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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 June 2009 (11.06.2009)

PCT

(10) International Publication Number
WO 2009/073368 A1

(51) International Patent Classification:
G06F 9/48 (2006.01) *G06F 9/46* (2006.01)

(21) International Application Number:
PCT/US2008/084060

(22) International Filing Date:
19 November 2008 (19.11.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2493/DEL/2007 29 November 2007 (29.11.2007) IN

(71) Applicant *(for all designated States except US)*: **MOTOROLA, INC.** [US/US]; 1303 East Algonquin Rd., Schaumburg, Illinois 60196 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **ENNAI, Anuraj Kunnummel** [IN/IN]; Flat No. 003, No. 26, M.S. Enclave, Link RoadKodihalli, Bangalore 560008 (IN).

(74) Agents: **BROWN, Larry G.** et al.; 1303 East Algonquin Road, Schaumburg, Illinois 60196 (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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND APPARATUS TO FACILITATE ASYNCHRONOUSLY INVOKING A BROWSER APPLICATION IN A CLIENT PLATFORM

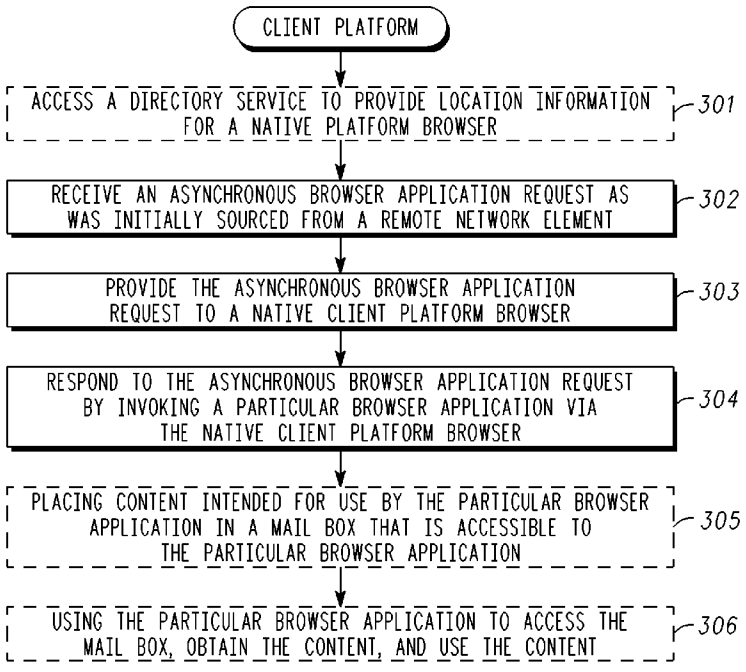


FIG. 3

(57) Abstract: A network element (209) can access (101) a directory service to obtain location information for a particular client platform browser (203) for which the network element does not have an already-established connection. That network element can then asynchronously issue (102) a browser application request using that location information to thereby facilitate asynchronously invoking a browser application in that client platform (201). That client platform, upon receiving (302) such an asynchronous browser application request, can responsively provide (303) that request to a native client platform browser. The client platform can then further respond to the asynchronous browser application request by invoking (304) a particular browser application via that native client platform browser.

WO 2009/073368 A1

Published:
— *with international search report*

METHOD AND APPARATUS TO FACILITATE ASYNCHRONOUSLY INVOKING A BROWSER APPLICATION IN A CLIENT PLATFORM

Technical Field

[0001] This invention relates generally to browser-based functionality and more particularly to browser applications.

Background

[0002] Browsers are known in the art. As used herein, a browser will be understood to refer to a software component capable of rendering Hypertext Markup content (including Hypertext Markup Language (HTML) pages) including content that can support at least one client side scripting technology such as JavaScript, and which can further allow for navigation of corresponding links such as HTML links. Explorer and Firefox are two current well known browser examples that many end users employ to navigate the World Wide Web. Browser capabilities have grown to permit increased functionality to meet growing end user demand for greater capabilities and flexibility when browsing. This includes, for example, the provision and use of browser applications. A browser application is a computer software application that is coded in a browser-supported language (such as HTML, Active Server Pages (ASP), PHP Hypertext Preprocessor (PHP), Practical Extraction and Reporting Language (Perl), and so forth) and that is reliant on a browser to render the application executable. The use of browser applications holds the potential for permitting access to a wide variety of processing capabilities without also requiring a priori native provisioning of such capabilities at the client platform.

[0003] To a large extent, however, the browser paradigm is a pull-based approach that presumes the browser to be a central and initiating component. To more fully realize the potential of browser applications, however, such may not always be the case. Asynchronous Javascript and eXtensible Markup Language (XML) (AJAX) represents one approach to addressing such needs. AJAX provides a mechanism to support asynchronous client requests to be made to a server without also blocking the

client platform user interface. This, in turn, enables richer applications that can be updated asynchronously to thereby provide a user experience somewhat closer to that which is more usually associated with native platform applications. COMET comprises another relevant example in this regard. COMET comprises an AJAX extension that permits server applications to asynchronously push information to a client platform using a long-lived already-established connection between the server and the client platform.

