US 20120151329A1

(19) United States

(12) Patent Application Publication

Cordasco

(10) Pub. No.: US 2012/0151329 A1

(43) Pub. Date: Jun. 14, 2012

(54) **ON-PAGE MANIPULATION AND REAL-TIME REPLACEMENT OF CONTENT**

(75) Inventor: **Matthew Cordasco**, San Francisco, CA (US)

(73) Assignee: **TeaLeaf Technology, Inc.**, San Francisco, CA (US)

(21) Appl. No.: **13/401,725**

(22) Filed: **Feb. 21, 2012**

Related U.S. Application Data

(63) Continuation-in-part of application No. 12/750,607, filed on Mar. 30, 2010.

(60) Provisional application No. 61/445,528, filed on Feb. 22, 2011.

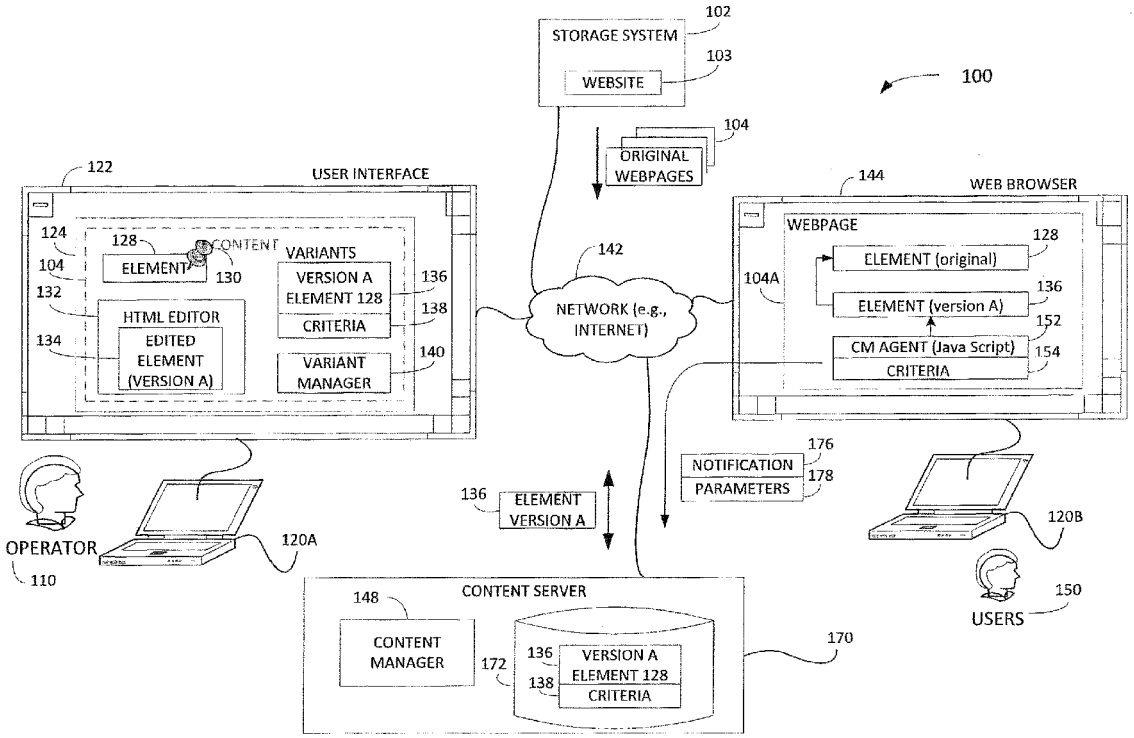
Publication Classification

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/234**

(57) **ABSTRACT**

A request to render a control content is intercepted by a client application as it is being served from a content source. Based on predetermined criteria, the control content is replaced with replacement content. The replacement content may be an alternative version of the control content differing from the control content in one or more alternative elements. The replacement content may be created with a graphical interface that overlays the control content. The graphical interface allows a user to select elements of the control content, edit the selected elements of the control content, and save the edited elements as part of the replacement content.



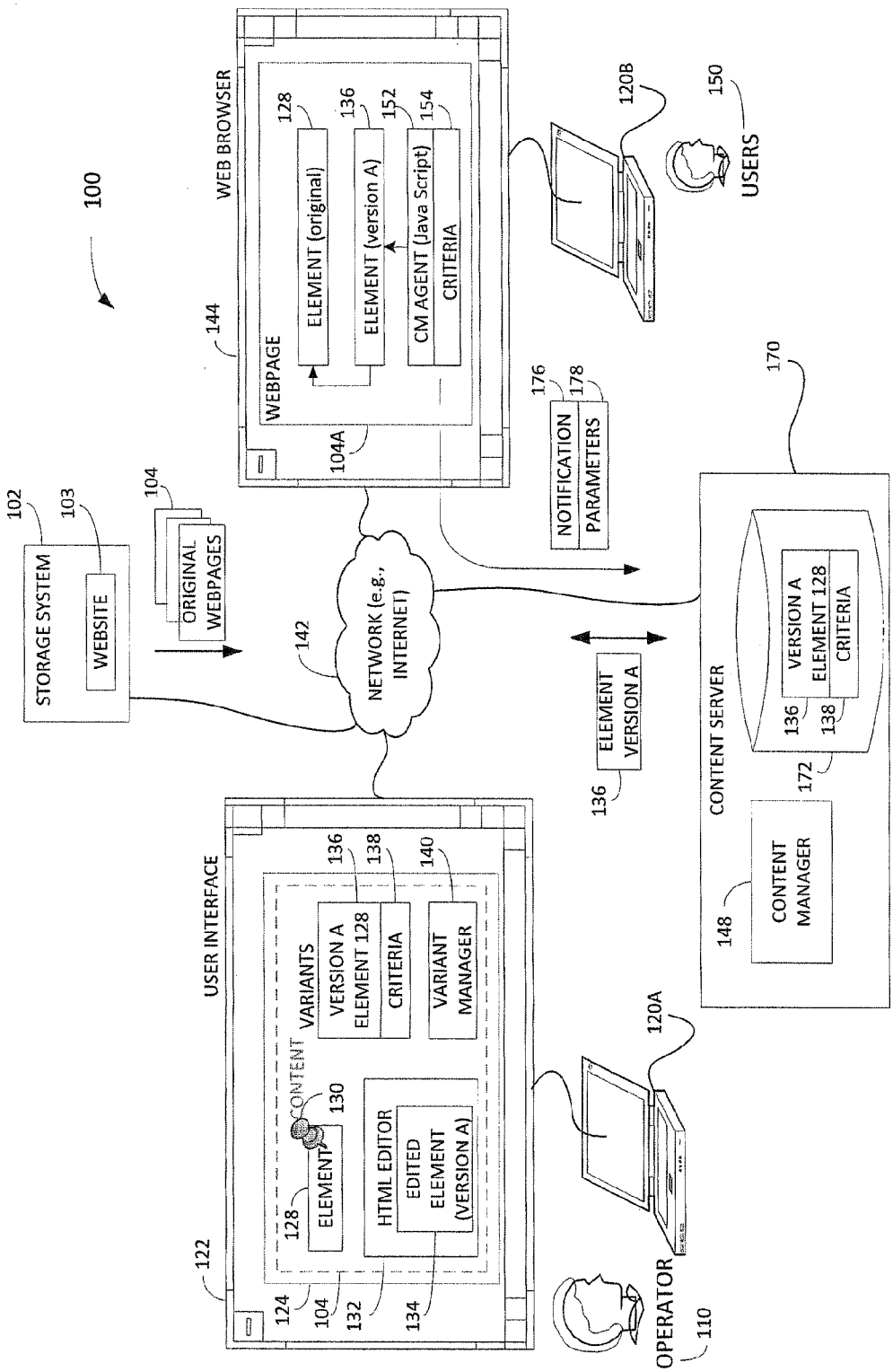

100
WEB BROWSER
WEBPAGE
ELEMENT (original)
ELEMENT (version A)
CM AGENT (Java Script)
CRITERIA
128
136
152
154
144
104A
120B
150
USERS
176
178
NOTIFICATION
PARAMETERS
170
102
103
STORAGE SYSTEM
WEBSITE
104
ORIGINAL WEBPAGES
142
NETWORK (e.g., INTERNET)
ELEMENT VERSION A
136
CONTENT SERVER
VERSION A
ELEMENT 128
CRITERIA
136
138
172
CONTENT MANAGER
148
USER INTERFACE
VARIANTS
VERSION A
ELEMENT 128
CRITERIA
VARIANT MANAGER
136
138
140
CONTENT
ELEMENT
128
130
HTML EDITOR
EDITED ELEMENT (VERSION A)
122
124
104
132
134
120A
OPERATOR
110

FIG. 1

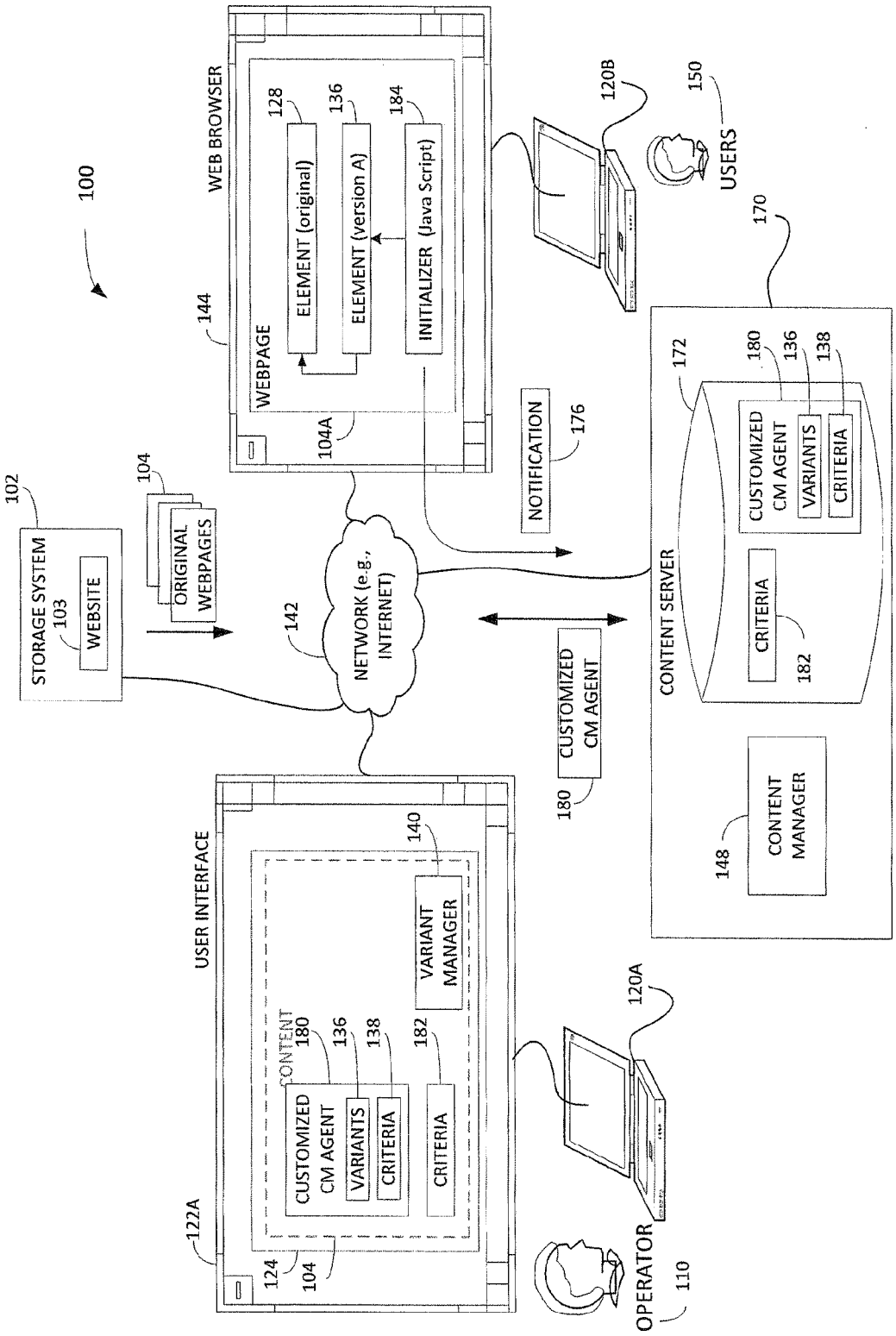

100
WEB BROWSER
144
WEBPAGE
128
ELEMENT (original)
136
ELEMENT (version A)
184
INITIALIZER (Java Script)
104A
120B
150
USERS
170
172
180
136
138
CUSTOMIZED CM AGENT
VARIANTS
CRITERIA
CONTENT SERVER
CRITERIA
182
CONTENT MANAGER
148
NOTIFICATION
176
102
STORAGE SYSTEM
103
WEBSITE
104
ORIGINAL WEBPAGES
142
NETWORK (e.g., INTERNET)
CUSTOMIZED CM AGENT
180
USER INTERFACE
140
VARIANT MANAGER
CONTENT
180
136
138
CUSTOMIZED CM AGENT
VARIANTS
CRITERIA
182
CRITERIA
122A
124
104
120A
OPERATOR
110

