribute=.example.com; and path attribute=/ is transformed by the web content proxy server 12 to a web optimized client cookie: SESSION+.example.com+/=n1234; path attribute=/mt/. Accordingly, in this illustrative example the new web optimized client cookie name represents the concatenation of the original server cookie name, original domain attribute, and original path attribute each separated by the plus sign character. Additionally, in this example, the new web optimized client cookie value represents the original server cookie value prefixed by an indicator character indicating whether the original server cookie value is encoded.

[0046] In this example, the original domain attribute and original path attribute are used because none of the characters of the original domain attribute and original path attribute were included in the established set of characters or matched the first or second characters (plus sign and equal sign, respectively, in this example). Other orders and manners for forming the name of the web optimized client cookie can also be used. In this example, the value of the web optimized client cookie is prefixed with an "n" character indicating that the value of the original server cookie did not include any characters in the established set of characters and, therefore, was not URL encoded. Additionally, in this example, the new path attribute corresponds to a prefix "/mt/" associated with this optimization method.

[0047] Next, in step 74 the web content proxy server 12 copies the remaining attributes in the original server cookie, such as an expiration date for the original server cookie by way of example, in the web optimized client cookie, although other amounts of the remaining attributes could be copied and other information also could be appended.

[0048] In step 76, the original server cookie which has been translated into the web optimized client cookie is now provided to the core module 21 in the web content proxy server 12. The core module 21 includes programmed instructions and/or logic to manage the transmission of the web optimized client cookie and the con-

tent from the web content proxy server 12 to the requesting one of the client devices 14(1)-14(n). The web browser at the requesting one of the client devices 14(1)-14(n) receives and saves the web optimized client cookie in the memory 22 at the requesting one of the client devices 14(1)-14(n).

[0049] Accordingly, in this illustrative example, the web optimized client cookie shown in FIG. 2C is stored in the memory 22 at the requesting one of the client devices 14(1)-14(n) and is not stored by the web content proxy server 12. While conformance with cookie name specifications could be maintained by URL encoding all characters, or all special characters, of the domain attribute, path attribute, and value of the original server cookie, in this example, only a limited number of characters are URL encoded, thereby reducing the size of the web optimized client cookie. By reducing the size of the web optimized client cookie, less space in the memory 22 of the requesting one of the client devices 14(1)-14(n) is utilized and the web optimized client cookie can be sent to the requesting one of the client devices 14(1)-14(n) in less time.

[0050] Referring now to FIGS. 1, 2D-2E, and 4, an exemplary method for translating a web optimized client cookie back to an original server cookie to optimize the implementation of the web content proxy server 12 will now be described. In step 100, in this example one of the client devices 14(1)-14(n) via a web browser submits another request to the web content proxy server 12 for page B.html at the website, "www.example.com" as shown in one example in FIG. 2D. This request includes a web optimized client cookie, which in this example comprises a name/value pair: SESSION+.example.com+/=n1234.

[0051] In step 102, the web content proxy server 12 extracts the original server cookie name and the encoded domain attribute and path attributes from the name of the web optimized client cookie. In this illustrative example, the original server cookie name and the domain attribute and path attributes are extracted by the web content proxy server from the name: SESSION+.example.com+/. The domain attribute can be extracted based on the characters following the first plus sign character separating the domain attribute from the original server cookie name and the path attribute can be extracted based on the characters following the second plus sign character separating the domain attribute from the path attribute. Additionally, the value can be extracted based on the characters of the web optimized client cookie following the equal sign character. The web content proxy server 12 further performs a URL decoding of the domain attribute and path attributes. In this example, no characters of the domain attribute and path attribute were encoded in step 62 and the extracted domain attribute and path attribute represent the original domain attribute and path attributes as included in the original server cookie.

[0052] In step 104, the web content proxy server 12 determines whether the extracted domain attribute and

path attribute identify a web optimized client cookie that is a match to a universal resource locator for the requested web page. If in step 104 the web content proxy server 12 determines the extracted domain attribute and path attributes identify a web optimized client cookie that is not a match, then the No branch is taken to step 106. In step 106, the web content proxy server 12 submits the request to the hosting one of the web servers 16(1)-16(n) hosting the request page without an original server cookie. In this illustrative example, the requested page is "B.html." If in step 104 the web content proxy server 12 determines the extracted domain attribute and path attributes identify a web optimized client cookie that is a match, then the Yes branch is taken to step 108.

[0053] In step 108, the web content proxy server 12 determines whether the web optimized client cookie included in the request received in step 100 includes a value, prefixed with the first indicator character or the second indicator character. If the web content proxy server 12 determines that the value of the web optimized client cookie is prefixed by the first indicator character, then the Yes branch is taken to step 110.

[0054] In step 110, the web content proxy server 12 generates a decoded value by performing a URL decoding of any characters following the first indicator character in the value of the web optimized client cookie extracted in step 102. Referring back to step 108, if the web content proxy server 12 determines that the value of the web optimized client cookie is prefixed by the second character, then the No branch is taken to step 112. In examples in which the second indicator is not used, and optional step 70 is not performed, the No branch is taken by web content proxy server 12 when the web content proxy server 12 determines the extracted value is not prefixed by the first indicator character.

[0055] In step 112, the web content proxy server 12 generates a decoded value including the characters following the second indicator character in the value of the web optimized client cookie extracted in step 102. In examples in which the second indicator character is not used, the value of the web optimized client cookie extracted in step 102 is used as the decoded value. Since, in the example illustrated in FIG. 2D, the value n1234 is prefixed with an "n", the No branch is taken from step 108 and the decoded value generated in step 112 is 1234. Accordingly, one of the indicator characters is inserted by the web content proxy server 12 to indicate to the web content proxy server 12 whether the value included in a subsequent request including a web optimized client has been URL encoded, as described and illustrated earlier with reference to step 68.

[0056] In step 114, the web content proxy server 12 creates a new cookie by associating the name extracted from the web optimized client cookie in step 102 with the value for the original server cookie decoded in step 110 or 112. The extracted name and decoded value comprise the original server cookie which is appended to the HTTP cookie header fields of the request to be sent to the one

of the web servers 16(1)-16(n) hosting the requested web page. In this illustrative example, the extracted name SESSION is associated with the value 1234.

[0057] In step 116, the web content proxy server 12 submits the request with the reconstituted original server cookie to the one of the web servers 16(1)-16(n) hosting the requested page. In this illustrative example, the request with the reconstituted original server cookie as shown in FIG. 2E is transmitted to the one of the web servers 16(1)-16(n) hosting the requested web page B.html.

[0058] Accordingly, as illustrated and described herein this technology provides a number of advantages including providing a method, computer readable medium and an apparatus that optimizes implementation of a web content proxy server for interactions involving cookies between client devices and remote web servers. With this technology, the web content proxy server is much more scalable because of the reduced memory storage demands and the web content proxy server poses a much lower security and privacy risk to information provided by the client devices 14(1)-14(n). Additionally, the size of the cookies, and associated time required to send cookies to client devices and storage requirements on the client devices, is advantageously reduced while conformance with cookie name specifications is maintained.

[0059] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for optimizing a web content proxy server, the method comprising:

obtaining at a web content proxy server content with an original server cookie comprising at least a name and a value from a content server in response to a request from a client device for the content;

determining at the web content proxy server whether the value of the original server cookie includes one or more of an established set of characters;

generating with the web content proxy server a

new value based on the value of the original server cookie and a uniform resource locator (URL) encoding of at least the one or more of the established set of characters and any percent characters included in the value of the original server cookie prefixed by a first indicator character, when the determining indicates the value of the original server cookie includes one or more of the established set of characters; generating at the web content proxy server a web optimized client cookie comprising the new value and a new name comprising at least the name of the original server cookie concatenated with at least a portion of a domain attribute and a path attribute associated with the obtained content; and providing with the web content proxy server at least the web optimized client cookie to the requesting client device.

2. The method as set forth in claim 1, further comprising generating with the web content proxy server a new value comprising the value of the original server cookie prefixed by a second indicator character, when it is determined that the value of the original server cookie does not include one or more of the established set of characters.

3. The method as set forth in claim 1, further comprising:
 generating with the web content proxy server a new domain attribute or a new path attribute based on the domain attribute or the path attribute associated with the content and a URL encoding of at least any percent character, first character, or second character of the domain attribute or the path attribute associated with the content; and wherein the domain attribute or the path attribute of the web optimized client cookie comprises the new domain attribute or the new path attribute and the new path attribute and the new value are separated by the second character.

4. The method as set forth in claim 3, further comprising:
 determining at the web content proxy server whether the original server cookie includes a domain attribute or a path attribute; generating with the web content proxy server the new domain attribute or the new path attribute based on the domain attribute or the path attribute included in the original server cookie, when it is determined that the original server cookie includes the domain attribute or the path attribute; and

generating with the web content proxy server the new domain attribute or the new path attribute based on a domain attribute or a path attribute of a network address of the obtained content, when it is determined that the original server cookie does not include the domain attribute or the path attribute.

5. The method as set forth in claim 3, wherein the first character is a plus character and the second character is an equal character and the established set of characters further comprises one or more of a control character, a double quote character, a space character, a comma character, a semicolon character, a backslash character, the plus character, the equal character, or one or more reserved characters identified in Request for Comment No. 6265.

6. The method of claim 1, further comprising:
 processing with the web content proxy server a subsequent request from the client device by reconstituting the original server cookie using a web optimized client cookie included in the subsequent request, the reconstituting comprising:

determining whether the web optimized client cookie included in the subsequent request includes a value prefixed with the first indicator character; and generating a URL decoding of a plurality of characters following the first indicator character included in the web optimized client cookie included in the subsequent request, when it is determined that the web optimized client cookie included in the subsequent request includes a value prefixed with the first indicator character.

7. The method as set forth in claim 6, wherein the processing further comprises:
 determining whether a domain attribute and a path attribute of the reconstituted original server cookie correspond with a network address of the subsequent request; and providing the subsequent request with the reconstituted original server cookie to the content server, when it is determined that the domain attribute and the path attribute of the reconstituted original server cookie correspond with the network address of the subsequent request.

8. The method as set forth in claim 1, wherein the providing further comprises providing the obtained content and the web optimized client cookie to the client device without storing the original server cookie or the web optimized client cookie.

9. The method as set forth in claim 1, wherein the generating the web optimized client cookie further comprises appending one or more original attributes of the original server cookie to the web optimized client cookie. 5

10. A non-transitory computer readable medium having stored thereon instructions for optimizing a web content proxy server comprising machine executable code which when executed by a processor, causes the processor to perform steps of the method according to any of claims from 1 to 9. 10

11. A web content proxy server apparatus, comprising: 15
a memory; and
a processor coupled to the memory, and configured to execute programmed instructions stored in the memory and comprising the steps of the method according to any of claims from 1 to 9. 20

25

30

35

40

45

50

55

10

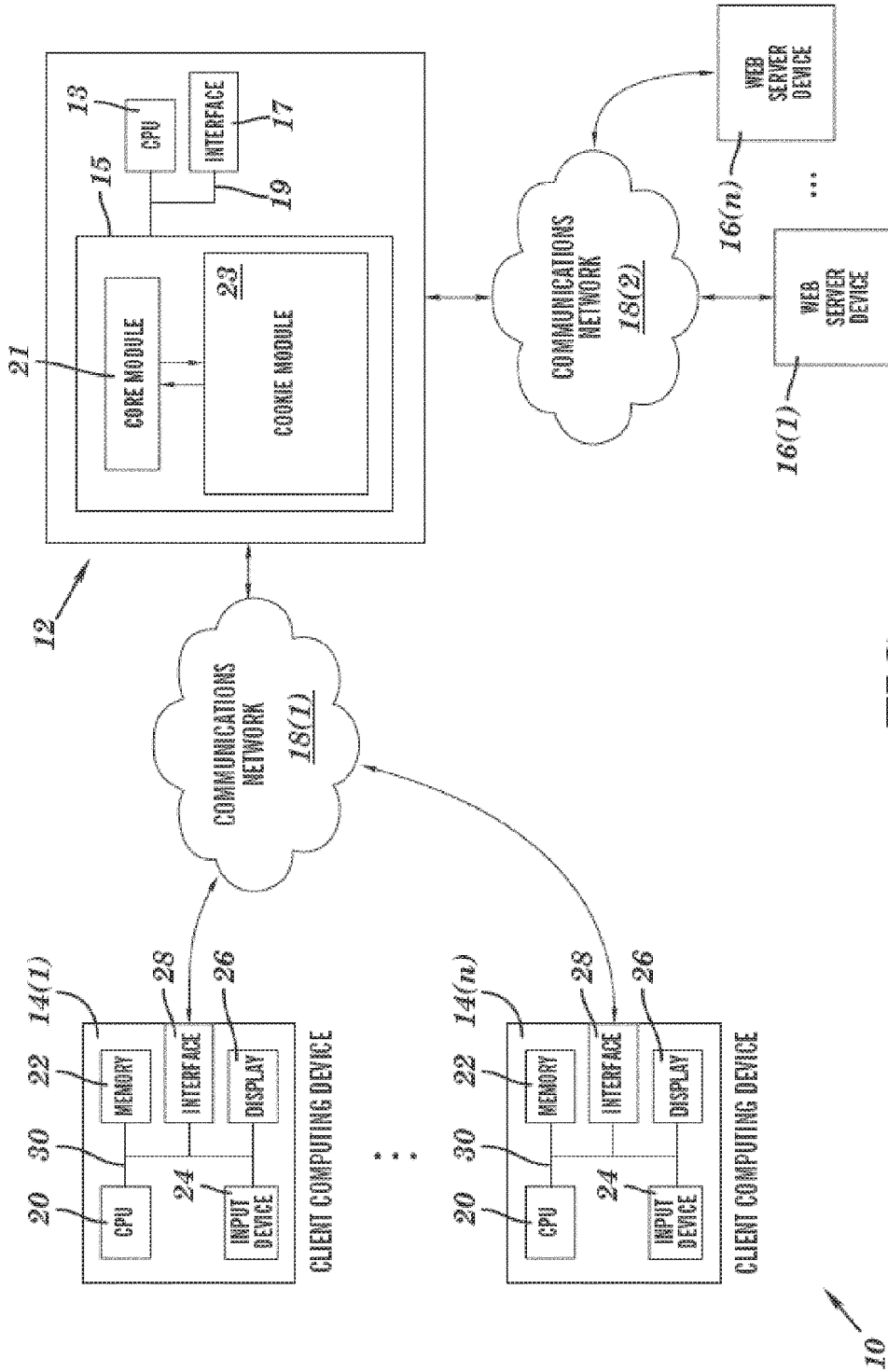


FIG. 1

```
GET /A.html HTTP/1.1
Host: www.example.com
Accept: */*
User-Agent: my-mobile-browser 1.0
```

FIG. 2A

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 5300
Set-Cookie: SESSION=1234; domain=.example.com; path=/
```

FIG. 2B

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 5300
Set-Cookie: SESSION+.example.com+/=n1234; path=mt/
```

FIG. 2C

```
GET /mt/www.example.com/B.html HTTP/1.1
User-Agent: my-mobile-browser 1.0
Host: m.proxy.com
Accept: */*
Cookie: SESSION+.example.com+/=n1234
```

FIG. 2D

```
GET /B.html HTTP/1.1
User-Agent: my-mobile-browser 1.0
Host: www.example.com
Accept: */*
Cookie: SESSION=1234
```

FIG. 2E

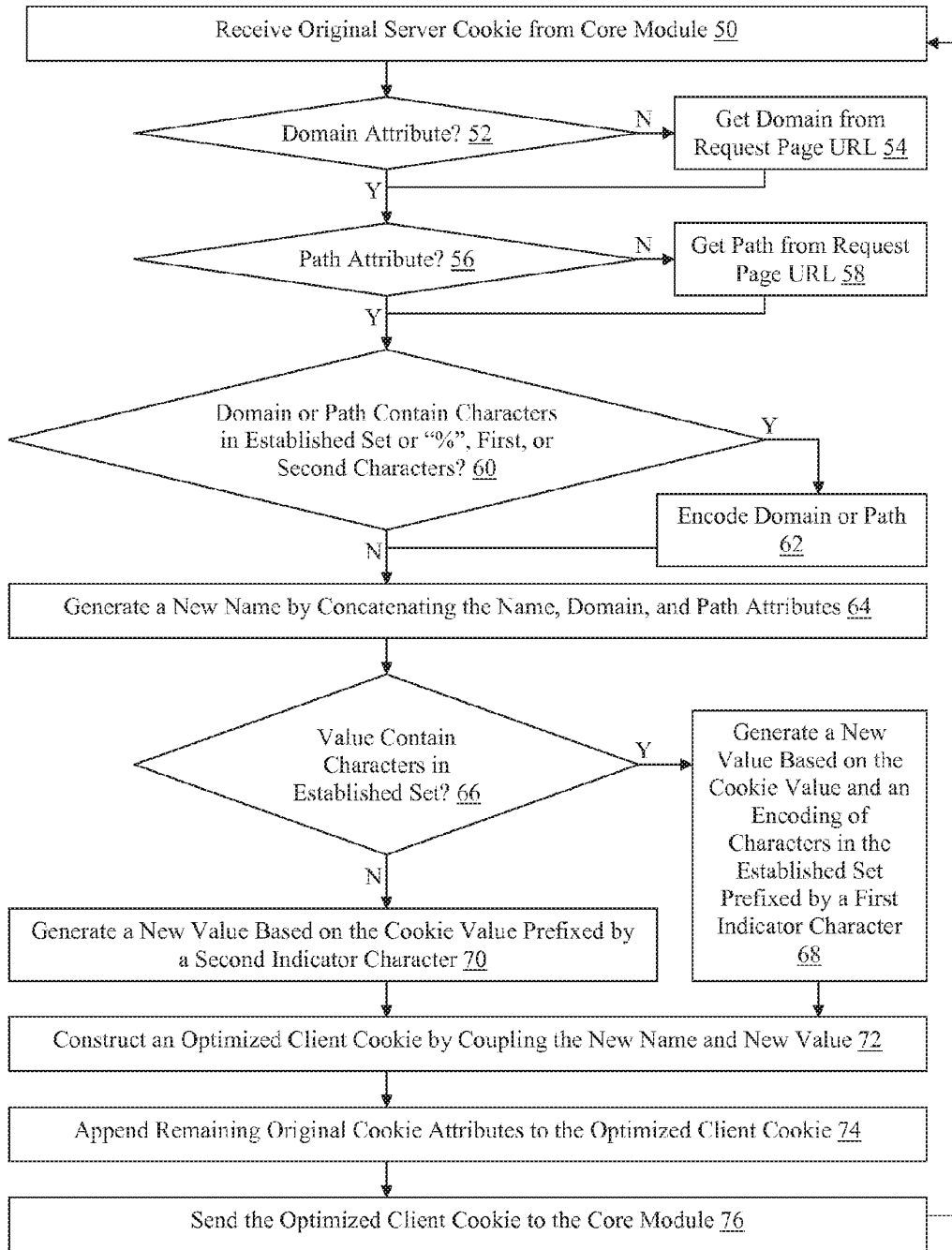


FIG. 3

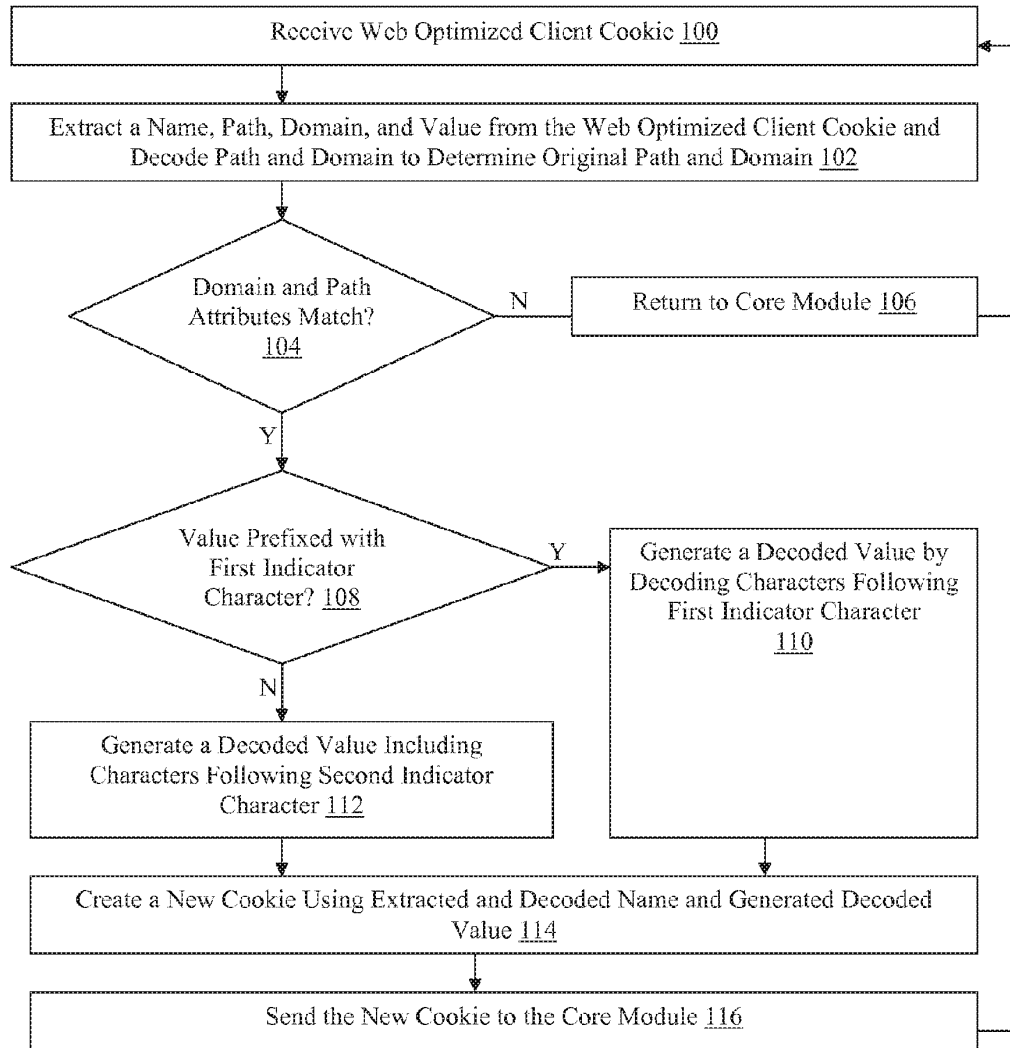


FIG. 4



EUROPEAN SEARCH REPORT

Application Number
EP 14 16 3353

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	EP 2 363 995 A1 (USABLENET INC [US]) 7 September 2011 (2011-09-07) * column 1, line 46 - line 47 * * column 7, line 58 - column 8, line 3 * * column 1, line 47 - line 54 * * column 8, paragraph [0034] * * column 9, line 17 - line 20 * * column 8, line 32 - line 38 * * column 8, line 10 - line 15 * * column 9, paragraph [0039] - column 10, paragraph [0041] * * column 9, paragraph [0036] *	1-11	INV. G06F17/30
A	Anonymous: "Percent-encoding - Wikipedia, the free encyclopedia", 14 March 2013 (2013-03-14), XP055130713, Retrieved from the Internet: URL:http://en.wikipedia.org/w/index.php?title=Percent-encoding&oldid=544059828 [retrieved on 2014-07-22] * the whole document *	1-11	TECHNICAL FIELDS SEARCHED (IPC) G06F
A	US 2004/044768 A1 (TAKAHASHI KOICHI [JP]) 4 March 2004 (2004-03-04) * the whole document *	1-11	
A	US 2012/117253 A1 (SCODA ENRICO [IT]) 10 May 2012 (2012-05-10) * the whole document *	1-11	
A	A Barth: "Request for Comments: 6265 - HTTP State Management Mechanism", 30 April 2011 (2011-04-30), XP055131032, Retrieved from the Internet: URL:http://tools.ietf.org/pdf/rfc6265.pdf [retrieved on 2014-07-23] * the whole document *	1-11	
The present search report has been drawn up for all claims			
Place of search The Hague		Date of completion of the search 24 July 2014	Examiner Papanikolaou, N
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 (03-02) (9/04/01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 14 16 3353

5

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

24-07-2014

10

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2363995 A1	07-09-2011	CA 2732540 A1	02-09-2011
		EP 2363995 A1	07-09-2011
		US 2011219057 A1	08-09-2011
		US 2013086157 A1	04-04-2013

US 2004044768 A1	04-03-2004	CN 1487711 A	07-04-2004
		JP 4179535 B2	12-11-2008
		JP 2004094805 A	25-03-2004
		US 2004044768 A1	04-03-2004

US 2012117253 A1	10-05-2012	CA 2816336 A1	18-05-2012
		EP 2638681 A2	18-09-2013
		US 2012117253 A1	10-05-2012
		WO 2012064856 A2	18-05-2012

15

20

25

30

35

40

45

50

55

EPO FORM P466

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 2 807 578 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2013/112285 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2013/112285 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2013/112285 (art. 153(3) CBE).

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau



(10) International Publication Number
WO 2013/112285 A1

(43) International Publication Date
1 August 2013 (01.08.2013)

- (51) International Patent Classification:
G06F 17/21 (2006.01) G06F 15/16 (2006.01)
- (21) International Application Number:
PCT/US2013/020726
- (22) International Filing Date:
9 January 2013 (09.01.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/360,357 27 January 2012 (27.01.2012) US
- (71) Applicant: USABLENET INC. [US/US]; 28 W. 23rd St.,
6th Floor, New York, NY 10010 (US).
- (72) Inventor; and
(71) Applicant : SCODA, Enrico [IT/IT]; Via Cividina 416/3,
I-33035 Martignacco (UD) (IT).
- (74) Agents: GALLO, Nicholas et al.; LeClairRyan, A Profes-
sional Corporation, 70 Linden Oaks, Suite 210, Rochester,
NY 14625 (US).
- (81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,
RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ,
TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA,
ZM, ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,
TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,
MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

Published:

--- with international search report (Art. 21(3))

(54) Title: METHODS FOR TRANSFORMING REQUESTS FOR WEB CONTENT AND DEVICES THEREOF

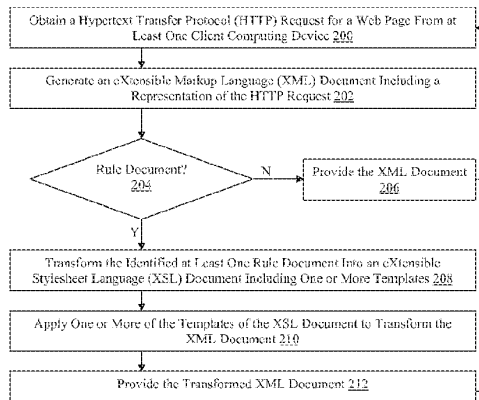


FIG. 2

(57) Abstract: A method, computer readable medium and apparatus for transforming a request for web content includes obtaining at a web content optimization computing apparatus a hypertext transfer protocol (HTTP) request for a web page from at least one client computing device. An eXtensible Markup Language (XML) document including a representation of the HTTP request is generated with the web content optimization computing apparatus. At least one rule document associated with the HTTP request is identified with the web content optimization computing apparatus. The identified at least one rule document is transformed with the web content optimization computing apparatus into an eXtensible Stylesheet Language (XSL) document including one or more templates. One or more of the templates of the XSL document are applied with the web content optimization computing apparatus to transform the XML document. The transformed XML document is provided by the web content optimization computing apparatus.

WO 2013/112285 A1

- 1 -

METHODS FOR TRANSFORMING REQUESTS FOR WEB CONTENT AND DEVICES THEREOF

[0001] This application claims the benefit of U.S. Patent Application
Serial No. 13/360,357, filed January 27, 2012, which is hereby incorporated by
5 reference in its entirety.

FIELD

[0002] This technology generally relates to methods and devices for
transforming requests for web content and, more particularly, for extending the
extensible stylesheet language (XSL) in order to manage hypertext transfer
10 protocol (HTTP) requests for web pages.

BACKGROUND

[0003] The introduction of extensible markup language (XML) and the
extensible stylesheet language (XSL) specifications has provided an easy way to
transform documents between various formats. This functionality has been
15 included in web development frameworks, giving them the ability to
automatically transform an XML document into a document with a different
format, such as hypertext markup language (HTML) or extensible hypertext
markup language (XHTML), integrating the original data with graphic layout and
user interface components. The XSL specifications are based on special
20 constructs called templates that match a single element or a set of similar elements
and rewrite them and their content based on instructions defined in the template.

[0004] Unfortunately, there is currently no effective way to define actions
to take, based on attributes of a hypertext transfer protocol (HTTP) request, or
otherwise to alter or control the flow of a web application and/or transaction
25 processing the HTTP request. Attributes of the HTTP request can include HTTP
header values and/or names of query or post parameters that correspond with web
page user interface functionality, such as a submit or purchase button, for
example. While XSL has been utilized to transform the format of web content
provided in response to an HTTP request resulting from engagement with such

- 2 -

user interface functionality, there is no effective method or device that is capable of manipulating an HTTP request based on the engaged functionality, on the type of request, or on any other attribute of the request, in order to communicate with a web application or otherwise affect the flow and/or processing of the request.

5

SUMMARY

[0005] A method for transforming a request for web content includes obtaining at a web content optimization computing apparatus a hypertext transfer protocol (HTTP) request for a web page from at least one client computing device. An extensible markup language (XML) document including a representation of the HTTP request is generated with the web content optimization computing apparatus. At least one rule document associated with the HTTP request is identified with the web content optimization computing apparatus. The identified at least one rule document is transformed with the web content optimization computing apparatus into an extensible stylesheet language (XSL) document including one or more templates. One or more of the templates of the XSL document are applied with the web content optimization computing apparatus to transform the XML document. The transformed XML document is provided by the web content optimization computing apparatus.

[0006] A computer readable medium having stored thereon instructions for transforming a request for web content comprising machine executable code which when executed by at least one processor, causes the processor to perform steps including obtaining an HTTP request for a web page from at least one client computing device. An XML document including a representation of the HTTP request is generated. At least one rule document associated with the HTTP request is identified. The identified at least one rule document is transformed into an XSL document including one or more templates. One or more of the templates of the XSL document are applied to transform the XML document. The transformed XML document is provided.

[0007] A web content optimization computing apparatus for transforming a request for web content includes one or more processors and a memory coupled

- 3 -

to the one or more processors which are configured to execute programmed instructions stored in the memory including obtaining an HTTP request for a web page from at least one client computing device. An XML document including a representation of the HTTP request is generated. At least one rule document
5 associated with the HTTP request is identified. The identified at least one rule document is transformed into an XSL document including one or more templates. One or more of the templates of the XSL document are applied to transform the XML document. The transformed XML document is provided.

[0008] This technology provides a number of advantages including
10 providing a method, a computer readable medium, and an apparatus that transforms requests for web content by utilizing XSL to manipulate an HTTP request for the content. More specifically, examples of this technology generate an XML document representing an HTTP request and apply one or more rules to the XML document, the rules being predefined and represented in one or more
15 templates of an XSL document. With this technology, the applied rules can manipulate one or more HTTP request headers or request parameters and/or the actions responsive to the HTTP request and/or flow of the application and/or web transaction configured to process the HTTP request.

BRIEF DESCRIPTION OF THE DRAWINGS

20 [0009] FIG. 1 is a block diagram of an exemplary environment with a web content optimization computing apparatus configured to transform a request for web content;

[0010] FIG. 2 is a flow chart of an exemplary method transforming a request for web content;

25 FIG. 3 is an exemplary hypertext transfer protocol (HTTP) request;

[0011] FIG. 4 is an exemplary extensible markup language (XML) document including a representation of the HTTP request of FIG. 3;

[0012] FIG. 5 is an exemplary rule document;

- 4 -

[0013] FIG. 6 is an exemplary extensible stylesheet language (XSL) document including a plurality of exemplary templates and resulting from an exemplary transformation of the rule document of FIG. 5; and

[0014] FIG. 7 is an exemplary XML document transformed according to an exemplary application of the templates of the XSL document of FIG. 6.

DETAILED DESCRIPTION

[0015] An exemplary environment 10 with a web content optimization computing apparatus 12 configured to transform requests for web content is illustrated in FIG. 1, although this technology can be implemented on other types of devices, such as one of the web server devices 16(1)-16(n), or any other server computing apparatus configured to receive and process hypertext transfer protocol (HTTP) requests, by way of example only. The exemplary environment 10 includes the web content optimization computing apparatus 12, client devices 14(1)-14(n), the web server devices 16(1)-16(n), and communication networks 18(1)-18(2), although other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can be used. This technology provides a number of advantages including providing a method, computer readable medium and an apparatus that transforms HTTP requests for web content, such as a hypertext markup language (HTML) web page, for example, in order to manipulate one or more actions taken based on the HTTP request headers and/or parameters.

[0016] Referring more specifically to FIG. 1, the web content optimization computing apparatus 12 includes a central processing unit (CPU) or processor 13, a memory 15, and an interface system 17 which are coupled together by a bus 19 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor 13 in the web content optimization computing apparatus 12 executes a program of stored instructions for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

- 5 -

[0017] The memory 15 in the web content optimization computing apparatus 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 13, can be used for the memory 15 in the web content optimization computing apparatus 12.

[0018] The interface system 17 in the web content optimization computing apparatus 12 is used to operatively couple and communicate between the web content optimization computing apparatus 12 and the client devices 14(1)-14(n) and the web server devices 16(1)-16(n) via the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. By way of example only, the communication networks 18(1) and 18(2) can use TCP/IP over Ethernet and industry-standard protocols, including HTTP, HTTPS, WAP, and SOAP, although other types and numbers of communication networks, such as a direct connection, a local area network, a wide area network, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[0019] Each of the client devices 14(1)-14(n) enables a user to request, receive, and interact with web pages from one or more web sites hosted by the web server devices 16(1)-16(n) through the web content optimization computing apparatus 12 via one or more communication networks 18(1), although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. Although multiple client devices 14(1)-14(n) are shown, other numbers and types of user computing systems could be used. In one example, the client devices 14(1)-14(n) comprise smart phones, personal digital assistants, computers, or mobile devices with Internet access that permit a website

- 6 -

form page or other retrieved web content to be displayed on the client devices 14(1)-14(n).

[0020] Each of the client devices 14(1)-14(n) in this example is a computing device that includes a central processing unit (CPU) or processor 20, a memory 22, user input device 24, a display 26, and an interface system 28, which are coupled together by a bus 30 or other link, although one or more of the client devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 20 in each of the client devices 14(1)-14(n) executes a program of stored instructions for one or more aspects of the present invention as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0021] The memory 22 in each of the client devices 14(1)-14(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 20 can be used for the memory 22 in each of the client devices 14(1)-14(n).

[0022] The user input device 24 in each of the client devices 14(1)-14(n) is used to input selections, such as requests for a particular website form page or to enter data in fields of a form page, although the user input device could be used to input other types of data and interact with other elements. The user input device can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of user input devices can be used.

[0023] The display 26 in each of the client devices 14(1)-14(n) is used to show data and information to the user, such as website page by way of example

- 7 -

only. The display in each of the client devices 14(1)-14(n) can be a mobile phone screen display, although other types and numbers of displays could be used depending on the particular type of client device 14(1)-14(n).

[0024] The interface system 28 in each of the client devices 14(1)-14(n) is used to operatively couple and communicate between the client devices 14(1)-14(n), the web content optimization computing apparatus 12, and the web server devices 16(1)-16(n) over the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

10 [0025] The web server devices 16(1)-16(n) provide web content such as one or more pages from one or more web sites for use by one or more of the client devices 14(1)-14(n) via the web content optimization computing apparatus 12, although the web server devices 16(1)-16(n) can provide other numbers and types of applications and/or content and can provide other numbers and types of
15 functions. Although the web server devices 16(1)-16(n) are shown for ease of illustration and discussion, other numbers and types of web server systems and devices can be used.

[0026] Each of the web server devices 16(1)-16(n) include a central processing unit (CPU) or processor, a memory, and an interface system which are
20 coupled together by a bus or other link, although each of the web server devices 16(1)-16(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations. The processor in each of the web server devices 16(1)-16(n) executes a program of stored instructions one or more aspects of the present invention as described and
25 illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[0027] The memory in each of the web server devices 16(1)-16(n) stores these programmed instructions for one or more aspects of the present invention as described and illustrated by way of the embodiments described and illustrated
30 herein, although some or all of the programmed instructions could be stored

and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor, can be used for the memory in each of the web server devices 16(1)-16(n).

[0028] The interface system in each of the web server devices 16(1)-16(n) is used to operatively couple and communicate between the web server devices 16(1)-16(n), the web content optimization computing apparatus 12, and the client devices 14(1)-14(n) via the communication networks 18(1) and 18(2), although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0029] Although embodiments of the web content optimization computing apparatus 12, the client devices 14(1)-14(n), and the web server devices 16(1)-16(n), are described and illustrated herein, each of the client devices 14(1)-14(n), the web content optimization computing apparatus 12, and the web server devices 16(1)-16(n), can be implemented on any suitable computer system or computing device. It is to be understood that the devices and systems of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0030] Furthermore, each of the systems of the embodiments may be conveniently implemented using one or more general purpose computer systems, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0031] In addition, two or more computing systems or devices can be substituted for any one of the systems in any of the embodiments. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and

performance of the devices and systems of the embodiments. The embodiments may also be implemented on computer system or systems that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0032] The embodiments may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0033] An exemplary method for transforming an HTTP request for web content with the web content optimization computing apparatus 12 will now be described with reference to FIGS. 2-7, although this technology can be executed by other types of devices, such as by one of the web server devices 16(1)-16(n) and without a web content optimization computing apparatus 12 or other proxy server, for example. Referring more specifically to FIG. 2, in step 200 the web content optimization computing apparatus 12 obtains an HTTP request for web content, such as an HTML web page, from at least one of the client devices 14(1)-14(n). The requested web content can be stored on one or more of the web server devices 16(1)-16(n), for example.

[0034] Referring to FIG. 3, an exemplary HTTP request 300 for the "http://processor.com/app/www.acme.com/sample/?a=1&b=2" uniform resource located (URL) is shown as communicated by one of the client devices 14(1)-14(n) using a web browser having a "my_browser" associated user agent identification. In this example, the "app" portion of the URL refers to the "app" application operating on the "processor.com" device, which can be the web content optimization computing apparatus 12. The "app" application is configured to

- 10 -

download the content referred to by the “www.acme.com/sample/?a=1&b=2”
portion of the URL and optionally perform some operation on the content, based
on the headers and/or query and/or post parameters of the HTTP request 300, such
as transforming the content into a desired format, such as HTML or XHTML
5 and/or a desired graphical format optimal for the browser identified by the user
agent, for example. With this technology, as described in detail below, the HTTP
request can be manipulated such that the “app” application downloads web
content referred to by a different URL than the original URL, performs a different
operation on the content than indicated by the headers and/or query and/or post
10 parameter values of the HTTP request 300, and/or performs an action different
than, or in addition to, the transformation of the content into a different format, by
way of example only.

[0035] In order to manipulate the HTTP request 300, an XML document
400 including a representation of the HTTP request is generated, at step 202, by
15 the web content optimization computing apparatus 12. The XML document 400
can include a plurality of elements, optionally nested by a request root element,
the values of which are based on the contents of the HTTP request 300. In this
example, the value of the URL element is the URL of the content to be
downloaded, without any query parameters, the value of the currentserver element
20 is the name of the device running the “app” application, such as the web content
optimization computing apparatus 12, that will process the HTTP request 300 to at
least retrieve the requested content from one of the web server devices 16, the
value of the remoteip element is the IP address of the requesting client device
14(1)-14(n), the value of the headers element is a param element list, the name
25 attribute of each element of which corresponds to each header of the HTTP
request 300 such as the host, user-agent, and accept headers and the value of each
element of which is included in the HTTP request 300, the value of the original-ua
element is the user agent header of the HTTP request 300, and the value of the
query element is a param element list, the name attribute of each element of which
30 corresponds to each query parameter of the URL associated with the HTTP
request 300 and the value of each element of which is included in the URL
associated with the HTTP request 300.

- 11 -

[0036] While empty in the exemplary XML document 400 shown in FIG. 4, a plurality of other elements can be provided in the XML document 400, and can contain value(s) based on the HTTP request 300, including a post element having a value of a param element list, the name attribute of each element of which corresponds to each post parameter of the HTTP request 300 and the value of each element of which is set forth in the HTTP request 300, a cookies element having a value of a param element list, the name attribute of each element of which corresponds to a cookie name as included in the HTTP request 300 and the value of each element of which corresponds to a cookie value as included in the HTTP request 300, and an imode element having a string value indicating whether the requesting client device 14(1)-14(n) supports i-mode services.

[0037] In other examples, the XML document 400 can contain a plurality of other elements representing actions that can be performed by the “app” application including a redirect element having a value of a URL of a location to which the HTTP request 300 is to be redirected, an auto_redirect element having a value of a string indicating the “app” application should implement an automatic redirect algorithm, such as that described in U.S. Patent Application Nos. 12/927,169 and 13/135,707, each of which is hereby incorporated by reference in its entirety, an encoding element having a string value indicating the character set to be used to read the web content requested by the HTTP request 300, a content-type element having a string value indicating the content or mime type to be used to read the web content requested by the HTTP request 300, a popup element having a value to be sent to the requesting client device 14(1)-14(n) to be displayed on the requesting client device 14(1)-14(n) instead of the requested web content, an error element having a string value including an error message to be displayed on the requesting client device 14(1)-14(n) when one or more attributes and/or values of the attributes of the HTTP request 300 are invalid, for example, and a ua element having a string value indicating a user agent to be used by the “app” application, prior to retrieving the requested web content from the web server device 16(1)-16(n), instead of the user agent indicated in the HTTP request 300.

- 12 -

[0038] In step 204, the web content optimization computing apparatus 12 determines whether a rule document exists for the HTTP request 300. The web content optimization computing apparatus 12 can identify at least one rule document 500 associated with the HTTP request 300 based on a match of at least a portion of the URL included in the HTTP request 300 or the value of any of the headers included in the HTTP request 300, for example. Accordingly, a plurality of rule documents can be stored in the memory 15 of the web content optimization computing apparatus 12 as associated, such as in a table, with one or more attributes of an HTTP request. In one example, a rule document is provided for a plurality of URLs and the web content optimization computing apparatus 12 is configured to identify the rule document applicable to the current HTTP request based on a match in the table of the URL included in the current HTTP request. In another example, a rule document is stored for one or more user agents and the web content optimization computing apparatus 12 is configured to identify the rule document applicable to the current HTTP request based on a match in the table of the value of the user agent indicated in the HTTP request or in the ua element of the XML document representing the HTTP request, for example. If no rule document is identified at step 204, the web content optimization computing apparatus 12 provides the XML document 400 at step 206, such that the requested web content can be retrieved from the web server device 16(1)-16(n) and communicated to the requesting client device 14(1)-14(n).

[0039] If a rule document 500 is identified for the HTTP request at step 204, the web content optimization computing apparatus 12 transforms, at step 208, the identified rule document 500 into an eXtensible Stylesheet Language (XSL) document 600 including one or more templates. An exemplary rule document 500 is shown in FIG. 5 as including two rule elements, each including a “for” condition satisfied based upon a match of a query or post parameter name, for example, and each also including one or more commands nested by an execute element.

[0040] In this example, the rule elements of the rule document 500 are established based on a match of a name of the query parameters included in a URL associated with the HTTP request 300, although in other examples one or

- 13 -

more rules can be applied based on a match of the name of a post parameter, or based on any other attribute of the HTTP request 300. In one example, the name of one of a query or a post parameter included in a URL associated with the HTTP request 300 corresponds to the name of user interface functionality that, when engaged by a user of one of the client computing devices 14(1)-14(n), results in generating the HTTP request 300 that is then communicated to the web content optimization computing apparatus 12.

[0041] In the example shown in FIG. 5, the parameter names included in the conditional expression are “a” and “c” and the commands in the first rule element are set-encoding and set-query-param and the command in the second rule is set-query-param. While the “c” query parameter is not included in the HTTP request 300, the “a” parameter name can indicate the name of user interface functionality such as a button that, when engaged by a user of one of the client computing devices 14(1)-14(n), causes the communication of the HTTP request to be sent to the web content optimization computing apparatus 12. In other examples, the name of the query parameter and associated user interface functionality can be “submit” or “purchase”, for example, or any other string value corresponding to a button, image, link, or any other web functionality initiating an HTTP request 300. Accordingly, a rule element, such as the first rule element in the exemplary rule document 500 of FIG. 5, can be included and can correspond to the name of the user interface functionality. The set-encoding command of the first rule element has a string value indicating the character set to be applied to the requested web content by the “app” application. The set-query-param command has a name attribute of a query parameter of the HTTP request 300 to be replaced by the value of the string indicated in the element.

[0042] Accordingly, in this example, whenever a user engages the “a” button on a web page, an HTTP request 300 for content is initiated whereby the content is stored on a web server device 16(1)-16(n) located in Japan, for example, and requires Japanese encoding for proper manipulation and/or display. Therefore, a rule element is recited in a rule document 500 conditional upon a match of the “a” query parameter name included in the HTTP request 300 and

- 14 -

including a set-encoding command with a "Shift_JIS" value indicating a Japanese character set.

[0043] In other examples, commands are set forth in one or more rule elements of the rule document 500 to manipulate one or more parameters of the HTTP request 300, including a remove-post-param command configured to remove a specified post parameter from the HTTP request 300, a remove-query param command configured to remove a specified query parameter from the HTTP request 300, a set-all-post-params command configured to replace all post parameters of the HTTP request 300 with a specified post parameter value, a set-all-query-params command configured to replace all query parameters of the HTTP request 300 with a specified query parameter value, and a set-post-param configured to set a specified post parameter of the HTTP request 300 with a specified value.

[0044] In other examples, commands are set forth in one or more rule elements of the rule document 500 to change one or more parameters of the HTTP request 300 and/or one or more parameters of an HTTP response, including a set-content-type command configured to set the content type of the requested web content including in the HTTP request 300 or an HTTP response, a set-cookie command configured to set a new cookie in the HTTP request 300 or an HTTP response, a set-header command configured to add a header to the HTTP request 300 or an HTTP response, and a set-user-agent command configured to change or set the user agent included in the HTTP request 300 or an HTTP response.

[0045] In yet other examples, commands are set forth in one or more rule elements of the rule document 500 to change the flow of a web transaction and/or an action performed by the identified "app" application including a set-auto-redirect command configured to enable an automatic redirect algorithm, as identified above, a set-error command configured to set an error message to be displayed on the requesting client device 14(1)-14(n), a set-popup command configured to display a virtual page on the requesting client device 14(1)-14(n) instead of the web content requested by the HTTP request 300, a set-redirect command configured to generate an HTTP response status code 302 to be

- 15 -

communicated to the requesting client device 14(1)-14(n), and a set-url command configured to change the URL of the requested web content included in the HTTP request 300.

[0046] Accordingly, in this example, the web content optimization
5 computing apparatus 12 identifies the rule document 500, at step 204, of FIG. 5, based on an attribute of the HTTP request 300, as represented by the XML document 400, generated at step 202, and, in order to apply the rule(s) to the HTTP request 300, transforms the rule document 500, at step 208, into an XSL document 600 including one or more templates, an example of which is shown in
10 FIG. 6. The XSL document 600 includes at least one template corresponding to each rule element included in the rule document 500 and, optionally, each command included in each rule element.

[0047] In the exemplary rule document 500 of FIG. 5, a first rule element is associated with a query or post parameter value of "a" and a second rule is
15 associated with a query or post parameter value of "c". The first rule includes a first command to set the encoding of the character set of the requested web content to "Shift_JIS", a Japanese encoding, and a second command to set the value of the query parameter having a value of "b" in the initial HTTP request to the value of "JP." The second rule includes a command to set the value of the
20 query parameter having a value of "e" in the HTTP request 300 to the value of "TRANSLATE". While the HTTP request 300 does not include a query parameter having a value of "e", as shown in FIG. 3, in one example all rule elements are included as part of the transformation of the rule document 500 into the XSL document 600, and have a corresponding template in the XSL document
25 600, irrespective of whether the rule element(s) are applicable to the current HTTP request 300.

[0048] In step 210, the web content optimization computing apparatus 12 applies one or more of the templates of the XSL document 600 in order to transform the XML document 400 into the transformed XML document 700, an
30 example of which is shown in FIG. 7. The exemplary XSL document 600, resulting from a transformation of the rule document 500, includes a first template

- 16 -

with a match expression for the “a” query parameter and configured to set a value for the encoding element to “Shift_JIS” when applied to the XML document 400, as shown in the resulting transformed XML document 700. The XSL document 600 further includes a second template with a match expression for the “a” query parameter and configured to change the value of the “b” query parameter from “2” to “JP” when applied to the XML document 400, as shown in the resulting transformed XML document 700. The XSL document 600 further includes a third template with a match expression for a “c” query parameter which is not satisfied by the current HTTP request 300. The rule elements included in the rule document 500 are exemplary only and the rule elements can include any number and type of commands as identified above, or any other command. Additionally, the associated templates of the XSL document 600, resulting from a transformation of the rule elements of the rule document 500, can change and/or set values for any element of the HTTP request 300.

15 [0049] In step 212, the web content optimization computing apparatus 12 provides the transformed XML document 700, such as to the “app” application identified by the URL associated with the HTTP request 300 and configured to process and service the HTTP request 300 according to the transformed XML document 700, including by retrieving the requested web content from the web server device 16(1)-16(n) and communicating it to the requesting client computing device 14(1)-14(n).

[0050] Accordingly, in this example, the “app” application interprets the HTTP request represented by the transformed XML document 700 to retrieve the requested content as identified by the “http://www.acme.com/sample” value of the url element and encodes the requested content with the Shift_JIS character set as indicated by the value of the encoding element, thereby overriding any other encoding that may have been applied by the “app” application. Accordingly, in this example, the HTTP request 300 is manipulated, based on predefined rules, to ensure the proper display of the requested web content on the client device 14(1)-14(n).

- 17 -

[0051] Accordingly, as illustrated and described herein this technology provides a number of advantages including providing a method, a computer readable medium, and an apparatus that transforms requests for web content by utilizing XSL to manipulate an HTTP request for the content. More specifically, 5 examples of this technology identify a rule document based on one or more attributes of an HTTP request, transform the rule document into an XSL document, and apply the XSL document to an XML representation of the HTTP request. With this technology, one or more HTTP request headers, one or more HTTP request parameters and/or parameter values, and/or one or more actions 10 performed in response to an HTTP request can be manipulated based on one or more predefined rules, thereby enabling control over the flow of a web application and/or transaction.

[0052] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is 15 intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of 20 processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

- 18 -

CLAIMS

What is claimed is:

1. A method for transforming a request for web content, comprising:
 - 5 obtaining at a web content optimization computing apparatus a hypertext transfer protocol (HTTP) request for a web page from at least one client computing device;
 - generating with the web content optimization computing apparatus an eXtensible Markup Language (XML) document including a representation of the HTTP request;
 - 10 identifying with the web content optimization computing apparatus at least one rule document associated with the HTTP request;
 - transforming with the web content optimization computing apparatus the identified at least one rule document into an eXtensible Stylesheet Language (XSL) document including one or more templates;
 - 15 applying with the web content optimization computing apparatus one or more of the templates of the XSL document to transform the XML document; and
 - providing by the web content optimization computing apparatus the transformed XML document.
 2. The method as set forth in claim 1 wherein the providing further comprises providing with the web content optimization computing apparatus the XML document when there is not at least one rule document associated with the HTTP request.
 - 25 3. The method as set forth in claim 1 wherein the XML document includes a plurality of elements selected from uniform resource locator (URL), redirect, auto__redirect, encoding, content-type, popup, error, currentserver, remoteip, headers, original-ua, ua, query, post, cookies, or imode.

- 19 -

4. The method as set forth in claim 1 wherein the at least one rule document includes at least one rule and wherein each rule includes at least one command selected from remove-post-param, remove-query-param, set-all-post-params, set-all-query-params, set-auto-redirect, set-content-type, set-cookie, set-encoding, set-error, set-header, set-popup, set-post-param, set-query-param, set-redirect, set-url, or set-user-agent.

5. The method as set forth in claim 1 wherein the rule document includes at least one rule including a conditional statement satisfied based on a match of a name of one of a query parameter or a post parameter included in a URL associated with the HTTP request wherein the name corresponds to a name of user interface functionality configured to initiate an HTTP request when engaged.

6. The method as set forth in claim 1 wherein the providing further comprises:
communicating with the web content optimization computing apparatus the transformed XML document to an application identified by a URL associated with the HTTP request; and
servicing with the web content optimization computing apparatus the HTTP request based at least in part on the transformed XML document.

7. A non-transitory computer readable medium having stored thereon instructions for transforming requests for web content comprising machine executable code which when executed by at least one processor, causes the processor to perform steps comprising:
obtaining a hypertext transfer protocol (HTTP) request for a web page from at least one client computing device;
generating an eXtensible Markup Language (XML) document including a representation of the HTTP request;
identifying at least one rule document associated with the HTTP request;

- 20 -

transforming the identified at least one rule document into an eXtensible Stylesheet Language (XSL) document including one or more templates;

5 applying one or more of the templates of the XSL document to transform the XML document; and
 providing the transformed XML document.

8. The medium as set forth in claim 7 wherein the providing further comprises providing the XML document when there is not at least one rule
10 document associated with the HTTP request.

9. The medium as set forth in claim 7 wherein the XML document includes a plurality of elements selected from uniform resource locator (URL), redirect, auto_redirect, encoding, content-type, popup, error,
15 currentserver, remoteip, headers, original-ua, ua, query, post, cookies, or imode.

10. The medium as set forth in claim 7 wherein the at least one rule document includes at least one rule and wherein each rule includes at least one command selected from remove-post-param, remove-qciry-param, set-all-
20 post-params, set-all-query-params, set-auto-redirect, set-content-type, set-cookie, set-encoding, set-error, set-header, set-popup, set-post-param, set-query-param, set-redirect, set-url, or set-user-agent.

11. The medium as set forth in claim 7 wherein the rule
25 document includes at least one rule including a conditional statement satisfied based on a match of a name of one of a query parameter or a post parameter included in a URL associated with the HTTP request wherein the name corresponds to a name of user interface functionality configured to initiate an
30 HTTP request when engaged.

12. The medium as set forth in claim 7 wherein the providing further comprises:

- 21 -

communicating the transformed XML document to an application identified by a URL associated with the HTTP request; and servicing the HTTP request based at least in part on the transformed XML document.

5

13. A web content optimization computing apparatus for transforming a request for web content, comprising:

- one or more processors;
- a memory coupled to the one or more processors which are

10 configured to execute programmed instructions stored in the memory comprising:

- obtaining a hypertext transfer protocol (HTTP) request for a web page from at least one client computing device;
- generating an eXtensible Markup Language (XML) document including a representation of the HTTP request;

15 identifying at least one rule document associated with the HTTP request;

- transforming the identified at least one rule document into an eXtensible Stylesheet Language (XSL) document including one or more templates;

20 applying one or more of the templates of the XSL document to transform the XML document; and

- providing the transformed XML document.

14. The apparatus as set forth in claim 13 wherein the providing further comprises providing the XML document when there is not at least one rule document associated with the HTTP request.

15. The apparatus as set forth in claim 13 wherein the XML document includes a plurality of elements selected from uniform resource locator (URL), redirect, auto_redirect, encoding, content-type, popup, error, currentserver, remoteip, headers, original-ua, ua, query, post, cookies, or imode.

- 22 -

16. The apparatus as set forth in claim 13 wherein the at least one rule document includes at least one rule and wherein each rule includes at least one command selected from remove-post-param, remove-query-param, set-all-post-params, set-all-query-params, set-auto-redirect, set-content-type, set-cookie, set-encoding, set-error, set-header, set-popup, set-post-param, set-query-param, set-redirect, set-url, or set-user-agent.

17. The apparatus as set forth in claim 13 wherein the rule document includes at least one rule including a conditional statement satisfied based on a match of a name of one of a query parameter or a post parameter included in a URL associated with the HTTP request wherein the name corresponds to a name of user interface functionality configured to initiate an HTTP request when engaged.

18. The apparatus as set forth in claim 13 wherein the providing further comprises:

- communicating the transformed XML document to an application identified by a URL associated with the HTTP request; and
- servicing the HTTP request based at least in part on the transformed XML document.

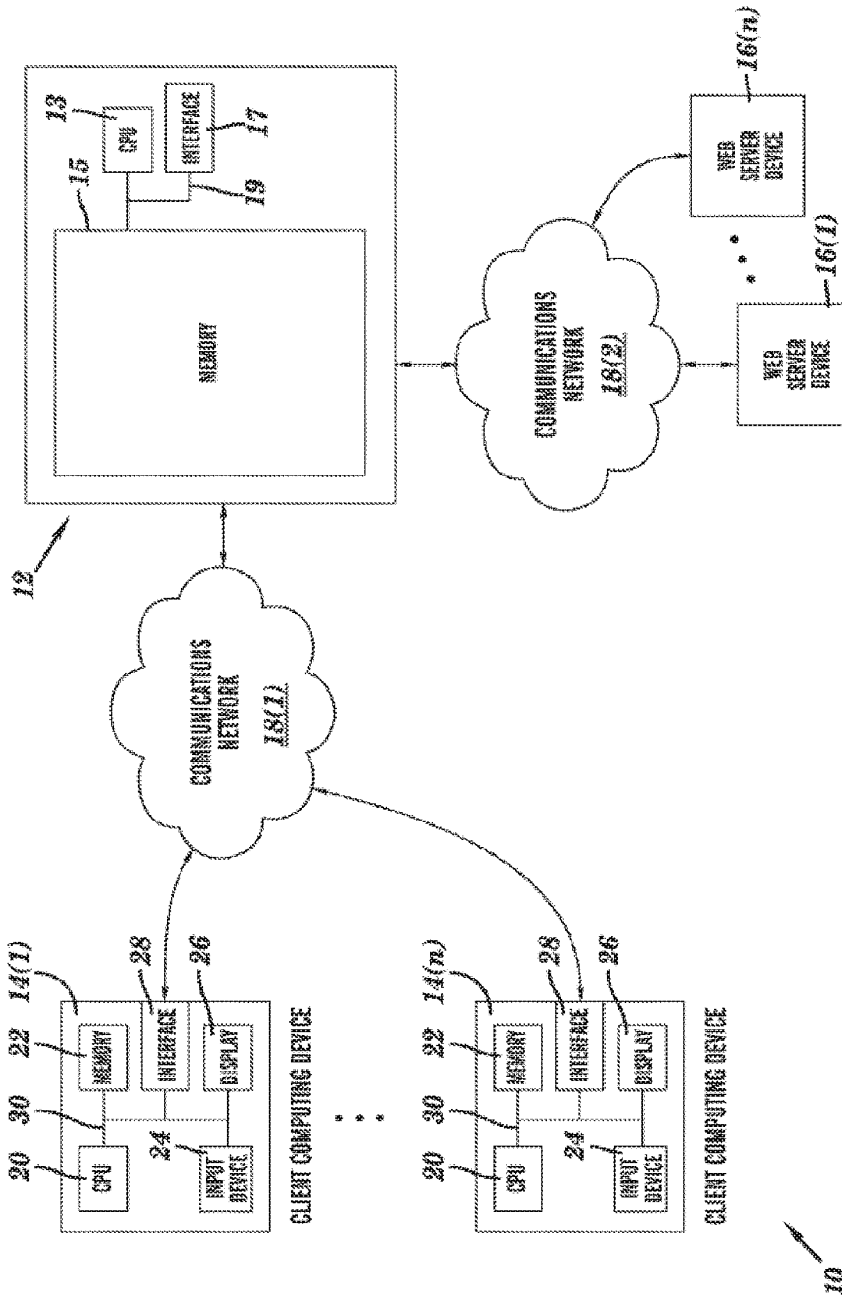


FIG. 1

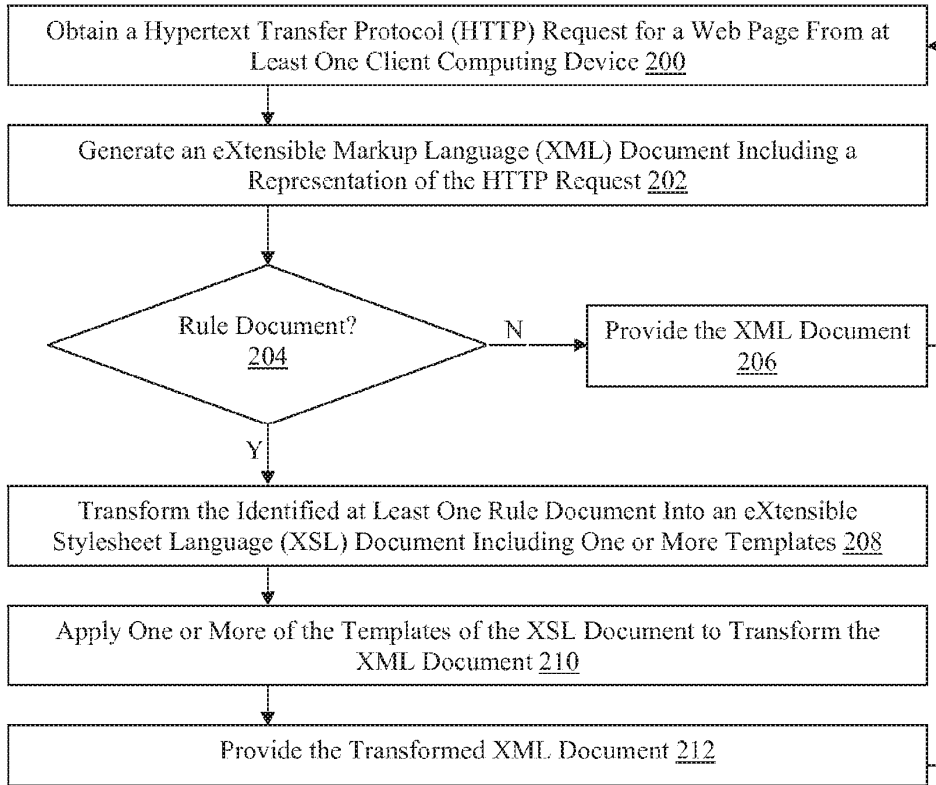


FIG. 2

3/7

```
* About to connect() to processor.com port 80 (#0)
*   Trying 63.70.164.32... connected
* Connected to processor.com (63.70.164.32) port 80 (#0)
> GET /app/www.acme.com/sample/?a=1&b=2 HTTP/1.1
> User-Agent: my_browser
> Host: processor.com
> Accept: */*
>
```

300
←

FIG. 3

4/7

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <url>http://www.acme.com/sample/</url>
  <redirect/>
  <auto_redirect/>
  <encoding/>
  <content-type/>
  <popup/>
  <error/>
  <currentserver>processor.com</currentserver>
  <remoteip>192.168.10.x</remoteip>
  <headers>
    <param name="host">processor.com</param>
    <param name="user-agent">my_browser</param>
    <param name="accept">*/*</param>
  </headers>
  <original-ua>my_browser</original-ua>
  <ua>my_browser</ua>
  <query>
    <param name="a">1</param>
    <param name="b">2</param>
  </query>
  <post></post>
  <cookies></cookies>
  <imode/>
</request>
```

400



FIG. 4

5/7

```
<request-rules>  
  <rule for="a">  
    <execute>  
      <set-encoding>Shift_JIS</set-encoding>  
      <set-query-param name="b">JP</set-query-param>  
    </execute>  
  </rule>  
  
  <rule for="c">  
    <execute>  
      <set-query-param name="e">TRANSLATE</set-query-param>  
    </execute>  
  </rule>  
</request-rules>
```

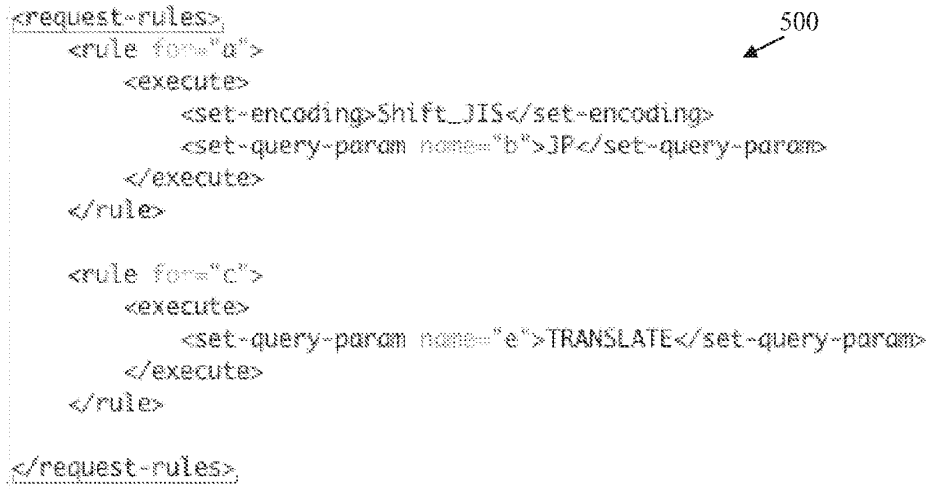
A diagram showing XML code for request rules. The code is enclosed in a dotted-line box. An arrow labeled '500' points to the first rule element. The code defines two rules: one for 'a' that sets encoding to Shift_JIS and query parameter 'b' to JP, and one for 'c' that sets query parameter 'e' to TRANSLATE.

FIG. 5

6/7

600

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:request="http://request-rules.org" version="1.0">

  <xsl:template match="encoding[request:query-param('a')!=''
                        or request:post-param('a')!='']">
    <xsl:copy>
      Shift_JIS
    </xsl:copy>
  </xsl:template>

  <xsl:template match="query[request:query-param('a')!=''
                            or request:post-param('a')!='']">
    <xsl:copy>
      <xsl:copy-of select="param[@name='b']"/>
      <param name="b">JP</param>
    </xsl:copy>
  </xsl:template>

  <xsl:template match="query[request:query-param('c')!=''
                            or request:post-param('c')!='']">
    <xsl:copy>
      <xsl:apply-templates/>
      <param name="e">TRANSLATE</param>
    </xsl:copy>
  </xsl:template>

  <xsl:template match="*">
    <xsl:copy>
      <xsl:copy-of select="@*" />
      <xsl:apply-templates/>
    </xsl:copy>
  </xsl:template>

</xsl:stylesheet>

```

FIG. 6

7/7



```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <url>http://www.acme.com/sample/</url>
  <redirect/>
  <auto_redirect/>
  <encoding>Shift_JIS</encoding>
  <content-type/>
  <popup/>
  <error/>
  <currentserver>processor.com</currentserver>
  <remoteip>192.168.10.x</remoteip>
  <headers>
    <param name="host">processor.com</param>
    <param name="user-agent">my_browser</param>
    <param name="accept">*/</param>
  </headers>
  <original-ua>my_browser</original-ua>
  <ua>my_browser</ua>
  <query>
    <param name="a">1</param>
    <param name="b">JP</param>
  </query>
  <post/>
  <cookies/>
  <imode/>
</request>
```

700

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/020726

A. CLASSIFICATION OF SUBJECT MATTER		
<i>G06F 17/21(2006.01)i, G06F 15/16(2006.01)i</i>		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F 17/21; G06F 15/16; G06F 15/00; G06F 17/00; G06F 15/12; G06F 9/00; G06F 3/12; G06F 9/45		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: web, request, XML, transform, XSL, and similar terms.		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 2003-0028377 A1 (CHIA-CHU DORLAND et al.) 06 February 2003 See paragraph [0019]-[0021]; claim 1; and figures 1 and 3.	2, 8, 14 1, 3-7, 9-13, 15-18
A	US 2005-0086584 A1 (NITHYALAKSHMI SAMPATHKUMAR et al.) 21 April 2005 See paragraphs [0049], [0051], [0070], [0085]-[0086], and [0107]-[0108]; and figures 19-20.	1-18
A	US 2004-0015891 A1 (ANNA M. ARELLANO-PAYNE et al.) 22 January 2004 See paragraphs [0022], [0025]-[0029], and [0064]; and figures 3 and 9.	1-18
A	US 2005-0021548 A1 (PHILIP L. BOHANNON et al.) 27 January 2005 See paragraphs [0005]-[0009] and [0062]-[0065]; claims 1 and 3; and figures 1, 5, and 7.	1-18
A	US 2005-0150953 A1 (BRUCE N. ALLESHOUSE) 14 July 2005 See paragraphs [0029], [0040], [0057], [0071], and [0084]; claims 1 and 9; and figures 2-3.	1-18
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 13 May 2013 (13.05.2013)		Date of mailing of the international search report 14 May 2013 (14.05.2013)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer NHO, Ji Myong Telephone No. 82-42-481-8528 

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2013/020726

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003-0028577 A1	06.02.2003	None	
US 2005-0086584 A1	21.04.2005	None	
US 2004-0015891 A1	22.01.2004	US 7047525 B2	16.05.2006
US 2005-0021548 A1	27.01.2005	None	
US 2005-0150953 A1	14.07.2005	AU 2002-352652 A1	30.06.2003
		CA 2487789 A1	26.06.2003
		CN 1605081 A	06.04.2005
		CN 1605081 B	31.08.2011
		CN 1605081 C0	06.04.2005
		CN 1836201 A0	20.09.2006
		CN 1836243 A0	20.09.2006
		CN 1836243 B	08.06.2011
		EP 1456790 A1	15.09.2004
		EP 1456790 A4	09.04.2008
		EP 1639445 A2	29.03.2006
		EP 1639517 A1	29.03.2006
		EP 2275967 A2	19.01.2011
		EP 2275967 A3	23.02.2011
		EP 2282270 A2	09.02.2011
		EP 2282270 A2	24.08.2011
		EP 2284767 A2	16.02.2011
		EP 2284767 A3	27.04.2011
		JP 2005-513627 A	12.05.2005
		JP 2006-528384 A	14.12.2006
		JP 4161314 B2	08.10.2008
		KR 10-2004-0063000 A	09.07.2004
		US 2003-0136841 A1	24.07.2003
		US 2004-0094632 A1	20.05.2004
		US 2004-0149826 A1	05.08.2004
		US 2006-0138227 A1	29.06.2006
		US 2006-0249566 A1	09.11.2006
		US 2007-0125859 A1	07.06.2007
		US 2008-0030771 A1	07.02.2008
		US 2009-0321513 A1	31.12.2009
		US 2011-0303753 A1	15.12.2011
		US 6540142 B1	01.04.2003
		US 6655593 B2	02.12.2003
		US 6908034 B2	21.06.2005
		US 7025268 B2	11.04.2006
		US 7172122 B2	06.02.2007
		US 7407102 B2	05.08.2008
		US 7600685 B2	13.10.2009
		US 7959082 B2	14.06.2011
		US 7959083 B2	14.06.2011
		US 7988051 B2	02.08.2011

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2013/020726

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		WO 03-052658 A1	26.06.2003
		WO 2005-001601 A2	06.01.2005
		WO 2005-001601 A3	14.04.2005
		WO 2005-122057 A1	22.12.2005

Form PCT/ISA/210 (patent family annex) (July 2009)



(11) **EP 2 827 261 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
21.01.2015 Bulletin 2015/04

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

(21) Application number: **14176090.0**

(22) Date of filing: **08.07.2014**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(72) Inventor: **Scoda, Enrico**
33035 MARTIGNACCO (IT)

(30) Priority: **09.07.2013 US 201313937596**

(74) Representative: **Gervasi, Gemma et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(54) **Method for bundling images and devices thereof**

(57) A method, non-transitory computer readable medium, and web server device that obtains a web page comprising a plurality of image elements each including a source attribute having a value identifying an image. Each of the image elements is modified to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data URI. A reference to a executable file is inserted into

the web page. The web page is sent to the client device and a request from the client device for the executable file is received. The executable file is sent to the client device and is configured when executed to replace the source attribute value of each of the image elements with a data URI of an image identified by the respective data attribute value.

EP 2 827 261 A1

Description**FIELD**

[0001] This technology generally relates to methods and devices for optimizing transmission of web page images and, more particularly, methods for bundling images and devices thereof.

BACKGROUND

[0002] Many web sites are increasingly sophisticated and provide rich multimedia experiences for users. Often, the multimedia content of a web page includes a significant number of images which are sent across a communication network. For example, product catalog and social network web pages are generally image-intensive, although many other types of web pages also have significant graphical content. The images of a web page are generally each retrieved through a hypertext transfer protocol (HTTP) request sent from a client device while rendering the web page. However, many networks have high latency and require a significant amount of time to transmit each request and server response including one of the requested images. Latency is a particularly significant issue with respect to networks generally utilized by mobile devices.

[0003] In order to reduce the number of HTTP requests sent by a client device for the images of a web page, web servers can parse web pages requested by client devices to identify referenced images, retrieve the images, generate an encoding of each of the images, and modify the web pages to include the encoding of the images in-line prior to sending the web pages to the requesting client devices. The embedded encoding can be a base 64 representation of each image and can be included in the web page according to a data uniform resource identifier (URI) scheme, for example. By including the referenced images in-line, web pages can be rendered by web browsers of client devices without requiring an HTTP request and response for each of the images.

[0004] While including images in-line can reduce the time required to render a web page, particularly in high latency communication networks, there are several drawbacks to this approach. For example, the images are not cached separately from the web page that includes the corresponding in-line encoding. Accordingly, every time a change is made to the web page, and a cached version of the web page becomes invalid, the images must again be encoded and embedded in-line. Another drawback is that multiple copies of the encoded version of images that are referenced more than once in a web page are generated, while externally-referenced images are downloaded only once irrespective of the number of references to the images in the web page.

SUMMARY

[0005] A method for bundling images includes obtaining, with a web server, a web page requested by a client device, the web page comprising a plurality of image elements each including a source attribute having a value identifying an image. Each of the plurality of image elements is modified, with the web server, to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data uniform resource indicator (URI). With the web server, a reference to an executable file is inserted into the requested web page, the requested web page is sent to the client device, and then a request from the client device for the executable file is received. The executable file is sent with the web server to the client device in response to the request and is configured when executed to replace the source attribute value of each of the plurality of image elements with a data URI of an image identified by the respective data attribute value.