[0004] As helpful and useful as these various methodologies and approaches are, there nevertheless remain application settings that are insufficiently served by the prior art. As a general observation, for example, existing approaches rely upon a number of significant preliminary requirements before a desired level of asynchronous server-to-client functionality becomes possible. In the case of COMET, for example, an already-established connection must exist between such network elements before a server can effectively asynchronously push content towards a browser-based client. Problems such as these, in turn, can be viewed as obstructing the deployment of an effective and robust Web operating system as envisioned by some industry participants.

Brief Description of the Drawings

[0005] The above needs are at least partially met through provision of the method and apparatus to facilitate asynchronously invoking a browser application in a client platform described in the following detailed description, particularly when studied in conjunction with the drawings, wherein:

[0006] FIG. 1 comprises a flow diagram as configured in accordance with various embodiments of the invention;

[0007] FIG. 2 comprises a block diagram as configured in accordance with various embodiments of the invention; and

[0008] FIG. 3 comprises a flow diagram as configured in accordance with various embodiments of the invention.

[0009] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been drawn to scale. For example, the dimensions and/or relative positioning of some of the elements in the figures may be exaggerated relative to other elements to help to improve understanding of various embodiments of the present invention. Also, common but well-understood elements that are useful or necessary in a commercially feasible embodiment are often not depicted in order to facilitate a less obstructed view of these various embodiments of the present invention. It will further be appreciated that certain actions and/or steps may be described or depicted in a particular order of occurrence while those skilled in the art will understand that such specificity with respect to sequence is not actually required. It will also be understood that the terms and expressions used herein have the ordinary meaning as is accorded to such terms and expressions with respect to their corresponding respective areas of inquiry and study except where specific meanings have otherwise been set forth herein.

Detailed Description

[0010] Generally speaking, pursuant to these various embodiments, a network element can access a directory service to obtain location information (i.e., access information) for a particular client platform browser for which the network element does not have an already-established connection. That network element can then asynchronously issue a browser application request using that location information to thereby facilitate asynchronously invoking a browser application in that client platform. As used herein, those skilled in the art will recognize and understand that the expression "asynchronously" refers to the client browser (and the application running on it) not listening for application requests/information from a network, but rather where these components are awoken by a callback whenever a remote network entity delivers an application invocation request.

[0011] This location information can comprise, for example, location information for the browser application in the client platform itself. By another approach, this location information can comprise location information for a push proxy that receives messages on behalf of this particular client platform browser.

[0012] By one approach, this browser application request can comprise, at least in part, a network address that is intended for use by the client platform browser. In some cases, as where the network address may comprise a non-translatable Internet Protocol (IP) address, this browser application request can additionally comprise relevant port information (such as a port pair or a Uniform Resource Identifier that is routable to an IP address:port pair as is known in the art).This usage can comprise, for example, downloading content at this network address that is necessary to running the invoked browser application. Illustrative content in this regard might comprise, for example, an application library, authorization credentials, and so forth. These teachings will also support, subsequent to the issuance of such a browser application request, again using the aforementioned location information to asynchronously issue additional content to be delivered to the browser application.

[0013] These teachings will also accommodate corresponding behaviors and functionality on the part of the client platform. This can comprise, for example, receiving such an asynchronous browser application request as was initially sourced from a remote network element and responsively providing that request to a native client platform browser. The client platform can then further respond to the asynchronous browser application request by invoking a particular browser application via that native client platform browser.

[0014] Such a client platform can also be configured and arranged to access a directory service to provide the aforementioned location to thereby facilitate the aforementioned network element activities.

[0015] By one approach, these teachings will also accommodate use of a mailbox mechanism that can receive all or part of the aforementioned browser application request and/or subsequent messages. This can comprise, for example, placing content from such requests/messages that is intended for use by the intended browser application into a mailbox that is accessible (exclusively or as shared with other applications) by the browser application. So configured, the browser application can then access that mailbox, obtain that content, and use that content as appropriate.

[0016] So configured and arranged, those skilled in the art will recognize and appreciate that these teachings readily and efficiently facilitate the remote asynchronous invocation of a browser, and particularly a browser application, that is resident to a given client platform with which the invoking network element does not otherwise have an already established connection. Accordingly, it will be appreciated that these teachings permit a degree of control and effect that is presently lacking in the art. In both form and substance, these teachings will also be understood to considerably advance the technological basis for establishing a Web-based operating system.