FIG. 2

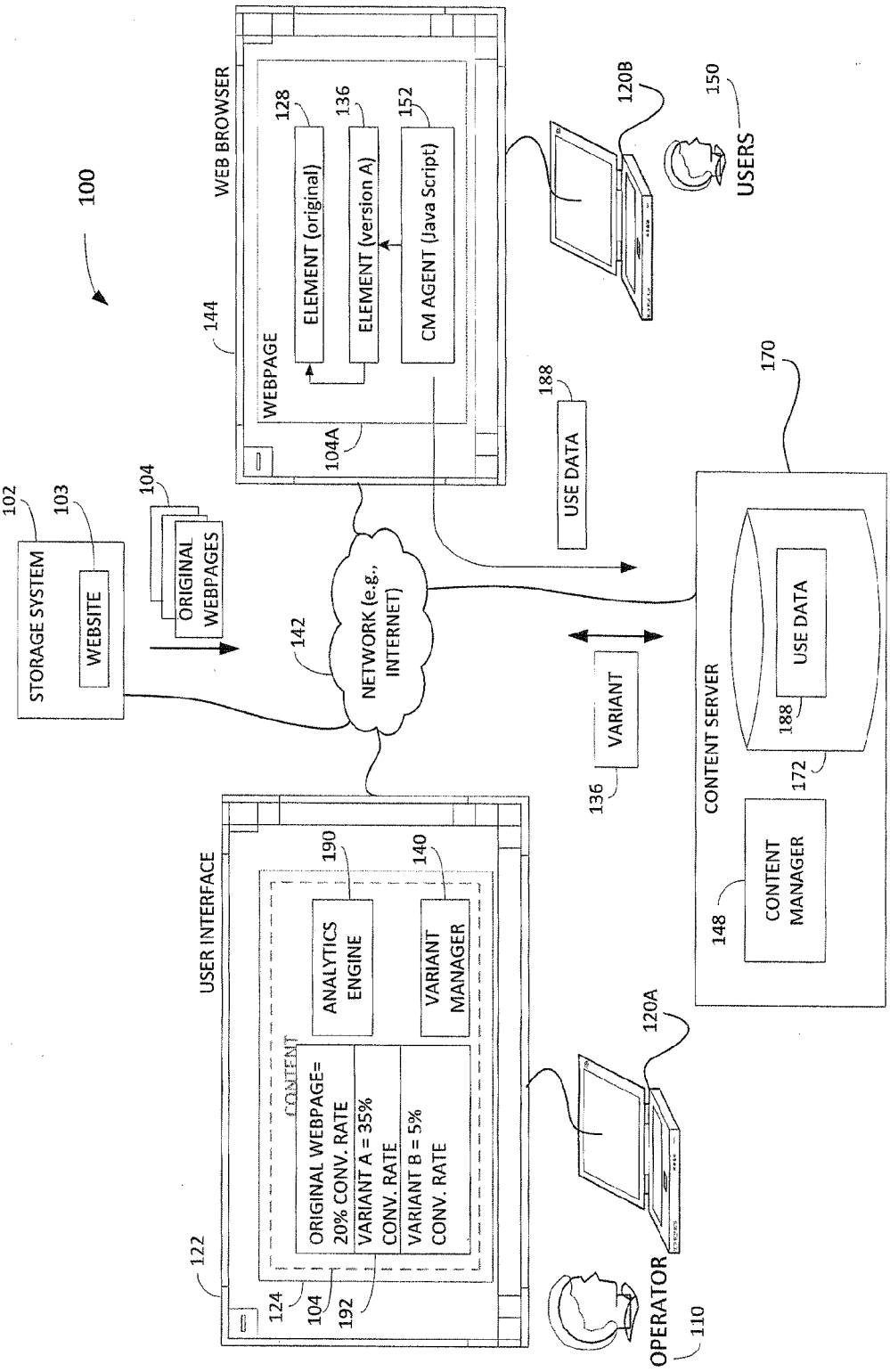

100
WEB BROWSER
WEBPAGE
ELEMENT (original)
ELEMENT (version A)
CM AGENT (Java Script)
128
136
152
144
104A
120B
150
USERS
188
USE DATA
170
STORAGE SYSTEM
WEBSITE
102
103
104
ORIGINAL WEBPAGES
142
NETWORK (e.g., INTERNET)
VARIANT
136
CONTENT SERVER
USE DATA
188
172
CONTENT MANAGER
148
USER INTERFACE
ANALYTICS ENGINE
190
VARIANT MANAGER
140
CONTENT
ORIGINAL WEBPAGE= 20% CONV. RATE
VARIANT A = 35% CONV. RATE
VARIANT B = 5% CONV. RATE
122
124
104
192
120A
OPERATOR
110

FIG. 3

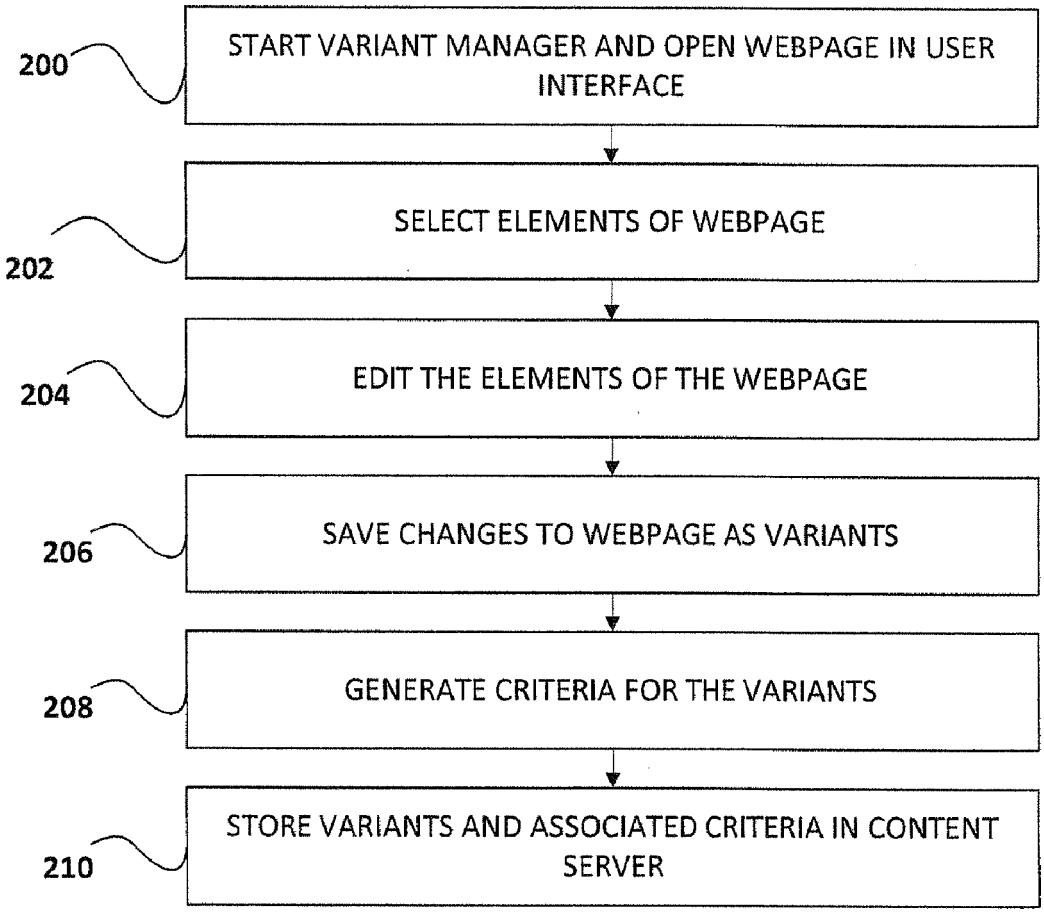

200
START VARIANT MANAGER AND OPEN WEBPAGE IN USER INTERFACE
202
SELECT ELEMENTS OF WEBPAGE
204
EDIT THE ELEMENTS OF THE WEBPAGE
206
SAVE CHANGES TO WEBPAGE AS VARIANTS
208
GENERATE CRITERIA FOR THE VARIANTS
210
STORE VARIANTS AND ASSOCIATED CRITERIA IN CONTENT SERVER

FIG. 4

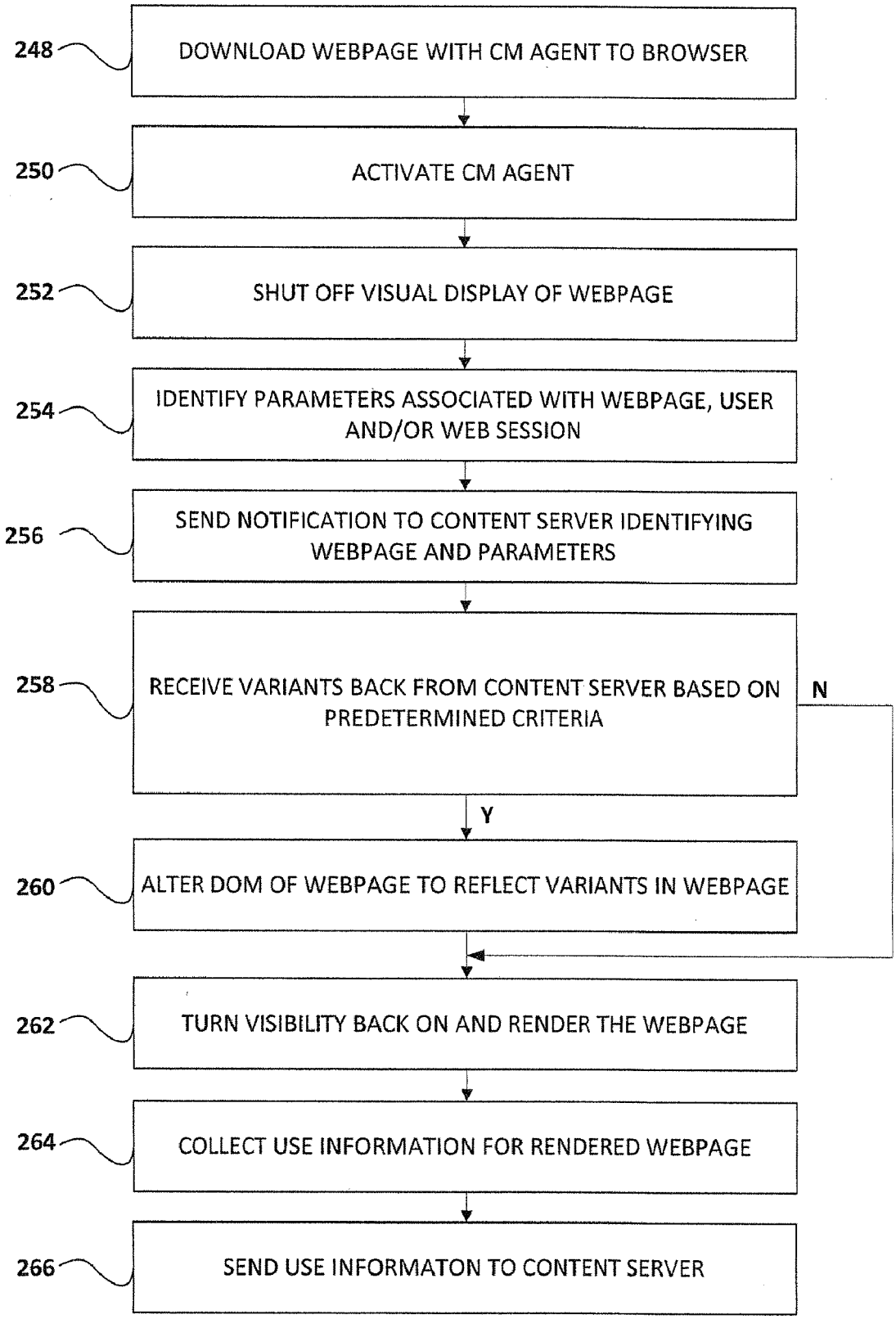

248
DOWNLOAD WEBPAGE WITH CM AGENT TO BROWSER
250
ACTIVATE CM AGENT
252
SHUT OFF VISUAL DISPLAY OF WEBPAGE
254
IDENTIFY PARAMETERS ASSOCIATED WITH WEBPAGE, USER AND/OR WEB SESSION
256
SEND NOTIFICATION TO CONTENT SERVER IDENTIFYING WEBPAGE AND PARAMETERS
258
RECEIVE VARIANTS BACK FROM CONTENT SERVER BASED ON PREDETERMINED CRITERIA
N
Y
260
ALTER DOM OF WEBPAGE TO REFLECT VARIANTS IN WEBPAGE
262
TURN VISIBILITY BACK ON AND RENDER THE WEBPAGE
264
COLLECT USE INFORMATION FOR RENDERED WEBPAGE
266
SEND USE INFORMATON TO CONTENT SERVER