[0006] A non-transitory computer readable medium having stored thereon instructions for bundling images comprising machine executable code which when executed by a processor, causes the processor to perform steps including obtaining a web page requested by a client device, the web page comprising a plurality of image elements each including a source attribute having a value identifying an image. Each of the plurality of image elements is modified to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data URI. A reference to a executable file is inserted into the web page, the web page is sent to the client device, and a request from the client device for the executable file is received. The executable file is sent to the client device in response to the request and is configured when executed to replace the source attribute value of each of the plurality of image elements with a data URI of an image identified by the respective data attribute value.

[0007] A web server device includes a processor coupled to a memory and configured to execute programmed instructions stored in the memory including obtaining a web page requested by a client device, the web page comprising a plurality of image elements each including a source attribute having a value identifying an image. Each of the plurality of image elements is modified to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data URI. A reference to a executable file is inserted into the web page, the web page is sent to the client device, and a request from the client device for the executable file is received. The executable file is sent to the client device in response to the request and is configured when executed to replace the source attribute value of each of the plurality of image elements with a data URI of an image identified by the respective data attribute value.

[0008] This technology provides a number of advan-

tages including methods, non-transitory computer readable medium, and a web server device that mitigates the negative affects of high network latency with respect to retrieval of web page images, while also reducing storage overhead. With this technology, web page images are encoded, bundled, and included in-line in web pages by a retrieved executable file. Advantageously, only one copy of each of the image encodings is stored irrespective of the number of references to the images in the web page. Additionally, the image encodings are stored separately from the associated web pages. Therefore, changes made to a web page that render a cached version invalid do not necessitate a subsequent downloading and encoding of the images of the web page.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]

FIG. 1 is a block diagram of an environment with an exemplary web content proxy server;

FIG. 2 is a flow chart of an exemplary method for bundling web page images;

FIG. 3 is exemplary hypertext markup language (HTML) code fragment defining a portion of a web page that references multiple images;

FIG. 4 is an exemplary version of the HTML code fragment of FIG. 3 with image elements modified to include a data uniform resource identifier (URI) of a default image and a respective data attribute; and

FIG. 5 is exemplary JavaScript code fragment configured to insert, when executed by a client device, a data URI of the images referenced by the HTML code fragment of FIG. 3.

DETAILED DESCRIPTION

[0010] An exemplary environment 10 with a web content proxy server 12 coupled to client devices 14(1)-14(n) and server devices 16(1)-16(n) by communication networks 18(1)-18(2) is illustrated in FIG. 1. Other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can also be used. This technology provides a number of advantages including providing methods, non-transitory computer readable medium, and devices for bundling and in-lining web page images to reduce the number of communications required to render the images on a client device while also reducing storage overhead.

[0011] Referring more specifically to FIG. 1, the web content proxy server 12 includes a central processing unit (CPU) 20 or processor, a memory 22, and an input/output device 24, which are coupled together by a

bus 26 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The CPU 20 in the web content proxy server 12 executes a program of stored instructions one or more aspects of the present invention as described and illustrated by way of the embodiments herein, although the CPU 20 could execute other numbers and types of programmed instructions.

[0012] The memory 22 in the web content proxy server 16 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the CPU 20, can be used for the memory 22 in the web content proxy server 12.

[0013] The input/output device 24 in the web content proxy server 12 is used to operatively couple and communicate between the web content proxy server 12, client devices 14(1)-14(n), and server devices 16(1)-16(n) via the communication networks 18(1)-18(2). One or more of the communication networks 18(1)-18(2) can include one or more networks, such as one or more local area networks (LANs) and/or wide area networks (WANs). By way of example only, the communication networks can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hard-wire communication technology, each having their own communications protocols, can be used.

[0014] The client devices 14(1)-12(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the server devices 16(1)-16(n) through the web content proxy server 12 via the communication network 18(1), although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. In some examples, the client devices 14(1)-14(n) comprise mobile computing devices with Internet access that enable web pages and other content stored by the server devices 16(1)-16(n) to be retrieved and rendered. By way of example only, the client devices 14(1)-14(n) can be smart phones, personal digital assistants, or computers.

[0015] Each of the client devices 14(1)-14(n) includes a CPU, a memory, an input device, a display device, and an input/output device, which are coupled together by a bus or other link, although one or more of client devices

14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The CPU in the client devices 14(1)-14(n) can execute a program of instructions stored in the memory of the client device 14(1)-14(n) for one or more aspects of the present invention as described and illustrated herein, although the CPU could execute other numbers and types of programmed instructions.

[0016] The input device in each of the client devices 14(1)-14(n) can be used to input selections, such as a request for a particular web page, although the input device could be used to input other types of requests and data and interact with other elements. The input device can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can be used.

[0017] The display device in each of the client devices 14(1)-14(n) can be used to show data and information to the user, such as web pages retrieved from the server devices 16(1)-16(n) by way of example only. The display device in each of the client devices 14(1)-14(n) can be a mobile phone screen display, although other types and numbers of displays could be used depending on the particular type of client device.

[0018] The input/output device in each of the client devices 14(1)-14(n) can be used to operatively couple and communicate between the client devices 14(1)-14(n), the web content proxy server 12, and the server devices 16(1)-16(n) over the communication networks 18(1)-18(2).

[0019] Each of the server devices 16(1)-16(n) provides content including web pages for use by one or more of the client devices 14(1)-14(n) via the web content proxy server 12, although the server devices 16(1)-16(n) can provide other numbers and types of content and perform other functions. Each of the server devices 16(1)-16(n) can include a CPU, a memory, and an input/output device, which are coupled together by a bus or other link, although each of the server devices 16(1)-16(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations.

[0020] The CPU in each of the server devices 16(1)-16(n) executes a program of instructions stored in the memory of the server devices 16(1)-16(n) for one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the CPU could execute other numbers and types of programmed instructions.

[0021] The input/output device in each of the server devices 16(1)-16(n) is used to operatively couple and communicate between the server devices 16(1)-16(n), the web content proxy server 12, and the client devices 14(1)-14(n) via the communication networks 18(1)-18(2).

[0022] Although embodiments web content proxy server 12, the client devices 14(1)-14(n), and the server devices 16(1)-16(n) are described and illustrated herein, each of the web content proxy server 12, the client de-

vices 14(1)-14(n), and the server devices 16(1)-16(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0023] Furthermore, each of the devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0024] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0025] The embodiments may also be embodied as one or more non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a CPU, cause the CPU to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0026] An exemplary method for bundling images will now be described with reference to FIGS. 1-5. In this example, in step 200, the web content proxy server 12 receives a request for a web page from one of the client devices 14(1)-14(n). While in this example, steps 200-224 illustrated in FIG. 2 are performed by the web content proxy server 12, one or more of steps 200-224 could be performed by one of the server devices 16(1)-16(n) or any other web server in communication with the one of the client devices 14(1)-14(n) through one or more of the communication networks 18(1) and 18(2). Accordingly, in this example, the request can be a hypertext transfer protocol (HTTP) request for a web page stored by one of the server devices 16(1)-16(n).

[0027] Upon receipt of the request from one of the client devices 14(1)-14(n), the web content proxy server 12 obtains the requested web page. In this example, the web content proxy server 12 obtains the requested web page

by retrieving the requested web page from the one of the server devices 16(1)-16(n) on behalf of the requesting one of the client devices 14(1)-14(n). In other examples, the requested web page can be obtained by retrieving the requested web page from local memory, such as the memory 22, or by generating the requested web page, for example.

[0028] In this example, the requested web page is a hypertext markup language (HTML) document, a fragment 300 of which is illustrated in FIG. 3. The fragment 300 of the HTML document references multiple images including "fb.png" and "twitter.png". The "fb.png" and "twitter.png" images are stored in the "icons" directory, as indicated in the source attribute values of the image elements 302 and 304.

[0029] In step 202, the web content proxy server 12 determines whether there is an image element with a source attribute value identifying a directory with a version file. In the fragment 300, the directory identified by the source attribute value of each of the image elements 302 and 304 is the "icons" directory. Accordingly, the web content proxy server 12 determines whether the icons directory in this example includes a version file, which can be a text file with a default file name, for example, although other file formats can also be used. The version file is created and stored in the icons directory by a developer of the web page and is used to indicate that any images in the icons directory should be bundled when provided to a client device. The version file can also be used for other functions, as described and illustrated later.

[0030] In some examples, not all image elements of the requested web page are bundled and have a source attribute value identifying a directory with a version file. For example, directories in which only one image is stored will require the same number of communications to retrieve the image as required to retrieve an image bundle, as described and illustrated in more detail later. In other examples, image elements may reference images that are not bundled for any number of other reasons according to developer preference and, in yet other examples, all images of a web page can be bundled.

[0031] Accordingly, if the web content proxy server 12 determines in step 202 that there is an image element with a source attribute value identifying a directory with a version file, then the Yes branch is taken to step 204. In step 204, the web content proxy server modifies the image element to include a data attribute and to replace the source attribute value with a data uniform resource identifier (URI).

[0032] In this example, the inserted data attribute has a value that is equivalent to the source attribute value of the image element. Referring to FIG. 4, an exemplary fragment 400 is illustrated. In the fragment 400, the image element 302 is modified to include a data attribute "data-bundle="icons/fb.png", among other modifications described and illustrated later, resulting in image element 402. Accordingly, image element 402 includes a data at-

tribute having a value equivalent to the source attribute value "icons/fb.png" of the image element 302. Other methods of preserving the source attribute value of the image element can also be used.

[0033] Additionally, in step 204, the web content proxy server 12 replaces the source attribute value of the image element with a data URI. The data URI is an encoded version of an image that will be rendered in place of the image identified by the source attribute value of the image element 302. The data URI can be a base 64 encoding, although any other encoding can also be used. The data URI is effectively a placeholder, as described and illustrated in more detail later.

[0034] In this example, the source attribute value of the image element 302 is replaced with a data URI of a spacer graphic interchange format (GIF) transparent image in the element 402, although a data URI of any other default image can also be used. By using a spacer GIF image that is transparent, the user of the requesting one of the client devices 14(1)-14(n) will not see the image when the web page is rendered. Additionally, the overhead of including the data URI of the spacer GIF image is minimal.

[0035] In step 206, the web content proxy server 12 determines whether a script reference associated with the directory identified by the value of the data attribute included in the image element in step 204 has been previously inserted into the web page. A script reference will not have been previously inserted for each directory first encountered by the web content proxy server 12 in a source attribute value of an image element.

[0036] In this example, the directory identified by the data attribute of element 402, as inserted into the image element 302, is the "icons" directory. Since the "icons" directory is encountered for the first time in a first iteration in this example, the script reference will not have been previously included. If the web content proxy server 12 determines that a script reference associated with the directory identified by the value of the data attribute included in the image element in step 204 has not been previously included, then the No branch is taken to step 208. In step 208, the web content proxy server 12 inserts a reference to a JavaScript file based on the version file included in the directory identified by the data attribute value of the image element 402. The reference to the JavaScript file can be a script element having a source attribute with a value identifying the JavaScript file or a jQuery function call, for example, although other script references can also be used. The version file includes content which is used as a portion of the file name included in the script reference. The content of the version file is inserted by a developer of the web page when the version file is stored in the directory, and is used to indicate whether a new JavaScript image bundle code should be generated, as described and illustrated in more detail later.

[0037] In this example, the fragment 400 includes a script reference as script element 406 having a source

attribute value of "icons/v1.js". Accordingly, the icons directory includes a version file having content "v1" which, along with the JavaScript file extension ".js" is used to form the "v1.js" name of the JavaScript file in the source attribute value of the script element 406. Other content of the version file and other methods of naming the JavaScript file can also be used.

[0038] Upon inserting the reference to the JavaScript file in step 208, the web content proxy server 12 proceeds to step 202. In step 202, the web content proxy server 12 determines whether there is another image element with a source attribute value identifying a directory with a version file, as described and illustrated earlier. In this example, the next image element 304 of the fragment 300 also has a source attribute value identifying a directory with a version file, as the identified "icons" directory was determined to have a version file, in this example, in the previous iteration of step 202.

[0039] Accordingly, in this second iteration, the Yes branch is taken from step 202 and the image element 304 is modified in step 204 by the web content proxy server 12, as described and illustrated earlier, resulting in image element 404 of fragment 400. In this example, the data URI that replaced the source attribute value of image element 304 is the same data URI used in the prior iteration of step 206 to replace the source attribute value of image element 302, although different data URIs can also be used.

[0040] Upon replacing the source attribute value of the image element 304, the web content proxy server 12 proceeds to step 206. In step 206, the web content proxy server 12 determines whether a script reference associated with the directory identified by the value of the data attribute included in the image element in this second iteration of step 204 has been previously inserted into the web page. In this iteration, the web content proxy server 12 will determine that script element 406 was previously inserted and has a source attribute value identifying a directory ("icons") matching the data attribute value of image element 404. Accordingly, in the second iteration in this example, the Yes branch is taken by the web content proxy server 12 from step 206 to step 202.

[0041] In step 202, the web content proxy server 12 determines whether there is another image element with a source attribute value identifying a directory with a version file, as described and illustrated earlier. In the fragment 300, the only other image element 306 does not include a source attribute value identifying a directory with a version file. Instead, the source attribute value of the image element 306 is a uniform resource locator (URL) that does not identify any directory. As there are no other image elements with a source attribute value identifying a directory with a version file, the No branch is taken to step 210.

[0042] In step 210, the web content proxy server 12 sends the web page to the requesting one of the client devices 14(1)-14(n). In other examples, the web content proxy server 12 can mark the directory, or otherwise

maintain a list of directories in step 208, and include each of the script references at substantially the same time prior to sending the web page to the requesting one of the client devices 14(1)-14(n) in step 210. In these examples, the web content proxy server 12 can determine in step 206 whether the data attribute of the image element identifies a directory that has been previously marked or included in the list of directories for which a script reference is to be inserted.

[0043] In step 212, the web content proxy server 12 receives a request for a JavaScript file identified by one of the script references inserted in step 208 from the requesting one of the client devices 14(1)-14(n). The requested is sent by the one of the client devices 14(1)-14(n) upon encountering the script reference while rendering the web page.

[0044] In step 214, the web content proxy server 12 extracts a directory and file name from a path included in the request. The request can be an HTTP request including a URL path identifying the JavaScript file. In this example, the JavaScript file will be identified in the URL path of the HTTP request as "icons/v1.js" based on the source attribute value of the script reference 406 and, accordingly, the web content proxy server 12 will extract "icons" as the directory and "v1.js" as the file name from the URL path.

[0045] In step 216, the web content proxy server 12 determines whether the requested JavaScript file was previously generated. The web content proxy server 12 can determine whether the requested JavaScript file was previously generated based on whether a JavaScript file with the extracted file name is stored in the extracted directory. If the web content proxy server 12 determines that the requested JavaScript file was not previously generated, then the No branch is taken to step 218.

[0046] In step 218, the web content proxy server 12 generates a data URI map for each image in the extracted directory. Each of the data URIs can be generated by generating an encoding of each of the images, such as a base 64 encoding, for example, although other encodings can also be used. Additionally, each of the data URIs is mapped to a directory and file name corresponding to the images encoded by the data URI. Accordingly, irrespective of the number of references to each of the images in the web page, only one encoding of each of the images is generated.

[0047] In step 220, the web content proxy server 12 generates the requested JavaScript file based on the data URIs generated in step 218. The generated JavaScript file includes JavaScript code that, when executed by the one of the client devices 14(1)-14(n), is configured to replace the source attribute value of each of the image elements with one of the data URIs. The source attribute values of the image elements are replaced based on a match of the value of the data attribute of the image elements inserted in step 204 with the directory and file name mapped to one of the data URIs. Optionally, the JavaScript code is also configured to remove the data

attribute from each of the image elements.

[0048] Referring to FIG. 5, an exemplary JavaScript code fragment 500 of a JavaScript file is illustrated. In this example, the fragment 500 includes a data URI 502 mapped to the "icon/fo.png" directory and file name and a data URI 504 mapped to the "icons/twitter.png" directory and file name. The fragment 500 is configured to retrieve the data attribute of all of the image elements of the web page. For each image element with a data attribute, the fragment 500 is configured to determine whether the value of the data attribute matches a directory and file name mapped to one of the data URIs. When it is determined that the data attribute value matches a directory and file name mapped to one of the data URIs, the fragment 500 is configured to replace the source attribute value of with the one data URI.

[0049] In step 222, the web content proxy server 12 stores the generated JavaScript file in the directory extracted in step 214. The JavaScript file is named according to the file name extracted in step 214, which corresponds with the contents of the version file stored in the extracted directory, as described and illustrated earlier. In this example, the JavaScript file is stored in the "icons" directory with a file name of "v1.js".

[0050] In step 224, the web content proxy server 12 sends the JavaScript file to the one of the client devices 14(1)-14(n) in response to the request received in step 212. Accordingly, in this example the images are sent as a bundle based on the encodings included in the JavaScript file. Accordingly, only one request and response will be required for the one of the client devices 14(1)-14(n) to retrieve all of the images stored in a respective local directory associated with the requested web page. Subsequent to sending the JavaScript file to the requesting one of the client devices 14(1)-14(n) in step 224, the web content proxy server 12 then proceeds to step 212 to receive another request for a JavaScript file from the same or a different one of the client devices 14(1)-14(n).

[0051] Upon receipt of the JavaScript file, the requesting one of the client devices 14(1)-14(n) will execute the JavaScript file resulting in the replacement of the source attribute value for one or more of the image elements of the associated web page with a data URI, as described and illustrated earlier. Subsequent to replacing the source attribute value(s), the one of the client devices 14(1)-14(n) will render the image encoded in and defined by the data URI that replaced the source attribute value for each of the image elements. Accordingly, in this example, source attribute values of the image elements 402 and 404, previously including the data URI of the default spacer GIF image, will be replaced with the data URIs 502 and 504 of the "fb.png" and "twitter.png" images, respectively.

[0052] Referring back to step 216, if the web content proxy server 12 determines that the requested JavaScript file has been previously generated, then the Yes branch is taken to step 224 and the requested JavaScript file is sent to the requesting one of the client devices 14(1)-

14(n). Whenever a change is made to the web page by a developer of the web page (e.g., addition or removal of an image), the developer can cause a new JavaScript file to be generated when requested by one of the client devices 14(1)-14(n) by changing the contents of the version file in the directory of the added or removed images (e.g., from "v1" to "v2").

[0053] By changing the contents of the version file, the reference to the JavaScript file inserted in step 208 will be to a JavaScript file with a file name not matching the file name of a previously generated JavaScript file in the directory (e.g., "v2.js"). Accordingly, the web content proxy server 12 will determine, in step 216, that the requested JavaScript file was not previously generated and will proceed to take the No branch in order to generate a new JavaScript file, which will be configured to insert the appropriate source attribute values of the image elements, as described and illustrated earlier.

[0054] Accordingly, as illustrated and described herein, this technology provides a number of advantages including providing methods, non-transitory computer readable medium, and a web content proxy server that facilitates image bundling and reduce the number of requests and responses required to retrieve web page images. Thereby, the amount of time required to render web pages is significantly reduced, particularly in high latency networks.

[0055] With this technology, data URIs including encodings of web page images are provided in a bundle through a JavaScript file that is configured to modify the web page to include the data URIs in respective image elements. Advantageously, only one encoding of each image is generated irrespective of the number of references to an image. Additionally, the encodings are stored separately from the associated web page in a JavaScript file in local directories associated with the images. Accordingly, the encodings in a directory advantageously do not have to be regenerated in response to web page changes that do not involve the images stored in the directory.

[0056] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for bundling images, the method comprising:

obtaining, with a web server, a web page requested by a client device, the web page comprising a plurality of image elements each including a source attribute having a value identifying an image;

modifying, with the web server, each of the plurality of image elements to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data uniform resource indicator (URI);

inserting, with the web server, a reference to an executable file into the requested web page, sending the requested web page to the client device, and receiving a request from the client device for the executable file; and

sending, with the web server, the executable file to the client device in response to the request, the executable file configured when executed to replace the source attribute value of each of the plurality of image elements with a data URI of an image identified by the respective data attribute value.

2. The method of claim 1, wherein the source attribute of one or more of the image elements has a value identifying a different directory than one or more other of the plurality of image elements and the reference to the executable file comprises a reference to a different executable file for each directory.

3. The method of claim 1, wherein the reference identifies the executable file based on a directory identified by the value of the source attribute included in the plurality of image elements and contents of a version file in the directory, the contents including at least a portion of a file name of the executable file.

4. The method of claim 1, wherein the reference to the executable file is a script element or a jQuery function call, the script element comprising a source attribute with a value identifying the executable file.

5. The method of claim 1, wherein the sending the executable file further comprises:

extracting at least a directory and a file name from a path included in the request received from the client device for the executable file; determining whether the executable file is stored in the directory based on the file name; and generating and storing the executable file, when it is determined that the executable file is not stored in the directory, the generating the exe-

cutable file further comprising generating a data URI for each image in the directory.

6. A non-transitory computer readable medium having stored thereon instructions for bundling images comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:

obtaining a web page requested by a client device, the web page comprising a plurality of image elements each including a source attribute having a value identifying an image;

modifying each of the plurality of image elements to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data uniform resource indicator (URI);

inserting a reference to an executable file into the requested web page, sending the requested web page to the client device, and receiving a request from the client device for the executable file; and

sending the executable file to the client device in response to the request, the executable file configured when executed to replace the source attribute value of each of the plurality of image elements with a data URI of an image identified by the respective data attribute value.

7. The medium of claim 6, wherein the source attribute of one or more of the image elements has a value identifying a different directory than one or more other of the plurality of image elements and the reference to the executable file comprises a reference to a different executable file for each directory.

8. The medium of claim 6, wherein the reference identifies the executable file based on a directory identified by the value of the source attribute included in the plurality of image elements and contents of a version file in the directory, the contents including at least a portion of a file name of the executable file.

9. The medium of claim 6, wherein the reference to the executable file is a script element or a jQuery function call, the script element comprising a source attribute with a value identifying the executable file.

10. The medium of claim 6, wherein the sending the executable file further comprises:

extracting at least a directory and a file name from a path included in the request received from the client device for the executable file; determining whether the executable file is stored in the directory based on the file name; and generating and storing the executable file, when

it is determined that the executable file is not stored in the directory.

11. A web server device, comprising:

a processor coupled to a memory and configured to execute programmed instructions stored in the memory comprising:

obtaining a web page requested by a client device, the web page comprising a plurality of image elements each including a source attribute having a value identifying an image;

modifying each of the plurality of image elements to insert a data attribute having a value of the respective source attribute value and to replace the source attribute value with a data uniform resource indicator (URI);

inserting a reference to a executable file into the requested web page, sending the requested web page to the client device, and receiving a request from the client device for the executable file; and

sending the executable file to the client device in response to the request, the executable file configured when executed to replace the source attribute value of each of the plurality of image elements with a data URI of an image identified by the respective data attribute value.

12. The device of claim 11, wherein the source attribute of one or more of the image elements has a value identifying a different directory than one or more other of the plurality of image elements and the reference to the executable file comprises a reference to a different executable file for each directory.

13. The device of claim 11, wherein the reference identifies the executable file based on a directory identified by the value of the source attribute included in the plurality of image elements and contents of a version file in the directory, the contents including at least a portion of a file name of the executable file.

14. The device of claim 11, wherein the reference to the executable file is a script element or a jQuery function call, the script element comprising a source attribute with a value identifying the executable file.

15. The device of claim 11, wherein the sending the executable file further comprises:

extracting at least a directory and a file name from a path included in the request received from the client device for the executable file;

determining whether the executable file is stored in the directory based on the file name; and generating and storing the executable file, when it is determined that the executable file is not stored in the directory.

5

10

15

20

25

30

35

40

45

50

55

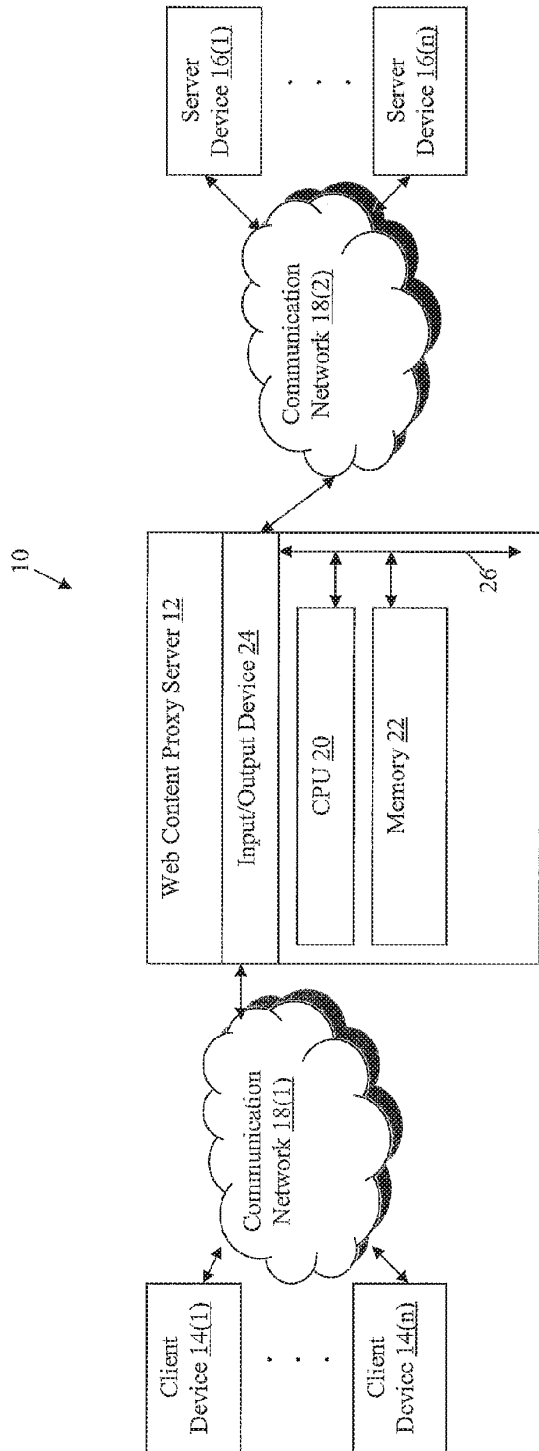


FIG. 1

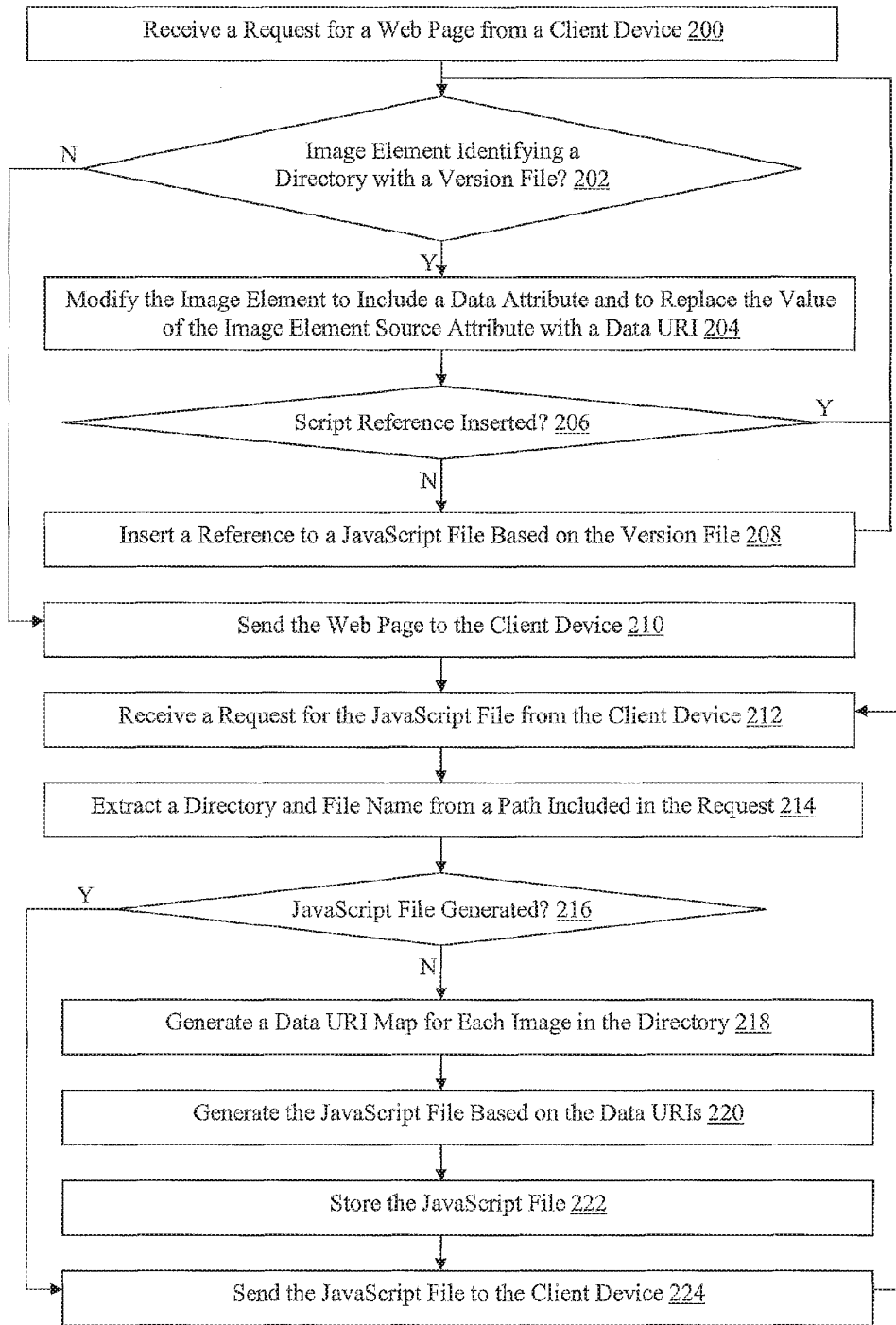


FIG. 2

```

300
<body>
  <div id="nav">
    
    
  </div>
  <div id="content">
    
  </div>
</body>
302
304
306

```

FIG. 3

```

400
<body>
  <div id="nav">
    
    
  </div>
  <div id="content">
    
  </div>
  <div>
    <script type="text/javascript" src="icons/v1.js"></script>
  </div>
</body>
402
404
406

```

FIG. 4

```

502
function unBundleTag(tag) {
  var a = {"icons/fb.png": 'data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAA...',
    "icons/twitter.png": 'data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAA...'}
  var elts=document.getElementsByTagName(tag);
  for (var i=0; i<elts.length; i++) {
    var src=a[elts[i].getAttribute('data-bundle')];
    if (src){
      elts[i].removeAttribute('data-bundle');
      elts[i].src = src;
    }
  }
}
504
function unBundle() {
  unBundleTag('img');
  unBundleTag('input');
};
unBundle();
500

```

FIG. 5



EUROPEAN SEARCH REPORT

Application Number
EP 14 17 6090

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	David Walsh: "JavaScript Canvas Image Conversion", 8 May 2012 (2012-05-08), pages 1-6, XP055154676, INTERNET Retrieved from the Internet: URL:http://davidwalsh.name/convert-canvas-image [retrieved on 2014-11-24] * pages 1,5,6 *	1-15	INV. G06F17/30
X	US 2010/235473 A1 (KOREN DAVID [IL] ET AL) 16 September 2010 (2010-09-16) * figures 1,3,5,6,7 * * paragraphs [0004], [0037] - [0046], [0057] - [0059] *	1-15	
			TECHNICAL FIELDS SEARCHED (IPC)
			G06F
The present search report has been drawn up for all claims			
Place of search Munich		Date of completion of the search 11 December 2014	Examiner Michalski, Stéphane
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document			

EPO FORM 1503 09/02 (P04/01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 14 17 6090

5

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EPO file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

11-12-2014

10

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010235473 A1	16-09-2010	US 2010235329 A1	16-09-2010
		US 2010235473 A1	16-09-2010

15

20

25

30

35

40

45

50

55

EPO FORM P/0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82



(11) **EP 2 851 813 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
25.03.2015 Bulletin 2015/13

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

(21) Application number: 14182403.7

(22) Date of filing: 27.08.2014

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(72) Inventor: **Scoda, Enrico**
33035 MARTIGNACCO (IT)

(74) Representative: **Gervasi, Gemma et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(30) Priority: 28.08.2013 US 201314012051

(54) **Methods for servicing web service requests using parallel agile web services and devices thereof**

(57) A method, non-transitory computer readable medium, and web content management server device that sends each of a plurality of jobs requiring execution in order to service a received web service request to one of a plurality of slave web services configured to execute

the plurality of jobs in parallel. A response from each of the plurality of slave web services is received. A web service response is generated based on the received responses. The generated web service response is provided in response to the received web service request.

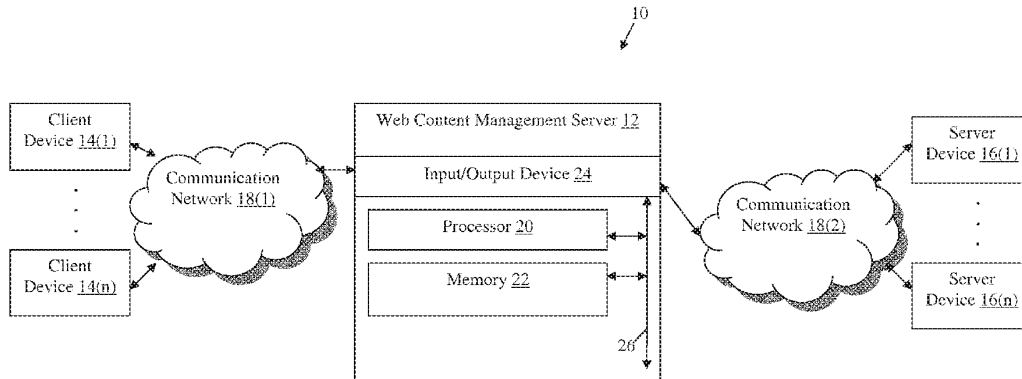


FIG. 1

EP 2 851 813 A1

Description**FIELD OF THE INVENTION**

[0001] This technology generally relates to methods and devices for optimizing delivery of web content and, more particularly, to methods for servicing web service requests using parallel agile web services and devices thereof.

BACKGROUND

[0002] Web services provide a standardized way of integrating web-based applications traditionally using extensible Markup Language (XML), SOAP, Web Services Description Language (WSDL), and/or Universal Description Discovery, for example, and Integration (UDDI) standards over an Internet Protocol (IP) backbone. XML can be used to tag data used by a web service, SOAP can be used to transfer the data, WSDL can be used for describing the web services available, and UDDI can be used for listing the available web services. Web services allow different applications located at different sources to communicate with each other efficiently and without custom coding which can require a significant amount of resources. Additionally, because communications can be in XML, web services are not tied to any operating system or programming language. Recently, JavaScript Object Notation (JSON) has emerged as a standard object-based messaging format alternative to XML that can be used with the Hypertext Transfer Protocol (HTTP) to generate web services for web applications.

[0003] Unlike traditional client/server models, web services do not provide an end user with a graphical user interface (GUI). Instead, web services share data and processes through an application interface across a network. These application interfaces are invoked and used to interpret any resulting data. Web services are increasingly popular since it is relatively easy to integrate them into applications to extend the features offered to end users. However, web service requests are also increasingly complex. For example, many web services require content from various sources to be mashed-up in order to provide a response. Accordingly, the processing of web service requests often requires significant resources, as well as a significant amount of time.

SUMMARY OF THE INVENTION

[0004] A method for servicing web service requests using parallel agile web services includes sending, with a web content management server, each of a plurality of jobs requiring execution in order to service a received web service request to one of a plurality of slave web services configured to execute the plurality of jobs in parallel. A response from each of the plurality of slave web services is received with the web content management server. A web service response is generated, with the

web content management server, based on the received responses. The generated web service response is provided, with the web content management server, in response to the received web service request

5 [0005] A non-transitory computer readable medium having stored thereon instructions for servicing web service requests using parallel agile web services comprising machine executable code which when executed by a processor, causes the processor to perform steps including sending each of a plurality of jobs requiring execution
10 in order to service a received web service request to one of a plurality of slave web services configured to execute the plurality of jobs in parallel. A response from each of the plurality of slave web services is received. A web
15 service response is generated based on the received responses. The generated web service response is provided in response to the received web service request.

[0006] A web content management server device includes a processor coupled to a memory and configured to execute programmed instructions stored in the memory including sending each of a plurality of jobs requiring execution in order to service a received web service request to one of a plurality of slave web services configured to execute the plurality of jobs in parallel. A response
20 from each of the plurality of slave web services is received. A web service response is generated based on the received responses. The generated web service response is provided in response to the received web service request.

[0007] This technology provides a number of advantages including methods, non-transitory computer readable medium, and devices that facilitate more efficient and effective processing of web service requests using parallel slave web services to retrieve content required to generate a web service response. The parallel slave web services advantageously run in an emulated JavaScript environment which is capable of executing other web service functionality implemented using JavaScript. Accordingly, with this technology, web service requests
25 can be processed by an increased number of devices, including mobile devices and tablets with special requirements, and in relatively less time.

BRIEF DESCRIPTION OF THE DRAWINGS**[0008]**

FIG. 1 is a block diagram of an environment with an exemplary web content management server;

FIG. 2 is a flow chart of an exemplary method for processing web service requests with a master web service using parallel agile web services;

FIG. 3 is a flow chart of an exemplary method for processing with parallel agile web service jobs received from a master web service;

FIG. 4 is an exemplary code fragment of an exemplary master web service configured to perform the exemplary method of FIG. 2;

FIG. 5 is an exemplary code fragment of an exemplary slave web service configured to perform the exemplary method of FIG. 3; and

FIG. 6 is an exemplary configuration file defining the role of an exemplary slave web service.

DETAILED DESCRIPTION OF THE INVENTION

[0009] An exemplary network environment 10 with a web content management server 12 coupled to client devices 14(1)-14(n) and server devices 16(1)-16(n) is illustrated in FIG. 1. In this example, the web content management server 12, client devices 14(1)-14(n), and server devices 16(1)-16(n) are coupled together by communication networks 18(1)-18(2), although other numbers and types of systems, devices, and/or elements in other configurations or network topologies can also be used. This technology provides a number of advantages including methods, non-transitory computer readable medium, and devices that facilitate more efficient and effective processing of web service requests using parallel slave web services to retrieve content required to generate a web service response.

[0010] The web content management server 12 is coupled to the client devices 14(1)-14(n) by the communication network 18(1) which can include one or more local area network(s) (LANs) and/or wide area network(s) (WANs). In this example, the web content management server 12 is further coupled to the server devices 16(1)-16(n) by the communication network 18(2), which may also include one or more LANs and/or WANs. Other network devices configured to generate, send, and receive network communications and coupled together via other topologies can also be used. While not shown, the network environment 10 also may include additional network components, such as routers, switches and other devices, which are well known to those of ordinary skill in the art and thus will not be described here.

[0011] The web content management server 12 may perform any number of functions including optimizing content retrieved from the server devices 16(1)-16(n) for display on the client devices 14(1)-14(n), for example. In this example the web content management server 12 includes a processor 20, a memory 22, and an input/output system 24, which are coupled together by a bus 26 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used.

[0012] The processor 20 in the web content management server 12 executes a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 20 could execute other numbers

and types of programmed instructions. The processor 20 of the web content management server 12 may comprise one or more central processing units or general purpose processors with one or more processing cores, for example.

[0013] The memory 22 in the web content management server 12 stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 20, can be used for the memory 22 in the web content management server 12.

[0014] The input/output system 24 in the web content management server 12 is used to operatively couple and communicate between the web content management server 12, client devices 14(1)-14(n), and server devices 16(1)-16(n), which are all coupled together via the communication networks 18(1)-18(2), although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. By way of example only, the communication networks 18(1)-18(2) can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[0015] The client devices 14(1)-14(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the server devices 16(1)-16(n) through the web content management server 12 and using the communication network 18(1), although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. In some examples, the client devices 14(1)-14(n) comprise mobile devices with Internet access that enable web pages and other content stored by the server devices 16(1)-16(n) to be retrieved and rendered. By way of example only, the client devices 14(1)-14(n) can be smart phones, personal digital assistants, tablets, or computers.

[0016] Each of the client devices 14(1)-14(n) includes a processor, a memory, an input device, a display device, and an input/output system, which are coupled together by a bus or other link, although one or more of client devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements

in other configurations. The processor in each of the client devices 14(1)-14(n) can execute a program of instructions stored in the memory the client device 14(1)-14(n) for one or more aspects of the present invention, as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0017] The input device in each of the client devices 14(1)-14(n) can be used to input selections, such as a request for a particular web page or other content stored by one or more of the server devices 16(1)-16(n), although the input device could be used to input other types of requests and data and interact with other elements. The input device can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can also be used.

[0018] The display device in each of the client devices 14(1)-14(n) can be used to show data and information to the user, such as web pages and other content retrieved from the server devices 16(1)-16(n) by way of example only. The display device in each of the client devices 14(1)-14(n) can be a mobile phone screen display, although other types and numbers of displays could be used depending on the particular type of client device.

[0019] The input/output system in each of the client devices 14(1)-14(n) can be used to operatively couple and communicate between the client devices 14(1)-14(n), the web content management server 12, and the server devices 16(1)-16(n) over the communication networks 18(1)-18(2).

[0020] Each of the server devices 16(1)-16(n) provides content including web pages and web applications for use by one or more of the client devices 14(1)-14(n) via the web content management server 12, although the server devices 16(1)-16(n) can provide other numbers and types of content and perform other functions. Each of the server devices 16(1)-16(n) can include a processor, a memory, and an input/output system, which are coupled together by a bus or other link, although each of the server devices 16(1)-16(n) can have other numbers and types of components, parts, devices, systems, and elements in other configurations.

[0021] The processor in each of the server devices 16(1)-16(n) executes a program of instructions stored in the memory of the server devices 16(1)-16(n) for one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[0022] The input/output system in each of the server devices 16(1)-16(n) is used to operatively couple and communicate between the server devices 16(1)-16(n), the web content management server 12, and the client devices 14(1)-14(n) via the communication networks 18(1)-18(2).

[0023] Although embodiments web content management server 12, the client devices 14(1)-14(n), and the server devices 16(1)-16(n) are described and illustrated

herein, each of the web content management server 12, the client devices 14(1)-14(n), and the server devices 16(1)-16(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0024] Furthermore, each of the devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0025] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0026] The embodiments may also be embodied as one or more non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0027] An exemplary method for servicing web service requests using parallel agile web services will now be described with reference to FIGS. 1-6. Referring more specifically to FIG. 2, a flow chart of an exemplary method for processing web service requests with a master web service using parallel agile web services is illustrated. In this example, in step 200, the web content management server 12 receives a web service request from one of the client devices 14(1)-14(n). The web service request can be for social media posts, blog summaries, or any other content stored by one or more of the server devices 16(1)-16(n), for example. In the example described and illustrated herein, the received web service request is for the last five social media (e.g., Facebook®) posts for each of twenty users of a social media service associated with a user (e.g., Facebook® friends) of the requesting one

of the client devices 14(1)-14(n).

[0028] In step 202, the web content management server 12 determines whether the request received in step 200 requires execution of a plurality of jobs in order to service the request. In the example described herein, the web content management server 12 may determine that the request requires execution of twenty jobs, one for each of the users for which the last five posts have been requested. In this example, the social media service will have to be called twenty times in order to retrieve the last five posts for each of the twenty users identified in the received web service request.

[0029] In other examples, the received web service request may be for the last ten posts for only one user of the social media service, in which case the web content management server 12 may determine in step 202 that a plurality of jobs are not required in order to service the request. The web content management server 12 can be configured by an administrator, for example, to determine that a received web service request requires execution of a plurality of jobs based on any criteria or type of analysis of the web service request.

[0030] Additionally, an administrator of the web content management server 12 can define a threshold number of jobs requiring execution in order to satisfy the condition in step 202. Alternatively, the condition in step 202 can be satisfied whenever the web content management server 12 determines more than one job must be executed in order to service the received web service request. If the web content management server 12 determines in step 202 that the web service request received in step 200 does not require execution of a plurality of jobs, then the web content management server 12 proceeds to step 204.

[0031] In step 204, the web content management server 12 retrieves content from one or more of the server devices 16(1)-16(n) and generates and sends a web service response to the requesting one of the client devices 14(1)-14(n). Accordingly, if the web content management server 12 determines a plurality of jobs are not required in order to service the received web service request, then the web service request is processed based on one or more sequential requests for content sent by the web content management server 12 to the server devices 16(1)-16(n) on behalf of the requesting one of the client devices 14(1)-14(n). Subsequent to servicing the web service request, or in parallel, the web content management server 12 may retrieve another web service request from one of the client devices 14(1)-14(n) in step 200.

[0032] Referring back to step 202, if the web content management server determines that the web service request received in step 200 does require execution of a plurality of jobs, then the web content management server 12 proceeds to step 206. In step 206, the web content management server 12 executes a master web service in an emulated JavaScript environment, although other types of environments could be used. The emulated

JavaScript environment can be generated, for example, as described and illustrated in U.S. Patent Application No. 12/802,670, entitled "Methods for Utilizing a JavaScript Emulator in a Web content management server and Devices Thereof," which is incorporated by reference herein in its entirety, although other methods of generating an emulated JavaScript environment can also be used.

[0033] In step 208, the master web service executed by the web content management server 12 inserts the plurality of jobs requiring execution in order to service the web service request into a queue. The queue can be maintained in the memory 22 for example, although the queue can also be stored elsewhere. In the example described earlier, the master web service inserts the twenty jobs that require a call to the social media service for the last five social media posts for one of the twenty specified users.

[0034] In step 210, the master web service executed by the web content management server 12 determines whether the queue is empty. In a first iteration, the web content management server 12 will determine in step 210 that the queue is not empty. However, in subsequent iterations, all of the jobs may have been sent by the master web service to threads configured to request execution of each of the jobs by one of a plurality of slave web services resulting in an empty queue. If the master web service executed by the web content management server 12 determines that the queue is not empty, then the No branch is taken to step 212.

[0035] In step 212, the master web service executed by the web content management server 12 determines whether a thread is available to receive one of the jobs stored in the queue. In a first iteration, the web content management server 12 will determine in step 212 that a thread is available. However, in subsequent iterations, slave web services may be executing jobs sent by each of the threads and a thread may not be therefore be available.

[0036] The total number of slave web services can be established by an administrator of the web content management server 12, for example. The number of slave web services can be based on a number of threads or processes the system administrator wants to allow the web content management server 12 to execute concurrently, although the number of slave web services can be based on any other criteria. Additionally, web services can be designated as master or slave based on a configuration established by an administrator of the web content management server 12, as described and illustrated in more detail later with reference to FIG. 6. Accordingly, if the master web service executed by the web content management server 12 determines that a slave web service is available, then the Yes branch is taken to step 214.

[0037] In step 214, the master web service executed by the web content management server 12 generates a thread and sends one of the jobs from the queue to the thread along with an index. The thread is configured to

request execution of the one of the jobs by one of the available slave web services. The index will be returned to the master web service context and used to reorder the content received from the slave web services, as described and illustrated in more detail later with reference to step 222. Upon sending one of the jobs from the queue, the master web service again determines whether the queue is empty in step 210.

[0038] Accordingly, in this example, steps 210-214 are performed for each of the jobs and the master web service proceeds to send a job from the queue to a generated thread configured to request execution of the job from one of the slave web services until the master web service determines that the queue is empty in step 210 or that no more threads are available in step 212. For example, assuming that none of the slave web services have processed a job as requested by one of the threads, in this example, the master web service will determine in an eleventh iteration of step 212 that a thread is not available.

[0039] If the master web service determines that a thread is not available, then the No branch is taken from step 212 back to step 212 and the master web service effectively waits until it determines a thread has finished processing a response sent by an associated slave web service, and is available to receive another job, prior to sending one of the jobs from the queue to the generated thread in step 214. Subsequent to generating a thread and sending another one of the jobs and an index to the generated thread in step 214, the master web service proceeds back to step 210, as described and illustrated earlier. Upon the master web service determining, in step 210, that the queue is empty, the Yes branch is taken to step 216.

[0040] In step 216, one of the threads generated by the master web service executed by the web content management server 12 receives a response from one of the slave web services. In this example, the response is a JavaScript Object Notation (JSON) response, although the response can be any type of response. The response includes content obtained from one or more of the server devices 16(1)-16(n) that satisfies the job the slave web service was requested to execute by the thread, as described and illustrated in more detail later with reference to FIG. 3.

[0041] In step 218, the one of the threads generated by the master web service executed by the web content management server 12 converts the received response into a JavaScript object and passes the JavaScript object and an index to a callback function executed in the master web service context, although the response and/or index can be converted and/or passed in other manners. The index passed in step 218 is the same index sent by the master web service to the one of the threads in step 214. In this example, the thread generated by the master web service converts the received response by using the JSON.parse() function, although other methods of converting the response received from the slave web service

can be used.

[0042] In step 220, the master web service executed by the web content management server 12 determines whether there are more responses required to satisfy the web service request received in step 200. The master web service can determine whether there are more responses based on whether it has received a response from the threads corresponding to each of the indices sent along with one of the jobs in step 214. Alternatively, one or more of steps 210-220 can be performed by the master web service within the context of a function call such that control will only return to the master web service context when all of the slave web services have returned a response which is received by the master web service in step 216.

[0043] If the master web service determines in step 220 that more responses are required to satisfy the web service requested received in step 200, then the Yes branch is taken to step 216. Steps 216-220 are repeated until the master web service determines in step 220 that no more responses from the slave web services are required. In this particular example, steps 216-220 will be repeated twenty times or one time for each of the jobs sent to the threads in step 214. If the master web service determines in step 220 that no more responses are required, then the No branch is taken to step 222.

[0044] In this example, steps 216-220 proceed in parallel with steps 210-214 such that a response may be received by the master web service from one of the threads prior to the queue being empty and all of the jobs being sent to one of the threads. Since the slave web services process the jobs in parallel, the responses from the slave web services can be received from the threads by the master web service in any order, generally depending on the time required for each slave web service to complete its job and retrieve the associated content from one or more of the server devices 16(1)-16(n).

[0045] In step 222, the master web service executed by the web content management server 12 generates a web service response based on the responses received in step 216, as converted in step 218, and the indices. As described and illustrated earlier, the threads can optionally pass the converted responses to a callback function in the master web service context in step 218. The callback function can be configured to store the JavaScript objects in a common location, optionally as associated with a corresponding one of the indices, extract one or more properties of the JavaScript objects, and/or perform any number of other functions.

[0046] Accordingly, upon control returning to the master web service context, the master web service can generate a response to the web service request received in step 200 by retrieving the JavaScript objects from the common location, for example. Once retrieved, the master web service can order the JavaScript objects, and/or content included therein, based on the index associated with each of the JavaScript objects, to formulate the web service response.

[0047] In some examples, a plurality of different callback functions configured to execute different functionality can be provided in the master web service context and used by a thread depending on the job sent to the thread by the master web service. For example, the master web service may send a first subset of the threads jobs associated with a first social media service and a second subset of the threads jobs associated with a second social media service. In this example, the master web service may provide, and the threads may pass JavaScript objects and indices to, different callback functions depending on the job received from the master web service in step 214, and the master web service can thereby mash up converted responses from the various slave web services to generate a web service response.

[0048] In step 224, the master web service executed by the web content management server 12 sends the generated web service response to the requesting one of the client device 14(1)-14(n). Upon sending the web service response with the master web service, or during any of steps 202-224, the web content management server 12 can receive another web service request from one of the client devices 14(1)-14(n) in step 200.

[0049] Referring more specifically to FIG. 3, a flow chart of an exemplary method for processing with parallel agile web service jobs received from a master web service is illustrated. In this example, in FIG. 300, one of the slave web services executed by the web content management server 12 receives one of the jobs, such as from a thread generated by the master web service, as described and illustrated earlier with reference to step 214 of FIG. 2.

[0050] In step 302, the slave web service executed by the web content management server 12 retrieves content satisfying the job received in step 300 from one or more of the server devices 16(1)-16(n). Accordingly, in this example, the slave web service retrieves the last five posts for one of the users of the social media service specified in the received job. The retrieved content can be HTML web pages, XML documents, RESTful web services, for example, or any other type of content.

[0051] In step 304, the slave web service executed by the web content management server 12 generates a response including the retrieved content. In this example, the response is a JSON response, which can be generated using the `JSON.stringify()` function, for example, although other methods of generating a JSON response and other types of responses can also be used.

[0052] In step 306, the slave web service executed by the web content management server 12 sends the response to the thread from which the job was received. After processing the response received from the slave web service, the thread from which the job was received becomes available to receive another job, as described and illustrated earlier with reference to step 212 of FIG. 2.

[0053] In this example, steps 300-306 are performed in parallel for each of the slave web services receiving one of the jobs from a thread generated by the master

web service in step 214 of FIG. 2. By breaking the web service request into a plurality of jobs, and processing the jobs in parallel with a plurality of slave web services, the time required for the web content management server 12 to generate a response to the web service request can be reduced.

[0054] Referring more specifically to FIG. 4, an exemplary code fragment 400 of an exemplary master web service configured to perform the exemplary method of FIG. 2 is illustrated. In this example, the master web service uses "counter" and "generic" callback functions, as described and illustrated earlier with reference to step 218 of FIG. 2, depending on whether a response is successfully or unsuccessfully, respectively, received by a thread from one of the slave web services. The "counter" callback function takes the index of a job and a JavaScript object, generated by a thread from a response received from slave web service, as parameters.

[0055] The total number of threads is established as "10" in this example which corresponds to the number of slave web services currently processing or available to process a job at any time. Additionally, the `executeBatch()` function can be used in this example to perform steps 216-220. Upon completion of the `executeBatch()` function, control returns to the master web service context to generate the web service response, as described and illustrated earlier with reference to step 222 of FIG. 2. Accordingly, the master web service will automatically determine whether there are more responses in step 220 of FIG. 2 based on whether the `executeBatch()` function has completed its execution.

[0056] Referring more specifically to FIG. 5, an exemplary code fragment 500 of an exemplary slave web service configured to perform the exemplary method of FIG. 3 is illustrated. Referring more specifically to FIG. 6, an exemplary configuration file 600 defining the role of the exemplary slave web service 500 is illustrated. In this example, the slave web service 500 called `catalogue-plp` is assigned the role of "slave." The configuration file 600 can be established by an administrator of the web content management server 12 and can be stored in the memory 22, for example, as described and illustrated earlier. Additionally, a corresponding configuration file can be established, or the same configuration file can be used, to define the role of a plurality of master and/or slave web services executable by the web content management sever 12.

[0057] With this technology, web service requests, such as requests requiring a large amount of content and/or content from a number of disparate sources, for example, can be divided into a plurality of jobs which are processed in parallel by slave agile web services to reduce the overall amount of time required to generate a response to the web service request. Accordingly, portions of content required to satisfy a web service request can be retrieved in parallel and mashed-up to generate a web service response. Advantageously, the process can utilize JavaScript executed in an emulated environ-

ment on behalf of client devices, including mobile computing or other devices with special requirements that may otherwise be unable to process such content.

[0058] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for servicing web service requests using parallel agile web services, the method comprising:

 sending, with a web content management server, each of a plurality of jobs requiring execution in order to service a received web service request to one of a plurality of slave web services configured to execute the plurality of jobs in parallel;
 receiving, with the web content management server, a response from each of the plurality of slave web services;
 generating, with the web content management server, a web service response based on the received responses; and
 providing, with the web content management server, the generated web service response in response to the received web service request.

2. The method of claim 1 wherein the sending further comprises sending each of the jobs and an index to one of a plurality of threads generated by a master web service executed in an emulated JavaScript environment, each of the threads configured to request execution of one of the jobs by one of the slave web services.

3. The method of claim 2 wherein each of the slave web services is further configured to retrieve content from a server device, convert the retrieved content into a JavaScript Object Notation (JSON) response, and return the JSON response.

4. The method of claim 3, wherein the generating further comprises:

 converting each of the returned JSON responses into a JavaScript object;
 passing each of the JavaScript objects and the index to a callback function, wherein the index corresponds to an order of a respective one of the jobs; and
 organizing the JavaScript objects based on the received indices.

5. The method of claim 4, wherein the passing further comprises passing each of the JavaScript objects and the index to one of a plurality of callback functions based on the job sent to each of the threads.

6. The method of claim 1, further comprising:
 determining, with the web content management server, when the received web service request requires execution of a plurality of jobs; and
 performing the executing, sending, receiving, generating, and providing steps when the determining indicates the received web service request requires execution of a plurality of jobs.

7. The method of claim 6, further comprising:
 determining, with the web content management server, when the plurality of jobs exceeds a maximum number of slave web services executable in parallel; and
 performing, with the web content management server, when the determining indicates the plurality of jobs exceeds a maximum number of slave web services executable in parallel, steps comprising:

 inserting, with the web content management server, the plurality of jobs into a queue;
 sending, with the web content management server, a number of jobs corresponding to the maximum number of slave web services executable in parallel to the plurality of slave web services from the queue; and
 sending, with the web content management server, one of the plurality of jobs remaining in the queue upon receiving each response from one of the plurality of slave web services and until the queue is empty.

8. A non-transitory computer readable medium having stored thereon instructions for servicing web service requests using parallel agile web services comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:

 sending each of a plurality of jobs requiring ex-

execution in order to service a received web service request to one of a plurality of slave web services configured to execute the plurality of jobs in parallel;
 receiving a response from each of the plurality of slave web services;
 generating a web service response based on the received responses; and
 providing the generated web service response in response to the received web service request.

- 9. The medium of claim 8, wherein the sending further comprises sending each of the jobs and an index to one of a plurality of threads generated by a master web service executed in an emulated JavaScript environment, each of the threads configured to request execution of one of the jobs by one of the slave web services.
- 10. The medium of claim 9, wherein each of the slave web services is further configured to retrieve content from a server device, convert the retrieved content into a JavaScript Object Notation (JSON) response, and return the JSON response.
- 11. The medium of claim 10, wherein the generating further comprises:

converting each of the returned JSON responses into a JavaScript object;
 passing each of the JavaScript objects and the index to a callback function, wherein the index corresponds to an order of a respective one of the jobs; and
 organizing the JavaScript objects based on the received indices.

- 12. The medium of claim 11, wherein the passing further comprises passing each of the JavaScript objects and the index to one of a plurality of callback functions based on the job sent to each of the threads.
- 13. The medium of claim 8, further having stored thereon instructions comprising machine executable code which when executed by the processor, causes the processor to perform steps further comprising:

determining when the received web service request requires execution of a plurality of jobs; and
 performing the executing, sending, receiving, generating, and providing steps when the determining indicates the received web service request requires execution of a plurality of jobs.

- 14. The medium of claim 13, further having stored thereon instructions comprising machine executable code which when executed by the processor, causes

the processor to perform steps further comprising:

determining when the plurality of jobs exceeds a maximum number of slave web services executable in parallel; and
 performing when the determining indicates the plurality of jobs exceeds a maximum number of slave web services executable in parallel, steps comprising:

inserting the plurality of jobs into a queue;
 sending a number of jobs corresponding to the maximum number of slave web services executable in parallel to the plurality of slave web services from the queue; and
 sending one of the plurality of jobs remaining in the queue upon receiving each response from one of the plurality of slave web services and until the queue is empty.

- 15. A web content management server device, comprising:

a processor coupled to a memory and configured to execute programmed instructions stored in the memory comprising:

sending each of a plurality of jobs requiring execution in order to service a received web service request to one of a plurality of slave web services configured to execute the plurality of jobs in parallel;
 receiving a response from each of the plurality of slave web services;
 generating a web service response based on the received responses; and
 providing the generated web service response in response to the received web service request.

- 16. The device of claim 15, the sending further comprises sending each of the jobs and an index to one of a plurality of threads generated by a master web service executed in an emulated JavaScript environment, each of the threads configured to request execution of one of the jobs by one of the slave web services.

- 17. The device of claim 16, wherein each of the slave web services is further configured to retrieve content from a server device, convert the retrieved content into a JavaScript Object Notation (JSON) response, and return the JSON response.

- 18. The device of claim 17, wherein the generating further comprises:

converting each of the returned JSON respons-

es into a JavaScript object;
 passing each of the JavaScript objects and the
 index to a callback function, wherein the index
 corresponds to an order of a respective one of
 the jobs; and
 organizing the JavaScript objects based on the
 received indices.

5

19. The device of claim 18, wherein the passing further
 comprises passing each of the JavaScript objects
 and the index to one of a plurality of callback func-
 tions based on the job sent to each of the threads.

10

20. The device of claim 15, wherein the processor is fur-
 ther configured to execute programmed instructions
 stored in the memory further comprising:

15

determining when the received web service re-
 quest requires execution of a plurality of jobs;
 and
 performing the executing, sending, receiving,
 generating, and providing steps when the deter-
 mining indicates the received web service re-
 quest requires execution of a plurality of jobs.

20

25

21. The device of claim 20, wherein the processor is fur-
 ther configured to execute programmed instructions
 stored in the memory further comprising:

determining when the plurality of jobs exceeds
 a maximum number of slave web services exe-
 cutable in parallel; and
 performing when the determining indicates the
 plurality of jobs exceeds a maximum number of
 slave web services executable in parallel, steps
 comprising:

30

35

inserting the plurality of jobs into a queue;
 sending a number of jobs corresponding to
 the maximum number of slave web services
 executable in parallel to the plurality of slave
 web services from the queue; and
 sending one of the plurality of jobs remain-
 ing in the queue upon receiving each re-
 sponse from one of the plurality of slave web
 services and until the queue is empty.

40

45

50

55

10

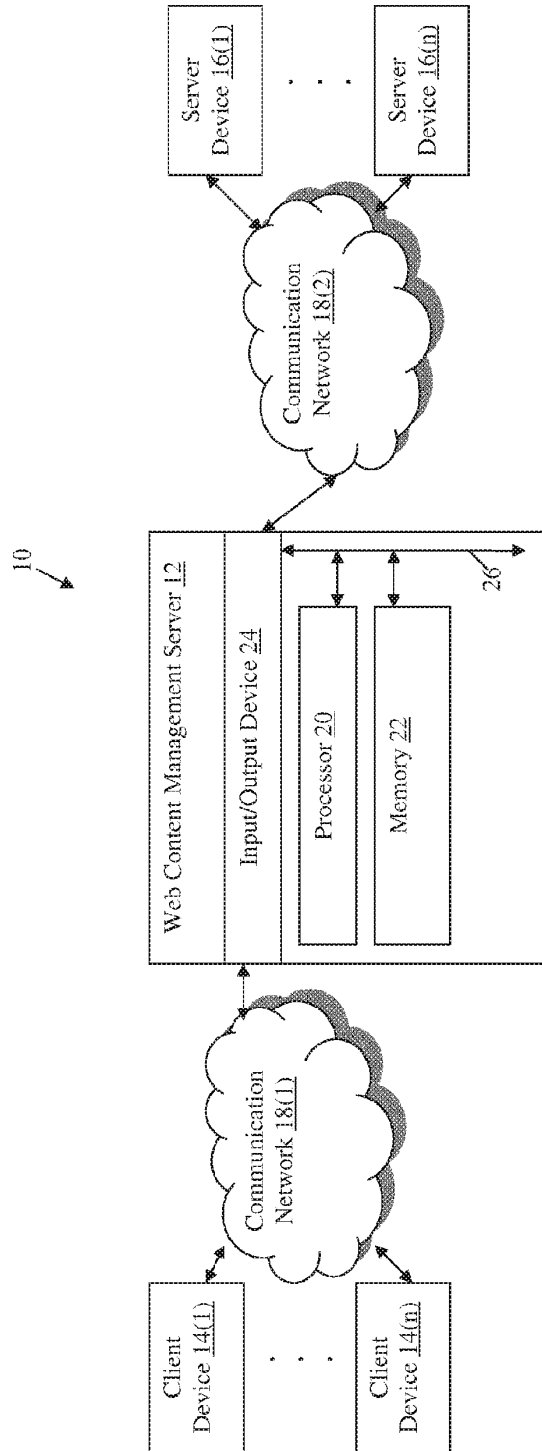


FIG. 1

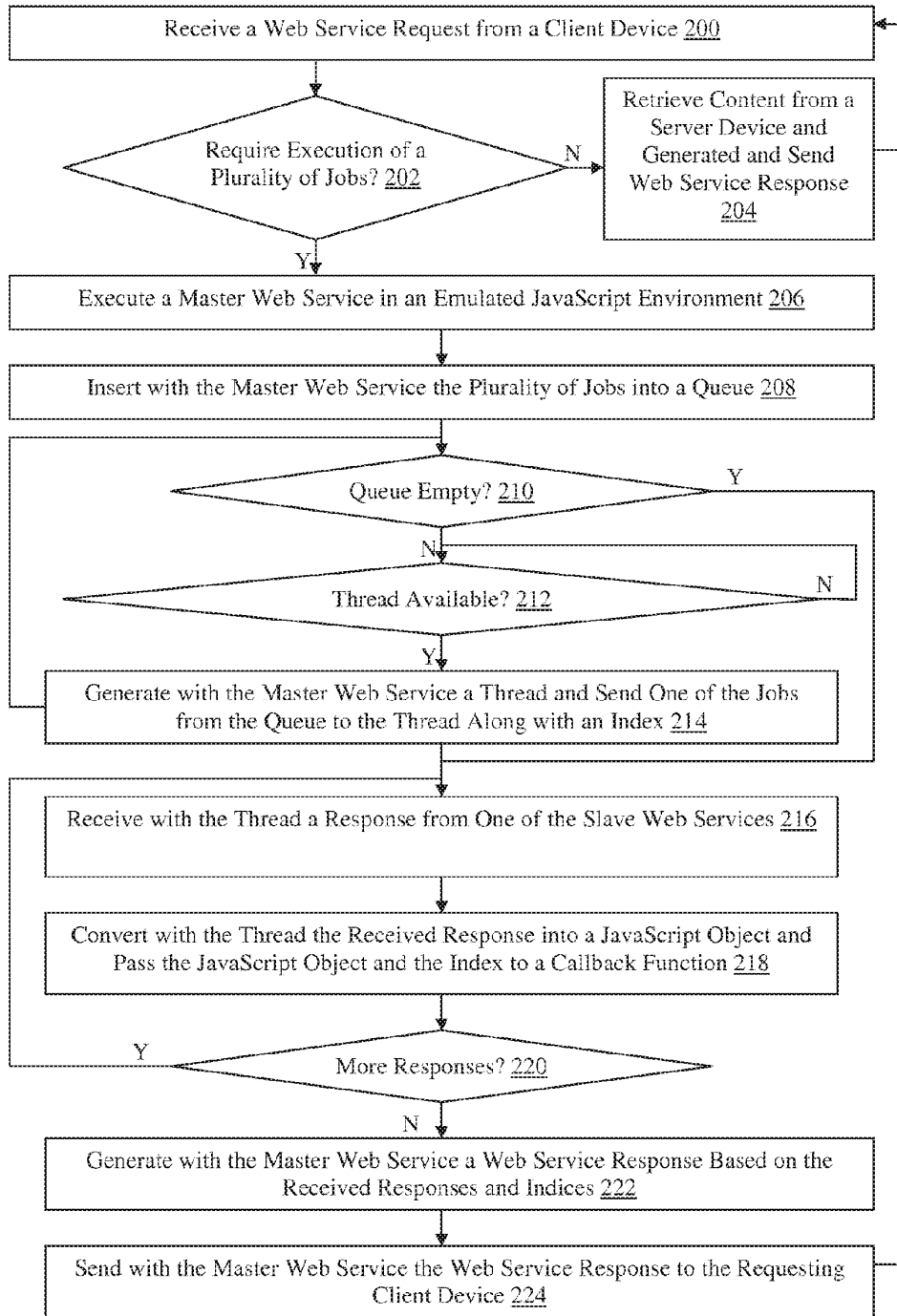


FIG. 2

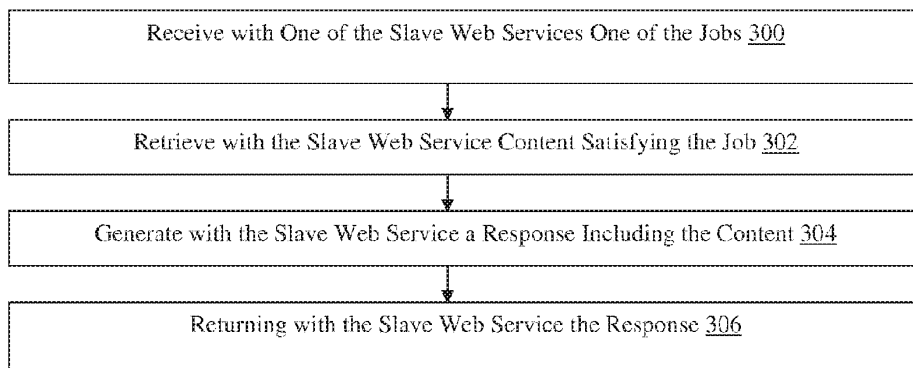


FIG. 3

```

var batch = [];
for (var i=0; i<pages; i++) {
  result.push(null);
  var o = {
    'service': 'catalogue-p-plp',           400
    'params': {'url': pages[i]},
    'cache': '1d',
    'retryTimeouts': 5,
    'success': 'counter',
    'failure': 'generic'
  }
  batch.push(o);
}
ws.executeBatch({
  'threads': 10,
  'batch': batch,
  'success': {
    'counter': function(index, obj) {
      result[index] = obj;
    }
  },
  'failure': {
    'generic': function(index, status) {
      ws.response.error("Error on page " + index + " " + status);
    }
  }
});
  
```

FIG. 4

```
function service (ws) {  
    jse.settings.httpTimeout = 30; 500  
    var url = ws.request.params['url'];  
    if (url) {  
        ws.load (url, function (ws, xhr) {  
            ws.response = JSON.stringify({  
                'title': $('title')  
            });  
        }, function (ws, xhr) {  
            ws.response.error(xhr.status, xhr.statusText);  
        });  
    }  
}
```

FIG. 5

```
<access_control> 600  
    <global-rule protocols="https">  
    </global-rule>  
    <rule for="catalogue-p-plp" role="slave"/>  
</access_control>
```

FIG. 6



EUROPEAN SEARCH REPORT

Application Number
EP 14 18 2403

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	KUT A ET AL: "An approach for parallel execution of web services", 6 July 2004 (2004-07-06), WEB SERVICES, 2004. PROCEEDINGS. IEEE INTERNATIONAL CONFERENCE ON SAN DIEGO, CA, USA 6-9 JULY 2004, PISCATAWAY, NJ, USA, IEEE, PAGE(S) 812 - 813, XP010708904, ISBN: 978-0-7695-2167-1 * the whole document *	1-21	INV. G06F17/30
X	Eric Elzinga: "Oracle Service Bus, Implementing Aggregator pattern by use of Split-Join - Oracle .. Java .. OpenSource .. SOA", 31 July 2011 (2011-07-31), XP055169498, Retrieved from the Internet: URL: http://www.xenta.nl/blog/2011/07/03/oracle-service-bus-implementing-aggregator-pattern-by-use-of-split-join/ [retrieved on 2015-02-12] * page 2, line 18 - line 31 * * page 5, line 50 - page 7, line 8 *	1-21	TECHNICAL FIELDS SEARCHED (IPC) G06F
The present search report has been drawn up for all claims			
Place of search Berlin		Date of completion of the search 13 February 2015	Examiner Écolivet, Stéphane
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

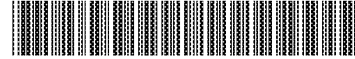
EPO FORM 1566 (03.02.10) (IPC/CPC)

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 802670 A [0032]



(11) **EP 2 869 222 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
06.05.2015 Bulletin 2015/19

(51) Int Cl.:
G06F 17/30 (2006.01) G06F 17/22 (2006.01)

(21) Application number: 14191628.8

(22) Date of filing: 04.11.2014

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(72) Inventor: **Scoda, Enrico**
33035 MARTIGNACCO (IT)

(74) Representative: **Gervasi, Gemma et al**
Notarbartolo & Gervasi S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(30) Priority: 04.11.2013 US 201314070764

(54) **Methods for extending a selector application programming interface and devices thereof**

(57) A method, non-transitory computer readable medium, and web content management server device that receives a cascading style sheet (CSS) selector expression comprising a plurality of expression components. Whether an equivalent XML path language (XPath) expression can be generated for each of the ex-

pression components is determined. Each of the expression components for which an equivalent XPath expression cannot be generated is transformed into a transformed expression comprising at least one XPath expression and an extension function.

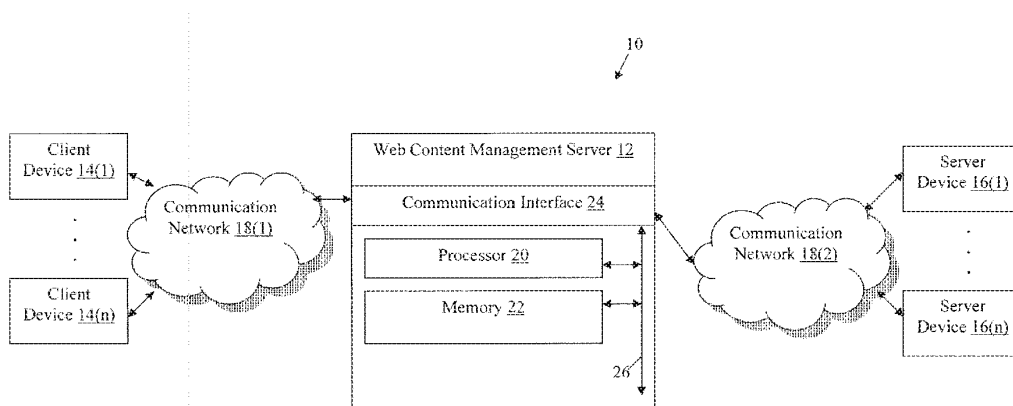


FIG. 1

EP 2 869 222 A1

Description**FIELD OF THE INVENTION**

[0001] This technology generally relates to methods and devices for optimizing delivery of web content and, more particularly, to methods for extending a selectors application programming interface (API) and devices thereof.