[0017] These and other benefits may become clearer upon making a thorough review and study of the following detailed description. Referring now to the drawings, and in particular to FIG. 1, an illustrative process that is compatible with many of these teachings will now be presented. This process 100 can be implemented, for example, by a network element of choice. As one example in this regard, this network element can comprise a server application/agent.

[0018] Pursuant to this process 100, the network element can optionally access 101 a directory service to obtain location information for a particular client platform browser for which the network element does not have an already-established connection. This directory service can comprise, for example, a presence server with which the client platform browser (or a surrogate or proxy on the browser's behalf) registers its actual and/or scheduled location on some regular and/or event-driven basis.

[0019] The location information itself can vary in accordance with the needs and/or opportunities as tend to characterize a given application setting. By one approach, for example, this location information can comprise location information for the browser application itself in the client platform. As an illustrative, non-limiting example in this regard, the location information might comprise an IP address-port pair such as 10.168.1.1:1234 or a globally routable Uniform Resource Locator (URL) such as sip:name@domain.com. By another approach, this location information can comprise location information for a push proxy that receives messages on behalf of the particular client platform browser. In this case, the location information might

comprise, for example, a Uniform Resource Locator (URL) such as a World Wide Web address. By yet another approach, this location information can comprise location information for a publish-subscribe message queue to which the particular client platform browser subscribes (such queues being known in the art and requiring no further elaboration here). In this case, and again, the location information itself might comprise a URL.

[0020] The specific way by which the network element accesses the directory service can also vary with the specifics of a given application setting. Those skilled in the art will recognize that, in many cases, such access can be achieved using an extranet such as the Internet. Various methodologies and technologies exist in this regard and others are likely to be promulgated in the future. Various approaches are known in the art in this regard. As these teachings are not overly sensitive to any particular selection in this regard, for the sake of brevity and the preservation of clarity, further elaboration in this regard will not be presented here.

[0021] In any event, and regardless of how the network element acquires the aforementioned location information, this process 100 then provides for asynchronously issuing 102 a browser application request using the location information. In particular, this browser application request is transmitted in a message using that location information to form the address of the intended message recipient. There are various ways by which such a message can be transmitted and it will again be recognized that a network such as the Internet will often serve well in this regard.

[0022] At a minimum, this browser application request comprises a message that will be understood, upon eventual receipt by the client platform, to comprise a request/instruction that the client platform browser enable at least one identified browser application. The purpose of such a request/instruction can of course vary. In many cases, however, the purpose of this request/instruction will be to thereby cause the client platform to asynchronously and automatically display, deliver, and/or process some information for the benefit of the client platform end user.

[0023] The specific client platform browser application that comprises the subject of the browser application request is essentially unlimited. Examples include,

but are not limited to, word processing applications, spreadsheet applications, slide presentation applications, graphic image processing applications, animation playback applications, audio and/or video streaming content playback applications, and so forth. Those skilled in the art will recognize that many other examples are possible and that others will likely become available in the future.

[0024] In some cases, the specific application will be fully native to the recipient client platform. In other cases, however, the specific application, or particular modules, functionality, or capabilities of that specific application will not be presently available at the client platform (as may be the case when implementing these teachings in a Web operating system context). In such cases, this browser application request can comprise, at least in part, a network address that is intended for use by the particular client platform browser to download content that is necessary (or useful) to running the browser application to be invoked. This content can of course vary with the needs and/or opportunities as characterize a given application setting. Content examples range from the complete browser application itself (as may be available, for example, via an application library) to application components to such items as authorization credentials as may be useful or necessary to acquiring and/or using the content, the application, or the like.

[0025] In some cases, the browser application request may also comprise, at least in part, content to be delivered to the browser application upon invoking the browser application. This might comprise, for the sake of illustration and not by way of limitation, programming to be executed by the browser application, a network address (such as a corresponding Web URL) for streaming content to be rendered by the browser application, data (such as, for example, a word processing document) to be displayed at the client platform by the browser application, and so forth are all relevant examples in this regard.

[0026] For some purposes, the foregoing steps may be sufficient to meet present needs. In other instances, however, it may be useful or necessary for the network element to transmit additional content for the application browser subsequent to the foregoing steps having been accomplished. To meet such a need, this process 100 will also optionally, if desired, accommodate also asynchronously issuing 103

additional content to be delivered to the browser application while again using the aforementioned location information. Such content can of course vary with the specifics of a given application setting. Non-limiting examples would include further data to be processed by the browser application, an additional network address (or addresses) at which the browser application can access or receive additional data or functionality, and so forth.

[0027] Those skilled in the art will appreciate that the above-described processes are readily enabled using any of a wide variety of available and/or readily configured platforms, including partially or wholly programmable platforms as are known in the art or dedicated purpose platforms as may be desired for some applications. Referring now to FIG. 2, an illustrative approach to such a platform will now be provided.

