US 20120117494A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2012/0117494 A1

(43) Pub. Date: May 10, 2012

Floyd et al.

(54) SYSTEM AND METHOD FOR EXPEDITING INFORMATION DISPLAY

(76) Inventors: **Michel Floyd**, Redwood City, CA (US); **Brian Spiegel**, San Jose, CA (US)

(21) Appl. No.: **13/350,696**

(22) Filed: **Jan. 13, 2012**

Related U.S. Application Data

(63) Continuation of application No. 11/903,272, filed on Sep. 21, 2007.

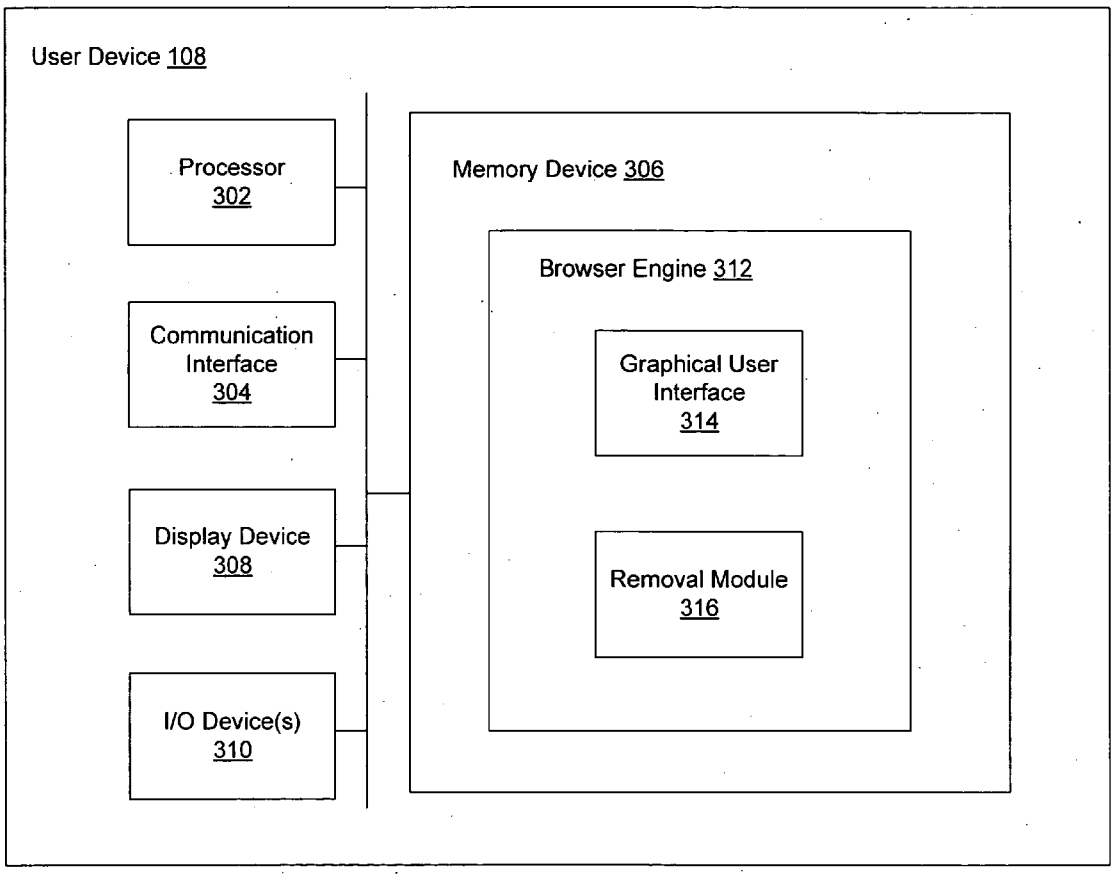
Publication Classification

(51) **Int. Cl.**
G06F 3/00 (2006.01)

(52) **U.S. Cl.** **715/760**

(57) **ABSTRACT**

Exemplary systems and methods for expediting information display on a user device are provided. In exemplary embodiments, a request for information is received. An application server then generates and provides source code for a web page containing the requested information. In exemplary embodiments, a removal module is returned with the source code for the first web page. The removal module is configured to remove unnecessary elements from subsequent source code in order to expedite loading and displaying of subsequent web pages on the user device. In some embodiments, the web page comprises a syndicated web page within a window of a main web page.



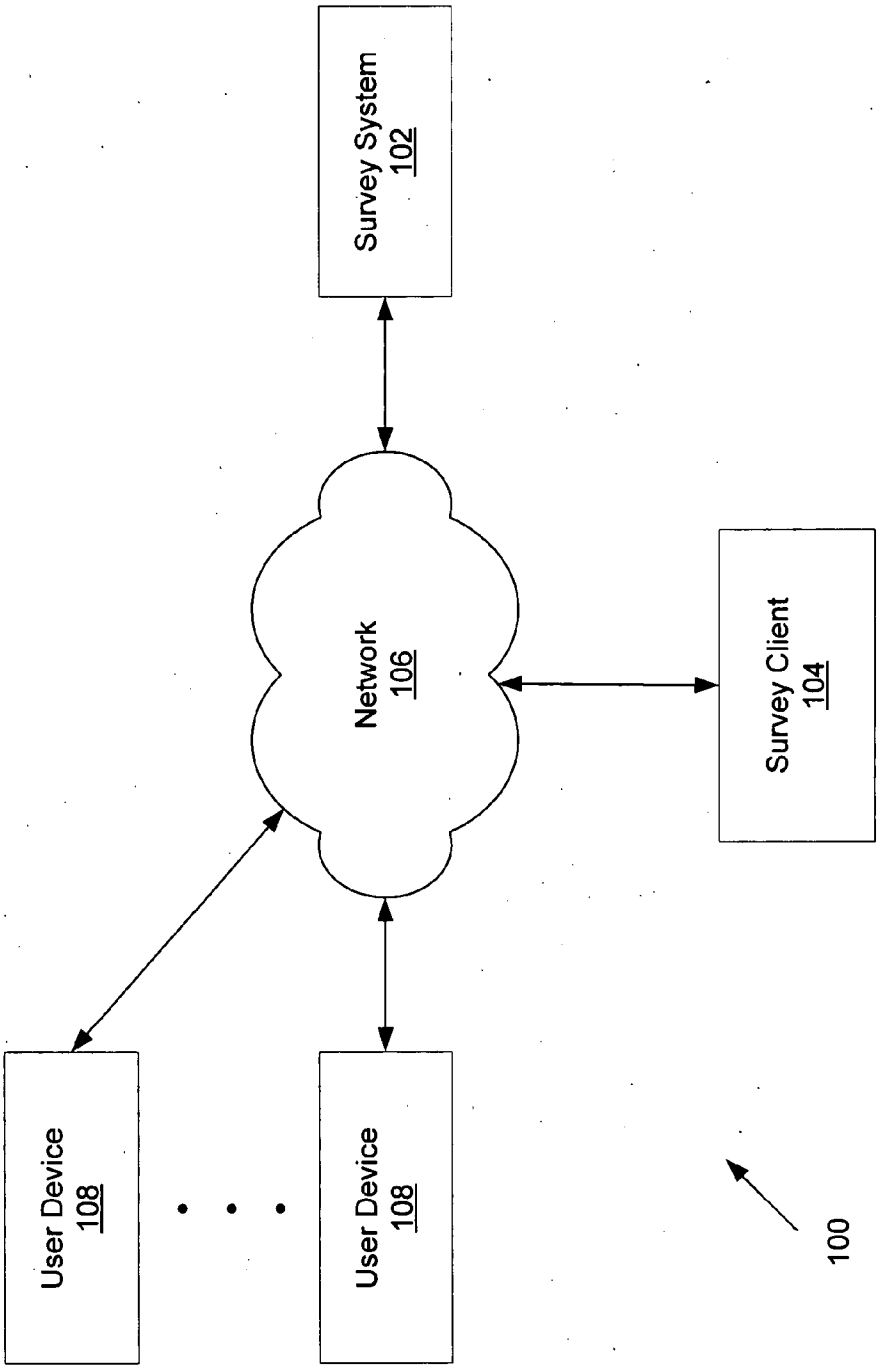

Survey System
102
Network
106
Survey Client
104
User Device
108
User Device
108
100