BACKGROUND

[0002] A web content management server is a server that optimizes web pages and web page interactions for client devices with special requirements, such as smart-phones. By way of example, a client device can send an HTTP request for a web page which is retrieved from a server device by the web content management server. Next, the web content management server can optimize the content of the web page by applying transformation rules tailored to the requesting client device. This optimization process can include extracting content relevant to the requesting client device and adapting the extracted content to fit the specifications of the requesting client device. In order to adapt the content, the web content management server can perform transformations including JavaScript removal, content linearization, and small screen adaptation, for example.

[0003] Although this process works well to optimize content for display at the requesting client device, the web page may not function properly when the original content heavily depends on JavaScript technology that cannot be implemented on some client devices. For example, if the requested web page includes JavaScript code responsible for populating form fields, validating form submissions, retrieving data from external resources (e.g., based on AJAX technology), and even generating components that may change the structure of the web page, the page at the client device will not be able to properly function.

[0004] Accordingly, JavaScript included in web pages can be removed, stored by the web content management server, and replaced by a hidden field with an identifier, for example. The identifier is subsequently sent by the client device in an HTTP request in response to a web page interaction requiring execution of the JavaScript, such as a request for validation of login credentials. Based on the identifier, the web content management server can retrieve the JavaScript, execute the JavaScript in an emulated JavaScript environment on behalf of the client device, and return a result to the client device.

[0005] In some implementations, the emulated JavaScript environment can support application programming interfaces (APIs) such as those defined by hypertext markup language (HTML), events, cascading style sheet (CSS), range, traversal, and views models, for example. Some of the supported APIs can allow web page developers to embed JavaScript defining functionality for ac-

cessing a document object model (DOM) associated with a web page in order to perform an operation. For example, the selector API provides CSS selector expressions or queries configured to be applied to a DOM to obtain a result set of elements of the DOM.

[0006] However, CSS selector expressions can be difficult to implement, can be complex, and can require specific code or libraries to be implemented. While some CSS selector expressions can be translated into XML path language (XPath) expressions or queries, which are relatively efficient to execute, many CSS selector expressions are not translatable.

SUMMARY

[0007] A method for extending a selector application programming interface (API) includes receiving, with a web content management server, a cascading style sheet (CSS) selector expression comprising a plurality of expression components. Whether an equivalent XML path language (XPath) expression can be generated for each of the expression components is determined with the web content management server. Each of the expression components for which an equivalent XPath expression cannot be generated is transformed with the web content management server into a transformed expression comprising at least one XPath expression and an extension function.

[0008] A non-transitory computer readable medium having stored thereon instructions for extending the selector API comprising machine executable code which when executed by a processor, causes the processor to perform steps including receiving a cascading style sheet (CSS) selector expression comprising a plurality of expression components. Whether an equivalent XML path language (XPath) expression can be generated for each of the expression components is determined. Each of the expression components for which an equivalent XPath expression cannot be generated is transformed into a transformed expression comprising at least one XPath expression and an extension function.

[0009] A web content management server device includes a processor coupled to a memory and configured to execute programmed instructions stored in the memory including receiving a cascading style sheet (CSS) selector expression comprising a plurality of expression components. Whether an equivalent XML path language (XPath) expression can be generated for each of the expression components is determined. Each of the expression components for which an equivalent XPath expression cannot be generated is transformed into a transformed expression comprising at least one XPath expression and an extension function.

[0010] This technology provides a number of advantages including methods, non-transitory computer readable medium, and devices that facilitate more efficient processing of CSS selector expressions using XPath. With this technology, CSS selector expressions are fully

translatable into XPath expressions which can be evaluated in an emulated JavaScript environment by a relatively efficient XPath processor without requiring the specific code otherwise necessary to execute the CSS selector expressions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011]

FIG. 1 is a block diagram of a network environment with an exemplary web content management server;

FIG. 2 is a flow chart of an exemplary method for extending the selector API to an emulated JavaScript environment;

FIG. 3 is a flow chart of an exemplary method performed by an extension function for evaluating the :first-of-type, :last-of-type, :nth-of-type, and :nth-last-of-type pseudo classes;

FIG. 4 is a flow chart of an exemplary method performed by an extension function for evaluating the :last-of-type, :only-of-type, and :nth-last-of-type pseudo classes;

FIG. 5 is a flow chart of an exemplary method performed by an extension function for evaluating the :lang pseudo class;

FIG. 6 is a flow chart of an exemplary method performed by an extension function for evaluating the :not pseudo class; and

FIG. 7 is a flow chart of an exemplary method performed by an extension function for evaluating the :selected jQuery pseudo class.

DETAILED DESCRIPTION OF THE INVENTION

[0012] An exemplary network environment 10 with a web content management server 12 coupled to client devices 14(1)-14(n) and server devices 16(1)-16(n) is illustrated in FIG. 1. In this example, the web content management server 12, client devices 14(1)-14(n), and server devices 16(1)-16(n) are coupled together by communication networks 18(1)-18(2), although other numbers and types of systems, devices, and/or elements in other configurations or network topologies can also be used. This technology provides a number of advantages including methods, non-transitory computer readable medium, and devices that facilitate more efficient processing of CSS selector expressions in an emulated JavaScript environment.

[0013] The web content management server 12 is coupled to the client devices 14(1)-14(n) by the communication network 18(1), which can include one or more local

area network(s) (LANs) and/or wide area network(s) (WANs). In this example, the web content management server 12 is further coupled to the server devices 16(1)-16(n) by the communication network 18(2), which may also include one or more LANs and/or WANs. Other network devices configured to generate, send, and receive network communications and coupled together via other topologies can also be used. While not shown, the network environment 10 also may include additional network components, such as routers, switches and other devices, which are well known to those of ordinary skill in the art and thus will not be described here.

[0014] The web content management server 12 may perform any number of functions including optimizing content retrieved from the server devices 16(1)-16(n) for display on the client devices 14(1)-14(n), for example. In this example the web content management server 12 includes a processor 20, a memory 22, and a communication interface 24, which are coupled together by a bus 26 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used.

[0015] The processor 20 in the web content management server 12 executes a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 20 could execute other numbers and types of programmed instructions. The processor 20 of the web content management server 12 may comprise one or more central processing units or general purpose processors with one or more processing cores, for example.

[0016] The memory 22 in the web content management server 12 stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 20, can be used for the memory 22 in the web content management server 12.

[0017] The communication interface 24 in the web content management server 12 is used to operatively couple and communicate between the web content management server 12, client devices 14(1)-14(n), and server devices 16(1)-16(n), which are all coupled together via the communication networks 18(1)-18(2), although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. By way of example only, the communication networks 18(1)-18(2) can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer

protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[0018] The client devices 14(1)-12(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the server devices 16(1)-16(n) through the web content management server 12 and using the communication network 18(1), although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. In some examples, the client devices 14(1)-14(n) comprise mobile devices with Internet access that enable web pages and other content stored by the server devices 16(1)-16(n) to be retrieved and rendered. By way of example only, the client devices 14(1)-14(n) can be smart phones, personal digital assistants, tablets, or computers.

[0019] Each of the client devices 14(1)-14(n) includes a processor, a memory, an input device, a display device, and a communication interface, which are coupled together by a bus or other link, although one or more of client devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor in each of the client devices 14(1)-14(n) can execute a program of instructions stored in the memory the client device 14(1)-14(n) for one or more aspects of the present invention, as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0020] The input device in each of the client devices 14(1)-14(n) can be used to input selections, such as a request for a particular web page or other content stored by one or more of the server devices 16(1)-16(n), although the input device could be used to input other types of requests and data and interact with other elements. The input device can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can also be used.

[0021] The display device in each of the client devices 14(1)-14(n) can be used to show data and information to the user, such as web pages and other content retrieved from the server devices 16(1)-16(n) by way of example only. The display device in each of the client devices 14(1)-14(n) can be a mobile phone screen display, although other types and numbers of displays could be used depending on the particular type of client device.

[0022] The communication interface in each of the client devices 14(1)-14(n) can be used to operatively couple and communicate between the client devices 14(1)-14(n), web content management server 12, and server devices 16(1)-16(n) over the communication networks 18(1)-18(2).

[0023] Each of the server devices 16(1)-16(n) provides content including web pages and web applications for use by one or more of the client devices 14(1)-14(n) via the web content management server 12, although the server devices 16(1)-16(n) can provide other numbers and types of content and perform other functions. Each of the server devices 16(1)-16(n) can include a processor, a memory, and a communication interface, which are coupled together by a bus or other link, although each of the server devices 16(1)-16(n) can have other numbers and types of components, parts, devices, systems, and elements in other configurations.

[0024] The processor in each of the server devices 16(1)-16(n) executes a program of instructions stored in the memory of the server devices 16(1)-16(n) for one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor could execute other numbers and types of programmed instructions.

[0025] The communication interface in each of the server devices 16(1)-16(n) is used to operatively couple and communicate between the server devices 16(1)-16(n), the web content management server 12, and the client devices 14(1)-14(n) via the communication networks 18(1)-18(2).

[0026] Although embodiments web content management server 12, the client devices 14(1)-14(n), and the server devices 16(1)-16(n) are described and illustrated herein, each of the web content management server 12, the client devices 14(1)-14(n), and the server devices 16(1)-16(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0027] Furthermore, each of the devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and microcontrollers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0028] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks,

G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0029] The embodiments may also be embodied as one or more non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0030] An exemplary method for extending a selector application programming interface (API) will now be described with reference to FIGS. 1-7. Referring more specifically to FIG. 2, a flow chart of an exemplary method for extending the selector API to an emulated JavaScript environment is illustrated. In this example, in step 200, the web content management server 12 receives an HTTP request for a web page from one of the client devices 14(1)-14(n) and retrieves the requested web page from one of the server devices 16(1)-16(n).

[0031] In step 202, the web content management server 12 processes the retrieved web page to generate an XML document object model (DOM) and to remove one or more JavaScript code portions. In this example, the web content management server 12 stores the DOM and each of the removed JavaScript portions in the memory 22. Each of the removed JavaScript portions can be stored as associated with a respective identifier.

[0032] Additionally, the web page can be modified by the web content management server 12 to insert hidden fields in place of the removed JavaScript. Each of the hidden fields include at least one of the identifiers and are inserted so as to cause the requesting one of the client devices 14(1)-14(n) to send one or more of the identifiers in a subsequent HTTP request. Optionally, the web page can be further processed to optimize the web page for a display on the requesting one of the client devices 14(1)-14(n) and/or to improve the transmission and/or load time of the web page, for example, although any other type of processing can also be performed on the web page.

[0033] In step 204, the web content management server 12 sends the processed web page to the requesting one of the client devices 14(1)-14(n). Once received, the requesting one of the client devices 14(1)-14(n) can load the web page in a web browser for example.

[0034] In step 206, the web content management server 12 receives an HTTP request which includes the identifier from the requesting one of the client devices 14(1)-14(n) in response to a user interaction with the web page. For example, the user may submit login credentials using a submit button which prompts the requesting one of the client devices 14(1)-14(n) to generate and send an HTTP request including one of the identifiers previously included in a tag associated with the submit button in step 202. In response to receiving the HTTP request, the web con-

tent management server 12 retrieves the JavaScript code corresponding to the identifier included in the request from the memory 22.

[0035] In step 208, the web content management server 12 executes the retrieved JavaScript code on behalf of the requesting one of the client devices 14(1)-14(n) in an emulated JavaScript environment. The emulated JavaScript environment essentially emulates the web browser environment of the requesting one of the client devices 14(1)-14(n). Methods and devices for utilizing an emulated JavaScript environment are described in U.S. Patent Application No. 12/802,670 entitled "Methods for utilizing a javascript emulator in a web content proxy server and devices thereof," which is incorporated herein by reference in its entirety.

[0036] During execution of the JavaScript code, the web content management server 12 determines, in step 210, whether a cascading stylesheet (CSS) selector expression is encountered. CSS selector expressions include a querySelector() or querySelectorAll() function call in this example and a plurality of expression components, although the CSS selector expressions could include other function calls and other content. Each of the querySelector() and querySelectorAll() functions are defined in the selector API and receive at least one CSS selector as a parameter. If the web content management server 12 determines that a CSS selector expression has been encountered, then the Yes branch is taken to step 212.

[0037] While steps 200-212 are illustrated for exemplary purposes, other methods of encountering or receiving a CSS selector expression can be used. Additionally, other types of environments other than an emulated JavaScript environment can be used. For example, the methods and devices described and illustrated herein are operable with any browser emulator executed on a web content management server in which the DOM for each retrieved web page is implemented using an XML DOM implementation paired with an XPath processor.

[0038] In step 212, the web content management server 12 extracts an expression component from the CSS selector expression, and more particularly, from the CSS selector(s) included as parameters in the CSS selector expression. One exemplary expression component can be "div.red" (e.g., as part of a CSS selector expression "querySelectorAll('div.red')"). This expression component would require selecting all the "div" elements of the DOM generated in step 202 having a class attribute containing "red". Other expression components can also be extracted in step 212.

[0039] In step 214, the web content management server 12 determines whether an equivalent XML path language (XPath) expression can be generated for the extracted expression component. The determination can be made based on a stored mapping of expression component formats or types to equivalent XPath expressions, for example, although other methods of determining whether the expression component has an equivalent

XPath expression can also be used. In this example, the "div.red" expression component is equivalent to the "//div[contains(concat(' ', @class, ' '), 'red')]" XPath expression. Accordingly, if the web content management server 12 determines that an equivalent XPath expression can be generated for the expression component, then the Yes branch is taken to step 216.

[0040] In step 216, the web content management server 12 generates the equivalent XPath expression from the expression component. Optionally, the web content management server 12 also stores the equivalent XPath expression at least temporarily in the memory 22.

[0041] In step 218, the web content management server 12 determines whether there are more expression components that can be extracted from the CSS selector expression. If the web content management server 12 determines there are more expression components that can be extracted from the CSS selector expression, then the Yes branch is taken back to step 212 and another expression component is extracted, as described and illustrated earlier.

[0042] The web content management server 12 then determines in step 214 whether an equivalent XPath expression can be generated for the extracted expression component, also as described and illustrated earlier. In this iteration, the exemplary extracted expression component is ".red:nth-of-type(odd)". This exemplary expression component would require selecting all the elements of the DOM generated in step 202 having a class attribute containing "red" and being in an odd position in the set of all children of the element's parents having the same name as the element. In this example, the :nth-of-type pseudo class does not have an equivalent XPath expression. Accordingly, in this iteration, the web content management server 12 determines in step 214 that an equivalent XPath expression cannot be generated for the extracted expression component and the No branch is taken to step 220.

[0043] In step 220, the web content management server 12 transforms the expression component into a transformed expression including at least one XPath expression and at least one extension function. Optionally, the web content management server 12 also stores the transformed expression at least temporarily in the memory 22. The extension functions in this example are custom functions written in Java, optionally defined in a file stored in the memory 22, and associated with an XPath processor of the web content management server 12. The extension functions can be mapped to one or more pseudo classes in the memory 22 and selected for a particular transformation based on the mapping.

[0044] Optionally, the extension functions are prefixed by a namespace which facilitates locating of the extension functions by the XPath processor. Also optionally, the transformed expression can include a mathematical operation (e.g., in order to implement an "odd" parameter of the pseudo class of the expression component). An exemplary set of extension functions is described and

illustrated in more detail later with reference to FIGS. 3-7.

[0045] In this example, the expression component can be transformed into a transformed expression including "//[contains(concat(' ', @class, ' '), 'red')][expr:css-type-position(.) mod 2 = 1]". Accordingly, in this example, the XPath expression of the transformed expression includes "//[contains(concat(' ', @class, ' '), 'red')]" The extension function of the transformed expression in this example is the "css-type-position()" extension function, although other extension functions with different names can also be used.

[0046] The extension function in this example takes in an indication of an element of the DOM as a parameter (or "." which indicates all elements as used in this example), although other parameters can also be used. In this example, "expr:" is used as a namespace prefix, although any namespace prefix can be used. Additionally, a modulus operation and comparison is performed on the result of the execution of the extension function in this example, although other mathematical or logical operations can also be used.

[0047] In step 218, the web content management server 12 determines whether there are any more expression components in the CSS selector expression, as described and illustrated earlier. If the web content management server 12 determines there are no more expression components in this iteration, then the No branch is taken to step 222.

[0048] In step 222, the web content management server 12 generates a combined expression including at least all of the transformed expressions and generated equivalent XPath expressions optionally stored in the memory 22 in steps 216 and step 220. In this example, the expressions can be merged together using the XPath operators corresponding to the CSS descendants operators, for example, although other operators and methods of combining the expressions can also be used.

[0049] In step 224, the web content management server 12 applies the combined expression to the DOM, generated in step 202 and corresponding to the web page, to generate a result. The result can be a set of elements of the DOM that satisfy or correspond to the CSS selector expression encountered in step 210, although other types and numbers of results are also possible. Subsequent to applying the combined in expression to the DOM or, referring back to step 210, if the web content management server 12 determines during execution of the JavaScript code in step 208 that a CSS selector expression has not been encountered and the No branch is taken, the web content management server proceeds to step 226.

[0050] In step 226, the web content management server 12 determines whether execution of the retrieved JavaScript has completed. If the web content management server 12 proceeds to step 226 subsequent to applying the combined expression to the DOM in step 224, it is likely that execution of the JavaScript is not complete. Accordingly, in these examples, the No branch is taken

back to step 208 and the web content management server 12 continues executing the JavaScript retrieved in step 206. Additionally, the result generated in step 224 can be used by the JavaScript in these examples such as to manipulate, filter, perform an operation on, or reduce the DOM elements indicated therein. During the continued execution of the JavaScript code, the web content management server 12 again proceeds to step 210 in a subsequent iteration, as described and illustrated earlier.

[0051] However, if the No branch is taken from step 210, execution of the JavaScript is likely complete and no CSS selector expressions were encountered, or no additional CSS selector expression subsequent to a prior iteration of steps 214-224 was encountered. Accordingly, in these examples, the Yes branch is taken from step 226 to step 228. In step 228, the web content management server 12 sends a response to the HTTP request received in step 206 to the requesting one of the client devices 14(1)-14(n). The contents of the HTTP response can be based on the execution of the JavaScript code, for example, including any CSS selector expressions embedded therein.

[0052] Referring to FIGS. 3-7, exemplary methods performed by an exemplary set of extension functions when executed during the application of the combined expression to the DOM in step 224 is illustrated. The extension functions can be executed by an XPath processor, which is optionally part of an extensible stylesheet language (XSL) processor, of the web content management server 12, for example, although other methods of executing the extension functions can also be used.

[0053] Referring more specifically to FIG. 3, a flow chart of an exemplary method performed by an extension function for evaluating the :first-of-type, :last-of-type, :nth-of-type, and :nth-last-of-type pseudo classes is illustrated. In step 300, the web content management server 12 receives an indication of a DOM element, such as passed through a parameter of the extension function. In this example, the extension function can be the "css-type-position(element)" extension function identified earlier for evaluating the :nth-of-type pseudo class of the ".red:nth-of-type(odd)" expression component of the example described and illustrated earlier, although any other name can be used for the extension function.

[0054] In step 302, the web content management server 12 instantiates a variable T having a value of the tag name of the DOM element received in step 300. In step 304, the web content management server 12 instantiates a variable C having a value of 1. Other variables and variable names can also be used.

[0055] In step 306, the web content management server 12 determines whether there are any preceding siblings in the DOM of the DOM element received in step 300. If the web content management server 12 determines there are any preceding siblings in the DOM, then the Yes branch is taken to step 308. In step 308, the web content management server 12 determines whether one of the preceding siblings is an element and has a tag

name equal to T. If the web content management server 12 determines that the preceding sibling is an element and has a tag name equal to T, then the Yes branch is taken to step 310. In step 310, the web content management server 12 increments the value of C. If the web content management server 12 determines that the preceding sibling is not an element or does not have a tag name equal to T, then the No branch is taken back to step 306.

[0056] In a subsequent iteration of step 306 in this example, the web content management server 12 again determines whether there are any additional preceding siblings of the element that have not yet been considered. If the web content management server 12 determines there is not any additional preceding siblings, then the No branch is taken to step 312. In step 312, the web content management server 12 returns the value of C. The returned value can be used to continue evaluating the combined expression applied to the DOM, for example. Accordingly, the extension function in this example returns a number of preceding sibling elements of the element of the DOM having the same tag name as the element plus one.

[0057] Referring more specifically to FIG. 4, a flow chart of an exemplary method performed by an extension function for evaluating the :last-of-type, :only-of-type, and :nth-last-of-type pseudo classes is illustrated. In step 400, the web content management server 12 receives an indication of a DOM element, such as passed through a parameter of the extension function. In this example, the extension function can be a "css-type-total(element)" extension function, although any other name can be used for the extension function.

[0058] In step 402, the web content management server 12 instantiates a variable T having a value of the tag name of the DOM element received in step 400. In step 404, the web content management server 12 instantiates a variable C having a value of 0. Other variables and variable names can also be used.

[0059] In step 406, the web content management server 12 determines whether there are any children in the DOM of the parent of the DOM element received in step 400 or, alternatively, any siblings of the DOM element. If the web content management server 12 determines there are children of the parent of the DOM element, then the Yes branch is taken to step 408. In step 408, the web content management server 12 determines whether one of the children is an element and has a tag name equal to T. If the web content management server 12 determines that the child is an element and has a tag name equal to T, then the Yes branch is taken to step 410. In step 410, the web content management server 12 increments the value of C. If the web content management server 12 determines that the child is not an element or does not have a tag name equal to T, then the No branch is taken back to step 406.

[0060] In a subsequent iteration of step 406 in this example, the web content management server 12 again

determines whether there are any additional children of the parent of the element that have not yet been considered. If the web content management server 12 determines there is not any additional children, then the No branch is taken to step 312. In step 312, the web content management server 12 returns the value of C. The returned value can be used to continue evaluating the combined expression applied to the DOM, for example. Accordingly, the extension function in this example returns a number of child elements of the parent of the element of the DOM having the tag name one of the element.

[0061] Referring more specifically to FIG. 5, a flow chart of an exemplary method performed by an extension function for evaluating the :lang pseudo class is illustrated. In step 500, the web content management server 12 receives an indication of a DOM node, such as passed through a parameter of the extension function. In this example, the extension function can be a "lang(node)" extension function, although any other name can be used for the extension function.

[0062] In step 502, the web content management server 12 instantiates a variable N having a value indicating the node received in step 500. In step 504, the web content management server 12 instantiates a variable L having a value of 'en' or any other default indication of a language. Other variables and variable names can also be used.

[0063] In step 506, the web content management server 12 determines whether N indicates a node that is an element. If the web content management server 12 determines N indicates a node that is an element, then the Yes branch is taken to step 508. In step 508, the web content management server 12 determines whether the node indicated by N has a language attribute. If the web content management server 12 determines that the node indicated by N does not have a language attribute, then the No branch is taken to step 510.

[0064] In step 510, the web content management server 12 sets the value of N to indicate the parent of the node indicated by the current value of N. The web content management server 12 then proceeds back to step 506 and again determines whether N indicates a node that is an element. Referring back to step 508, if the web content management server 12 determines that the node indicated by N does have a language attribute, then the Yes branch is taken to step 512.

[0065] In step 512, the value of L is assigned the value of the language attribute of the node indicated by N. In step 514, the web content management server 12 returns the value of L. Referring back to step 506, if the web content management server 12 determines in a first or subsequent iteration that N indicates a node that is not an element, then the No branch is taken to step 514 and the default value of L is returned. The returned value can be used to continue evaluating the combined expression applied to the DOM, for example. Accordingly, the extension function in this example returns a value of a first identified language attribute based on a traversal of the

element of the DOM and any ancestors of the element of the DOM.

[0066] Referring more specifically to FIG. 6, a flow chart of an exemplary method performed by an extension function for evaluating the :not pseudo class is illustrated. In step 600, the web content management server 12 receives an indication of a DOM element and a string including a CSS selector expression, such as passed through a parameter of the extension function. In this example, the extension function can be a "filter-out(element, css-selector)" extension function, although any other name can be used for the extension function.

[0067] In step 602, the web content management server 12 applies the received CSS selector expression to the DOM to generate a result, such as described and illustrated earlier with reference to steps 210-224, for example, although other methods of evaluating the received CSS selector expression can also be used. In this example, the CSS selector expression includes a CSS selector function querySelectorAll(), although other functions can also be included in the CSS selector expression of the string received in step 600.

[0068] In step 604, the web content management server 12 determines whether there are any nodes in the result generated by the application of the CSS selector expression in step 602. If the web content management server 12 determines there are node(s) in the result, then the Yes branch is taken to step 606. In step 606, the web content management server 12 determines whether one of the node(s) in the result is equal to the DOM element received in step 600. If the web content management server 12 determines that the one of the node(s) in the result is equal to the DOM element, then the Yes branch is taken to step 608. In step 608, the web content management server 12 returns a value of false.

[0069] Referring back to step 606, if the web content management server 12 determines that the one of the node(s) in the result is not equal to the DOM element, then the No branch is taken back to step 604. If the web content management server 12 determines in step 604 that the no nodes in the result of the evaluation of the CSS selector expression, or no additional nodes not previously considered, then the No branch is taken to step 610. In step 610, the web content management server 12 returns a value of true. Accordingly, the extension function in this example returns a true value when the element of the DOM received in step 600 is not included in a result obtained by applying the CSS selector expression received in step 600 and a false value when the element of the DOM is included in the result.

[0070] Referring more specifically to FIG. 7, a flow chart of an exemplary method performed by an extension function for evaluating the :selected jQuery pseudo class is illustrated. In step 700, the web content management server 12 receives an indication of a DOM option element, such as passed through a parameter of the extension function. In this example, the extension function can be an "option-selected(option)" extension function, al-

though any other name can be used for the extension function. The indication of the option element can represent an option in a dropdown list in an HTML form, for example, although the option can also refer to other types of elements.

[0071] In step 701, the web content management server 12 determines whether the option is explicitly selected. If the web content management server 12 determines that the option is explicitly selected, then the Yes branch is taken to step 704. In step 704, the web content management server returns a value of true. Referring back to step 702, if the web content management server 12 determines that the option is not explicitly selected, then the No branch is taken to step 706.

[0072] In step 706, the web content management server 12 obtains a select element from the DOM that contains the option element corresponding to the indication received in step 700. In step 708, the web content management server 12 determines whether the select element is a single selection element with size attribute equal to 1. If the web content management server 12 determines that the select element is a single selection element with size attribute equal to 1 then the Yes branch is taken to step 210.

[0073] In step 710, the web content management server 12 determines whether the option element corresponding to the indication received in step 700 is the first option element descendant of the select element. If the web content management server 12 determines that the option element is the first option element descendant of the select element, then the Yes branch is taken to step 712.

[0074] In step 712, the web content management server 12 determines whether all of the other option elements descendant from the select element are not selected. If the web content management server 12 determines that all of the other option elements descendant from the select element are not selected then the Yes branch is taken to step 714. In step 714, the web content management server 12 returns a value of true.

[0075] Referring back to steps 708, 710, and 712, if the web content management server determines that the select element is not a single selection element or does not have a size attribute equal to 1, the option element corresponding to the indication received in step 700 is not the first option element descendant of the select element, or any of the other option elements descendant from the select element are selected, respectively, then one of the respective No branches is taken to step 716. In step 716, the web content management server 12 returns a value of false. Accordingly, the extension function in this example returns a true value if the option element of the DOM is selected and a false value if the option element of the DOM is not selected. In this example, the option is considered selected if it is explicitly selected or if it is the first option of a select element that has no explicitly selected options and is not a multiple selections select element.

[0076] With this technology, web page scripts including JavaScript code can be emulated by a web content management server on behalf of client devices that may not otherwise be able to take advantage of such functionality.

Advantageously, any CSS selector expressions embedded in the JavaScript can be translated to XPath expressions based on an equivalent XPath expression or, in the case of CSS selector expressions without equivalent XPath expressions, by using an extension function. Thereby, CSS selector expressions can be fully translatable to XPath expressions and implemented using a relatively efficient XPath processor without the significant additional code and associated complexity required to implement the CSS selector expressions of the selector API.

[0077] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for extending a selector application programming interface (API), the method comprising:

receiving, with a web content management server, a cascading style sheet (CSS) selector expressions comprising a plurality of expression components;
determining, with the web content management server, whether an equivalent XML path language (XPath) expression can be generated for each of the expression components; and
transforming, with the web content management server, each of the expression components for which an equivalent XPath expression cannot be generated into a transformed expression comprising at least one XPath expression and an extension function.

2. The method of claim 1 further comprising generating, with the web content management server, the equivalent XPath expression for each of the plurality of expression components for which the equivalent XPath expression can be generated.

3. The method of claim 2 further comprising:
 generating, with the web content management server, a combined expression comprising each of the transformed expressions and each of the equivalent XPath expressions; and
 applying, with the web content management server, the combined expression to a document object model (DOM) to generate a result.

4. The method of claim 3 wherein the extension function comprises JavaScript code and the applying further comprises executing the extension function in an emulated JavaScript environment.

5. The method of claim 3 wherein:
 at least one of the expression components for which an equivalent XPath expression cannot be generated comprises a pseudo class comprising :first-of-type, :last-of-type, :nth-of-type, or :nth-last-of-type; and
 the extension function of the transformed expression for the at least one of the expression components is configured to receive an indication of an element of the DOM, the indication comprising a tag name of the element of the DOM, and return a number of one or more preceding sibling elements of the element of the DOM having the tag name.

6. The method of claim 3 wherein:
 at least one of the expression components for which an equivalent XPath expression cannot be generated comprises a pseudo class comprising :last-of-type, :only-of-type, or :nth-last-of-type; and
 the extension function of the transformed expression for the at least one of the expression components is configured to receive an indication of an element of the DOM, the indication comprising a tag name of the element of the DOM and return a number of child elements of the parent of the element of the DOM having the tag name plus one.

7. The method of claim 3 wherein:
 at least one of the expression components for which an equivalent XPath expression cannot be generated comprises a :lang pseudo class; and
 the extension function of the transformed expression for the at least one of the expression components is configured to receive an indication of an element of the DOM and return a value of a first identified language attribute based on

a traversal of the element of the DOM and any ancestors of the element of the DOM.

8. The method of claim 3 wherein:
 at least one of the expression components for which an equivalent XPath expression cannot be generated comprises a :not pseudo class; and
 the extension function of the transformed expression for the at least one of the expression components is configured to receive an indication of an element of the DOM and a second CSS selector expression and return a true value when the element of the DOM is not included in a second result obtained by applying the second CSS selector expression and a false value when the element of the DOM is included in the second result.

9. The method of claim 3 wherein:
 at least one of the expression components for which an equivalent XPath expression cannot be generated comprises a :selected jQuery pseudo class; and
 the extension function of the transformed expression for the at least one of the expression components is configured to receive an option element of the DOM and return a true value if the option element of the DOM is selected and a false value if the option element of the DOM is not selected.

10. A non-transitory computer readable medium having stored thereon instructions for extending a selector application programming interface (API) comprising machine executable code which when executed by a processor carries out the method according to claims from 1 to 9.

11. A management server device, comprising:
 a processor coupled to a memory and configured to execute programmed instructions stored in the memory to carry out the method according to claims from 1 to 9..

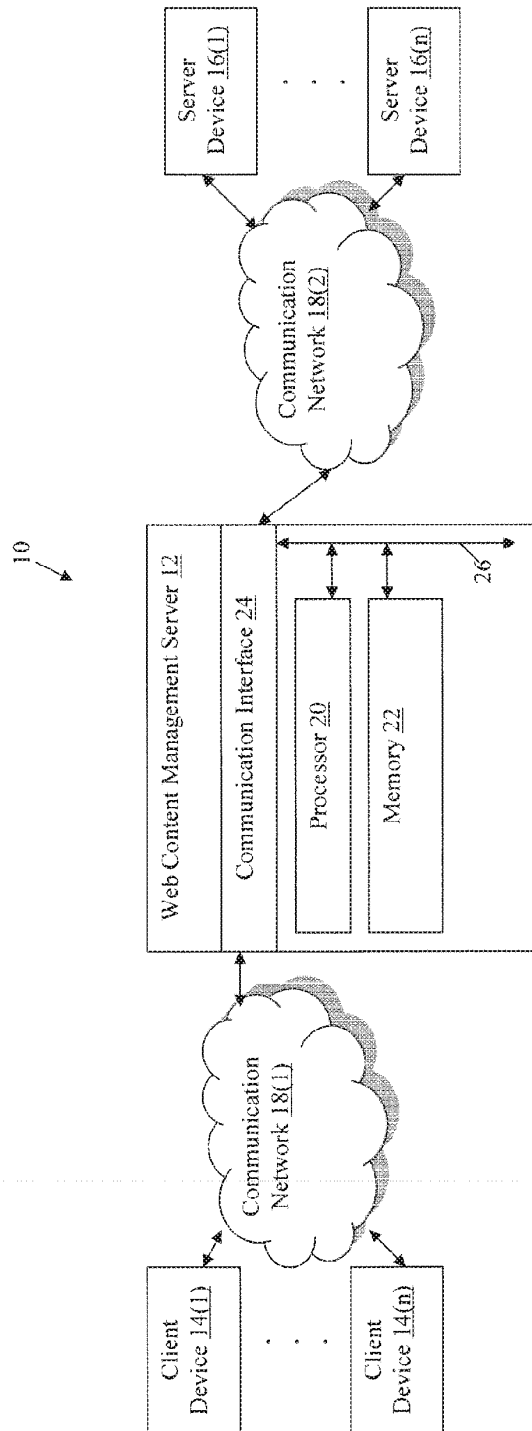


FIG. 1

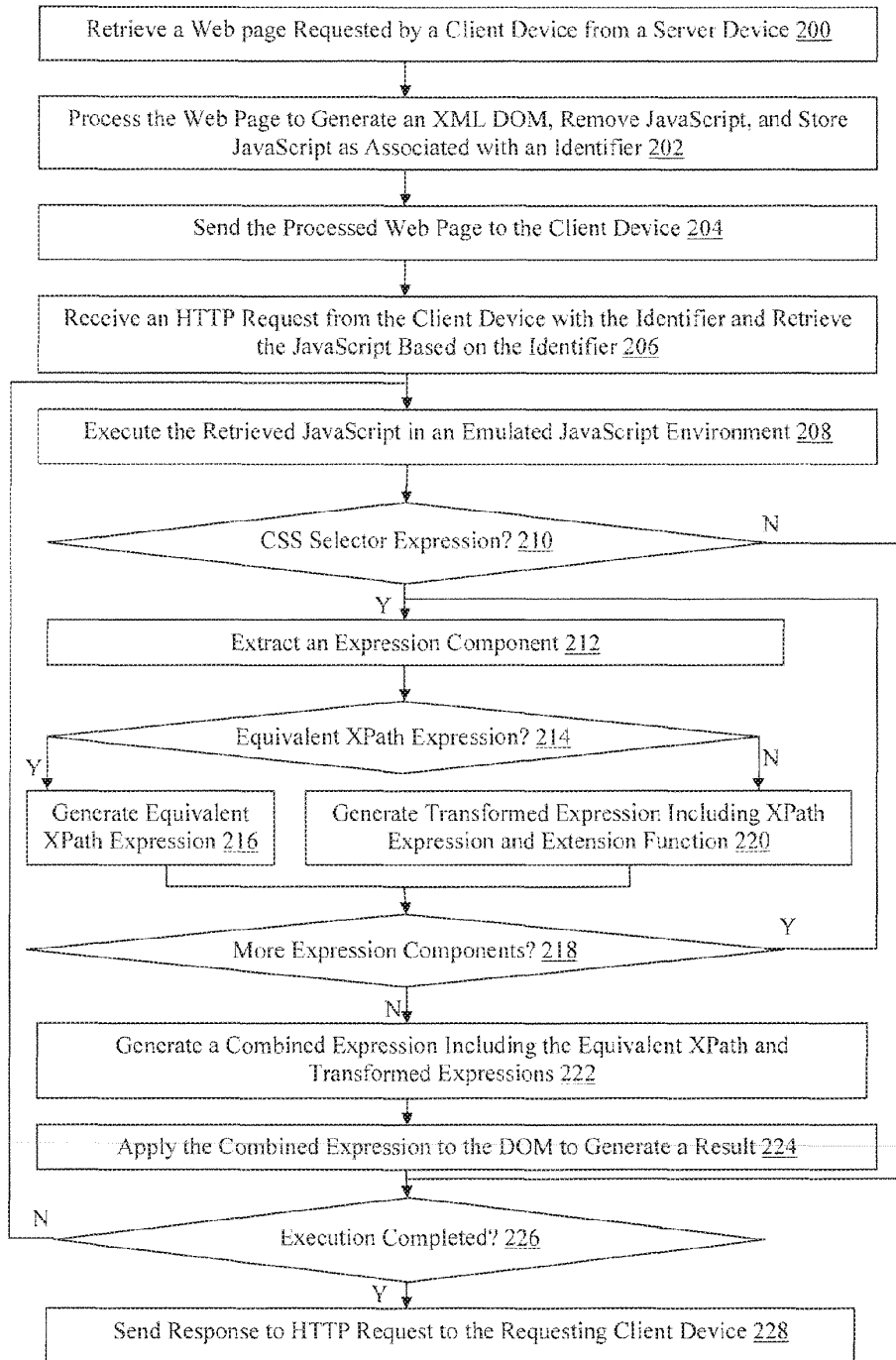


FIG. 2

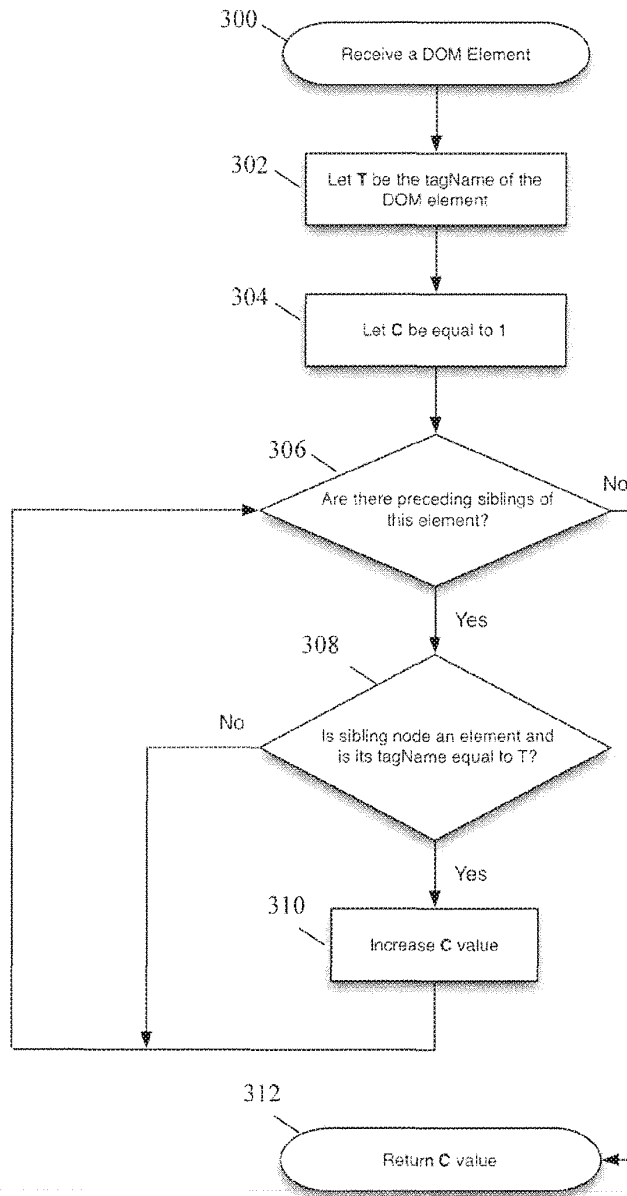


FIG. 3

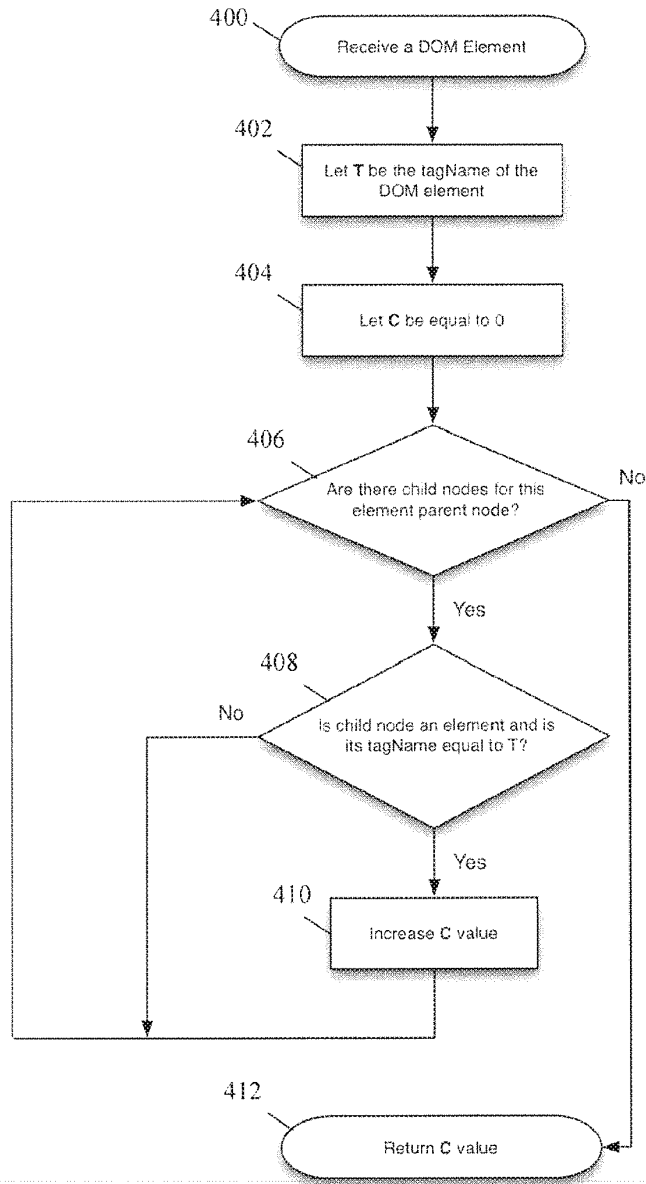


FIG. 4

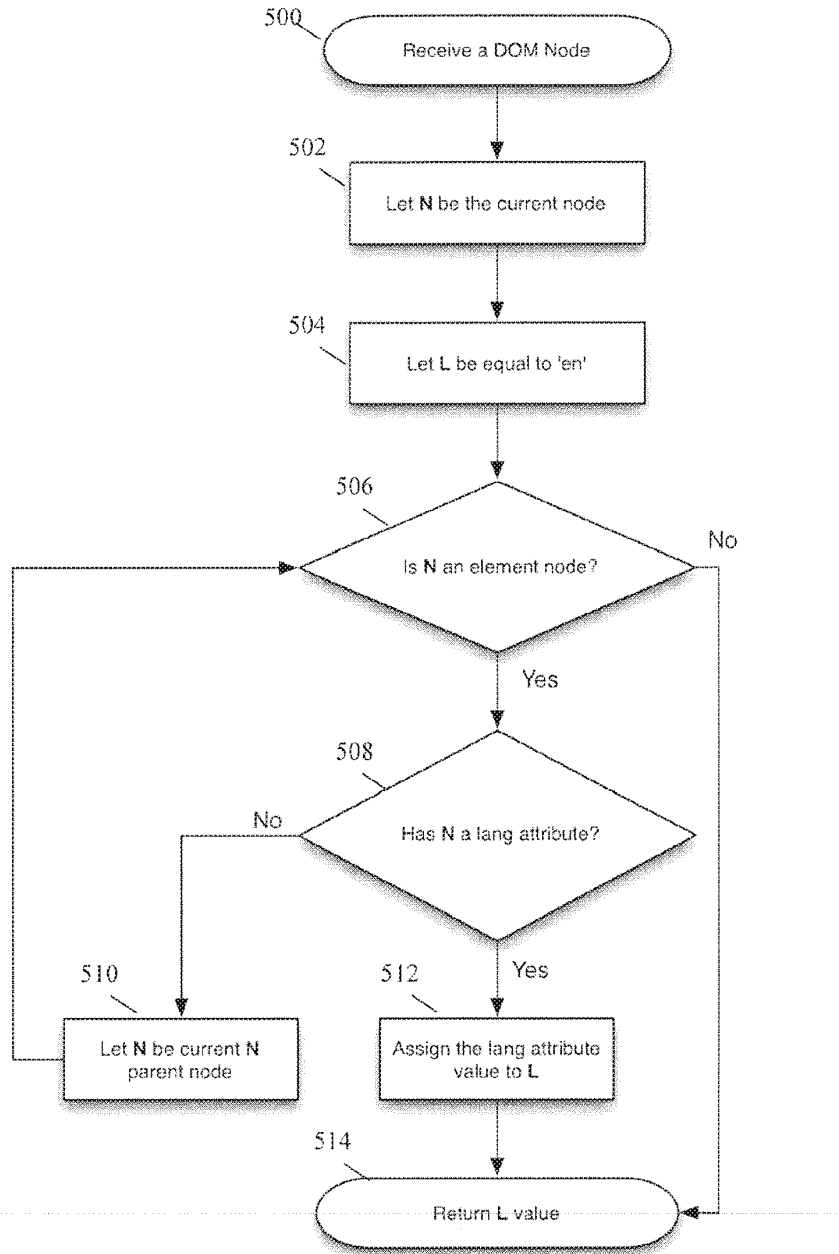


FIG. 5

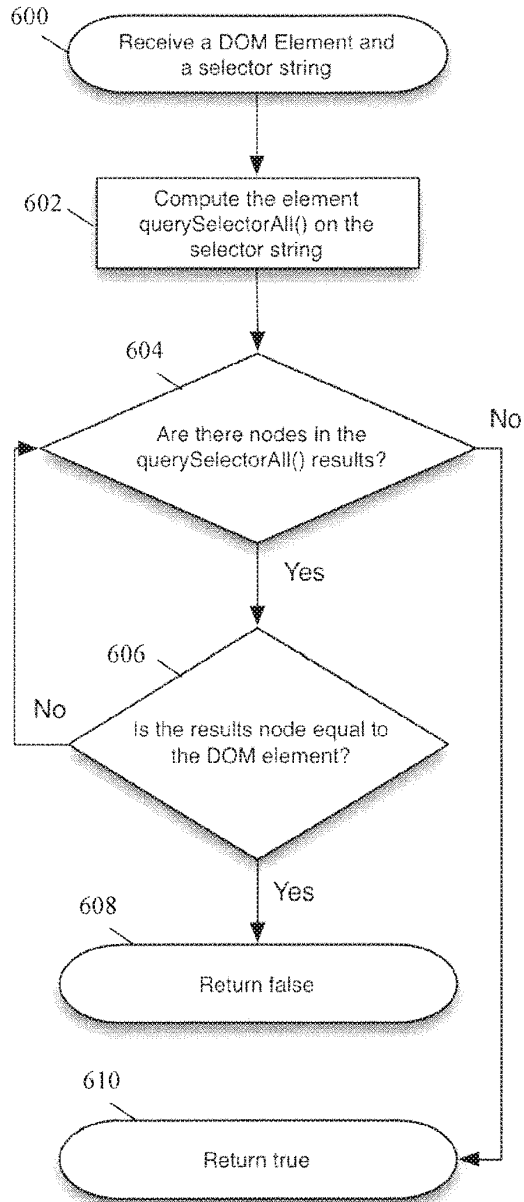


FIG. 6

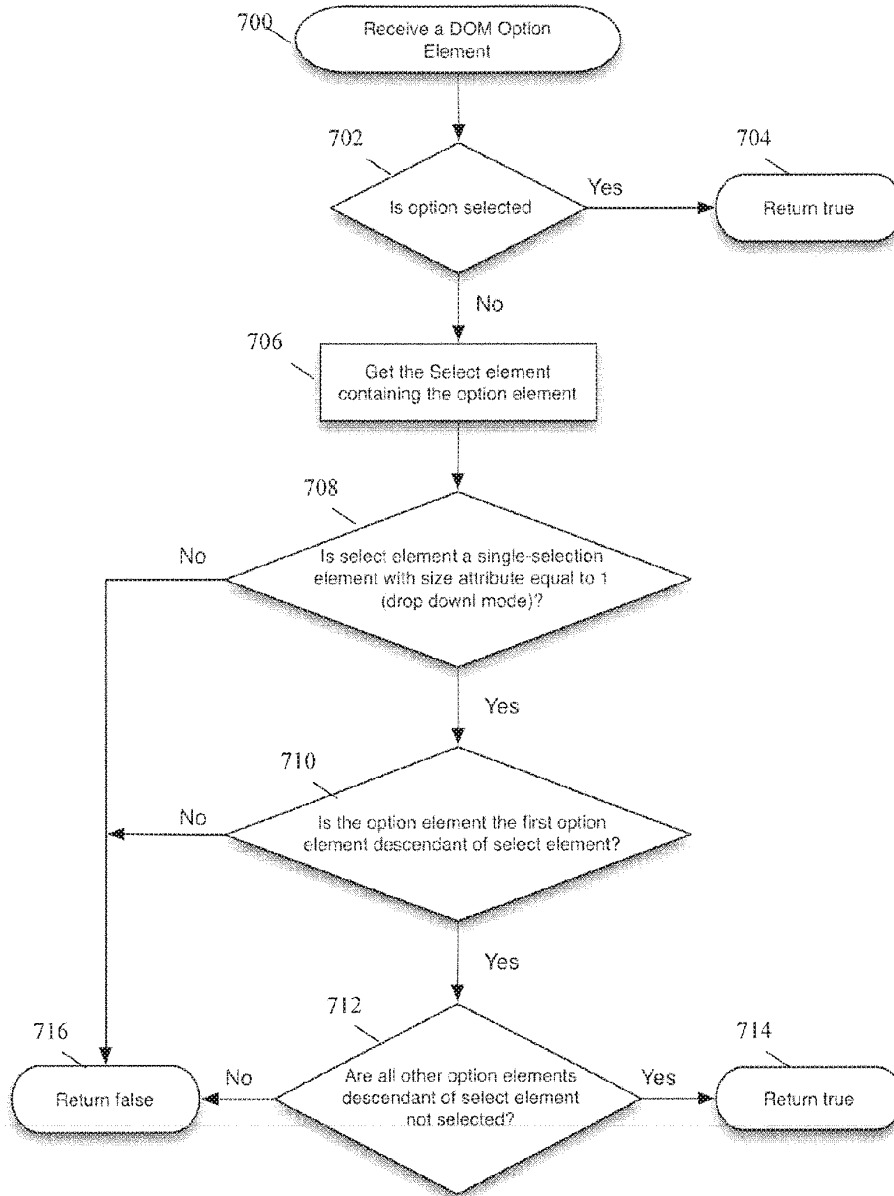


FIG. 7



EUROPEAN SEARCH REPORT

Application Number
EP 14 19 1628

5

10

15

20

25

30

35

40

45

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	ALI MESBAH ET AL: "Automated analysis of CSS rules to support style maintenance", SOFTWARE ENGINEERING (ICSE), 2012 34TH INTERNATIONAL CONFERENCE ON, IEEE, 2 June 2012 (2012-06-02), pages 408-418, XP032195346, DOI: 10.1109/ICSE.2012.6227174 ISBN: 978-1-4673-1066-6 * abstract * * page 408, left-hand column, line 20 - page 410, left-hand column, line 24 * * page 410, right-hand column, line 49 - page 411, right-hand column, line 39 * * page 412, left-hand column, line 47 - page 413, right-hand column, line 26 *	1-11	INV. G06F17/30 G06F17/22
A	W3c: "XSL Transformations (XSLT) Version 2.0 - W3C Working Draft 4 April 2005", 4 April 2005 (2005-04-04), pages 1-180, XP055028208, Retrieved from the Internet: URL:http://www.w3.org/TR/2005/WD-xslt20-20050404/ [retrieved on 2012-05-25] * page 4, line 12 - page 4, line 34 * * page 22, line 1 - page 24, line 24 * * page 61, line 1 - page 62, line 3 * * page 109, line 23 - page 111, line 21 *	1-11	TECHNICAL FIELDS SEARCHED (IPC) G06F
A	US 2011/307238 A1 (SCODA ENRICO [IT]) 15 December 2011 (2011-12-15) * abstract * * paragraph [0001] - paragraph [0010] * * paragraph [0013] - paragraph [0016] * * paragraph [0043] - paragraph [0048] * * paragraph [0056] - paragraph [0066] *	1-11	
----- -/--			
The present search report has been drawn up for all claims			
Place of search The Hague		Date of completion of the search 13 March 2015	Examiner Boyadzhiev, Yavor
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EPO FORM 508 (3-12) (P/M/EN)

55



EUROPEAN SEARCH REPORT

Application Number
EP 14 19 1628

5

10

15

20

25

30

35

40

45

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	<p>WILLIAM P ZELLER AND EDWARD W FELTEN {WZELLER ET AL: "SVC: Selector-based View Composition for Web Frameworks", USENIX,, 19 May 2010 (2010-05-19), pages 1-12, XP061011478, [retrieved on 2010-05-19] * abstract * * page 1, left-hand column, line 34 - page 2, right-hand column, line 12 * * page 2, right-hand column, line 43 - page 4, right-hand column, line 50 * * page 5, left-hand column, line 31 - page 8, left-hand column, line 52 * -----</p>	1-11	
			TECHNICAL FIELDS SEARCHED (IPC)
The present search report has been drawn up for all claims			
Place of search		Date of completion of the search	Examiner
The Hague		13 March 2015	Boyadzhiev, Yavor
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

2

EP0 14191628 (3.12) (P/M/C/D/H)

50

55

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 14 19 1628

5

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

13-03-2015

10

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011307238 A1	15-12-2011	CA 2800723 A1	15-12-2011
		EP 2580686 A2	17-04-2013
		US 2011307238 A1	15-12-2011
		WO 2011156743 A2	15-12-2011

15

20

25

30

35

40

45

50

55

EP O FORM P/MS9

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 802670 A [0035]

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 3 140 733 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2015/171228 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2015/171228 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2015/171228 (art. 153(3) CBE).



- (51) International Patent Classification:
G06F 9/445 (2006.01)
- (21) International Application Number:
PCT/US2015/024172
- (22) International Filing Date:
2 April 2015 (02.04.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/988,639 5 May 2014 (05.05.2014) US
- (71) Applicant: USABLENET INC. [US/US]; 142 W. 57th Street, 7th Floor, New York, NY 10019 (US).
- (72) Inventor: SCODA, Enrico; Via Cividina 416/3, 33035 Martignacco Ud (IT).
- (74) Agents: GALLO, Nicholas, J. et al.; LeClairRyan, 70 Linden Oaks, Suite 210, Rochester, NY 14625 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— with international search report (Art. 21(3))

(54) Title: METHODS FOR FACILITATING A REMOTE INTERFACE AND DEVICES THEREOF

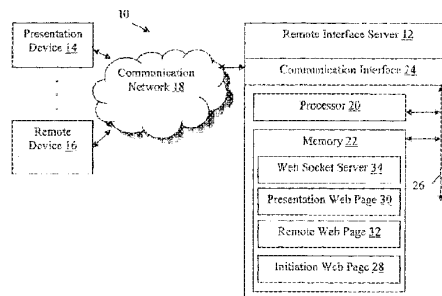


FIG. 1

(57) Abstract: A method, non-transitory computer readable medium, remote interface server computing device, and system that provides a presentation web page to a presentation device and a remote web page to a remote device. The remote web page is configured to, when executed by the remote device, register the remote device as associated with the presentation device and render a swipe panel on a display of the remote device. A first message is received from the remote device in response to an interaction with the swipe panel. A second message is sent to the presentation device in response to receiving the first message. The second message, when executed by the presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

WO 2015/171228 A1

- 1 -

METHODS FOR FACILITATING A REMOTE INTERFACE AND DEVICES THEREOF

[0001] This application claims the benefit of U.S. Provisional Patent
5 Application Serial No. 61/988,639 filed on May 5, 2014, which is hereby
incorporated by reference in its entirety.

FIELD

[0002] This technology generally relates to kiosk and other presentation
10 devices, and more particularly to methods and devices for facilitating a remote
interface for interacting with such presentation devices.

BACKGROUND

[0003] Presentation devices, such as kiosks and other devices with
relatively large screen sizes, are often available for interaction in commercial and
other settings. Presentation devices can display product information associated
15 with a catalog of available products for a retailer, for example, advertising
information, or any other information directed to consumers or other members of
the public.

[0004] The method of interaction with presentation devices is often
through a multi-touch screen. However, such presentation devices are generally
20 complex and have relatively high associated cost due to the multi-touch screens
and required processing power. Additionally, presentation devices with relatively
large screen sizes are currently unable to effectively present, and/or allow users to
input, private information (e.g. personally identifiable information or credit card
numbers) in a discreet manner.

25 **SUMMARY**

[0005] A method for facilitating a remote interface includes providing, by
a remote interface server computing device, a presentation web page to a
presentation device and a remote web page to a remote device. The remote web

- 2 -

page is configured to, when executed by the remote device, register the remote device as associated with the presentation device and render a swipe panel on a display of the remote device. A first message is received, by the remote interface server computing device, from the remote device in response to an interaction with the swipe panel. A second message is sent, by the remote interface server computing device, to the presentation device in response to receiving the first message. The second message, when executed by the presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

10 [0006] A non-transitory computer readable medium having stored thereon instructions for facilitating a remote interface comprising executable code which when executed by a processor, causes the processor to perform steps including providing a presentation web page to a presentation device and a remote web page to a remote device. The remote web page is configured to, when executed by the
15 remote device, register the remote device as associated with the presentation device and render a swipe panel on a display of the remote device. A first message is received from the remote device in response to an interaction with the swipe panel. A second message is sent to the presentation device in response to receiving the first message. The second message, when executed by the
20 presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

[0007] A remote interface server computing device including a processor and a memory coupled to the processor which is configured to be capable of
25 executing programmed instructions comprising and stored in the memory to provide a presentation web page to a presentation device and a remote web page to a remote device. The remote web page is configured to, when executed by the remote device, register the remote device as associated with the presentation device and render a swipe panel on a display of the remote device. A first
30 message is received from the remote device in response to an interaction with the swipe panel. A second message is sent to the presentation device in response to receiving the first message. The second message, when executed by the

- 3 -

presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

[0008] A system for facilitating a remote interface includes a remote
5 interface server computing device including a first processor and a first memory coupled to the first processor. The first process is configured to be capable of executing programmed instructions comprising and stored in the first memory to provide a presentation web page to a presentation device and a remote web page to a remote device. The remote web page configured to, when executed by the
10 remote device, register the remote device as associated with the presentation device and render at least a swipe panel on a display of the remote device. The system further includes a web socket server computing device including a second processor and a second memory coupled to the second processor. The second processor is configured to be capable of executing programmed instructions
15 comprising and stored in the second memory to receive a first message from the remote device in response to a user interaction with the swipe panel. A second message is sent to the presentation device in response to receiving the first message. The second message, when executed by the presentation device, is configured to cause the presentation device to perform an action on the
20 presentation web page corresponding to the user interaction with the swipe panel.

[0009] This technology provides a number of advantages including providing methods, non-transitory computer readable media, devices, and systems that facilitate remote interfaces for presentation devices. With this technology, presentation device (e.g., a kiosk) can be seamlessly controlled by a remote device
25 (e.g., mobile phones) using messages exchanged based on the web socket protocol. By leveraging remote devices, this technology allows presentation devices to be less complex and less costly. Additionally, private information can be advantageously submitted without displaying the information in a visible format on the display of a presentation device

BRIEF DESCRIPTION OF THE DRAWINGS

- [0010] FIG. 1 is a block diagram of a network environment with an exemplary remote interface server coupled to a presentation device and a remote device;
- 5 [0011] FIG. 2 is a block diagram of another network environment with an exemplary remote interface server coupled to a presentation device, a remote device, and a web socket server;
- [0012] FIG. 3 is a flowchart of an exemplary method for facilitating a remote interface;
- 10 [0013] FIG. 4 is an exemplary initiation web page;
- [0014] FIG. 5 is an exemplary presentation web page;
- [0015] FIG. 6 is an exemplary remote web page;
- [0016] FIG. 7 is an exemplary presentation web page;
- [0017] FIG. 8 is an exemplary remote web page modified according to a
15 horizontal swipe gesture with a swipe panel;
- [0018] FIG. 9 is an exemplary presentation web page with input fields;
- [0019] FIG. 10 is an exemplary remote web page with input fields;
- [0020] FIG. 11 is an exemplary remote web page subsequent to user interaction with an edit button;
- 20 [0021] FIG. 12 is an exemplary remote web page with a virtual keyboard;
- [0022] FIG. 13 is an exemplary presentation web page with input fields subsequent to a user editing content;
- [0023] FIG. 14 is an exemplary presentation web page subsequent to user interaction with a play video button of a remote web page; and

- 5 -

[0024] FIG. 15 is an exemplary remote web page subsequent to user interaction with a play video button of the remote web page.

DETAILED DESCRIPTION

[0025] An exemplary network environment 10 with a remote interface server 12 coupled to a presentation device 14 and a remote device 16 is illustrated in FIG. 1. In this example, the remote interface server 12, presentation device 14, and remote device 16 are coupled together by at least one communication network 18, although other numbers and types of systems, devices, and/or elements in other configurations or network topologies can also be used. This technology provides a number of advantages including methods, non-transitory computer readable media, devices, and systems that facilitate a remote interfaces to effectively replicate, on the presentation device 14, user interactions with a web page rendered on the remote device 16 while maintaining information privacy.

[0026] The remote interface server 12 (also referred to herein as a remote interface server computing device) in this particular example is coupled to the presentation device 14 and the remote device 16 by the communication network 18 which can include one or more local area network(s) (LANs) and/or wide area network(s) (WANs). Other network devices configured to generate, send, and receive network communications and coupled together via other topologies can also be used. While not shown, the network environment 10 also may include additional network components, such as routers, switches and other devices, which are well known to those of ordinary skill in the art and thus will not be described here.

[0027] The remote interface server 12 may perform any number of functions including hosting and providing web content and facilitating communications between the presentation device 14 and the remote device 16 according to the web socket protocol, for example. In this example, the remote interface server 12 includes a processor 20, a memory 22, and a communication interface 24, which are coupled together by a bus 24 or other communication link,

- 6 -

although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used.

[0028] The processor 20 in the remote interface server 12 executes a program of stored instructions for one or more aspects of this technology, as described and illustrated by way of the embodiments herein, although the processor 20 could execute other numbers and types of programmed instructions. The processor 20 of the remote interface server 12 may include one or more central processing units or general purpose processors with one or more processing cores, for example.

10 [0029] The memory 24 in the remote interface server 12 stores these programmed instructions for one or more aspects of this technology, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. Optionally, the memory 24 in this example stores a plurality of web pages including at least one initiation web page 28, 15 presentation web page 30, and remote web page 32, as described and illustrated in more detail later. A variety of different types of memory storage devices, such as a random access memory (RAM), read only memory (ROM), hard disk drive(s), flash, solid state drive(s), or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system 20 that is coupled to the processor, can be used for the memory 22 in the remote interface server 12.

[0030] In this particular example, the memory 24 also includes a web socket server 34. The web socket server 34 in this example is a software module that includes programmed instructions that, when executed by the processor, 25 generate a web socket server configured to facilitate communications between the presentation device 14 and the remote device 16 according to the web socket protocol, as described and illustrated in more detail later.

[0031] The communication interface 24 in the remote interface server 12 is used to operatively couple and communicate between the remote interface 30 server 12, the presentation device 14, and the remote device 16, which are all

- 7 -

coupled together via the communication network 18, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. By way of example only, the communication network 18 can use TCP/IP over
5 Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), and/or secure HTTP (HTTPS), although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

10 [0032] The presentation device 14 and the remote device 16 in this particular example enable a user to request, receive, and interact with applications, web services, and content hosted by the remote interface server 12 using the communication network 18, although one or more of the presentation device 14 or
15 remote device 16 could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user.

[0033] Each of the presentation device 14 and remote device 16 in this example includes a processor, a memory, an input device, a display device, and a
20 communication interface, which are coupled together by a bus or other communication link, although one or more of presentation device 14 or remote device 16 can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor in each of the presentation device 14 and remote device 16 can execute a program of instructions stored in the memory the client device for one or more aspects of this technology,
25 as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0034] The input device in each of the presentation device 14 and remote device 16 can be used to input selections, such as a request for a particular web page or other content stored by the remote interface server 12 or another web
30 content server, although the input device could be used to input other types of requests and data and interact with other elements. The input device can include

- 8 -

keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can also be used.

[0035] The display device in each of the presentation device 14 and remote device 16 can be used to show data and information to a user, such as web pages and other content retrieved from the remote interface server 12 or another web content server by way of example only. The display device in the presentation device 14 can be a television screen and the display device in the remote device 16 can be a mobile phone screen, for example, although other types and numbers of display devices could be used depending on the particular type of presentation device 14 and remote device 16. The communication interface in each of the presentation device 14 and remote device 16 can be used to operatively couple and communicate between the presentation device 14, remote device 16, and remote interface server 12 over the communication network 18.

[0036] By way of example only, the presentation device 14 can be relatively less mobile than the remote device 16 and can include a television, kiosk, or other device with a relatively large display as compared to that of the remote device 16, although other types of presentation devices can also be used. Accordingly, in some examples, the remote device 16 is relatively more mobile than the presentation device 14 and can be a smartphone, personal digital assistant, tablet, netbook, notebook, or other device with a relatively small display as compared to that of the presentation device 14, although other types of remote devices can also be used.

[0037] Referring more specifically to FIG. 2 another exemplary network environment 36 with a remote interface server 12 coupled to a presentation device 14, a remote device 16, and a web socket server 38 is illustrated. The remote interface server 12, presentation device 14, remote device 16, and communication network 18 in this example are the same as described and illustrated earlier with reference to FIG. 1 except that the remote interface server 12 does not include the web socket server 24. Instead, in this particular example, the web socket server 38 is provided as a separate web socket server computing device in the environment 36 that is also configured to communicate with the presentation

- 9 -

device 14 and the remote device 16 via the communication network 18. Other network topologies and numbers of remote interface servers and/or web socket servers can also be provided in network environment 10 or 36.

[0038] The web socket server 38 in the particular example illustrated in FIG. 2 includes a processor 40, a memory 42, and a communication interface 44, which are coupled together by a bus 46 or other communication link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor 40 in the web socket server 38 executes a program of stored instructions one or more aspects of this technology, as described and illustrated by way of the embodiments herein, although the processor 40 could execute other numbers and types of programmed instructions. The processor 40 of the web socket server 38 may include one or more central processing units or general purpose processors with one or more processing cores, for example.

[0039] The memory 42 in the web socket server 38 stores these programmed instructions for one or more aspects of this technology, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM), read only memory (ROM), hard disk drive(s), flash, solid state drive(s), or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 40, can be used for the memory 42 in the web socket server 38.

[0040] The communication interface 44 in the web socket server 38 is used to operatively couple and communicate between the web socket server 38, the presentation device 14, and the remote device 16, which are all coupled together via the communication network 18, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. By way of example only, the communication network 18 can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP) and the

- 10 -

web socket protocol, although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can also be used.

5 [0041] The embodiments of the remote interface server 12, web socket server 38, presentation device 14, and remote device 16 are described and illustrated herein for exemplary purposes and many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s). Furthermore, each of the
10 devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

15 [0042] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The
20 embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular
25 communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0043] The embodiments may also be embodied as one or more non-transitory computer readable media having instructions stored thereon for one or
30 more aspects of this technology as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor,

- 11 -

cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0044] An exemplary method for facilitating a remote interface will now be described with reference to FIGS. 1-15. Referring more specifically to FIG. 3, in step 300 in this example, the remote interface server 12 sends the initiation web page 28 to the presentation device 14 and registers the presentation device 14 with the web socket server 34 or 38. The remote interface server 12 can send the initiation web page 28 in response to a request for the initiation web page 28 received from the presentation device 14. In one example, the presentation device 10 14 is a smart television executing a web browser which facilitates the retrieval of the initiation web page 28 at the request of a user, although other types of presentation devices and other methods of providing the initiation web page 28 can also be used. Upon receipt of the initiation web page 28, the presentation device 14 executes JavaScript code included with the initiation web page 28, 15 which is configured to communicate with the remote interface server 12 to register the presentation device 12 by establishing a connection between the presentation device and the web socket server 34 or 38.

[0045] In step 302, the remote interface server 12 sends a remote web page 32 to the remote device 16, registers the remote device 16 with the web 20 socket server 34 or 38 as associated with the presentation device 14, and sends a presentation web page 30 to the presentation device 14. The remote web page 32 and presentation web page 30 can be sent by the remote interface server 12 in response to a request from the remote device 16 initiated based on an interaction by the remote device 16 with at least a portion of the initiation web page 28 25 rendered on the display of the presentation device 14.

[0046] Referring more specifically to FIG. 4, an exemplary initiation web page 28 is illustrated. In this example, the initiation web page 28 includes a portion with an interactive mode interface 400, which is a three dimensional bar code in this example, although other types of interactive mode interfaces and 30 portions of the initiation web page 28 can also be used.

- 12 -

[0047] Accordingly, a user of the remote device 16 in this example can scan the interactive mode interface 400 which encodes at least a Uniform Resource Locator (URL) and causes a web browser executed by the remote device 16 to request the remote web page 32 located at the URL from the remote interface server 12, which sends the remote web page 32 to the remote device 16 in response. In this example, the remote web page 32 is configured to, when executed by the web browser of the remote device 16, register the remote device 16 with the web socket server 34 or 38 as associated with the presentation device 14.

10 [0048] Accordingly, the remote web page 32 can include JavaScript code executed by the remote device 16 that facilitates communication by the remote device 16 with the web socket server 34 or 38 to establish a connection between the remote device 16 and the web socket server 34 or 38, as well as an association with the presentation device 14. Optionally, the interactive mode interface 400 of the initiation web page 28 can further encode an identifier of the presentation device 14 which can be used to facilitate the association of the presentation device 14 and the remote device 16 with the web socket server 34 or 38. Other methods of initiating the association of the presentation device 14 and the remote device 16 with the web socket server 34 or 38 can also be used.

20 [0049] In response to receipt of the request from the remote device 16 for the remote web page 32, or in response to a subsequent communication to the remote interface server 12 by the remote device 16 executing the JavaScript code of the remote web page 32, the remote interface server 12 also sends the presentation web page 30 to the presentation device 14. The remote web page 32 and presentation web page 30 can be different versions of a same web page such that the remote web page 32 includes at least a portion of the content of the presentation web page 30, although other types of presentation and remote web pages can also be used.

[0050] Referring more specifically to FIG. 5, an exemplary presentation web page 30 is illustrated and referring more specifically to FIG. 6, an exemplary remote web page 32 is illustrated. In this example, the presentation web page 30

- 13 -

and remote web page 32 are different versions of the same web page as the presentation web page includes multiple panels that can be manipulated, as described and illustrated in more detail later, whereas the remote web page 32 is a mobile version of the web page which includes content of only one of the panels included in the presentation web page 30. The remote web page 32 is also configured to, when executed by the remote device 16, render a swipe panel 600, and optionally one or more buttons, on the display of the remote device 16. In this example, the swipe panel 600 includes the content of the panel corresponding to one of the panels of the presentation web page 30, although the swipe panel 600 can be located elsewhere in the web page and/or display of the remote device 16.

[0051] Referring back to FIG. 3, in step 304 the web socket server 34 or 38 receives a message from the remote device 16 in response to a user interaction with the remote web page 32. The message can comply with the web socket protocol and can be received by the web socket server 34 or 38 using the connection established with the remote device 16. Accordingly, the JavaScript code of the remote web page 32 executed by the remote device 16 can determine when a user has interacted with the remote web page 32 and send a message to the web socket server 34 or 38 corresponding to the interaction in response. The message can include information regarding the type of interaction and any other contextual information, for example.

[0052] In step 306, the web socket server 34 or 38 determines whether the user interaction corresponding to the message received in step 304 is a swipe panel interaction, and optionally whether the interaction was a horizontal or vertical swipe gesture, for example. If the web socket server 34 or 38 determines that the user interaction is a swipe panel interaction, then the Yes branch is taken to step 308. In step 308, the web socket server 34 or 38 sends a message to the presentation device 14 to cause the presentation device 14 to perform an action on the presentation web page 30 corresponding to the swipe panel interaction.

[0053] Referring more specifically to FIG. 7, a presentation web page 30 subsequent to performing an action corresponding to a horizontal swipe gesture with the swipe panel 600 of the remote web page 32 is illustrated. In this example,

- 14 -

the panel with a video illustrated in the foreground in FIG. 5 has been rotated to the left, such as by three dimensional rotation, for example, resulting in the modified presentation web page 30 illustrated in FIG. 7 in which a new panel has been rotated to the foreground.

5 [0054] Referring more specifically to FIG. 8, a remote web page 32 modified according to a horizontal swipe gesture with the swipe panel 600 of the remote web page 32 is illustrated. In this example, the panel with the video illustrated in the swipe panel 600 in FIG. 6 has been moved off screen, such as by two dimensional slide animation, for example, resulting in the modified remote
10 web page 32 illustrated in FIG. 8 in which a new panel with different content has replaced the previous panel in the swipe panel 600.

[0055] In other examples, the user interaction can be a vertical swipe gesture and the action can be a vertical scroll. For example, a user can perform a vertical swipe gesture on the swipe panel 600 of the remote web page illustrated in
15 FIG. 8 resulting in a vertical scroll action on the presentation web page 30 illustrated in FIG. 7. Other exemplary gestures and interactions with the swipe panel 600 and corresponding actions, as well as animations and rotations can also be used. Accordingly, in this example, a user of the remote device 16 can contemporaneously control the display of the presentation device 14, and in
20 particular the presentation web page 30, without physically interacting with the presentation device 14 and using only the interface provided on the remote device 16 through the remote web page 32.

[0056] Referring back to FIG. 3, if the web socket server 34 or 38 determines that the user interaction is not a swipe panel interaction in step 306,
25 then the No branch is taken to step 310. In step 310, the web socket server 34 or 38 determines whether the user interaction corresponding to the message received in step 304 is a save button interaction. If the web socket server 34 or 38 determines that the user interaction is a save button interaction, then the Yes branch is taken to step 312.