[0028] In this illustrative example, the client platform 201 comprises a processor 202 having the aforementioned native browser 203. Various browsers are known in the art and the selection of a particular browser will depend in a given instance upon the needs and/or requirements as characterize that given instance. It will also be understood that this client platform 201 may comprise a plurality of differing browsers if desired. Those skilled in the art will recognize and appreciate that such a processor can comprise a fixed-purpose hard-wired platform or can comprise a partially or wholly programmable platform. All of these architectural options are well known and understood in the art and require no further description here.

[0029] In this illustrative embodiment the processor 202 operably couples to a receiver 204 (which may comprise, if desired, a wireless receiver of choice). This receiver 204 can be configured and arranged, for example, to receive the aforementioned asynchronous browser application request as was initially sourced from a remote network element 209 as per the description provided above. This can comprise, for example, an ability to interact compatibly with one or more external networks 207 such as, but not limited to, the Internet. As also was noted above, this may involve eventual reception of such a request via a push proxy 210 or a message

queue 212 via, for example, an intervening network (or networks) 207 such as, but not limited to, the Internet.

[0030] This client platform 201 may also optionally comprise, if desired, a mailbox memory 205 that is operably coupled to the receiver 204 and the processor 202. This may comprise, for example and as illustrated, coupling to the receiver 204 via that processor 202 though other possibilities of course exist. This mailbox memory 205 comprises an area of memory that is specifically (and, in many cases, largely or fully exclusively) accessible by the aforementioned browser 203 and/or a particular browser application. Such partitioning and access control is known in the art and need not be explained here in more detail.

[0031] Shown below is an illustrative embodiment of the mailbox as rendered in pseudo JavaScript style and which is easily implementable as a JavaScript Object by persons versed in the art:

```
interface XMLHTTPMailBox {

//If a new message is received, event listener shall be invoked if specified.
attribute EventListener   onreadystatechange;

//state of the object. One of Uninitialized, receiving, received, completed.
readonly attribute unsigned short  readyState;

//Create new MailBox for this app. Credentials are optional
XMLHTTPMailBox(in DOMString user, in DOMString password, in DOMString
appId)

//Check for any messages destined for the app.
int checkForMessages();

//Get content type of message
DOMString getContentType(int messageId);
```

```
//Retrieve message
DOMString getMessage(int messageId);

//Get all message headers
DOMString getAllMessageHeaders();

//Get a message header
DOMString getMessageHeader(DOMString headerLabel);

//Gets URL of server application which invoked current application.
DOMString getInvokerURL();
}
```

The following now represents how such a mailbox can be used to read content by an invoked application.

```
//Obtain a Mailbox instance
var mBox = new XMLHTTPMailBox("user1","password1", "app001");
//Check if one or more messages are waiting
var n = mBox.checkForMessages();
if(n>=1)
{
  for(i=0;i<n;i++)
  {
//read each message
    var msg = mBox.getMessage(n);
    //Do some thing with message....
    .....
  }
}
```

```
//Listen for further messages
mBox.onreadyStateChange = processMessage();

//Listener implementation
function processMessage()
{
 var msg = mBox.getMessage(1);
 //Do something with message
 .....
}
```

[0032] So configured, and as will also be described below, the processor 202 can be configured and arranged (via, for example, corresponding programming as will be well understood by those skilled in the art) to place, for example, received information that is intended for use by the particular browser application in this mailbox memory 205 to thereby make such received information available to the invoked browser application. This can comprise information as was received in an original asynchronous browser application request as described above and/or can comprise information as was received in a subsequent transmission by the corresponding network element.

[0033] Also optionally and if desired, this client platform 201 can comprise a transmitter 206. This transmitter 206 can also comprise a wireless and/or wireline transmitter of choice and can serve to facilitate, for example, transmitting location information to a directory service 208 (as described below). Like the receiver 204, this transmitter 206 can be configured and arranged to interact compatibly with, for example, the aforementioned network(s) 207 to facilitate such functionality. The specifics of this compatible interaction will of course vary with the specifics of a given application setting as will be well understood by those skilled in the art.

[0034] So configured, the client platform 201 can receive an asynchronous browser application request as sourced by a given network element such as the

depicted server application/agent 209. This request may be directed specifically to the client platform 201 or may be conveyed via a push proxy 201, a message queue 212, or the like as determined by the location information as may have been obtained by the serve application/agent 209 from the directory service 208. This asynchronous browser application request, in turn, can be employed by the processor 202 to invoke its native browser 203 and, more particularly, a particular browser application. By one approach, all of part of the enabling code for the browser application may, in turn, be downloaded by the processor 202 from one or more application sources 211 (such as application servers configured and arranged to provide this service and functionality) using, for example, network address information as may have been received with the initial asynchronous browser application request and then retrieved by the browser 203 from the mailbox memory 205.