FIG. 5

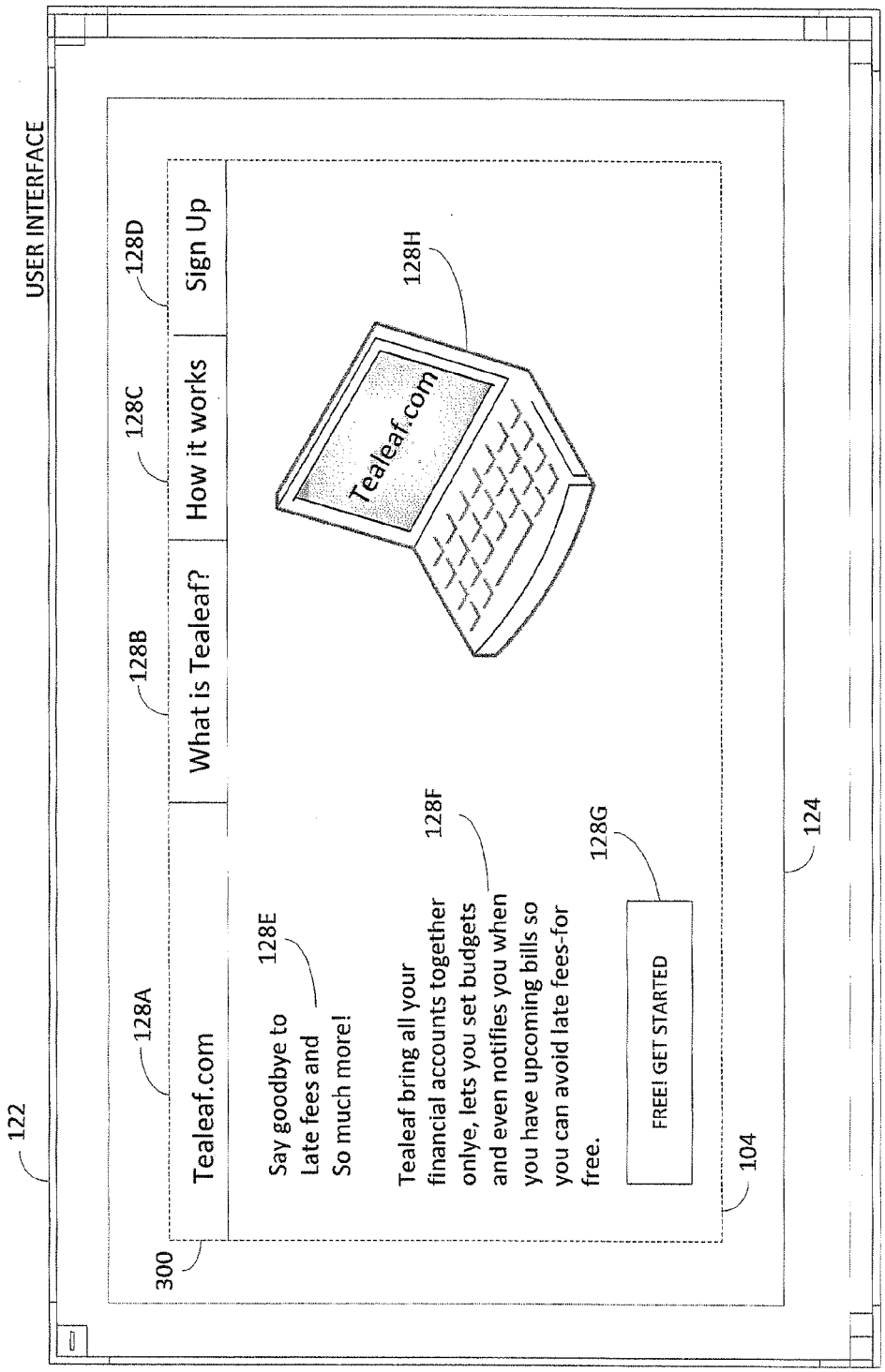

140
USER INTERFACE
122
300
128A
Tealeaf.com
128B
What is Tealeaf?
128C
How it works
128D
Sign Up
128E
Say goodbye to
Late fees and
So much more!
128F
Tealeaf bring all your
financial accounts together
onlye, lets you set budgets
and even notifies you when
you have upcoming bills so
you can avoid late fees-for
free.
128G
FREE! GET STARTED
128H
Tealeaf.com
104
124

FIG. 6

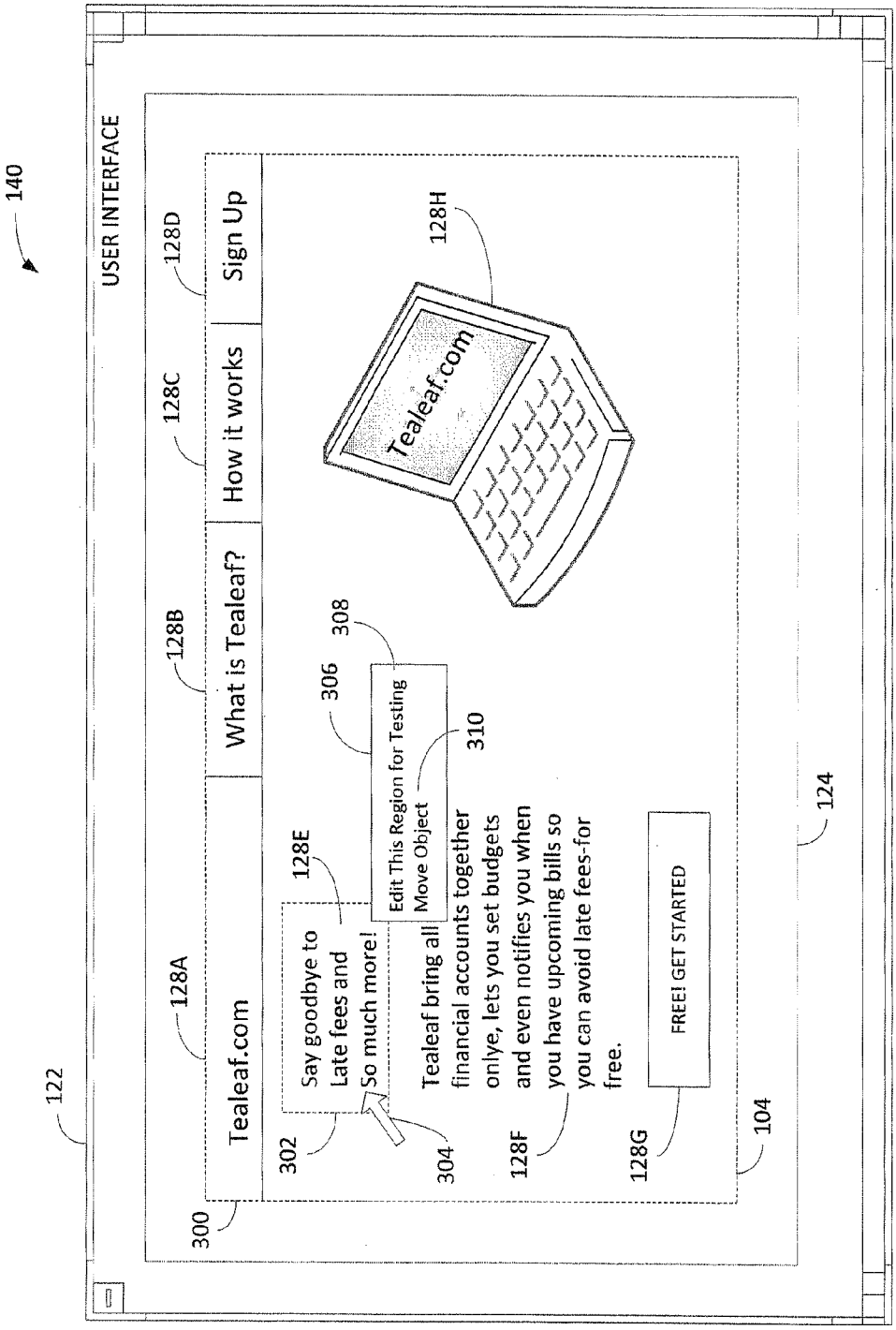

140
USER INTERFACE
122
Tealeaf.com
What is Tealeaf?
How it works
Sign Up
128A
128B
128C
128D
300
302
Say goodbye to Late fees and So much more!
128E
304
Edit This Region for Testing
Move Object
306
308
310
Tealeaf bring all financial accounts together onlye, lets you set budgets and even notifies you when you have upcoming bills so you can avoid late fees-for free.
128F
FREE! GET STARTED
128G
Tealeaf.com
128H
104
124

FIG. 7

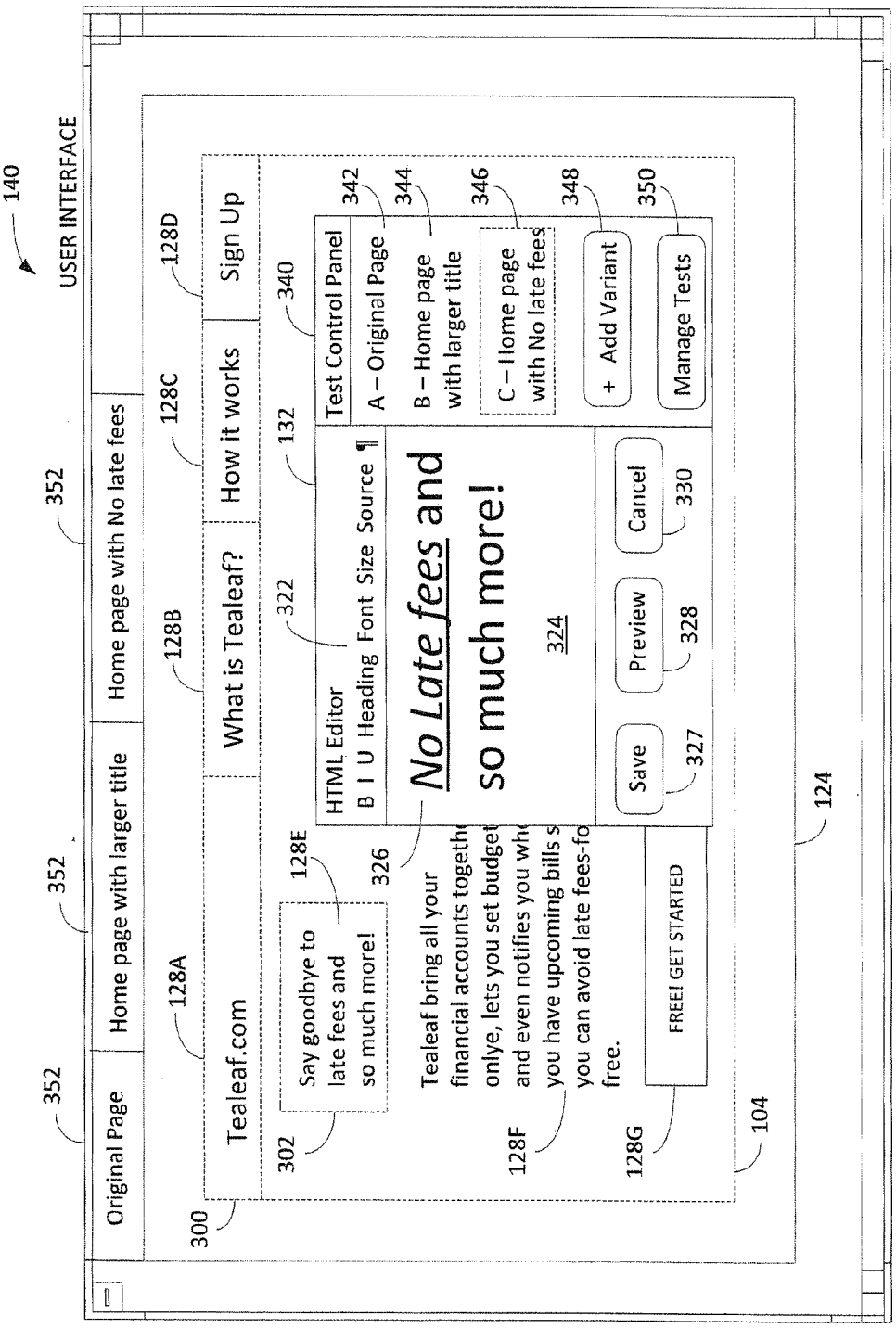

140
USER INTERFACE
352
Original Page
Home page with larger title
Home page with No late fees
128A
Tealeaf.com
128B
What is Tealeaf?
128C
How it works
128D
Sign Up
300
302
Say goodbye to late fees and so much more!
128E
128F
Tealeaf bring all your financial accounts togeth onlye, lets you set budget and even notifies you wh you have upcoming bills s you can avoid late fees-fo free.
128G
FREE! GET STARTED
104
124
326
HTML Editor
B I U Heading Font Size Source ¶
322
No Late fees and so much more!
324
132
Save
327
Preview
328
Cancel
330
340
Test Control Panel
342
A – Original Page
344
B – Home page with larger title
346
C – Home page with No late fees
348
+ Add Variant
350
Manage Tests