- 15 -

[0057] Optionally, the buttons of the remote web page 32 as rendered on the display of the remote device 16 can change based on functionality present in the remote web page 32 and/or presentation web page 30. For example, referring back to FIG. 6, the remote web page 32 includes play button 602 corresponding to the video content rendered in the swipe panel 600. Referring more specifically to FIG. 9, a presentation web page 30 with input fields 900 is illustrated and referring more specifically to FIG. 10, a remote web page 32 with the input fields 1000 is illustrated. In this example, the buttons are modified by the JavaScript code of the remote web page 32 to include an edit button 1002 corresponding to the content of the input fields 1000 of the remote web page 32.

[0058] Referring more specifically to FIG. 11, the remote web page 32 of FIG. 10 is illustrated subsequent to user interaction with the edit button 1002. Upon user interaction with the edit button 1002, the remote web page 32 is configured to render a save button 1100 in place of the edit button 1002 as well as editable input fields 1000 corresponding to the input fields 900 of the presentation web page 30 of FIG. 9. In this example, private information such as a credit card number is optionally obfuscated in the presentation web page 30 since the presentation web page 30 is rendered on a presentation device 14 which may have a relatively large display and/or may be visible to the environment or other members of the public. However, the editable input fields 1000 rendered by the remote web page 32 in response to the user interaction with the edit button 1002 are rendered without the obfuscation to allow user editing.

[0059] Referring more specifically to FIG. 12, optionally, the remote device 16 is configured to display a virtual keyboard 1200 upon user selection of one of the editable input fields 1000 allowing the user to edit the information. In this example, the user has edited the name, credit card number, and CVV fields. Upon entering the new information, the user can select the save button 1100 as illustrated in FIG. 11. In response to user selection of the save button 1100, the message received by the web socket server 34 or 38 in step 304 of FIG. 3 is sent by the remote device 16 and includes at least any information updated by the user.

- 16 -

[0060] In step 312, the web socket server 34 or 38 sends a message to the presentation device 14 including information included in the message received from the remote device 16 in step 304. Referring more specifically to FIG. 13, the presentation web page 30 with input fields 900 subsequent to user editing of the content is illustrated. In this example, any private information continues to be rendered in an obfuscated manner in the presentation web page 30 rendered on the display of the presentation device 14.

[0061] Referring back to FIG. 3, if the web socket server 34 or 38 determines in step 310 that the user interaction is not a save button interaction, then the No branch is taken to step 314. In step 314, the web socket server 34 or 38 determines whether the user interaction corresponding to the message received in step 304 is a video button interaction. If the web socket server 34 or 38 determines that the user interaction is a video button interaction, then the Yes branch is taken to step 316.

[0062] In step 316, the web socket server 34 or 38 sends a message to the presentation device 14 corresponding to the video button interacted with by the user of the remote device 16. Referring back to FIG. 6, user interaction with the play button 602, for example, can cause a message to be sent to the web socket server 34 or 38 which, in step 316 of FIG. 3, sends a message to the presentation device 16 to initiate the video of the presentation web page 30 in response. Referring more specifically to FIG. 14, the presentation web page 30 subsequent to user interaction with the play button 602 of the remote web page 32 is illustrated.

[0063] In FIG. 15, the remote web page 32 subsequent to user interaction with the play button 602 of the remote web page 32 is illustrated. Optionally, in this example, the remote web page 32 is configured to convert the swipe panel 600 to indicate that the video is playing and to render a pause button 1500 in place of the play button 602, although the remote web page 32 can be configured to provide other functionality in response to the user interaction with the play button 602. In another example, user interaction with the pause button 1500 of the remote web page 32 can be determined in step 316, which can cause a message to

- 17 -

be sent to the presentation device 14 to pause the video of the presentation web page 30 in response. In yet other examples, a stop button can be rendered on the remote web page 32 and any other type of button can also be used.

[0064] Referring back to FIG. 3, in step 318, the web socket server 34 or
5 38 optionally determines whether a message is received from the presentation device 14 in response to the message sent to the presentation device 14 in step 316 in examples in which the video button interaction is a user interaction with a play button 602. Optionally, one or more callbacks can be received by the web socket server 34 or 38 from the presentation device 14 after any of the messages sent in
10 the after any of steps 308, 312, or 316. However, in this example, the message received from the presentation device 14 in step 318 optionally includes video information (e.g., elapsed time) and/or a URL. The URL can correspond with content displayed in the video. For example, if the video is of a model on a runway at a fashion show, the URL can point to content including information
15 regarding an article of the clothing worn by the model including associated cost and purchase information.

[0065] Accordingly, if the web socket server 34 or 38 determines that a message is received from the presentation device 14 in step 318, then the Yes branch is taken to step 320. In step 320, the web socket server 34 or 38 sends a
20 message to the remote device 16 in response to the message received from the presentation device 14 in step 316. The message sent by the web socket server 34 or 38 in step 320 can include the video information and/or the URL included in the message received from the presentation device 14 in step 318. In response, the remote web page 32 can be configured to render the video information and/or
25 content located at the URL on the display of the remote device 16, such as on the swipe panel 600 for example.

[0066] Referring back to step 318, if the web socket server determines a message is not received from the presentation device 14, then the No branch is taken back to step 304 and the web socket server 34 or 38 receives another
30 message from the remote device 16 in response to a subsequent user interaction with the remote web page 32. Referring back to step 314, if the web socket server

- 18 -

34 or 38 determines that the user interaction is not a video button interaction, then the No branch is taken to step 322. In step 322, the web socket server 34 or 38 sends a message to the presentation device 14, if necessary, to reproduce activity associated with the remote web page 32.

5 [0067] Accordingly, while the swipe panel, save button, and video button user interactions have been described and illustrated earlier by way of example only, other interactions with the remote web page 32 are possible. In response to the user interactions, the remote web page 32 is configured to send a message to the web socket server 34 or 38, if necessary, which is configured to identify the
10 associated presentation device 14 and send a corresponding message to the presentation device 14 to modify the presentation web page 30 accordingly. Thereby, a user of the remote device 16 interacting with the remote web page 32 can effectively control the presentation web page 30 rendered on the display of the presentation device 14.

15 [0068] Accordingly, with this technology, presentation devices can be seamlessly controlled by remote devices using messages exchanged based on the web socket protocol. As the remote devices are used to facilitate an interface, using specially programmed web pages and without any dedicated hardware or software, the presentation devices can be less complex and less costly and do not
20 require multi-touch displays or any other physical interfaces. Additionally, private information can be advantageously submitted, such as with respect to facilitating product purchases, without displaying the information in a visible format on the display of the presentation device, which is visible publicly in many environments.

25 [0069] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations,
30 improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of

- 19 -

processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

5

- 20 -

CLAIMS

What is claimed is:

1. A method for facilitating a remote interface, the method comprising:
 - 5 providing, by a remote interface server computing device, a presentation web page to a presentation device and a remote web page to a remote device, the remote web page configured to, when executed by the remote device, register the remote device as associated with the presentation device and render at least a swipe panel on a display of the remote device;
 - 10 receiving, by the remote interface server computing device, a first message from the remote device in response to a received indication of a user interaction with the swipe panel; and
 - 15 sending, by the remote interface server computing device, a second message to the presentation device in response to receiving the first message, wherein the second message, when executed by the presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

2. The method of claim 1, further comprising providing, by
 - 20 the remote interface server computing device, an initiation web page to the presentation device, the initiation web page configured to register the presentation device and to establish a first connection with a web socket server, wherein: - 25 the remote web page is further configured to register the remote device as associated with the presentation device and to establish a second connection with the web socket server;
 - the first and second messages comply with a web socket protocol; and
 - 30 the presentation and remote web pages are provided to the presentation and remote devices, respectively, in response to a received indication of an interaction by the remote device with at least a portion of the initiation web page.

- 21 -

3. The method of claim 1, wherein the received indication of the user interaction further comprises a received indication of at least one of:

a horizontal swipe gesture resulting in a first panel transition on the remote device and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a second panel transition; or

a vertical swipe gesture and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a vertical scroll.

10

4. The method of claim 1, wherein:

the remote web page is further configured to, when executed by the remote device, render an edit button on the display of the remote device;

the presentation web page further comprises a first input field with obfuscated sensitive information; and

the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the edit button, render an editable second input field with the sensitive information visible on the display of the remote device.

15

5. The method of claim 1, wherein the remote web page is further configured to, when executed by the remote device, render a play button on the display of the remote device, the presentation web page and the remote web page comprise a video, the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the play button, modify the swipe panel to display information retrieved using a uniform resource locator (URL), and the method further comprises:

receiving, by the remote interface server computing device, a third message from the remote device in response to a user selection of the play button;

sending, by the remote interface server computing device, a fourth message to the presentation device in response to the third message, the

20

25

30

- 22 -

fourth message indicating to the presentation device that the video has been initiated;

receiving, by the remote interface server computing device, a fifth message from the presentation device, the fifth message including the URL;
5 and

sending, by the remote interface server computing device, a sixth message to the remote device in response to the fifth message, the sixth message including the URL.

10 6. A remote interface server computing device, comprising a processor and a memory coupled to the processor which is configured to be capable of executing programmed instructions comprising and stored in the memory to:

provide a presentation web page to a presentation device
15 and a remote web page to a remote device, the remote web page configured to, when executed by the remote device, register the remote device as associated with the presentation device and render at least a swipe panel on a display of the remote device;

receive a first message from the remote device in response
20 to a received indication of a user interaction with the swipe panel; and

send a second message to the presentation device in response to receiving the first message, wherein the second message, when executed by the presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user
25 interaction with the swipe panel.

7. The remote interface server computing device of claim 6, wherein the processor is further configured to be capable of executing at least one additional programmed instruction comprising and stored in the memory to
30 provide an initiation web page to the presentation device, the initiation web page configured to register the presentation device and to establish a first connection with a web socket server, wherein:

- 23 -

the remote web page is further configured to register the remote device as associated with the presentation device and to establish a second connection with the web socket server;

5 the first and second messages comply with a web socket protocol; and

the presentation and remote web pages are provided to the presentation and remote devices, respectively, in response to a received indication of an interaction by the remote device with at least a portion of the initiation web page.

10

8. The remote interface server computing device of claim 6, wherein the received indication of the user interaction further comprises a received indication of at least one of:

15 a horizontal swipe gesture resulting in a first panel transition on the remote device and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a second panel transition; or

20 a vertical swipe gesture and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a vertical scroll.

9. The remote interface server computing device of claim 6, wherein:

25 the remote web page is further configured to, when executed by the remote device, render an edit button on the display of the remote device;

the presentation web page further comprises a first input field with obfuscated sensitive information; and

30 the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the edit button, render an editable second input field with the sensitive information visible on the display of the remote device.

- 24 -

10. The remote interface server computing device of claim 6, wherein the remote web page is further configured to, when executed by the remote device, render a play button on the display of the remote device, the presentation web page and the remote web page comprise a video, the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the play button, modify the swipe panel to display information retrieved using a uniform resource locator (URL), and the processor is further configured to be capable of executing at least one additional programmed instruction comprising and stored in the memory to:
- 5 receive a third message from the remote device in response to a user selection of the play button;
- send a fourth message to the presentation device in response to the third message, the fourth message indicating to the presentation device that the video has been initiated;
- 15 receive a fifth message from the presentation device, the fifth message including the URL; and
- send a sixth message to the remote device in response to the fifth message, the sixth message including the URL.

- 20 11. A non-transitory computer readable medium having stored thereon instructions for facilitating a remote interface comprising executable code which when executed by a processor, causes the processor to perform steps comprising:
- 25 providing a presentation web page to a presentation device and a remote web page to a remote device, the remote web page configured to, when executed by the remote device, register the remote device as associated with the presentation device and render at least a swipe panel on a display of the remote device;
- 30 receiving a first message from the remote device in response to a received indication of a user interaction with the swipe panel; and
- sending a second message to the presentation device in response to receiving the first message, wherein the second message, when executed by the presentation device, is configured to cause the presentation device

- 25 -

to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

12. The non-transitory computer readable medium of claim 11,
5 wherein the executable code when executed by the processor further causes the processor to perform at least one additional step comprising providing an initiation web page to the presentation device, the initiation web page configured to register the presentation device and to establish a first connection with a web socket server, wherein:
- 10 the remote web page is further configured to register the remote device as associated with the presentation device and to establish a second connection with the web socket server;
- the first and second messages comply with a web socket protocol; and
- 15 the presentation and remote web pages are provided to the presentation and remote devices, respectively, in response to a received indication of an interaction by the remote device with at least a portion of the initiation web page.

- 20 13. The non-transitory computer readable medium of claim 11, wherein the received indication of the user interaction further comprises a received indication of at least one of:
- a horizontal swipe gesture resulting in a first panel transition on the remote device and the second message, when executed by the
25 presentation device, is configured to cause the presentation device to execute a second panel transition; or
- a vertical swipe gesture and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a vertical scroll.

- 30 14. The non-transitory computer readable medium of claim 11, wherein:

- 26 -

the remote web page is further configured to, when executed by the remote device, render an edit button on the display of the remote device;

5 the presentation web page further comprises a first input field with obfuscated sensitive information; and

the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the edit button, render an editable second input field with the sensitive information visible on the display of the remote device.

10

15 15. The non-transitory computer readable medium of claim 11, wherein the remote web page is further configured to, when executed by the remote device, render a play button on the display of the remote device, the presentation web page and the remote web page comprise a video, the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the play button, modify the swipe panel to display information retrieved using a uniform resource locator (URL), and the executable code when executed by the processor further causes the processor to perform at least one additional step comprising:

20 receiving a third message from the remote device in response to a user selection of the play button;

sending a fourth message to the presentation device in response to the third message, the fourth message indicating to the presentation device that the video has been initiated;

25 receiving a fifth message from the presentation device, the fifth message including the URL; and

sending a sixth message to the remote device in response to the fifth message, the sixth message including the URL.

30 16. A system for facilitating a remote interface, the system comprising:

a remote interface server computing device comprising a first processor and a first memory coupled to the first processor which is

- 27 -

configured to be capable of executing programmed instructions comprising and stored in the first memory to:

provide a presentation web page to a presentation device and a remote web page to a remote device, the remote web page configured to, when executed by the remote device, register the remote device as associated with the presentation device and render at least a swipe panel on a display of the remote device; and

a web socket server computing device comprising a second processor and a second memory coupled to the second processor which is configured to be capable of executing programmed instructions comprising and stored in the second memory to:

receive a first message from the remote device in response to a user interaction with the swipe panel; and

send a second message to the presentation device in response to receiving the first message, wherein the second message, when executed by the presentation device, is configured to cause the presentation device to perform an action on the presentation web page corresponding to the user interaction with the swipe panel.

17. The system of claim 16, wherein the first processor is further configured to be capable of executing at least one additional programmed instruction comprising and stored in the first memory to provide an initiation web page to the presentation device, the initiation web page configured to register the presentation device and to establish a first connection with the web socket server computing device, wherein:

the remote web page is further configured to register the remote device as associated with the presentation device and to establish a second connection with the web socket server computing device;

the first and second messages comply with a web socket protocol; and

the presentation and remote web pages are provided to the presentation and remote devices, respectively, in response to a received indication

- 28 -

of an interaction by the remote device with at least a portion of the initiation web page.

18. The system of claim 16, wherein the received indication of the user interaction further comprises a received indication of at least one of:
- a horizontal swipe gesture resulting in a first panel transition on the remote device and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a second panel transition; or
 - a vertical swipe gesture and the second message, when executed by the presentation device, is configured to cause the presentation device to execute a vertical scroll.

19. The system of claim 16, wherein:
- the remote web page is further configured to, when executed by the remote device, render an edit button on the display of the remote device;
 - the presentation web page further comprises a first input field with obfuscated sensitive information; and
 - the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the edit button, render an editable second input field with the sensitive information visible on the display of the remote device.

20. The system of claim 16, wherein the remote web page is further configured to, when executed by the remote device, render a play button on the display of the remote device, the presentation web page and the remote web page comprise a video, the remote web page is further configured to, when executed by the remote device and in response to receiving a user selection of the play button, modify the swipe panel to display information retrieved using a uniform resource locator (URL), and the second processor is further configured to be capable of executing at least one additional programmed instruction comprising and stored in the second memory to:

- 29 -

receive a third message from the remote device in response to a user selection of the play button;

send a fourth message to the presentation device in response to the third message, the fourth message indicating to the presentation
5 device that the video has been initiated;

receive a fifth message from the presentation device, the fifth message including the URL; and

send a sixth message to the remote device in response to the fifth message, the sixth message including the URL.

10

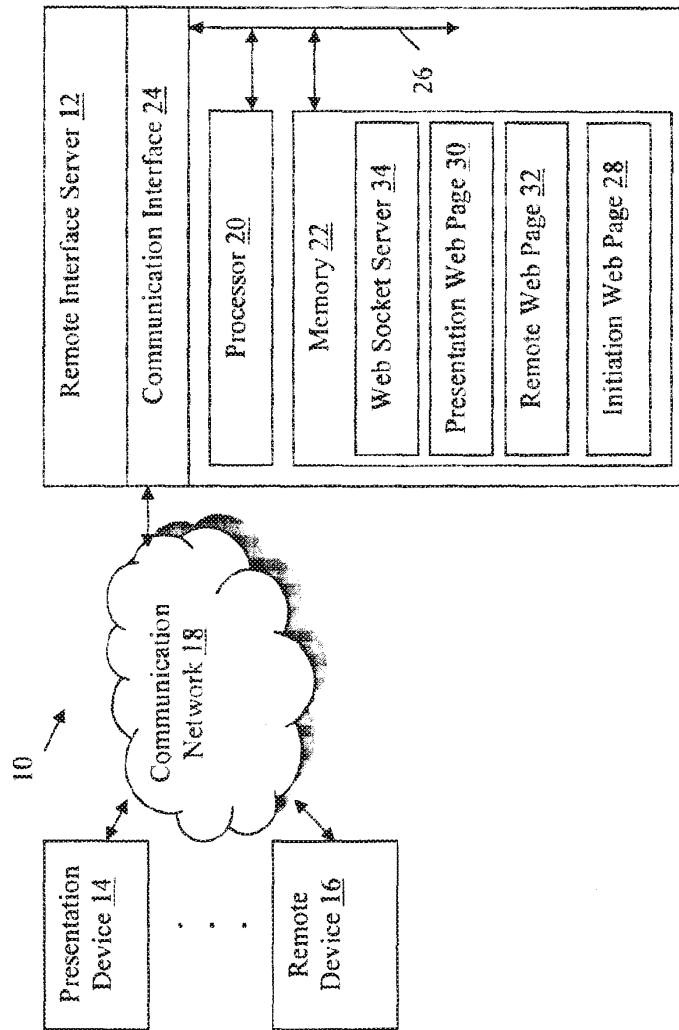


FIG. 1

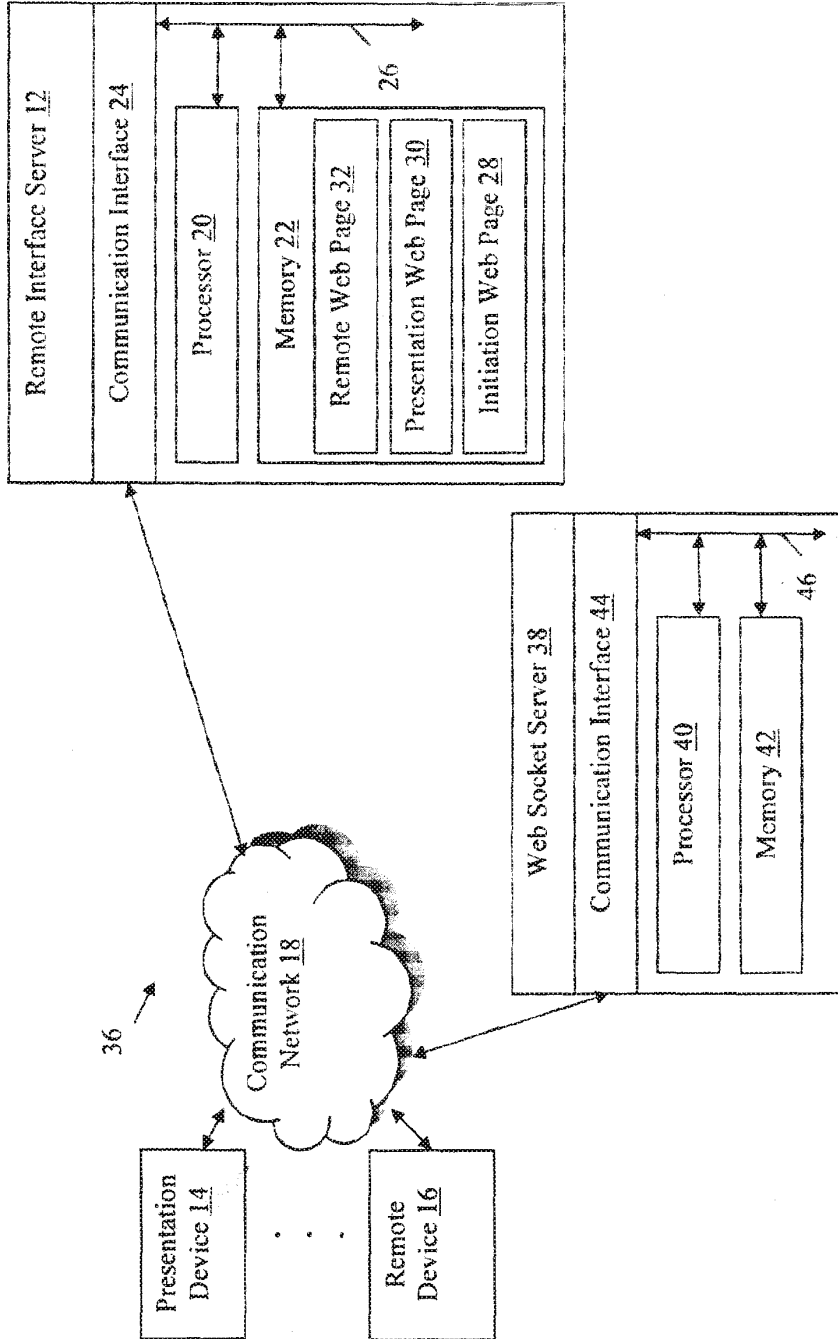


FIG. 2

3/10

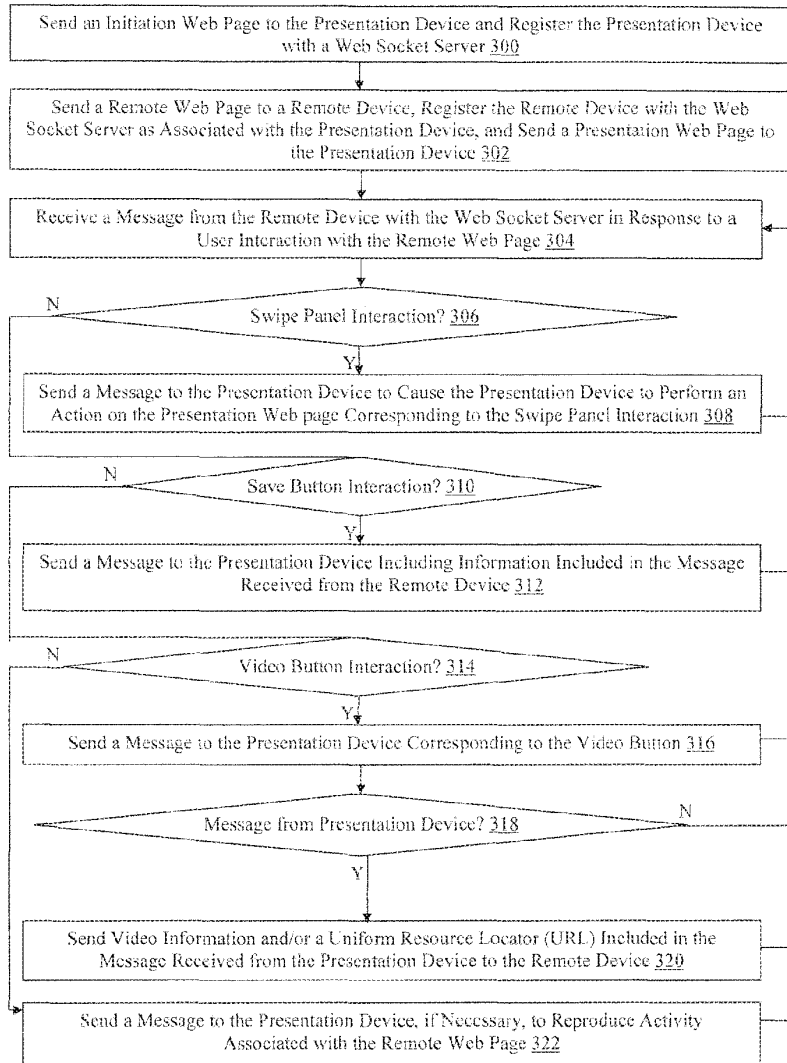


FIG. 3

4/10

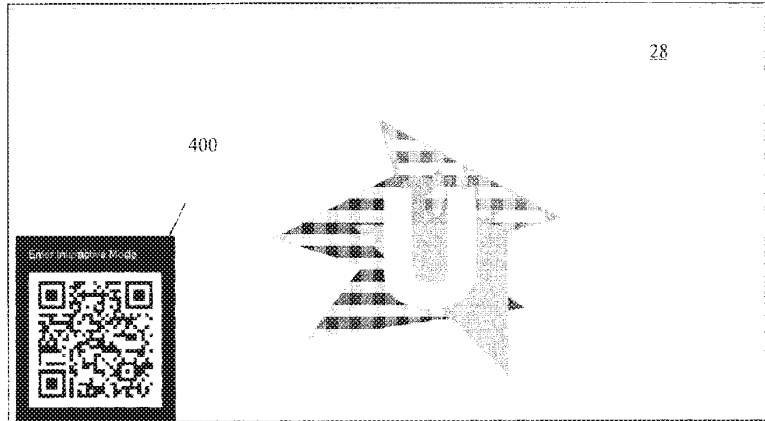


FIG. 4



30

FIG. 5

5/10

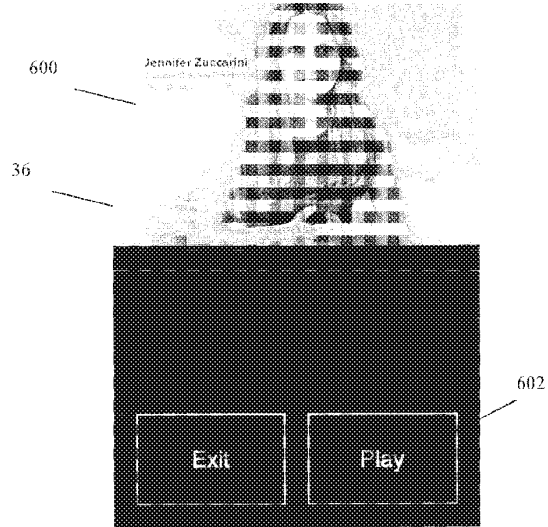


FIG. 6

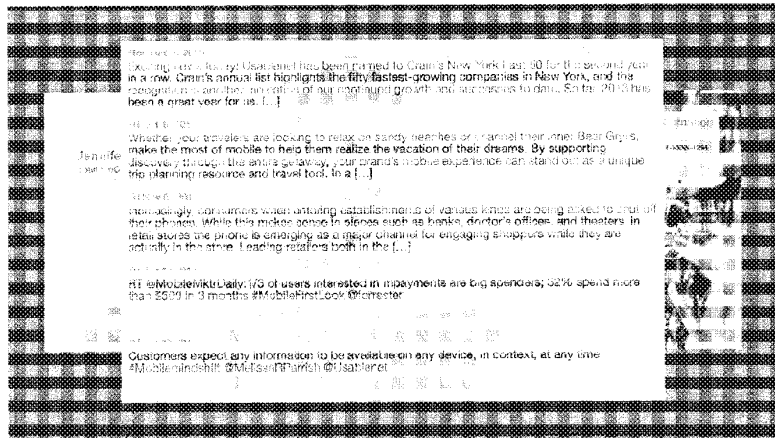


FIG. 7

30

6/10

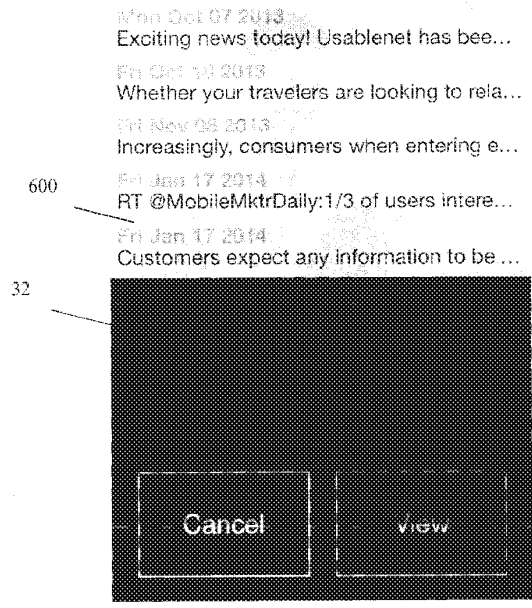


FIG. 8

7/10



FIG. 9

30

900

Payment Information

Name: Arthur Dent

Card: XXXX-XXXX-XXXX-1234

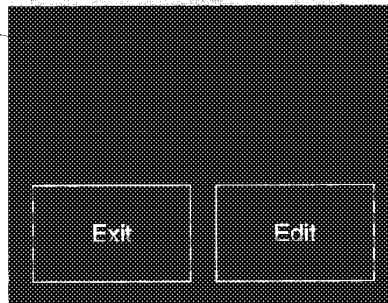
Expires: 07/2014

CVV: XXX

600

1000

32



1002

FIG. 10

8/10

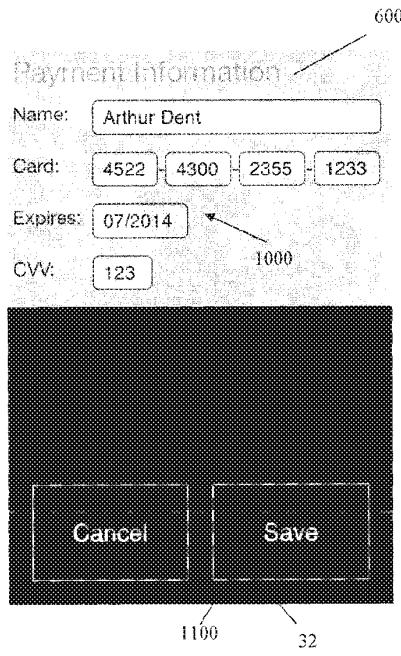


FIG. 11

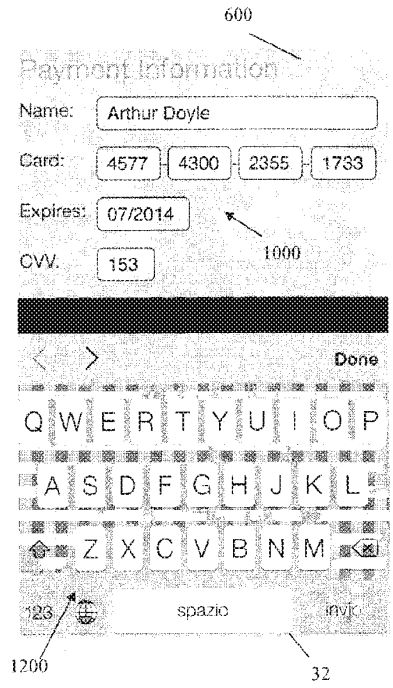


FIG. 12

9/10



FIG. 13

30

900

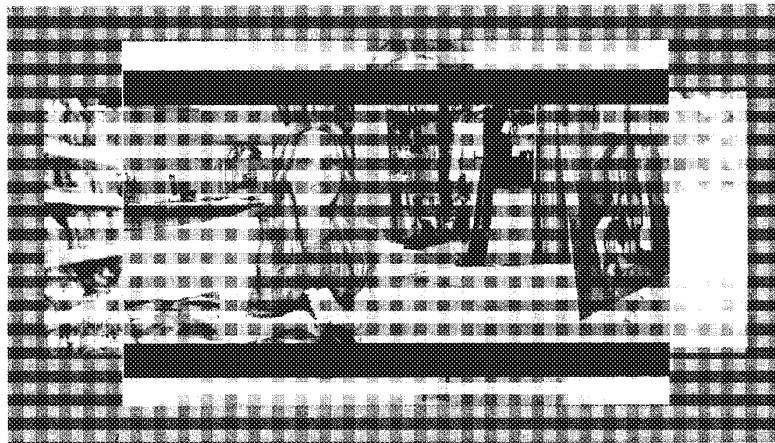


FIG. 14

30

10/10

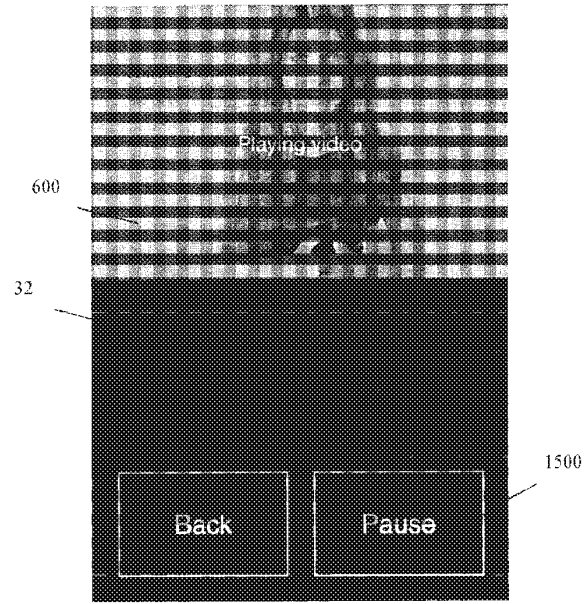


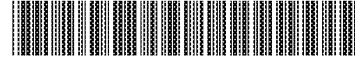
FIG. 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/024172

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 9/445 (2015.01) CPC - G06F 9/54 (2015.04) According to International Patent Classification (IPC) or to both national classification and IPC</p>																												
<p>B. FIELDS SEARCHED</p> <p>Minimum documentation searched (classification system followed by classification symbols) IPC(8) - G06F 9/445, H04L 29/06, G06F 13/00, G06F 12/00, H04L 29/08, G06F 15/16, G06F 9/46, G06F 15/00, G06F 9/44 (2015.01) USPC - 709/203, 717/103, 715/740</p> <p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched CPC - G06F 9/54, H04L 51/06, H04L 67/36, H04L 67/14, H04L 69/329, H04L 67/34, G06F 8/60, H04L 29/06, G06F 8/71, H04L 67/125, H04L 12/2818 (2015.04) (keyword delimited)</p> <p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) Orbit, Google Patents, Google Scholar, Google. Search terms used: facilitate remote interface, presentation web page, presentation device, remote web page, remote device, register remote device</p>																												
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1"> <thead> <tr> <th>Category*</th> <th>Citation of document, with indication, where appropriate, of the relevant passages</th> <th>Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>Y</td> <td>US 2009/0156251 A1 (CANNISTRARO et al) 18 June 2009 (18.06.2009), entire document</td> <td>1-20</td> </tr> <tr> <td>Y</td> <td>US 2013/0339850 A1 (HARDI et al) 19 December 2013 (19.12.2013), entire document</td> <td>1-20</td> </tr> <tr> <td>Y</td> <td>US 2011/0072355 A1 (CARTER et al) 24 March 2011 (24.03.2011), entire document</td> <td>4, 9, 14, 19</td> </tr> <tr> <td>Y</td> <td>US 2002/0162121 A1 (MITCHELL) 31 October 2002 (31.10.2002), entire document</td> <td>5, 10, 15, 20</td> </tr> <tr> <td>A</td> <td>US 2013/0024499 A1 (FRANCO et al) 24 January 2013 (24.01.2013), entire document</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>US 2011/0307855 A1 (SCODA et al) 15 December 2011 (15.12.2011), entire document</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>US 2013/0283182 A1 (DADU et al) 24 October 2013 (24.10.2013), entire document</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>US 2013/0212484 A1 (JOSHI et al) 15 August 2013 (15.08.2013), entire document</td> <td>1-20</td> </tr> </tbody> </table>		Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	Y	US 2009/0156251 A1 (CANNISTRARO et al) 18 June 2009 (18.06.2009), entire document	1-20	Y	US 2013/0339850 A1 (HARDI et al) 19 December 2013 (19.12.2013), entire document	1-20	Y	US 2011/0072355 A1 (CARTER et al) 24 March 2011 (24.03.2011), entire document	4, 9, 14, 19	Y	US 2002/0162121 A1 (MITCHELL) 31 October 2002 (31.10.2002), entire document	5, 10, 15, 20	A	US 2013/0024499 A1 (FRANCO et al) 24 January 2013 (24.01.2013), entire document	1-20	A	US 2011/0307855 A1 (SCODA et al) 15 December 2011 (15.12.2011), entire document	1-20	A	US 2013/0283182 A1 (DADU et al) 24 October 2013 (24.10.2013), entire document	1-20	A	US 2013/0212484 A1 (JOSHI et al) 15 August 2013 (15.08.2013), entire document	1-20
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.																										
Y	US 2009/0156251 A1 (CANNISTRARO et al) 18 June 2009 (18.06.2009), entire document	1-20																										
Y	US 2013/0339850 A1 (HARDI et al) 19 December 2013 (19.12.2013), entire document	1-20																										
Y	US 2011/0072355 A1 (CARTER et al) 24 March 2011 (24.03.2011), entire document	4, 9, 14, 19																										
Y	US 2002/0162121 A1 (MITCHELL) 31 October 2002 (31.10.2002), entire document	5, 10, 15, 20																										
A	US 2013/0024499 A1 (FRANCO et al) 24 January 2013 (24.01.2013), entire document	1-20																										
A	US 2011/0307855 A1 (SCODA et al) 15 December 2011 (15.12.2011), entire document	1-20																										
A	US 2013/0283182 A1 (DADU et al) 24 October 2013 (24.10.2013), entire document	1-20																										
A	US 2013/0212484 A1 (JOSHI et al) 15 August 2013 (15.08.2013), entire document	1-20																										
<p><input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/></p>																												
<p>* Special categories of cited documents:</p> <table border="0"> <tr> <td> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </td> <td> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p> </td> </tr> </table>		<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>																									
<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>																											
<p>Date of the actual completion of the international search</p> <p>14 June 2015</p>	<p>Date of mailing of the international search report</p> <p>06 JUL 2015</p>																											
<p>Name and mailing address of the ISA/US</p> <p>Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300</p>	<p>Authorized officer:</p> <p>Blaine R. Copenheaver</p> <p>PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774</p>																											

Form PCT/ISA/210 (second sheet) (July 2009)



(11) **EP 3 156 919 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
19.04.2017 Bulletin 2017/16

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

(21) Application number: 16193291.8

(22) Date of filing: 11.10.2016

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
MA MD

(72) Inventors:
• **SCODA, Enrico**
33035 MARTIGNACCO (IT)
• **PITTINO, Luca**
33050 POZZUOLO DEL FRIULI (IT)
• **PEZZANO, Simone**
33010 TAVAGNACCO (IT)

(30) Priority: 15.10.2015 US 201514883886

(74) Representative: **Savi, Massimiliano et al**
NOTARBARTOLO & GERVASI S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(54) **METHODS FOR TRANSFORMING A SERVER SIDE TEMPLATE INTO A CLIENT SIDE TEMPLATE AND DEVICES THEREOF**

(57) A method, non-transitory computer readable medium, and web content server that assists with transforming server side template to client side template includes obtaining a server side template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive to a request for webpage received from a client device. The obtained server side template is parsed to generate a document object model. The parsed document object model is traversed to identify the plurality of HTML and logical instructions. The obtained server side template is transformed to a client side template by replacing each of the identified plurality of HTML and logical instructions during the traversing with an equivalent JavaScript code.

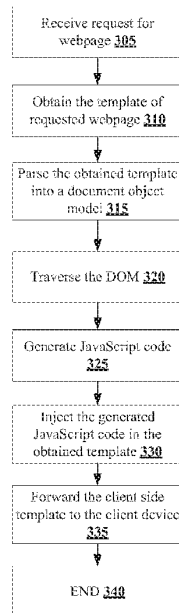


FIG. 3

EP 3 156 919 A1

Description**FIELD**

[0001] This technology generally relates to methods and devices for optimizing web content and, more particularly, methods for transforming a server side template into a client side template and devices thereof.

BACKGROUND

[0002] Many web sites are increasingly sophisticated and provide rich user experiences. Accordingly, in web development, the user or client side dynamic code is often crafted for each specific case to provide rich user experiences. In prior technologies, a dynamic Hypertext Markup Language (HTML) based template (server side template) is used to describe the websites to provide rich user experience. The dynamic HTML based template includes references to variables, express conditions, description of iterations and other types of HTML instructions. Accordingly, with the growth of client side web technologies, a need of having client side templates which would generate HTML directly on the browser grew as well.

[0003] Most commonly, the existing products working as client side templates are made of HTML extensions and are using JavaScript as logical language. However, as previously illustrated, the server side templates are dynamic HTML based templates which do not share most peculiarities with the current client side implementations. Therefore with prior technologies, software developers were required to know multiple programming and markup languages in order to develop and transform the server side template to client side templates. As noted earlier, with many websites having a sophisticated layout to provide rich user experience, the manual transformation of both server side templates and client side templates of one into another became a tedious task. Additionally, due to large amounts of human intervention (web developers), the methods used in the prior technologies also become inefficient and error prone.

SUMMARY

[0004] A method for transforming server side template to a client side template includes obtaining, by a web content server, a server side template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive to a request for webpage received from a client device. The obtained server side template is parsed by the web content server to generate a document object model. The parsed document object model is traversed by the web content server to identify the plurality of HTML and logical instructions. The obtained server side template is transformed by the web content server to a client side template by replacing each of the identified plurality of HTML instructions and logical

during the traversing with an equivalent JavaScript code.

[0005] A non-transitory computer readable medium having stored thereon instructions transforming server side template to a client side template comprising machine executable code which when executed by a processor, causes the processor to perform steps including obtaining a server side template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive to a request for webpage received from a client device. The obtained server side template is parsed to generate a document object model. The parsed document object model is traversed to identify the plurality of HTML and logical instructions. The obtained server side template is transformed to a client side template by replacing each of the identified plurality of HTML and logical instructions during the traversing with an equivalent JavaScript code.

[0006] A web content server includes a processor coupled to a memory and configured to execute programmed instructions stored in the memory including obtaining a server side template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive to a request for webpage received from a client device. The obtained server side template is parsed to generate a document object model. The parsed document object model is traversed to identify the plurality of HTML and logical instructions. The obtained server side template is transformed to a client side template by replacing each of the identified plurality of HTML and logical instructions during the traversing with an equivalent JavaScript code.

[0007] This technology provides a number of advantages including methods, non-transitory computer readable medium, and a web content server for transforming templates designed for the server side execution to templates that are suitable to be executed on the client side. Additionally, this technology allows developers to use one language for both developing server side pages and client side content. Further with this technology, the server side templates can be transformed to client side templates that allow the creation of complex web pages where the displayed data can change without the page being reloaded. Even further, with this technology by directly transforming the server side template to a client side template, an autonomous JavaScript based template is created for the client side without the need of an intermediate template language to be reinterpreted on the browser of the client device.

BRIEF DESCRIPTION OF THE DRAWINGS**[0008]**

FIG. 1 is a block diagram of an environment with an example of a web content server that effectively transforms a server side template into a client side template;

FIG. 2 is a block diagram of the example of the web content server shown in FIG. 1; and

FIG. 3 is a flow chart of an example of a method for transforming server side template into a client side template;

FIGS. 4-5 are exemplary server side templates; and

FIGS. 6-7 are exemplary client side templates that are transformed from the server side templates.

DETAILED DESCRIPTION

[0009] An exemplary environment 10 with a web content server 14 coupled to plurality of client computing devices 12(1)-12(n) and plurality of server devices 16(1)-16(n) by communication network 30 is illustrated in FIG. 1. Other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can also be used. This technology provides a number of advantages including providing methods, non-transitory computer readable medium, and devices for transforming templates designed for the server side execution to templates that are suitable to be executed on the client side.

[0010] Referring more specifically to FIG. 1, the web content server 14 includes a central processing unit (CPU) 18 or processor, a memory 20, and a communication interface 24, which are coupled together by a bus 26 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used.

[0011] The processor 18 of the web content server 14 may execute one or more programmed instructions stored in the memory 20 for transforming templates designed for the server side execution to templates that are suitable to be executed on the client side as illustrated and described in the examples herein, although other types and numbers of functions and/or other operation can be performed. The processor 18 of the web content server 14 may include one or more central processing units ("CPUs") or general purpose processors with one or more processing cores, such as AMD® processor(s), although other types of processor(s) could be used (e.g., Intel®).

[0012] The memory 20 of the web content server 14 stores the programmed instructions and other data for one or more aspects of the present technology as described and illustrated herein, although some or all of the programmed instructions could be stored and executed elsewhere. A variety of different types of memory storage devices, such as a non-volatile memory, random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and written to by a magnetic, optical, or other reading and writing system that is coupled to the processor 18, can

be used for the memory 20.

[0013] The communication interface 24 in the web content server 14 is used to operatively couple and communicate between the web content server 14, plurality of client computing devices 12(1)-12(n), and plurality of server devices 16(1)-16(n) via the communication network 30. One or more of the communication network 30 can include one or more networks, such as one or more local area networks (LANs) and/or wide area networks (WANs). By way of example only, the communication networks 30 can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[0014] The plurality of client devices 12(1)-12(n) enable a user to request, receive, and interact with applications, web services, and content hosted by the plurality of server devices 16(1)-16(n) through the web content server 14 via the communication network 30, although one or more of the plurality of client computing devices 12(1)-12(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. In some examples, the plurality of client computing devices 12(1)-12(n) comprise mobile computing devices with Internet access that enable web pages and other content stored by the plurality of server devices 16(1)-16(n) to be retrieved and rendered. By way of example only, the plurality of client computing devices 12(1)-12(n) can be smart phones, personal digital assistants, or computers.

[0015] Each of the plurality of client computing devices 12(1)-12(n) includes a CPU, a memory, an input device, a display device, and an input/output device, which are coupled together by a bus or other link, although one or more of plurality of client computing devices 12(1)-12(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The CPU in the plurality of client computing devices 12(1)-12(n) can execute a program of instructions stored in the memory of the plurality of client computing devices 12(1)-12(n) for one or more aspects of the present invention as described and illustrated herein, although the CPU could execute other numbers and types of programmed instructions.

[0016] The input device in each of the plurality of client computing devices 12(1)-12(n) can be used to input selections, such as a request for a particular web page, although the input device could be used to input other types of requests and data and interact with other elements. The input device can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can be used.

[0017] The display device in each of the plurality of client computing devices 12(1)-12(n) can be used to show data and information to the user, such as web pages retrieved from the plurality of server devices 16(1)-16(n) by way of example only. The display device in each of the plurality of client computing devices 12(1)-12(n) can be a mobile phone screen display, although other types and numbers of displays could be used depending on the particular type of client device.

[0018] The input/output device in each of the plurality of client computing devices 12(1)-12(n) can be used to operatively couple and communicate between the plurality of client computing devices 12(1)-12(n), the web content server 14, and the plurality of server devices 16(1)-16(n) over the communication network 30.

[0019] Each of the plurality of server devices 16(1)-16(n) provides content including web pages for use by one or more of the plurality of client computing devices 12(1)-12(n) via the web content server 14, although the plurality of server devices 16(1)-16(n) can provide other numbers and types of content and perform other functions. Each of the plurality of server devices 16(1)-16(n) can include a CPU, a memory, and an input/output device, which are coupled together by a bus or other link, although each of the plurality of server devices 16(1)-16(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations.

[0020] The CPU in each of the plurality of server devices 16(1)-16(n) executes a program of instructions stored in the memory of the plurality of server devices 16(1)-16(n) for one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the CPU could execute other numbers and types of programmed instructions.

[0021] The input/output device in each of the plurality of server devices 16(1)-16(n) is used to operatively couple and communicate between the plurality of server devices 16(1)-16(n), the web content server 14, and the plurality of client computing devices 12(1)-12(n) via the communication network 30.

[0022] Although embodiments web content server 14, the plurality of client computing devices 12(1)-12(n), and the plurality of server devices 16(1)-16(n) are described and illustrated herein, each of the web content server 14, the plurality of client computing devices 12(1)-12(n), and the plurality of server devices 16(1)-16(n) can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0023] Furthermore, each of the devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, pro-

grammed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0024] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0025] The embodiments may also be embodied as one or more non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the embodiments, as described and illustrated herein.

[0026] An exemplary method for transforming templates designed for the server side execution to templates that are suitable to be executed on the client side will now be described with reference to FIGS. 1-7.

[0027] The exemplary method begins at step 305 where the web content server 14 receives a request for a web page from one of the plurality of client computing devices 12(1)-12(n). In this example, the requested web page is a hypertext markup language (HTML) or extensible markup language template of web page stored by one of the plurality of server devices 16(1)-16(n).

[0028] Upon receipt of the request from one of the plurality of client computing devices 12(1)-12(n), in step 310 the web content server 14 obtains the template of the requested web page. In this example, the web content server 14 obtains the requested web page by retrieving the template of requested web page from the one of the plurality of server devices 16(1)-16(n) on behalf of the requesting one of the plurality of client computing devices 12(1)-12(n). By way of example only, the template of the requested webpage that is obtained by the web content server 14 is illustrated in FIG. 4. Additionally, another example of the server side template obtained by the web content server 14 is illustrated in FIG. 5. As illustrated in FIGS. 4-5, the template (server side template) of the requested webpage obtained by the web content server 14 includes full support of HTML tags and support of special tags that are able to perform some special operations, such as iteration over a collection of items; evaluation of

logical conditions; set of variables; and print of variables, although the server side template can include other types and/or amounts of information.

[0029] In step 315, the web content server 14 parses the obtained server side template and generates a document object model (DOM) based on the parsed content.

[0030] In step 320, the web content server 14 traverses through each line in the DOM to identify: the types of HTML instructions, such as static HTML instructions; HTML tags including any special tags, condition tag, iteration tag; variable print instructions; and variable set instructions, although the web content server 14 can identify other types and/or amounts of instructions or other information. By way of example only, line 402 in FIG. 4 illustrates the condition tag and line 502 of FIG. 5 illustrates the iteration tag that is identified by the web content server 14.

[0031] In step 325, the web content server 14 generates JavaScript code representing the exact same procedure illustrated in the HTML instructions in the DOM as the client side template for each identified type of HTML instruction. By way of example only, the web content server 14 generates the JavaScript code 602 illustrated in FIG. 6 for the condition tag in the HTML instruction illustrated in line 402 of FIG. 4. In this example, when the web content server 14 encounters a condition tag while traversing the DOM, it generates a JavaScript fragment that illustrates the condition and injects the inner content inside the condition tag as illustrated in line 602 of FIG. 6. Additionally, the web content server 14 generates a JavaScript code illustrated in FIG. 7 for the iteration tag illustrated in line 502 of FIG. 5. In this example, when the web content server 14 encounters an iteration tag while traversing the DOM, it invokes a specifically crafted utility function passing the whole iteration tag as a parameter. The function will take care of interpreting the iteration code and transform it into a JavaScript iterator that will also evaluate all the inner content as illustrated in FIG. 7, by way of example only. The dynamic HTML based template extends the basic HTML instructions by adding logical commands such as conditions, iterations and transformations. The result of the server side interpretation of the HTML template is an HTML page that differs in the content by contextual factors.

[0032] In step 330, upon transforming each identified type of HTML instruction to an equivalent JavaScript code, the web content server 14 transforms the server side template to a client side template by injecting the generated JavaScript code in the obtained server side template, although the web content server 14 can inject other types and/or amounts of information. In this example, the resulting client side template includes a number of utility JavaScript functions that can receive parameters. By generating the client side template using this technique, this example of the technology transforms a server side template into an autonomous JavaScript based client side template, without the need of an intermediate template language to be reinterpreted on the

browser of the requesting one of the plurality of client computing devices 12(1)-12(n).

[0033] In step 335, the web content server 14 forwards the client side template to the requesting one of the plurality of client computing devices 12(1)-12(n) responsive to the received request for the webpage. Upon receipt of the client side template, the web browser in the requesting one of the plurality of client computing devices 12(1)-12(n) may execute the client side template resulting in the rendering of the requested webpage. While the web browser of the requesting one of the plurality of client computing devices 12(1)-12(n) executes the client side template, at a certain stage of the flow in the execution of the client side template will trigger the JavaScript code that will retrieve data in JavaScript Object Notion (JSON) format from one of the plurality of server devices 16(1)-16(n) hosting the requested webpage, although the data can be obtained in other formats. This example of the method ends in step 340.

[0034] Accordingly, as illustrated and described by way of the examples herein this technology effectively transforms templates designed for the server side execution to templates that are suitable to be executed on the client side. Additionally, this technology allows developers to use one language for both developing server side pages and client side content. Further with this technology, the server side templates can be transformed to client side templates that allow the creation of complex web pages where the displayed data can change without the page being reloaded. Even further, with this technology by directly transforming the server side template to a client side template, an autonomous JavaScript based template is created for the client side without the need of an intermediate template language to be reinterpreted on the browser of the client device.

[0035] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for transforming server side template to a client side template, the method comprising:
 - obtaining, by a web content server, a server side

- template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive to a request for webpage received from a client device;
- 5 parsing, by the web content server, the obtained server side template to generate a document object model;
- traversing, by the web content server, the generated document object model to identify the plurality of HTML and logical instructions; and
- 10 transforming, by the web content server, the obtained server side template to a client side template by replacing each of the identified plurality of HTML and logical instructions during the traversing with an equivalent JavaScript code.
- 15
2. The method as set forth in claim 1 wherein traversing further comprises identifying, by the web content server, one or more instruction types in the identified plurality of HTML and logical instructions, wherein the one or more instruction types comprises a condition tag, an iteration tag, a variable print instruction and a variable set instruction.
- 20
3. The method as set forth in claim 2 further comprising generating, by the web content server, the equivalent JavaScript code configured to express a condition and inject an inner content of the condition inside a branch of the condition when the identified one or more instruction types is the condition tag.
- 25
4. The method as set forth in claim 2 further comprising generating, by the web content server, the equivalent JavaScript code configured to invoke a specifically crafted utility function to pass the iteration tag as a parameter when the identified one or more instruction types is the iteration tag.
- 30
5. The method as set forth in claim 2 further comprising transforming, by the web content server, the variable print instruction into a JavaScript print instruction when the identified one or more instruction types is the variable print instruction.
- 35
6. The method as set forth in claim 2 further comprising transforming, by the web content server, the variable set instruction into a JavaScript variable set instruction when the identified one or more instruction types is the variable set instruction.
- 40
7. A non-transitory computer readable medium having stored thereon instructions for transforming server side template to a client side template comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:
- 45
- obtaining a server side template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive to a request for webpage received from a client device;
- 50 parsing the obtained server side template to generate a document object model;
- traversing the generated document object model to identify the plurality of HTML and logical instructions; and
- transforming the obtained server side template to a client side template by replacing each of the identified plurality of HTML and logical instructions during the traversing with an equivalent JavaScript code.
- 55
8. The medium as set forth in claim 7 wherein traversing further comprises identifying one or more instruction types in the identified plurality of HTML and logical instructions, wherein the one or more instruction types comprises a condition tag, an iteration tag, a variable print instruction and a variable set instruction.
9. The medium as set forth in claim 8 further comprising generating the equivalent JavaScript code configured to express a condition and inject an inner content of the condition inside a branch of the condition when the identified one or more instruction types is the condition tag.
10. The medium as set forth in claim 8 further comprising generating the equivalent JavaScript code configured to invoke a specifically crafted utility function to pass the iteration tag as a parameter when the identified one or more instruction types is the iteration tag.
11. The medium as set forth in claim 8 further comprising transforming the variable print instruction into a JavaScript print instruction when the identified one or more instruction types is the variable print instruction.
12. The medium as set forth in claim 8 further comprising transforming the variable set instruction into a JavaScript variable set instruction when the identified one or more instruction types is the variable set instruction.
13. A web content server, comprising:
- a processor;
- a memory, wherein the memory coupled to the processor which are configured to execute programmed instructions stored in the memory comprising:
- obtaining a server side template comprising a plurality of Hypertext Markup Language (HTML) and logical instructions responsive

to a request for webpage received from a client device;
 parsing the obtained server side template to generate a document object model;
 traversing the generated document object model to identify the plurality of HTML and logical instructions; and
 transforming the obtained server side template to a client side template by replacing each of the identified plurality of HTML and logical instructions during the traversing with an equivalent JavaScript code.

- 14. The device as set forth in claim 13 wherein the processor is further configured to execute programmed instructions stored in the memory for the traversing further comprises identifying one or more instruction types in the identified plurality of HTML and logical instructions, wherein the one or more instruction types comprises a condition tag, an iteration tag, a variable print instruction and a variable set instruction.
- 15. The device as set forth in claim 14 wherein the processor is further configured to execute programmed instructions stored in the memory further comprising generating the equivalent JavaScript code configured to express a condition and inject an inner content of the condition inside a branch of the condition when the identified one or more instruction types is the condition tag.
- 16. The device as set forth in claim 14 wherein the processor is further configured to execute programmed instructions stored in the memory further comprising generating the equivalent JavaScript code configured to invoke a specifically crafted utility function to pass the iteration tag as a parameter when the identified one or more instruction types is the iteration tag.
- 17. The device as set forth in claim 14 wherein the processor is further configured to execute programmed instructions stored in the memory further comprising transforming the variable print instruction into a JavaScript print instruction when the identified one or more instruction types is the variable print instruction.
- 18. The device as set forth in claim 14 wherein the processor is further configured to execute programmed instructions stored in the memory further comprising transforming the variable set instruction into a JavaScript variable set instruction when the identified one or more instruction types is the variable set instruction.

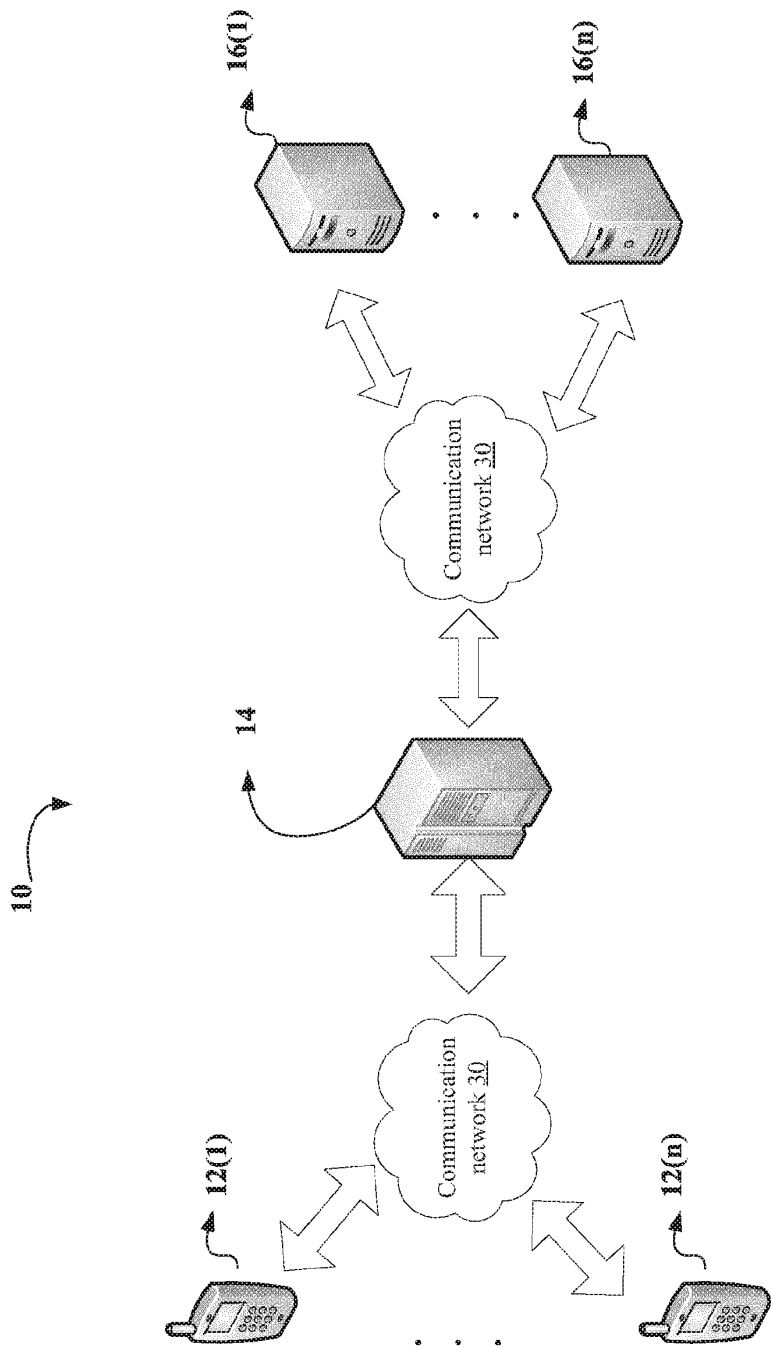


FIG. 1

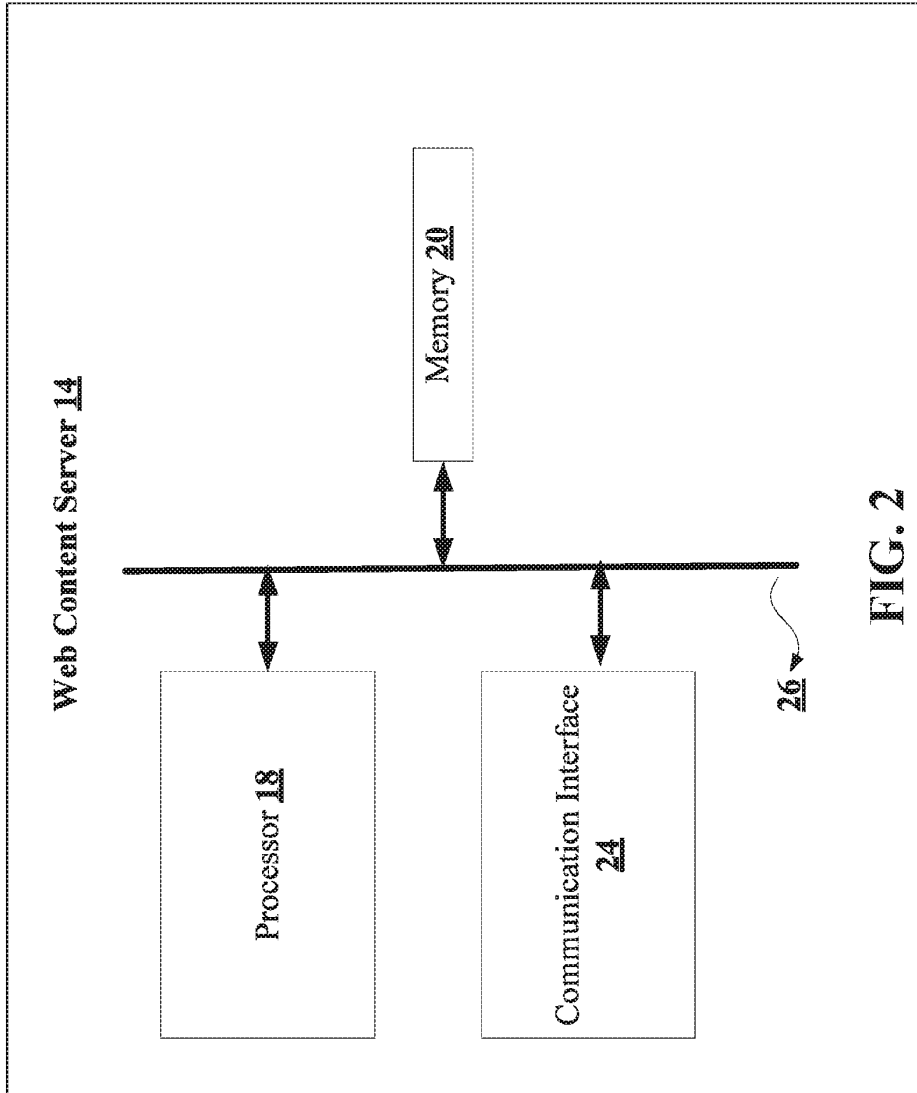


FIG. 2

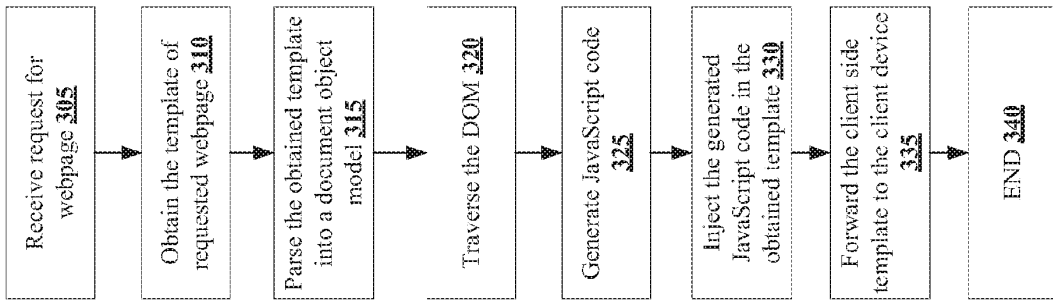


FIG. 3

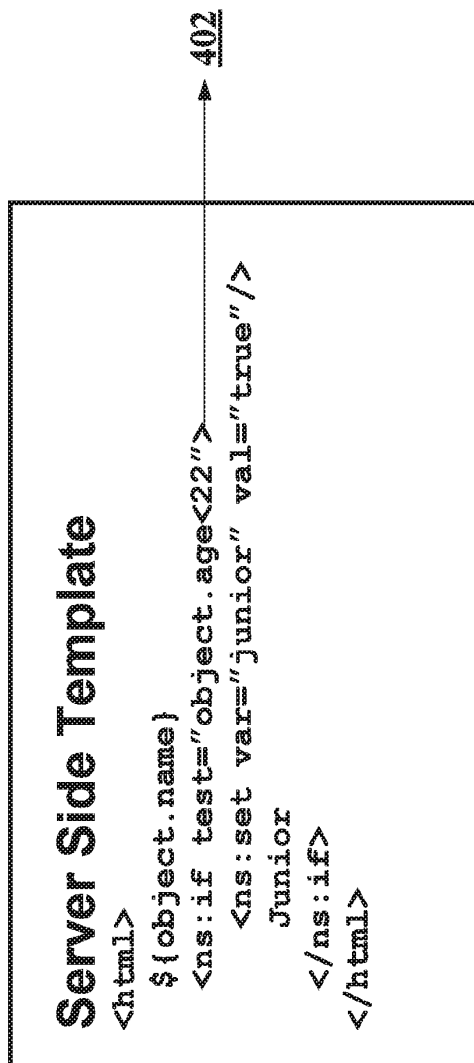


FIG. 4

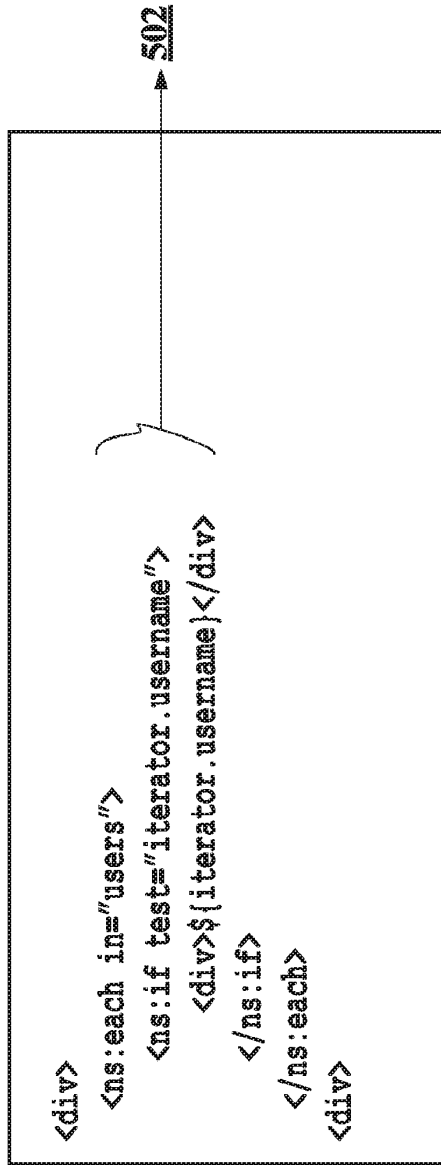


FIG. 5

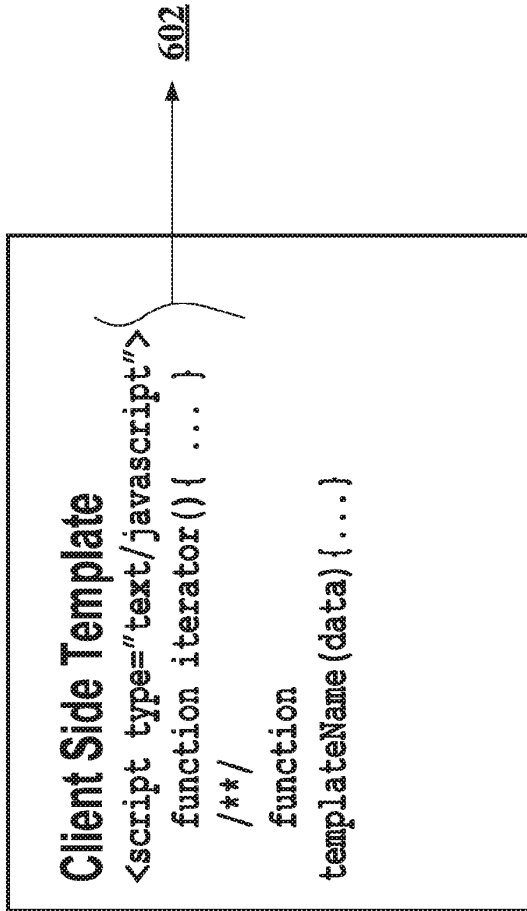


FIG. 6

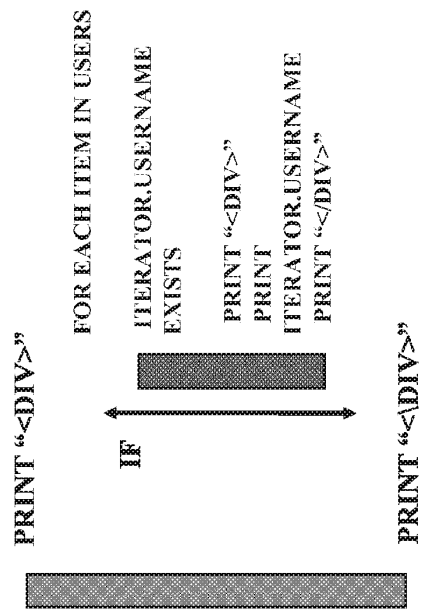


FIG. 7



EUROPEAN SEARCH REPORT

Application Number
EP 16 19 3291

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	Anonymous: "PHP-to-Javascript/README.md at master · Danack/PHP-to-Javascript · GitHub", 1 October 2013 (2013-10-01), XP055339479, Retrieved from the Internet: URL:https://github.com/Danack/PHP-to-Javascript/blob/master/README.md [retrieved on 2017-01-26] * the whole document *	1-18	INV. G06F17/30
X	WO 2008/043050 A2 (DERRICOTT BRETT W [US]; WEBER DOUGLAS M [US]) 10 April 2008 (2008-04-10) * page 5, lines 21-27 * * page 7, lines 17-22 *	1,7,13	
X	EP 2 824 578 A1 (MURAKUMO CORP [JP]) 14 January 2015 (2015-01-14) * paragraphs [0006], [0007], [0033], [0043], [0045], [0053] *	1,7,13	
A	US 2011/161840 A1 (WONG CESAR AUGUSTO [US] ET AL) 30 June 2011 (2011-06-30) * paragraphs [0004], [0015], [0016] *	1-18	
The present search report has been drawn up for all claims			TECHNICAL FIELDS SEARCHED (IPC) G06F
Place of search The Hague		Date of completion of the search 27 January 2017	Examiner Correia Martins, F
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1509 (03/02) (F04C01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 16 19 3291

5

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

27-01-2017

10

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008043050 A2	10-04-2008	US 2008086682 A1 WO 2008043050 A2	10-04-2008 10-04-2008
EP 2824578 A1	14-01-2015	EP 2824578 A1 JP 5154716 B1 JP WO2013132643 A1 US 2014379790 A1 WO 2013132643 A1	14-01-2015 27-02-2013 30-07-2015 25-12-2014 12-09-2013
US 2011161840 A1	30-06-2011	CN 102741811 A DE 112010004980 T5 GB 2488693 A US 2011161840 A1 WO 2011076505 A2	17-10-2012 03-01-2013 05-09-2012 30-06-2011 30-06-2011

15

20

25

30

35

40

45

50

55

EPO FORM P0489

For more details about this annex : see Official Journal of the European Patent Office, No. 12/02



(11) **EP 3 185 145 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
28.06.2017 Bulletin 2017/26

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

(21) Application number: 16203840.0

(22) Date of filing: 13.12.2016

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
MA MD

(72) Inventors:
• **Scoda, Enrico**
33035 Martignacco (IT)
• **Brondani, Marco**
33010 Colloredo di Monte Albano (IT)

(74) Representative: **Savi, Massimiliano et al**
NOTARBARTOLO & GERVASI S.p.A.
Corso di Porta Vittoria 9
20122 Milano (IT)

(30) Priority: 22.12.2015 US 201514977995

(71) Applicant: **Usablenet Inc.**
New York, NY 10019 (US)

(54) **METHODS FOR ANALYZING WEB SITES USING WEB SERVICES AND DEVICES THEREOF**

(57) Methods, non-transitory computer readable media, and devices that determine when a job extracted from a stack is a pending job. When the determining indicates that the job is a pending job, a web service indicated in the job is executed. Another job is extracted from a web service response, and the another job is inserted into the stack. The web service is configured to obtain a web page to be analyzed based on the URL, execute the

web page in an emulated JavaScript environment, and return the web service response. When the determining indicates that the job is not a pending job, a data collector event handler indicated in the job is executed. The data collector event handler is configured to update an output resource based on content of the analyzed web page included in the job.