[0035] Those skilled in the art will recognize and understand that the system configuration described comprises an illustrative example and that other architectural configurations are possible. Those skilled in the art will also recognize and understand that such a client platform 201 may be comprised of a plurality of physically distinct elements as is suggested by the illustration shown in FIG. 2. It is also possible, however, to view this illustration as comprising a logical view, in which case one or more of these elements can be enabled and realized via a shared platform. It will also be understood that such a shared platform may comprise a wholly or at least partially programmable platform as are known in the art.

[0036] Referring now to FIG. 3, a corresponding process 300 for such a client platform will be described. For the purposes of this illustrative example, the aforementioned processor can be configured and arranged (via, for example, corresponding software and/or firmware programming) to carry out some or all of the now described steps, actions, and functionality.

[0037] Pursuant to this client platform process 300, the client platform can optionally, if desired, access 301 a directory service (such as the directory service described above) to provide the aforementioned location information for the native client platform browser to thereby facilitate the remote network element being able to source the aforementioned asynchronous browser application request. This step can

occur on a regular or an irregular schedule if desired. By one approach such a step can be triggered as a function of time (such as a time of day, an amount of time as may have passed since a last location information update, and so forth). By another approach, in addition to or in lieu of the time-based approach just mentioned, this step can be event driven. By this approach, for example, the directory service can be updated with fresh location information when a given event of interest occurs. Examples of such events might include, but are certainly not limited to, the client platform sensing a new location, receiving a location update probe from the directory service, first becoming attached to a mobile network, and so forth.

[0038] In any event, this process 300 then provides for receiving 302 an asynchronous browser application request as was initially sourced from a remote network element as described above. As noted, this can comprise receiving the request directly from the network element or via some intermediary such as the aforementioned push proxy, message queue, or the like. As is also noted above, this asynchronous browser application request can include content intended for use by the particular browser application (which may comprise executable code, data, and/or one or more network addresses where executable code and/or data can be acquired by the browser application).

[0039] This process 300 then provides 303 this asynchronous browser application request to the native client platform browser such that the latter can now respond 304 to the asynchronous browser application request by invoking a particular corresponding browser application. This may or may not comprise literally passing the request in its received form to the browser; in some cases it may be useful or necessary to first translate that request into a format that is compatible with and usable by the browser. In such a case, the aforementioned processor can be programmed to carry out this translation in a direct or indirect manner.

[0040] So configured and arranged, those skilled in the art will recognize and appreciate that these teachings indeed permit a browser application to be asynchronously invoked by a remote network element with which the client platform does not presently have an already-established connection. This, in turn, opens a wide

variety of possibilities that are simply not otherwise readily available via a browser application-based capability using prior techniques and technologies in this regard.

[0041] As noted above, the network element may have additional content to provide to the invoked browser application subsequent to the above actions having been undertaken. In such a case, if desired, this process 300 will also optionally accommodate placing 305 such subsequently received content in the aforementioned mailbox. As the contents of that mailbox are accessible to this particular browser application, this process 300 can then optionally provide for using 305 this particular browser application to access the mailbox, obtain its contents and hence the aforementioned message (in substance if not in form), and use that message as appropriate. Depending upon the message contents itself, this might comprise, for example, accessing a network resource to download additional executable code and/or data. As another example, the message contents themselves might comprise executable code and/or data to be used by the browser application. (Those skilled in the art will appreciate that an invoked application can, for example, poll the aforementioned mailbox for further data or can receive callbacks by registering a handler function to the mailbox that would be invoked by the mailbox when a subsequent item of content becomes available.)

[0042] So configured, these teachings permit a browser application in a client platform to be asynchronously invoked notwithstanding an absence of an already-established connection between that client platform and a remote application-invoking network element. This, in turn, opens the door to numerous opportunities to better facilitate a more traditional and/or seamless end user experience.

[0043] Those skilled in the art will recognize that a wide variety of modifications, alterations, and combinations can be made with respect to the above described embodiments without departing from the spirit and scope of the invention, and that such modifications, alterations, and combinations are to be viewed as being within the ambit of the inventive concept. As one example in this regard, those skilled in the art will recognize and appreciate that these teachings can be employed to permit a given browser (which may be remote or native) to invoke a given client application instance and to pass some content thereto. As a further example, these teachings can

also be employed to permit a given client application to invoke a browser application and to pass content to that invoked browser application as desired. And as yet a further example in this regard, it will be clearly understood that these teachings will readily accommodate invoking, partially or fully simultaneously, a plurality of browser applications as per the dictates of one or more request-sourcing network elements. Those skilled in the art will further appreciate that these teachings can be readily applied in a manner designed to support specific security measures. For example, one could restrict access to the aforementioned directory services by client and/or server (using, for example, mechanisms similar to the authentication and authorization techniques of Secure Sockets Layer (SSL) processing), push information securely to thereby avoid at least some spamming and unsecure content, and/or restrict access the aforementioned mailbox using security credentials so that only authorized applications can access content retained therein.