FIG. 1

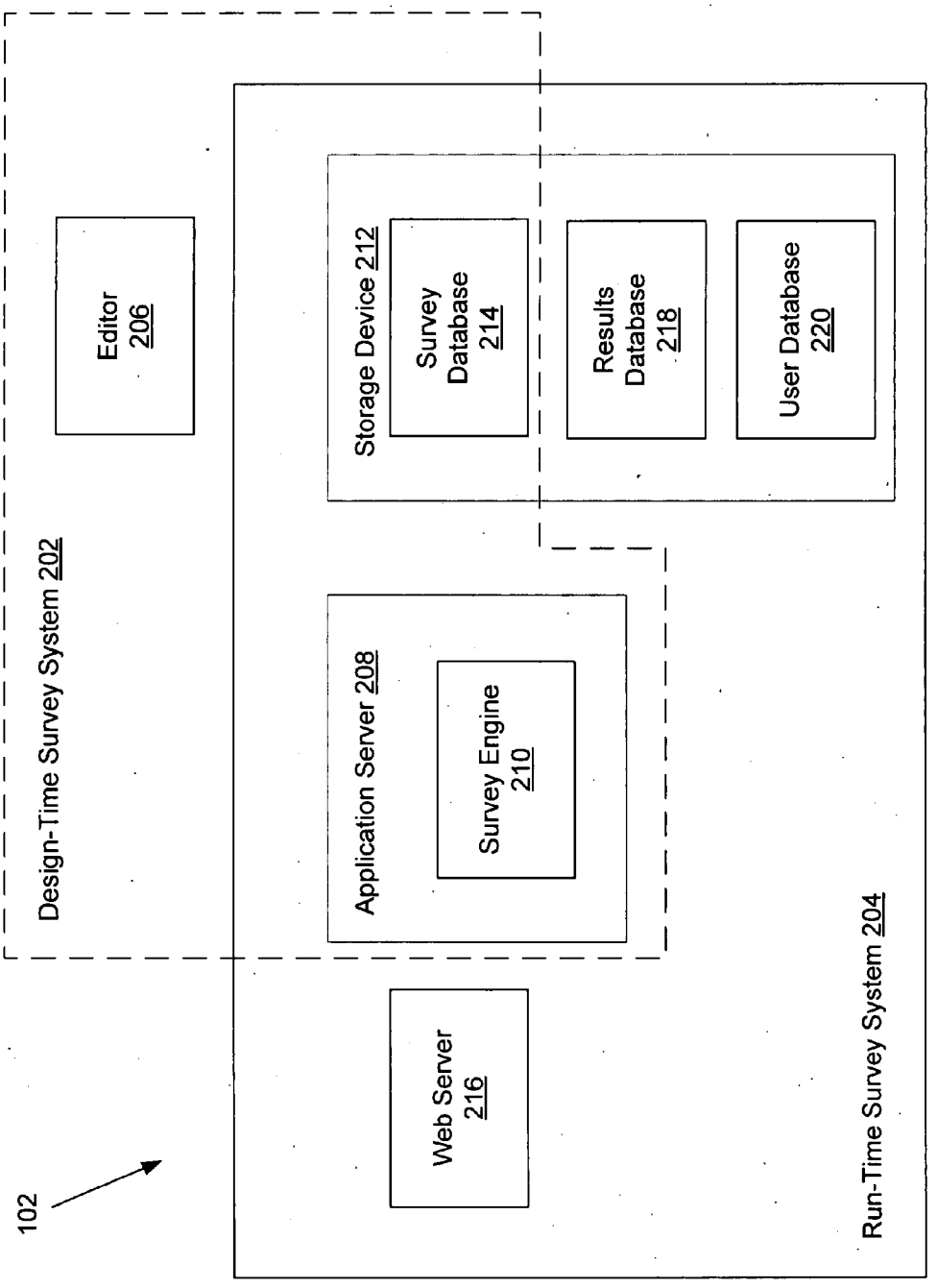

102
Design-Time Survey System 202
Editor 206
Application Server 208
Survey Engine 210
Run-Time Survey System 204
Web Server 216
Storage Device 212
Survey Database 214
Results Database 218
User Database 220

FIG. 2

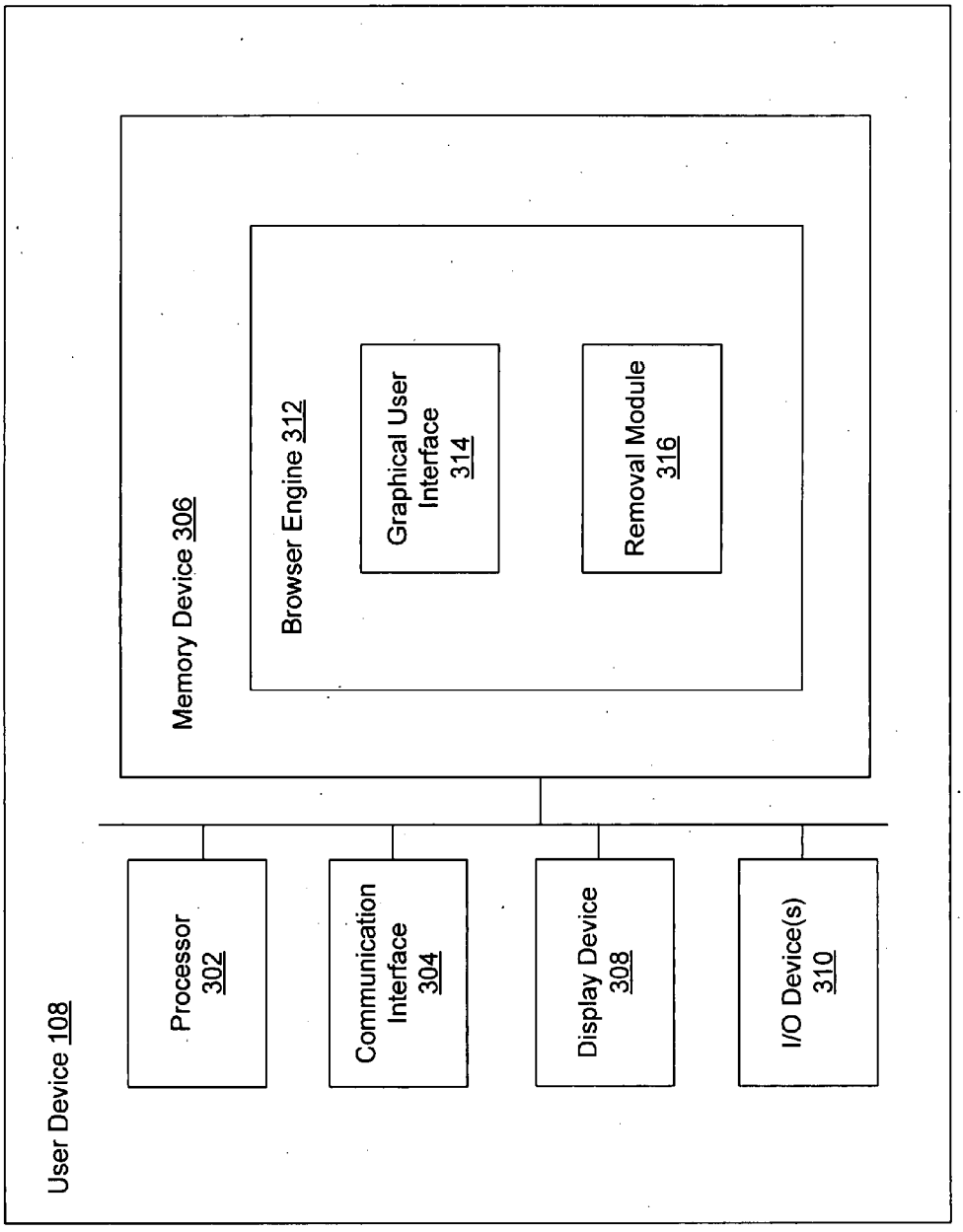

User Device 108
Processor 302
Communication Interface 304
Display Device 308
I/O Device(s) 310
Memory Device 306
Browser Engine 312
Graphical User Interface 314
Removal Module 316