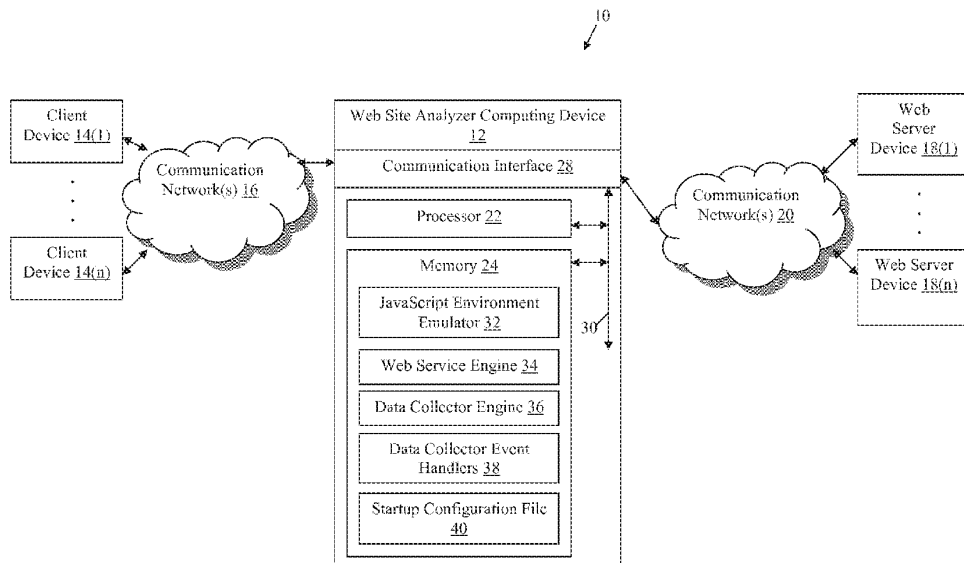


FIG. 1

EP 3 185 145 A1

Description

BACKGROUND

[0001] Web site analyzers, which are often referred to as indexers, spiders, bots, or crawlers, for example, navigate web sites and collect information regarding their structure or content. These analyzers have many uses including identifying security threats in a web site, evaluating web pages associated with a web site for implementation quality, and producing a list or sitemap of web pages of a web site that should be indexed by search engines, for example. In some instances, analyzers can be used to generate client-facing content, such as a list of available products with current offers in a retailer web site or a static version of a retailer web site catalog that can be used as a catalog navigation menu by a mobile application associated with the retailer or web site, for example.

[0002] However, current web site analyzers are limited to inspecting static web page documents associated with web sites. As a result, current web site analyzers do not effectively execute some web pages, including those web pages that include client-side JavaScript code. More specifically, current web site analyzers are unable to extract information from web pages that is hidden inside the client-side JavaScript code. In one particular example, a catalog navigation menu may be generated, when a web page is executed client-side, by downloading and processing a JavaScript Object Notation (JSON) resource, which would not be accessible or executable by current web site analyzers. Therefore, current analyzers have limited functionality and visibility into certain web sites resulting in relatively inaccurate or incomplete results that have limited utility.

SUMMARY

[0003] A method for analyzing web sites using web services includes determining, by a web site analyzer computing device, when a job extracted from a stack is a pending job. When the determining indicates that the job is a pending job, a web service indicated in the job is executed, by the web site analyzer computing device, by passing a Uniform Resource Locator (URL) included in the job as a parameter to the web service. Another job is extracted, by the web site analyzer computing device, from a web service response, and the another job is inserted, by the web site analyzer computing device, into the stack. The web service is configured to obtain a web page to be analyzed based on the URL, execute the web page in an emulated JavaScript environment, and return the web service response. When the determining indicates that the job is not a pending job, then a data collector event handler indicated in that job is executed, by the web site analyzer computing device, by passing that job as a parameter to the data collector event handler. The data collector event handler is configured to update

an output resource based on content of the analyzed web page included in that job.

[0004] A non-transitory computer readable medium having stored thereon programmed instructions for analyzing web sites using web services and includes executable code that, when executed by at least one processor, causes the processor to perform steps including determining when a job extracted from a stack is a pending job. When the determining indicates that the job is a pending job, a web service indicated in the job is executed by passing a URL included in the job as a parameter to the web service. Another job is extracted from a web service response, and the another job is inserted into the stack. The web service is configured to obtain a web page to be analyzed based on the URL, execute the web page in an emulated JavaScript environment, and return the web service response. When the determining indicates that the job is not a pending job, a data collector event handler indicated in that job is executed by passing that job as a parameter to the data collector event handler. The data collector event handler is configured to update an output resource based on content of the analyzed web page included in that job.

[0005] A web site analyzer computing device includes one or more processors coupled to a memory and configured to execute programmed instructions including and stored in the memory to determine when a job extracted from a stack is a pending job. When the determining indicates that the job is a pending job, a web service indicated in the job is executed by passing a URL included in the job as a parameter to the web service. Another job is extracted from a web service response, and the another job is inserted into the stack. The web service is configured to obtain a web page to be analyzed based on the URL, execute the web page in an emulated JavaScript environment, and return the web service response. When the determining indicates that the job is not a pending job, a data collector event handler indicated in that job is executed by passing that job as a parameter to the data collector event handler. The data collector event handler is configured to update an output resource based on content of the analyzed web page included in that job.

[0006] This technology provides a number of advantages including providing methods, non-transitory computer readable media, and web site analyzer computing devices that utilize web services and emulated JavaScript environments to more effectively analyze web pages of web sites. In particular, this technology extracts web page information, including information inside client-side JavaScript code, to facilitate a more thorough analysis of web sites. This technology also advantageously utilizes data collector event handlers that provide flexibility with respect to the type of provided output.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007]

FIG. 1 is a block diagram of an environment with an exemplary web site analyzer computing device;

FIG. 2 is a flow chart of an exemplary method for analyzing web sites using web services and the exemplary web site analyzer computing device of FIG. 1;

FIG. 3A is an exemplary stack including a portion of a pending job generated based on a startup configuration file;

FIG. 3B is a portion of an exemplary web service response generated by the web service indicated in the exemplary pending job in the exemplary stack of FIG. 3A and using the URL of the exemplary pending job in the exemplary stack of FIG. 3A;

FIG. 3C is an exemplary stack including a portion of each of a plurality of exemplary jobs resulting from processing of the exemplary web service response of FIG. 3B;

FIG. 4 is a plurality of exemplary data collector event handlers that process a subset of the plurality of exemplary jobs included in the exemplary stacks of FIGS. 3C, 6B, and 6B;

FIG. 5 is an exemplary output resource generated by the exemplary method for analyzing web sites using web services of FIG. 2.

FIG. 6A is a portion of an exemplary web service response generated by the web service indicated in a pending one of the exemplary jobs in the exemplary stack of FIG. 3C;

FIG. 6B is an exemplary stack including a portion of each of a plurality of a plurality of exemplary jobs resulting from processing the web service response of FIG. 6A;

FIG. 7A is a portion of an exemplary web service response generated by the web service indicated in a pending one of the exemplary jobs in the exemplary stack of FIG. 6B; and

FIG. 7B is an exemplary stack including a portion of each of a plurality of a plurality of exemplary jobs resulting from processing the web service response of FIG. 7A.

DETAILED DESCRIPTION

[0008] An exemplary environment 10 with a web site analyzer computing device 12 coupled to client devices 14(1)-14(n) via communication network(s) 16 and web server devices 18(1)-18(n) via communication networks

20 is illustrated in FIG. 1. Other numbers and types of systems, devices, and/or elements in other configurations and environments with other communication network topologies can also be used. This technology provides a number of advantages including providing methods, non-transitory computer readable media, and web site analyzer computing devices that more effectively analyze web sites using web services that execute web pages of the web sites in an emulated JavaScript environment in order to extract information from client-side JavaScript code of the web pages.

[0009] The web site analyzer computing device 12 in this particular example includes a processor 22, a memory 26, and a communication interface 28 which are coupled together by a bus 30 or other communication link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor 22 in the web site analyzer computing device 12 executes a program of stored instructions for one or more aspects of this technology as described and illustrated by way of the examples herein, although the processor 22 could execute other numbers and types of programmed instructions.

[0010] The memory 24 in the web site analyzer computing device 12 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM), a read only memory (ROM), solid state drives, flash, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 22, can be used for the memory 24 in the web site analyzer computing device 12.

[0011] In this particular example, the memory 24 includes a JavaScript environment emulator 32, a web service engine 34, a data collector engine 36, data collector event handlers 38, and a startup configuration file 40, although other types or numbers of modules or applications can be included in the memory 24 in other examples. By way of example only, the JavaScript environment emulator 32, web service engine 34, and/or data collector engine 36, can be implemented as executable modules of programmed instructions and/or configurable hardware logic for one or more of aspects of the technology described and illustrated herein, which are stored in the memory 24 and executed by the processor 22 in the web site analyzer computing device 12.

[0012] The JavaScript environment emulator 32 in this example can include programmed instructions and/or hardware logic configured to simulate a JavaScript environment for executing JavaScript code that may be included in web pages, as described and illustrated in more detail below. By way of example only, a JavaScript environment emulator 20 is illustrated and described in U.S. Patent Application Serial No. 12/802,670 entitled, "Meth-

ods For Utilizing A JavaScript Emulator In A Web content proxy Server And Devices Thereof," which is incorporated herein by reference in its entirety.

[0013] The web service engine 34 in this example can include programmed instructions and/or hardware logic configured to execute web services. Web services provide a standardized way of integrating web-based applications using eXtensible Markup Language (XML) and/or REpresentational State Transfer (REST) (e.g., using Java Universal Description Discovery and Integration (jUDDI) and/or a Swagger™ framework for a description standard) over an Internet Protocol (IP) backbone.

[0014] In this particular example, the web service engine 34 executes web services that execute web pages obtained from the server devices 18(1)-18(n) in an emulated JavaScript environment using the JavaScript environment emulator 32, extract information from the web pages, and return web service responses in a preconfigured format. By executing the web pages in an emulated JavaScript environment, the web services are able to extract information from client-side JavaScript code, as described and illustrated in more detail later. The returned web service responses can include pending jobs, as well as start and end jobs that indicate data collector event handlers 38 and include information regarding the analyzed web pages that the data collector event handlers 38 use to generate an output resource, as described and illustrated in more detail later.

[0015] The data collector engine 34 in this example can include programmed instructions and/or hardware logic configured to process web service responses including the jobs included therein. Accordingly, the data collector engine 34 collects and executes pending jobs and calls the data collector event handlers 38 for start and end jobs, as described and illustrated in more detail later.

[0016] The data collector event handlers 38 in this example are JavaScript functions that are called by the data collector engine 34 and take in start and end jobs as parameters. Based on the start and end jobs passed as parameters, the data collector event handlers 38 updates an output resource, also as described and illustrated in more detail later.

[0017] The startup configuration file 40 includes an indication of an initial web service to be called as well as a URL of an initial web page of a web site to be analyzed. Optionally, the startup configuration file 40 further includes a maximum number of web pages to process or a function callback configured to validate the output resource, for example, although the startup configuration file 40 can also include other information.

[0018] One or more of the JavaScript environment emulator 32, web service engine 34, and/or data collector engine 36, can also have other types and numbers of functions as described and illustrated herein. Additionally, one or more of the JavaScript environment emulator 32, web service engine 34, data collector engine 36, data collector event handlers 38, or startup configuration file

40 can be stored at and/or implemented by a separate device coupled to the web site analyzer computing device 12 by one or more of the communication network(s) 16 and 20, such as one or more of the web server devices 18(1)-18(n).

[0019] The communication interface 28 in the web site analyzer computing device 12 is used to operatively couple and communicate between the web site analyzer computing device 12, the client devices 14(1)-14(n) and the server devices 18(1)-18(n) via the communication network(s) 16 and 20, although other types and numbers of communication networks with other types and numbers of connections and configurations can be used. Additionally, one or more of the communication network(s) 16 and 20 can include one or more local area networks (LANs) and/or wide area networks (WANs). By way of example only, the communication network(s) 16 and 20 can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), secure HTTP (HTTPS), wireless application protocol (WAP), and/or SOAP, although other types and numbers of communication networks each having their own communications protocols, can be used.

[0020] The client devices 14(1)-14(n) in this example enable a user to request, receive, and interact with applications, web services, and content hosted by the server devices 18(1)-18(n) through the web site analyzer computing device 12 via one or more communication network(s) 16, although one or more of the client devices 14(1)-14(n) could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for the user. In some examples, the client devices 14(1)-14(n) comprise mobile computing devices with Internet access that enable one or more web services to be accessed. By way of example only, the client devices 14(1)-14(n) can be smart phones, personal digital assistants, or computers.

[0021] Each of the client devices 14(1)-14(n) includes one or more processors, a memory, a user input device, a display device, and a communication interface, which are coupled together by a bus or other communication link, although one or more of client devices 14(1)-14(n) can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor(s) in the client devices 14(1)-14(n) can execute a program of instructions stored in the memory of the client devices 14(1)-14(n) for one or more aspects of this technology as described and illustrated herein, although the processor(s) could execute other numbers and types of programmed instructions.

[0022] The user input device in the client devices 14(1)-14(n) can be used to input selections, such as a request for a particular web site, although the user input device could be used to input other types of requests and data and interact with other elements. The user input device can include keypads, touch screens, and/or vocal input processing systems although other types and numbers

of user input devices can be used.

[0023] The display device the client devices 14(1)-14(n) can be used to output data and information to the user, such as a requested web page by way of example only. The display device in the client devices 14(1)-14(n) can be a phone screen display, although other types and numbers of display devices could be used depending on the particular type of client device. The communication interface in the client devices 14(1)-14(n) can be used to operatively couple and communicate between the client devices 14(1)-14(n), the web site analyzer computing device 12, and the server devices 18(1)-18(n) over the communication networks 16 and 20.

[0024] The server devices 18(1)-18(n) provide content including web pages for use by one or more of the client devices 14(1)-14(n) or to be analyzed by the web site analyzer computing device 12, although the server devices 18(1)-18(n) can provide other numbers and types of functions. Each of the server devices 14(1)-14(n) in this example includes one or more processors, a memory, and a communication interface which are coupled together by a bus or other communication link, although each of the web server devices 18(1)-18(n) could have other numbers and types of components, parts, devices, systems, and elements in other configurations and locations.

[0025] The processor in each of the server devices 18(1)-18(n) executes a program of instructions stored in the memory of the server devices 18(1)-18(n) for one or more aspects of this technology, as described and illustrated by way of the examples herein, although the processor could execute other numbers and types of programmed instructions. The communication interface in each of the server devices 18(1)-18(n) is used to operatively couple and communicate between the server devices 18(1)-18(n), the web site analyzer computing device 12, and the client devices 14(1)-14(n) via communication networks 16 and 20.

[0026] Although the exemplary web site analyzer computing device 12, client devices 14(1)-14(n), and server devices 18(1)-18(n), are described and illustrated herein, each of the web site analyzer computing device 12, client devices 14(1)-14(n), and server devices 18(1)-18(n), can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

[0027] The examples of this technology described and illustrated herein may also be implemented on computer apparatuses or devices that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular commu-

nications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0028] The examples of this technology described and illustrated herein may also be embodied as one or more non-transitory computer readable media having instructions stored thereon for one or more aspects of this technology, as described and illustrated by way of the embodiments herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0029] An exemplary method for analyzing web sites using web services will now be described with reference to FIGS 1-7. Referring more specifically to FIG. 2, in step 200 in this particular example, the web site analyzer computing device 12 obtains the startup configuration file 40 from the memory 24 or from another location or network device. The startup configuration file includes at least a Uniform Resource Locator (URL) associated with a web site to be analyzed, such as a URL of a home web page for the web site, and an indication of a web service that is the initial web service that will initiate the analysis of the web site, as described and illustrated in more detail later. Optionally, the startup configuration file 40 can include other configuration information, such as a maximum number of web pages to process or a function callback configured to validate an output resource corresponding to a result of the analysis, for example, although other configuration information can also be included in the startup configuration file 40.

[0030] In step 202 in this example, the web site analyzer computing device 12 inserts a pending job including an indication of the web service and the URL included in the startup configuration file into a last-in-first-out (LIFO) data structure, which is referred to herein as a stack. The LIFO structure facilitates a depth-first inspection of the web pages of the web site, as described and illustrated in more detail later. In this particular example, a job is a structured record that can have a type indicating that the job is a pending job, a start job, or an end job.

[0031] A pending job in this example includes at least an indication of a web service, an indication of an endpoint of the web service, and a URL of a web page to be analyzed, and optionally also includes a name or a canonical URL associated with the URL of the web page to be analyzed. A start job in this example includes at least an indication of one of the data collector event handlers 38 and optionally also includes a name or the content of an analyzed web page obtained based on the execution of the web page in the emulated JavaScript environment, as described and illustrated in more detail later. In this example, an end job includes at least an indication of one of the data collector event handlers 38, and optionally also includes a name. Other types of information can also be included in one or more of the jobs and jobs having other types can also be used in other

examples.

[0032] Referring to FIG. 3A, an exemplary stack 300 including a portion of a pending job generated based on the startup configuration file 40 is illustrated. In this example, the web site analyzer computing device 12 generates the job 302 to have a pending type, an indication of the "root" web service included in the startup configuration file 40, and the "http://acme.com" URL included in the startup configuration file 40. Subsequent to generating the job 302, the web site analyzer computing device 12 inserts the job 302 into the stack 300.

[0033] Referring back to FIG. 2, in step 204 in this example, the web site analyzer computing device 12 determines whether the stack 300 is empty. In the first iteration, the stack 300 will never be empty. Accordingly, if the web site analyzer computing device 12 determines that the stack is not empty, then the No branch is taken to step 206. In step 206 in this example, the web site analyzer computing device 12 extracts a job from the stack 300, which is pending job 302 in the first iteration in this example.

[0034] In step 208 in this example, the web site analyzer computing device 12 determines whether the extracted job is a pending job based on a type identified in the job. In the first iteration, the job 302 generated based on the information contained in the startup configuration file 40 will always be of a pending type. Other nomenclature can also be used in other examples to indicate that a job includes an indication of a web service and a URL. Accordingly, if the web site analyzer computing device 12 determines that the job 302 is a pending job, then the Yes branch is taken to step 210.

[0035] In step 210 in this example, the web site analyzer computing device 12 optionally determines whether the extracted job 302 is a duplicate job. In the first iteration in this example, the job 302 will never be a duplicate job. However, in subsequent iterations, a canonical URL included in the job can be compared by the web site analyzer computing device 12 to a stored set of canonical URLs associated with previously analyzed web pages. The canonical URL can be included in the job by a web service that generate a web service response defining the job, as described and illustrated in more detail later. If the web site analyzer computing device 12 determines that the canonical URL included in the job matches one of the stored set of canonical URLs, then the web site analyzer computing device 12 will determine that the job is a duplicate job and take the Yes branch from step 210 back to step 204 without performing steps 212 and 214 for the job.

[0036] However, if the web site analyzer computing device 12 determines that the canonical URL included in the job does not match one of the stored set of canonical URLs, then the web site analyzer computing device 12 will determine that the job is not a duplicate job. If the web site analyzer computing device 12 determines that the job is not a duplicate job, then the canonical URL included in the job can be added to the stored set of

canonical URLs to facilitate subsequent identification of duplicate jobs. By configuring the web services to include canonical URLs in definitions of jobs in web service responses, and performing the comparison in step 210, the web site analyzer computing device 12 can avoid entering an infinite loop that could otherwise occur based on the topologies of some web site. Accordingly, if the web site analyze computing device 12 determines that the job 302 extracted in step 216 is not a duplicate job, then the No branch is taken from step 210 to step 212.

[0037] In step 212 in this example, the web site analyzer computing device 12 executes a web service indicated in the job 302, which is the "root" web service in job 302 in this example. The web service is configured to obtain the web page corresponding to the URL included in the job 302, execute the web page in an emulated JavaScript environment provided by the JavaScript Environment emulator 32, and return a web service response. By executing the web page in an emulated JavaScript environment, the web service is advantageously able to collect information from the web page that is only accessible by executing client-site JavaScript code. Accordingly, the web site analyzer computing device 12 calls the "root" web service in this example and passes the "http://acme.com" URL to the "root" web service as a parameter.

[0038] In step 214 in this example, the web site analyzer computing device 12 extracts one or more jobs from the web service response and insert the job(s) into the stack 300 between start and end jobs. Web service responses in this example include definitions of one or more job(s) and an indication of one of the data collector event handlers 38, and optionally also include a description the analyzed web page or a hash generated from the web page. The start and end jobs include an indication of one of the data collector event handlers 38 indicated in the web service response outside of the job definitions.

[0039] Referring to FIG. 3B, a portion of an exemplary web service response 304 generated by the "root" web service indicated in the pending job 302 in the stack 300 using the "http://acme.com" URL of the pending job 302 is illustrated. In this particular example, the "root" web service executes the web page corresponding to the "http://acme.com" URL and returns the web service response 304. The web service response identifies the "home" one of the data collector event handlers 38 (referred to in FIG. 3B as the "type"). The web service response 304 does not include a hash, but the optional hash can be used, in addition to or in place of the canonical URLs, to identify duplicated jobs based on web pages having corresponding content. The web service 304 also does not include any description of the web page (referring to in FIG. 3B as the "content"). However, the description can include content of the web page to be passed to one of the data collector event handlers 38, as described and illustrated in more detail later.

[0040] Additionally, the web service response 304 in this particular examples includes two job definitions, each

of which identifies the same one of the data collector event handlers 38 (referred to as the "action" in FIG. 3B), which is the "plp" data collector event handler. Each of the jobs are defined to have a pending type and include a name, URL, and canonical URL, although other information can also be included in the job definitions in other examples. Optionally, the web services can be configured to define pending jobs based on specified criteria (e.g., type of URL associated with an identified web page).

[0041] Referring to FIG. 3C, the stack 300 including a portion of each of the jobs resulting from processing of the exemplary web service response 304 of FIG. 3B is illustrated. In this particular example, the web site analyzer computing device 12 inserts a start job 306, pending jobs 308 and 310 extracted from the web service response 304, and an end job 312 into the stack 300. The pending jobs 308 and 310 include an indication of the "plp" one of the data collector event handlers 38 identified in the definition of each of the jobs 308 and 310 in the web service response 304. The web site analyzer computing device 12 inserts the pending jobs 308 and 310 into the stack between the start job 306 and the end job 312. Each of the start job 306 and the end job 312 includes an indication of the "home" one of the data collector event handlers 38 that is identified in the web service response 304.

[0042] Referring back to FIG. 2, subsequent to updating the stack 300 in step 214, the web site analyzer computing device 12 proceeds back to step 204 and again determines whether the stack 300 is empty. In this iteration in this example, the web site analyzer computing device 12 will again determine that the stack is not empty and will proceed to extract job 306 from the stack 300 in step 206. In step 208, the web site analyzer computing device 12 again determines whether the job is of a pending type. Since job 306 is a start job, the web site analyzer computing device 12 determines in this iteration that the extracted job is not a pending job and the No branch is taken to step 216.

[0043] In step 216 in this example, the web site analyzer computing device 12 executes one of the data collector event handlers 38 indicated in the job 306, which is the "home" one of the data collector event handlers 38 in this example. The data collector event handlers 38 are JavaScript functions that configured to update an output resource, although the data collector event handlers 38 can be written in other languages and can be configured to provide other functionality in other examples.

[0044] Referring to FIG. 4, a plurality of exemplary data collector event handlers 38(1)-38(6) that process jobs is illustrated. Accordingly, in this example, the job is a start job indicating the "home" data collector event handlers 38(1). Accordingly, the web site analyzer computing device 12 executes the data collector event handler 38(1) by passing the job 306 in this iteration. Upon execution, the data collector event handler 38(1) updates an output resource.

[0045] Referring to FIG. 5 an exemplary output resource 500 is illustrated. In this particular example the output resource is an eXtensible Markup Language (XML) document, but the output resource can be a JavaScript Object Notation (JSON) file, one or more records configured to be stored in a database, an e-mail or other electronic communication, or any other type of resource in other examples.

[0046] Accordingly, the data collector event handler 38(1) in this example is configured in this example to update the output resource 500 in this example to include a "<catalog>" start tag. In other examples, one or more of the data collector event handlers 38, such as data collector event handlers 38(3) and 38(5), for example, are configured to update the output resource based on content of the analyzed web page included in the job, as described and illustrated in more detail later.

[0047] However, in this example, subsequent to executing the data collector event handler 38(1) indicated in the extracted job 306 in step 216, the web site analyzer computing device 12 again proceeds back to step 204 and again determines whether the stack 300 is empty. Since the stack is not empty subsequent to the extraction of job 306, the web site analyzer computing device 12 will again take the no branch to step 206 and extract the job 308 from the stack 300. In this iteration, the extracted job 308 is a pending job and, accordingly, the web site analyzer computing device 12 will take the Yes branch from step 208 to step 210. Additionally, since the job 308 is not a duplicate job, the web site analyzer computing device 12 will take the No branch from step 210 to step 212.

[0048] In step 212 in this iteration, the web site analyzer computing device 12 will execute the "plp" web service indicated in the job 308 by passing the job 308 as a parameter. The "plp" web service in this iteration is configured to obtain the web page corresponding to the "http://acme.com/c2141" URL included in the job 308, executes the web page in an emulated JavaScript environment, and returns a web service response, such as the web service response 600 illustrated in FIG. 6A, for example.

[0049] In this example, the web service response 600 identifies a "plp" one of the data collector event handlers 38, a description of the web page, and four jobs including two start and two end jobs. Referring back to FIG. 2, in step 214 in this iteration, the web site analyzer computing device extracts the four jobs from the web service response 600 and inserts the four jobs into the stack 300 between start and end jobs that include an indication of the "plp" one of the data collector event handlers 38.

[0050] Referring to FIG. 6B, the stack 300 subsequent to step 214 in this iteration is illustrated. Accordingly, the four jobs defined in the web service response 600 are jobs 602, 604, 606, and 608, and jobs 602, 604, 606, and 608 are inserted into the stack 300 between start job 610 and end job 612. Accordingly, jobs 602-612 will be processed before steps 310 and 312 resulting in a depth-first

inspection of the web site due to the LIFO structure of the stack. Referring back to FIG. 2, subsequent to extracting the jobs 602, 604, 606, and 608 and inserting the jobs 602, 604, 606, 608, 610, and 612 into the stack 300, the web site analyzer computing device 12 proceeds back to step 204.

[0051] In this iteration, the web site analyzer computing device 12 will again determine that the stack 300 is not empty and take the No branch from step 204 to step 206. In step 206, the web site analyzer computing device 12 extracts job 610 from the stack. Job 610 is not a pending job and, accordingly, the website analyzer computing device 12 will take the No branch from step 208 to step 216. In step 216 in this iteration, the web site analyzer computing device 12 executes the data collector event handler 38(3) corresponding to the "plp" data collector event handler indicated in the job 610, as included based on the web service response 600, by passing the job 610 as a parameter.

[0052] As illustrated in FIG. 4, the data collector event handler 38(3) is configured to update the output resource to include the URL, name, and content from the job 610, as included in the definition of the job 610 in the web service response 600. Accordingly, the exemplary output resource 500 illustrated in FIG. 5 is updated to include the second through seventh lines of XML code, which correspond to the "http://acme.com/c2141/women_dresses" URL, "women dresses" name, and "Create the perfect holiday wardrobe with 20% off swim and beachwear" description or content, and associated tags and XML code.

[0053] Referring back to FIG. 2, subsequent to executing the data collector event handler 38(3) indicated in the job 610, the web site analyzer computing device 12 again proceeds to step 204 and determines whether the stack is empty. Accordingly, the web site analyzer computing device 12 repeats steps 204, 206, 208, and 216, as described and illustrated earlier, for jobs 602, 604, 606, 608, and 612. Subsequent to processing jobs 602, 604, 606, 608, and 612, in another iteration, the web site analyzer computing device 12 will again determine in step 204 that the stack 300 is not empty and take the No branch to step 206.

[0054] In step 206 in this iteration, the web site analyzer computing device 12 extracts job 310 and performs steps 208 and 210 for job 310. In step 212 in this iteration, the "plp" web service is executing by passing the "http://acme.com/c1550" URL indicated in the job 310. The "plp" web service returns the web service response 700, as illustrated in FIG. 7A in this example, which includes four jobs including two start and two end jobs. Accordingly, in step 214 in this iteration, the web site analyzer computing device 12 inserts the four jobs 702, 704, 706, and 708 between start job 710 and end job 712, resulting in the stack 300 illustrated in FIG. 7B.

[0055] Accordingly, referring back to FIG. 2, the web site analyzer computing device 12 then proceeds to perform steps 204, 206, 208, and 216 for jobs 710, 702, 704,

706, 708, 712, and 312. Subsequent to executing the data collector event handler 38(2) indicated in the job 312 in step 216, the web site analyzer computing device 12 proceeds back to step 204. Since there are no pending jobs in the stack 300 illustrated in FIG. 7B, the web site analyzer computing device 12 will not execute any web services while processing jobs 710, 702, 704, 706, 708, 712, and 312, and will not therefore insert any new jobs into the stack 300.

[0056] Accordingly, in the next iteration subsequent to processing job 312, the web site analyzer computing device 12 will determine in step 204 that the stack 300 is empty and the Yes branch will be taken to step 218. In step 218, the web site analyzer computing device 12 provides the output resource. The output resource 500 can be provided to a bot, a spider, or an indexer, for example, or any other type of application. Alternatively, as in the example describe and illustrated herein, the output resource 500 can be used to generate a mobile navigation menu, although different types of output resources can be generated and the output resource 500 can be used for different purposes or to facilitate different functionality in other examples.

[0057] Thus, as illustrated and described herein this technology provides a number of advantages including methods, non-transitory computer readable media, and web site analyzer computing devices that more effectively navigate web sites to collect information. With this technology, web services execute web pages in emulated JavaScript environments, which advantageously allows the web services to access and evaluate client-side JavaScript code and results in a more thorough and accurate inspection of the web pages.

[0058] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

Claims

1. A method for analyzing web sites using web services, the method comprising:

determining, by the web site analyzer computing device, when a job extracted from a stack is a pending job;

executing, by the web site analyzer computing device, a web service indicated in the job comprising passing a Uniform Resource Locator (URL) included in the job as a parameter to the web service, extracting another job from a web service response, and inserting the another job into the stack, when the determining indicates that the job is a pending job, wherein the web service is configured to obtain a web page to be analyzed based on the URL, execute the web page in an emulated JavaScript environment, and return the web service response; and executing, by the web site analyzer computing device, a data collector event handler indicated in the job comprising passing the job as a parameter to the data collector event handler, when the determining indicates that the job is not a pending job, wherein the data collector event handler is configured to update an output resource based on content of the analyzed web page included in the job.

- 2. The method as set forth in claim 1, further comprising, when the determining indicates that the job is a pending job:

determining, by the web site analyzer computing device, when the job is a duplicate job based on a canonical URL included in the job;
 storing, by the web site analyzer computing device, the canonical URL, when the determining indicates that the job is not a duplicate job; and
 extracting, by the web site analyzer computing device, an additional job from the stack, without executing the web service indicated in the job, extracting the another job from the web service response, or inserting the another job into the stack, when the determining indicates that the job is a duplicate job.

- 3. The method as set forth in claim 1, further comprising, when the determining indicates that the job is a pending job, inserting, by the web site analyzer computing device, an end job into the stack before the another job and inserting a start job into the stack after the another job, wherein the start and end jobs identify another data collector event handler indicated in the web service response.

- 4. The method as set forth in claim 1, wherein the data collector event handler is a JavaScript function, the web service response comprises one or more of the another job, an indication of another data collector event handler, a description the web page, or a hash generated from the web page, and the job comprises a structured record and is:

a pending job comprising one or more of the

indication of the web service, an indication of an endpoint of the web service, the URL of the web page to be analyzed, a name, or a canonical URL;
 a start job comprising one or more of an indication of the data collector event handler, another name, or the content of the analyzed web page obtained based on the execution of the web page in the emulated JavaScript environment;
 or
 an end job comprising one or more of the indication of the data collector event handler or the another name.

- 5. The method as set forth in claim 1, further comprising:

obtaining, by the web site analyzer computing device, a startup configuration file comprising one or more of an indication of the web service, the URL, a maximum number of web pages to process, or a function callback configured to validate the output resource;
 generating, by the web site analyzer computing device, the job to have a pending type, an indication of the web service, and the URL; and
 inserting, by the web site analyzer computing device, the job into the stack.

- 6. The method as set forth in claim 1, further comprising:

determining, by the web site analyzer computing device, when the stack is empty; and
 providing, by the web site analyzer computing device, the output resource when the determining indicates that the stack is empty, wherein the output resource comprises an eXtensible Markup Language (XML) document, a JavaScript Object Notation (JSON) file, one or more records configured to be stored in a database, or an e-mail or other electronic communication.

- 7. A non-transitory computer readable medium having stored thereon programmed instructions for analyzing web sites using web services comprising executable code that, when executed by at least one processor, causes the processor to perform steps comprising:

determining when a job extracted from a stack is a pending job;
 executing a web service indicated in the job comprising passing a Uniform Resource Locator (URL) included in the job as a parameter to the web service, extracting another job from a web service response, and inserting the another job into the stack, when the determining indicates

that the job is a pending job, wherein the web service is configured to obtain a web page to be analyzed based on the URL, execute the web page in an emulated JavaScript environment, and return the web service response; and
 5 executing a data collector event handler indicated in the job comprising passing the job as a parameter to the data collector event handler, when the determining indicates that the job is not a pending job, wherein the data collector
 10 event handler is configured to update an output resource based on content of the analyzed web page included in the job.

- 8. The non-transitory computer readable medium as set forth in claim 7, further having stored thereon one or more additional programmed instructions comprising executable code that, when executed by the processor further cause the processor to perform one or more additional steps comprising, when the determining indicates that the job is a pending job:

determining when the job is a duplicate job based on a canonical URL included in the job; storing the canonical URL, when the determining indicates that the job is not a duplicate job; and
 25 extracting an additional job from the stack, without executing the web service indicated in the job, extracting the another job from the web service response, or inserting the another job into the stack, when the determining indicates that the job is a duplicate job.

- 9. The non-transitory computer readable medium as set forth in claim 7, further having stored thereon one or more additional programmed instructions comprising executable code that, when executed by the processor further cause the processor to perform one or more additional steps comprising, when the determining indicates that the job is a pending job, inserting an end job into the stack before the another job and inserting a start job into the stack after the another job, wherein the start and end jobs identify another data collector event handler indicated in the web service response.

- 10. The non-transitory computer readable medium as set forth in claim 7, wherein the data collector event handler is a JavaScript function, the web service response comprises one or more of the another job, an indication of another data collector event handler, a description the web page, or a hash generated from the web page, and the job comprises a structured record and is:

a pending job comprising one or more of the indication of the web service, an indication of an

endpoint of the web service, the URL of the web page to be analyzed, a name, or a canonical URL;

a start job comprising one or more of an indication of the data collector event handler, another name, or the content of the analyzed web page obtained based on the execution of the web page in the emulated JavaScript environment; or

an end job comprising one or more of the indication of the data collector event handler or the another name.

- 11. The non-transitory computer readable medium as set forth in claim 7, further having stored thereon one or more additional programmed instructions comprising executable code that, when executed by the processor further cause the processor to perform one or more additional steps comprising:

obtaining a startup configuration file comprising one or more of an indication of the web service, the URL, a maximum number of web pages to process, or a function callback configured to validate the output resource;
 generating the job to have a pending type, an indication of the web service, and the URL; and inserting the job into the stack.

- 12. The non-transitory computer readable medium as set forth in claim 7, further having stored thereon one or more additional programmed instructions comprising executable code that, when executed by the processor further cause the processor to perform one or more additional steps comprising:

determining when the stack is empty; and providing the output resource when the determining indicates that the stack is empty, wherein the output resource comprises an eXtensible Markup Language (XML) document, a JavaScript Object Notation (JSON) file, one or more records configured to be stored in a database, or an e-mail or other electronic communication.

- 13. A web site analyzer computing device, comprising one or more processors coupled to a memory and configured to execute programmed instructions comprising and stored in the memory to:

determine when a job extracted from a stack is a pending job;
 execute a web service indicated in the job and pass a Uniform Resource Locator (URL) included in the job as a parameter to the web service, extract another job from a web service response, and insert the another job into the stack, when the determining indicates that the job is a pend-

ing job, wherein the web service is configured to obtain a web page to be analyzed based on the URL, execute the web page in an emulated JavaScript environment, and return the web service response; and
 5 execute a data collector event handler indicated in the job and pass the job as a parameter to the data collector event handler, when the determining indicates that the job is not a pending job, wherein the data collector event handler is configured to update an output resource based on content of the analyzed web page included in the job.

- 14. The web site analyzer computing device as set forth in claim 13, wherein the processor are further configured to execute one or more additional programmed instructions comprising and stored in the memory to, when the determining indicates that the job is a pending job:

determine when the job is a duplicate job based on a canonical URL included in the job;
 store the canonical URL, when the determining indicates that the job is not a duplicate job; and
 25 extract an additional job from the stack, without executing the web service indicated in the job, extracting the another job from the web service response, or inserting the another job into the stack, when the determining indicates that the job is a duplicate job.

- 15. The web site analyzer computing device as set forth in claim 13, wherein the processor are further configured to execute one or more additional programmed instructions comprising and stored in the memory to, when the determining indicates that the job is a pending job, insert an end job into the stack before the another job and insert a start job into the stack after the another job, wherein the start and end jobs identify another data collector event handler indicated in the web service response.

- 16. The web site analyzer computing device as set forth in claim 13, wherein the data collector event handler is a JavaScript function, the web service response comprises one or more of the another job, an indication of another data collector event handler, a description the web page, or a hash generated from the web page, and the job comprises a structured record and is:

a pending job comprising one or more of the indication of the web service, an indication of an endpoint of the web service, the URL of the web page to be analyzed, a name, or a canonical URL;
 55 a start job comprising one or more of an indica-

tion of the data collector event handler, another name, or the content of the analyzed web page obtained based on the execution of the web page in the emulated JavaScript environment;
 or
 an end job comprising one or more of the indication of the data collector event handler or the another name.

- 17. The web site analyzer computing device as set forth in claim 13, wherein the processor are further configured to execute one or more additional programmed instructions comprising and stored in the memory to:

obtain a startup configuration file comprising one or more of an indication of the web service, the URL, a maximum number of web pages to process, or a function callback configured to validate the output resource;
 generate the job to have a pending type, an indication of the web service, and the URL; and
 insert the job into the stack.

- 18. The web site analyzer computing device as set forth in claim 13, wherein the processor are further configured to execute one or more additional programmed instructions comprising and stored in the memory to:

determine when the stack is empty; and
 provide the output resource when the determining indicates that the stack is empty, wherein the output resource comprises an eXtensible Markup Language (XML) document, a JavaScript Object Notation (JSON) file, one or more records configured to be stored in a database, or an e-mail or other electronic communication.

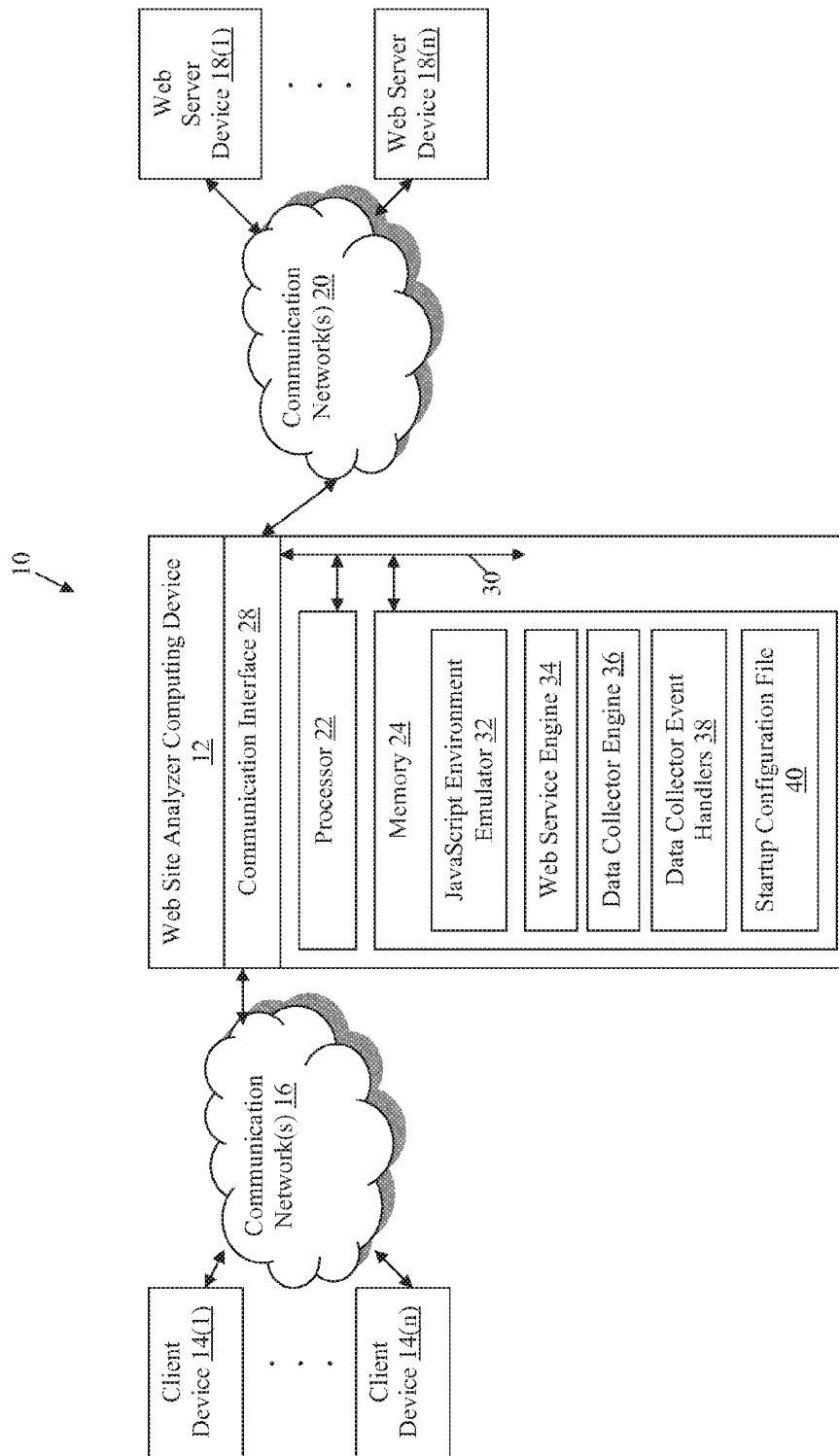


FIG. 1

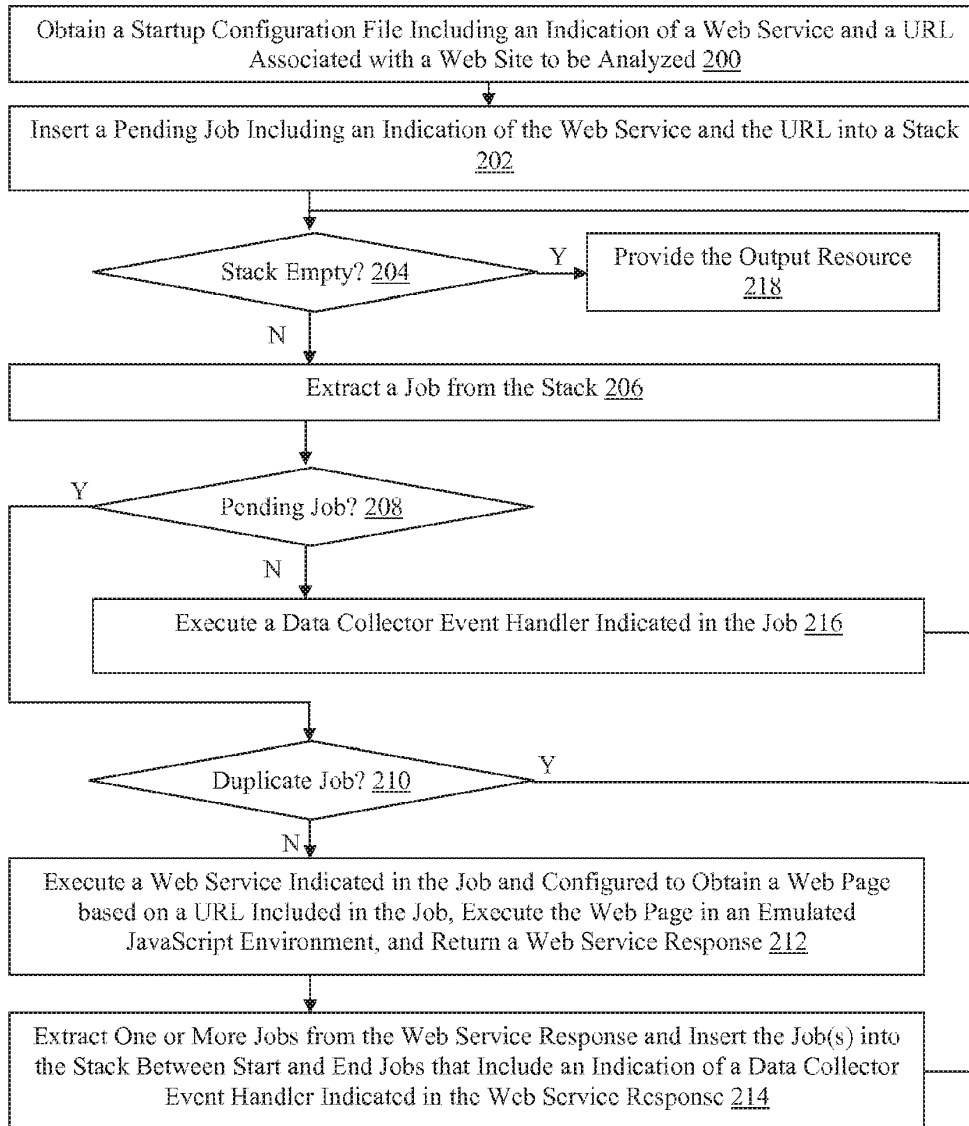


FIG. 2

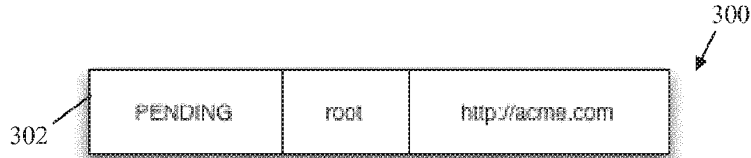


FIG. 3A

```

{
  "type": "home",
  "hash": "",
  "content": null,
  "jobs": [{
    "name": "women dresses",
    "type": "PENDING",
    "action": "plp",
    "url": "http://acme.com/c2141/women_dresses",
    "curl": "http://acme.com/c2141"
  }, {
    "name": "food & wine",
    "type": "PENDING",
    "action": "plp",
    "url": "http://acme.com/c1550/food_wine",
    "curl": "http://acme.com/c1550"
  }
]
}
    
```

FIG. 3B

306	START	home	
308	PENDING	plp	http://acme.com/c2141
310	PENDING	plp	http://acme.com/c1550
312	END	home	

FIG. 3C

```

function home_start(job) {
  out.write('<catalog>'); ← 38(1)
}

function home_end(job) {
  out.write('</catalog>'); ← 38(2)
}

function plp_start(job) {
  out.write('<plp url="' + job.url + '">');
  out.write('<name>' + job.name + '</name>');
  out.write('<description>' + job.content + '</description>');
}
                                     ← 38(3)

function plp_end(job) {
  out.write('</catalog>'); ← 38(4)
}

function pdp_start(job) {
  out.write('<product>');
  out.write('<name>' + job.name + '</name>');
  for (var i=0; i<job.content.sizes.length; i++)
    out.write('<size>' + job.content.sizes[i] + '</size>');
  if (job.content.bottles)
    out.write('<bottles>' + job.content.bottles + '</bottles>');
  out.write('<price>' + job.content.price + '</price>');
  out.write('</product>'); ← 38(5)
}

function pdp_end(job) { ← 38(6)
  // do nothing
}

```

FIG. 4

500

```
<catalog>
  <plp url="http://acme.com/c2141/women_dresses">
    <name>women dresses</name>
    <description>
      Create the perfect holiday wardrobe with 20% off
      swim and beachwear.
    </description>
    <product>
      <name>Long Sleeve Bow Dress</name>
      <size>small</size>
      <size>medium</size>
      <price>65$</price>
    </product>
    <product>
      <name>Bouclé Longline Coat</name>
      <size>small</size>
      <size>medium</size>
      <price>125$</price>
    </product>
  </plp>
  <plp url="http://acme.com/c1550/food_wine">
    <name>food & wine</name>
    <description>
      Save 25% when you raise a glass with this special
      discount on wine, champagne and beer.
    </description>
    <product>
      <name>Vinalta Malbec</name>
      <bottles>6</bottles>
      <price>48$</price>
    </product>
    <product>
      <name>Conte Priuli Prosecco</name>
      <bottles>4</bottles>
      <price>125$</price>
    </product>
  </plp>
</catalog>
```

Fig. 5

```

{
  "handler": "plp",
  "hash": "",
  "content": "Create the perfect holiday wardrobe with 20% off swim and beachwear.",
  "jobs": [
    {
      "name": "Long Sleeve Bow Dress",
      "type": "START",
      "handler": "pdp",
      "content": {
        "sizes": ["small", "medium"],
        "price": "65$"
      }
    },
    {
      "name": "Long Sleeve Bow Dress",
      "type": "END",
      "handler": "pdp"
    },
    {
      "name": "Bouclé Longline Coat",
      "type": "START",
      "handler": "pdp",
      "content": {
        "sizes": ["small", "medium"],
        "price": "125$"
      }
    },
    {
      "name": "Bouclé Longline Coat",
      "type": "END",
      "handler": "pdp"
    }
  ]
}

```

600

FIG. 6A

610	START	plp	
602	START	pdp	Long Sleeve Bow Dress
604	END	pdp	Long Sleeve Bow Dress
606	START	pdp	Bouclé Longline Coat
608	END	pdp	Bouclé Longline Coat
612	END	plp	
310	PENDING	plp	http://acme.com/c1550
312	END	home	

300

FIG. 6B

```

{
  "handler": "pip",
  "hash": "",
  "content": "Save 25% when you raise a glass with this special discount on wine, champagne and beer.",
  "jobs": [{
    "name": "Vinalta Malbec",
    "type": "START",
    "handler": "pdp",
    "content": {
      "price": "48$",
      "bottles": 6
    }
  }, {
    "name": "Vinalta Malbec",
    "type": "END",
    "handler": "pdp"
  }, {
    "name": "Conte Priuli Prosecco",
    "type": "START",
    "handler": "pdp",
    "content": {
      "price": "125$",
      "bottles": 4
    }
  }, {
    "name": "Conte Priuli Prosecco",
    "type": "END",
    "handler": "pdp"
  }
  ]
}

```

700

FIG. 7A

710	START	pip	
702	START	pdp	Vinalta Malbec
704	END	pdp	Vinalta Malbec
706	START	pdp	Conte Priuli Prosecco
708	END	pdp	Conte Priuli Prosecco
712	END	pip	
312	END	home	

700

FIG. 7B



EUROPEAN SEARCH REPORT

Application Number
EP 16 20 3840

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 7 519 902 B1 (KRAFT REINER [US] ET AL) 14 April 2009 (2009-04-14) * figures 3a, 4-8 * * column 7, line 50 - column 8, line 26 * * claim 1 * -----	1-18	INV. G06F17/30
			TECHNICAL FIELDS SEARCHED (IPC)
			G06F
The present search report has been drawn up for all claims			
Place of search Munich		Date of completion of the search 27 April 2017	Examiner Michalski, Stéphane
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1509 (03/02) (F04CO1)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 16 20 3840

5

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

27-04-2017

10

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7519902	B1	14-04-2009	NONE

15

20

25

30

35

40

45

50

55

EPO FORM P0489

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 12802670 B [0012]

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 3 195 136 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2016/032602 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2016/032602 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2016/032602 (art. 153(3) CBE).

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(10) International Publication Number
WO 2016/032602 A1

(43) International Publication Date
3 March 2016 (03.03.2016)

- (51) International Patent Classification:
G06F 15/16 (2006.01)
- (21) International Application Number:
PCT/US2015/036717
- (22) International Filing Date:
19 June 2015 (19.06.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
14/467,210 25 August 2014 (25.08.2014) US
- (71) Applicant: **USABLENET INC.** [US/US]; 142 W. 57th Street, 7th Floor, New York, NY 10019 (US).
- (72) Inventor: **SCODA, Enrico**; Via Cividina 416/3, I-33035 Martignacco Ud (IT).
- (74) Agents: **GALLO, Nicholas, J.** et al.; LeClairRyan, A Professional Corporation, 70 Linden Oaks, Suite 210, Rochester, NY 04625 (US).

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— with international search report (Art. 21(3))

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

(54) Title: METHODS FOR SYNCHRONIZING WEB SESSIONS AND DEVICES THEREOF

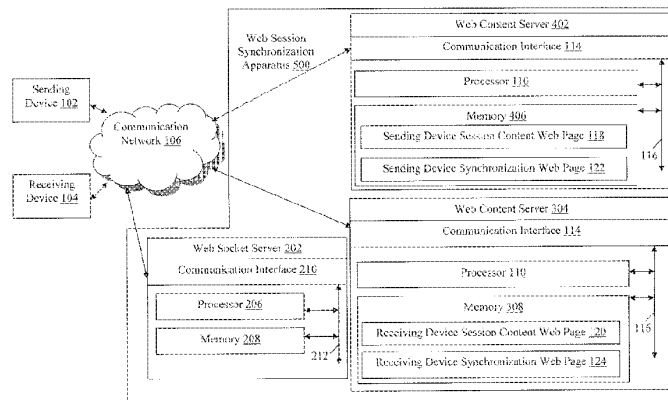


FIG. 5

(57) Abstract: A first web socket connection is established with a sending device and a second web socket connection is established with a receiving device. The sending device is notified when the second web socket connection is established. One or more cookies including session information and a redirect uniform resource locator (URL) are received from the sending device in response to the notification and over the first web socket connection. The one or more cookies and the redirect URL are forwarded to the receiving device over the second web socket connection, wherein the redirect URL is associated with a web page that, when executed by the receiving device, is configured to comprise the session information.

WO 2016/032602 A1

METHODS FOR SYNCHRONIZING WEB SESSIONS AND DEVICES THEREOF**FIELD**

[0001] This technology generally relates to methods and devices for synchronizing web pages and, more particularly, to maintaining session data in web sessions synchronized between
5 computing devices.

BACKGROUND

[0002] Many computing device users are increasingly using and switching between multiple computing devices such as mobile phones, tablets, and desktop computers. In order to
10 improve the user experience, many devices are able to maintain continuity with respect to a web browsing experience. Accordingly, functionality, such as HandoffTM provided in recent operating systems on computing devices made available by Apple Inc. of Cupertino, California, has been developed to allow users to load a web page on one device and continue browsing the same web page by selecting an icon on another device. For example, a user may want to view a
15 web page currently rendered on a mobile phone on a desktop computer instead, since the desktop computer may have a larger display. In another example, a user may want to switch from a tablet to a desktop computer to take advantage of the attached keyboard in order to input large amounts of data.

[0003] Currently, continuity is maintained by providing a receiving device with a
20 uniform resource locator (URL) of the web page that the user wants to load on the receiving device. However, user web sessions are not currently transferable as associated session information is not maintained. Accordingly, the user experience is reduced for certain web pages associated with user sessions and having session information. For example, a user's web browsing experience cannot be effectively continued on a receiving device for a web page
25 associated with a shopping cart and having session information such as the items the user has added to the shopping cart. Should the receiving device receive the URL for the shopping cart web page, the user session information will not be maintained and the user may have to repeat an operation, such as adding an item to the shopping cart, which is undesirable.

SUMMARY

30 [0004] A method for synchronizing web sessions includes receiving, with a web session synchronization apparatus, a request to establish a first web socket connection with a sending

- 2 -

device and establishing the first web socket connection in response to the request. A request to establish a second web socket connection is received, with the web session synchronization apparatus, from a receiving device and the second web socket connection is established with the receiving device in response to the request. The request to establish the second web socket
5 connection also includes the synchronization identifier. The sending device is notified when the second web socket connection is established. One or more cookies including session information are received, with the web session synchronization apparatus, as well as a redirect uniform resource locator (URL) from the sending device in response to the notification and over the first web socket connection. The one or more cookies and the redirect URL are forwarded,
10 with the web session synchronization apparatus, to the receiving device over the second web socket connection, wherein the redirect URL is associated with a web page that, when executed by the receiving device, is configured to comprise the session information.

[0005] A web session synchronization apparatus includes a processor coupled to a memory and configured to execute programmed instructions stored in the memory, including
15 receiving a request to establish a first web socket connection with a sending device and establishing the first web socket connection in response to the request. A request to establish a second web socket connection is received from a receiving device and the second web socket connection is established with the receiving device in response to the request. The request to establish the second web socket connection also includes the synchronization identifier. The
20 sending device is notified when the second web socket connection is established. One or more cookies including session information are received as well as a redirect uniform resource locator (URL) from the sending device in response to the notification and over the first web socket connection. The one or more cookies and the redirect URL are forwarded to the receiving device
25 over the second web socket connection, wherein the redirect URL is associated with a web page that, when executed by the receiving device, is configured to comprise the session information.

[0006] A non-transitory computer readable medium having stored thereon instructions for synchronizing web sessions comprising machine executable code which when executed by a processor, causes the processor to perform steps including receiving a request to establish a first web socket connection with a sending device and establishing the first web socket connection in
30 response to the request. A request to establish a second web socket connection is received from a receiving device and the second web socket connection is established with the receiving device in response to the request. The request to establish the second web socket connection also includes the synchronization identifier. The sending device is notified when the second web

- 3 -

socket connection is established. One or more cookies including session information are received as well as a redirect uniform resource locator (URL) from the sending device in response to the notification and over the first web socket connection. The one or more cookies and the redirect URL are forwarded to the receiving device over the second web socket
5 connection, wherein the redirect URL is associated with a web page that, when executed by the receiving device, is configured to comprise the session information.

[0007] This technology provides a number of advantages including methods, non-transitory computer readable media, and devices that facilitate synchronization of web pages with session information between computing devices. This technology allows users to maintain
10 continuity of a web browsing experience between computing devices. By maintaining session information, a user's web browsing experience across computing devices is improved since the user will not have to repeat operations performed within a web site and, instead, can continue a web browsing experience without loss of session information. Additionally, web socket connections are advantageously used to facilitate synchronization of a web session, without
15 requiring that session information be stored server-side, thereby providing increased security.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0008] FIG. 1 is a block diagram of a network environment with an exemplary web session synchronization apparatus with sending and receiving device session content and synchronization web pages and a web socket server;
- 20 [0009] FIG. 2 is a block diagram of a network environment with an exemplary web session synchronization apparatus with a web socket server and a web content server with sending and receiving device session content and synchronization web pages;
- [0010] FIG. 3 is a block diagram of a network environment with an exemplary web session synchronization apparatus with a web content server with a web socket server, sending
25 device session content web page, and sending device synchronization web page and another web content server with a receiving device session content web page and receiving device synchronization web page;
- [0011] FIG. 4 is a block diagram of a network environment with an exemplary web session synchronization apparatus with a web content server with a web socket server, receiving
30 device session content web page, and receiving device synchronization web page and another

web content server with a sending device session content web page and sending device synchronization web page;

[0012] FIG. 5 is a block diagram of a network environment with an exemplary web session synchronization apparatus with a web socket server, a web content server with a receiving device session content web page, and receiving device synchronization web page, and another web content server with a sending device session content web page and sending device synchronization web page;

[0013] FIG. 6 is a flow chart of an exemplary method for initiating transfer of a web session by a sending device to a receiving device;

10 [0014] FIG. 7 is an exemplary sending device session content web page;

[0015] FIG. 8 is an exemplary sending device synchronization web page;

[0016] FIG. 9 is a flow chart of an exemplary method of sending session information to a receiving device using a connection between a sending device and a web socket server;

[0017] FIG. 10 is an exemplary receiving device synchronization web page;

15 [0018] FIG. 11 is a flow chart of an exemplary method of receiving by a receiving device session information using a connection with a web socket server;

[0019] FIG. 12 is an exemplary receiving device session content web page; and

[0020] FIG. 13 is a flow chart of an exemplary method of facilitating synchronization of a web session between sending and receiving devices by a web socket server.

20

DETAILED DESCRIPTION

[0021] An exemplary network environment with a web session synchronization apparatus 100 coupled to a sending device 102 and a receiving device 104 is illustrated in FIG. 1. In this example, the web session synchronization apparatus 100, sending device 102 and receiving device 104 are coupled together by at least one communication network 106, although other numbers and types of systems, devices, and/or elements in other configurations or network topologies can also be used. This technology provides a number of advantages including methods, non-transitory computer readable media, and devices that securely facilitate

synchronization of web pages associated with web sessions between computing devices while maintaining session information and thereby improving the user experience.

[0022] The web session synchronization apparatus 100 in this example includes a web content server 108 coupled to the sending device 102 and receiving device 104 by the communication network 106, which can include one or more local area network(s) (LANs) and/or wide area network(s) (WANs). Other network devices configured to generate, send, and receive network communications and coupled together via other topologies can also be used. While not shown, the network environment also may include additional network components, such as routers, switches and other devices, which are well known to those of ordinary skill in the art and thus will not be described herein.

[0023] The web session synchronization apparatus 100 may perform any number of functions including hosting and providing web pages and facilitating synchronization of the web pages, including any associated web session information, between the sending device 102 and the receiving device 104 using web socket connections. In this example, the web content server 108 includes a processor 110, a memory 112, and a communication interface 114, which are coupled together by a bus 116 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used.

[0024] The processor 110 in the web content server 108 executes a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of the embodiments herein, although the processor 110 could execute other numbers and types of programmed instructions. The processor 110 of the web content server 108 may include one or more central processing units or general purpose processors with one or more processing cores, for example.

[0025] The memory 112 in the web content server 108 stores these programmed instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor 110, can be used for the memory 112 in the web content server 108. The memory 112 in this example stores a plurality of web pages including at least one sending device session content web page 118, receiving

- 6 -

device session content web page 120, sending device synchronization web page 122, and receiving device synchronization web page 124 as well as a web socket server 126.

[0026] Optionally, the sending device session content web page 118 and receiving device session content web page 120 can be the same web page or a different version of the same web page. In the first case, the web server can deliver both desktop and mobile pages, for example, using the same URL (e.g., using responsive web design or adaptive web design). In the second case, the web server hosts two web sites: one for desktop web page versions and one for mobile web page versions, which can be located at different URLs) For example, if the sending device 102 and receiving device 104 are both desktop computers, the sending device session content web page 118 and receiving device session content web page 120 can be the same web page, such as a desktop version of a shopping cart web page for a web site. In another example in which the sending device 102 is a mobile phone and the receiving device 104 is a desktop computer, the sending device session content web page 118 can be a mobile version of the shopping cart web page for the web site and the receiving device session content web page 120 can be a desktop version of the shopping cart web page for the web site.

[0027] Additionally, the sending device synchronization web page 122 and the receiving device synchronization web page 124 can be the same web page. For example, the web content server 108 can inject JavaScript code into a same synchronization web page to provide certain functionality of the sending device synchronization web page 122 or the receiving device synchronization web page 124 depending on whether a cookie is present in the HTTP request for the synchronization web page, as described and illustrated in more detail later. In another example, a same synchronization web page can be stored in the memory 112 as including the certain functionality of both the sending device synchronization web page 122 and the receiving device synchronization web page 124 with the synchronization web page determining the behavior based on the presence of a cookie, also as described and illustrated in more detail later. Other permutations of the web pages 118, 120, 122, and 124 with other functionality can also be used and other web pages can be provided in the memory 112.

[0028] The memory 112 of the web content server 108 also includes a web socket server 126 in this example, although one or more of the web pages 118, 120, 122, and 124 and/or the web socket server 126 can be provided elsewhere in the network environment, such as described and illustrated later with reference to FIGS. 2-5, for example. The web socket server 126 in this example is a software module that includes programmed instructions that, when executed by the processor 110, generate a web socket server configured to facilitate communications over web

socket connections between the sending device 102 and the receiving device 104 according to the web socket protocol, as described and illustrated in more detail later.

[0029] The communication interface 114 in the web content server 108 is used to operatively couple and communicate between the w web content server 108, the sending device
5 102, and the receiving device 104, which are all coupled together via the communication network 106, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. By way of example only, the communication network can use TCP/IP over Ethernet and industry-standard protocols, including hypertext transfer protocol (HTTP), and/or
10 secure HTTP (HTTPS), although other types and numbers of communication networks, such as a direct connection, modems and phone lines, e-mail, and wireless and hardwire communication technology, each having their own communications protocols, can be used.

[0030] The sending device 102 and the receiving device 104 in this example enable a user to request, receive, interact with, and synchronize applications and web pages hosted by the
15 web session synchronization apparatus 100 and using the communication network 106, although one or more of the sending device 102 or receiving device 104 could access content and utilize other types and numbers of applications from other sources and could provide a wide variety of other functions for a user. The sending device 102 and receiving device 104 can be the same type of computing device (e.g., mobile phone or desktop computer) or the sending device 102
20 and receiving device 104 can be different types of devices.

[0031] Each of the sending device 102 and receiving device 104 in this example includes a processor, a memory, an input device, a display device, and a communication interface, which are coupled together by a bus or other link, although one or more of sending device 102 or
remote device 104 can include other numbers and types of components, parts, devices, systems,
25 and elements in other configurations. The processor in each of the sending device and receiving device can execute a program of instructions stored in the memory for one or more aspects of the present invention, as described and illustrated herein, although the processor could execute other numbers and types of programmed instructions.

[0032] The input device in each of the sending device 102 and receiving device 104 can
30 be used to input selections, such as a request for a particular web page or other content stored by the web session synchronization apparatus, although the input device could be used to input other types of requests and data and interact with other elements. The input device can include

keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can also be used.

[0033] The display device in each of the sending device 102 and receiving device 104 can be used to show data and information to a user, such as web pages and other content
5 retrieved from the web session synchronization apparatus 100 by way of example only. The display device in one or more of the sending device 102 and receiving device 104 can be a television screen, a mobile phone screen display, a laptop screen, a tablet screen, or a monitor for example, although other types and numbers of displays could be used depending on the particular type of sending device 102 and receiving device 104. The communication interface in
10 each of the sending device 102 and receiving device 104 can be used to operatively couple and communicate between the sending device 102 and receiving device 104 and the web session synchronization apparatus 100 over the communication network 106.

[0034] Referring to FIG. 2 another exemplary network environment with an exemplary web session synchronization apparatus 200, the sending device 102, the receiving device 104,
15 and the communication network 106 is illustrated. In this example, the web session synchronization apparatus 200 includes a web socket server 202 and web content server 204 provided as separate devices. The web socket server 202 in this example includes a processor 206, a memory 208, and a communication interface 210 coupled together by a bus 212 or other link, although other numbers and types of components, parts, devices, systems, and elements in
20 other configurations and locations can be used. The web socket server 202 performs the same functions as described and illustrated earlier with reference to the web socket server 124 but is a hardware device separate from any web content server.

[0035] The processor 206 in the web socket server 202 executes a program of stored instructions one or more aspects of the present invention, as described and illustrated by way of
25 the embodiments herein, although the processor 206 could execute other numbers and types of programmed instructions. The processor 206 of the web socket server 202 may include one or more central processing units or general purpose processors with one or more processing cores, for example.

[0036] The memory 208 in the web socket server 202 stores these programmed
30 instructions for one or more aspects of the present invention, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM)

or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to the processor, can be used for the memory 208 in the web socket server 202.

5 [0037] The communication interface 210 in the web socket server 202 is used to operatively couple and communicate between the web socket server 202, the sending device 102, and the receiving device 104, which are all coupled together via the communication network 106, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used.

10 [0038] The web content server 204 in this example includes a processor 110, a memory 214, and a communication interface 114 coupled together by a bus 116 or other link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The web content server 204 may perform any number of functions including hosting and providing web pages in response to requests received from the
15 sending device 102 and receiving device 104. The web content server 204 is the same as the web content server 108 except that the memory 214 does not include web socket server 124 and instead only includes the sending device session content web page 116, receiving device session content web page 118, and sending device synchronization web page 120, and receiving device synchronization web page 122, although the sending device session content web page 116 and
20 receiving device session content web page 118 could be the same session content web page and the sending device synchronization web page 120 and receiving device synchronization web page 122 could also be the same synchronization device web page, as described and illustrated earlier.

[0039] Referring to FIG. 3 another exemplary network environment with an exemplary
25 web session synchronization apparatus 300, the sending device 102, the receiving device 104, and the communication network 106 is illustrated. In this example, the web session synchronization apparatus 300 includes a web content server 302 and another web content server 304. The web content servers 302 and 304 are the same as the web content servers 108 and 204 except that the memory 306 of the web content server 302 include the web socket server 124,
30 sending device session content web page 118, and sending device synchronization web page 122 and the memory 308 of the separate web content server 304 includes the receiving device session content web page 120 and the receiving device synchronization web page 124. Accordingly, in this example, the sending device and the receiving device could be different types of devices

(e.g., a mobile phone and a desktop computer) and the sending device session content web page 118 and receiving device session content web page 120 are different versions of the same web page but hosted on different content servers.

[0040] Referring to FIG. 4 another exemplary network environment with an exemplary web session synchronization apparatus 400, the sending device 102, the receiving device 104, and the communication network 106 is illustrated. In this example, the web session synchronization apparatus 400 includes a web content server 402 and another web content server 404. The web content servers 402 and 404 are the same as the web content servers 108, 204, 302, and 304 except that the memory 406 of the web content server 302 include only the sending device session content web page 118 and sending device synchronization web page 122 and the memory 408 of the separate web content server 404 includes the web socket server 124, the receiving device session content web page 120, and the receiving device synchronization web page 124. Accordingly, in this example as in the example described and illustrated earlier with reference to FIG. 3, the sending device and the receiving device could be different types of devices.

[0041] Referring to FIG. 5 another exemplary network environment with an exemplary web session synchronization apparatus 400, the sending device 102, the receiving device 104, and the communication network 106 is illustrated. In this example, the web session synchronization apparatus 400 includes the web content server 402, the web content sever 304, and the web socket server 202, all as separate devices. Accordingly, in this example as in the example described and illustrated earlier with reference to FIG. 3, the sending device and the receiving device could be different types of devices. In other examples, the web session synchronization apparatus 100, 200, 300, 400, and/or 500 can include different permutations of the web content server 108, 204, 302, 304, 402, or 404 and/or web socket server 124 or 202 and each of the web content servers 108, 204, 302, 304, 402, and 404 can store different web pages and content.

[0042] Although embodiments of the web session synchronization apparatus 100, 200, 300, 400, and 500, sending device 102, and receiving device 104 are described and illustrated herein, each of the web session synchronization apparatus 100, 200, 300, 400, and 500, sending device 102, and receiving device 104 can be implemented on any suitable computer apparatus or computing device. It is to be understood that the apparatuses and devices of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the embodiments are possible, as will be appreciated by those skilled

in the relevant art(s). Furthermore, each of the devices of the embodiments may be conveniently implemented using one or more general purpose computers, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and illustrated herein, and as will be appreciated by those ordinary skill in the art.

5 [0043] In addition, two or more computing apparatuses or devices can be substituted for any one of the devices in any embodiment described herein. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices of the embodiments. The embodiments may also be implemented on computer apparatuses or devices
10 that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet Data Networks (PDNs), the Internet,
15 intranets, and combinations thereof.

[0044] The embodiments may also be embodied as one or more non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present invention as described and illustrated by way of the embodiments herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to
20 implement the methods of the embodiments, as described and illustrated herein.

[0045] An exemplary method for synchronizing web sessions will now be described with reference to FIGS. 1-13. For purposes of this example only, the network environment described and illustrated with reference to FIG. 5 will be used, although any of the network environments described and illustrated with reference to FIGS. 1-4 can also be used in other examples.
25 Additionally, in this example, the sending device 102 is a mobile device and the receiving device 104 is a desktop computer, although, as described and illustrated earlier, the sending device 102 and receiving device 104 could also be any other types of devices as well as the same type of device.

[0046] Referring more specifically to FIG. 6, an exemplary method for initiating transfer
30 of a web session by a sending device to a receiving device is illustrated. In step 600, the sending device 102 obtains and executes, in a web browser for example, the sending device session content web page 118, which is associated with a web session. The sending device session

content web page 118 can be a mobile shopping cart web page associated with a web session for the user and having session content, such as an item added to the shopping cart, for example. Upon obtaining and executing the sending device session content web page 118, the web browser of the sending device 102 sets one or more cookies including information corresponding to the session content, such as an indication of the item added to the shopping cart in this example.

[0047] Referring more specifically to FIG. 7, an exemplary sending device session content web page 118 is illustrated. In this example, the session content includes at least the daily dental dog treat item added to the shopping cart for the web site, although the session content can include other information (e.g., number of items, number of each item, price, other user information) and the web session can be associated with types of web pages other than a shopping cart web page for a web site.

[0048] In this example, the sending device session content web page 118 corresponds with a Uniform Resource Locator (URL), such as "https://m.acme.com/cart", although any other type of web page associated with a web session and located at any other URL can also be used. Accordingly, the web content server 402 in this example can be hosting mobile versions of various web pages, including the sending device session content web page 118, that are associated with various mobile sites.

[0049] Referring back to FIG. 6, in step 602, the sending device 102 receives a request to synchronize the web session with the receiving device 104. In this example, the sending device session content web page 118 includes a transfer cart to desktop button 702. Accordingly, the sending device session content web page 118 can receive the request to initiate synchronization of the web session upon user interaction with the transfer cart to desktop button 702, although other types of user inputs can be used and the request to synchronize the web session can be generated in other ways.