CLAIMS

1. A method to facilitate asynchronously invoking a browser application in a client platform comprising:
at a network element:

accessing a directory service to obtain location information for a particular client platform browser for which the network element does not have an already-established connection;

asynchronously issuing a browser application request using the location information.

2. The method of claim 1 wherein the location information comprises location information for at least one of:

the browser application in the client platform;

a push proxy that receives messages on behalf of the particular client platform browser.

3. The method of claim 1 wherein the browser application request comprises, at least in part, a network address that is intended for use by the particular client platform browser to download content necessary to running the browser application to be invoked.

4. The method of claim 3 wherein the content comprises, at least in part, at least one of:

an application library;

authorization credentials;

application data.

5. The method of claim 1 wherein the browser application request comprises, at least in part, content to be delivered to the browser application upon invoking the browser application.

6. The method of claim 1 further comprising:

subsequent to issuing the browser application request, asynchronously issuing additional content to be delivered to the browser application using the location information.

7. The method of claim 1 wherein the location information comprises location for a publish-subscribe message queue to which the particular client platform browser subscribes.

8. A client platform comprising:

a receiver configured and arranged to receive an asynchronous browser application request as was initially sourced from a remote network element;

a processor operably coupled to the receiver and having a browser configured and arranged to run at least one browser application, wherein the processor is further configured and arranged to respond to the asynchronous browser application request by invoking a particular browser application via the browser.

9. The client platform of claim 8 wherein the processor is further configured and arranged to access a directory service to provide location information for the browser to thereby facilitate the remote network element sourcing the asynchronous browser application request.

10. The client platform of claim 8 further comprising:

mailbox memory operably coupled to the receiver and the processor;

and wherein the processor is further configured and arranged to place additional received information intended for use by the particular browser application in the mailbox memory to thereby make such received information available to the particular browser application.

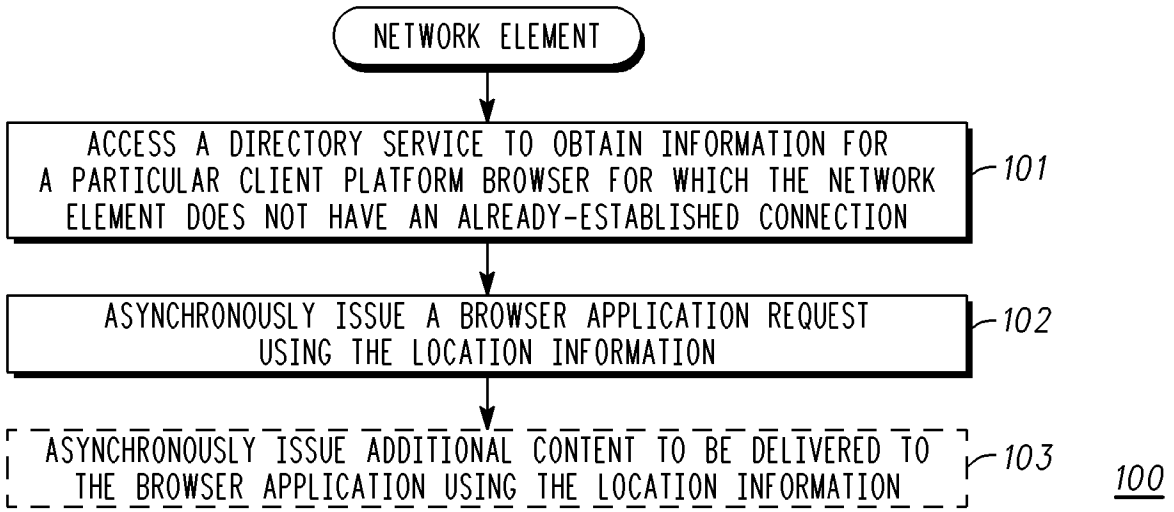

NETWORK ELEMENT
ACCESS A DIRECTORY SERVICE TO OBTAIN INFORMATION FOR A PARTICULAR CLIENT PLATFORM BROWSER FOR WHICH THE NETWORK ELEMENT DOES NOT HAVE AN ALREADY-ESTABLISHED CONNECTION
101
ASYNCHRONOUSLY ISSUE A BROWSER APPLICATION REQUEST USING THE LOCATION INFORMATION
102
ASYNCHRONOUSLY ISSUE ADDITIONAL CONTENT TO BE DELIVERED TO THE BROWSER APPLICATION USING THE LOCATION INFORMATION
103
100