FIG. 3

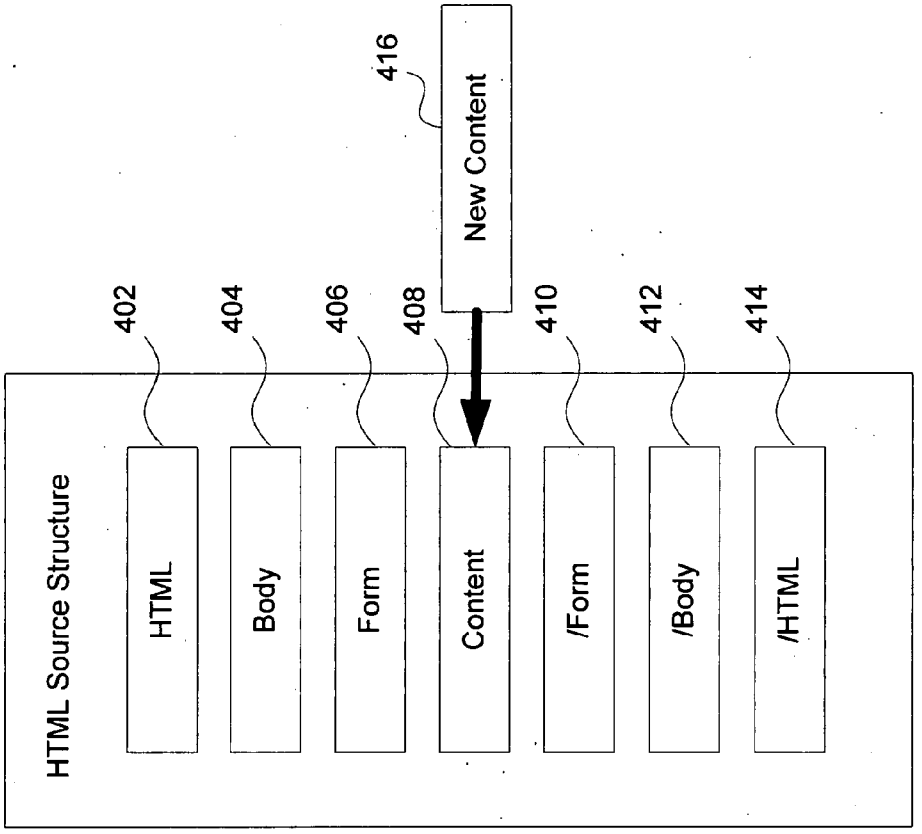

HTML Source Structure
HTML
Body
Form
Content
/Form
/Body
/HTML
New Content
402
404
406
408
410
412
414
416

FIG. 4b

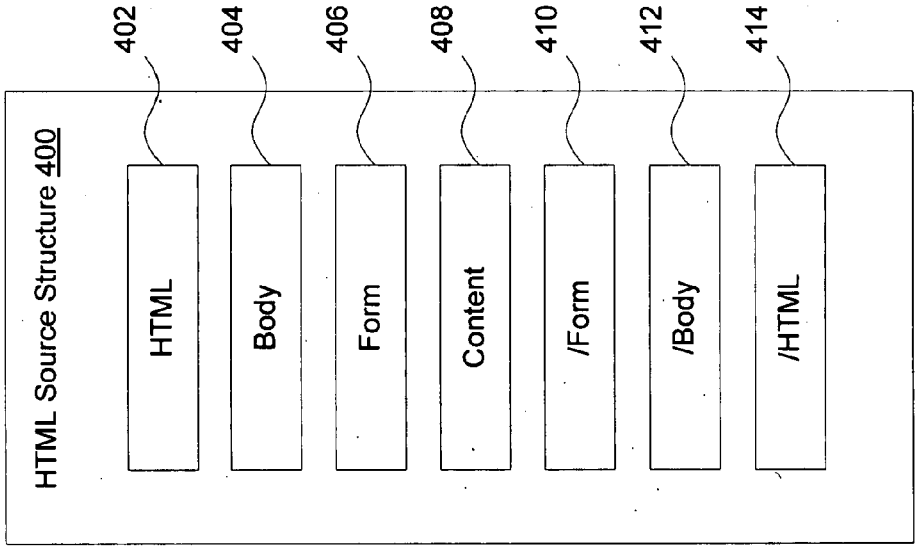

HTML Source Structure 400
HTML
Body
Form
Content
/Form
/Body
/HTML
402
404
406
408
410
412
414