[0050] In step 604, the sending device 102 generates a synchronization identifier and, optionally, a cookie with a value of the synchronization identifier. Accordingly, interaction by the user with the transfer cart to desktop button 702 of the sending device session content web page 118 in this example can cause the web browser of the sending device 102 to execute JavaScript code of the sending device session content web page 118 to generate the synchronization identifier (e.g., "3823329234") and, optionally, set the cookie having the value of the synchronization identifier. The cookie with the value of the synchronization identifier is

optional and not required in this example since the sending device synchronization web page 122 and receiving device synchronization web page 124 are different web pages provided by different web content servers 304 and 402. However, in examples in which the sending device synchronization web page 122 and receiving device synchronization web page 124 are the same
5 web page having both sending device and receiving device synchronization functionality, or having sending device or receiving device synchronization functionality injected by a web content server, the cookie with the value of the synchronization identifier is required, as described and illustrated in more detail later.

[0051] In step 606, the sending device 102 processes a redirect based on a URL of the
10 sending device synchronization web page 122. Accordingly, the JavaScript code executed by the web browser of the sending device 102, upon user interaction with the transfer cart to desktop button 702 in this example, can be configured to initiate the redirect based on a URL of the sending device synchronization web page 122 that includes the synchronization identifier. In this example, the URL of the sending device synchronization web page 122 can be
15 "https://m.acme.com/sync?id=3823329234", although any other URL that includes the synchronization identifier can also be used.

[0052] In other examples in which the cookie set in step 604 is required, the request for the sending device synchronization web page 122 sent to a web content server as part of processing the redirect includes the cookie with the synchronization identifier. Since, in these
20 examples, the sending device synchronization web page 122 and receiving device synchronization web page 124 may be identified based on the same URL, the web content server can determine which synchronization web page to provide in response to the request based on whether the cookie with the synchronization identifier is included in the request. If the cookie with the synchronization identifier is present, then the web content server will respond with the
25 sending device synchronization web page 122, or a synchronization web page injected with JavaScript code having the sending device synchronization functionality described and illustrated later with reference to FIG. 9. In other examples, the web content server will send the same synchronization web page in response to the request, but the JavaScript code will determine whether the sending or receiving device synchronization functionality is executed
30 based on the cookie, as described and illustrated in more detail later.

[0053] Referring more specifically to FIG. 8, an exemplary sending device synchronization web page 122 is illustrated. In this example, the sending device synchronization web page 122 includes text indicating that the process of synchronizing the web session has been

initiated, although the sending device synchronization web page 122 can include any other text or content. The sending device synchronization web page 122 is obtained and executed by the web browser of the sending device 102 in step 606 in this example as part of processing the redirect.

5 [0054] Referring more specifically to FIG. 9, an exemplary method for sending session information to the receiving device 104 using a connection between the sending device 102 and the web socket server 202 is illustrated. The method described and illustrated with reference to FIG. 9 can be performed by the sending device 102 executing JavaScript code of the sending device synchronization web page 122. In examples in which the sending device synchronization
10 web page 122 and receiving device synchronization web page 124 are the same synchronization web page, with both sending and receiving device synchronization functionality, the JavaScript code of the synchronization web page can be configured to determine whether the cookie with the synchronization identifier, as set in step 604 as described and illustrated earlier with reference to FIG. 6, was returned with the response that included the synchronization web page.
15 If the cookie with the synchronization identifier was included in the response that included the synchronization web page, then the JavaScript code of the synchronization web page implements the sending device synchronization functionality described and illustrated with reference to FIG. 9.

[0055] Accordingly, in step 900 in this example, the sending device 102 executing the
20 sending device synchronization web page 122 sends a request including the synchronization identifier to the web socket server 202 and establishes a web socket connection with the web socket server 202. The synchronization identifier can be obtained from the URL used to obtain the sending device synchronization web page 122 or from a cookie, such as the cookie optionally set as described and illustrated earlier with reference to step 604 of FIG. 6, for example. In other
25 examples, the web socket server 202 can be a module of a web content server, such as described and illustrated earlier with reference to web content servers 302 or 404, or of a single web content server, such as described and illustrated earlier with reference to web content servers 108 or 204.

[0056] In step 902, the sending device 102 executing the sending device synchronization
30 web page 122 determines whether a connection has been established with the receiving device 104. A connection is established with the receiving device 104 as described and illustrated in more detail later with reference to FIG. 11. However, upon the web socket server 202 establishing a connection with the receiving device 104, the web socket server 202 sends a

- 15 -

confirmation message to the sending device 102 using the connection established in step 900. Accordingly, in step 902, the sending device 102 determines whether the confirmation message confirming establishment of a connection between the web socket server 202 and the receiving device 104 has been received. If the sending device 102 determines that the connection with the receiving device 104 has not been established, then the No branch is taken back to step 902 and the sending device 102 effectively waits until the connection with the receiving device 104 is established.

[0057] However, if the sending device 102 determines in step 902 that the connection has been established with the receiving device 104, then the Yes branch is taken to step 904. In step 904, the sending device 102 executing the sending device synchronization web page 122 sends one or more cookies with session information and a redirect URL to the web socket server 202 over the connection established in step 900. The one or more cookies with the session information could have been set upon establishing the sending device session content web page 118, as described and illustrated in more detail earlier with reference to step 600 of FIG. 6. The redirect URL can be included in the JavaScript code of the sending device synchronization web page 122 and can be associated with the receiving device session content web page 120, which in this example is located at "https://www.acme.com/cart" since the receiving device 104 is a desktop computer. However, the sending device 102 may not be aware of the type of receiving device 104 and so the redirect URL can also be "https://m.acme.com/cart", which would cause the web browser of the receiving device 104 to redirect to "https://www.acme.com/cart" in this example.

[0058] In step 906, the sending device 102 executing the sending device synchronization web page 122 determines whether an end of communication message has been received from the web socket server 202 indicating that the one or more cookies with session information and redirect URL was successfully sent to the receiving device 104. If the sending device 102 determines that an end of communication message has not been received then the No branch is taken back to step 906 and the sending device 102 effectively waits for the end of communication message. In other examples, the sending device 102 can abort, resend the one or more cookies with session information and the redirect URL, or take another action.

[0059] However, if the sending device 102 determines in step 906 that an end of communication message has been received, then the Yes branch is taken to step 908. In step 908, the sending device 102 executing the sending device synchronization web page 122 generates a redirect to return to the sending device session content web page 118, which is

processed by the web browser of the sending device 102. Accordingly, the JavaScript code of the sending device synchronization web page 122 generates a redirect to the sending device session content web page 118 located at the "https://m.acme.com/cart" URL in this example, although other actions can also be performed by the sending device 102 in step 908.

- 5 Accordingly, subsequent to processing the redirect in step 908, the web browser of the sending device 102 returns to the web page rendered prior to and at the time of the user interaction with the transfer cart to desktop button 702 that initiated the synchronization in this example.

[0060] Referring more specifically to FIG. 10, an exemplary receiving device synchronization web page 124 is illustrated. Subsequent to the redirect being processed by the web browser of the sending device 102, as described and illustrated earlier with reference to step 10 606 of FIG. 6, the user of the sending device 102 initiates the obtaining and executing of the receiving device synchronization web page 124 by the receiving device 104. The user causes the web browser of the receiving device 104 to obtain and execute the receiving device synchronization web page 124 by manually entering the URL of the sending device 15 synchronization web page 122, which is rendered by the web browser of the sending device 102 after processing the redirect in step 606, into the web browser of the receiving device 104 or using a Handoff™ feature, for example, although other methods can also be used.

[0061] In this example, the web browser of the receiving device will attempt to obtain and execute a web page based on the "https://m.acme.com/sync?id=3823329234" URL of the 20 sending device synchronization web page 122. Since the receiving device 104 is a desktop computer in this example, the web content server 402 will receive the request, recognize the user-agent header of the request as being associated with a desktop web browser, and generate a redirect to the desktop version of the receiving device synchronization web page 124 hosted by the web content server 304 in this example. In this example, the receiving device 25 synchronization web page 124 includes text indicating that the process of synchronizing the web session has been initiated, although the receiving device synchronization web page 124 can include any other text or content.

[0062] Referring more specifically to FIG. 11, an exemplary method for receiving by the receiving device 104 session information using a connection with the web socket server 202 is 30 illustrated. The method described and illustrated with reference to FIG. 11 can be performed by the receiving device 104 executing JavaScript of the receiving device synchronization web page 124, for example. In examples in which the cookie set in step 604 is required, as described and illustrated earlier with reference to FIG. 6, the request for the receiving device synchronization

web page 124 sent to a web content server, as part of obtaining the receiving device synchronization web page 124 by the receiving device 104, will not include the cookie with the synchronization identifier. As described and illustrated earlier, if the cookie with the synchronization identifier is not present, then a web content server will respond with the receiving device synchronization web page 124, or a synchronization web page injected with JavaScript code having the receiving device synchronization functionality described and illustrated with reference to FIG. 11.

[0063] In examples in which the sending device synchronization web page 122 and receiving device synchronization web page 124 are the same synchronization web page with sending and receiving device synchronization functionality, the JavaScript code of the synchronization web page can be configured to determine whether the cookie with the synchronization identifier, set as described and illustrated earlier with reference to step 604 of FIG. 6, was returned with the response that included the synchronization web page. If the cookie with the synchronization identifier was not included with the response that included the synchronization web page, then the synchronization web page can be configured to implement the receiving device synchronization functionality described and illustrated with reference to FIG. 11.

[0064] Accordingly, in step 1100 in this example, the receiving device 104 executing the receiving device synchronization web page 124 sends a request including the synchronization identifier to the web socket server 202 and establishes a session with the web socket server 202. The receiving device 104 can obtain the synchronization identifier from the URL input to the web browser of the receiving device 104 in order to obtain the receiving device synchronization web page 124, for example, although the synchronization identifier can be obtained in other ways.

[0065] In step 1102, the receiving device 104 executing the receiving device synchronization web page 124 determines whether a connection has been established with the sending device 102. A connection is established with the sending device 104 as described and illustrated in detail earlier with reference to FIG. 9. Upon the web socket server 202 establishing a connection with the sending device 102, and subsequent to the receiving device 104 establishing the connection with the web socket server 202, the web socket server 202 sends a confirmation message to the receiving device 104 using the connection established in step 1100. Accordingly, in step 1102, the receiving device 104 determines whether the message confirming establishment of a connection between the web socket server 202 and the sending device 102 has

- 18 -

been received. If the receiving device 104 determines that the connection with the sending device 102 has not been established, then the No branch is taken back to step 1102 and the receiving device 104 effectively waits until the connection with the sending device 102 is established.

5 [0066] However, if the receiving device 104 determines in step 1102 that the connection has been established with the sending device 102, then the Yes branch is taken to step 1104. In step 1104, the receiving device 104 executing the receiving device synchronization web page 124 receives the one or more cookies with session information and the redirect URL from the web socket server 202 over the connection established in step 1100. The one or more cookies
10 with the session information and the redirect URL were sent by the sending device 102 to the web socket server 202 in this example as described and illustrated in more detail earlier with reference to step 904 of FIG. 9.

[0067] In step 1106, the receiving device 104 executing the receiving device synchronization web page 124, determines whether an end of communication message has been
15 received from the web socket server 202 indicating the end of information to be received from the web socket server 202, as sent to the web socket server 202 by the sending device 102. If the receiving device 104 determines that an end of communication message has not been received then the No branch is taken back to step 1106 and the receiving device 104 effectively waits for the end of communication message.

20 [0068] However, if the receiving device 104 determines in step 1106 that an end of communication message has been received, then the Yes branch is taken to step 1108. In step 1108, the receiving device 104 executing the receiving device synchronization web page 124 manages the cookie data, including associated session information, received in step 1104. In this
25 example, the web browser of the receiving device 104 sets one or more cookies including the session information according to the current domain and path.

[0069] In step 1110, the receiving device 104 executing the receiving device synchronization web page 124 generates a redirect based on the redirect URL received in step 1104. The generated redirect is processed by the web browser of the receiving device 104 which obtains and executes the receiving device session content web page 120 that is located at the
30 redirect URL.

[0070] Referring more specifically to FIG. 12, an exemplary receiving device session content web page 120 is illustrated. The receiving device session content web page 120 is hosted by the web content server 304 and located at “https://www.acme.com/cart” in this example, which is the redirect URL sent by the sending device 102, as described and illustrated earlier with reference to step 904 of FIG. 9, and received by the receiving device 104, as described and illustrated earlier with reference to step 1104 of FIG. 11.

[0071] Subsequent to obtaining the receiving device session content web page 120, as described and illustrated earlier with reference to step 1108 of FIG. 11, the receiving device 104 can execute the receiving device session content web page 120 by rendering the receiving content web page 120 including the session information included in the one or more cookies received by the receiving device 104 as described and illustrated earlier with reference to step 1104 of FIG. 11. Accordingly, in this example, the web browser of the receiving device 104 can render a desktop version of the sending device session content web page 118 that includes the web session information, thereby advantageously maintaining continuity of the user’s web browsing experience across devices.

[0072] Referring more specifically to FIG. 13, an exemplary method for facilitating synchronization of a web session between the sending device 102 and the receiving device 104 by the web socket server 202 of the web session synchronization apparatus 500 is illustrated. In step 1300, the web socket server 202 of the web session synchronization apparatus 500 receives a request to establish a connection from the sending device 102 and establishes a connection with the sending device 102 in response. The request is sent by the sending device 102 in this example as described and illustrated earlier with reference to step 900 of FIG. 9. Accordingly, the request includes the synchronization identifier which is optionally stored in a table in the memory 208 of the web socket server 202 allowing the web socket server 202 to identify active connections and associated synchronization identifiers.

[0073] In step 1302, the web socket server 202 receives a request to establish a connection from the receiving device 104 and establishes a connection with the receiving device 104 in response. The request is sent by the receiving device 104 in this example as described and illustrated earlier with reference to step 1100 of FIG. 11. Accordingly, the request includes the synchronization identifier.

[0074] In step 1304, the web socket server 202 of the web session synchronization apparatus 500 sends confirmation messages to the sending device 102 and the receiving device

104. Accordingly, the web socket server 202 can compare the synchronization identifier received from the receiving device 104 to synchronization identifiers of active connections and send a confirmation message confirming the establishment of a connection with the receiving device 104 to the sending device 102 associated with the active connection having a matching
5 associated synchronization identifier. Subsequent to sending the confirmation message to the sending device 102, the web socket server 202 can send a confirmation message confirming the establishment of a connection with the sending device 102 to the receiving device 104. The confirmation messages can be received by the sending device 102 as described and illustrated earlier with reference to step 902 of FIG. 9 and by the receiving device 104 as described and
10 illustrated earlier with reference to step 1102 of FIG. 11.

[0075] In step 1306, the web socket server 202 of the web session synchronization apparatus 500 receives one or more cookies with session information and a redirect URL from the sending device 102 over the established connection with the sending device 102 and forwards the one or more cookies and the redirect URL to the receiving device 104 over the
15 established connection with the receiving device 104. The one or more cookies with session information and the redirect URL are sent by the sending device 102 in this example as described and illustrated earlier with reference to step 904 of FIG. 9. Additionally, the one or more cookies and the redirect URL are received by the receiving device 104 in this example as described and illustrated earlier with reference to step 1104 of FIG. 11.

20 [0076] In step 1308, the web socket server 202 of the web session synchronization apparatus 500 sends an end of communication message to each of the sending device 102 and the receiving device 104. The end of communication messages are received by the sending device 102 in this example as described and illustrated earlier with reference to step 906 of FIG. 9 by the receiving device 104 in this example as described and illustrated earlier with reference to
25 step 1106 of FIG. 11. Optionally, the web socket server 202 closes the connections with each of the sending device 102 and the receiving device 104 following the sending of the end of communication messages.

[0077] With this technology, web sessions can be synchronized between devices such that web session information is maintained, thereby improving continuity of a user's web
30 browsing experience. Accordingly, users can switch between devices and maintain a web session and web browsing experience without having to repeat interactions with a web site associated with the web session. Additionally, web socket connections to a web socket server are advantageously utilized to transfer the web session information between a sending and a

receiving device. Accordingly, security is increased as the session information is not stored server-side, such as on a web content server.

[0078] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

- 22 -
CLAIMS

What is claimed is:

1. A method for synchronizing web sessions, the method comprising:
establishing, with a web session synchronization apparatus, in response to
5 a request a first web socket connection with a sending device;
establishing, with the web session synchronization apparatus, in response
to another request from a receiving device a second web socket connection;
notifying, with the web session synchronization apparatus, the sending
device when the second web socket connection is established;
10 receiving, with the web session synchronization apparatus, over the first
web socket connection one or more cookies comprising session information and a redirect
uniform resource locator (URL) from the sending device in response to the notifying; and
forwarding, with the web session synchronization apparatus, the one or
more cookies and the redirect URL to the receiving device over the second web socket
15 connection, wherein the redirect URL is associated with a web page that, when executed by the
receiving device, is configured to comprise the session information.
2. The method of claim 1, further comprising providing, with the web
session synchronization apparatus, a sending device session content web page to the sending
20 device, the session content web page configured to, when executed by the sending device,
generate and send a request for a sending device synchronization web page, the request for the
sending device synchronization web page including the synchronization identifier.
3. The method of claim 1, further comprising providing, with the web
25 session synchronization apparatus, a sending device synchronization web page to the sending
device, the sending device synchronization web page configured to, when executed by the
sending device, send the request to establish the first web socket connection and the one or more
cookies and redirect URL over the first web socket connection.
- 30 4. The method of claim 3, further comprising providing, with the web
session synchronization apparatus, a receiving device synchronization web page to the receiving
device, the receiving device synchronization web page configured to, when executed by the
receiving device, generate the request to establish the second web socket connection, receive the

one or more cookies and the redirect URL, and redirect a web browser of the receiving device based on the redirect URL.

5 5. The method of claim 4, wherein the sending device synchronization web
page and the receiving device web page are the same synchronization web page and the
synchronization web page is configured to, when executed by the sending device or the receiving
device, determine a behavior based on whether another cookie including the synchronization
identifier is included in a request for the synchronization web page or a response including the
synchronization web page.

10

6. The method of claim 1, wherein the receiving device session content web
page is a same version of the sending device session content web page or a different version of
the sending device session content web page adapted for a type of the receiving device and
served from a different location than sending device session content web page.

15

7. A web session synchronization apparatus, comprising:
a processor coupled to a memory and configured to execute programmed
instructions stored in the memory, comprising:

20 receiving a request to establish a first web socket connection with a
sending device, the request to establish the first web socket connection, and establishing the first
web socket connection in response to the request to establish the first web socket connection;

 receiving a request to establish a second web socket connection
from a receiving device, the request to establish the second web socket connection including the
synchronization identifier, establishing the second web socket connection in response to the
25 request to establish the second web socket connection, and notifying the sending device when the
second web socket connection is established;

 receiving one or more cookies including session information and a
redirect uniform resource locator (URL) from the sending device in response to the notification
and over the first web socket connection; and

30

 forwarding the one or more cookies and the redirect URL to the
receiving device over the second web socket connection, wherein the redirect URL is associated
with a web page that, when executed by the receiving device, is configured to comprise the
session information.

- 24 -

8. The apparatus of claim 7, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising providing a sending device session content web page to the sending device, the session content web page configured to, when executed by the sending device, generate and send a request for a sending device synchronization web page, the request for the sending device synchronization web page including the synchronization identifier.

9. The apparatus of claim 7, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising providing a sending device synchronization web page to the sending device, the sending device synchronization web page configured to, when executed by the sending device, send the request to establish the first web socket connection and the one or more cookies and redirect URL over the first web socket connection.

10. The apparatus of claim 9, wherein the processor is further configured to execute programmed instructions stored in the memory further comprising providing a receiving device synchronization web page to the receiving device, the receiving device synchronization web page configured to, when executed by the receiving device, generate the request to establish the second web socket connection, receive the one or more cookies and the redirect URL, and redirect a web browser of the receiving device based on the redirect URL.

11. The apparatus of claim 10, wherein the sending device synchronization web page and the receiving device web page are the same synchronization web page and the synchronization web page is configured to, when executed by the sending device or the receiving device, determine a behavior based on whether another cookie including the synchronization identifier is included in a request for the synchronization web page or a response including the synchronization web page.

12. The apparatus of claim 7, wherein the receiving device session content web page is a same version of the sending device session content web page or a different version of the sending device session content web page adapted for a type of the receiving device and served from a different location than sending device session content web page.

- 25 -

13. A non-transitory computer readable medium having stored thereon instructions for synchronizing web sessions comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:

- receiving a request to establish a first web socket connection with a
5 sending device, the request to establish the first web socket connection, and establishing the first web socket connection in response to the request to establish the first web socket connection;
receiving a request to establish a second web socket connection from a
receiving device, the request to establish the second web socket connection including the
synchronization identifier, establishing the second web socket connection in response to the
10 request to establish the second web socket connection, and notifying the sending device when the second web socket connection is established;
receiving one or more cookies including session information and a redirect
uniform resource locator (URL) from the sending device in response to the notification and over
the first web socket connection; and
15 forwarding the one or more cookies and the redirect URL to the receiving
device over the second web socket connection, wherein the redirect URL is associated with a
web page that, when executed by the receiving device, is configured to comprise the session
information.

- 20 14. The medium of claim 13, wherein the machine executable code when executed by the processor further causes the processor to perform steps further comprising
providing a sending device session content web page to the sending device, the session content
web page configured to, when executed by the sending device, generate and send a request for a
sending device synchronization web page, the request for the sending device synchronization
25 web page including the synchronization identifier.

15. The medium of claim 13, wherein the machine executable code when
executed by the processor further causes the processor to perform steps further comprising
providing a sending device synchronization web page to the sending device, the sending device
30 synchronization web page configured to, when executed by the sending device, send the request
to establish the first web socket connection and the one or more cookies and redirect URL over
the first web socket connection.

- 26 -

16. The medium of claim 15, wherein the machine executable code when executed by the processor further causes the processor to perform steps further comprising providing a receiving device synchronization web page to the receiving device, the receiving device synchronization web page configured to, when executed by the receiving device, generate
5 the request to establish the second web socket connection, receive the one or more cookies and the redirect URL, and redirect a web browser of the receiving device based on the redirect URL.

17. The medium of claim 16, wherein the sending device synchronization web page and the receiving device web page are the same synchronization web page and the
10 synchronization web page is configured to, when executed by the sending device or the receiving device, determine a behavior based on whether a cookie including the synchronization identifier is included in a request for the synchronization web page or a response including the synchronization web page.

15 18. The medium of claim 13, wherein the receiving device session content web page is a same version of the sending device session content web page or a different version of the sending device session content web page adapted for a type of the receiving device and served from a different location than sending device session content web page.

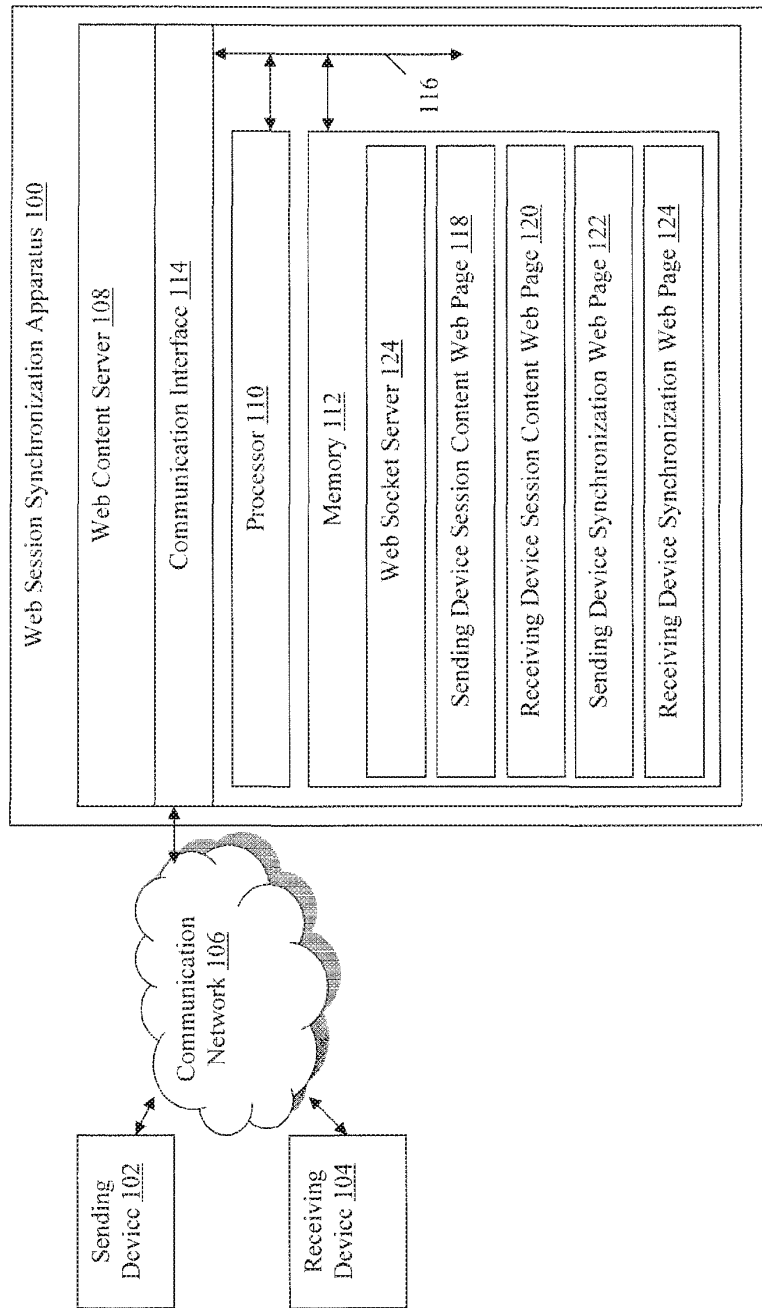


FIG. 1

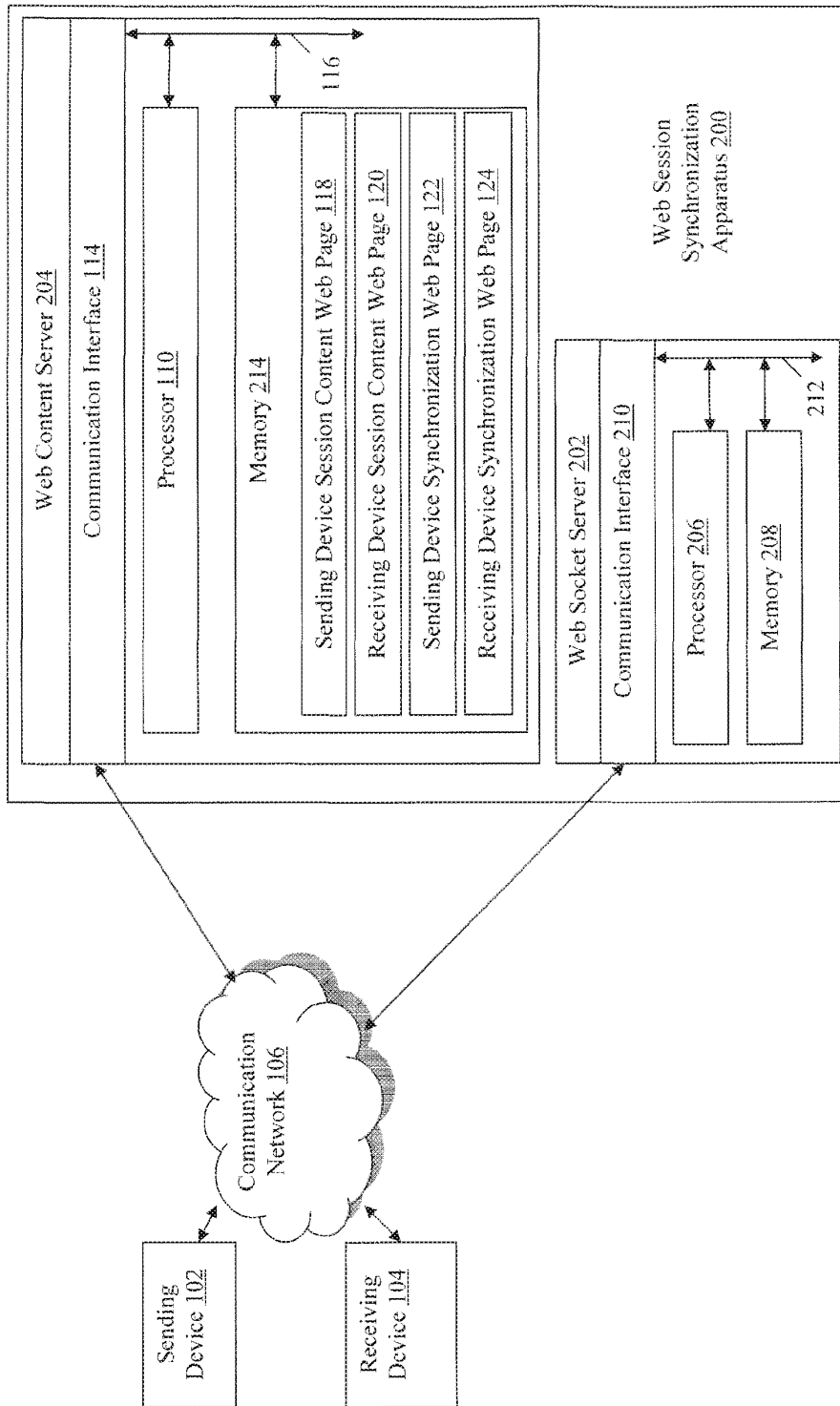


FIG. 2

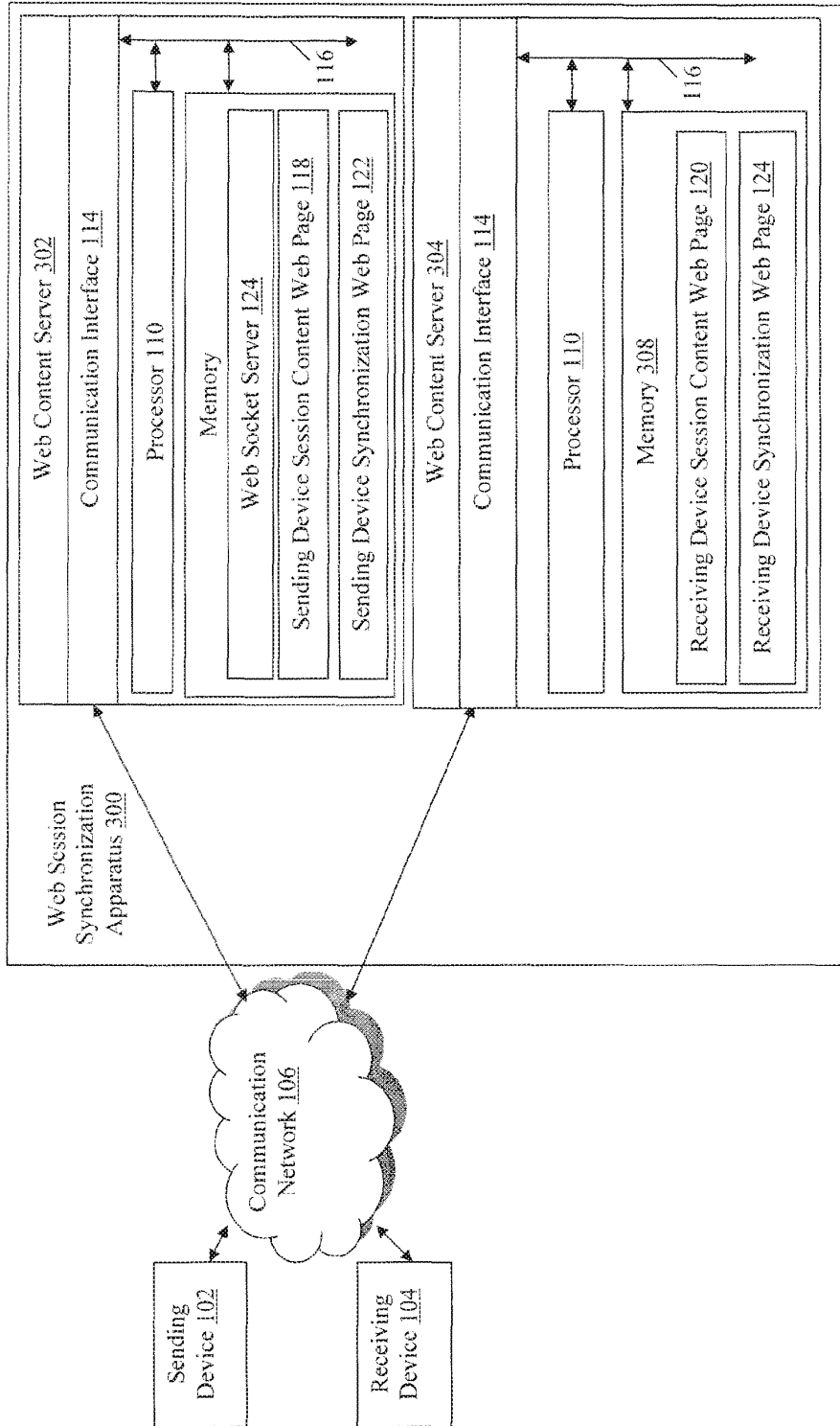


FIG. 3

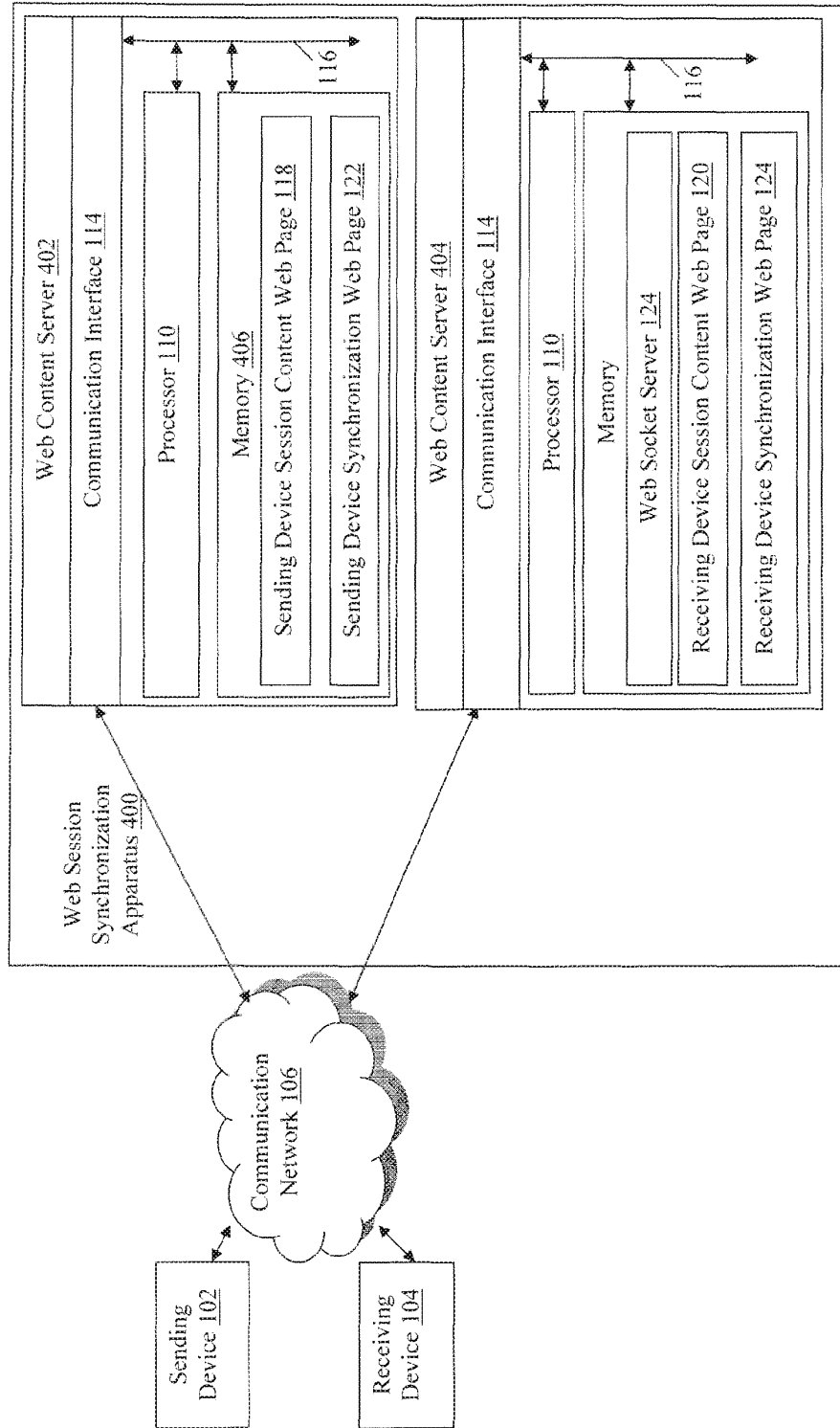


FIG. 4

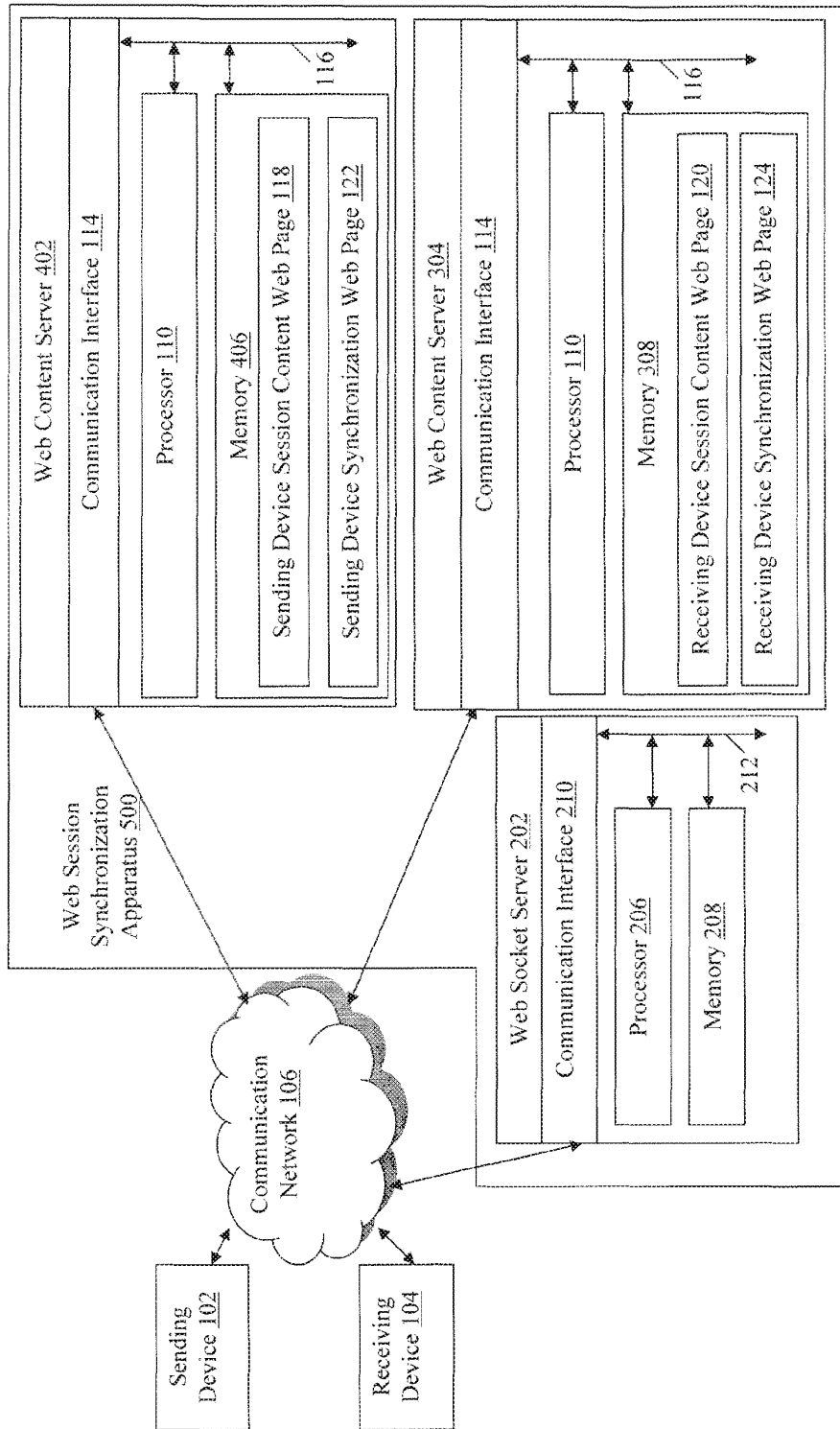


FIG. 5

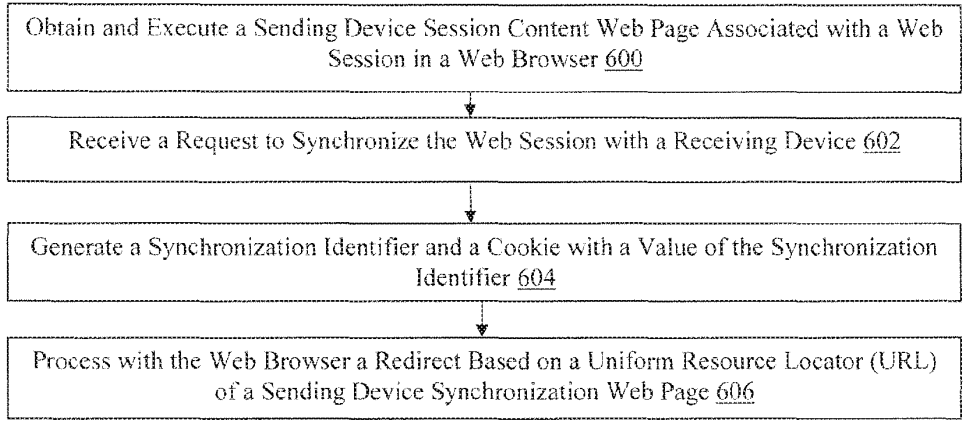


FIG. 6

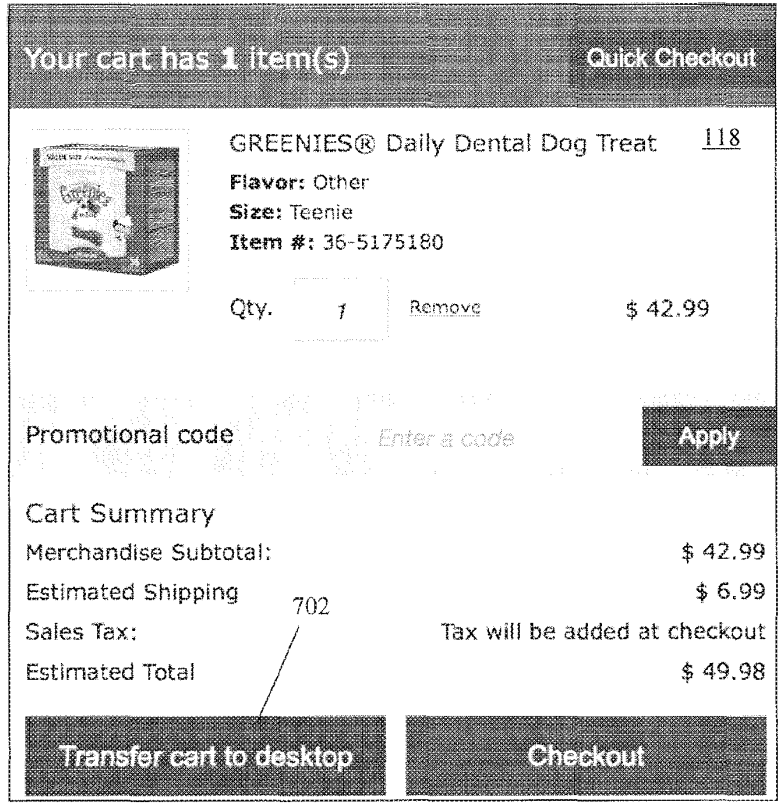


FIG. 7

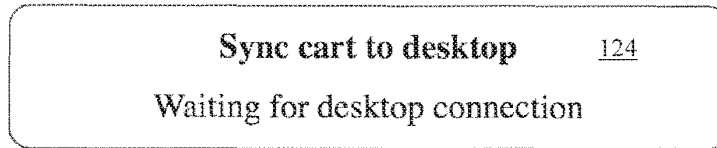


FIG. 8

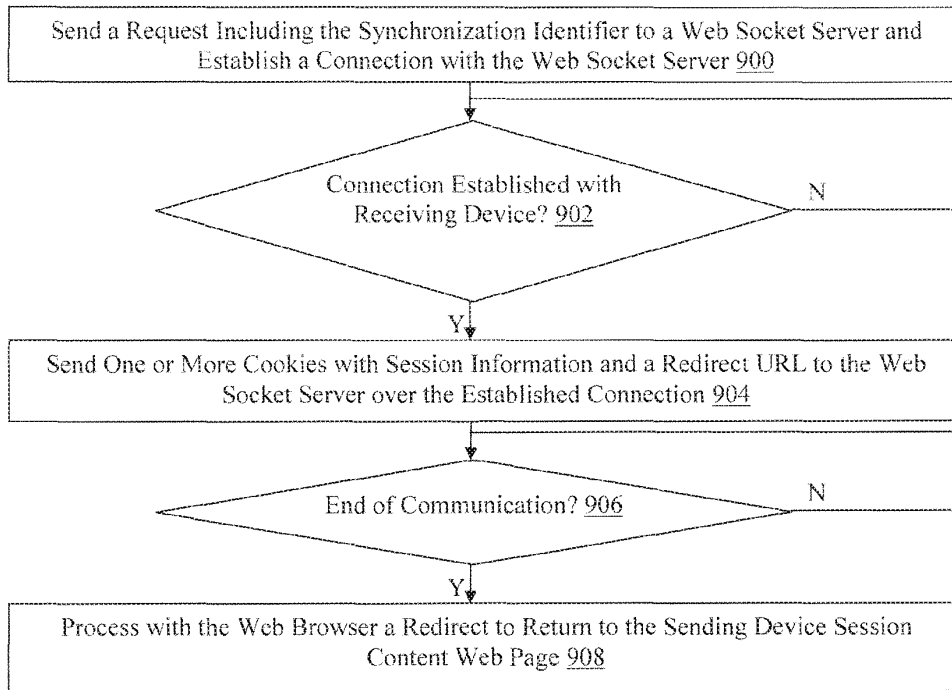


FIG. 9

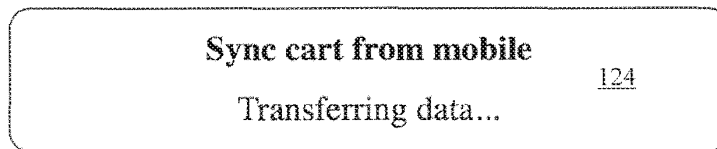


FIG. 10

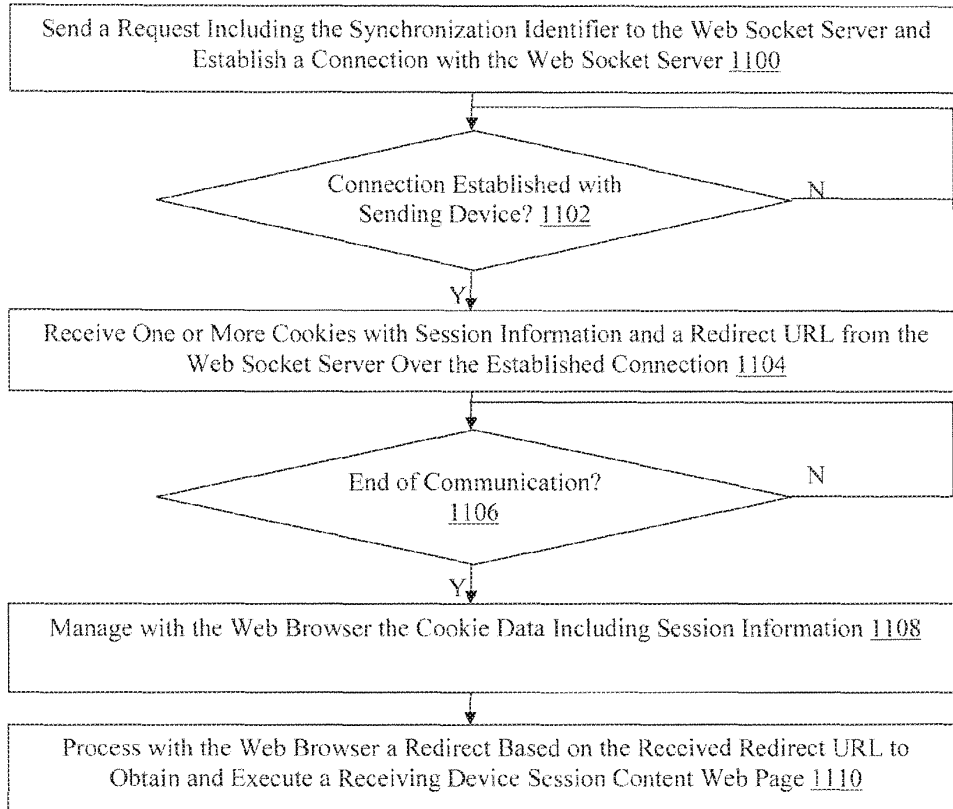


FIG. 11

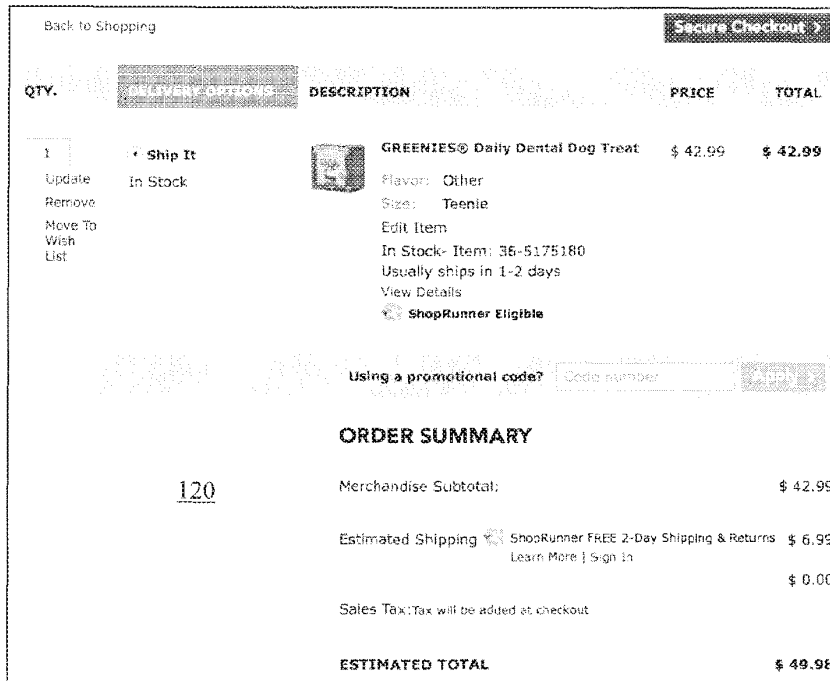


FIG. 12

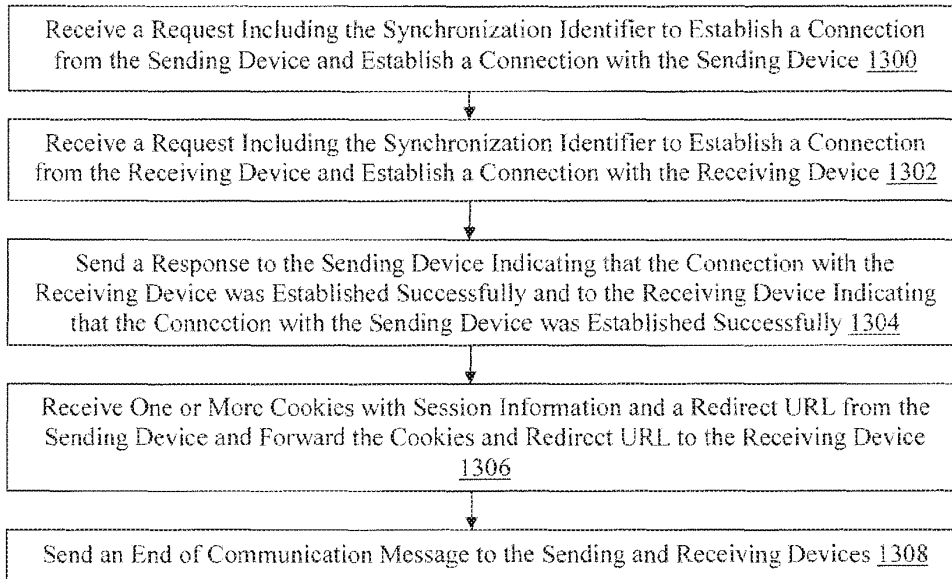


FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 15/36717

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 15/16 (2015.01) CPC - H04L 29/0854; H04L 29/08072; H04L 29/06 According to International Patent Classification (IPC) or to both national classification and IPC</p>																														
<p>B. FIELDS SEARCHED</p> <p>Minimum documentation searched (classification system followed by classification symbols) CPC: H04L 29/0854; H04L 29/08072; H04L 29/06; IPC(8): G06F 15/16 (2015.01)</p> <p>Documentation searched other than <i>minimum documentation</i> to the extent that such documents are included in the fields searched USPC: 709/248; 709/227; 709/217; CPC: H04L 29/0854; H04L 29/08072; H 04L 29/06; H04L 29/0809; G06F 1/14; IPC(8): G06F 15/16 (2015.01) (keyword limited, terms below)</p> <p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) PatBase, Google Patents, IEEE; Search Terms: synchronizing; web session; web socket connection; token; cookie; uniform resource locator, URL, uniform; resource identifier, URI; notifying, forwarding</p>																														
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1"> <thead> <tr> <th>Category*</th> <th>Citation of document, with indication, where appropriate, of the relevant passages</th> <th>Relevant to claim No</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>US 2014/0164447 A1 (Tarafdar et al.) 12 June 2014 (12.04.2014), entire document especially paras [0042], [0043], [0044], [0045], [0046], [0051], [0057], [0058], [0060], [0092]</td> <td>1 - 18</td> </tr> <tr> <td>A</td> <td>US 2009/0094688 A1 (Roy) 09 April 2009 (09.04.2009), entire document</td> <td>1 - 18</td> </tr> <tr> <td>A</td> <td>US 2010/0281107 A1 (Falows et al.) 04 November 2010 (04.11.2010), entire document</td> <td>1 - 18</td> </tr> <tr> <td>A</td> <td>US 2003/0131045 A1 (McGee et al.) 10 July 2003 (10.07.2003), entire document</td> <td>1 - 18</td> </tr> <tr> <td>A</td> <td>US 2009/0106349 A1 (Harris) 23 April 2009 (23.04.2009), entire document</td> <td>1 - 18</td> </tr> </tbody> </table> <p><input type="checkbox"/> Further documents are listed in the continuation of Box C <input type="checkbox"/></p> <p>* Special categories of cited documents:</p> <table border="0"> <tr> <td>"A" document defining the general state of the art which is not considered to be of particular relevance</td> <td>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</td> </tr> <tr> <td>"E" earlier application or patent but published on or after the international filing date</td> <td>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</td> </tr> <tr> <td>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</td> <td>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</td> </tr> <tr> <td>"O" document referring to an oral disclosure, use, exhibition or other means</td> <td>"&" document member of the same patent family</td> </tr> <tr> <td>"P" document published prior to the international filing date but later than the priority date claimed</td> <td></td> </tr> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No	X	US 2014/0164447 A1 (Tarafdar et al.) 12 June 2014 (12.04.2014), entire document especially paras [0042], [0043], [0044], [0045], [0046], [0051], [0057], [0058], [0060], [0092]	1 - 18	A	US 2009/0094688 A1 (Roy) 09 April 2009 (09.04.2009), entire document	1 - 18	A	US 2010/0281107 A1 (Falows et al.) 04 November 2010 (04.11.2010), entire document	1 - 18	A	US 2003/0131045 A1 (McGee et al.) 10 July 2003 (10.07.2003), entire document	1 - 18	A	US 2009/0106349 A1 (Harris) 23 April 2009 (23.04.2009), entire document	1 - 18	"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention	"E" earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone	"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art	"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family	"P" document published prior to the international filing date but later than the priority date claimed	
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No																												
X	US 2014/0164447 A1 (Tarafdar et al.) 12 June 2014 (12.04.2014), entire document especially paras [0042], [0043], [0044], [0045], [0046], [0051], [0057], [0058], [0060], [0092]	1 - 18																												
A	US 2009/0094688 A1 (Roy) 09 April 2009 (09.04.2009), entire document	1 - 18																												
A	US 2010/0281107 A1 (Falows et al.) 04 November 2010 (04.11.2010), entire document	1 - 18																												
A	US 2003/0131045 A1 (McGee et al.) 10 July 2003 (10.07.2003), entire document	1 - 18																												
A	US 2009/0106349 A1 (Harris) 23 April 2009 (23.04.2009), entire document	1 - 18																												
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention																													
"E" earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone																													
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art																													
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family																													
"P" document published prior to the international filing date but later than the priority date claimed																														
<p>Date of the actual completion of the international search</p> <p>24 August 2015 (24.08.2015)</p>		<p>Date of mailing of the international search report</p> <p>21 SEP 2015</p>																												
<p>Name and mailing address of the ISA/US</p> <p>Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300</p>		<p>Authorized officer:</p> <p>Lee W. Young</p> <p>PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774</p>																												

Form PCT/ISA/210 (second sheet) (January 2015)

(19)



(11) Veröffentlichungsnummer:

(11) Publication number: **EP 3 204 874 A0**

(11) Numéro de publication:

Internationale Anmeldung veröffentlicht durch die
Weltorganisation für geistiges Eigentum unter der Nummer:

WO 2016/057092 (Art. 153(3) EPÜ).

International application published by the World
Intellectual Property Organization under number:

WO 2016/057092 (Art. 153(3) EPC).

Demande internationale publiée par l'Organisation
Mondiale de la Propriété Intellectuelle sous le numéro:

WO 2016/057092 (art. 153(3) CBE).



- (51) International Patent Classification:
G06F 17/30 (2006.01)
- (21) International Application Number:
PCT/US2015/036956
- (22) International Filing Date:
22 June 2015 (22.06.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
14/509,235 8 October 2014 (08.10.2014) US
- (71) Applicant: **USABLENET INC.** [US/US]; 142 W. 57th Street, 7th Floor, New York, NY 10019 (US).
- (72) Inventor: **SCODA, Enrico**; Via Cividina 416/3, Martignacco UD 33035 (IT).
- (74) Agents: **GALLO, Nicholas, J.** et al.; LeClairRyan, A Professional Corporation, 70 Linden Oaks, Suite 210, Rochester, NY 14625 (US).
- (81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: METHODS FOR FACILITATING REFERENCES IN A CHAT CONTEXT AND DEVICES THEREOF

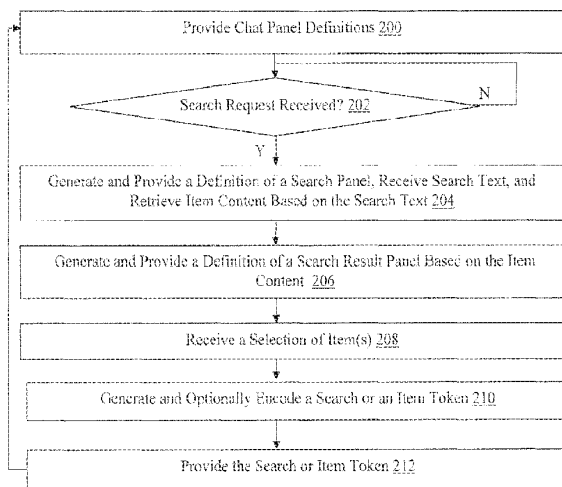


FIG. 2

(57) Abstract. A method, non-transitory computer readable medium, and chat management server apparatus that receives a search request via a search panel provided in response to a user interaction with a chat panel. A token including a special character is generated based on search text in the search request or a unique identifier for one of a plurality of items identified based on a search performed using the search text. The token is provided to a source of the search request for inclusion in the chat panel as a hyperlink. A preview panel request including the token is received in response to a user interaction with the hyperlink. Content for the items or for the one item is retrieved based on the special character included in the token. The content is provided to a source of the preview panel request.

WO 2016/057092 A1

METHODS FOR FACILITATING REFERENCES IN A CHAT CONTEXT AND DEVICES THEREOF**FIELD**

[0001] This technology generally relates to web-based chat contexts and, more particularly, to methods, non-transitory computer readable media, and apparatuses that facilitate the inclusion of references in chat panels.

BACKGROUND

[0002] Increasingly, web developers are providing chat functionality in websites via chat contexts that include chat panels that are displayed to a user as embedded within a web page or via a pop-up window, for example. A chat panel can allow interaction and communication between the user and a representative of the website host. Accordingly, such functionality is particularly useful for websites providing user support, although many other types of websites also implement chat contexts. In chat contexts, the speed of an exchange is often critical to an effective experience for users and, accordingly, it is preferable that messages are simple and short.

[0003] As one example, in a commercial website context, sales representatives may use chat panels to communicate with potential customers to answer questions regarding products or services in order to facilitate and increase sales. In this example, a sales representative may want to refer a prospective customer to content hosted on other portions of the website, such as product catalog content relating to products that might satisfy desired criteria communicated by the prospective customer.

[0004] In order to refer the prospective customer to the content, the sales representative may copy and paste Uniform Resource Locators (URLs) as hyperlinks. However, URLs are often very long and inconvenient for use in a chat context. While available services can process a URL and replace the URL with a relatively short link, the process is cumbersome and would still result in numerous hyperlinks and a relatively long message when the content is located at a number of URLs (e.g., corresponding to a number of different products). Additionally, upon selecting any of the hyperlinks, the prospective customer may be taken to a different web page in a new tab or window, which is inconvenient and does not allow the customer to preview products prior to navigating to a different web page associated with one of the products. Accordingly, there is currently no way for the representative of a website host to identify items

and provide preview content associated with the items to the prospective customer in an efficient and effective manner in a chat context.

SUMMARY

5 [0005] A method for facilitating references in a chat context includes receiving by a chat management server apparatus a search request via a search panel provided in response to a user interaction with a chat panel. A token including a special character is generated by the chat management server apparatus based on search text in the search request or a unique identifier for one of a plurality of items identified based on a search performed using the search text. The token is provided by the chat management server apparatus to a source of the search request for inclusion in the chat panel as a hyperlink. A preview panel request including the token is received by the chat management server apparatus in response to a user interaction with the hyperlink. Content for the items or for the one item is retrieved by the chat management server apparatus based on the special character included in the token. The content is provided by the chat management server apparatus to a source of the preview panel request.

15 [0006] A non-transitory computer readable medium having stored thereon instructions for facilitating references in a chat context comprising executable code which when executed by a processor, causes the processor to perform steps including receiving a search request via a search panel provided in response to a user interaction with a chat panel. A token including a special character is generated based on search text in the search request or a unique identifier for one of a plurality of items identified based on a search performed using the search text. The token is provided to a source of the search request for inclusion in the chat panel as a hyperlink. A preview panel request including the token is received in response to a user interaction with the hyperlink. Content for the items or for the one item is retrieved based on the special character included in the token. The content is provided to a source of the preview panel request.

25 [0007] A chat management server apparatus including a processor and a memory coupled to the processor which is configured to be capable of executing programmed instructions comprising and stored in the memory to receive a search request via a search panel provided in response to a user interaction with a chat panel. A token including a special character is generated based on search text in the search request or a unique identifier for one of a plurality of items identified based on a search performed using the search text. The token is provided to a source of the search request for inclusion in the chat panel as a hyperlink. A preview panel request including the token is received in response to a user interaction with the hyperlink.

Content for the items or for the one item is retrieved based on the special character included in the token. The content is provided to a source of the preview panel request.

- [0008] This technology provides a number of advantages including methods, non-transitory computer readable media, and apparatuses that more effectively facilitate references in a chat context. With this technology, website host representatives can identify content responsive to a search request from a user and communicate a reference to the content using a relatively short token. The token includes a special character and is introduced to a chat panel as a hyperlink. Upon selection of the hyperlink, the content is retrieved based on the token and included special character, and a preview panel is display to the user that includes the content.
- 5
- [0009] Accordingly, using the tokens, the size of the communicated reference(s) can be reduced, particularly when multiple URLs would otherwise have been required to communicate references to content associated with multiple items responsive to a search request. Additionally, the user does not have to navigate away from the current web page to see the content. Moreover, the tokens can be reused, advantageously allowing the host representative to respond relatively quickly to certain search requests matching previously searched criteria.
- 10
- 15

BRIEF DESCRIPTION OF THE DRAWINGS

- [0010] FIG. 1 is a block diagram of a network environment which incorporates an exemplary chat management server apparatus;
- [0011] FIG. 2 is a flowchart of an exemplary method of generating an item preview panel based on a token;
- 20
- [0012] FIG. 3 is an exemplary product web page with an exemplary chat panel link;
- [0013] FIG. 4 is an exemplary host chat panel with a search request button;
- [0014] FIG. 5 is an exemplary search panel for receiving search text;
- [0015] FIG. 6 is an exemplary search result panel displaying content for a plurality of selectable items identified based on search text;
- 25
- [0016] FIG. 7 is the exemplary chat panel of FIG. 4 with an item set link corresponding to a search token;

- 4 -

[0017] FIG. 8 is a flowchart of an exemplary method of generating a token for inclusion in a customer chat panel as a reference to item content;

[0018] FIG. 9 is an exemplary customer chat panel with the item set link and an exemplary multi-item preview panel with a navigational structure; and

5 [0019] FIG. 10 is the exemplary chat panel of FIG. 9 with an item link and an exemplary single item preview panel.

DETAILED DESCRIPTION

[0020] An exemplary network environment 10 is illustrated in FIG. 1 as including an exemplary chat management server apparatus 12. In this example, the chat management server apparatus 12 is coupled to a host representative device 14 by a local area network (LAN) 16 and a client device 18 by the LAN 16 and a wide area network (WAN) 20, although other types and numbers of devices, components, and elements in other topologies could be used. This technology provides a number of advantages including methods, non-transitory computer readable media, and apparatuses that more efficiently and effectively facilitate identifying and providing references to content matching user search criteria in a chat context.

[0021] In this example, the chat management server apparatus 12 includes a processor 22, a memory 24, and an interface device 26, which are coupled together by a bus 28 or other communication link, although other numbers and types of components, parts, devices, systems, and elements in other configurations and locations can be used. The processor 22 of the chat management server apparatus 12 may execute one or more stored programmed instructions for one or more aspects of this technology as described and illustrated by way of the embodiments herein, although the processor 22 could execute other numbers and types of programmed instructions.

[0022] The memory 24 of the chat management server apparatus 12 stores these programmed instructions for one or more aspects of this technology, as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. The memory 24 of the chat management server apparatus 12 may include one or more tangible storage media and/or devices, such as RAM, ROM, flash memory, hard disk drive(s), solid state memory, or any other memory storage types or devices, including combinations thereof, which are known to those of ordinary skill in the art.

[0023] In this example, the memory 24 of the chat management server apparatus 12 includes an item content catalog 30, a search web service 32, a preview web service 34, and an optional encoded token database 36, although the memory 24 can include other types and numbers of systems, devices, and elements in other configurations. Additionally, while the item content catalog 30, search web service 32, preview web service 34, and encoded token database 36 are illustrated in this example as being stored in the memory 24 of the chat management server apparatus 12, one or more of the item content catalog 30, search web service 32, preview web service 34, or encoded token database 36 could be stored elsewhere, including on another network device not shown in the network environment 10.

10 [0024] The item content catalog 30 in this example includes content for items, which can represent products or services, for example. The content can include a unique identifier for the item and information regarding the item including a description of the item, an item price, item options (e.g., colors), and/or any pictures or graphics associated with the item, for example, although any other type of content can also be stored in the item content catalog.

15 [0025] The search web service 32 in this example is configured to receive a request including search criteria and to identify matching item(s) in the item content catalog. The criteria can include search text, for example, when associated with a request to generate a search panel or a search token, for example, when associated with a request to generate a preview panel, as described and illustrated in more detail later. In response to the request, the search web service 32 retrieves at least a portion of the content (e.g., as used to present a preview of the item to a user) for each of the identified item(s) from the item content catalog 30 and generates and returns a HyperText Markup Language (HTML) fragment including the content, as described and illustrated in more detail later.

25 [0026] The preview web service 34 in this example is configured to receive a request to generate a preview panel that includes an item token. In response to the request, the preview web service 34 retrieves at least a portion of the content associated with an item identified based on the item token from the item content catalog 30 and generates and returns an HTML fragment including the content, as described and illustrated in more detail later.

30 [0027] The optional encoded token database 36 in this example stores encoded token values as associated with actual values. The actual values can be search text in the case of an encoded search token value or a unique item identifier in the case of an encoded item token value. As described and illustrated in more detail later, tokens can advantageously be encoded

with this technology to facilitate the inclusion of references in a chat panel that are even shorter than search text or item identifiers, for example.

[0028] The interface device 26 in the chat management server apparatus 12 is used to operatively couple and communicate between the chat management server apparatus 12 and the client device 18 via LAN 16 and WAN 20 and the host representative device 14 via the LAN 16, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements can also be used. The LAN 16 and WAN 20 can use TCP/IP over Ethernet and industry-standard protocols, including NFS, CIFS, SOAP, XML, LDAP, and SNMP, for example, although other types and numbers of communication networks can also be used.

[0029] The client device 18 in this example enables a user to request, receive and interact with services and content hosted by the chat management server apparatus 12 via the LAN 16 and WAN 20, although the client device 18 could access content and utilize other types and numbers of content or applications from other sources and could provide a wide variety of other functions for a user. By way of example only, the client device 16 can be a mobile computing device, smart phone, personal digital assistant, or computer, for example.

[0030] The client device 18 includes a processor 38, a memory 40, an interface device 42, an input device 43, and a display device 44 which are coupled together by a bus 45 or other communication link, although the client device 18 can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 38 in the client device 18 executes a program of stored instructions for one or more aspects of the present invention as described and illustrated herein, although the processor 38 could execute other numbers and types of programmed instructions.

[0031] The memory 40 in the client device 18 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM, ROM, hard disk drive(s), solid state storage device(s), and/or other storage device which is read from and/or written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 38 can be used for the memory 40 in the client device 18. In this example, the client device 18 is configured to access web services and web content through a web browser 46 stored in the memory 40. The web browser 46 in this example is configured to process programmed instructions (e.g., JavaScript

code) to render chat panels and preview panels, as well as provide other functionality, as described and illustrated in more detail later.

[0032] The interface device 42 in the client device 18 is used to operatively couple and communicate between the client device 18 and the chat management server apparatus 12 via the LAN 16 and the WAN 20, although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

[0033] The input device 43 in the client device 18 can be used to input selections, such as a request for a chat or preview panel, as well as messages to be exchanged with the user of the host representative device 14, although the input device 43 could be used to input other types of data and interact with other elements. The input device 43 can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can also be used.

[0034] The display device 44 in the client device 18 can be used to show data and information to a user, such as the requested chat or preview panel, although the display device 44 could be used to display other types of data and interact with other elements. The display device 44 can be television screen, a mobile phone screen display, a laptop screen, a tablet screen, or a monitor for example, although other types and numbers of displays could be used depending on the particular type of client device 18.

[0035] The host representative device 14 includes a processor 48, a memory 50, an interface device 52, an input device 53, and a display device 54, which are coupled together by a bus 55 or other communication link, although the host representative device 14 can include other numbers and types of components, parts, devices, systems, and elements in other configurations. The processor 48 in the host representative device 14 executes a program of stored instructions for one or more aspects of the present invention as described and illustrated herein, although the processor 48 could execute other numbers and types of programmed instructions.

[0036] The memory 50 in the host representative device 14 stores these programmed instructions for one or more aspects of the present invention as described and illustrated herein, although some or all of the programmed instructions could be stored and/or executed elsewhere. A variety of different types of memory storage devices, such as a RAM, ROM, hard disk drive(s), solid state storage device(s), and/or other storage device which is read from and/or

written to by a magnetic, optical, or other reading and/or writing system that is coupled to processor 38 can be used for the memory 40 in the host representative device 14.

[0037] In this example, the host representative device 14 is operated by a representative of a host of the website associated with the item content, although the host representative device
5 14 could be operated by other users, in order to engage in a chat with a user of the client device 18. Accordingly, the memory 50 in this example includes a web browser 56 through which the user of the host representative device can access web services and web content. The web browser 56 in this example is configured to process programmed instructions (e.g., JavaScript code) to render chat panels, search panels, and search result panels, as well as provide other
10 functionality, as described and illustrated in more detail later.

[0038] The interface device 52 in the host representative device 14 is used to operatively couple and communicate between the host representative device 14 and the chat management server apparatus 12 via the LAN 16, although other types and numbers of communication networks with other types and numbers of connections and configurations can be used.

15 [0039] The input device 53 in the host representative device 18 can be used to input selections, such as a request for a search panel, as well as messages to be exchanged with the user of the client device 18, although the input device 53 could be used to input other types of data and interact with other elements. The input device 53 can include keypads, touch screens, and/or vocal input processing systems, although other types and numbers of input devices can
20 also be used.

[0040] The display device 54 in the host representative device 18 can be used to show data and information to a user, such as the requested search panel, although the display device 54 could be used to display other types of data and interact with other elements. The display device 54 can be television screen, a mobile phone screen display, a laptop screen, a tablet screen, or a
25 monitor for example, although other types and numbers of displays could be used depending on the particular type of host representative device 18.

[0041] Although embodiments of the chat management server apparatus 12, host representative device 14, and client device 18 are described and illustrated herein, each of these devices can be implemented on any suitable computer system or computing device. It is to be
30 understood that the devices and systems of the embodiments described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the

embodiments are possible, as will be appreciated by those skilled in the relevant art(s).

Furthermore, each of the systems of the embodiments may be conveniently implemented using one or more general purpose computer systems, microprocessors, digital signal processors, and micro-controllers, programmed according to the teachings of the embodiments, as described and
5 illustrated herein, and as will be appreciated by those ordinary skill in the art.