FIG. 8

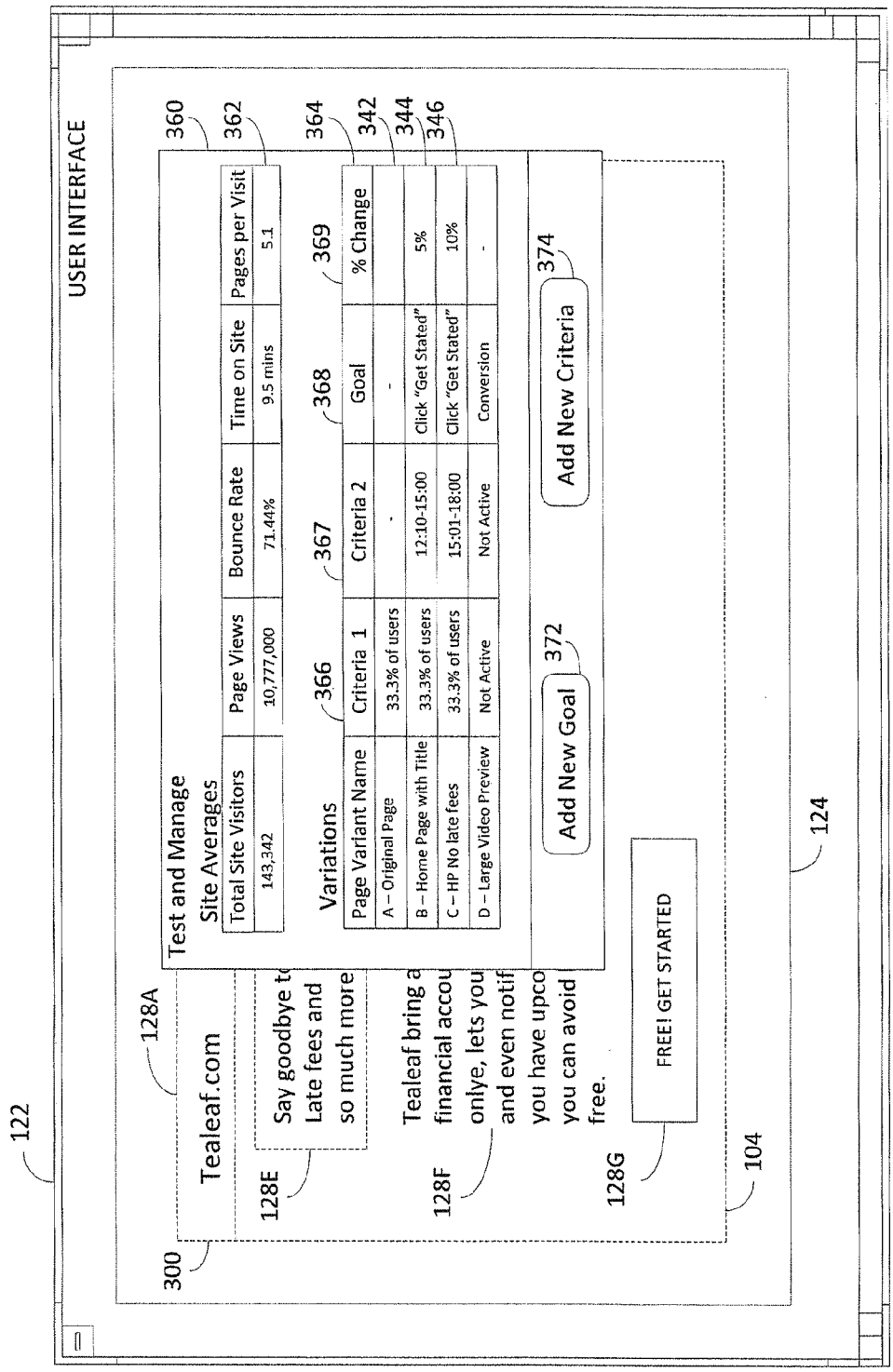

140
USER INTERFACE
122
124
104
300
128A
Tealeaf.com
128E
Say goodbye to
Late fees and
so much more
128F
Tealeaf bring a
financial accou
onlye, lets you
and even notif
you have upco
you can avoid
free.
128G
FREE! GET STARTED
Test and Manage
Site Averages
360
362
Total Site Visitors
Page Views
Bounce Rate
Time on Site
Pages per Visit
143,342
10,777,000
71.44%
9.5 mins
5.1
Variations
366
367
368
369
364
342
344
346
Page Variant Name
Criteria 1
Criteria 2
Goal
% Change
A – Original Page
33.3% of users
-
-
B – Home Page with Title
33.3% of users
12:10-15:00
Click "Get Stated"
5%
C – HP No late fees
33.3% of users
15:01-18:00
Click "Get Stated"
10%
D – Large Video Preview
Not Active
Not Active
Conversion
-
Add New Goal
372
Add New Criteria
374

FIG. 9

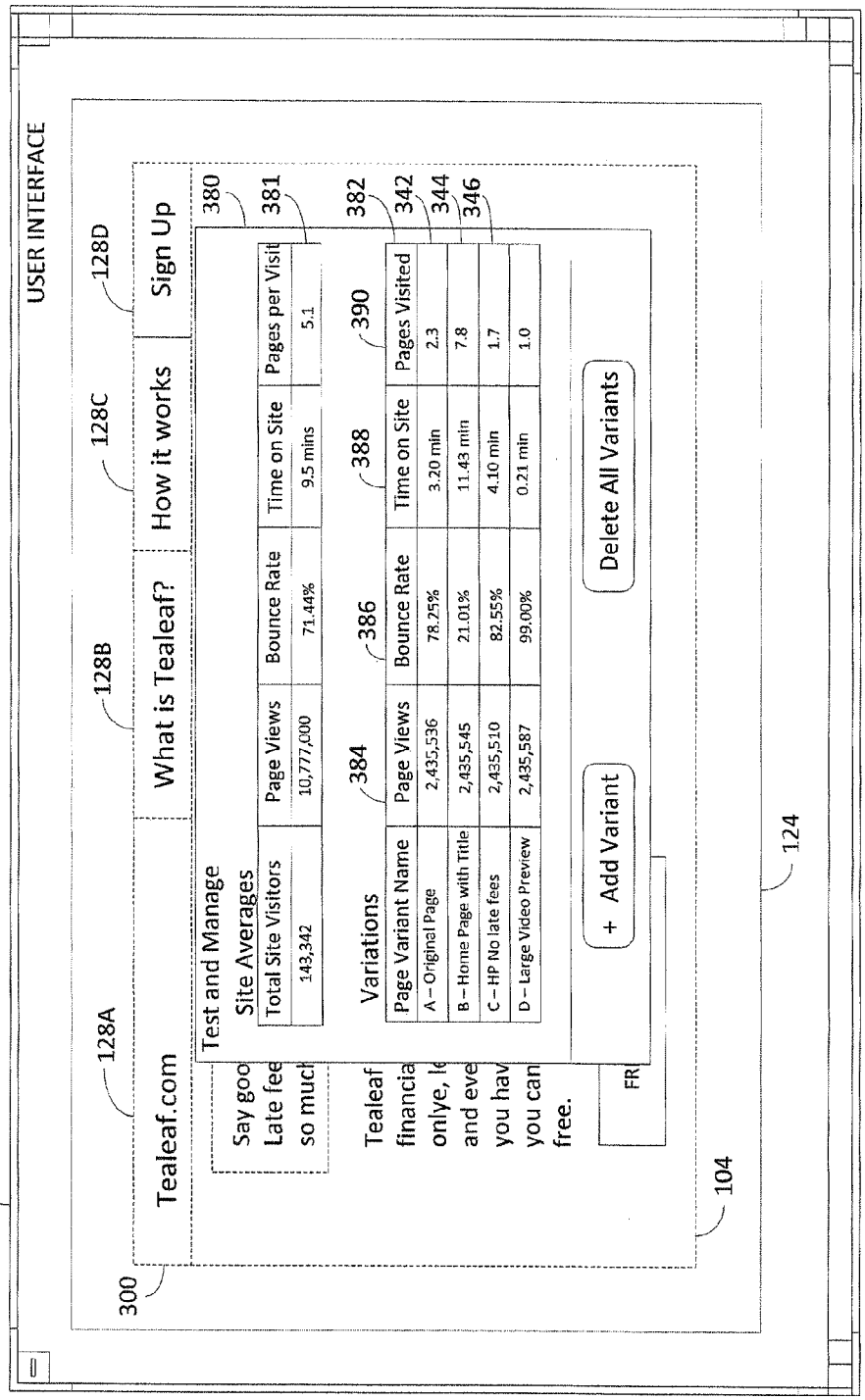

140
USER INTERFACE
122
128A
128B
128C
128D
Tealeaf.com
What is Tealeaf?
How it works
Sign Up
300
Test and Manage
Site Averages
380
381
Total Site Visitors
Page Views
Bounce Rate
Time on Site
Pages per Visit
143,342
10,777,000
71.44%
9.5 mins
5.1
Variations
382
384
386
388
390
342
344
346
Page Variant Name
Page Views
Bounce Rate
Time on Site
Pages Visited
A – Original Page
2,435,536
78.25%
3.20 min
2.3
B – Home Page with Title
2,435,545
21.01%
11.43 min
7.8
C – HP No late fees
2,435,510
82.55%
4.10 min
1.7
D – Large Video Preview
2,435,587
99.00%
0.21 min
1.0
+ Add Variant
Delete All Variants
124
104

FIG. 10

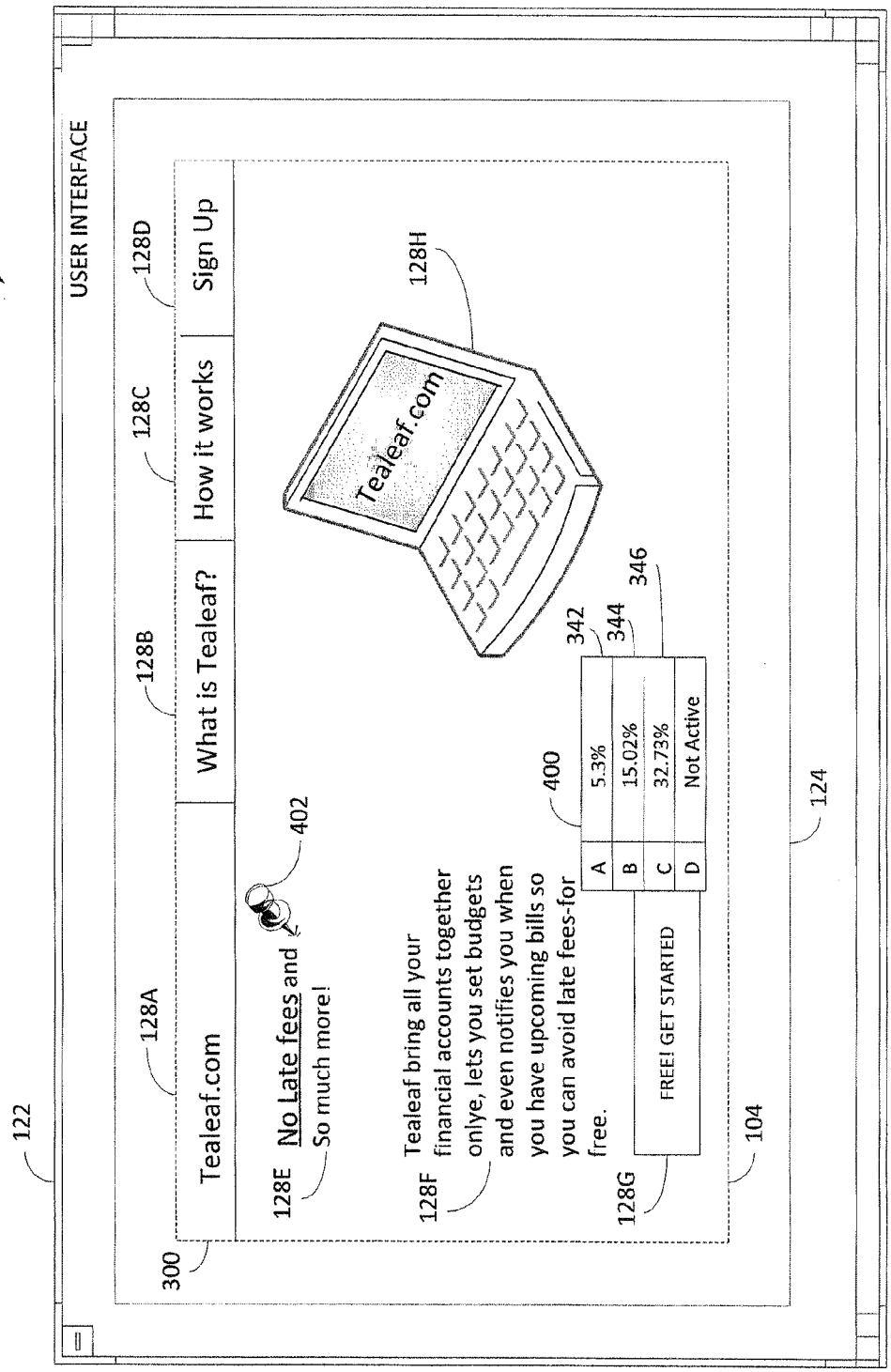

140
USER INTERFACE
122
128A
Tealeaf.com
128B
What is Tealeaf?
128C
How it works
128D
Sign Up
300
128E
No Late fees and
So much more!
402
128F
Tealeaf bring all your financial accounts together onlye, lets you set budgets and even notifies you when you have upcoming bills so you can avoid late fees-for free.
128G
FREE! GET STARTED
128H
Tealeaf.com
400
342
344
346
A
5.3%
B
15.02%
C
32.73%
D
Not Active
104
124