FIG. 4a

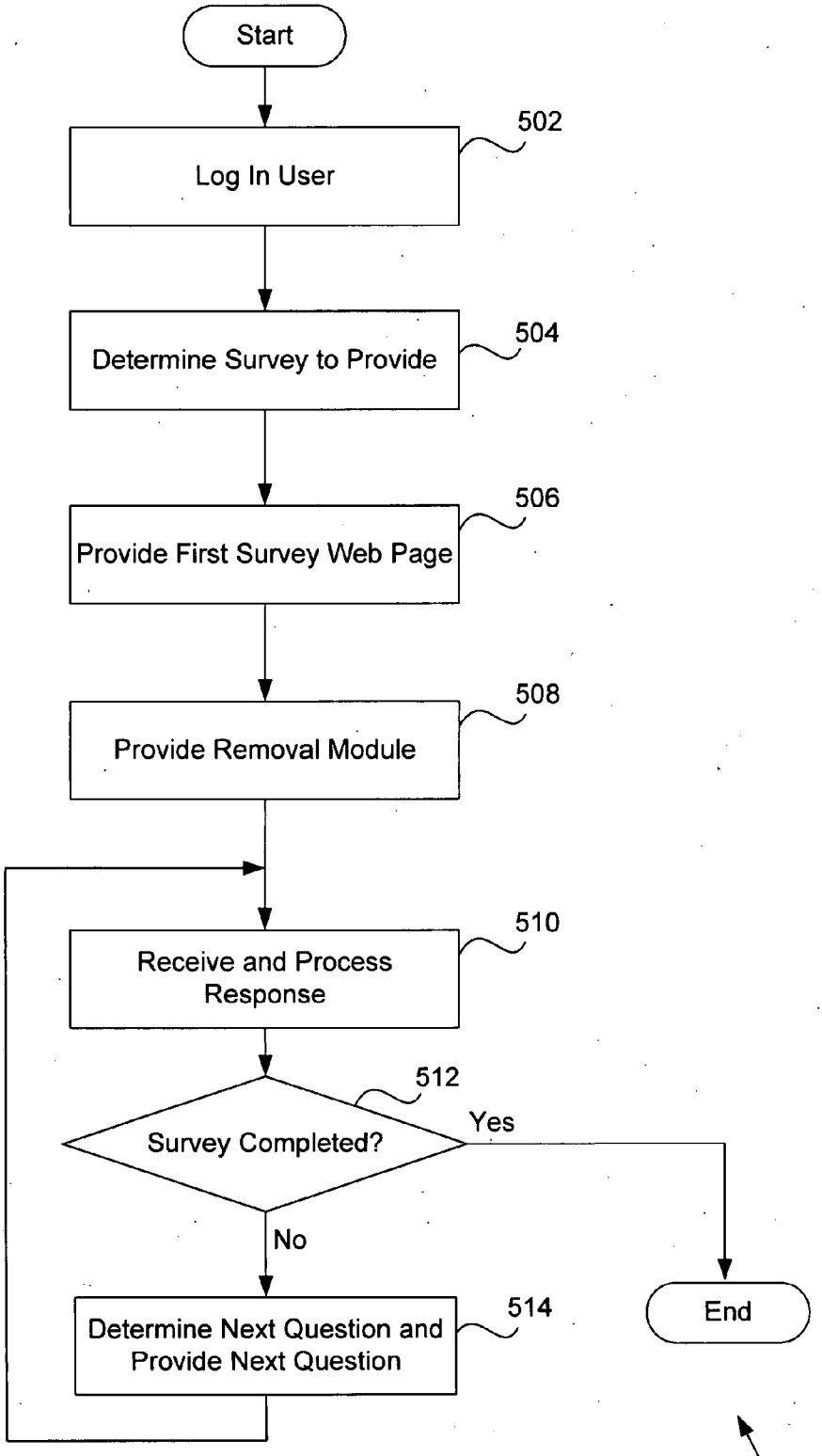

Start
Log In User
502
Determine Survey to Provide
504
Provide First Survey Web Page
506
Provide Removal Module
508
Receive and Process Response
510
Survey Completed?
512
Yes
No
Determine Next Question and Provide Next Question
514
End
500

FIG. 5

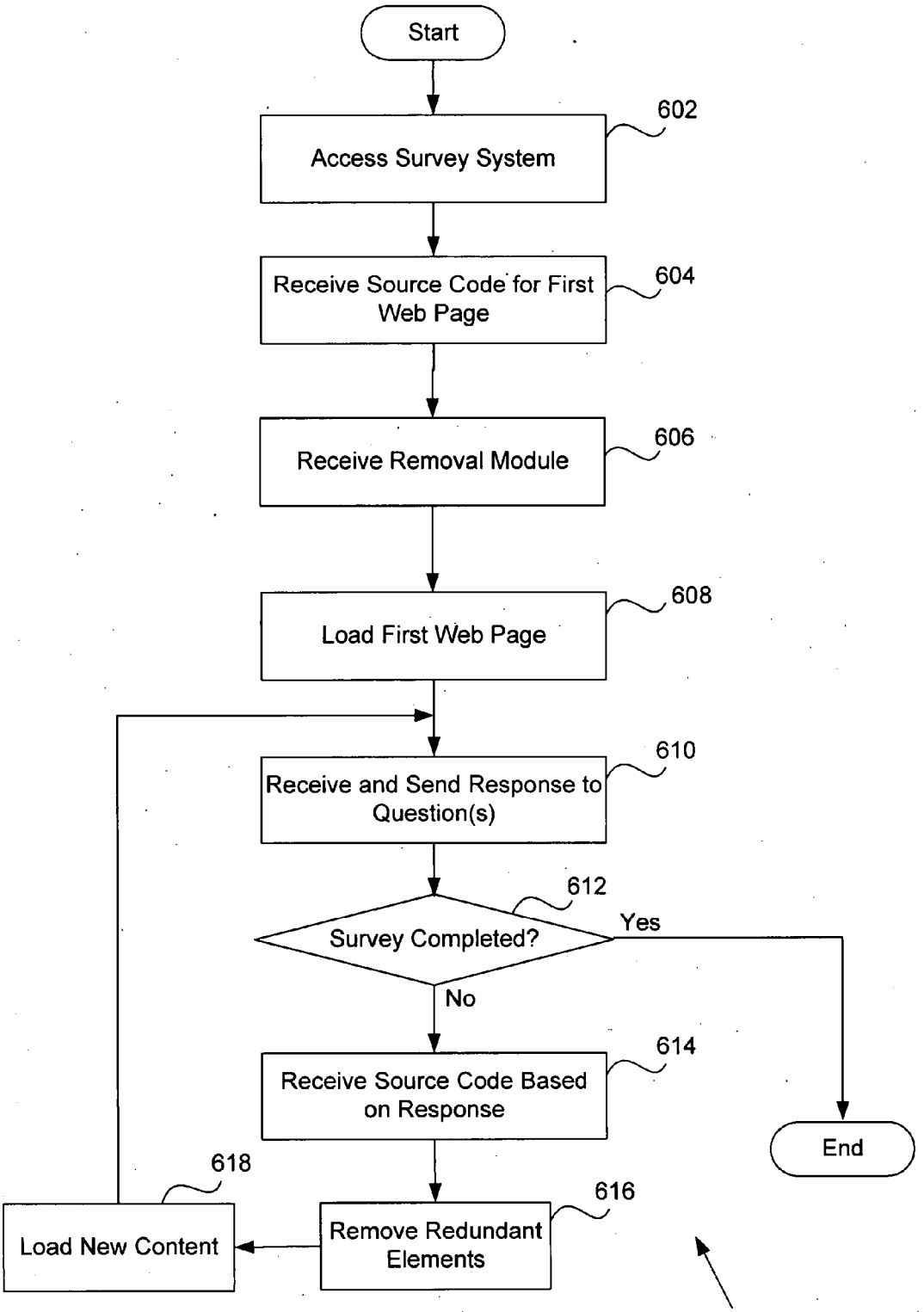

Start
Access Survey System
602
Receive Source Code for First Web Page
604
Receive Removal Module
606
Load First Web Page
608
Receive and Send Response to Question(s)
610
Survey Completed?
612
Yes
No
Receive Source Code Based on Response
614
End
Remove Redundant Elements
616
Load New Content
618
600