FIG. 1

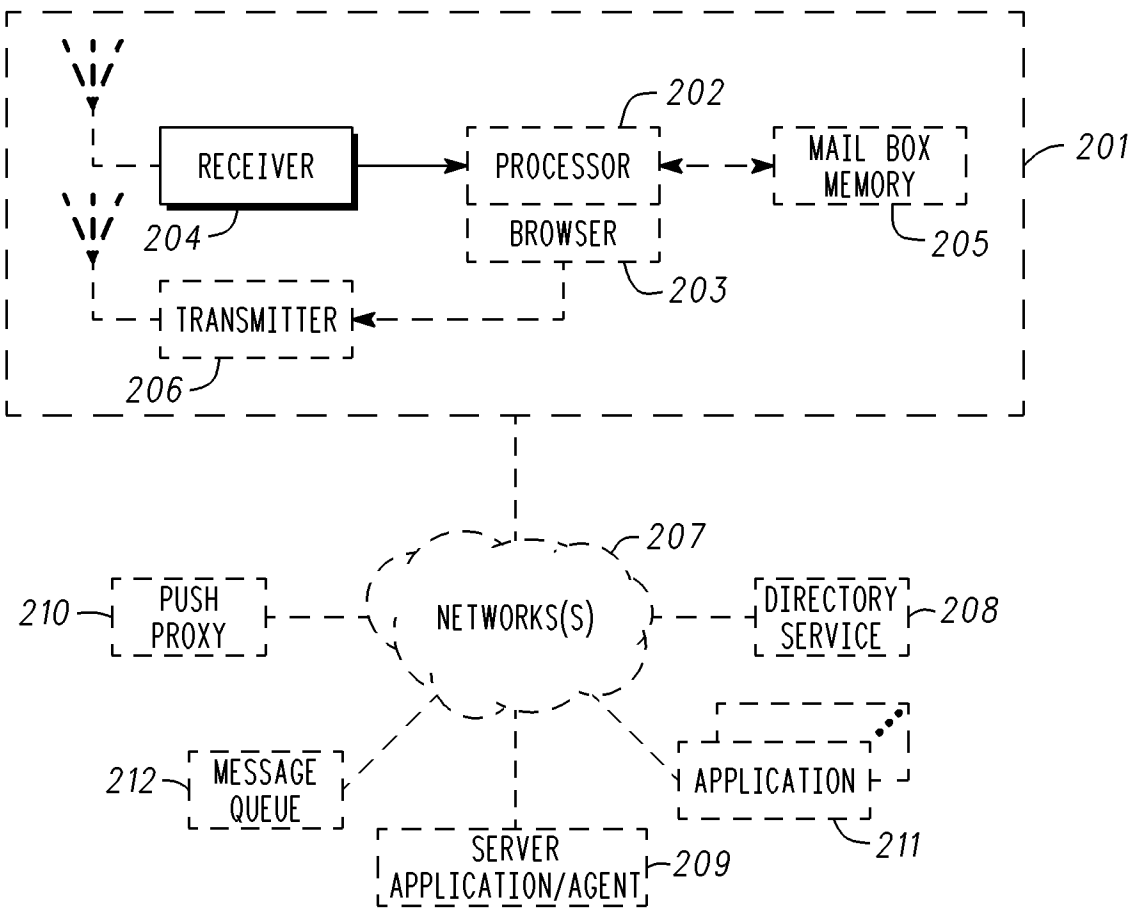
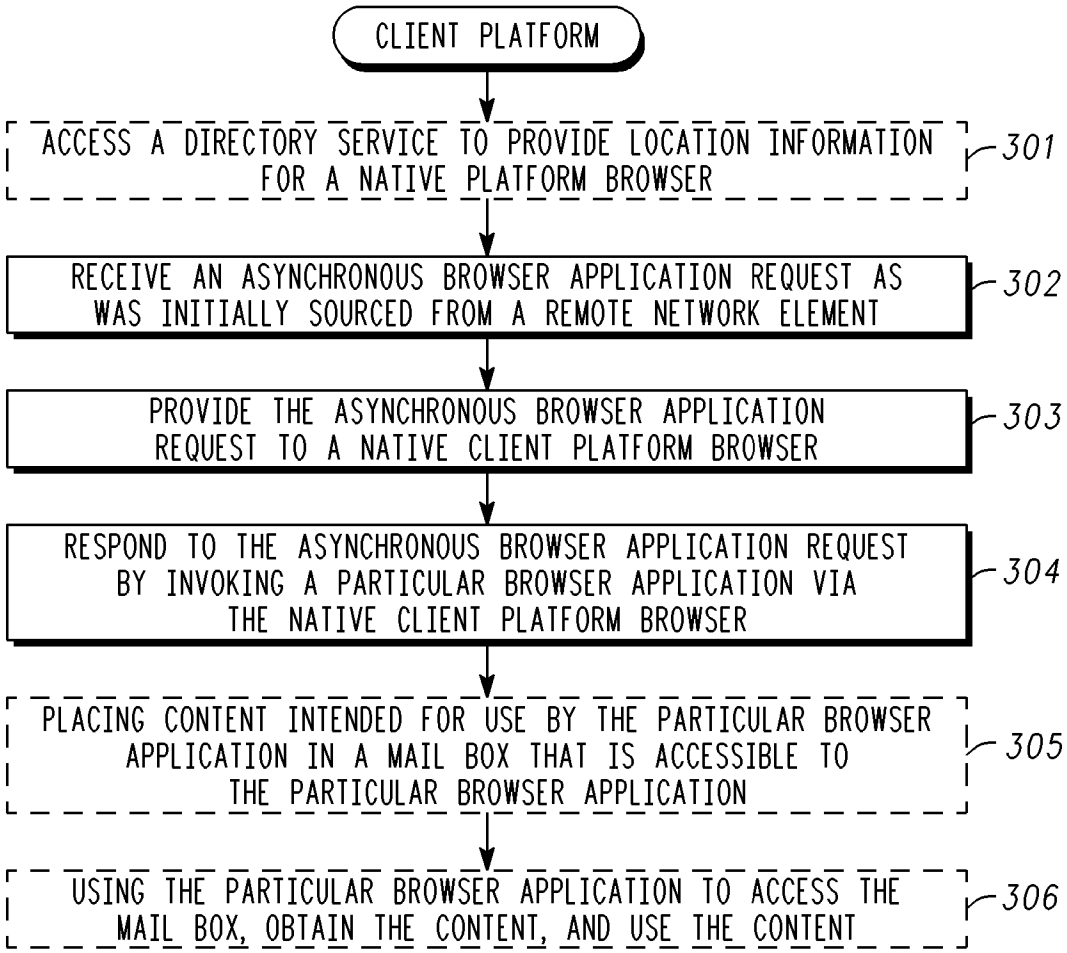