[0042] In addition, two or more computing systems or devices can be substituted for any one of the devices in any embodiment. Accordingly, principles and advantages of distributed processing, such as redundancy and replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the embodiments. The
10 embodiments may also be implemented on computer system(s) that extend across any suitable network using any suitable interface mechanisms and communications technologies, including by way of example only telecommunications in any suitable form (e.g., voice and modem), wireless communications media, wireless communications networks, cellular communications networks, G3 communications networks, Public Switched Telephone Network (PSTNs), Packet
15 Data Networks (PDNs), the Internet, intranets, and combinations thereof.

[0043] The examples may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the
20 methods of the examples, as described and illustrated herein.

[0044] An exemplary method for facilitating references in a chat context will now be described with reference to FIGS. 1-10. Referring specifically to FIG. 2, an exemplary method of generating a token for inclusion in a chat panel as a reference to item content will now be described. In this example, in step 200, the chat management server apparatus 12 provides chat
25 panel definitions in response to a request received from the client device 18. Referring more specifically to FIG. 3, an exemplary product web page 300 with an exemplary chat panel link 302 is illustrated. In this example, a user of the client device 18 use the input/display device 43 to select the chat panel link 302 thereby initiating a chat session and requesting a customer chat panel from the chat management server apparatus 12, although the chat session can be initiated
30 in other manners.

[0045] In response, the chat management server apparatus 12 sends a definition of a customer chat panel to the client device 18 and a host chat panel to the host representative device

14. The definitions can define the chat panels using HTML, for example, which, when interpreted by the web browsers 46 and 56 cause the client device 18 and host representative device 14 to display the customer and host chat panels, respectively, in a pop-up window or an overlay, for example, although any other type of display for the chat panels can also be used.

5 The chat panels facilitate communication of messages between the users of the client device 18 and host representative device 14.

[0046] Referring more specifically to FIG. 4, an exemplary host chat panel 400 is illustrated. In this example, a definition of the host chat panel 400 is provided by the chat management server 12 to the host representative device 14 in response to receiving an initiation
10 of the chat session in step 200 from the client device 18. The definition sent by the chat management server 12 to the client device 18 in step 200 can define a customer chat panel similar to the host chat panel 400 but without the search request button 402, for example, although the chat panels can be similar or different in other ways and other methods of distributing and generating the chat panels can also be used. An exemplary customer chat panel
15 will be described and illustrated in more detail later with reference to FIGS. 9-10.

[0047] In this example, the user of the client device 18 is a prospective customer of the host of the product web page 300 that is looking for assistance completing an outfit prior to purchasing, although the web page 300 can be any other type of web page and the user of the client device 14 can be any other type of user. Accordingly, the customer user of the client
20 device 18 initiates the chat session in order to communicate with a representative of the host of the website, that is currently using the host representative device 14, to obtain the requested assistance in an efficient manner.

[0048] Referring back to FIG. 2, in step 202, the chat management server apparatus 12 determines when a search request has been received from the host representative device 14. The
25 search request can be for content that a user of the host representative device 14 would like to refer the customer to in order to attempt to assist the customer and facilitate a purchase, for example. If the chat management server apparatus 12 determines that a search request has not been received, then the No branch is taken back to step 202 and the chat management server apparatus 12 effectively waits for a search request to be received.

30 [0049] Referring back to FIG. 4, the search request button 402 of the host chat panel 400 can be used to initiate a search request, although any other type of interface for initiating a search request can also be used. In this example, the customer user of the client device 18 is looking for

white shoes to pair with a specified dress. Accordingly, the host representative using the input/display device 53 of the host representative device 14 begins to type a message in an input field 404, having the text “No problem, here is a set of options for you:” in this example. Next, the host representative using the input/display device 53 of the host representative device 14
5 selects the search request button 402 in order to initiate a search for items responsive to the customer’s request.

[0050] Referring back to step 202 of FIG. 2, upon selection of the search request button 402, or if the chat management server apparatus 12 otherwise determines that a search request has been received, then the Yes branch is taken to step 204. In step 204, the chat management
10 server apparatus 12 generates, and provides to the host representative device 14, a definition of a search panel, receives search text, and retrieves item content based on the search text. The definition can define a search panel using HTML, for example, which, when interpreted by the web browser 56 can cause the host representative device 14 to display a search panel in a pop-up window or an overlay, for example, although the search panel can be displayed in other manners.
15 The search panel is configured to receive search text, and optionally other search criteria, from the host representative using the host representative device 14.

[0051] Referring more specifically to FIG. 5, an exemplary search panel 500 for receiving search text via a text input box 502 is illustrated, although in other examples the search panel 500 can include inputs for other search criteria. In this example, the search text “faith
20 court shoes white” is input by the host representative using the input/display device 53 of the host representative device 14. Upon selection of a search button 504 by the host representative, the search criteria are sent to the chat management server apparatus 12. In this example, the search criteria, including the search text, is sent to the search web service 32 of the chat management server apparatus 12, which is configured to process the criteria as described and
25 illustrated in more detail later. Other methods of receiving search criteria can also be used.

[0052] In response to receiving the search text, the search web service 32 of the chat management server apparatus 12 identifies and retrieves item content responsive to the request in step 204. The item content can be retrieved from the item content catalog 30, which in this example includes content associated with a plurality of items for sale by the website host,
30 although any other type of content associated with any other type of item can also be used. The content can include an item description, an item depiction, an item price, or any other information associated with each of the items. Optionally, the content includes at least

information suitable to provide the customer with a preview of the item to allow the customer to decide whether to learn more about the item in order to make a purchasing decision.

5 [0053] In step 206, the search web service 32 of the chat management server apparatus 12 generates and provides to the host representative device 14 a definition of a search result panel based on the item content retrieved in step 204. In this example, the item content can include a picture, a short description, and a price of various white faith court shoes identified based on the search text. Accordingly, the definition includes one or more HTML fragments for the identified item(s) that includes the content and is configured to generate a search result panel including the content when rendered by the web browser 56 of the host representative device 14.

10 [0054] Referring more specifically to FIG. 6, an exemplary search result panel 600 displaying content for a plurality of selectable items identified based on the search text is illustrated. In this example, the search result panel 600 includes item content 602(1) and 601(2) for two items (“white heeled court shoes” and “patent heeled court shoes”) satisfying the search criteria received in step 204. Any number of items can be identified and included in the search result panel 600 and the search results can be displayed by the web browser 56 of the host representative device 14 in other manners.

20 [0055] Referring back to FIG. 2, in step 208, the chat management server 12 receives a selection from the host representative using the host representative device 14 of one or more of the item(s) for which content 602(1) and 601(2) was identified and retrieved in step 204, and provided to the host representative device 14 in step 206. The selected item(s) are those item(s) responsive to the customer’s request and for which the host representative would like to specifically refer the customer to in order to assist the customer in making a purchasing decision in this example.

25 [0056] Referring back to FIG. 6, in this example, the content 602(1) and 602(2) for each item is associated with a select button 604(1) and 604(2), respectively. Additionally, the search result panel 600 includes a select all button 606. Upon selection of one of the buttons 604(1), 604(2), or 606 by the host representative using the input/display device 53 of the host representative device 14, an indication of the selection is sent to the chat management server apparatus 12. Accordingly, the definition of the search result panel 600 sent to the host representative device 14 in step 206 of FIG. 2 is configured to facilitate the selection of items, although other methods of facilitating the selection of item(s) can also be used.

[0057] In step 210, the chat management server apparatus 12 generates, and optionally encodes, a search or an item token. In this example, the tokens are prefixed by a first special character (e.g., “@”) or a second special character (e.g., “#”) according to whether the token is a search token or an item token, respectively, although the first or second special character can be included in the tokens in any location. The token is a search token if all of the items for which content is displayed in the search panel 600 are selected by the host representative using the host representative device 14 (e.g., by selecting the select all button 606). Additionally, the token is an item token if fewer than all of the items or which content is displayed in the search panel 600 are selected (e.g., using one or more of the select buttons 604(1) or 604(2)).

10 [0058] In this example, if the host representative using the host representative device 14 selects the select all button 606, an exemplary token including the first special character and the search text could be “@faith_court_shoes_white”. By including the search text, the host representative can advantageously reuse this token in subsequent chat sessions as it will be relatively easy to remember. For example, the host representative can reuse the token with other prospective customers the host representative would like to refer to the same content, as described and illustrated in more detail later.

[0059] In another example, as described and illustrated in more detail later with reference to FIG. 10, if the host representative using the host representative device 14 selected only one of the items for which content was displayed on the search panel 600, an exemplary token including the second special character and a unique item identifier could be “#3611369”. Accordingly, in this example, the “3611369” portion of the token corresponds to a unique identifier for the one selected item, as stored as associated with the content for the item in the content catalog 30.

[0060] However, in yet another example, the chat management server apparatus 12 can encode the token in order to reduce the size instead of merely using the search text or the unique item identifier for the portion of the token not including the special character. Accordingly, the chat management server apparatus 12 can decide to encode the token based on whether the number of characters in the search text or unique item identifier exceeds a threshold, for example, although the decision of whether to encode a token can be based on any other criteria. Additionally, the chat management server apparatus 12 can be configured to encode all or none of the tokens as a default setting.

[0061] If the chat management server apparatus 12 determines in step 210 that the search token in this example should be encoded, an exemplary encoded search token could be

- 14 -

“@42ad42”, although any other encoding can be used. The chat management server apparatus 12 can replace the “faith_court_shoes_white” search text in the token by encoding the search text to generate an output of “42ad42”. Any type of encoding function or formula can be used. Optionally, the output of the encoding can be limited to a certain number of characters in order to
5 optimize the benefit of using an encoded token in place of a token including search text or an item identifier. Additionally, if the chat management server apparatus 12 determines in step 210 that the token should be encoded, then the chat management server apparatus 12 in this example stores at least the encoded portion of the search token as associated with the search criteria (e.g., the search text) in the encoded token database 36 so that the encoded token can subsequently be
10 decoded, as described and illustrated in more detail later with reference to step 804 of FIG. 8.

[0062] Referring back to FIG. 2, in step 212, the chat management server apparatus 12 provides the search or item token to the host representative device 14. Referring more specifically to FIG. 7, the exemplary host chat panel 400 is illustrated with an item set link 702 corresponding to the search token “@42ad42”, which is an encoded search token in this
15 example. Accordingly, upon receipt by the host representative device 14 of the search token, the host chat panel 400 inserts the search token text into the input field 404.

[0063] Upon the host representative selecting the send button 700, using the input/display device 53 of the host representative device 14, the input text including the search token is sent to the chat management server 14, which routes the text to the customer chat panel currently
20 rendered on the client device 18 using an established connection. Upon display of the input text in the host chat panel 400, as well as the customer chat panel currently rendered on the client device 18, the search token becomes the item set link 702.

[0064] Accordingly, the definition of the chat panel 400 sent to the host representative device 14 is configured to insert a token returned in step 212 into the input field 404 and render
25 the token as a hyperlink (the item set link 702 in this example) used as described and illustrated in more detail with reference to FIG. 8. Optionally, at least the customer chat panel is configured, based on its definition, to render any text sent from a host representative in a chat session that includes the first or second special character as an item set link or an item link, respectively.

30 [0065] Referring more specifically to FIG. 8, an exemplary method of generating an item preview panel based on a token will now be described. In step 800 in this example, the chat management server apparatus 12 receives a request from the client device 18 for a preview panel.

- 15 -

The request includes a token and is sent in response to a selection by the customer, using the input/display device 43 of the client device 18, of a hyperlink including the token that was rendered in the customer chat panel. The hyperlink can be rendered in the customer chat panel subsequent to the host representative submitting a message including the token. The token can
5 be manually entered by the host representative or provided by the chat management server apparatus 12, as described and illustrated earlier with reference to step 212.

[0066] In step 802, the chat management server apparatus 12 determines when the token included in the request for the preview panel received in step 800 is encoded. In order to determine whether the token is decoded, the chat management server apparatus 12 can compare
10 the token to entries of the encoded token database 36 to determine where there is a match in this example, although other methods of determining whether the token is encoded can also be used. If the chat management server apparatus 12 determines that the token is encoded, then the Yes branch is taken to step 804.

[0067] In step 804, the chat management server apparatus 12 decodes the token. In order to decode the token in this example, the chat management server apparatus 12 retrieves the actual
15 value of the portion of the token not including the special character from the matching entry of the encoded token database 36. The actual value could have been stored in the encoded token database 36 as described and illustrated in more detail earlier with reference to step 210 of FIG. 2. Accordingly, in this example, the chat management server 12 can obtain the
20 "faith_court_shoes_white" actual value by decoding the "42ad42" encoded token value. Other methods of encoding or decoding the tokens, including using a reversible encoding function that does not require a database look-up, can also be used. Subsequent to decoding the token, or if the chat management server apparatus 12 determines in step 802 that the token is not encoded and the No branch is taken, the chat management server apparatus 12 proceeds to step 806.

[0068] In step 806, the chat management server apparatus 12 retrieves item content based
25 on the token. In order to retrieve the item content, in this example, the chat management server apparatus 12 first determines whether the token is a search or an item token based on whether the token includes the first or second special character. If the chat management server apparatus 12 determines that the token is a search token, then the portion of the token not including the first
30 special character, or the decoded actual value in examples in which step 804 is performed, is processed by the search web service 34. Accordingly, the search web service 32 searches the item content catalog 30, as described and illustrated earlier with reference to step 206 of FIG. 2, using the token value as the search text in order to identify and retrieve responsive item content.

[0069] However, if the chat management server apparatus 12 determines that the token is an item token, then the portion of the token not including the first special character, or the decoded actual value in examples in which step 804 is performed, is processed by the preview web service 34. Accordingly, the preview web service 34 searches the item content catalog 30 using the token value to identify and retrieve content for an item having a unique item identifier matching the token value.

[0070] In step 808, the chat management server apparatus 12 provides a preview panel definition, including HTML fragment(s) including the content retrieved in step 806, to the client device 18 in response to the request for the preview panel received in step 800. The preview panel definition is configured to, when rendered, cause the web browser 46 of the client device 18 to generate a preview panel that includes the item content. The preview panel can be generated a pop-up window, an overlay, or any other type of display that does not require navigation by the web browser 46 away from the customer chat panel. Additionally, the preview panel can be a multi-item preview panel with a navigation structure or a single item preview panel based on whether the token, received with the request for the preview panel in step 800, is a search token or an item token, respectively.

[0071] Optionally, at least a portion of the content displayed by the preview panel is, or another portion of the preview panel includes, a link that is selectable by the customer using the client device 18 in order to allow the customer to navigate to a different web page associated with the item that provides additional content. Also optionally, at least a portion of the content displayed by the preview panel is, or another portion of the preview panel includes, a link that is selectable by the customer using the client device 18 in order to allow the customer to navigate to an item purchase web page or add the item to a shopping cart, for example. Other types of links and other content can also be provided in the preview panel.

[0072] Referring more specifically to FIG. 9, an exemplary customer chat panel 900 with the item set link 702 corresponding to the encoded search token "@42ad42" and an exemplary multi-item preview panel 902 are illustrated. In this example, the multi-item preview panel 902 is generated, based on the definition provided in step 808, subsequent to the customer selecting the item set link 702 using the input/display device of the client device 18. The multi-item preview panel 902 includes the content 602(1) for one of the items that the host representative selected to be referred to the customer in this example.

[0073] Additionally, the multi-item preview panel 902 includes a navigation structure, which in this example includes a next button 904, which facilitates navigation between content associated with a plurality of items, including at least the content 602(2) of another of the items that the host representative selected to be referred to the customer in this example. Other types of
5 navigational structures can also be used.

[0074] Referring more specifically to FIG. 10, the exemplary customer chat panel 900 of FIG. 9 with an item link 1000 and an exemplary single item preview panel 1002 are illustrated. In this example, the item link 1000 corresponds to an item token “#3611369” which includes a unique item identifier “3611369” for a single item selected by the host representative in an
10 iteration of steps 202-212 of FIG. 2 performed prior to the example iteration described and illustrated in detail earlier. The item preview panel 1002 includes content 1004 for a navy colored shoe item that the customer in this example is not interested in. Instead, the customer indicated to the host representative a preference for a white shoe and the host representative submitted the request received in step 202 in the example iteration described and illustrated
15 earlier in order to identify items responsive to the customer’s preference.

[0075] Accordingly, with this technology, representatives of website hosts can more easily and effectively refer website users to preview content for items in a chat context. The references can be sent using tokens which are generally, or can be encoded to be, shorter than URLs associated with web pages corresponding to the items. Additionally, host representatives
20 can refer users to preview content for item(s) by reusing tokens thereby facilitating relatively quick responsiveness. Moreover, items identified by a host representative can advantageously be displayed by a user in a preview panel without requiring the user to navigate away from the chat panel or the current web page and thereby significantly improving the functioning of the user’s client device.

[0076] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within
30 the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

- 18 -
CLAIMS

What is claimed is:

1. A method for facilitating references in a chat context, the method
5 comprising:
 - receiving, by a chat management server apparatus, a search request via a
search panel provided in response to a user interaction with a chat panel;
 - generating, by the chat management server apparatus, a token including a
special character and generated based on search text in the search request or a unique identifier
10 for one of a plurality of items identified based on a search performed using the search text;
 - providing, by the chat management server apparatus, the token to a source
of the search request for inclusion in the chat panel as a hyperlink;
 - receiving, by the chat management server apparatus, a preview panel
request in response to a user interaction with the hyperlink, the preview panel request including
15 the token;
 - retrieving, by the chat management server apparatus, content for the items
or for the one item based on the special character included in the token; and
 - providing, by the chat management server apparatus, the content to a
source of the preview panel request.
20
2. The method as set forth in claim 1, further comprising, prior to generating
the token:
 - retrieving, by the chat management server apparatus, the content for the
items based on the search text;
 - 25 providing, by the chat management server apparatus, the content for the
items to the source of the search request for inclusion in a search result panel; and
 - receiving, by the chat management server apparatus, a selection of one or
more of the items from the source of the search request.
- 30 3. The method as set forth in claim 2, wherein the selection is of all of the
items and the generating further comprises generating a search token comprising a first special
character and the request for the preview panel includes the search token.
4. The method as set forth in claim 2, wherein the selection is of one or more

of the items, the generating further comprises generating an item token for each of the one or more items, each item token comprises a second special character, and the request for the preview panel includes at least one of the item tokens.

5 5. The method as set forth in claim 3, wherein the search token includes at least one or more terms included in the search text.

 6. The method as set forth in claim 1, wherein:
 the generating further comprises encoding the search text or the unique
10 identifier and storing the encoded search text or the encoded unique identifier in an encoded token database as associated with the corresponding search text or unique identifier, wherein the token is generated based on the encoded search text or the encoded unique identifier; and
 the retrieving further comprises decoding the token comprising comparing the token to the encoded token database to retrieve the search text or unique identifier.

15 7. The method as set forth in claim 1, wherein the retrieving further comprises:
 determining when the token included in the preview panel request is a search token based on a match of a specified character of the token with a first special character;
20 performing a search of an item content catalog using the search text, as determined based on the token included in the preview panel request, to retrieve the content for the items, when the token included in the preview panel request is determined to be the search token; and
 retrieving content for the one item based on the unique identifier, as
25 determined based on the token included in the preview panel request, when the token included in the preview panel request is not determined to be the search token.

 8. A non-transitory computer readable medium having stored thereon instructions for facilitating references in a chat context comprising executable code which when
30 executed by a processor, causes the processor to perform steps comprising:
 receiving a search request via a search panel provided in response to a user interaction with a chat panel;
 generating a token including a special character and generated based on search text in the search request or a unique identifier for one of a plurality of items identified

based on a search performed using the search text;

providing the token to a source of the search request for inclusion in the chat panel as a hyperlink;

receiving a preview panel request in response to a user interaction with
5 the hyperlink, the preview panel request including the token;

retrieving content for the items or for the one item based on the special character included in the token; and

providing the content to a source of the preview panel request.

10 9. The non-transitory computer readable medium as set forth in claim 8, further having stored thereon instructions that when executed by the processor cause the processor to perform steps further comprising, prior to generating the search token or the item token:

retrieving the content for the items based on the search text;

15 providing the content for the items to the source of the search request for inclusion in a search result panel; and

receiving a selection of one or more of the items from the source of the search request.

20 10. The non-transitory computer readable medium as set forth in claim 9, wherein the selection is of all of the items and the generating further comprises generating a search token comprising a first special character and the request for the preview panel includes the search token.

25 11. The non-transitory computer readable medium as set forth in claim 9, wherein the selection is of one or more of the items, the generating further comprises generating an item token for each of the one or more items, each item token comprises a second special character, and the request for the preview panel includes at least one of the item tokens.

30 12. The non-transitory computer readable medium as set forth in claim 10, wherein the search token includes at least one or more terms included in the search text.

13. The non-transitory computer readable medium as set forth in claim 8, wherein:

- 21 -

the generating further comprises encoding the search text or the unique identifier and storing the encoded search text or the encoded unique identifier in an encoded token database as associated with the corresponding search text or unique identifier, wherein the token is generated based on the encoded search text or the encoded unique identifier; and

5 the retrieving further comprises decoding the token comprising comparing the token to the encoded token database to retrieve the search text or unique identifier.

14. The non-transitory computer readable medium as set forth in claim 8, wherein the retrieving further comprises:

10 determining when the token included in the preview panel request is a search token based on a match of a specified character of the token with a first special character; performing a search of an item content catalog using the search text, as determined based on the token included in the preview panel request, to retrieve the content for the items, when the token included in the preview panel request is determined to be the search
15 token; and

retrieving content for the one item based on the unique identifier, as determined based on the token included in the preview panel request, when the token included in the preview panel request is not determined to be the search token.

20 15. A chat management server apparatus, comprising a processor and a memory coupled to the processor which is configured to be capable of executing programmed instructions comprising and stored in the memory to:

receive a search request via a search panel provided in response to a user interaction with a chat panel;

25 generate a token including a special character and generated based on search text in the search request or a unique identifier for one of a plurality of items identified based on a search performed using the search text;

provide the token to a source of the search request for inclusion in the chat panel as a hyperlink;

30 receive a preview panel request in response to a user interaction with the hyperlink, the preview panel request including the token;

retrieve content for the items or for the one item based on the special character included in the token; and

provide the content to a source of the preview panel request.

16. The chat management server apparatus as set forth in claim 15, wherein the processor coupled to the memory is further configured to be capable of executing programmed instructions further comprising and stored in the memory to, prior to generating the search token or the item token:

5 retrieve the content for the items based on the search text;
provide the content for the items to the source of the search request for inclusion in a search result panel; and
receive a selection of one or more of the items from the source of the search request.

17. The chat management server apparatus as set forth in claim 16, wherein the selection is of all of the items, the processor coupled to the memory is further configured to be capable of executing at least one additional programmed instruction further comprising and stored in the memory to generate a search token comprising a first special character, and the request for the preview panel includes the search token.

18. The chat management server apparatus as set forth in claim 16, wherein the selection is of one or more of the items, the processor coupled to the memory is further configured to be capable of executing at least one additional programmed instruction further comprising and stored in the memory to generate an item token for each of the one or more items, each item token comprises a second special character, and the request for the preview panel includes at least one of the item tokens.

19. The chat management server apparatus as set forth in claim 17, wherein the search token includes at least one or more terms included in the search text.

20. The chat management server apparatus as set forth in claim 15, wherein the processor coupled to the memory is further configured to be capable of executing programmed instructions further comprising and stored in the memory to:

30 encode the search text or the unique identifier and storing the encoded search text or the encoded unique identifier in an encoded token database as associated with the corresponding search text or unique identifier, wherein the token is generated based on the encoded search text or the encoded unique identifier; and

- 23 -

decode the token comprising comparing the token to the encoded token database to retrieve the search text or unique identifier.

21. The chat management server apparatus as set forth in claim 15, wherein
- 5 the processor coupled to the memory is further configured to be capable of executing programmed instructions further comprising and stored in the memory to:
- determine when the token included in the preview panel request is a search token based on a match of a specified character of the token with a first special character;
 - perform a search of an item content catalog using the search text, as
 - 10 determined based on the token included in the preview panel request , to retrieve the content for the items, when the token included in the preview panel request is determined to be the search token; and
 - retrieve content for the one item based on the unique identifier, as
 - determined based on the token included in the preview panel request, when the token included in
 - 15 the preview panel request is not determined to be the search token.

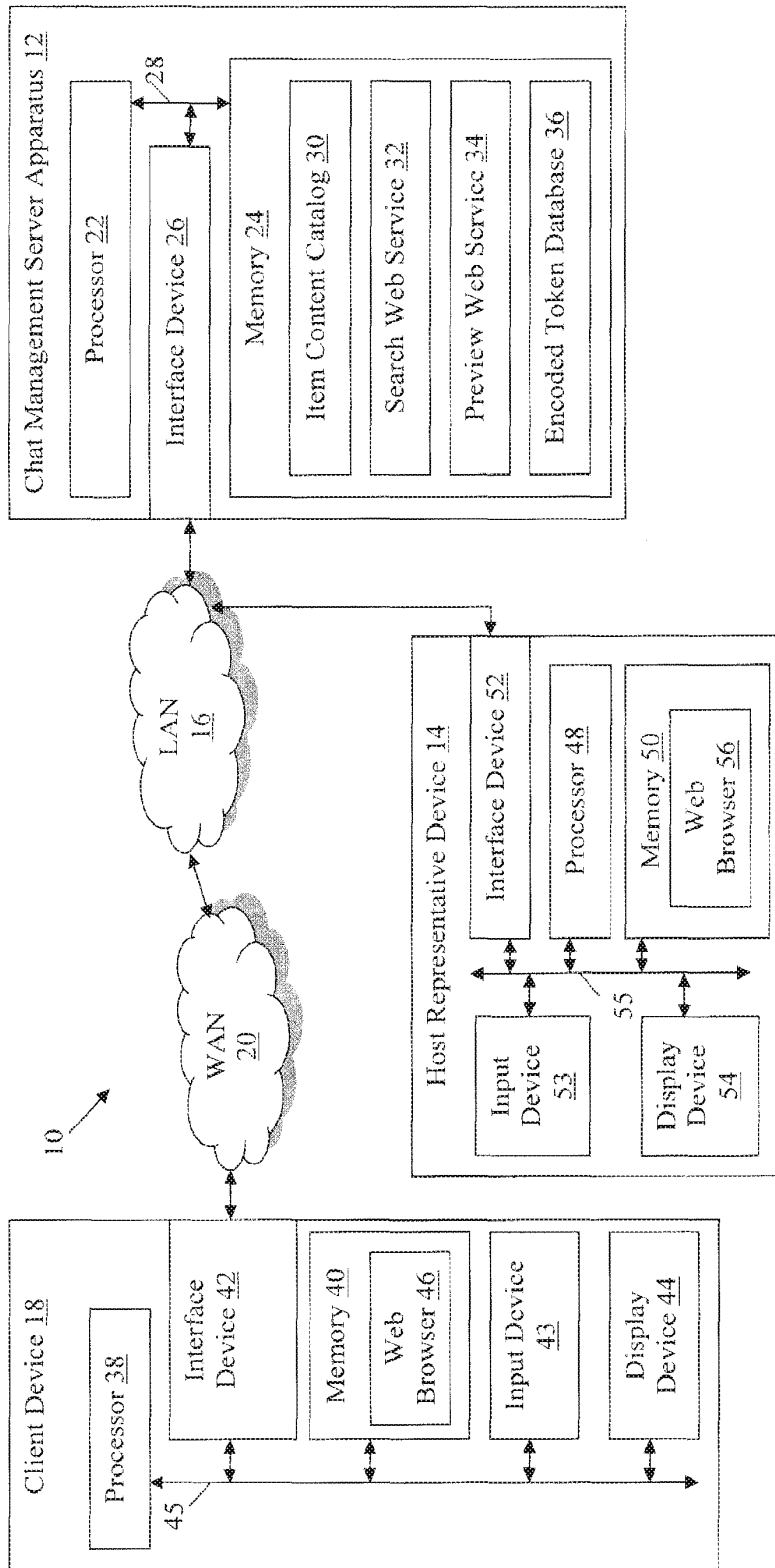


FIG. 1

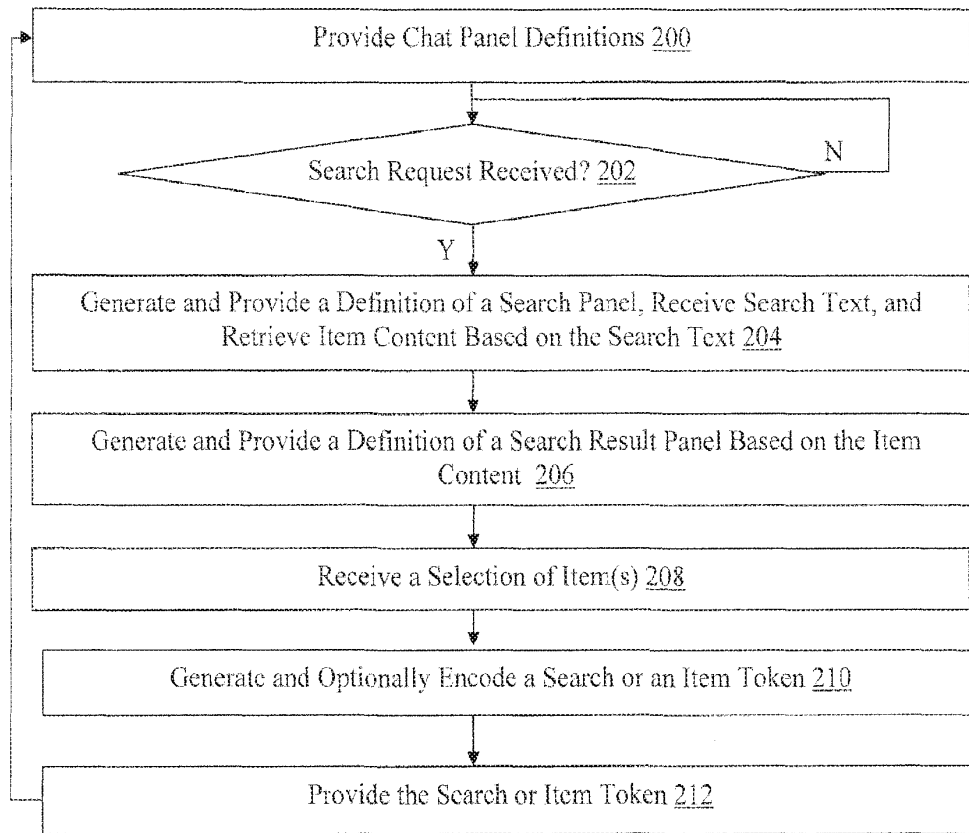


FIG. 2

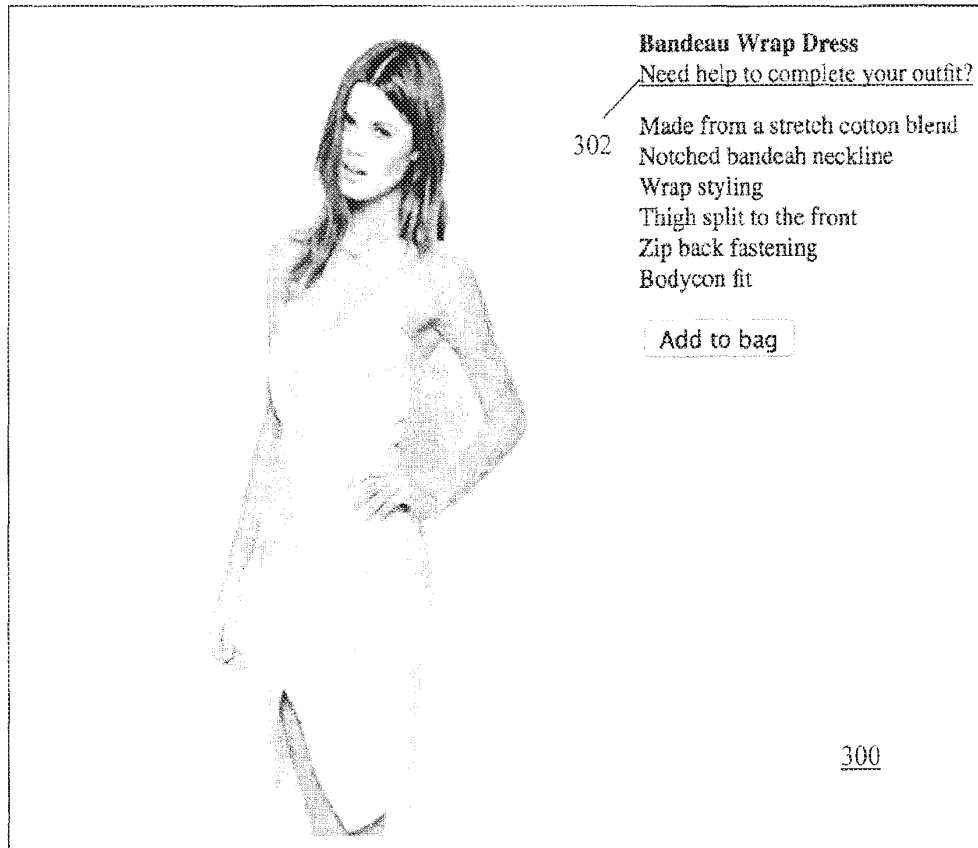


FIG. 3

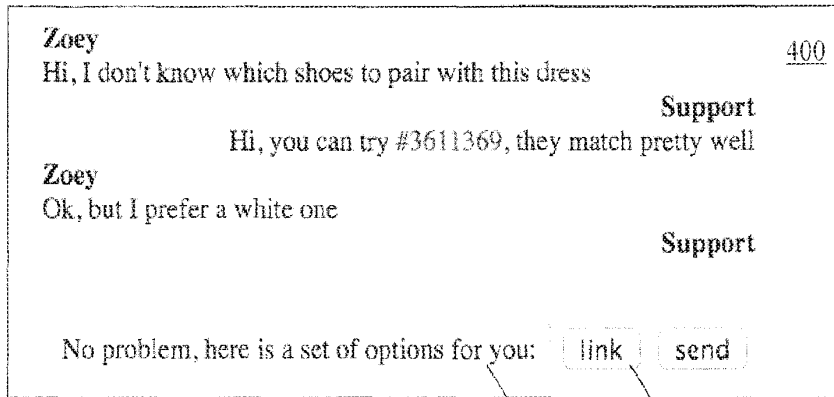


FIG. 4

404

402

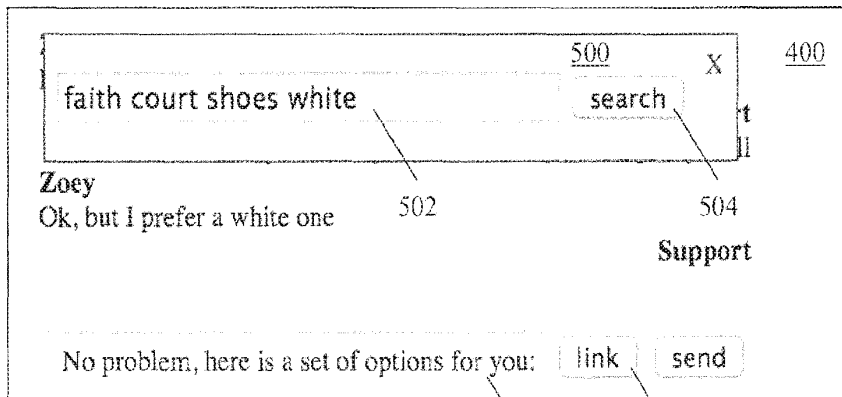


FIG. 5

404

402

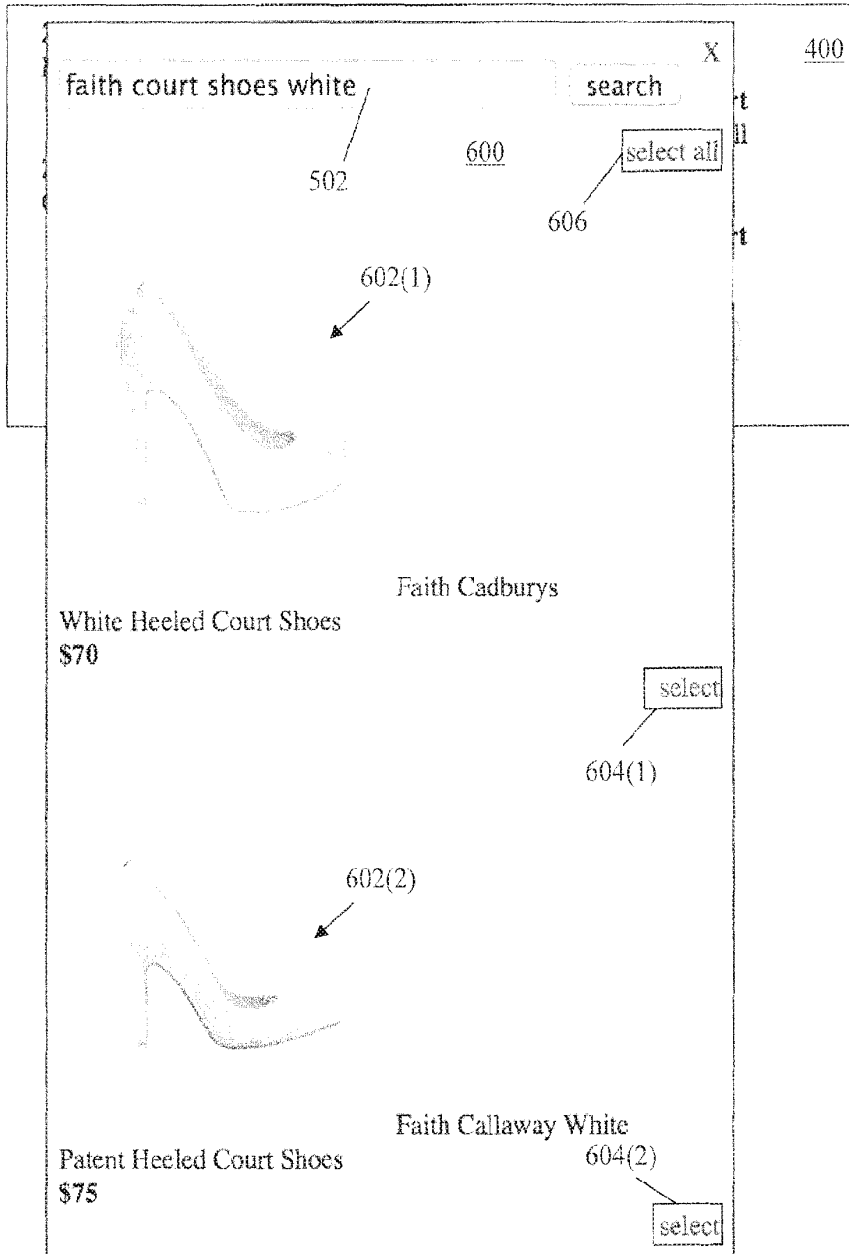


FIG. 6

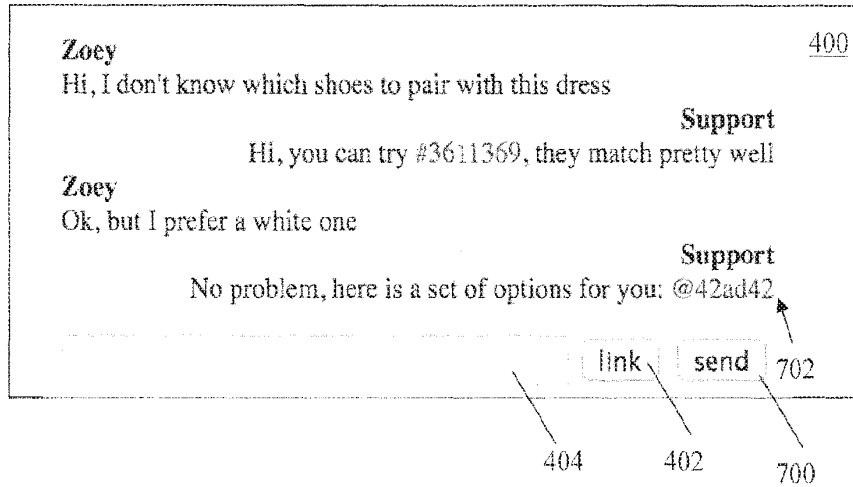


FIG. 7

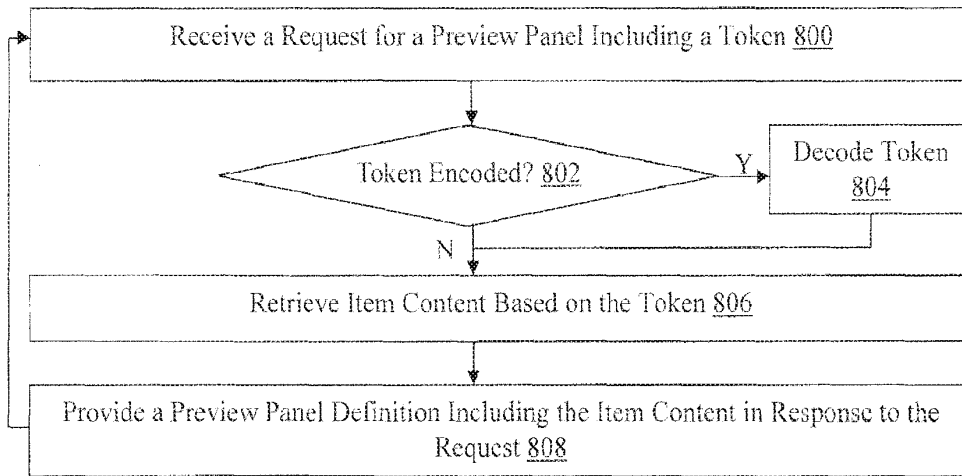


FIG. 8

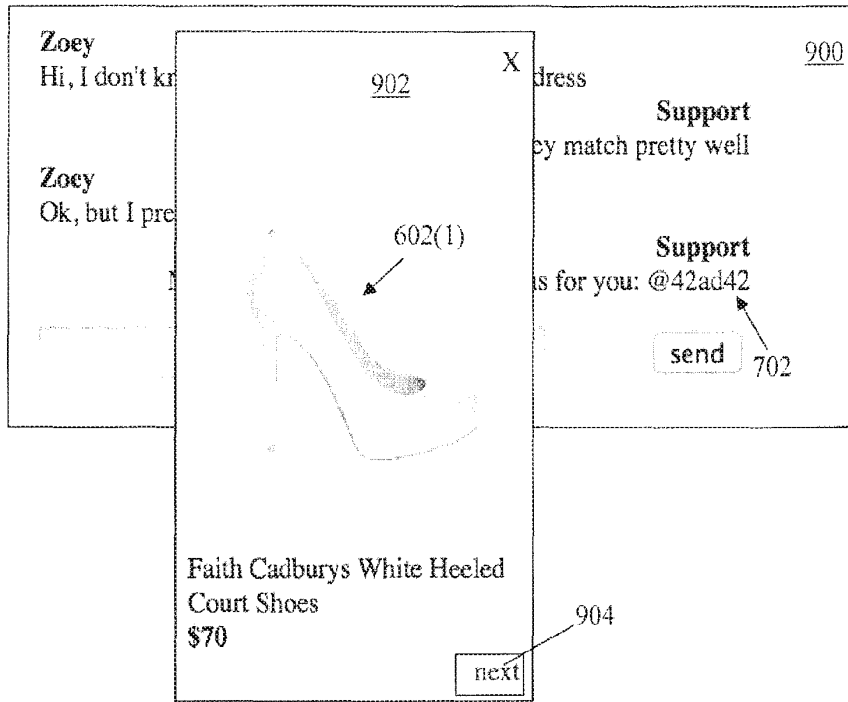


FIG. 9

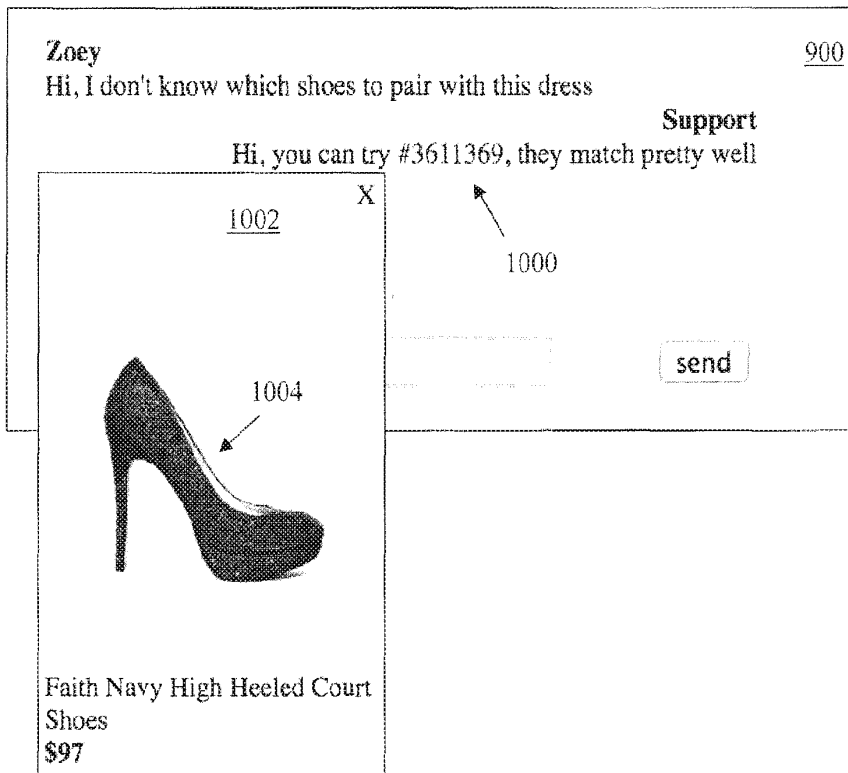


FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US15/36956

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 17/30 (2015.01) CPC - G06F 17/30 According to International Patent Classification (IPC) or to both national classification and IPC</p>																	
<p>B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC(8) Classification(s): G06F 17/30, 13/00; G06Q 30/00 (2015.01) CPC Classification(s): G06F 17/30, 17/30864; Y10S 707/99933; G06Q 30/00; H04L 51/04 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) PatSeer (US, EP, WO, JP, DE, GB, CN, FR, KR, ES, AU, IN, CA, Other Countries (INPADOC), RU, AT, CH, TH, BR, PH); IEEE/EEEExplore; Google/Google Scholar; IP.com; Keywords: web chat, live chat, text, SMS, panel, window, search, query, keyword, terms, token, symbol, character, icon, image, avatar, identifier, reference, results, response, preview, portion, segment, area, website, URI, URL, hyperlink</p>																	
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1"> <thead> <tr> <th>Category*</th> <th>Citation of document, with indication, where appropriate, of the relevant passages</th> <th>Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>US 6678673 B1 (EVES, D et al.) January 13, 2004; abstract; column 3, lines 46-52; column 4, lines 44-55; column 7, lines 10-16; column 7, lines 20-27</td> <td>1-5, 7-12, 14-19, 21</td> </tr> <tr> <td>Y</td> <td></td> <td>6, 13, 20</td> </tr> <tr> <td>Y</td> <td>US 6804664 B1 (HARTMAN, B et al.) October 12, 2004; abstract</td> <td>6, 13, 20</td> </tr> <tr> <td>A</td> <td>US 2013/0311339 A1 (JEREMIAS, L) November 21, 2013; entire document</td> <td>1-21</td> </tr> </tbody> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	X	US 6678673 B1 (EVES, D et al.) January 13, 2004; abstract; column 3, lines 46-52; column 4, lines 44-55; column 7, lines 10-16; column 7, lines 20-27	1-5, 7-12, 14-19, 21	Y		6, 13, 20	Y	US 6804664 B1 (HARTMAN, B et al.) October 12, 2004; abstract	6, 13, 20	A	US 2013/0311339 A1 (JEREMIAS, L) November 21, 2013; entire document	1-21
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.															
X	US 6678673 B1 (EVES, D et al.) January 13, 2004; abstract; column 3, lines 46-52; column 4, lines 44-55; column 7, lines 10-16; column 7, lines 20-27	1-5, 7-12, 14-19, 21															
Y		6, 13, 20															
Y	US 6804664 B1 (HARTMAN, B et al.) October 12, 2004; abstract	6, 13, 20															
A	US 2013/0311339 A1 (JEREMIAS, L) November 21, 2013; entire document	1-21															
<p><input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.</p>																	
<p>* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family</p>																	
<p>Date of the actual completion of the international search 28 August 2015 (28.08.2015)</p>		<p>Date of mailing of the international search report 16 SEP 2015</p>															
<p>Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300</p>		<p>Authorized officer Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774</p>															

Form PCT/ISA/210 (second sheet) (January 2015)

INFORMATION DISCLOSURE STATEMENT

First Inventor :	Sean D. Bradley
App. No. :	16/430210
Filed :	June 3, 2019
For :	MODULAR SYSTEMS AND METHODS FOR SELECTIVELY ENABLING CLOUD-BASED ASSISTIVE TECHNOLOGIES
Examiner :	Zuberi, Mohammed H.
Art Unit :	2177
Conf. No. :	6600

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. Copies of any listed foreign and non-patent literature references are being submitted. Any foreign references may also include English abstract(s) and/or machine translation(s), but no representation is made as to their accuracy.

If the Examiner would like additional information regarding these references or if anything is unclear, the Examiner is invited to contact the undersigned for assistance.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed before the receipt of a First Office Action on the merits, and presumably no fee is required. If a First Office Action on the merits was mailed before the mailing date of this Statement, the Commissioner is authorized to charge the fee set forth in 37 CFR 1.17(p) to Deposit Account No. 11-1410.

Docket No.: AUDEY.003C1
App. No.: 16/430210

Page 2 of 2

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: August 13, 2020

By: Xiaoyu Wang
Xiaoyu Wang
Registration No. 75,640
Registered Practitioner
(212) 849-3000

33306741

Doc Code: ECOMM.AUTH/ECOMM.WTDW

Doc Description: Internet Communications Authorized/Internet Communications Authorization Withdrawn

PTO/SB/439 (11-15)

AUTHORIZATION FOR INTERNET COMMUNICATIONS IN A PATENT APPLICATION OR REQUEST TO WITHDRAW AUTHORIZATION FOR INTERNET COMMUNICATIONS	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2177
	Examiner Name	Zuberi, Mohammed H.
	Practitioner Docket No.	AUDEY.003C1

To: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. To authorize permission for Internet Communications.

Recognizing that Internet communications are not secure, I hereby authorize the USPTO to communicate with the undersigned and practitioners in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via video conferencing, instant messaging, or electronic mail. I understand that a copy of these communications will be made of record in the application file. (MPEP 502.03)

II. To withdraw authorization for Internet Communications.

The authorization given on _____, to the USPTO to communicate with the undersigned and any practitioner in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via Internet communications such as video conferencing, instant messaging, or electronic mail is hereby withdrawn. I understand that the withdrawal is effective when approved rather than when received.

I am the

- applicant.
- attorney or agent of record. Registration number 57013
- attorney or agent acting under 37 CFR 1.34. Registration number _____

/THOMAS Y. YEE/ _____ 10/29/2020 _____
Signature Date

Thomas Y. Yee _____ 212-849-3077 _____
Typed or printed name Telephone Number

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Juristic entities must be represented by a patent practitioner (see 37 CFR 1.31, which is applicable to any paper filed on or after September 16, 2012, that is presented on behalf of a juristic entity, regardless of application filing date). Submit multiple forms if more than one signature is required, see below*.

* Total of 1 forms are submitted.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Acknowledgement Receipt

EFS ID:	40991573
Application Number:	16430210
International Application Number:	
Confirmation Number:	6600
Title of Invention:	Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies
First Named Inventor/Applicant Name:	Sean D. Bradley
Customer Number:	20995
Filer:	Thomas Yee/Daisy Perez
Filer Authorized By:	Thomas Yee
Attorney Docket Number:	AUDEY.003C1
Receipt Date:	30-OCT-2020
Filing Date:	03-JUN-2019
Time Stamp:	12:49:40
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	IDS.pdf	48389 <small>7b63935e8bb4d62bc666b7a5ad0fe9da206c2115</small>	no	3

Warnings:

Information:					
This is not an USPTO supplied IDS fillable form					
2	Internet Communications Authorized	Internet_Authorization.pdf	88714	no	2
			07db75b2b857e2e5984a9e508232a90c0e036697		
Warnings:					
Information:					
3	Non Patent Literature	Sato_Accessibility_Evaluation_2006.pdf	173631	no	2
			b7d25b9ce7adff8435eeeb252ffeeca77c7984eb5		
Warnings:					
Information:					
4	Non Patent Literature	Shadi_Artificial_Intelligence_for_Web.pdf	405698	no	5
			befd9b3c3c27d60657beadb093f0f1b5f31c152ba		
Warnings:					
Information:					
Total Files Size (in bytes):			716432		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

INFORMATION DISCLOSURE STATEMENT

First Inventor :	Sean D. Bradley
App. No. :	16/430210
Filed :	June 3, 2019
For :	MODULAR SYSTEMS AND METHODS FOR SELECTIVELY ENABLING CLOUD-BASED ASSISTIVE TECHNOLOGIES
Examiner :	Zuberi, Mohammed H.
Art Unit :	2177
Conf. No. :	6600

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed before the receipt of a First Office Action on the merits, and presumably no fee is required. If a First Office Action on the merits was mailed before the mailing date of this Statement, the Commissioner is authorized to charge the fee set forth in 37 CFR 1.17(p) to Deposit Account No. 11-1410.

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: October 30, 2020

By: /THOMAS Y. YEE/_____
Thomas Y. Yee
Registration No. 57,013
Registered Practitioner
(212) 849-3000

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2177
<i>(Multiple sheets used when necessary)</i>	Examiner	Zuberi, Mohammed H.
SHEET 1 OF 2	Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number <i>Number - Kind Code (if known)</i> Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	7,546,531	06-09-2009	Celik	
	2	8,452,600	05-28-2013	Fleizach	
	3	8,855,423	10-07-2014	Boncyk et al.	
	4	10,037,318	07-31-2018	Liu	
	5	10,425,501	06-09-2016	Nasson et al.	
	6	10,762,280 (Our Ref. AUDEY.002C1), including its prosecution history, the cited references, and the Office Actions therein	09-01-2020	Bradley et al.	
	7	10,809,877(Our Ref. AUDEY.003C4), including its prosecution history, the cited references, and the Office Actions therein	10-20-2020	Bradley et al.	
	8	2002/0122053	09-05-2002	Dutta	
	9	2012/0254723	10-04-2012	Kasa et al.	
	10	2013/0104029	04-25-2013	Hendry et al.	
	11	2015/0106686	04-16-2015	Blitzstein	
	12	2017/0371975	12-28-2017	Chen et al.	
	13	2019/0340212	11-07-2019	Isager	
	14	16/991329**	08-12-2020	Bradley et al.	
	15	16/991346**	08-12-2020	Bradley et al.	
	16	16/991416**	08-12-2020	Bradley et al.	
	17	16/991392**	08-12-2020	Bradley et al.	
	18	16/991304**	08-12-2020	Bradley et al.	
	19	16/991381**	08-12-2020	Bradley et al.	
	20	16/991434**	08-12-2020	Bradley et al.	

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2177
<i>(Multiple sheets used when necessary)</i>	Examiner	Zuberi, Mohammed H.
SHEET 2 OF 2	Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number <i>Number - Kind Code (if known)</i> Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	21	16/991405**	08-12-2020	Bradley et al.	
	22	16/991505**	08-12-2020	Bradley et al.	
	23	16/991584**	08-12-2020	Bradley et al.	
	24	16/991578**	08-12-2020	Bradley et al.	
	25	16/991671**	08-12-2020	Bradley et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document <i>Country Code-Number-Kind Code</i> Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	26	Sato et al., "Accessibility Evaluation based on Machine Learning Technique", October 22-25, 2006, Portland, Oregon, USA, ACM, pages: 2 (Year: 2006)	
	27	Shadi et al., "Artificial Intelligence for Web Accessibility- Conformance Evaluation as a Way Forward?", April 23-25, 2018, Lyon, France, ACM, pages: 5 (Year: 2018)	

**Reference is co-owned by Applicant and signature indicates consideration of publication and file history, including references cited and office actions issued therein. The Examiner has access to these materials through PTO computer systems. If additional copies are desired, please notify Applicant through its attorneys.
33780263

Examiner Signature	Date Considered
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

Doc Code: DIST.E.FILE Document Description: Electronic Terminal Disclaimer - Filed		PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce
Electronic Petition Request	TERMINAL DISCLAIMER TO OBTAIN A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	
Application Number	16430210	
Filing Date	03-Jun-2019	
First Named Inventor	Sean Bradley	
Attorney Docket Number	AUDEY.003C1	
Title of Invention	Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies	
<input checked="" type="checkbox"/> Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action <input checked="" type="checkbox"/> This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
AUDIOEYE, INC.	100%	
<p>The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)</p> <p>10444934</p> <p>as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.</p> <p>In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:</p> <ul style="list-style-type: none"> - expires for failure to pay a maintenance fee; - is held unenforceable; - is found invalid by a court of competent jurisdiction; - is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321; - has all claims canceled by a reexamination certificate; - is reissued; or - is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer. <p><input checked="" type="radio"/> Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.</p>		

I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- Small Entity
- Micro Entity
- Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application

Registration Number 57013
- A sole inventor
- A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- A joint inventor; all of whom are signing this request

Signature	/THOMAS Y. YEE/
Name	THOMAS Y. YEE

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal

Application Number:	16430210				
Filing Date:	03-Jun-2019				
Title of Invention:	Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies				
First Named Inventor/Applicant Name:	Sean D. Bradley				
Filer:	Thomas Yee				
Attorney Docket Number:	AUDEY.003C1				
Filed as Large Entity					
Filing Fees for Utility under 35 USC 111(a)					
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)	
Basic Filing:					
STATUTORY OR TERMINAL DISCLAIMER	1814	1	170	170	
Pages:					
Claims:					
Miscellaneous-Filing:					
Petition:					
Patent-Appeals-and-Interference:					
Post-Allowance-and-Post-Issuance:					

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				170

Doc Code: DISQ.E.FILE
Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 16430210

Filing Date: 03-Jun-2019

Applicant/Patent under Reexamination: Bradley

Electronic Terminal Disclaimer filed on December 31, 2020

APPROVED

This patent is subject to a terminal disclaimer

DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt

EFS ID:	41534634
Application Number:	16430210
International Application Number:	
Confirmation Number:	6600
Title of Invention:	Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies
First Named Inventor/Applicant Name:	Sean D. Bradley
Customer Number:	20995
Filer:	Thomas Yee
Filer Authorized By:	
Attorney Docket Number:	AUDEY.003C1
Receipt Date:	31-DEC-2020
Filing Date:	03-JUN-2019
Time Stamp:	16:18:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	\$170
RAM confirmation Number	E2020BUG18148221
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

--	--	--	--	--	--

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	33378	no	2
			00233fcb0460cd4a96a4cd188a8777c0addae76c		

Warnings:

Information:

2	Fee Worksheet (SB06)	fee-info.pdf	30346	no	2
			349a1ab2539aff53713198347d943f14c62de526		

Warnings:

Information:

Total Files Size (in bytes):	63724
-------------------------------------	-------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

20995 7590 01/11/2021
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER
ZUBERI, MOHAMMED H

ART UNIT PAPER NUMBER

2177

DATE MAILED: 01/11/2021

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO. Values: 16/430,210, 06/03/2019, Sean D. Bradley, AUDEY.003C1, 6600

TITLE OF INVENTION: Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies

Table with 7 columns: APPLN. TYPE, ENTITY STATUS, ISSUE FEE DUE, PUBLICATION FEE DUE, PREV. PAID ISSUE FEE, TOTAL FEE(S) DUE, DATE DUE. Values: nonprovisional, UNDISCOUNTED, \$1200, \$0.00, \$0.00, \$1200, 04/12/2021

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

20995 7590 01/11/2021
KNOBBE MARTENS OLSON & BEAR LLP
 2040 MAIN STREET
 FOURTEENTH FLOOR
 IRVINE, CA 92614

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

_____ (Typed or printed name)
_____ (Signature)
_____ (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/430,210	06/03/2019	Sean D. Bradley	AUDEY.003C1	6600

TITLE OF INVENTION: Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	04/12/2021

EXAMINER	ART UNIT	CLASS-SUBCLASS
ZUBERI, MOHAMMED H	2177	715-747000

<p>1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).</p> <p><input type="checkbox"/> Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.</p> <p><input type="checkbox"/> "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-09 or more recent) attached. Use of a Customer Number is required.</p>	<p>2. For printing on the patent front page, list</p> <p>(1) The names of up to 3 registered patent attorneys or agents OR, alternatively, _____ 1</p> <p>(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. _____ 2</p> <p>_____ 3</p>
---	---

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE _____ (B) RESIDENCE: (CITY and STATE OR COUNTRY) _____

Please check the appropriate assignee category or categories (will not be printed on the patent): Individual Corporation or other private group entity Government

4a. Fees submitted: Issue Fee Publication Fee (if required) Advance Order - # of Copies _____

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

Electronic Payment via EFS-Web Enclosed check Non-electronic payment by credit card (Attach form PTO-2038)

The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. Change in Entity Status (from status indicated above)

Applicant certifying micro entity status. See 37 CFR 1.29

Applicant asserting small entity status. See 37 CFR 1.27

Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO. Includes details for application 16/430,210, inventor Sean D. Bradley, and attorney AUDEY.003C1.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 16/430,210	Applicant(s) Bradley et al.	
	Examiner MOHAMMED H ZUBERI	Art Unit 2177	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. This communication is responsive to CON filed 6/3/2019.
 A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
2. An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. The allowed claim(s) is/are 11-16 and 42-55. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.
4. Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) All b) Some *c) None of the:
1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.
Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|--|
| 1. <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 5. <input checked="" type="checkbox"/> Examiner's Amendment/Comment |
| 2. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date <u>See Continuation Sheet</u> . | 6. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance |
| 3. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material _____. | 7. <input type="checkbox"/> Other _____. |
| 4. <input checked="" type="checkbox"/> Interview Summary (PTO-413),
Paper No./Mail Date. <u>12/31/2020</u> . | |

/MOHAMMED H ZUBERI/
Primary Examiner, Art Unit 2177

Continuation of Attachment(s) 2. Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date: 10/30/2020, 8/13/2020, 6/22/2020, 8/12/2019

DETAILED ACTION

Claims **1 – 22** are pending in the case. Claims **1 and 10** are independent claims.

Information Disclosure Statement

The information disclosure statements (IDS) submitted on **10/30/2020, 8/13/2020, 6/22/2020, 8/12/2019**, are in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

Drawings

The drawings filed on **6/3/2019** have been accepted by the Examiner.

EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with **Thomas Yee (Registration Number 57013) on 12/31/2020.**

The application has been amended as follows:

CROSS-REFERENCE TO RELATED APPLICATIONS

1-10 (Cancelled)

11. (Amended) A non-transitory electronic storage medium with computer code stored thereon, in a non-transient medium the computer code configured to programmatically assign a descriptive attribute to an element on a web page to provide an audible description of the element, the web page having an associated document object model (DOM), and the computer code configured to perform, when executed by a computer processor, the steps of:
accessing code associated with the web page, the code comprising HTML or the DOM;
retrieving at least one remediation code from a remote server, the at least one remediation code configured to associate a descriptive attribute to an element in the web page, the element in the web page lacking an adequate descriptive attribute;
amending the web page, based on applying the retrieved at least one remediation code, to assign the descriptive attribute to the element in the web page, the descriptive attribute configured to enable an assistive technology to speak the descriptive attribute to a user;
~~superimposing~~ amending the web page to display a timed-toggle-able of a branding icon over a universal accessibility icon on a the web page; and

~~revealing~~displaying, in response to input from the a-user command to the displayed toggle-able universal accessibility icon, an assistive technology tool bar including user controls for independently enabling remediating the web page at least an audio player and a reader tool suite, wherein user controls of the assistive technology tool bar are configured to enable the user to dynamically change on the web page a font type, magnification, and contrast of colors displayed to the user.

12. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the user ~~command~~ input comprises one or more of mouse hovering over and or clicking on the branding icon toggle-able universal accessibility icon.

13. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein independently enabling comprises selecting one of: a first option to enable ~~the~~ an audio player without enabling the reader tool suite assistive technology tool bar; a second option to enable the ~~reader tool suite~~ assistive technology tool bar without enabling the audio player; ~~and or~~ a third option to simultaneously enable both the audio player and the ~~reader tool suite~~ assistive technology tool bar.

14. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein ~~superimposing~~ amending the web page to display the timed toggle-able universal accessibility icon comprises using one of a cascading style sheet (CSS) ~~and~~ or JavaScript to periodically animate the ~~branding icon~~ universal accessibility icon.

15. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the assistive technology tool bar further includes a control for independently enabling a help desk function.

16. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the computer code comprises JavaScript.

17.-41. (Cancelled)

42. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the computer code comprises one or more program codes.

43. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the amending the web page based on applying the retrieved at least one remediation code comprises amending the DOM or HTML associated with the web page.

44. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the at least one remediation code comprises javascript.

45. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the element lacking an adequate descriptive attribute is a graphic.

46. (New) A computer-implemented method of programmatically assigning a descriptive attribute to an element on a web page to enable an audible description of the element, the web page having an associated document object model (DOM), the computer-implemented method comprising:

accessing code associated with the web page, the code comprising HTML or the DOM;

retrieving at least one remediation code from a remote server, the at least one remediation code configured to associate a descriptive attribute to an element in the web page, the element in the web page lacking an adequate descriptive attribute;

amending the web page, based on applying the retrieved at least one remediation code to identify the element in the code associated with the web page, to associate the descriptive attribute to the element in the web page, the descriptive attribute configured to enable an assistive technology to speak the descriptive attribute to a user;

amending the web page to display a universal accessibility icon on the web page, the universal accessibility icon configured to receive input from the user; and

displaying, in response to input from the user to the displayed universal accessibility icon, an assistive technology tool bar including user controls for independently enabling remediating the web page,

wherein user controls of the assistive technology tool bar are configured to enable the user to dynamically change on the web page a font type, magnification, and contrast of colors displayed to the user.

47. (New) The computer-implemented method of claim 46, wherein the amending the web page based on applying the retrieved at least one remediation code comprises amending the DOM or HTML associated with the web page.

48. (New) The computer-implemented method of claim 46, wherein the element lacking an adequate descriptive attribute is a graphic.

49. (New) The computer-implemented method of claim 46, wherein the at least one remediation code comprises javascript.

50. (New) The computer-implemented method of claim 46, wherein the element lacking an adequate descriptive attribute is a video.

51. (New) The computer-implemented method of claim 46, wherein the descriptive attribute associated with the element in the web page is an alt-text tag.

52. (New) The computer-implemented method of claim 46, wherein the descriptive attribute is generated by an artificial intelligence algorithm.

53. (New) The computer-implemented method of claim 46, wherein the computer-implemented method is based on executing one or more program codes.

54. (New) The computer-implemented method of claim 46, wherein the computer-implemented method is based on executing a JavaScript.

55. (New) The computer-implemented method of claim 46, wherein the element lacking an adequate descriptive attribute is an input field.

REASONS FOR ALLOWANCE

The following is an examiner's statement of reasons for allowance: The prior art of record discusses aspects of the claimed invention, however the combination of cited prior art fails to teach the all of features claimed. The closest prior art of record discuss aspects of the features present in the claims, however they fail to individually, or in combination, teach each of the features present in the allowable claims.

Blitzstein (USPUB 20150106686 A1), discusses generating content rules and additional content for a webpage, the content rules directed to providing accessibility content to visually impaired users. Additional fields, images, descriptions, text or other information may be added to existing webpage simply by creating rules for the inclusion of the additional content and storing the rules and the additional content to the rules database. Additional content may be derived based on standards and guidelines provided, and utilized in the analysis of a web page. A web page is provided to a client device, a determination is made as to whether there is a rule set and additional content available for the URL in a rules database, if a match is found then rule set is accessed which includes additional content to be injected into the webpage (0044-56).

Hendry (USPUB 20130104029) discusses automated addition of accessibility features to documents, wherein a first document is analyzed to identify a first set of one or more tags and responsive to identifying the first set of one or more tags, automatically producing a second document based in part on first set of one or more tags, where the second document includes one or more accessibility features that were not in the first document.

Trujillo (USPUB 20100070872) discusses adaptive techniques used to render dynamic web content for accessibility software applications, such as screen readers, additional content is rendered on the webpage to screen reader applications, and is placed at the position of the screen reader focus.

Bradley (USPUB 20120240045) discusses audio files representing files intended primarily for viewing being created and organized into hierarchies that mimic those of the original files as instantiated at the original websites so as to provide access to visually impaired users with access to and navigation of the audio files in a way that mimics the original website.

Edwards (USPUB 20080133500) discusses a website evaluation and recommendation tool which accesses a website, evaluates the website, and provides recommended changes.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee.

Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MOHAMMED H ZUBERI whose telephone number is (571)270-7761. The examiner can normally be reached on Monday-Thursday 10AM-8PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Cesar Paula can be reached on 571-272-4128. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MOHAMMED H ZUBERI/
Primary Examiner, Art Unit 2177

<i>Examiner-Initiated Interview Summary</i>	Application No. 16/430,210	Applicant(s) Bradley et al.	
	Examiner MOHAMMED H ZUBERI	Art Unit 2177	AIA (FITF) Status Yes

All participants (applicant, applicant's representative, PTO personnel):

(1) MOHAMMED H. ZUBERI. (3) ____.

(2) THOMAS YEE. (4) ____.

Date of Interview: 31 December 2020.

Type: Telephonic Video Conference
 Personal [copy given to: applicant applicant's representative]

Exhibit shown or demonstration conducted: Yes No.
If Yes, brief description: ____.

Issues Discussed 101 112 102 103 Others
(For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: 1-55.

Identification of prior art discussed: ____.

Substance of Interview

(For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Discussed restriction requirement, elected group and agreed to amendments to the claims that would place the application in condition for allowance. Terminal disclaimer was also discussed and filed..