FIG. 6

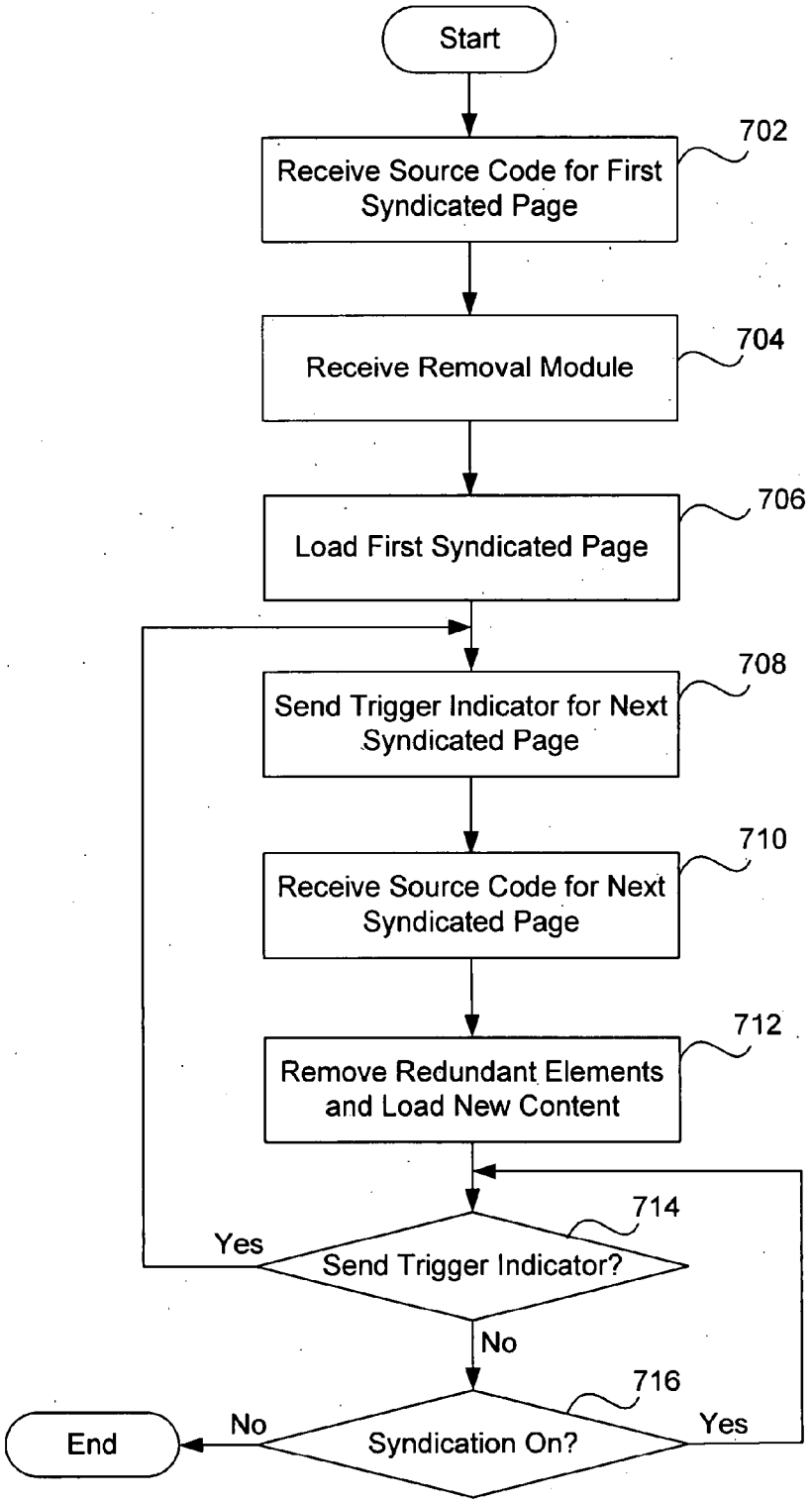

Start
Receive Source Code for First Syndicated Page
702
Receive Removal Module
704
Load First Syndicated Page
706
Send Trigger Indicator for Next Syndicated Page
708
Receive Source Code for Next Syndicated Page
710
Remove Redundant Elements and Load New Content
712
Send Trigger Indicator?
714
Yes
No
Syndication On?
716
No
Yes
End
700

FIG. 7

SYSTEM AND METHOD FOR EXPEDITING INFORMATION DISPLAY

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] Embodiments of the present invention relate generally to expediting information display, and more particularly to expediting display of new content on web pages.

[0003] 2. Description of Related Art

[0004] Conventionally, content is provided by a content provider over a network. In survey embodiments, surveys may be provided to survey participants via the Internet. Typically, the survey participants go to a website to access one or more surveys. The survey is then provided to a browser associated with each survey participant.

[0005] Bi-directional communication of HTTP traffic occurs during a survey process. HTML source code is provided to the survey participant's browser which enables display of a survey page comprising one or more questions. The survey participant provides answer(s) to the one or more questions and submits the answer(s). The browser performs a form post and sends the answer(s) to a web server and/or application server providing the survey. Based on the answer (s), the application server determines a next set of one or more questions. The HTML source code is generated for the next set and sent back to the browser. This process continues until the survey is complete. Additionally, survey testers may also access surveys via a website or network connection in a similar manner.

[0006] Conventionally, the HTML source code returned by the application server will consist of code for an entire survey web page. That is the returned HTML source code will include body tags, form tags, and content. As a result, a full web page is loaded by the browser every time a next set of questions is provided even though only the content (i.e., the questions, themselves) may have changed.

[0007] An amount of time a survey participant is interested in responding to a survey is typically short. If a survey is too long, the survey participant may lose interest and stop responding. Thus, load time of a survey web page becomes an important factor for surveys that are longer in length. Additionally, as surveys are likely using "wizards" where a user has to go screen by screen, the surveys may be tedious to test. Therefore, it is desirable to be able to reduce the load time of web pages and desirable to be able to navigate to different parts of a survey.

SUMMARY OF THE INVENTION

[0008] Embodiments of the present invention provide systems and methods for expediting information display on a user device. In exemplary embodiments, a request for information is received. The information may be associated, in some embodiments, with a survey. In other embodiments, the information is associated with a syndication web page. The request for the information may comprise a user's name and password or comprise a trigger indicator for new or additional information. Alternatively, for example, the survey may be accessed via a link in a survey invitation received by the survey participant.

[0009] An application server then generates and provides source code for each web page of the selected information. In one embodiment, the source code may comprise HTML source code. In exemplary survey embodiments, a survey engine of the application server will generate the source code based on survey subject matter and an edited survey template associated with the selected survey. The survey template may be generated via an editor.

[0010] In exemplary embodiments, a removal module is returned with source code for the first web page. The removal module is configured to remove unnecessary or redundant elements from subsequent source code in order to expedite building and displaying of subsequent web pages on the user device in situ, for example, without the appearance of a page being loaded. In an alternative embodiment, the removal module is not needed if the source code is based on XML formatted data which only consists of new content.

[0011] Upon a subsequent request for information, source code for the new information is received. Because many elements of the source code are redundant or unnecessary (e.g., repetitive of what has been received in the past such as headers and background elements), information display would be expedited if these elements are not reloaded onto a web page that already has such elements loaded. As such, the removal module will remove these redundant or unnecessary elements. What remains is the new content which is loaded onto the web page. In one embodiment, the web page comprises a syndication page comprised within a main web page.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. **1** is a diagram of an environment in which embodiments of the present invention may be practiced;

[0013] FIG. **2** is a block diagram of an exemplary survey system;

[0014] FIG. **3** is a block diagram of an exemplary user device;

[0015] FIG. **4***a* is a block diagram of an exemplary HTML source structure, according to one embodiment;

[0016] FIG. **4***b* illustrates an operation of an embodiment of the present invention;

[0017] FIG. **5** is a flowchart of an exemplary method for providing a survey to a requesting user.

[0018] FIG. **6** is a flowchart of an exemplary method for providing the survey on a user device.

[0019] FIG. **7** is a flowchart of an exemplary method for providing syndicated information on the user device.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0020] Embodiments of the present invention provide a system and method that allows faster web page loads by limiting the load/refresh to new content. In exemplary embodiments, portions of a web page that do not change (e.g., a header, background design, etc.) will not be reloaded by a browser. While some embodiments of the present invention will be discussed in connection to surveys, the present invention may be utilized in non-survey environments whereby only a portion of a web page needs to be updated. The present invention may also be applied to syndication in a window of a main web page whereby any new content is loaded into the window according to exemplary embodiments as described herein.

[0021] Referring to FIG. **1**, an exemplary survey environment **100** in which embodiments of the present invention may be practiced is shown. The environment **100** comprises a survey system **102** configured to provide surveys based on information received from a survey client **104**. The survey

client **104** is an entity (e.g., business) that is interested in having some aspect of their product(s) or service(s) reviewed via a survey. In exemplary embodiments, the survey client **104** forms a relationship with a survey company associated with the survey system **102** such that the survey company will, based on information received from the survey client **104**, generate the survey, determine an appropriate set of users (e.g., survey participants) to take the survey, and collect survey results. In some embodiments, the survey company may also test the survey. The survey company then provides the results to the survey client **104**. Any number of survey clients **104** may be present in the environment **100**.

[0022] In some embodiments, the survey client **104** interacts with the survey company via a network **106**. The network **106** may comprise the Internet. In other embodiments, the survey client **104** and the survey company may interact via other networks **106** or communication means such as via telephone, in person, or the interaction may involve a combination of communication means.