FIG. 11

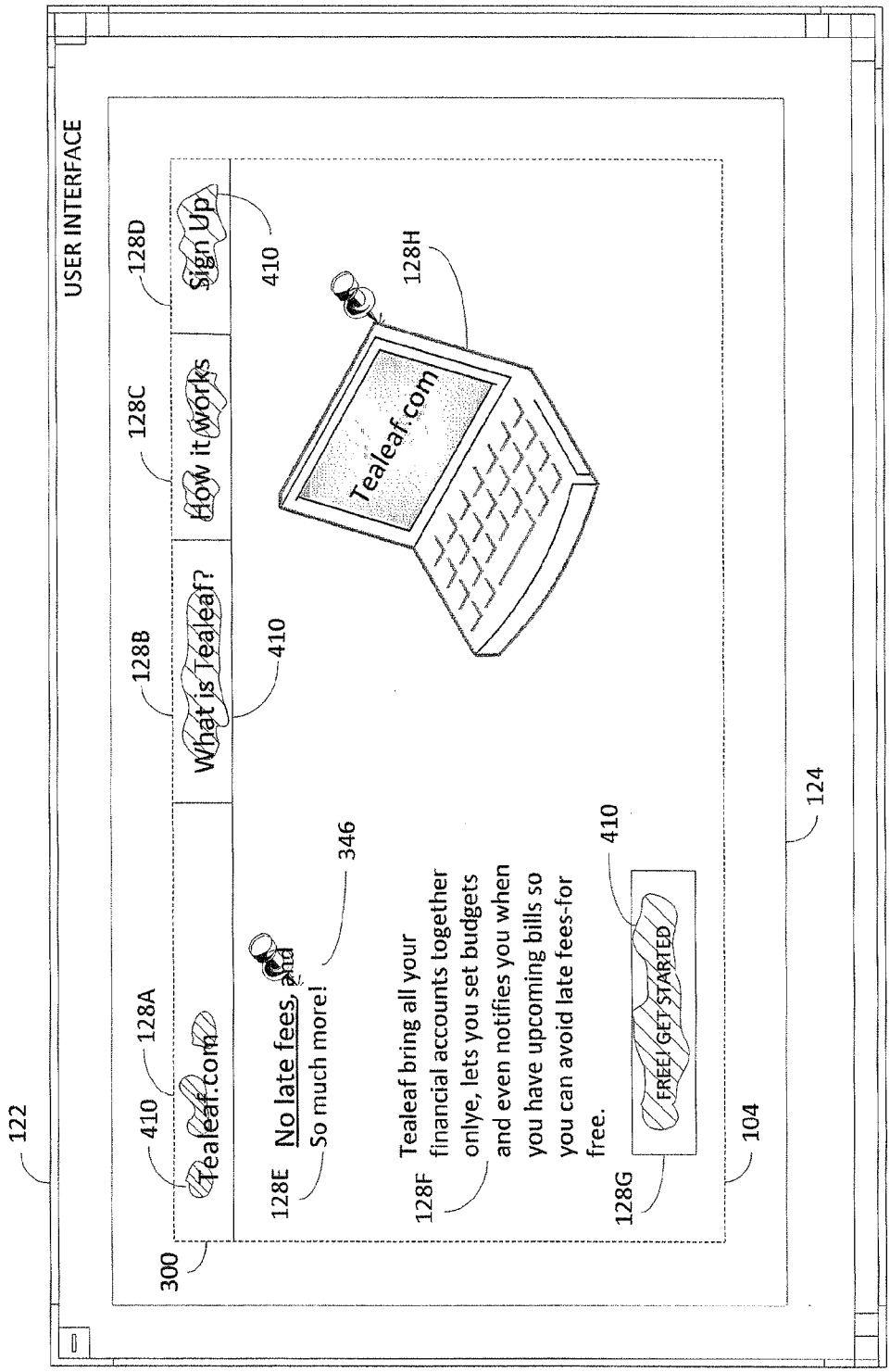

140
USER INTERFACE
122
124
104
300
410
128A
Tealeaf.com
128B
What is Tealeaf?
128C
How it Works
128D
Sign Up
128E
No late fees, and
So much more!
346
128F
Tealeaf bring all your financial accounts together onlye, lets you set budgets and even notifies you when you have upcoming bills so you can avoid late fees-for free.
128G
FREE! GET STARTED
128H
Tealeaf.com

FIG. 12

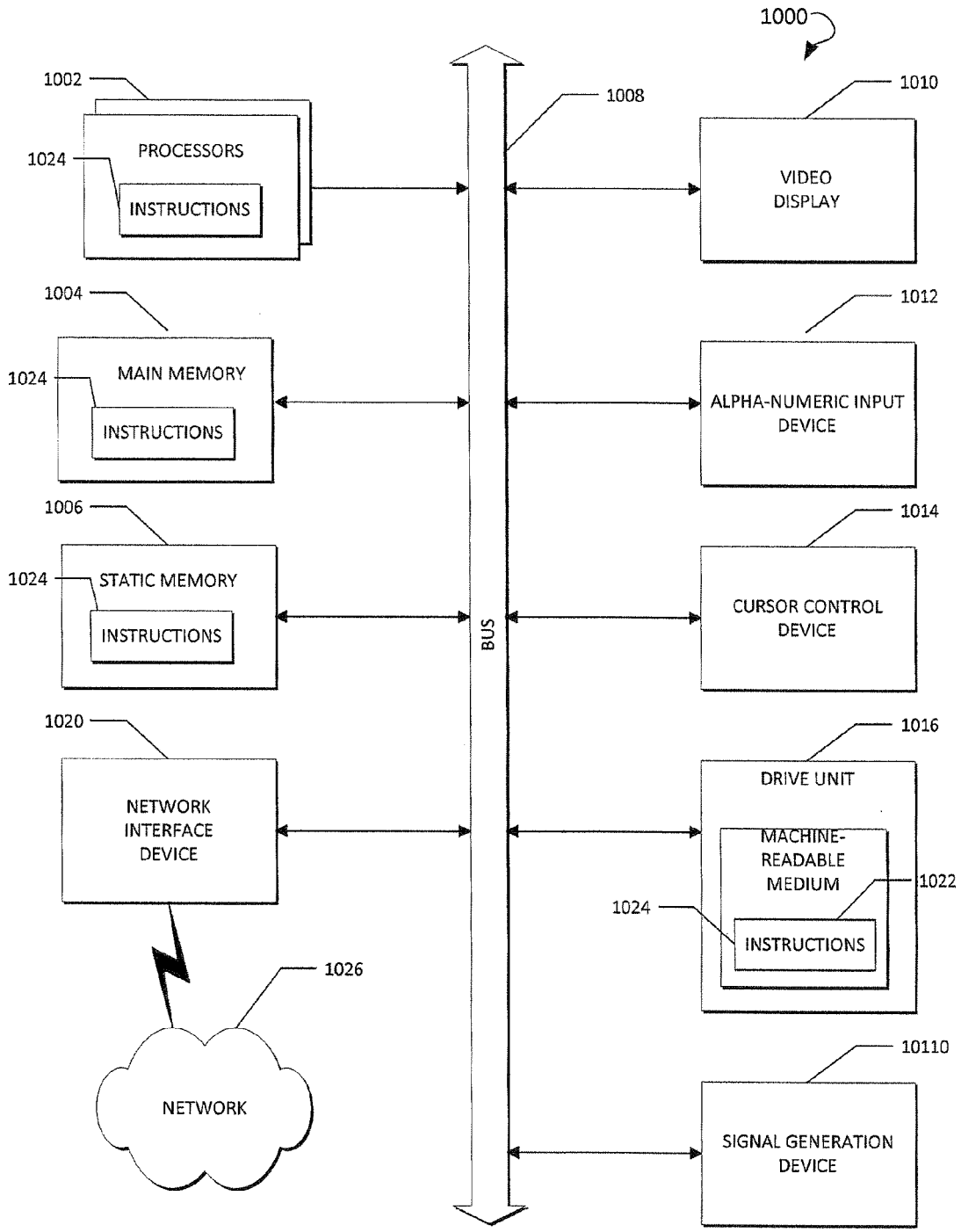

1000
1002
PROCESSORS
1024
INSTRUCTIONS
1004
MAIN MEMORY
1024
INSTRUCTIONS
1006
STATIC MEMORY
1024
INSTRUCTIONS
1020
NETWORK INTERFACE DEVICE
1026
NETWORK
1008
BUS
1010
VIDEO DISPLAY
1012
ALPHA-NUMERIC INPUT DEVICE
1014
CURSOR CONTROL DEVICE
1016
DRIVE UNIT
MACHINE-READABLE MEDIUM
1022
1024
INSTRUCTIONS
10110
SIGNAL GENERATION DEVICE

FIG. 13

ON-PAGE MANIPULATION AND REAL-TIME REPLACEMENT OF CONTENT

[0001] The present application claims priority to U.S. Provisional Patent Application, Ser. No. 61/445,528, entitled: USING RICH INTERNET APPLICATIONS FOR ON-PAGE MANIPULATION OF UNDERLYING CONTENT AND REAL-TIME CONTENT REPLACEMENT, filed Feb. 22, 2011, which is incorporated by reference in its entirety. The present application is also a continuation-in-part of co-pending U.S. patent application Ser. No. 12/750,607, entitled: VISUALIZATION OF WEBSITE ANALYTICS, filed Mar. 30, 2010, which is incorporated by reference in its entirety.

FIELD

[0002] This application relates generally to data processing and, more specifically, to on-page manipulation of underlying content and real-time content replacement.

BACKGROUND

[0003] Different testing schemes such as A/B, split, and multivariate testing, can be used on webpages to compare a baseline control webpage to a variety of webpage test samples. Based on the results of such testing, different content can be used to improve response rates for the webpages. This approach can be used to test tactics such as banner ads, emails, and landing pages. Different versions of a test webpage may be compared to a control sample webpage to identify elements that tend to produce better results. For example, different variations of the webpage may be distributed to users to determine which variables are most effective at increasing a response rate or other desired outcome. In order to be effective, the test webpages should reach enough users to detect meaningful differences between the different versions of the webpage.

[0004] Many companies, including Amazon®, Google®, and Microsoft® employ or provide multivariate testing tools that help in making marketing decisions. For example, Google® provides a website optimizer that allows users to upload different versions of a webpage to a server. Based on request attributes, search requests may be directed to specific versions of the webpage. Systems also exist that use overlays to manipulate some aspects of an underlying content. However, these systems use pure Hyper Text Markup Language (HTML) and are therefore limited on what features of the webpages can be manipulated.

BRIEF DESCRIPTION OF DRAWINGS

[0005] FIG. **1** depicts an example of a content management system.

[0006] FIG. **2** depicts an example of a content management system generating customized content management agents.

[0007] FIG. **3** depicts an example of a content management system providing analytics for content variants.

[0008] FIG. **4** depicts an example of a process for generating variants for webpages.

[0009] FIG. **5** depicts an example of a process for real-time alteration of webpages.

[0010] FIG. **6** depicts a user interface for a webpage editor.

[0011] FIG. **7** depicts a user interface for identifying elements on a webpage for editing.