Applicant recordation instructions: It is not necessary for applicant to provide a separate record of the substance of interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

Attachment

/MOHAMMED H ZUBERI/
Primary Examiner, Art Unit 2177

Notice of References Cited	Application/Control No. 16/430,210	Applicant(s)/Patent Under Reexamination Bradley et al.	
	Examiner MOHAMMED H ZUBERI	Art Unit 2177	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20080133500-A1	06-2008	Edwards; Joshua Charles	G06F16/24578	1/1
*	B	US-10425501-B2	09-2019	Nasson; Randy	G06F9/45529	1/1
*	C	US-20170371975-A1	12-2017	Chen; Hsiao-Yung	G06F40/14	1/1
*	D	US-10037318-B2	07-2018	Liu; Agnes	G06F3/0481	1/1
*	E	US-8855423-B2	10-2014	Boncyk; Wayne C.	A63F13/20	382/181
*	F	US-8452600-B2	05-2013	Fleizach; Christopher B.	G10L13/00	704/260
*	G	US-7653544-B2	01-2010	Bradley; Nathan T.	G06F3/0484	704/270.1
*	H	US-20130073949-A1	03-2013	Barrell; Dylan	G06F16/9577	715/234
*	I	US-20150106686-A1	04-2015	Blitzstein; Jared	G06F16/958	715/234
*	J	US-20130104029-A1	04-2013	Hendry; Robert Olson	G06F40/154	715/234
*	K	US-7546531-B2	06-2009	Celik; Tantek	G06F16/958	715/273
*	L	US-20020122053-A1	09-2002	Dutta, Rabindranath	G09B21/007	715/729
*	M	US-20100070872-A1	03-2010	Trujillo; Damian	G06F3/04892	715/745


FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	


*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Issue Classification 	Application/Control No. 16/430,210	Applicant(s)/Patent Under Reexamination Bradley et al.
	Examiner MOHAMMED H ZUBERI	Art Unit 2177

CPC					
Symbol				Type	Version
G06F		3	04817	F	2013-01-01
G10L	/	15	22	I	2013-01-01
G06F	/	40	14	I	2020-01-01
G06F	/	40	117	I	2020-01-01
G06F	/	3	0482	I	2013-01-01
G06F	/	3	167	I	2013-01-01
G10L	/	13	027	I	2013-01-01
G10L	/	15	26	I	2013-01-01
H04L	/	67	22	I	2013-01-01
G10L	/	2015	221	A	2013-01-01
G10L	/	13	02	A	2013-01-01
H04L	/	67	02	A	2013-01-01

CPC Combination Sets				
Symbol	Type	Set	Ranking	Version

NONE			Total Claims Allowed:	
(Assistant Examiner)	(Date)			20
/MOHAMMED H ZUBERI/ Primary Examiner, Art Unit 2177	31 December 2020	O.G. Print Claim(s)	O.G. Print Figure	
(Primary Examiner)	(Date)	1	13	


Issue Classification 	Application/Control No. 16/430,210	Applicant(s)/Patent Under Reexamination Bradley et al.
	Examiner MOHAMMED H ZUBERI	Art Unit 2177

INTERNATIONAL CLASSIFICATION			
CLAIMED			
G06F		17	00
NON-CLAIMED			

US ORIGINAL CLASSIFICATION	
CLASS	SUBCLASS

CROSS REFERENCES(S)					
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				


NONE			Total Claims Allowed:	
(Assistant Examiner)	(Date)			20
/MOHAMMED H ZUBERI/ Primary Examiner, Art Unit 2177	31 December 2020	O.G. Print Claim(s)	O.G. Print Figure	
(Primary Examiner)	(Date)	1	13	

Issue Classification 	Application/Control No. 16/430,210	Applicant(s)/Patent Under Reexamination Bradley et al.
	Examiner MOHAMMED H ZUBERI	Art Unit 2177

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIMS															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	11	10	45	19	54										
2	12	11	46	20	55										
3	13	12	47												
4	14	13	48												
5	15	14	49												
6	16	15	50												
7	42	16	51												
8	43	17	52												
9	44	18	53												

NONE	Total Claims Allowed:	
(Assistant Examiner)	(Date)	20
/MOHAMMED H ZUBERI/ Primary Examiner, Art Unit 2177	31 December 2020	O.G. Print Claim(s)
(Primary Examiner)	(Date)	1
		O.G. Print Figure
		13

<i>Search Notes</i> 	Application/Control No. 16/430,210	Applicant(s)/Patent Under Reexamination Bradley et al.
	Examiner MOHAMMED H ZUBERI	Art Unit 2177


CPC - Searched*		
Symbol	Date	Examiner
H04L67/02	12/31/2020	/MZ/
G10L2015/221	12/31/2020	/MZ/
G10L13/02	12/31/2020	/MZ/
H04L67/22	12/31/2020	/MZ/
G10L15/26	12/31/2020	/MZ/
G10L15/22	12/31/2020	/MZ/
G10L13/027	12/31/2020	/MZ/
G06F3/167	12/31/2020	/MZ/
G06F3/0482	12/31/2020	/MZ/
G06F40/14	12/31/2020	/MZ/
G06F40/117	12/31/2020	/MZ/
G06F3/04817	12/31/2020	/MZ/

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

/MOHAMMED H ZUBERI/ Primary Examiner, Art Unit 2177	
--	--

<i>Search Notes</i> 	Application/Control No. 16/430,210	Applicant(s)/Patent Under Reexamination Bradley et al.
	Examiner MOHAMMED H ZUBERI	Art Unit 2177

Search Notes		
Search Notes	Date	Examiner
EAST	12/31/2020	/MZ/
CPC Classification + Keyword Search (EAST)	12/31/2020	/MZ/
Google Scholar	12/31/2020	/MZ/

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
	EAST	12/31/2020	/MZ/

/MOHAMMED H ZUBERI/ Primary Examiner, Art Unit 2177	
--	--

EAST Search History

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L16	8	(DOM AND compliance AND remediation AND code AND attribute).clm.	USPAT	OR	OFF	2020/12/31 22:08

12/31/2020 10:08:47 PM

C:\Users\mzuberi\Documents\EAST\Workspaces\16430210.wsp

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	13	(US-20100070872-\$ or US-20130073949-\$ or US-20080133500-\$ or US-20150106686-\$ or US-20020122053-\$ or US-20130104029-\$ or US-20170371975-\$).did. or (US-7546531-\$ or US-7653544-\$ or US-10425501-\$ or US-8452600-\$ or US-8855423-\$ or US-10037318-\$).did.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:47
L2	157	("0000021" "0002002" "10037318" "10425501" "10762280" "10809877" "16991304" "16991329" "16991346" "16991381" "16991392" "16991416" "16991434" "16991505" "16991578" "16991584" "16991671" "20120254723" "20130104029" "20150106686" "20170371975" "20190340212" "7546531" "8452600" "8855423" "10015226" "10049089" "10116726" "10120847" "10198414" "10218775" "10282401" "10423709" "10444934" "10452730" "20110307855" "20130305139" "20140149447" "20150319215" "20170111431" "20180191764" "20180239516" "20180349004" "8321502" "8516362" "8527862" "8589484" "8650474" "8868638" "8983935" "8984164" "9275023" "9311281" "9319244" "9473592" "9547633" "9736245" "9846572" "9846686" "9876776" "9971636" "0000028" "10237334" "20020010715" "20030098803" "20040006478" "20040056882" "20050160065" "20110231192" "20120239405" "20120240045" "20120322384" "20140047337" "20150113410" "6751544" "7028306" "7181692" "7194411" "7246058" "7702674" "8055713" "8090800" "8103956" "8260616" "8694319" "8874716" "8903978" "9208783" "9268753" "9356574" "9384183" "9727660" "9736524" "9875671" "9996613" "20020003547" "20020007379" "20020010584" "20020023020" "20020065658"	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:53

		"20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030115546" "20030158737" "20040002808" "20040128136" "20040158429" "20040218451" "20040254988" "20050033577" "20050108338" "20050195077" "20060212466" "20070100628" "20080133500" "20080168049" "20080208593" "20100070872" "20100095210" "20100114578" "20100166165" "20100251217" "20110154212" "20110295623" "20110307259" "20120116872" "20120176509" "20120216243" "20120278714" "20130073949" "20140164352" "20140195653" "20150156084" "20160306784" "20160337426" "20170269816" "20180075488" "20180165258" "20180267847" "20180293504" "20180336356" "20190075081" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6934684" "6940953" "6959279" "7106220" "7124366" "7174293" "7194752" "7219136" "7284255" "7653544" "7810087" "7966184" "8046229" "8296150" "8589169").PN.				
L3	17,475	G06F3/04817.cpc.	US- PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L4	2,431	G06F40/117.cpc.	US- PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L5	10,286	G06F40/14.cpc.	US- PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L6	40,745	G06F3/0482.cpc.	US- PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L7	12,715	G06F3/167.cpc.	US- PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L8	653	G10L13/027.cpc.	US- PGPUB; USPAT	OR	OFF	2020/12/31 21:57

L9	13,796	G10L15/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L10	11,765	G10L15/26.cpc.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L11	19,271	H04L67/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L12	105,920	L3 L4 L5 L6 L7 L8 L9 L10 L11	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:57
L13	840	G10L13/02.cpc.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:58
L14	663	G10L2015/221.cpc.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:58
L15	42,598	H04L67/02.cpc.	US-PGPUB; USPAT	OR	OFF	2020/12/31 21:58
S1	56	("20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030115546" "20030158737" "20040002808" "20040128136" "20040218451" "20050033577" "20050108338" "20050195077" "20060212466" "20070100628" "20070208687" "20080168049" "20080208593" "20100095210" "20100114578" "20100166165" "20110154212" "20110295623" "20110307259" "20120116872" "20120176509" "20120278714" "20140164352" "20140195653" "20140365860" "20180075488" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6934684" "6940953" "6959279" "7106220" "7124366" "7174293" "7194752" "7219136" "7284255" "8296150" "8589169").PN.	US-PGPUB; USPAT	OR	OFF	2019/05/27 12:50
S2	14	("1458787" "1458790" "14587902" "1458791" "20120239405" "20120240045" "20140047337"	US-PGPUB; USPAT	OR	OFF	2019/05/27 12:50

		"20150113410" "7653544" "7966184" "8046229" "8260616" "8296150" "8589169" "8903978").PN.				
S3	68	S1 S2	US-PGPUB; USPAT	OR	OFF	2019/05/27 12:53
S4	7	S3 AND (script SAME audio SAME element)	US-PGPUB; USPAT	OR	ON	2019/05/27 12:54
S5	311,086	"3" AND (impaired)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:17
S6	56	("20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030115546" "20030158737" "20040002808" "20040128136" "20040218451" "20050033577" "20050108338" "20050195077" "20060212466" "20070100628" "20070208687" "20080168049" "20080208593" "20100095210" "20100114578" "20100166165" "20110154212" "20110295623" "20110307259" "20120116872" "20120176509" "20120278714" "20140164352" "20140195653" "20140365860" "20180075488" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6934684" "6940953" "6959279" "7106220" "7124366" "7174293" "7194752" "7219136" "7284255" "8296150" "8589169").PN.	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:17
S7	14	("1458787" "1458790" "14587902" "1458791" "20120239405" "20120240045" "20140047337" "20150113410" "7653544" "7966184" "8046229" "8260616" "8296150" "8589169" "8903978").PN.	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:17
S8	68	S6 S7	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:17
S9	26	S8 AND (impaired)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:17

S10	7	S9 AND audio WITH element	US-PGPUB; USPAT	OR	ON	2019/05/27 18:18
S11	5	S8 AND (alt\$8 ADJ text)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:19
S12	5	S8 AND ((alt alternate alternative) ADJ text)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:19
S13	0	S8 AND ((alt alternate alternative) ADJ text) WITH (untagged tagless)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:20
S14	0	S8 AND ((alt alternate alternative) ADJ text) SAME (untagged tagless)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:20
S15	3	S8 AND ((alt alternate alternative) ADJ text) SAME (untagged tagless tag tagged)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:20
S16	0	((alt alternate alternative) ADJ text) WITH untagged	US-PGPUB; USPAT	OR	ON	2019/05/27 18:21
S17	0	((alt alternate alternative) ADJ text) SAME untagged	US-PGPUB; USPAT	OR	ON	2019/05/27 18:21
S18	299	((alt alternate alternative) ADJ text) WITH (tag tagged)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:21
S19	63	S18 AND accessibility	US-PGPUB; USPAT	OR	ON	2019/05/27 18:21
S20	2	S19 AND script SAME (alt alternate alternative) ADJ text)	US-PGPUB; USPAT	OR	ON	2019/05/27 18:22
S21	0	"like us on"	US-PGPUB; USPAT	OR	ON	2019/05/27 18:24
S22	2	web ADJ accessibility AND compliance AND (add adding assign assigning) NEAR3 (alt alternate alternative) ADJ text)	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:28
S23	2	web ADJ accessibility AND (add adding assign assigning) NEAR3 (alt alternate alternative) ADJ text)	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:30
S24	422	web ADJ accessibility AND (audio description)	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:32

S25	4	web ADJ accessibility AND (audio ADJ description)	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:32
S26	7	S8 AND (audio ADJ description)	US-PGPUB; USPAT	OR	OFF	2019/05/27 18:42
S27	4	US-20140164352-\$.did. US-20080168049-\$.did. US-20120176509-\$.did. US-20020010715-\$.did.	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:05
S28	7	US-20140164352-\$.did. US-20080168049-\$.did. US-20120176509-\$.did. US-20020010715-\$.did. US-20180075488-\$.did. US-20110154212-\$.did. US-20120116872-\$.did.	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:07
S29	9	US-20140164352-\$.did. US-20080168049-\$.did. US-20120176509-\$.did. US-20020010715-\$.did. US-20180075488-\$.did. US-20110154212-\$.did. US-20120116872-\$.did. US-20110154212-\$.did. US-20140195653-\$.did. US-20020194388-\$.did.	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:08
S30	1	crawl\$4 WITH (DOM page) AND ((accessibility ADJ standards) (compliance ADJ issues)) AND audio WITH element	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:20
S31	3	crawl\$4 WITH (DOM page) AND ((accessibility ADJ standards) (compliance ADJ issues)) AND audio WITH (page description)	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:21
S32	56	("20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030115546" "20030158737" "20040002808" "20040128136" "20040218451" "20050033577" "20050108338" "20050195077" "20060212466" "20070100628" "20070208687" "20080168049" "20080208593" "20100095210" "20100114578" "20100166165" "20110154212" "20110295623" "20110307259" "20120116872" "20120176509" "20120278714" "20140164352" "20140195653" "20140365860" "20180075488" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6934684" "6940953" "6959279"	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:36

		"7106220" "7124366" "7174293" "7194752" "7219136" "7284255" "8296150" "8589169").PN.				
S33	14	("1458787" "1458790" "14587902" "1458791" "20120239405" "20120240045" "20140047337" "20150113410" "7653544" "7966184" "8046229" "8260616" "8296150" "8589169" "8903978").PN.	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:36
S34	68	S32 S33	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:36
S35	5	hearing ADJ impaired AND S34	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:36
S36	27	(blind (visually ADJ impaired)) AND S34	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:39
S37	11	S36 AND (add adding including include) NEAR3 audio	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:43
S38	2	S37 AND DOM	US-PGPUB; USPAT	OR	OFF	2019/05/28 12:43
S39	749	sensus	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:15
S40	4	sensus AND visually ADJ impaired	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:15
S41	7	audioeye.as.	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:17
S42	1	S40 AND S34	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:17
S43	38,119	G06F3/04817.cpc. G06F3/0482.cpc. G06F3/157.cpc.	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:30
S44	44,240	G06F3/04817.cpc. G06F3/0482.cpc. G06F3/167.cpc.	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:30
S45	8,317	G10L13/027.cpc. G10L15/26.cpc.	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:32

S46	51,623	S43 S44 S45	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:33
S47	908	S46 AND (add adding embed embedding) NEAR3 (audio)	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:33
S48	64	S47 AND (accessibility)	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:33
S49	11	S48 AND (remediation resolve resolution resolving remediate remediating)	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:34
S50	14	audioeye	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:34
S51	77	(modify modification update adjust fix updating resolving remediate remediation adjusting modifying) NEAR3 (web\$page) SAME (accessible accessibility impaired)	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:49
S52	6	S51 AND impaired	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:49
S53	0	S52 AND S34	US-PGPUB; USPAT	OR	OFF	2019/05/28 13:52
S55	1	US-10444934-\$.did.	US-PGPUB; USPAT	OR	OFF	2020/09/02 13:58
S56	5,706	G06F40/103.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/03 11:14
S57	27,908	G06F3/04842.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/03 11:16
S58	6	(web\$page web\$site) WITH audio ADJ description	US-PGPUB; USPAT	OR	OFF	2020/09/03 14:30
S59	1	US-10444934-\$.did.	US-PGPUB; USPAT	OR	OFF	2020/09/10 00:00
S60	0	accessibility ADJ standards AND (audible ADJ description) WITH field	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:46
S61	0	accessibility ADJ standards AND (descriptive ADJ attribute) WITH field	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:47

S62	5	(US-20100070872-\$ or US-20130073949-\$ or US-20080133500-\$).did. or (US-7546531-\$ or US-7653544-\$).did.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:47
S63	6,608	G06F3/04817.cpc.	USPAT	OR	OFF	2020/09/10 10:47
S64	1,033	G06F40/117.cpc.	USPAT	OR	OFF	2020/09/10 10:47
S65	16,769	G06F3/04817.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:47
S66	2,357	G06F40/117.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:47
S67	10,055	G06F40/14.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:48
S68	38,893	G06F3/0482.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:48
S69	11,647	G06F3/167.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:48
S70	607	G10L13/027.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:48
S71	12,714	G10L15/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:48
S72	11,200	G10L15/26.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:48
S73	18,429	H04L67/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:49
S74	123	compliance SAME field WITH attribute	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:50
S75	629	ADA ADJ complian\$2	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:52
S76	7	S75 AND (speech audio audible speak speakable voice) WITH field	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:53
S77	5	ADA ADJ complian\$2 WITH (web\$page web\$site)	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:54

S78	233	WCAG	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:54
S79	101,015	S63 S64 S65 S66 S67 S68 S69 S70 S71 S72 S73	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:54
S80	57	S78 AND S79	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:54
S81	11	S80 AND (speech audio audible speak speakable voice) WITH field	US-PGPUB; USPAT	OR	OFF	2020/09/10 10:55
S82	5	("2015/0106686").URPN.	USPAT	OR	OFF	2020/09/10 13:24
S83	449	generat\$3 NEAR3 (audible audio spoken) NEAR4 field	USPAT	OR	OFF	2020/09/10 14:37
S84	922	generat\$3 NEAR3 (audible audio spoken) NEAR4 field	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:37
S85	6,608	G06F3/04817.cpc.	USPAT	OR	OFF	2020/09/10 14:39
S86	1,033	G06F40/117.cpc.	USPAT	OR	OFF	2020/09/10 14:39
S87	16,769	G06F3/04817.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S88	2,357	G06F40/117.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S89	10,055	G06F40/14.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S90	38,893	G06F3/0482.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S91	11,647	G06F3/167.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S92	607	G10L13/027.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S93	12,714	G10L15/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39

S94	11,200	G10L15/26.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S95	18,429	H04L67/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S96	101,015	S85 S86 S87 S88 S89 S90 S91 S92 S93 S94 S95	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S97	78	S96 AND S84	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:39
S98	629	ADA ADJ complian\$2	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:40
S99	233	WCAG	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:40
S100	862	S98 S99	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:40
S101	0	S97 AND S100	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:40
S102	1	US-20020122053-\$.did.	US-PGPUB; USPAT	OR	OFF	2020/09/10 14:42
S103	149	("2002/0122053").URPN.	USPAT	OR	OFF	2020/09/10 15:02
S104	291	((visually ADJ impaired) blind) WITH (web\$navigation web\$page web\$site)	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:18
S105	0	S84 AND S104	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:19
S106	49	S104 AND S96	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:19
S107	1,657	screen ADJ reader	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:29
S108	387	S96 AND S107	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:30

S109	0	S84 AND S108	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:30
S110	79	(voice\$over talk\$back microsoft ADJ narrator emacspeak yasr speakup) AND S107	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:37
S111	24	screen ADJ reader WITH web\$page	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:46
S112	0	brousealoud	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:54
S113	0	Brousealoud	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:54
S114	11	texthelp	US-PGPUB; USPAT	OR	OFF	2020/09/10 15:55
S115	146	("16991304" "16991329" "16991381" "16991392" "16991405" "16991416" "16991434" "16991505" "16991578" "16991584" "16991671" "10015226" "10049089" "10116726" "10120847" "10198414" "10218775" "10237334" "10282401" "10423709" "10444934" "10452730" "20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030098803" "20030115546" "20030158737" "20040002808" "20040006478" "20040056882" "20040128136" "20040158429" "20040218451" "20040254988" "20050033577" "20050108338" "20050160065" "20050195077" "20060212466" "20070100628" "20070208687" "20080133500" "20080168049" "20080208593" "20080235564" "20100070872" "20100095210" "20100114578" "20100166165" "20100251217" "20110154212" "20110231192" "20110295623" "20110307259" "20110307855" "20120116872" "20120176509" "20120216243" "20120239405" "20120240045" "20120278714" "20120322384"	US-PGPUB; USPAT	OR	OFF	2020/09/11 17:40

		"20130073949" "20130305139" "20140047337" "20140149447" "20140164352" "20140195653" "20140365860" "20150113410" "20150156084" "20150319215" "20160306784" "20160337426" "20170111431" "20170269816" "20180075488" "20180165258" "20180191764" "20180239516" "20180267847" "20180293504" "20180336356" "20180349004" "20190075081" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6751544" "6934684" "6940953" "6959279" "7028306" "7106220" "7124366" "7174293" "7181692" "7194411" "7194752" "7219136" "7246058" "7284255" "7653544" "7702674" "7810087" "7966184" "8046229" "8055713" "8090800" "8103956" "8260616" "8296150" "8321502" "8516362" "8527862" "8589169" "8589484" "8650474" "8694319" "8868638" "8874716" "8903978" "8983935" "8984164" "8990392" "9208783" "9268753" "9275023" "9311281" "9319244" "9356574" "9384183" "9473592" "9547633" "9727660" "9736245" "9736524" "9846572" "9846686" "9875671" "9876776" "9971636" "9996613").PN.				
S117	146	("16991304" "16991346" "16991381" "16991392" "16991405" "16991416" "16991434" "16991505" "16991578" "16991584" "16991671" "10015226" "10049089" "10116726" "10120847" "10198414" "10218775" "10237334" "10282401" "10423709" "10444934" "10452730" "20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030098803" "20030115546" "20030158737" "20040002808" "20040006478" "20040056882" "20040128136" "20040158429" "20040218451" "20040254988" "20050033577" "20050108338"	US- PGPUB; USPAT	OR	OFF	2020/09/13 12:00

		"20050160065" "20050195077" "20060212466" "20070100628" "20070208687" "20080133500" "20080168049" "20080208593" "20080235564" "20100070872" "20100095210" "20100114578" "20100166165" "20100251217" "20110154212" "20110231192" "20110295623" "20110307259" "20110307855" "20120116872" "20120176509" "20120216243" "20120239405" "20120240045" "20120278714" "20120322384" "20130073949" "20130305139" "20140047337" "20140149447" "20140164352" "20140195653" "20140365860" "20150113410" "20150156084" "20150319215" "20160306784" "20160337426" "20170111431" "20170269816" "20180075488" "20180165258" "20180191764" "20180239516" "20180267847" "20180293504" "20180336356" "20180349004" "20190075081" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6751544" "6934684" "6940953" "6959279" "7028306" "7106220" "7124366" "7174293" "7181692" "7194411" "7194752" "7219136" "7246058" "7284255" "7653544" "7702674" "7810087" "7966184" "8046229" "8055713" "8090800" "8103956" "8260616" "8296150" "8321502" "8516362" "8527862" "8589169" "8589484" "8650474" "8694319" "8868638" "8874716" "8903978" "8983935" "8984164" "8990392" "9208783" "9268753" "9275023" "9311281" "9319244" "9356574" "9384183" "9473592" "9547633" "9727660" "9736245" "9736524" "9846572" "9846686" "9875671" "9876776" "9971636" "9996613").PN.				
S118	233	WCAG	US- PGPUB; USPAT	OR	OFF	2020/09/13 12:12
S119	629	ADA ADJ complian\$2	US- PGPUB; USPAT	OR	OFF	2020/09/13 12:12

S120	862	S118 S119	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:12
S121	21	S120 AND (scan scanning recognition recognize) NEAR4 (image graphic)	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:15
S122	33,051	(document page web\$page web\$site) WITH (scan scanning recognition recognize) NEAR4 (image graphic)	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:16
S123	955	(document page web\$page web\$site) WITH (scan scanning recognition) NEAR4 (image graphic) WITH (content subject ADJ matter)	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:18
S124	16,781	G06F3/04817.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S125	2,357	G06F40/117.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S126	10,058	G06F40/14.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S127	38,916	G06F3/0482.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S128	11,683	G06F3/167.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S129	609	G10L13/027.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S130	12,722	G10L15/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S131	11,206	G10L15/26.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S132	18,453	H04L67/22.cpc.	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S133	101,080	S124 S125 S126 S127 S128 S129 S130 S131 S132	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S134	95	S123 AND S133	US-PGPUB; USPAT	OR	OFF	2020/09/13 12:19

S135	0	S134 AND screen\$reader	US- PGPUB; USPAT	OR	OFF	2020/09/13 12:19
S137	146	("16991304" "16991329" "16991346" "16991381" "16991392" "16991405" "16991434" "16991505" "16991578" "16991584" "16991671" "10015226" "10049089" "10116726" "10120847" "10198414" "10218775" "10237334" "10282401" "10423709" "10444934" "10452730" "20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030098803" "20030115546" "20030158737" "20040002808" "20040006478" "20040056882" "20040128136" "20040158429" "20040218451" "20040254988" "20050033577" "20050108338" "20050160065" "20050195077" "20060212466" "20070100628" "20070208687" "20080133500" "20080168049" "20080208593" "20080235564" "20100070872" "20100095210" "20100114578" "20100166165" "20100251217" "20110154212" "20110231192" "20110295623" "20110307259" "20110307855" "20120116872" "20120176509" "20120216243" "20120239405" "20120240045" "20120278714" "20120322384" "20130073949" "20130305139" "20140047337" "20140149447" "20140164352" "20140195653" "20140365860" "20150113410" "20150156084" "20150319215" "20160306784" "20160337426" "20170111431" "20170269816" "20180075488" "20180165258" "20180191764" "20180239516" "20180267847" "20180293504" "20180336356" "20180349004" "20190075081" "5918223" "5930752" "5991781" "6092039" "6240448" "6606374" "6665642" "6742021" "6751544" "6934684" "6940953" "6959279" "7028306" "7106220" "7124366" "7174293" "7181692")	US- PGPUB; USPAT	OR	OFF	2020/09/24 14:56

		"7194411" "7194752" "7219136" "7246058" "7284255" "7653544" "7702674" "7810087" "7966184" "8046229" "8055713" "8090800" "8103956" "8260616" "8296150" "8321502" "8516362" "8527862" "8589169" "8589484" "8650474" "8694319" "8868638" "8874716" "8903978" "8983935" "8984164" "8990392" "9208783" "9268753" "9275023" "9311281" "9319244" "9356574" "9384183" "9473592" "9547633" "9727660" "9736245" "9736524" "9846572" "9846686" "9875671" "9876776" "9971636" "9996613").PN.				
S138	16,860	G06F3/04817.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S139	2,375	G06F40/117.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S140	10,091	G06F40/14.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S141	39,139	G06F3/0482.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S142	11,779	G06F3/167.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S143	614	G10L13/027.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S144	12,853	G10L15/22.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S145	11,279	G10L15/26.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S146	18,546	H04L67/22.cpc.	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02
S147	101,671	S138 S139 S140 S141 S142 S143 S144 S145 S146	US- PGPUB; USPAT	OR	OFF	2020/09/24 15:02

S148	234	WCAG	US-PGPUB; USPAT	OR	OFF	2020/09/24 15:13
S149	632	ADA ADJ complian\$2	US-PGPUB; USPAT	OR	OFF	2020/09/24 15:13
S150	866	S148 S149	US-PGPUB; USPAT	OR	OFF	2020/09/24 15:13
S151	21	S150 AND (scan scanning recognition recognize) NEAR4 (image graphic)	US-PGPUB; USPAT	OR	OFF	2020/09/24 15:13
S153	13	(US-20100070872-\$ or US-20130073949-\$ or US-20080133500-\$ or US-20150106686-\$ or US-20020122053-\$ or US-20130104029-\$ or US-20170371975-\$).did. or (US-7546531-\$ or US-7653544-\$ or US-10425501-\$ or US-8452600-\$ or US-8855423-\$ or US-10037318-\$).did.	US-PGPUB; USPAT	OR	OFF	2020/10/08 22:10
S154	146	("16991329" "16991346" "16991381" "16991392" "16991405" "16991416" "16991434" "16991505" "16991578" "16991584" "16991671" "10015226" "10049089" "10116726" "10120847" "10198414" "10218775" "10237334" "10282401" "10423709" "10444934" "10452730" "20020003547" "20020007379" "20020010584" "20020010715" "20020023020" "20020065658" "20020074396" "20020124020" "20020194388" "20020198705" "20030033434" "20030098803" "20030115546" "20030158737" "20040002808" "20040006478" "20040056882" "20040128136" "20040158429" "20040218451" "20040254988" "20050033577" "20050108338" "20050160065" "20050195077" "20060212466" "20070100628" "20070208687" "20080133500" "20080168049" "20080208593" "20080235564" "20100070872" "20100095210" "20100114578" "20100166165" "20100251217" "20110154212" "20110231192" "20110295623" "20110307259" "20110307855" "20120116872" "20120176509" "20120216243" "20120239405" "20120240045" "20120278714" "20120322384"	US-PGPUB; USPAT	OR	OFF	2020/10/09 23:02

"20130073949" "20130305139"				
"20140047337" "20140149447"				
"20140164352" "20140195653"				
"20140365860" "20150113410"				
"20150156084" "20150319215"				
"20160306784" "20160337426"				
"20170111431" "20170269816"				
"20180075488" "20180165258"				
"20180191764" "20180239516"				
"20180267847" "20180293504"				
"20180336356" "20180349004"				
"20190075081" "5918223" "5930752"				
"5991781" "6092039" "6240448"				
"6606374" "6665642" "6742021"				
"6751544" "6934684" "6940953"				
"6959279" "7028306" "7106220"				
"7124366" "7174293" "7181692"				
"7194411" "7194752" "7219136"				
"7246058" "7284255" "7653544"				
"7702674" "7810087" "7966184"				
"8046229" "8055713" "8090800"				
"8103956" "8260616" "8296150"				
"8321502" "8516362" "8527862"				
"8589169" "8589484" "8650474"				
"8694319" "8868638" "8874716"				
"8903978" "8983935" "8984164"				
"8990392" "9208783" "9268753"				
"9275023" "9311281" "9319244"				
"9356574" "9384183" "9473592"				
"9547633" "9727660" "9736245"				
"9736524" "9846572" "9846686"				
"9875671" "9876776" "9971636"				
"9996613").PN.				

12/31/2020 10:08:09 PM
C:\Users\mzuberi\Documents\EAST\Workspaces\16430210.wsp

Bibliographic Data

Application No: 16/430,210

Foreign Priority claimed: Yes No

35 USC 119 (a-d) conditions met: Yes No Met After Allowance

Verified and Acknowledged: /MOHAMMED H ZUBERI/

Examiner's Signature

Initials

Title:

Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
06/03/2019	715	2177	AUDEY.003C1
RULE			

APPLICANTS

AudioEye, Inc., Tucson, AZ,

INVENTORS

Sean D. Bradley, Tucson, AZ, UNITED STATES

Mark D. Baker, Marietta, GA, UNITED STATES

Jeffrey O. Jones, Mountain Park, GA, UNITED STATES

Kenny P. Hefner, Buchanan, GA, UNITED STATES

Adam Finkelstein, Alpharetta, GA, UNITED STATES

Douglas J. Gilormo, Cumming, GA, UNITED STATES

Taylor R. Bodnar, Tucson, AZ, UNITED STATES

David C. Pinckney, Roswell, GA, UNITED STATES

Charlie E. Blevins, Atlanta, GA, UNITED STATES

Trevor C. Jones, Kennesaw, GA, UNITED STATES

Helena Laymon, Duluth, GA, UNITED STATES

CONTINUING DATA

This application is a CON of 15074818 03/18/2016 PAT 10444934

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

06/10/2019

STATE OR COUNTRY

UNITED STATES

ADDRESS

KNOBBE MARTENS OLSON & BEAR LLP

2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614
UNITED STATES

FILING FEE RECEIVED

\$1,880

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2179	
<i>(Multiple sheets used when necessary)</i>		Examiner	ZUBERI, MOHAMMED H
SHEET 1 OF 3		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,751,544	06-15-2004	Hashimoto et al.	
	2	7,028,306	04-11-2006	Boloker et al.	
	3	7,181,692	02-20-2007	Siegel	
	4	7,194,411	03-20-2007	Slotznick et al.	
	5	7,246,058	07-17-2007	Burnett	
	6	7,702,674	04-20-2010	Hyder et al.	
	7	8,055,713	11-08-2011	Simske et al.	
	8	8,090,800	01-03-2012	Yee	
	9	8,103,956	01-24-2012	Trujillo	
	10	8,260,616	09-04-2012	O'Connor et al.	
	11	8,694,319	04-08-2014	Bodin et al.	
	12	8,874,716	08-22-2000	Daiwa Seiko Inc.	
	13	8,903,978	12-02-2014	Zerr et al.	
	14	9,208,783	12-08-2015	Ativanichayaphong et al.	
	15	9,268,753	02-23-2016	Hendry et al.	
	16	9,356,574	05-31-2016	Denninghoff	
	17	9,384,183	07-05-2016	Kasa et al.	
	18	9,727,660	08-08-2017	Barrell et al.	
	19	9,736,524	08-15-2017	Aravamudan et al.	
	20	9,875,671	01-23-2018	Gharpure et al.	
	21	9,996,613	06-12-2018	Jadhav et al.	
	22	10,237,334	03-19-2019	Alexander et al.	
	23	2002/0010715	01-24-2002	Chinn et al.	
	24	2003/0098803	05-29-2003	Gourgey et al.	
	25	2004/0006478	01-08-2004	Alpdemir et al.	
	26	2004/0056882	03-25-2004	Foreman et al.	
	27	2005/0160065	07-21-2005	Seeman	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2179	
<i>(Multiple sheets used when necessary)</i>		Examiner	ZUBERI, MOHAMMED H
SHEET 2 OF 3		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number <i>Number - Kind Code (if known)</i> Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	28	2008/0235564	09-25-2008	Erol et al.	
	29	2011/0231192	09-22-2011	O'Connor et al.	
	30	2012/0239405	09-20-2012	O'Connor et al.	
	31	2012/0240045	09-20-2012	Bradley et al.	
	32	2012/0322384	12-20-2012	Zerr et al.	
	33	2014/0047337	02-13-2014	Bradley et al.	
	34	2015/0113410	04-23-2015	Bradley et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document <i>Country Code-Number-Kind Code</i> Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	35	WO 2006/065877	06-22-2006	FREEDOM SCIENTIFIC INC		
	36	WO 2013/184273	12-12-2013	FREEDOM SCIENTIFIC INC		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	37	European Search Report and Written Opinion of EP 17767449.6 dated October 15, 2019 in 10 pages	
	38	International Search Report and Written Opinion of PCT/US2019/046361 dated December 5, 2019 in 17 pages	
	39	SCHIAVONE et al., "An extensible environment for guideline-based accessibility evaluation of dynamic Web applications", Univ Access Inf Soc., January 2015, 22 pages	
	40	BIGHAM et al., "WebInSight: Making Web Images Accessible", Proceedings of the 8th International ACM SIG ACCESS Conference on Computers and Accessibility, 2006, pages 181-188	
	41	BRUNET, et al., "Accessibility requirements for systems design to accommodate users with vision impairments", IBM Systems Journal, 2005, 44(3): 445-466	
	42	IACCARINO et al., "Personalizable Edge Services for Web Accessibility", Univ Access Inf Soc 6, 2007, pages 285-306 https://doi.org/10.1007/s10209-007-0091-y	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2179
<i>(Multiple sheets used when necessary)</i>	Examiner	ZUBERI, MOHAMMED H
SHEET 3 OF 3	Attorney Docket No.	AUDEY.003C1

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	43	MAEDA et al., "Web accessibility technology at the IBM Tokyo Research Laboratory", IBM Journal of Research and Development, Nov. 2004, 48(5/6): 735-749 http://dialog.proquest.com/professional/docview/220685504?accountid=157282	

32463761

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Inventor	:	Sean D. Bradley
App. No.	:	16/430210
Filed	:	June 3, 2019
For	:	MODULAR SYSTEMS AND METHODS FOR SELECTIVELY ENABLING CLOUD-BASED ASSISTIVE TECHNOLOGIES
Examiner	:	Unassigned
Art Unit	:	2143
Conf. No.	:	6600

Application No.: 16/430210
Filing Date: June 3, 2019

AMENDMENTS TO THE CLAIMS

1-10 (Cancelled)

11. (Amended) A non-transitory electronic storage medium with computer code stored thereon, in a non-transient medium the computer code configured to programmatically assign a descriptive attribute to an element on a web page to provide an audible description of the element, the web page having an associated document object model (DOM), and the computer code configured to perform, when executed by a computer processor, the steps of:

accessing code associated with the web page, the code comprising HTML or the DOM;

retrieving at least one remediation code from a remote server, the at least one remediation code configured to associate a descriptive attribute to an element in the web page, the element in the web page lacking an adequate descriptive attribute;

amending the web page, based on applying the retrieved at least one remediation code, to assign the descriptive attribute to the element in the web page, the descriptive attribute configured to enable an assistive technology to speak the descriptive attribute to a user;

superimposing amending the web page to display a timed-toggle-able of a branding icon over a universal accessibility icon on a the web page; and

revealing displaying, in response to input from the a-user command to the displayed toggle-able universal accessibility icon, an assistive technology tool bar including user controls for independently enabling remediating the web page at least an audio player and a reader tool suite, wherein user controls of the assistive technology tool bar are configured to enable the user to dynamically change on the web page a font type, magnification, and contrast of colors displayed to the user.

12. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the user ~~command~~ input comprises one or more of mouse hovering over and or clicking on the branding icon toggle-able universal accessibility icon.

13. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein independently enabling comprises selecting one of: a first option to enable ~~the~~ an audio player without enabling the ~~reader tool suite~~ assistive technology

Application No.: 16/430210
Filing Date: June 3, 2019

tool bar; a second option to enable the ~~reader tool suite~~ assistive technology tool bar without enabling the audio player; ~~and~~ or a third option to simultaneously enable both the audio player and the ~~reader tool suite~~ assistive technology tool bar.

14. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein ~~superimposing~~ amending the web page to display the timed toggle-able universal accessibility icon comprises using one of a cascading style sheet (CSS) ~~and~~ or JavaScript to periodically animate the ~~branding icon~~ universal accessibility icon.

15. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the assistive technology tool bar further includes a control for independently enabling a help desk function.

16. (Amended) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the computer code comprises JavaScript.

17.-41. (Cancelled)

42. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the computer code comprises one or more program codes.

43. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the amending the web page based on applying the retrieved at least one remediation code comprises amending the DOM or HTML associated with the web page.

44. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the at least one remediation code comprises javascript.

45. (New) The non-transitory electronic storage medium with computer code stored thereon of claim 11, wherein the element lacking an adequate descriptive attribute is a graphic.

46. (New) A computer-implemented method of programmatically assigning a descriptive attribute to an element on a web page to enable an audible description of the element, the web page having an associated document object model (DOM), the computer-implemented method comprising:

accessing code associated with the web page, the code comprising HTML or the DOM;

Application No.: 16/430210
Filing Date: June 3, 2019

retrieving at least one remediation code from a remote server, the at least one remediation code configured to associate a descriptive attribute to an element in the web page, the element in the web page lacking an adequate descriptive attribute;

amending the web page, based on applying the retrieved at least one remediation code to identify the element in the code associated with the web page, to associate the descriptive attribute to the element in the web page, the descriptive attribute configured to enable an assistive technology to speak the descriptive attribute to a user;

amending the web page to display a universal accessibility icon on the web page, the universal accessibility icon configured to receive input from the user; and

displaying, in response to input from the user to the displayed universal accessibility icon, an assistive technology tool bar including user controls for independently enabling remediating the web page,

wherein user controls of the assistive technology tool bar are configured to enable the user to dynamically change on the web page a font type, magnification, and contrast of colors displayed to the user.

47. (New) The computer-implemented method of claim 46, wherein the amending the web page based on applying the retrieved at least one remediation code comprises amending the DOM or HTML associated with the web page.

48. (New) The computer-implemented method of claim 46, wherein the element lacking an adequate descriptive attribute is a graphic.

49. (New) The computer-implemented method of claim 46, wherein the at least one remediation code comprises javascript.

50. (New) The computer-implemented method of claim 46, wherein the element lacking an adequate descriptive attribute is a video.

51. (New) The computer-implemented method of claim 46, wherein the descriptive attribute associated with the element in the web page is an alt-text tag.

52. (New) The computer-implemented method of claim 46, wherein the descriptive attribute is generated by an artificial intelligence algorithm.

53. (New) The computer-implemented method of claim 46, wherein the computer-implemented method is based on executing one or more program codes.

Application No.: 16/430210
Filing Date: June 3, 2019

54. (New) The computer-implemented method of claim 46, wherein the computer-implemented method is based on executing a JavaScript.

55. (New) The computer-implemented method of claim 46, wherein the element lacking an adequate descriptive attribute is an input field.

34140291

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2143
<i>(Multiple sheets used when necessary)</i>	Examiner	Unassigned
SHEET 1 OF 4	Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	5,918,223	06-29-1999	Blum et al.	
	2	5,930,752	07-27-1999	Kawaguchi et al	
	3	5,991,781	11-23-1999	Nielsen	
	4	6,092,039	07-18-2000	Zingher	
	5	6,240,448	05-29-2001	Imielinski et al	
	6	6,606,374	08-12-2003	Rokoff et al.	
	7	6,665,642	12-16-2003	Kanevsky et al.	
	8	6,742,021	05-25-2004	Halverson et al.	
	9	6,940,953	09-06-2005	Eberle et. Al	
	10	6,959,279	10-25-2005	Jackson et al.	
	11	6,934,684	08-23-2005	Alpdemir et al.	
	12	7,106,220	09-12-2006	Gourgey et al.	
	13	7,124,366	10-17-2006	Foreman et al.	
	14	7,174,293	02-06-2007	Kenyon et al.	
	15	7,194,752	03-20-2007	Kenyon et al.	
	16	7,219,136	05-15-5007	Danner et al.	
	17	7,284,255	10-16-2007	Apel et al.	
	18	7,653,544	01-26-2010	Bradley et al.	
	19	7,810,087	10-01-2010	O'Brien	
	20	7,966,184	06-21-2011	Bradley et al.	
	21	8,046,229	10-25-2011	Bradley et al.	
	22	8,296,150 (Our Ref. AUDEY.004A2), including its prosecution history, the cited references, and the Office Actions therein (AUDEY.004A2)	10-23-2012	Bradley et al.	
	23	8,589,169 (Our Ref. AUDEY.004C2) including its prosecution history, the cited references, and the Office Actions therein	11-19-2013	Bradley et al.	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2143
<i>(Multiple sheets used when necessary)</i>	Examiner	Unassigned
SHEET 2 OF 4	Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	24	8,990,392	03-01-2015	Stamos	
	25	2002/0003547	01-10-2002	Wang	
	26	2002/0007379	01-17-2002	Wang	
	27	2002/0010584	01-24-2002	Schultz et al.	
	28	2002/0010715	01-24-2002	Chinn et al.	
	29	2002/0023020	02-21-2002	Kenyon et al.	
	30	2002/0065658	05-30-2002	Kanevsky et al.	
	31	2002/0074396	06-20-2002	Rathus et al.	
	32	2002/0124020	09-05-2002	Janakiraman et al .	
	33	2002/0194388	12-19-2002	Boloker et al.	
	34	2002/0198705	12-26-2002	Burnett	
	35	2003/0033434	02-13-2003	Kavacheri et al.	
	36	2003/0115546	06-19-2003	Dubey et al.	
	37	2003/0158737	08-21-2003	Csicsatka	
	38	2004/0002808	01-01-2004	Hashimoto et al.	
	39	2004/0128136	07-01-2004	Irani	
	40	2004/0158429	08-01-2004	Bary	
	41	2004/0218451	11-04-2004	Said, et al.	
	42	2004/0254988	12-01-2004	Rodriguez	
	43	2005/0033577 (Our Ref. AUDEY.004A1) including its prosecution history, the cited references, and the Office Actions therein	02-10-2005	Bradley et al.	
	44	2005/0108338	05-19-2005	Simske et al.	
	45	2005/0195077	09-08-2005	McCulloch et al.	
	46	2006/0212466	09-21-2006	Hyder et al.	
	47	2007/0100628	05-03-2007	Bodin et al.	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2143
<i>(Multiple sheets used when necessary)</i>	Examiner	Unassigned
SHEET 3 OF 4	Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number <i>Number - Kind Code (if known)</i> Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	48	2007/0208687 (Our Ref. AUDEY.005A) including its prosecution history, the cited references, and the Office Actions therein	09-06-2007	Bradley et al.	
	49	2008/0133500	06-05-2008	Edwards	
	50	2008/0168049	07-10-2008	Gao et al.	
	51	2008/0208593	08-28-2008	Ativanichayaphong et al.	
	52	2010/0070872	03-18-2010	Trujillo	
	53	2010/0251217	09-01-2010	Miller	
	54	2010/0114578	05-06-2010	Yuen et al	
	55	2010/0095210 (Our Ref. AUDEY.004C1) including its prosecution history, the cited references, and the Office Actions therein	04-15-2010	Bradley et al.	
	56	2010/0166165	07-01-2010	Langseth et al.	
	57	2011/0154212	0-23-2011	Gharpure et al.	
	58	2011/0295623	12-01-2011	Behringer et al.	
	59	2011/0307259 (Our Ref. AUDEY.004A2) including its prosecution history, the cited references, and the Office Actions therein	12-15-2011	Bradley et al.	
	60	2012/0116872	05-10-2012	Hicken et al.	
	61	2012/0176509	07-12-2012	Aravamudan et al.	
	62	2012/0216243	08-01-2012	Gill	
	63	2012/0278714 (Our Ref. AUDEY.004C2) including its prosecution history, the cited references, and the Office Actions therein	11-01-2012	Bradley et al.	
	64	2013/0073949	03-21-2013	Barrell	
	65	2014/0164352	06-12-2014	Denninghoff	
	66	2014/0195653	07-10-2014	Alexander et al.	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2143	
<i>(Multiple sheets used when necessary)</i>		Examiner	Unassigned
SHEET 4 OF 4		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	67	2014/0365860	12-11-2014	Spielberg et al.	
	68	2015/0156084	06-01-2015	Kaminsky	
	69	2016/0306784	10-01-2016	Bradley	
	70	2016/0337426	11-17-2016	Shribman	
	71	2017/0269816 (Our Ref. AUDEY.003A) including its prosecution history, the cited references, and the Office Actions therein	09-21-2017	Bradley et al.	
	72	2018/0336356	11-22-2018	Papaxenopoulos	
	73	2018/0075488	03-15-2018	Celik et al.	
	74	2018/0165258	06-01-2018	Scoda	
	75	2018/0267847	09-01-2018	Smith	
	76	2018/0293504	10-01-2018	Meruva	
	77	2019/0075081	03-07-2019	Adam	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	78	WO 2017/161008	09-21-2017	AUDIOEYE, INC.		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	79	Sensus Internet Browser, printed from www/sensus.dk/sibl0uk.htm on Jul. 23, 2002.	
	80	International Search Report and Written Opinion of PCT/US17/22542 dated June 20, 2017 in 10 pages	
	81	International Preliminary Report on Patentability dated January 25, 2019 for International Application No. PCT/US2017/022542 in 36 pages	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2177
<i>(Multiple sheets used when necessary)</i>	Examiner	Zuberi, Mohammed H.
SHEET 1 OF 2	Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	7,546,531	06-09-2009	Celik	
	2	8,452,600	05-28-2013	Fleizach	
	3	8,855,423	10-07-2014	Bonczyk et al.	
	4	10,037,318	07-31-2018	Liu	
	5	10,425,501	06-09-2016	Nasson et al.	
	6	10,762,280 (Our Ref. AUDEY.002C1), including its prosecution history, the cited references, and the Office Actions therein	09-01-2020	Bradley et al.	
	7	10,809,877(Our Ref. AUDEY.003C4), including its prosecution history, the cited references, and the Office Actions therein	10-20-2020	Bradley et al.	
	8	2002/0122053	09-05-2002	Dutta	
	9	2012/0254723	10-04-2012	Kasa et al.	
	10	2013/0104029	04-25-2013	Hendry et al.	
	11	2015/0106686	04-16-2015	Blitzstein	
	12	2017/0371975	12-28-2017	Chen et al.	
	13	2019/0340212	11-07-2019	Isager	
	14	16/991329**	08-12-2020	Bradley et al.	
	15	16/991346**	08-12-2020	Bradley et al.	
	16	16/991416**	08-12-2020	Bradley et al.	
	17	16/991392**	08-12-2020	Bradley et al.	
	18	16/991304**	08-12-2020	Bradley et al.	
	19	16/991381**	08-12-2020	Bradley et al.	
	20	16/991434**	08-12-2020	Bradley et al.	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2177	
<i>(Multiple sheets used when necessary)</i>		Examiner	Zuberi, Mohammed H.
SHEET 2 OF 2		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	21	16/991405**	08-12-2020	Bradley et al.	
	22	16/991505**	08-12-2020	Bradley et al.	
	23	16/991584**	08-12-2020	Bradley et al.	
	24	16/991578**	08-12-2020	Bradley et al.	
	25	16/991671**	08-12-2020	Bradley et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document Country Code-Number-Kind Code Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	26	Sato et al., "Accessibility Evaluation based on Machine Learning Technique", October 22-25, 2006, Portland, Oregon, USA, ACM, pages: 2 (Year: 2006)	
	27	Shadi et al., "Artificial Intelligence for Web Accessibility- Conformance Evaluation as a Way Forward?", April 23-25, 2018, Lyon, France, ACM, pages: 5 (Year: 2018)	

**Reference is co-owned by Applicant and signature indicates consideration of publication and file history, including references cited and office actions issued therein. The Examiner has access to these materials through PTO computer systems. If additional copies are desired, please notify Applicant through its attorneys.
33780263

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2177	
<i>(Multiple sheets used when necessary)</i>		Examiner	Zuberi, Mohammed H.
SHEET 1 OF 5		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	8,321,502	11-27-2012	Scoda	
	2	8,516,362	08-20-2013	Scoda	
	3	8,527,862	09-03-2013	Scoda et al.	
	4	8,589,484	11-19-2013	Scoda	
	5	8,650,474	02-11-2014	Scoda	
	6	8,868,638	10-21-2014	Scoda	
	7	8,983,935	03-17-2015	Scoda	
	8	8,984,164	03-17-2015	Scoda	
	9	9,275,023	03-01-2016	Scoda	
	10	9,311,281	04-12-2016	Scoda	
	11	9,319,244	04-19-2016	Taylor et al.	
	12	9,473,592	10-18-2016	Scoda	
	13	9,547,633	01-17-2017	Scoda	
	14	9,736,245	08-15-2017	Scoda	
	15	9,846,572	12-19-2017	Scoda	
	16	9,846,686	12-19-2017	Scoda	
	17	9,876,776	01-23-2018	Scoda	
	18	9,971,636	05-15-2018	Scoda et al.	
	19	10,015,226	07-03-2018	Scoda et al.	
	20	10,049,089	08-14-2018	Scoda et al.	
	21	10,116,726	10-30-2018	Scoda	
	22	10,120,847	11-06-2018	Scoda	
	23	10,198,414	02-05-2019	Scoda	
	24	10,282,401	05-07-2019	Scoda	
	25	10,218,775	02-026-2019	Scoda	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2177	
<i>(Multiple sheets used when necessary)</i>		Examiner	Zuberi, Mohammed H.
SHEET 2 OF 5		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number <i>Number - Kind Code (if known)</i> Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	26	10,423,709 (Our Ref. AUDEY.002A), including its prosecution history, the cited references, and the Office Actions therein	09-24-2019	Bradley et al.	
	27	10,444,934 (Our Ref. AUDEY.003A), including its prosecution history, the cited references, and the Office Actions therein	10-15-2019	Bradley et al.	
	28	10,452,730	10-22-2019	Scoda	
	29	2011/0307855	12-15-2011	Scoda et al.	
	30	2013/0305139	11-14-2013	Scoda	
	31	2014/0149447	05-29-2014	Scoda	
	32	2015/0319215	11-05-2015	Scoda	
	33	2017/0111431	04-20-2017	Scoda et al.	
	34	2018/0191764	07-05-2018	Chawla et al.	
	35	2018/0239516	08-23-2018	Scoda	
	36	2018/0349004	12-06-2018	Scoda et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document <i>Country Code-Number-Kind Code</i> Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	37	CA2716635C	11-01-2016	USABLENET INC.		
	38	CA2732540C	02-27-2018	USABLENET INC.		
	39	CA2773088A1	03-17-2011	USABLENET INC.		
	40	CA2800723A1	12-15-2011	USABLENET INC.		
	41	CA2800790A1	12-15-2011	USABLENET INC.		
	42	CA2807320A1	02-23-2012	USABLENET, INC.		

Examiner Signature / MOHAMMED H ZUBERI /	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2177
<i>(Multiple sheets used when necessary)</i>	Examiner	Zuberi, Mohammed H.
SHEET 3 OF 5	Attorney Docket No.	AUDEY.003C1

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document <i>Country Code-Number-Kind Code</i> Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	43	CA2811207A1	03-22-2012	USABLENET, INC.		
	44	CA2816336A1	05-18-2012	USABLENET INC.		
	45	CA2816338A1	05-18-2012	USABLENET INC.		
	46	CA2821769A1	06-28-2012	USABLENET INC.		
	47	CA2824861A1	03-10-2014	USABLENET INC.		
	48	CA2834466A1	05-29-2014	USABLENET INC.		
	49	CA2839006C	08-01-2017	USABLENET INC.		
	50	CA2839013A1	01-03-2013	USABLENET INC.		
	51	CA2843938A1	09-13-2014	USABLENET INC.		
	52	CA2845279A1	09-13-2014	USABLENET INC.		
	53	CA2848175A1	10-03-2014	USABLENET INC.		
	54	CA2855420A1	01-09-2015	USABLENET INC.		
	55	CA2858590A1	02-28-2015	USABLENET INC.		
	56	CA2861602A1	08-01-2013	USABLENET INC.		
	57	CA2868162A1	05-04-2015	USABLENET INC.		
	58	CA2944659A1	04-15-2017	USABLENET INC.		
	59	CA2947402A1	11-12-2015	USABLENET INC.		
	60	CA2951659A1	06-22-2017	USABLENET INC.		
	61	CA2958235A1	03-03-2016	USABLENET INC.		
	62	CA2963393A1	04-14-2016	USABLENET INC.		
	63	EP2363995B1	07-29-2015	USABLENET INC.		
	64	EP2476063A4	02-21-2018	USABLENET INC.		
	65	EP2580686A4	11-30-2016	USABLENET INC.		
	66	EP2580699A4	07-27-2016	USABLENET INC.		
	67	EP2606436A4	11-15-2017	USABLENET INC.		
	68	EP2616962A4	01-17-2018	USABLENET INC.		
	69	EP2638681B1	07-19-2017	USABLENET INC.		

Examiner Signature	/MOHAMMED H ZUBERI/	Date Considered	12/31/2020
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>			

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2177	
<i>(Multiple sheets used when necessary)</i>		Examiner	Zuberi, Mohammed H.
SHEET 4 OF 5		Attorney Docket No.	AUDEY.003C1

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document <i>Country Code-Number-Kind Code</i> Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	70	EP2638683B1	03-15-2017	USABLENET INC.		
	71	EP2656303A4	07-30-2014	USABLENET INC.		
	72	EP2724251B1	08-08-2018	USABLENET INC.		
	73	EP2724253A4	04-20-2016	USABLENET INC.		
	74	EP2732385A4	04-29-2015	USABLENET INC.		
	75	EP2738696A1	06-04-2014	USABLENET INC.		
	76	EP2778975A1	09-17-2014	USABLENET INC.		
	77	EP2778976B1	12-27-2017	USABLENET INC.		
	78	EP2787454A1	10-08-2014	USABLENET INC.		
	79	EP2807578A4	11-18-2015	USABLENET INC.		
	80	EP2827261A1	01-21-2015	USABLENET INC.		
	81	EP2851813A1	03-25-2015	USABLENET INC.		
	82	EP2869222A1	05-06-2015	USABLENET INC.		
	83	EP3140733A4	12-13-2017	Usablenet Inc.		
	84	EP3156919A1	04-19-2017	USABLENET INC.		
	85	EP3185145A1	06-28-2017	USABLENET INC.		
	86	EP3195136A4	04-25-2018	USABLENET INC.		
	87	EP3204874A4	07-04-2018	USABLENET INC.		
	88	WO 2001/75678	10-11-2001	USABLENET, INC.		
	89	WO2009111251A1	09-11-2009	USABLENET INC.		
	90	WO2011031868A1	03-17-2011	USABLENET INC.		
	91	WO2011156739A2	12-15-2011	USABLENET INC.		
	92	WO2011156743A2	12-15-2011	USABLENET INC.		
	93	WO2012024380A3	06-14-2012	USABLENET, INC.		
	94	WO2012036833A9	12-20-2012	USABLENET, INC.		
	95	WO2012064856A3	10-04-2012	USABLENET INC.		
	96	WO2012064857A3	09-27-2012	USABLENET INC.		

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
<p>*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.</p>	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210
	Filing Date	June 03, 2019
	First Named Inventor	Bradley, Sean D.
	Art Unit	2177
<i>(Multiple sheets used when necessary)</i>	Examiner	Zuberi, Mohammed H.
SHEET 5 OF 5	Attorney Docket No.	AUDEY.003C1

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document <i>Country Code-Number-Kind Code</i> Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T ¹
	97	WO2012088326A9	09-20-2012	USABLENET INC.		
	98	WO2012178167A3	02-28-2013	USABLENET INC.		
	99	WO2013003455A3	03-14-2013	USABLENET INC.		
	100	WO2013112285A1	08-01-2013	USABLENET INC.		
	101	WO2014040080A1	03-13-2014	USABLENET INC.		
	102	WO2015171228A1	11-12-2015	USABLENET INC.		
	103	WO2016032602A1	03-03-2016	USABLENET INC.		
	104	WO2016057092A1	04-14-2016	USABLENET INC.		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ¹
	105		

33320834

Examiner Signature / MOHAMMED H ZUBERI /	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/



website accessibility standards



All News Maps Images Videos More Settings Tools

About 716,000,000 results (0.41 seconds)

Ad · www.audioeye.com/ · (866) 331-5324

ADA Website Standards - Is Your Website ADA Compliant?

Quick, Cost-Effective Solutions Compliant with ADA & WCAG Laws on Day 1 of Implementation. Make Your Website Accessible Fast. Get Started Now with a Free Analysis from an Expert. Types: Accessibility Testing, ADA Website Compliance.

Ad · www.accessibe.com/ ·

The #1 Fully Automated - Web Accessibility Solution

Trusted to protect tens of thousands of small businesses and industry leaders alike. Protect your business from lawsuits. Open your website to the world! Machine Learning. Image Recognition. Giant Tech Collaborations. Contextual Understanding. Examples & Demo · Pricing · Technology · Story & Vision

Ad · www.siteimprove.com/accessibility ·

Siteimprove Web Accessibility - Check Your Sites Accessibility

Increase user experience and brand reputation with Siteimprove Accessibility. Check the...

Ad · www.onlineada.com/ ·

Get Your Website Accessible - WCAG A/AA Certification

Get your website audited, updated, and Accessibility Certified by the industry leader. Full...

WCAG 2.0 and WCAG 2.1 are stable, referenceable technical standards. They have 12-13 guidelines that are organized under 4 principles: perceivable, operable, understandable, and robust. For each guideline, there are testable success criteria, which are at three levels: A, AA, and AAA.

www.w3.org · WAI · standards-guidelines · wcag

Web Content Accessibility Guidelines (WCAG) Overview ...

About Featured Snippets Feedback

www.w3.org · WAI · standards-guidelines ·

W3C Accessibility Standards Overview | Web Accessibility ...

The Website of the World Wide Web Consortium's Web Accessibility Initiative. WCAG · WAI-ARIA · WCAG 2.1 at a Glance · Mobile

www.usability.gov · what-and-why · accessibility ·

Accessibility Basics | Usability.gov

By making your website accessible, you are ensuring that all of your potential users, including people with disabilities, have a decent user experience and are ...

medium.com · the-ada-checklist-website-compliance-g... ·

The ADA Checklist: Website Compliance Guidelines for 2019 ...

Nov 7, 2018 · Website accessibility can mean two things depending on the context: 1) ... law but it is the most frequently referenced set of technical standards.

www.essentialaccessibility.com · Blog ·

How To Ensure Website Compliance With Accessibility ...

Jun 7, 2019 · A complete guide to ensuring website accessibility. Learn more about web compliance accessibility standards and laws from experts.

[www.boia.org](#) > [blog](#) > [is-there-a-legal-requirement-to-i...](#) ▾

Is There a Legal Requirement to Implement WCAG?

Aug 22, 2019 - The Web Content Accessibility Guidelines (WCAG) provide the ... any specific standard like WCAG, but their websites do have to be accessible.

[en.wikipedia.org](#) > [wiki](#) > [Web_Content_Accessibility...](#) ▾

Web Content Accessibility Guidelines - Wikipedia

In 2017, a Federal Court in Florida identified the WCAG guidelines as the "industry standard" for website accessibility and found that Winn Dixie Store, Inc., ...

[www.searchenginejournal.com](#) > [Web Development](#) ▾

Website Accessibility & the Law: Why Your Website Must Be ...

Jan 9, 2019 - We know these standards as the World Wide Web Consortium, or W3C. Accessibility guidelines are known as WCAG. As of 2018, we are at ...

[www.ada.gov](#) > [pcatoolkit](#) ▾

ADA Tool Kit: Website Accessibility Under Title II of the ADA

May 7, 2007 - Accessible website design recognizes these differences and does not require people to see, hear, or use a standard mouse in order to access ...

People also ask

What is the accessibility of a website? ▾

What is WCAG standard? ▾

What does ADA compliant mean for websites? ▾

How do you ensure accessibility? ▾

[Feedback](#)

[www.accessibilityworks](#) > [blog](#) > [2020-ada-wcag-websi...](#) ▾

2020 ADA Website Accessibility Standards - Accessibility.Works

Jul 21, 2020 - Practical guide to current requirements for websites to avoid ADA lawsuits and achieve universal accessibility.

Searches related to website accessibility standards

website accessibility checklist	latest accessibility standards
website accessibility checker	web accessibility initiative
web accessibility	w3c accessibility checklist
wcag	accessibility testing

1 2 3 4 5 6 7 8 9 10 [Next](#)

Washington DC (Hagerstown MD), Virginia - From your internet address - Use precise location - [Learn more](#)

[Help](#) [Send feedback](#) [Privacy](#) [Terms](#)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Sean D. Bradley
App. No	: 16/430210
Filed	: June 3, 2019
For	: MODULAR SYSTEMS AND METHODS FOR SELECTIVELY ENABLING CLOUD-BASED ASSISTIVE TECHNOLOGIES
Examiner	: ZUBERI, Mohammad H.
Art Unit	: 2177
Conf No.	: 6600

COMMENTS ON THE EXAMINER'S STATEMENT OF REASONS FOR ALLOWANCE**Mail Stop Issue Fee**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant takes no position regarding any reasons for allowance presented by the Examiner (in the Notice of Allowance or elsewhere in the application's file history) other than the positions Applicant may have taken during prosecution. Therefore, the Examiner's reasons for allowance should not be attributed to Applicant as an indication of the basis for Applicant's belief that the claims are patentable. Furthermore, Applicant respectfully asserts that there may also be additional reasons for patentability of the claimed subject matter not explicitly stated in this record and Applicant does not waive its rights to such arguments by not further addressing such reasons herein.

Applicant also respectfully reserves the right to traverse the characterizations of what any particular reference shows or teaches, of what any combination of references shows or teaches, or the appropriateness of combining references. Further, by making certain amendments to the claims, Applicant is not conceding that previously pending

Application No.: 16/430210
Filing Date: June 3, 2019

claims are not patentable. Rather, the amendments are being made to facilitate expeditious prosecution of this application. Applicant reserves the right to pursue at a later date any previously pending or other broader or narrower claims that capture any subject matter supported by the application's disclosure. Accordingly, reviewers of this or any child or related prosecution history shall not reasonably infer that Applicant has made any disclaimers or disavowals of any subject matter.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: January 12, 2021

By: /THOMAS Y. YEE/
Thomas Y. Yee
Registration No. 57,013
Attorney of Record
Customer No. 20,995
212-849-3077

34195198

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

20995 7590 01/11/2021
KNOBBE MARTENS OLSON & BEAR LLP
 2040 MAIN STREET
 FOURTEENTH FLOOR
 IRVINE, CA 92614

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

_____ (Typed or printed name)
_____ (Signature)
_____ (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/430,210	06/03/2019	Sean D. Bradley	AUDEY 003C1	6600

TITLE OF INVENTION: Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	04/12/2021

EXAMINER	ART UNIT	CLASS-SUBCLASS
ZUBERI, MOHAMMED H	2177	715-747000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- "Fee Address" indication (or "Fee Address" indication form PTO/SB/47; Rev 03-09 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
- (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

- 1 Knobbe, Martens,
- 2 Olson & Bear, LLP
- 3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

AudioEye, Inc.

Tucson, AZ

Please check the appropriate assignee category or categories (will not be printed on the patent): Individual Corporation or other private group entity Government

4a. Fees submitted: Issue Fee Publication Fee (if required) Advance Order - # of Copies _____

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

Electronic Payment via EFS-Web Enclosed check Non-electronic payment by credit card (Attach form PTO-2038)

The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. 11-1410

5. Change in Entity Status (from status indicated above)

- Applicant certifying micro entity status. See 37 CFR 1.29
- Applicant asserting small entity status. See 37 CFR 1.27
- Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature /THOMAS Y. YEE/ Date 1/12/2021

Typed or printed name Thomas Y. Yee Registration No. 57,013

Electronic Patent Application Fee Transmittal

Application Number:	16430210			
Filing Date:	03-Jun-2019			
Title of Invention:	Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies			
First Named Inventor/Applicant Name:	Sean D. Bradley			
Filer:	Thomas Yee/Daisy Perez			
Attorney Docket Number:	AUDEY.003C1			
Filed as Large Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
UTILITY APPL ISSUE FEE	1501	1	1200	1200

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1200

Electronic Acknowledgement Receipt

EFS ID:	41618376
Application Number:	16430210
International Application Number:	
Confirmation Number:	6600
Title of Invention:	Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies
First Named Inventor/Applicant Name:	Sean D. Bradley
Customer Number:	20995
Filer:	Thomas Yee/Christina Gaul
Filer Authorized By:	Thomas Yee
Attorney Docket Number:	AUDEY.003C1
Receipt Date:	12-JAN-2021
Filing Date:	03-JUN-2019
Time Stamp:	12:58:43
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$1200
RAM confirmation Number	E20211BC59050816
Deposit Account	111410
Authorized User	Christina Gaul

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

37 CFR 1.16 (National application filing, search, and examination fees)

37 CFR 1.17 (Patent application and reexamination processing fees)

--	--	--	--	--	--

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Post Allowance Communication - Incoming	Comments.pdf	20323	no	2
			59b6bcae747c92f378ad63fbb415bd67e1a603da		

Warnings:

Information:

2	Issue Fee Payment (PTO-85B)	Issue_Fee.pdf	182299	no	1
			38d69fdb2ee63064fdce42782868ee4942305c6		

Warnings:

Information:

3	Fee Worksheet (SB06)	fee-info.pdf	30191	no	2
			d997d8ad1ffaf88f2d9b5cddb427dba1da5336f4		

Warnings:

Information:

Total Files Size (in bytes):	232813
-------------------------------------	--------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2177	
<i>(Multiple sheets used when necessary)</i>		Examiner	Zuberi, Mohammed H.
SHEET 1 OF 2		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	7,546,531	06-09-2009	Celik	
	2	8,452,600	05-28-2013	Fleizach	
	3	8,855,423	10-07-2014	Bonczyk et al.	
	4	10,037,318	07-31-2018	Liu	
	5	10,425,501	06-09-2018	Nasson et al. 09/2019	
Change(s) applied to document, /S.B./ 1/15/2021	6	10,762,280 (Our Ref. AUDEY.002C1), including its prosecution history, the cited references, and the Office Actions therein	09-01-2020	Bradley et al.	
	7	10,809,877(Our Ref. AUDEY.003C4), including its prosecution history, the cited references, and the Office Actions therein	10-20-2020	Bradley et al.	
	8	2002/0122053	09-05-2002	Dutta	
	9	2012/0254723	10-04-2012	Kasa et al.	
	10	2013/0104029	04-25-2013	Hendry et al.	
	11	2015/0106686	04-16-2015	Blitzstein	
	12	2017/0371975	12-28-2017	Chen et al.	
	13	2019/0340212	11-07-2019	Isager	
	14	16/991329**	08-12-2020	Bradley et al.	
	15	16/991346**	08-12-2020	Bradley et al.	
	16	16/991416**	08-12-2020	Bradley et al.	
	17	16/991392**	08-12-2020	Bradley et al.	
	18	16/991304**	08-12-2020	Bradley et al.	
	19	16/991381**	08-12-2020	Bradley et al.	
	20	16/991434**	08-12-2020	Bradley et al.	

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	16/430210	
	Filing Date	June 03, 2019	
	First Named Inventor	Bradley, Sean D.	
	Art Unit	2179	
<i>(Multiple sheets used when necessary)</i>		Examiner	ZUBERI, MOHAMMED H
SHEET 1 OF 3		Attorney Docket No.	AUDEY.003C1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number Number - Kind Code (if known) Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,751,544	06-15-2004	Hashimoto et al.	
	2	7,028,306	04-11-2006	Boloker et al.	
	3	7,181,692	02-20-2007	Siegel	
	4	7,194,411	03-20-2007	Slotznick et al.	
	5	7,246,058	07-17-2007	Burnett	
	6	7,702,674	04-20-2010	Hyder et al.	
	7	8,055,713	11-08-2011	Simske et al.	
	8	8,090,800	01-03-2012	Yee	
	9	8,103,956	01-24-2012	Trujillo	
	10	8,260,616	09-04-2012	O'Connor et al.	
	11	8,694,319	04-08-2014	Bodin et al.	
	12	8,874,716	08-22-2000	Daiwa Seiko Inc. (Withdrawn)	
	13	8,903,978	12-02-2014	Zerr et al.	
	14	9,208,783	12-08-2015	Ativanichayaphong et al.	
	15	9,268,753	02-23-2016	Hendry et al.	
	16	9,356,574	05-31-2016	Denninghoff	
	17	9,384,183	07-05-2016	Kasa et al.	
	18	9,727,660	08-08-2017	Barrell et al.	
	19	9,736,524	08-15-2017	Aravamudan et al.	
	20	9,875,671	01-23-2018	Gharpure et al.	
	21	9,996,613	06-12-2018	Jadhav et al.	
	22	10,237,334	03-19-2019	Alexander et al.	
	23	2002/0010715	01-24-2002	Chinn et al.	
	24	2003/0098803	05-29-2003	Gourgey et al.	
	25	2004/0006478	01-08-2004	Alpdemir et al.	
	26	2004/0056882	03-25-2004	Foreman et al.	
	27	2005/0160065	07-21-2005	Seeman	

Change(s) applied
to document,
/S.B./
1/15/2021

Examiner Signature /MOHAMMED H ZUBERI/	Date Considered 12/31/2020
*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T¹ - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.H.Z/



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/430,210	02/23/2021	10928978	AUDEY.003C1	6600

20995 7590 02/03/2021
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment is 161 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Sean D. Bradley, Tucson, AZ;
AudioEye, Inc., Tucson, AZ;
Mark D. Baker, Marietta, GA;
Jeffrey O. Jones, Mountain Park, GA;
Kenny P. Hefner, Buchanan, GA;
Adam Finkelstein, Alpharetta, GA;
Douglas J. Gilormo, Cumming, GA;
Taylor R. Bodnar, Tucson, AZ;
David C. Pinckney, Roswell, GA;
Charlie E. Blevins, Atlanta, GA;
Trevor C. Jones, Kennesaw, GA;
Helena Laymon, Duluth, GA;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

IR103 (Rev. 10/09)



**UNITED STATES
PATENT AND TRADEMARK OFFICE**

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #
16/430,210

RECEIPT DATE / TIME
04/19/2021 06:27:07 PM ET

ATTORNEY DOCKET #
AUDEY.003C1

Title of Invention

Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	10928978
CONFIRMATION #	6600	FILED BY	Kim Skoff
PATENT CENTER #	60179246	AUTHORIZED BY	Bryan Johnson
CUSTOMER #	20995	FILING DATE	06/03/2019
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Sean D. Bradley

Payment Information

PAYMENT METHOD CARD / 2564	PAYMENT TRANSACTION ID E20214R29470146	PAYMENT AUTHORIZED BY Fabiola Esmerio
PRE-AUTHORIZED ACCOUNT 111410	PRE-AUTHORIZED CATEGORY 37 CFR 1.16 (National application filing, search, and examination fees); 37 CFR 1.17 (Patent application and reexamination processing fees)	

FEE CODE	DESCRIPTION	ITEM PRICE(\$)	QUANTITY	ITEM TOTAL(\$)
1811	CERTIFICATE OF CORRECTION	160.00	1	160.00
			TOTAL AMOUNT:	\$160.00

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement

Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #
16/430,210

RECEIPT DATE / TIME
04/19/2021 06:27:07 PM ET

ATTORNEY DOCKET #
AUDEY.003C1

Title of Invention

Modular Systems and Methods For Selectively Enabling Cloud-Based Assistive Technologies

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	10928978
CONFIRMATION #	6600	FILED BY	Kim Skoff
PATENT CENTER #	60179246	FILING DATE	06/03/2019
CUSTOMER #	20995	FIRST NAMED INVENTOR	Sean D. Bradley
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Bryan Johnson

Documents

TOTAL DOCUMENTS: 2

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
Transmittal.pdf	-	Transmittal Letter	16 KB
Certificate of Correction.pdf	1	Request for Certificate of Correction	16 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
Transmittal.pdf	6C50C828BA48864F887BF8F110F92A56E630EE0998B2EDC78 D4F88AD0CD3EA00434120E2F445FA1F62D6BF6DC23F4E09B6 8CA1417D065E5FE2CB7E611FF1A7CD

Certificate of Correction.pdf

9D7BA2BCEB0160E82A28EC57603D9E6EC24356581EC2D719
257B89982BB0D85F387D67F3EDF736A45080E790E298766E7A
7F77900501EF59879E1C00A102CA39

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

REQUEST FOR CERTIFICATE OF CORRECTION

First Inventor : Sean D. Bradley
App. No. : 16/430210
Filed : June 3, 2019
Patent No. : 10,928,978
Issue Date : February 23, 2021
Title : MODULAR SYSTEMS AND METHODS FOR SELECTIVELY
ENABLING CLOUD-BASED ASSISTIVE TECHNOLOGIES
Conf. No. : 6600

Commissioner for Patents
Office of Data Management
Attention: Certificates of Correction Branch
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

Enclosed for filing is a Certificate of Correction in connection with the above-identified patent.

Some of the errors cited in the Certificate of Correction appear to have been incurred through the fault of the PTO (see 35 USC § 254, 37 CFR § 1.322, and MPEP § 1480). However, because this may not apply to each item in the Certificate of Correction, the \$160 fee under 37 CFR § 1.20(a) is submitted herewith. Please charge any additional fees to our Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: April 19, 2021

By: /Bryan Johnson/
Bryan J. Johnson
Registration No. 77,244
Registered Practitioner
(212) 849-3000

34690617

**UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION**

PATENT NO. : 10,928,978
APPLICATION NO. : 16/430210
ISSUE DATE : February 23, 2021
INVENTOR(S) : Sean D. Bradley

Page 1 of 1

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On page 2, in column 1, field (56), U.S. Patent Documents, line 27, delete "Bradley et al." and insert --O'Connor et al.--.

On page 3, in column 2, field (56), Other Publications, line 21, delete "www/" and insert --www.--.

In sheet 4 of 13, FIG. 4, reference numeral 403, line 1, delete "403." and insert --403--.

In column 1, line 60, delete "Brousealoud™" and insert --Browsealoud™--.

In column 6, line 62, delete "mark-up" and insert --markup--.

In column 17, line 60, delete "SIT" and insert --STT--.

34690486

MAILING ADDRESS OF SENDER:

Bryan J. Johnson
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, 14th Floor
Irvine, California 92614

DOCKET NO. AUDEY.003C1

PTO/SB/44 Equivalent

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 10,928,978 B2
APPLICATION NO. : 16/430210
DATED : February 23, 2021
INVENTOR(S) : Sean D. Bradley et al.

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

On page 2, in Column 1, item (56), U.S. Patent Documents, Line 27, delete "Bradley et al." and insert --O'Connor et al.--.

On page 3, in Column 2, item (56), Other Publications, Line 21, delete "www/" and insert --www.--.

In the Drawings

In sheet 4 of 13, FIG. 4, reference numeral 403, Line 1, delete "403." and insert --403--.

In the Specification

In Column 1, Line 60, delete "Brousealoud™" and insert --Browsealoud™--.

In Column 6, Line 62, delete "mark-up" and insert --markup--.

In Column 17, Line 60, delete "SIT" and insert --STT--.

Signed and Sealed this
Twenty-fifth Day of May, 2021



Drew Hirshfeld
*Performing the Functions and Duties of the
Under Secretary of Commerce for Intellectual Property and
Director of the United States Patent and Trademark Office*