[0023] A plurality of user devices **108** are also coupled to the survey system **102** via the network **106**. In some embodiments, the user devices **108** are associated with a survey participant. In other embodiments, the user devices **108** are associated with a survey tester or reviewer. As such, the user device **108** may be associated with any user that wants to access a survey. The user device **108** may comprise a personal computer, mobile communication device, personal digital assistant, or any other type of computing device. Any number of user devices **108** may be present in the environment **100**. In exemplary embodiments, the users access a website associated with the survey company. The users are identified by the survey system **102**, and an appropriate survey is provided to a browser of the user device **108** associated with the user. In alternative embodiments, the user (e.g., a tester), and thus the user device **108**, may be embodied within the survey company or survey client **104**.

[0024] In exemplary embodiments of the present invention, a first survey web page (e.g., web page comprising survey questions) is entirely loaded onto the user device **108**. Subsequent survey web pages may only load new content, as will be described in more details below.

[0025] Referring now to FIG. **2**, the survey system **102** is shown in more detail. In exemplary embodiments, the survey system **102** comprises a design-time survey system **202** and a run-time survey system **204**. The design-time survey system **202** provides components used to create a survey. In contrast, the run-time survey system **204** provides components used to provide the created survey to users. These two survey systems **202** and **204** may share common components.

[0026] In exemplary embodiments, the design-time survey system **202** comprises an editor **206** and an application server **208**. The editor **206** is configured to process survey content into a form of a survey template which is then provided to a survey engine **210** of the application server **208**. According to one embodiment, the survey content is scripted, via the editor **206**, into the survey template for use by the survey engine **210**. The survey template may be stored in a storage device **212**. The survey content may comprise questions for the survey based on data received from the survey client **104**.

[0027] The editor **206** may also make updates to the surveys. In exemplary embodiments, the updates are made to the survey engine **210** which may then affect a survey database **214**. Alternatively, the editor **206** may make updates directly to the survey database **214**. These edits to the survey engine **210** or survey database **214** may comprise updates to the survey. For example, if a question is changed, an update to the survey database **214** may be made to edit columns to match the new questions. In exemplary embodiments, every column in the survey database **214** is associated with a question or a variable location.

[0028] The run-time system **204** comprises a web server **216**, the application server **208** along with the survey engine **210** and the storage device **212**. The storage device **212** may further comprise a results database **218** and a user database **220**.

[0029] The exemplary web server **216** is configured to deliver web page content to browsers and other files to applications via, for example, a HTTP protocol. In some embodiments, the web server **216** may comprise software, and as such, may be a HTTP server (e.g., IIS, Apache, etc.) that manages requests from the browser and delivers HTML documents and files (e.g., HTML source code) in response. The .web server **216** may also execute server-side scripts (e.g., CGI scripts, JSPs, ASPs, etc.). Alternative embodiments may utilize other protocols. While the following discussion is provided with respect to the use of HTML source code, alternative embodiments may utilize other source code formats.

[0030] In some embodiments, the web server **216** provides a removal module to the user device **108** which removes unnecessary or redundant portions of source code provided to the user device **108**, so that only new content will be loaded onto subsequent web pages associated with the survey. In one embodiment, the removal module comprises an AJAX (Asynchronous JavaScript and XML) type module. Alternative embodiments may utilize other scripting language modules to remove redundant or unnecessary portions of the source code. In yet other embodiments, only new content is returned to the user (e.g., using XML).

[0031] The exemplary application server **208** is configured to host a variety of language systems used to program database queries and/or general processing. These scripts and services, such as JavaScript and Java server pages (JSPs), typically access databases in a storage device **212** to retrieve up-to-date data (e.g., a next set of one or more survey questions) to present to the user. The survey data may be stored in the survey database **214**.

[0032] The exemplary survey engine **210** takes the survey content and survey template from the editor **206** and generates source code representing a web page, which is sent to the user upon request. The survey engine **210** may also receive responses from the user and store the responses in the storage device **212**. Based on the responses, the survey engine **210** determines a next set of one or more questions to be provided to the user. The corresponding source code is then generated by the survey engine **210** for the next set of questions and forwarded to the user device **108** associated with the survey participant.

[0033] In some embodiments, the survey engine **210** may comprise a plurality of routing and logic which transforms the survey content received from the editor **206** into the source code. This source code may comprise coding which may not be necessary for the purpose of presenting a next set of questions as described in accordance with embodiments of the present invention. For example, if the background and header of the web page does not change from one web page to the next, then only new content (e.g., a new set of questions)

needs to be loaded onto the web page in order to reduce load or refresh time. This process will be discussed in more detail below.

[0034] In some embodiments, the editor **206** and the survey engine **210** may be under the control of two different entities. For example, the editor **206** may be associated with the survey company, while the survey engine **210** may be associated with a survey services provider. In these embodiments, the survey company has no control as to the operation of the survey engine **210** or the output of the survey engine **210**. As such, the survey company utilizes the editor **206** to develop templates that will enable the functionalities described in connection with embodiments of the present invention.

[0035] In an alternative embodiment, the survey engine **210** may comprise a logic driven survey engine **210** which outputs in an XML format. In some of these embodiments, a conversion of the XML output to HTML format may be required. For example, a XSLT processor may be utilized to transform the XML output of the survey engine **210** into HTML format for delivery to the user device **108**. According to exemplary embodiments, the XML-based data consists only of the new content (i.e., there are no redundant elements).

[0036] The exemplary storage device **212** is configured to store data associated with surveys provided by the survey system **102**. The survey database **212** comprises data associated with the survey and survey questions. In some embodiments, the survey database **212** may store the templates that are generated by the editor **206** and provided to the survey engine **210**. The results database **218** may store answers to the surveys. In some embodiments, the results database **218** may include processed survey results. For example, the answers to a particular survey may be processed to provide an average and median score or result for each survey question.

[0037] The user database **220** comprises user information. The user information may be utilized to verify users (e.g., login data), determine which users should respond to a survey, and send an invitation to selected users, for example. Alternative embodiments may comprise other databases or combine databases (e.g., the survey database **214** and the results database **218**).

[0038] Referring now to FIG. **3**, a block diagram of the exemplary user device **108** is shown. The user device **108** may comprise a processor **302**, a communication interface **304**, and at least one memory device **306**. The user device **108** further comprises other conventional components, such as a display device **308** and input and output devices **310**. The communication interface **304** allows the user device **108** to communicate with the network **106**. The memory device **306** may be any storage medium type, and may comprise a browser engine **312**. The exemplary browser engine **312** may be any type of application or engine that allows the user device **108** to interact with the survey system **102** or any other content provider via the network **106**.

[0039] The browser engine **312** may further comprise a graphical user interface **314** and a downloaded removal module **316**. The exemplary graphical user interface **314** comprises a graphical user interface which is configured to provide a graphical representation of the survey to the display device **208** of the user device **108**. The graphical user interface **314** may be further configured to receive inputs (e.g., survey answers/responses) from a user via the I/O devices **310**. The browser engine **312** may comprise other conventional components necessary for the operation of the browser on the user device **108**. These conventional components will not be discussed as one skilled in the art will understand these functionalities.

[0040] In exemplary embodiments, the removal module **316** is downloaded from the survey system **102** at a beginning of a survey or from a content provider with a first web page from the content provider. For example, the removal module **316** may be downloaded with a first web page associated with the survey. This first web page may be an introductory web page or a first web page of the survey. In some embodiments, the removal module **316** may comprise JavaScript. Alternative embodiments may comprise a removal module **316** using other forms of scripting language. In yet other embodiments, the removal module **316** may be provided from other sources or via other means such as a compact disc.