RECEIVER
204
PROCESSOR
202
MAIL BOX MEMORY
205
BROWSER
203
TRANSMITTER
206
201
207
NETWORKS(S)
PUSH PROXY
210
DIRECTORY SERVICE
208
MESSAGE QUEUE
212
APPLICATION
211
SERVER APPLICATION/AGENT
209

FIG. 2


CLIENT PLATFORM
ACCESS A DIRECTORY SERVICE TO PROVIDE LOCATION INFORMATION FOR A NATIVE PLATFORM BROWSER
301
RECEIVE AN ASYNCHRONOUS BROWSER APPLICATION REQUEST AS WAS INITIALLY SOURCED FROM A REMOTE NETWORK ELEMENT
302
PROVIDE THE ASYNCHRONOUS BROWSER APPLICATION REQUEST TO A NATIVE CLIENT PLATFORM BROWSER
303
RESPOND TO THE ASYNCHRONOUS BROWSER APPLICATION REQUEST BY INVOKING A PARTICULAR BROWSER APPLICATION VIA THE NATIVE CLIENT PLATFORM BROWSER
304
PLACING CONTENT INTENDED FOR USE BY THE PARTICULAR BROWSER APPLICATION IN A MAIL BOX THAT IS ACCESSIBLE TO THE PARTICULAR BROWSER APPLICATION
305
USING THE PARTICULAR BROWSER APPLICATION TO ACCESS THE MAIL BOX, OBTAIN THE CONTENT, AND USE THE CONTENT
306

300

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2008/084060

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/48
ADD. G06F9/46

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 H04L

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 practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 092 703 B1 (PAPINEAU SCOTT [US]) 15 August 2006 (2006-08-15) column 20, line 42 - column 22, line 20; figure 17 column 25, line 9 - column 28, line 26	1-10
X	EP 1 308 858 A (OPENWAVE SYS INC [US]) 7 May 2003 (2003-05-07) paragraph [0020] - paragraph [0021]; figures 1,3A	1-10

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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 document 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

Date of the actual completion of the international search	Date of mailing of the international search report
27 February 2009	10/03/2009

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Carciofi, Andrea

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2008/084060

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SANJAY: "J2ME MIDP apps launching wap browser" TECHIESABODE TIPS, [Online] 21 December 2005 (2005-12-21), XP002516950 Retrieved from the Internet: URL:http://www.techiesabode.com/show/show_tips_w.php?tip_id=131> [retrieved on 2009-02-27] the whole document	1,8
A	"WAP PUSH ARCHITECTURAL OVERVIEW" WAP PUSH ARCHITECTURAL OVERVIEW, XX, XX, 8 November 1999 (1999-11-08), pages 1-26, XP002942202 the whole document	1-10

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2008/084060

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7092703 B1	15-08-2006	US 7471947 B1	30-12-2008
EP 1308858 A	07-05-2003	NONE	