[0012] FIG. **8** depicts a user interface for editing elements on a webpage.

[0013] FIG. **9** depicts a user interface for generating criteria for displaying different variants on a webpage.

[0014] FIG. **10** depicts a user interface for displaying analytics for different variants displayed on a webpage.

[0015] FIG. **11** depicts a user interface for identifying performance results for different variants.

[0016] FIG. **12** depicts a user interface displaying a usage map for variants of a webpage.

[0017] FIG. **13** depicts a computing device for implementing a content management system.

DETAILED DESCRIPTION

[0018] A content management system may create different versions of content. By viewing statistics associated with the different versions, an operator can determine which versions attract more user traffic. The different versions of the webpage may provide substantially the same message but use different presentations. For example, one version may have a large header, a second version may have a slide show, and a third version may have an embedded video. In other examples, different versions of the webpage may comprise different colors, different sizes, and/or different content for text, images, icons, fields, etc., or the like, or any variations thereof.

[0019] An operator may enter a special key to access webpages on an associated website. An overlay may extend over the webpages providing the operator with a visual test setting. Once the overlay appears, the operator may select elements within the webpages for editing. The operator can use an HTML editor to alter the selected elements. In one example, the HTML editor may comprise a What You See Is What You Get (WYSIWYG) interface.

[0020] The content management system allows the operator to change the appearance of webpages viewed by users without changing the underlying content in the original webpages. This allows webpages to be changed on the fly and different versions of the webpages to be distributed to different groups of users based on selectable criteria. The content management system also may quantify the changes made to the webpages by gathering associated statistics and determining which versions of the webpages provide the best results. A user interface can present the test results in overlays as visualized analytic data.

[0021] FIG. **1** is a block diagram depicting a content management system **100** for manipulating content. A storage system **102** may store content. For example, storage system **102** may store webpages **104** for a website **103** operated by an enterprise. The enterprise may be a company that sells products on-line to users **150**. Of course, the enterprise or operating storage system **102** and/or website **103** may be any service, business, company, individual, or other entity that provides information to users **150**. Original webpages **104** may comprise a home webpage for website **103** and any additional webpages stored on storage system **102** associated with website **103**.

[0022] A network **142** may connect storage system **102** to different computing devices **120**. Network **142** may comprise any combination of local area networks (LANs), wide area networks (WANs), Internet Protocol (IP) networks, phone networks, Public Services Telephone Networks (PSTN), wireless networks, cellular networks, Wi-Fi networks, Bluetooth networks, cable networks, data buses, or the like, or any

combination thereof used for transferring information between storage system **102** and computing devices **120**.

[0023] Computing devices **120** may comprise any device that can access content on storage system **102**. For example, computing devices **120** may comprise tablet computers, hand-held devices, smart telephones, mobile telephones, personal digital assistants (PDAs), laptop computers, personal computers, computer terminals, PSTN telephones, voice over internet protocol (VoIP) phones, or the like, or any combination thereof.

[0024] Some of computing devices **120**A may be used by operators **110** of website **103** and some of computing devices **120**B may be used by users **150** of website **103**. For example, operators **110** may be persons employed by the enterprise to manage content in webpages **104**. Users **150** may be customers or subscribers of the enterprise that operates website **103**.

[0025] Users **150** may use computing devices **120**B to access website **103**. For example, one of users **150** may open a web browser **144** and send a Hypertext Transfer Protocol (HTTP) request over network **142** to website **103**. Website **103** may send webpages **104** back to web browser **144** in response to the HTTP requests. Web browser **144** then may display the webpages **104** on a computer screen of computing device **120**B.

[0026] Operator **110** may need to modify content on webpages **104**. For example, operator **110** may want to modify content in webpages **104** to increase sales for products sold on webpages **104**. In another example, operator **110** may want to redirect user activity to different webpages **104** on web site **103** or add new content for new products or services. The content management system **100** allows operator **110** to change the content displayed on webpages **104** and send test results for the changes back to operator **110**.

[0027] A variant manager **140** may operate on computing device **120**A, storage system **102**, and/or on a content server **170**. Variant manager **140** may open a user interface **122** on computing device **120**A and download one of webpages **140** from storage system **102**. Variant manager **140** also may display an overlay **124** over webpage **104**. Overlay **124** may be transparent, semitransparent and/or opaque renderings and allows operator **110** to see and manipulate the underlying content contained within webpage **104**.

[0028] User interface **122** may communicate with a code snippet embedded into webpage **130**. This code snippet can interact with the webpage Data Object Model (DOM), add or extract elements to or from the source code of webpage **104**, initialize webpage **104**, and retrieve bootstrap scripts from content server **170**. The bootstrap script, in turn, can load a library script, which provides broader functionality. Both the bootstrap script and library script can load overlay **124** and can be enabled remotely by a content manager **148** in content server **170**. In another example, there may be no overlay **124** and operator **110** may access and modify elements in webpage **104**B simply by accessing the webpage **104** through user interface **122**.

[0029] Webpages **104** may contain one or more original elements **128**. Elements **128** may comprise any information or control that may be contained and/or displayed within webpages **104**. For example, elements **128** may comprise text, icons, images, links, fields, hypertext markup language (HTML) code, extensible markup language (XML) code, or the like, or any combination thereof.

[0030] As mentioned above, operator **110** may want to modify one or more elements **128** in an attempt to increase sales activity from webpage **104**. For example, element **128** may comprise an icon that a user selects when purchasing an item. Operator **110** may want to change the color of the icon from green to red with the hope of causing more users to select the icon and purchase more associated items.

[0031] Operator **110** may select element **128** via overlay **124** causing variant manager **140** to display an HTML editor **132** within interface **122**. HTML editor **132** may display the selected original element **128** within an edit window **134**. In one example, HTML editor **132** may comprise a WYSIWYG interface that operates similar to a word processing application and allows operator **110** to edit element **128** within window **134**. An edited version A of element **128** is stored and identified as variant **136** in user interface **122**.

[0032] Operator **110** may make other changes to element **128** and save those other changes as other variants **136**. For example, operator **110** may change the color of the icon associated with element **128** from green to red and save the red icon as version A variant **136**. Operator **110** also may change the color of the icon associated with element **128** from green to blue and save the blue icon as a second version B variant **136**.

[0033] In another example, operator **110** may modify other elements in webpage **104**. For example, operator **110** also may change a font on text located in another location within webpage **104**. Operator **110** may select the text for the second element **128** and the text may be displayed in HTML editor **132**. Operator **110** may make the desired changes to the font of the text and save the changes as another variant **136**. Operator **110** can make changes to any elements within webpage **104** and the changes may be saved by variant manager **140** as different variants **136**. Variant manager **140** then may upload the variants **136** of webpage **104** to content server **170**.

[0034] Operator **110** also may specify different criteria **138** for the variants **136**. For example, operator **110** may create three different variants **136** for webpage **104** and may want to distribute the three different variants **136** evenly among users **150**. Accordingly, operator **110** may designate a criteria **138** for a version A of element **126** to be displayed to 33% of users **150** accessing website **103**. Operator **110** also may create criteria **138** for distributing each of the other two variants **136** to 33% of users **150**. Criteria **138** are uploaded with the associated variants **136** to content server **170**.

[0035] Content management system **100** then may perform real-time content replacement for testing the variants **136**. Content management (CM) agents **152** may be embedded into original webpages **104** and may operate in conjunction with content manager **148**. CM agents **152** and content manager **148** determine what variants **136** are displayed to users **150** within webpages **104**. In one example, CM agent **152** may comprise Java script, Adobe Flash, or Microsoft Silverlight.

[0036] User **150** may download original webpage **104** from website **103** into web browser **144**. Webpage **104** may be embedded with CM agent **152**. CM agent **152** may send a notification **176** to content manager **148** identifying webpage **104**. Notification **176** also may provide other parameters **178** needed by content manager **148** to determine what variants **136**, if any, to send back to CM agent **152**. For example, parameters **178** may provide additional information regarding user **150** and may be compared with criteria **138**. In another example, CM agent **152** may compare parameters **178** with local criteria **154** to determine when to send notifi-

cation **176** to content manager **148** and/or determine which variants **136** to display in webpage **104**.

[0037] Based on the criteria **138**, content manager **148** may send variants **136** back to CM agent **152**. CM agent **152** then may dynamically replace original element **128** in webpage **104** with the variants **136** received from content manger **148**. CM agent **152** may turn off the visibility of webpage **104** in web browser **144** until replacement HTML code in variant **136** is received from content server **170**. CM agent **152** replaces HTML code for original element **128** with the HTML code for variant **136** and then turns the visibility of now modified webpage **104**A back on in web browser **144**.

[0038] For example, variant **136** may comprise the red icon created by operator **110** in user interface **122** and uploaded to content server **170**. CM agent **152** replaces the green icon in original element **128** with the red icon in variant **136** and renders modified webpage **104**A to user **150** in web browser **144**. Thus, user **150** will see the red icon associated with variant **136** without operator **110** having to change original webpage **104** in website **103**.

[0039] Different variants **136** may be used for the same webpage **104** based on criteria **154** and criteria **138**. For example, operator **110** may have created a first variant **136** and a second variant **136** for element **128**. As explained above, element **128** may be a green icon and the first variant **136** may comprise a red icon. The second variant **136** may comprise a green icon. Operator **110** may have created first criteria **138** for the first variant **136** specifying a first time period and created second criteria **138** for the second variant **136** specifying a second time period.

[0040] A user **150** may download webpage **104** from website **103**. CM agent **152** may send notification **176** identifying webpage **104** to content manager **148**. Content manager **148** may compare a current time of day with the times identified in the first and second criteria **138**. Content manager **148** may send the first variant **136** with the red icon to CM agent **152** when the current time is within the first time period identified in first criteria **138**. Content manager **148** may send the second variant **136** with the blue icon to CM agent **152** when the current time is within the second time period identified in second criteria **138**.

[0041] CM agent **152** replaces original element **128** in original webpage **104** with variant **136** received from content manager **148** and displays modified webpage **104**A. Thus, user **150** may see a red icon during the first time period and see a green icon during the second time period. Based on criteria **138**, content manager **148** may only send the first and second variants **136** during the first and second time periods, respectively. During any other time period, content manager **148** may not send any variant **136** and original webpage **104** may be rendered in web browser **144** with original element **128**.

[0042] Criteria **154** and/or **138** may specify any variety of conditions for using any combination of different variants **136**. For example, criteria **154** and/or **138** may be based on, demographic, economic, psychographic, user interests, or the like, or any combination thereof. Other criteria **154** and/or **138** may be based on fixed times, fixed time periods, keywords in keyword searches entered by user **150**, user grouping criteria, or randomly for each user **150**.

[0043] For example, CM agent **152** may identify a source where user **150** accesses website **103**, such as from a particular web browser, search engine, social network, etc. CM agent **152** may identify the source from HTTP requests sent from computing device **120**B, an IP address, profile information for user **150**, etc. The source may be sent to content manager **148** as parameters **178** with notification **176**. Content manager **148** then may send different variants **136** back to CM agent **152** based on a comparison of the source in parameters **178** with criteria **138**.

[0044] In another example, a third party service may aggregate demographic data for users **150** and associate the demographic data with a unique identifier, such as an IP address or email address. The third party service also may receive demographic data directly from users and/or may scrape demographic data from user profiles on social networks.

[0045] Content manager **148** may receive an IP address for user **150** as part of parameters **178** with notification **176**. Content manager **148** may access the third party service and obtain demographic information associated with the user IP address, such as gender or household income. Criteria **138** may designate a first variant **136** for a male user **150** and may designate a second variant **136** for a female user **150**. Criteria **138** also may designate different variants **136** based on any identified household income. Content manager **148** then sends variants **136** with matching criteria to CM agent **152**.

[0046] FIG. **2** depicts a content management system that uses customized content management agents (CM agents). In one example, operator **110** may use user interface **122**A to generate criteria **138** associated with variants **136**. Variant manager **140** may generate a customized CM agent **180** for implementing variants **136** and criteria **138**. For example, customized CM agent **180** may comprise Java code that identifies different user parameters, compares the parameters with criteria **138** and then replaces different elements **128** in original webpage **104** with variants **136** based on the comparisons.

[0047] Variant manager **140** may generate and upload customized CM agents **180** to content server **170**. In another example, variant manager **140** may embed customized CM agents **180** in original webpages **104** and upload the webpages **104** to website **103**. Variant manager **140** may generate additional criteria **182** that determine when customized CM agents **180** are uploaded to users **150**. Criteria **182** may comprise any of the conditions described above for criteria **138** or may comprise other conditions, such as the amount of network traffic.