[0041] The exemplary removal module **316** is configured to intercept communications being exchanged between the browser engine **312** and the network **106**. However, the browser engine **312**, the web server **216**, and/or the application server **208** function normally as if the removal module **316** is not present. Thus, when a user submits responses to a set of one or more survey questions, the removal module **316** intercepts the submit request and performs a POST operation while preventing the browser engine **312** from performing a POST itself.

[0042] In exemplary embodiments, the removal module **316** prevents the browser engine **312** from loading a completely new web page when only select content on the web page changes. The removal module **316** does this by intercepting the source code received from the survey system **102** or content provider, parsing the source code, and striping out portions of the source code that are not necessary or redundant for a subsequent web page. These unnecessary or redundant portions may comprise areas of the web page that do not change from one web page to a subsequent web page (e.g., header, background, etc.) and/or source code that is redundant.

[0043] In embodiments where a user stops a survey before completion, a save point is stored in the storage device **212**. This save point comprises a marker that may return the user back to a previous point in the survey. By using a single survey template to present survey questions throughout the survey, the user may be returned to the save point in the survey including styling and format of non-changing portions of the web page.

[0044] Referring now to FIG. **4***a*, an exemplary HTML source structure **400** is shown. The HTML source structure **400** represents the HTML source code that is loaded for each web page. As shown, the HTML source structure **400** comprises a HTML tag **402**, a body tag **404**, a form tag **406**, content **408**, a form closing tag **410**, a body closing tag **412**, and a HTML closing tag **414**. One skilled in the art will understand the usage of these various tags **402**, **404**, **406** and closing tags **410**, **412**, and **414**. As such, no further discussion will be provided with respect to these elements.

[0045] After the initial loading of the HTML source code for the first web page, subsequent web pages may only have a change in the content. As such, the tags **402**, **404**, **406** and closing tags **410**, **412**, and **414** are merely, duplicates of the HTML source code of the first web page. Therefore, these repeated tags **402**, **404**, **406** and closing tags **410**, **412**, and **414** may be unnecessary in a load or refresh of the subsequent web page.

[0046] The exemplary removal module **316** is configured to intercept HTML source code of a subsequent web page, parses the HTML source code, and removes the unnecessary/duplicate elements of the HTML source code such that only new content (e.g., new content **416** of FIG. **4***b*) is loaded to create the next web page. In exemplary survey embodiments, the survey engine **210** will provide the same HTML source structure for each subsequent survey web page of the survey based on the survey template. With this knowledge, the removal module **316** may remove any duplicate elements prior to and including the form tag **406** and any duplicate elements after and including the closing form tag **410**. What remains is the new content **416**. This new content **416** may then be passed to the graphical user interface **314** which generates the next survey web page to display to the survey participant by replacing the previous content **408**. As such, the graphical user interface **314** does not rebuild the entire HTML source structure of the subsequent survey web page, but limits the rebuild to the new content **416**. Therefore, there is no browser refresh, and the associated time to load the next set of one or more questions is reduced.

[0047] While embodiments of the present invention have been described utilizing a removal module **316** for removing redundant elements from subsequent HTML source code, alternative embodiments may utilizing other source structures and other forms of remote scripting modules for removing redundant elements from the source structure.

[0048] In some embodiments, a transition effect may be utilized to display the next web page. By using transition effects, an interest level of the survey participant may be maintained. Additionally, the transition effect may make the load of the new content **416** less perceivable to the survey participant. Examples of transition effects include cover left/right, fade in/out, box in/out, and so forth. Any type of transition effect may be used in various embodiments of the present invention.

[0049] Referring now to FIG. **5**, a flowchart **500** of an exemplary method for providing a survey to a user is shown. In step **502**, the user is logged in with the survey system **102**. In one embodiment, the user may provide a user name and password to log in with the survey system **102**. In another embodiment, the user may access the survey system **102** via a link provided on a communication (e.g., survey invitation sent by the survey company via e-mail). In this embodiment, the user's identity may be determined based on the accessed link or information automatically returned when the link is activated.

[0050] In step **504**, the survey system **102** determines which survey to provide the user. For example, upon receiving the user name and password from the user, the application server **208** may access the survey database **214** to determine which survey(s) the user is eligible to take or allowed to access. If more than one survey is applicable, the user may be asked to make a selection. In the embodiment where the user accesses the survey system **102** via the survey invitation link, the application server **208** may, via the link or information automatically returned, determine a specific survey to provide to the user.

[0051] A first survey web page is provided to the user device **108** associated with the user in the form of source code in step **506**. In some embodiments, the first survey web page presents a first set of one or more questions to the survey participants. Along with the first survey web page, the removal module **316** may be provided by the web server **216** in step **508**. In embodiments that utilize XML in order to provide updated content, the removal module **316** is not needed. It should be noted that steps **506** and **508** may occur substantially simultaneous or in parallel.

[0052] In some embodiments, an introductory web page may precede the first survey web page. This introductory web page may comprise an explanation or purpose associated with the survey, thank the survey participants for their time and cooperation, and/or describe incentives for completing the survey. In these embodiments, the removal module **316** may be provided with the introductory web page. A next web page of the survey may then comprise the first set of questions (i.e., the first survey web page).

[0053] A response or a set of responses is then received from the user device **108** based on a current set of questions being displayed to the user in step **510**. For example, if the first survey web page is currently being provided, then the response is for the first set of questions from the first survey web page.

[0054] A determination is made in step **512** as to whether the survey is completed. If the survey is not completed, then the survey engine **210** will determine a next set of one or more questions to present to the user in step **514**. The next set of questions may be based on the received response or set of responses. Thus, the survey engine **210** utilizes logic to determine a next set of content based of a previous response. The survey engine **210** then provides the next set of one or more questions to the user, and waits for a next set of responses. The process is repeated until the survey is completed in step **512**. In some embodiments, a last web page of the survey may comprise set up and content that differs from the survey web pages. For example, the last web page may present a thank you message or a survey complete message with a different style format than that of the survey web pages.

[0055] FIG. **6** illustrates a flowchart **600** of an exemplary method for providing a survey display on the user device **108**. In step **602**, the user accesses the survey system **102**. In one embodiment, the user may, via their user device **108**, access a website associated with the survey company and provide a user name and password to log in with the survey system **102**. In another embodiment, the user may access the survey system **102** via a link provided on a communication (e.g., e-mail) received from, for example, the survey company.

[0056] Upon determining a survey that the user is eligible to participant in or has permission to access, source code of a first survey web page is received by the user device **108** in step **604**. In step **606**, the removal module **316** is received at the user device **106**. In some embodiments, steps **604** and **606** may occur in parallel, substantially simultaneously, or in reverse order. The removal module **316** may be stored in the memory device **306** of the user device **108** for subsequent usage during the survey process. In embodiments utilizing XML, the removal module **316** may not be needed.

[0057] In some embodiments, an introductory web page may precede the survey web pages. In these embodiments, the removal module **316** may be received with, or substantially simultaneous with, the introductory web page source code. If the style formatting of the introductory web page is different than that of the survey web page (e.g., different header, background, placement of content, etc.), then the introductory source code along with the source code for the first survey web page will both be fully loaded by the browser engine **312**. If the only difference between the introductory web page and

the survey web page is content, then the survey web page may be provided by implementing the expedited display process as discussed herein.

[0058] In step **608**, the first survey web page may be graphically displayed to the user. In exemplary embodiments, the graphical user interface **314** processes the source code received from the survey system **102** and loads the entire first survey web page.

[0059] As described in connection with FIG. **4***a*, the source code of the first survey web page may comprise a plurality of tags **402**, **404**, **406**, content **408**, and closing tags **410**, **412**, and **414**. In some embodiments, the first web page comprises a set of one or more survey questions.

[0060] The user responds to the set of survey questions and submits the responses in step **610**. In some embodiments, the removal module **316** will intercept the submit request and perform a post to the survey system **102**.

[0061] If the survey is completed in step **612**, then the process ends. However, if the survey is not completed, then the user device **108** will receive the next source code in step **614**. The removal module **316** may intercept the next source code prior to the source code reaching the graphical user interface **314**. In step **616**, the removal module **316** will, according to the process described in connection with FIG. **4***a* and FIG. **4***b*, remove unnecessary or redundant elements from the source code. In exemplary embodiments, the removed elements may comprise various tags **402**, **404**, **406** and closing tags **410**, **412**, and **414**. If the source code is based on XML data, then step **616** is not needed since the XML data is configured to provide only the new content.

[0062] After the removal of unnecessary or redundant code/elements, the remaining content (e.g., new content **416**) is provided to the graphical user interface **314**. The graphical user interface **314** then builds a next survey web page with the new content **416** and provides the next survey web page to the survey participant in step **618**. In some embodiments, the new content comprises the next set of questions of the survey. The survey process continues until the survey is completed in step **612**.

[0063] In some embodiments, a last web page of the survey may comprise set up and content that differs from the survey web pages. For example, the last web page may present a thank you message or a survey complete message with a different style format than that of the survey web pages. In these embodiments, the removal module **316** may remove little or no elements from the source code, and the source code for this last web page may be fully loaded by the browser engine **312**.

[0064] Referring now to FIG. **7**, a flowchart **700** of an exemplary method for providing a syndicated page using embodiments of the present invention is provided. In step **702**, source code for a first syndication page is received by the user device **108**. In exemplary embodiments, the syndication page may be provided within a window of a main web page. The syndication page may provide a web page, content, or other information from a website or content provider that is different from the main web page. In one example, the source code for a first syndication page may be source code that contains or links to content from a content provider that is different from that of the main web page.

[0065] In step **704**, the removal module **316** is received at the user device **108**. In some embodiments, steps **702** and **704** may occur in parallel, substantially simultaneously, or in reverse order. The removal module **316** may be stored in the memory device **306** of the user device **108** for subsequent usage. In embodiments utilizing XML, the removal module **316** may not be needed. It should be noted that if the main web page also utilizes the expedited information display mechanisms of the present invention, then the removal module **316** may be provided to the user device **108** with source code for a first page of the main web page. In these embodiments, step **704** may be unnecessary.

[0066] In some embodiments, the removal module **316** may be retrieved from memory of the user device **108**. In one example, a browser of the user device **108** may have previously visited a website that downloaded the removal module **316** to the user device **108**. In various embodiments, prior to downloading the removal module **316**, the browser may be instructed (e.g., by the downloaded source code) to determine if a removal module **316** had been previously downloaded and is available (e.g., in a cache). If the removal module **316** is available, the browser may be further instructed to determine if the removal module **316** is current (e.g., the correct version) and/or if the removal module **316** is the correct type (e.g., there may be multiple types of removal modules **316** which may work with different pages, web servers, software, and/or technologies).

[0067] In step **706**, the first syndicated page is graphically displayed to the user. In exemplary embodiments, the graphical user interface **314** processes the source code received from the content provider and loads the entire first syndicated page. As described in connection with FIG. **4***a*, the source code of the first syndicated page may, for example, comprise a plurality of tags **402**, **404**, **406**, content **408**, and closing tags **410**, **412**, and **414**.

[0068] In step **708**, a trigger indicator is sent requesting a next syndicated page from the content provider. In some embodiments, the trigger indicator may comprise a manual request initiated by the user of the user device **108** for the next syndicated page. In other embodiments, the trigger indicator may comprise a request generated based on a predetermined amount of lapsed time. For example, the trigger indicator may be sent in order to update the syndicated page every ten minutes.

[0069] Based on the trigger indicator, source code for a next syndicated page is received from the content provider in step **710**. The source code may comprise redundant or unnecessary elements with respect to a currently loaded syndicated page. As such in step **712**, the removal module **316** will, according to, for example, the process described in connection with FIG. **4***a* and FIG. **4***b*, remove unnecessary or redundant elements from the source code. In exemplary embodiments, the removed elements may comprise various tags **402**, **404**, **406** and closing tags **410**, **412**, and **414**. If the source code is based on XML data, then step **616** may not be needed since the XML data may be configured to provide only the new content.

[0070] After the removal of unnecessary or redundant code/elements, the remaining content (e.g., new content **416**) is provided to the graphical user interface **314**.

[0071] The graphical user interface **314** then builds a syndicated web page with the new content and provides the next syndicated web page to the user.

[0072] In step **714**, a determination is made as to whether another trigger indicator should be sent. For example, if a predetermined period of time has past or a user selects to manually send a trigger indicator (e.g., by the user selecting

"reload" within the browser), then a next trigger indicator will be sent in step **708**. The process continues until the syndication is inactivated in step **716**.

[0073] In various embodiments, when first accessed by the user, only a portion of the syndication page may load. In one example, the source code for a main page may be loaded by a browser on the user device **108**. The source code for the main page may include instructions to draw a window configured to contain content from another content provider (e.g., headlines from a news website). The source code may contain a link to content from the other content provider as well as the removal module **316**. The browser may activate the link to download an entire web page (e.g., the syndication page) from the other content provider. The removal module **316** may then remove unnecessary or redundant elements from the downloaded web page (e.g., the source code of the web page) from the other content provider as described herein. The user interface **314** may display the main page and the remaining content of the web page from the other content provider within the window.

[0074] Those skilled in the art will appreciate that this process may work in conjunction with any kind of software or technology and is not limited to syndication. This process may be used with, but limited to, RSS feeds, network feeds, blogging, sports data, traffic information, video, images, audio, and news. Further, the removal module **316** may assist in displaying content or organizing information from any number of content providers on one or more web pages.

[0075] The above-described functions and components can be comprised of instructions that are stored on a storage medium. The instructions can be retrieved and executed by a processor. Some examples of instructions are software, program code, and firmware. Some examples of storage medium are memory devices, tape, disks, integrated circuits, and servers. The instructions are operational when executed by the processor to direct the processor to operate in accord with embodiments of the present invention. Those skilled in the art are familiar with instructions, processor(s), and storage medium.

[0076] The present invention has been described above with reference to exemplary embodiments. It will be apparent to those skilled in the art that various modifications may be made and other embodiments can be used without departing from the broader scope of the invention. Therefore, these and other variations upon the exemplary embodiments are intended to be covered by the present invention.

What is claimed is:

1-15. (canceled)

16. A method for expediting information display on a user device, comprising:

receiving and loading source code for an entire web page associated with a request for information from the user device;

receiving a removal module;

receiving additional source code comprising new content; and

removing unnecessary elements from the additional source code such that only new content is loaded into the web page without a full web page load.

17. The method of claim **16** further comprising providing a request for the additional source code.

18. The method of claim **17** wherein the request comprises a time out trigger based on a predetermined expiration of time.

19. The method of claim **17** wherein the response comprises a request for the new content based on a manual trigger by the user.

20. The method of claim **17** wherein the response comprises a response to one or more questions of a survey.

21. The method of claim **16** wherein a syndicated web page comprises the web page additional source code.

22. A machine readable medium having embodied thereon a program, the program having instructions operable by a machine for expediting information display on a user device, comprising:

receiving and loading source code for an entire web page associated with a request for information from the user device;

receiving a removal module;

receiving additional source code comprising new content; and

removing unnecessary elements from the additional source code such that only new content is loaded into the web page without a full web page load.

23. The method of claim **20** wherein source code for the second entire web page comprises a second set of one of more questions based on the response to the one or more questions of the survey.

* * * * *