[0048] An initializer **184** comprising Java script may be embedded in original webpages **104**. After webpage **104** is downloaded to browser **144**, initializer **184** may send notification **176** to content manager **148**. Notification **176** may identify webpage **104** and may identify any other parameters. In response to notification **176** and based on criteria **182**, content manager **148** may upload customized CM agent **180** to web browser **144**. After uploading into web browser **144**, customized CM agent **180** may take over rendering and variant replacement operations as previously described for CM agent **152** in FIG. **1**. In another example, a tag management service may be used for sending notification **176** to content manager **148** instead of embedding script in webpages **104**.

[0049] Dynamically uploading customized CM agent **180** into web browser **144** allows more robust and customized rendering of variants **136** in webpages **104**. For example, customized CM agent **180** may be programmed to replace a wider variety of elements **128** with a wider variety of variants **136** based on a wider variety of programmable criteria **138**. Customized CM agents **180** may operate more autonomously on different computing devices **120**B, thus requiring less

communication with content server **170**. This may result in better scaling and faster rendering of modified webpages **104**A.

[0050] FIG. **3** depicts an example of a content management system **100** that provides analytics for webpages that use variants. User interface **122** may operate an analytics engine **190** directly on top of webpage **104** that views, changes, and reports on specific variants used in webpages **104**. For example, analytics engine **190** may use overlay **124** as an interactive graphical tool for generating and displaying reports **192**. The on-page analytics engine **190** may use Rich Internet Application (RIA) based, for example, on HTML 5, Adobe Flash or Microsoft Silverlight. In one example, analytics engine **190** may be implemented and operate as described in co-pending U.S. patent application Ser. No. 12/750607, entitled: VISUALIZATION OF WEBSITE ANALYTICS, filed Mar. 30, 2010 which is incorporated by reference in its entirety.

[0051] CM agent **152** operating in web browser **144** may capture use data **188** associated with any versions of webpage **104**. For example, a first original webpage **104** may be downloaded to a first web browser **144** during a first web session with a first user **150**. CM agent **152** in the first web browser **144** may not replace original element **128** with variant **136** during the first web session. CM agent **152** may track use of original webpage **104** by the first user **150**. For example, CM agent **152** may capture mouse clicks, opened original webpages **104**, conversion rates, bounce rates, amount of time spent on website **103**, etc. CM agent **152** may send the use data **188** to variant sever **170**.

[0052] A second original webpage **104** may be downloaded to a second web browser **144** during a second web session with a second user **150**. CM agent **152** in the second web browser **144** may replace original element **128** with variant **136** during the second web session. CM agent **152** then may track use of variant webpage **104**A by the second user **150** and send the use data **188** to variant sever **170**.

[0053] Analytics engine **190** then may conduct analytics on the use data **188** to determine relative popularities of the different variants for webpage **104**. For example, analytics engine **190** may display in report **192** a conversion rate for original webpage **104** and a conversion rate for modified webpage **104**A that used variant **136**. Statistics can be overlaid over the versions of webpage **104** that were displayed to the user. For example, the statistics for original webpage **104** may be displayed in overlay **124** over original webpage **104** and statistics for modified webpage **104**A may be displayed in overlay **124** over modified webpage **104**A.

[0054] Analytics engine **190** may be utilized as a proxy for the communication between overlay **124** and the underlying webpage **104**. Analytics engine **190** may manage the rendering and display of the underlying webpages **104** and **104**A and provide a plurality of automatic actions, which may trigger additional real-time content replacement based on other predefined criteria. Example applications of such real-time content replacement may include A/B testing, uptime monitoring, email management, and so forth. In one example, dynamic A/B testing may be performed by substituting elements on any webpage **104** of the website **103** according to preconfigured settings, which can be defined by operator **110** via the analytics engine **190**.

[0055] The use data **188** captured by CM agent **152** may be dynamically changed based on different programmable criteria. For example, operator **110** may generate testing criteria when creating variant **136**. The testing criteria may be uploaded to content server **170** along with variant **136**. CM agent **152** may send a notification back to content manager **148** when webpage **104** is initially downloaded into web browser **144**. Content manager **148** may send the testing criteria along with variant **136** back to CM agent **152** in response to the notification. CM agent **152** then may capture different data during the web session based on the testing criteria.

[0056] FIG. **4** depicts an example of a process for generating variants for webpages. In operation **200**, an operator starts a variant manager and downloads a webpage from a website in a user interface. In one example, the variant manager may access HTML code in the webpage through an embedded snippet. In operation **202**, the operator may select an element within the webpage via the user interface. For example, the user interface may display an overlay over the webpage and the operator may move a cursor over the overlay and over an element in the webpage and click on the element.

[0057] In operation **204**, the operator may edit the selected element. For example, a HTML editor may display the selected element and the operator may change a size, shape, content, location, etc. of the element, or completely replace, or remove the element. In operation **206**, the changes to the webpage may be saved as variants. For example, different elements, and/or different combinations of elements, may be changed in multiple different ways. The different combinations of the changes may be saved in different files as different variants. Each different file may be assigned a different name identifying the associated changes.

[0058] In operation **208**, the operator may generate different criteria for the variants. For example, the operator may create criteria that only send the variant to users in Europe. In operation **210**, the variants and associated criteria are stored in the content server. The content server then supplies the variants to web browsers based on the associated criteria. For example, the content server may send the variant in response to notifications received from a content management agent embedded in the webpage indicating the webpage has been downloaded to a web browser and that the user has an associated European IP address.

[0059] FIG. **5** depicts an example of a process for real-time replacement of webpage elements with variants. In operation **248**, an original webpage is downloaded from a website to a web browser. For example, a user may perform a search via a search engine and select an identified link to the website. The webpage downloaded to the web browser may include JavaScript code that operates as a content management (CM) agent.

[0060] In operation **250**, the CM agent is activated. For example, the CM agent may start operating on the web browser before the webpage is rendered and displayed on the web browser. In operation **252**, the CM agent may temporarily shut off the visual display of the webpage. For example, the Java script in the CM agent may modify a visibility setting in the webpage HTML, such as in a cascading style sheet (CSS) property.

[0061] In operation **254**, the CM agent may identify parameters that may determine what variants, if any, are used with the downloaded webpage. The parameters may be associated with the webpage, the user, and/or the web session. For example, the CM agent may locate a webpage identifier in the downloaded webpage and also may identify other parameters

such as an email address or demographic information for the user, or an originating portal where the user initiated the web session.

[0062] In operation **256**, the CM agent may send a notification to the content server including the webpage identifier and any other parameters as described above. In operation **258**, the content server may send variants back to the CM agent based on associated criteria. For example, the content server may send a particular variant, or particular set of variants, to users with a particular demographic identified in the parameters.

[0063] In operation **260**, the CM agent alters the document object model (DOM) for the webpage to display the variants received back from the content server. If no variants are received back from the content server, the CM agent may not alter the original DOM for the webpage. In operation **262**, the CM agent turns the visibility back on for the webpage in the web browser and the webpage is rendered and displayed to the user. The rendered webpage may now contain variants that are downloaded in real-time from the content server based on operator configured criteria.

[0064] In operation **264**, the CM agent may collect use information for the rendered webpage. For example, the CM agent may detect and collect user cursor movements and mouse clicks within individual webpages and user movements between webpages from the website. In operation **266**, the collected use data is sent back to the content server. The content server and an analytics engine may perform different analytics for the different webpage variants. For example, the analytics engine may derive conversion rates, time on website, locations of mouse clicks, heat maps, bounce rates, etc., for the original webpage and for each variant used with the webpage. The analytics engine may display the results to the operator as overlays over the associated original webpage and the modified webpages displaying the variants.

[0065] FIG. **6** depicts an example of a user interface **122** for manipulating content in a webpage **104**. User interface **122** may be displayed on a computer screen of a computing device **120**A as previously shown in FIG. **1**. In one example, user interface **122** may displayed through a web browser after an operator logs into an associated website using an authorized user name and password. In another example, user interface **122** may be displayed via an application operating on a mobile device. User interface **122** also may display a semi-transparent overlay **124** over webpage **104**.

[0066] Webpage **104** may contain multiple different elements **128**A-**128**H. For example, a tool bar **300** in webpage **104** may contain an element **128**A comprising text that identifies an Internet domain name for the website. An element **128**E may comprise text for a link to another webpage containing additional information regarding the website. An element **128**C may comprise text for a link to another webpage containing additional information regarding the website. An element **128**D may comprise text for a link to another webpage for signing up to the website.

[0067] A main body of webpage **104** may include an element **128**E comprising a first block of text and an element **128**F comprising a second block of text. An element **128**G may comprise an icon and text that when selected may link to yet another webpage for the website. An element **128**H may comprise an image that may represent additional information related to the website. Elements **128** are just examples and any content contained in a webpage may comprise elements **128**.

[0068] A variant manager **140** operating on a computing device displays user interface **122** on a screen of the computing device. An overlay **124** is displayed over webpage **104** and allows the operator to select and change elements **128** within webpage **104**. Variant manager **140** then saves the changes as variants for later displaying to users of the website.

[0069] FIG. **7** depicts an example of a user interface configured to select elements on a webpage. The operator may select one of elements **128** within webpage **104**. For example, the operator may use a computer mouse to move a cursor **304** over element **128**E. Variant manager **140** may automatically highlight element **128**E. For example, variant manager **140** may identify HTML code within webpage **104** that displays the text in selected element **128**E and display a box **302** around the identified text.

[0070] The operator may select element **128**E by moving cursor **304** over the text of element **128**E and then pressing a left mouse button or by entering a combination of keystrokes. Selecting element **128**E may cause variant manager **140** to display a panel **306**. Selecting a first item **308** in panel **306** may cause variant manager **140** to open a HTML editor for modifying element **128**E. Selecting a second item **310** in panel **306** may cause variant manager **140** to allow the operator to move element **128**E to another location within webpage **104**. For example, selecting item **310** may cause variant manager **140** to drag and drop element **128**E to a new location of cursor **304**.

[0071] FIG. **8** depicts an example of a HTML editor **132** for editing elements within a webpage. Variant manager **140** may display HTML editor **132** in response to selection of item **308** in FIG. **7**. HTML editor **132** may initially display original selected element **128**E within an edit box **324** and may display a tool bar **322** for changing the format and content of element **128**E. Thus, the operator may edit element **128**E in a same manner as using a conventional word processing application. In this example, the operator changes the original text of element **128**E from "Say goodbye to late fees and so much more!" to "No Late fees and so much more!" in a modified element **326**.

[0072] The operator may save the modified element **326** as a variant **346** by selecting a save icon **327**. The operator may preview webpage **104** with the modified element **326** by selecting a preview icon **328**. For example, selecting preview icon **328** may display webpage **104** with original text element **128**E replaced with modified text element **326**. Selecting a cancel icon **330** may delete modified text element **326** and close HTML editor **132**.

[0073] Variant manager **140** also may display a test control panel **340** in response to selection of element **128**E and/or in response to selection of save icon **327**. Test control panel **340** may identify different variants created for webpage **104**. For example, a first variant **342** in control panel **340** may comprise the original webpage **104** received from the website. In one example, original webpage **104** cannot be edited or removed with editor **132**. A second variant **344** may comprise a first change made to webpage **104**. For example, variant **344** may have created a larger title for webpage **104**. A third variant **346** may comprise a second change made to webpage **104**. For example, variant **346** may comprise modified text element **326** created in HTML editor **132**.

[0074] Selection of an icon **348** may create a file for another variant of webpage **104**. For example, the operator may select another element **128**, change the selected element **128** in HTML editor **132**, and save the changed element as the new

variant created in text control panel **340**. A manage tests icon **350** may be selected to control how and when variants **342**, **344**, and **346** are presented to users and how test results are analyzed and displayed back to the operator. For example, selecting icon **350** may open a window for creating criteria that determine when variants **342-346** are displayed to users and determine what test data to capture while the variants are being displayed to the users.

[0075] FIG. **9** depicts an example of a text management panel **360** for controlling variant tests. A section **362** may display different averages for the website. For example, section **362** may identify a total number of visitors to the website over some period of time, an average number of webpages viewed over the time period, an average bounce rate, an average time each user spent on the website, and an average number of webpages visited by the users. Of course, any other test data can also be displayed.

[0076] Section **364** may contain fields for setting criteria for the different variants **342-346**. For example, the operator may enter values into fields of a criteria column **366** that determine how variants **342-346** are displayed to users. For example, the operator may enter values of 33% into the fields of column **366** associated with variants **342**, **344**, and **346**. Variant manager **140** then may assign criteria to each of the variants that cause the content server to distribute each of variants **342-346** to 33% of the users accessing the website.

[0077] Other criteria can also be created. For example, the operator may enter time values into fields of a second criteria column **367**. Variant manager **140** then may assign criteria to each of variants **342-346** that cause the content server to use variants **342-346** during associated time periods specified in criteria column **367**. For example, variant **344** may be used between 12:10-3:00pm, variant **346** may be used between 3:01-6:00pm, and original page variant **342** may be used during all other time periods.

[0078] The operator may enter values into fields of a goal column **368** to specify specific user actions/goals for monitoring while variants **342-346** are being displayed. For example, one goal may be to increase the number of mouse clicks to the icon element **128**G in webpage **104**. The operator may select the icon element **128**G and then select the field for one of variants **342-346** in column **368** to link to icon element **128**G.

[0079] Variant manager **140** may generate Java code that the CM agent uses to track clicks on icon element **128**G. The CM agent already may track the mouse clicks for icon element **128**G. Variant manger **140** then may only need to send a message to the content server or to an analytic engine to aggregate the mouse clicks on icon element **128**G for each of selected variants **342-346**.

[0080] The number or percentage of mouse clicks on icon element **128**G may be calculated and displayed in column **369** for each of variants **342-346**. For example, a number of mouse clicks on icon element **128**G may be tracked while original page variant **342** is displayed to users. The number of mouse clicks on icon element **128**G also may be tracked while variants **344** and **346** are displayed to users. The analytics engine then may compare the number of mouse clicks for each of the variants **342-346**. For example, a percentage increase or decrease in the number of mouse clicks for variants **344** and **346** may be compared with the number of mouse clicks on original webpage **104** in column **369**. A icon **372** may be selected to add new goals for the variants and a icon **374** may be selected to add new criteria for the variants.

[0081] FIG. **10** depicts an example of a panel **380** showing tests results for different variants. The data shown in panel **380** may be captured by CM agents and sent to content server **170** in FIG. **1** for further analytic analysis. A section **381** may display averages for user visits to the website as described above in FIG. **9**. For example, section **381** may identify a total number of visitors to the website over some period of time, the number of webpages viewed by the users, a bounce rate, an average time each user spent on the website, and an average number of webpages opened by the users per website visit.

[0082] Section **382** may identify results for each the different variants **342-346**. For example, column **384** may show the number of times the original webpage **104** associated with variant **342** was viewed. Column **384** also may show the total number times the modified webpages with variants **344** and **346** were viewed. A column **386** shows a bounce rate indicating a percentage of users who enter the associated webpage and leave the website rather than continue viewing other webpages within the same website. For example, original webpage **104** associated with variant **342** has a bounce rate of 78.25% and modified webpage with variant **344** has a bounce rate of 21.01%.

[0083] Column **388** indicates the average amount of time spent on the website by users for each of variants **342-346**. For example, users spent on average of 3.20 minutes on the website when original page **104** is displayed. Users on average spent 11.43 minutes on the website when variant **344** was displayed on the modified webpage.

[0084] Column **390** indicates the average number of pages downloaded per user visit for each of variants **342-346**. For example, users downloaded on average 2.3 webpages from the website when original webpage **104** was displayed. Users downloaded on average 7.8 webpages from the website when variant **344** was displayed. Thus, in this example, it appears that variant **344** may cause users to spend more time on the website and view more pages while on the website.

[0085] FIG. **11** depicts an example of a user interface displaying locations of variants in a webpage. A push pin **402** is displayed next to element **128**E indicating that the original content of element **128**E has been changed to the content currently displayed within modified webpage **104**. The operator may delete the variant for element **128**E by selecting pushpin **402** and entering a delete command.

[0086] A performance table **400** is displayed in user interface **122** next to icon element **128**G. Performance table **400** may identify a percentage of mouse clicks on icon element **128**G for each of variants **342-346**. For example, the data captured by the CM agent indicates that icon element **128**G is selected 5.3% of the time when the original webpage **104** associated with variant **342** is displayed. The data also indicates that icon element **128**G is selected 15.02% of the time when variant **344** is displayed on modified webpage **104** and icon **128**G is selected 32.73% of the time when variant **346** is displayed on webpage **104**. Thus, it appears that using variant **346** is likely to increase the number of mouse clicks on element **128**G.

[0087] FIG. **12** depicts an example of a user interface configured to display heat maps for different variants of webpage **104**. In this example, the heat map is displayed in overlay **124** over modified webpage **104** displaying variant **346** of element **128**E. Shaded areas **410** represent places on webpage **104** where a user performed a mouse click. In one example, shaded areas **410** are darker when more mouse clicks are

detected in the same area. Heat maps can be displayed for each variant indicating how the variants influence user behavior on webpage **104**.

[0088] FIG. **13** shows a diagrammatic representation of a computing device for a machine in the example electronic form of a computer system **1000**, within which a set of instructions cause the machine to perform any one or more of the methodologies discussed herein can be executed. In various example embodiments, the machine operates as a standalone device or can be connected (e.g., networked) to other machines. In a networked deployment, the machine can operate in the capacity of a server or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine can be a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a portable music player (e.g., a portable hard drive audio device such as an Moving Picture Experts Group Audio Layer 3 (MP3) player, a web appliance, a network router, a switch, a bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0089] The example computer system **1000** includes an engine or multiple engines **1002** (e.g., a central processing unit (CPU), a graphics processing unit (GPU), or both), and a main memory **1004** and a static memory **1006**, which communicate with each other via a bus **1008**. The computer system **1000** can further include a video display unit **1010** (e.g., a liquid crystal displays (LCD) or a cathode ray tube (CRT)). The computer system **1000** also includes an alphanumeric input device **1012** (e.g., a keyboard), a cursor control device **1014** (e.g., a mouse), a disk drive unit **1016**, a signal generation device **1018** (e.g., a speaker) and a network interface device **1020**.

[0090] The disk drive unit **1016** includes a computer-readable medium **1022** on which is stored one or more sets of instructions and data structures (e.g., instructions **1024**) embodying or utilized by any one or more of the methodologies or functions described herein. The instructions **1024** can also reside, completely or at least partially, within the main memory **1004** and/or within the engines **1002** during execution thereof by the computer system **1000**. The main memory **1004** and the engines **1002** also constitute machine-readable media.

[0091] The instructions **1024** can further be transmitted or received over a network **1026** via the network interface device **1020** utilizing any one of a number of well-known transfer protocols (e.g., Hyper Text Transfer Protocol (HTTP), CAN, Serial, Modbus).

[0092] While the computer-readable medium **1022** is shown in an example embodiment to be a single medium, the term "computer-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that store the one or more sets of instructions. The term "computer-readable medium" shall also be taken to include any medium that is capable of storing, encoding, or carrying a set of instructions for execution by the machine and that causes the machine to perform any one or more of the methodologies of the present application, or that is capable of storing, encoding, or carrying data structures utilized by or associated with such a set of instructions. The term "computer-readable medium" shall accordingly be taken to include, but not be limited to, solid-state memories, optical and magnetic media. Such media can also include, without limitation, hard disks, floppy disks, flash memory cards, digital video disks, random access memory (RAMs), read only memory (ROMs), and the like.

[0093] The example embodiments described herein can be implemented in an operating environment comprising computer-executable instructions (e.g., software) installed on a computer, in hardware, or in a combination of software and hardware. The computer-executable instructions can be written in a computer programming language or can be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interfaces to a variety of operating systems. Although not limited thereto, computer software programs for implementing the present method can be written in any number of suitable programming languages such as, for example, Hyper text Markup Language (HTML), Dynamic HTML, Extensible Markup Language (XML), Extensible Stylesheet Language (XSL), Document Style Semantics and Specification Language (DSSSL), Cascading Style Sheets (CSS), Synchronized Multimedia Integration Language (SMIL), Wireless Markup Language (WML), Java™, Jini™, C, C++, Perl, UNIX Shell, Visual Basic or Visual Basic Script, Virtual Reality Markup Language (VRML), ColdFusion™ or other compilers, assemblers, interpreters or other computer languages or platforms.

[0094] Thus, systems and methods for page setup and real-time content replacement in A/B testing have been described. Although embodiments have been described with reference to specific example embodiments, it will be evident that various modifications and changes can be made to these example embodiments without departing from the broader spirit and scope of the present application. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

[0095] The detailed description includes references to the accompanying drawings, which form a part of the detailed description. The drawings show illustrations in accordance with example embodiments. These example embodiments, which are also referred to herein as "examples," are described in enough detail to enable those skilled in the art to practice the present subject matter. The embodiments can be combined, other embodiments can be utilized, or structural, logical and electrical changes can be made without departing from the scope of what is claimed. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope is defined by the appended claims and their equivalents.

[0096] In this document, the terms "a" or "an" are used, as is common in patent documents, to include one or more than one. In this document, the term "or" is used to refer to a nonexclusive "or," such that "A or B" includes "A but not B," "B but not A," and "A and B," unless otherwise indicated. Furthermore, all publications, patents, and patent documents referred to in this document are incorporated by reference herein in their entirety, as though individually incorporated by reference. In the event of inconsistent usages between this document and those documents so incorporated by reference, the usage in the incorporated reference(s) should be consid-

ered supplementary to that of this document; for irreconcilable inconsistencies, the usage in this document controls.

[0097] Having described and illustrated the principles of various examples, it should be apparent that the examples may be modified in arrangement and detail without departing from such principles. Claim is made to all modifications and variations coming within the spirit and scope of the following claims.

1. A method, comprising:
intercepting a request to render a control content by a client application;
intercepting the control content as it is being served from a content source;
based on predetermined criteria, dynamically replacing the control content with a replacement content, the replacement content being an alternative version of the control content differing from the control content in one or more alternative elements; and
selectively rendering the replacement content via the client application.

2. The method of claim **1**, wherein the replacement content is created by using a graphical interface overlaying the control content, the graphical interface configured to:
directly select an element of the control content;
select the alternative version of the control content;
edit the element of the control content; and
save the element as part of the replacement content.

3. The method of claim **2**, wherein the graphical interface overlaying the control content displays relative popularities of elements.

4. The computer-implemented method of claim **2**, wherein the editing of the control content is performed within a Hyper Text Markup Language (HTML) editor, image editor, or media editor.

5. The method of claim **2**, wherein an Hyper Text Markup Language (HTML), image, or media selector allows the selecting of the element of the control content underlying the graphical interface.

6. The method of claim **2**, further comprising rendering statistics related to the real-time content replacement on the graphical interface overlaying the control content.

7. The method of claim **6**, wherein the statistics are communicated to an operator associated via the graphical interface, reports files, or report.

8. The method of claim **1**, wherein the one or more alternative elements are tracked internally by being associated with one or more parent elements.

9. The method of claim **1**, wherein the replacement of the control content with the alternative version is enabled by a Rich Internet Application (RIA).

10. The method of claim **1**, wherein the replacement of the control content with the alternative version is enabled by Java script embedded in the control content.

11. The method of claim **1**, wherein the predetermined criteria comprises distributing various versions of the alternative content evenly to different users.

12. The method of claim **1**, wherein the predetermined criteria comprises distributing various versions based on geographical location of users.

13. The method of claim **1**, wherein the predetermined criteria comprises distributing various versions based on gender, income, or interests of users.

14. The method of claim **1**, wherein the dynamic replacement is performed by code embedded in a webpage containing the control content.

15. A system, comprising:
a processing device configured to:
intercept a request to render a control content and to intercept the control content as it is being served from a content source;
dynamically replace, based on predetermined criteria, the control content with a replacement content, the replacement content being an alternative version of the control content differing from the control content in one or more alternative elements; and
selectively render the replacement content.

16. The system of claim **15**, further comprising logic circuitry configured to overlay a graphical interface over the control content, the graphical interface enabling editing the element of the control content and saving the element as part of the replacement content.

17. The system of claim **16**, wherein the logic circuitry is configured to display relative popularities of elements in the control content.

18. The system of claim **16**, wherein the logic circuitry is configured to use a Hyper Text Markup Language (HTML) editor, image editor, or media editor to select and edit elements of the control content.

19. The system of claim **16**, wherein the logic circuitry is configured to render statistics related to the real-time content replacement of elements in the control content.

20. The system of claim **16**, wherein the logic circuitry is configured to track usage of control content displaying the alternative elements.

21. The system of claim **16**, wherein the logic circuitry is configured to generate the criteria for controlling how replacement content is distributed to users.

22. The system of claim **16**, wherein the logic circuitry is configured to enable distribution of various versions of the replacement content based on geographical location of the users

23. The system of claim **16**, wherein the logic circuitry is configured to enable distribution of various versions of the replacement content based on gender, income, or interests of users.

24. A computer-readable medium comprising instructions, which when executed by one or more computers, perform the following operations:
intercepting a request to render a control content by a client application;
intercepting the control content as it is being served from a content source;
based on predetermined criteria, dynamically replacing the control content with a replacement content, the replacement content being an alternative version of the control content differing from the control content in one or more alternative elements; and
rendering the replacement content.

25. The computer-readable medium of claim **24** wherein the control content is contained in webpages and the alternative versions of the control content comprise variants that are rendered in a web Brower instead of original elements in the webpages.

26. The computer-readable medium of claim **25** wherein the control content is replaced by changing a hyper text markup language (HTML) of the webpage to display the replacement content.

* * * * *