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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2007 (11.10.2007)

PCT

(10) International Publication Number
WO 2007/113573 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/GB2007/001275

(22) International Filing Date: 5 April 2007 (05.04.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0606889.4 5 April 2006 (05.04.2006) GB

(71) Applicant *(for all designated States except US)*: **BOX UK LIMITED** [GB/GB]; 46-48 Charles Street, Cardiff, CF10 2GE, United Kingdom (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ZAMBONINI, Daniel** [GB/GB]; c/o Box UK Limited, 46-48 Charles Street, Cardiff CF10 2GE (GB). **PEARSE, Jon** [GB/GB]; c/o Box UK Limited, 46-48 Charles Street, Cardiff CF10 2GE (GB).

(74) Agents: **RICHARDSON, Mark Jonathan** et al.; Keltie, Fleet Place House, 2 Fleet Place, London EC4M 7ET (GB).

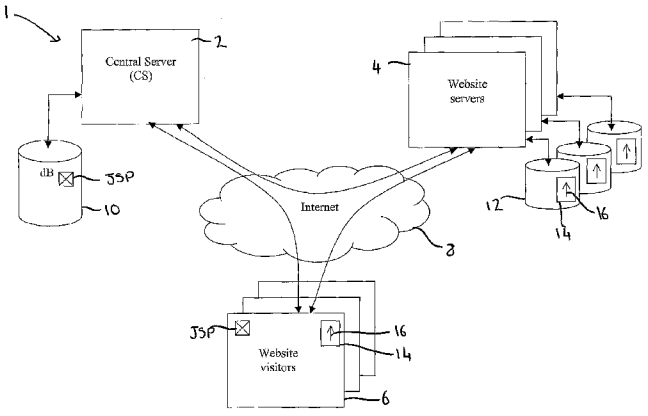
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** AUTOMATED MEASURING OF INTERACTION WITH USER INTERFACES



(57) **Abstract:** A recorder for recording interaction data relating to multiple user interaction events with a GUI presented at a user terminal is described. The recorder comprises: script transmitting means for transmitting an executable script to the user terminal in response to a first request; interaction data receiving means for receiving interaction data generated by execution of the script on the user terminal, the interaction data describing each user interaction event occurring with the GUI and including a set of co-ordinates specifying the location within the GUI of each user interaction event and an identifier identifying the GUI to which the interaction data relates; and a data store arranged to store the received interaction data. The interaction data receiving means and the data store are arranged to receive and store a sequence of user interaction data relating to the GUI in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI is possible by viewing of the currently stored interaction data. A data capture device and a data interaction analyser are also disclosed.

WO 2007/113573 A2

Automated Measuring of Interaction with User Interfaces

Technical Field

The present invention relates to automated measuring of interaction with user interfaces. In particular, the invention relates to the recordal and reporting of user interaction with a user interface. More particularly, though not exclusively the present invention relates to an interaction recorder for recording user interaction with a website, and an interaction analyser for generating visual reports for identifying the usability of the website.

Background to the Invention

Numerous web analytics tools exist, which can be used to identify users' interactions with a web site, the majority of which identify the number of times a page or content item has been served by the web site. In addition, there exist software solutions that provide an analysis of the clicks on a single web page, presenting the data to the user purely as a mass of data points.

One disadvantage with these tools is that they do not provide clear visual analysis of the click information of any web page and as a result they require that users have prior technical knowledge and analytical skills in order to interpret the result data.

Existing web analysis tools operate on a campaign basis whereby a website is monitored for a period of time. The results collated at the end of that period are then interpreted and provided to the user for consideration. This is useful for many users but has limited value for websites in which content and/or format varies on a frequent basis, for example a news website. In these cases the reporting can become out of date relatively quickly.

A known analysis tool provided by Atlas Solutions™ plots a user's mouse clicks on any webpage. However, a severe limitation with this tool is that the clicks are merely represented as dots on a page. As such, a large number of clicks in any particular area results in a large number of dots, but multiple clicks on the same co-ordinate do not appear any different than a single click on the same location. Additionally, it is very

difficult for a user to look at these analysis results in order to determine accurately the areas that receive the majority of clicks.

Summary of the Invention

According to a first aspect of the invention there is provided an interaction recorder for recording interaction data relating to multiple user interaction events with a GUI presented at a user terminal, the recorder comprising: script transmitting means for transmitting an executable script to the user terminal in response to a first request, interaction data receiving means for receiving interaction data generated by execution of the script on the user terminal, the interaction data describing each user interaction event occurring with the GUI and including a set of co-ordinates specifying the location within the GUI of each user interaction event and an identifier identifying the GUI to which the interaction data relates; and a data store arranged to store the received interaction data; wherein interaction data receiving means and the data store are arranged to receive and store a sequence of user interaction data relating to the GUI in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI is possible by viewing of the currently stored interaction data.

One advantage of the above aspect is that details regarding how visitors interact with a GUI/website are recorded at the time of the interaction and can be viewed by a user instantaneously, as required, without the need to wait for a statistically significant subset of user behaviour to be recorded for analysis to be carried out.

Preferably, the data store is arranged to store the interaction data in a plurality of session logs wherein each session log comprises data relating to a single user's interaction with a GUI over a period of time.

More preferably, the interaction data includes session data relating to session attributes describing the environment in which the interaction events are recorded.

Optionally, the session data comprises one or more attributes selected from the group comprising: a session start date, a session start time, a browser name, a browser version, a screen resolution, a screen colour depth, a default font size, an IP address, and an operating system of the remote terminal.

Information regarding the environment in which the interaction took place may be advantageously used to provide further insight regarding the users of the GUI.

Typically, the interaction data includes event data relating to the single user's interaction with the GUI.

Optionally, the event data comprises one or more attributes selected from the group comprising: positional co-ordinates, an event date, an event time, a target positional co-ordinate, a target ID, a horizontal scroll variable, a vertical scroll variable, a key element positional co-ordinate, a browser window size, a previous GUI page ID, an elapsed time variable, a session sequence number, a GUI sequence number, a current GUI page ID, and a URI of a selected element.

In a preferred embodiment, the interaction recorder further comprises a session ID generator which is arranged to generate and transmit a session ID to the user terminal in response to a script generated request.

Preferably, the data store is arranged to store the interaction data using the session ID, the interaction data being stored in a relational table. This offers the advantage that all of the event data regarding interactions with the GUI can be linked together to provide insight regarding a single user's interaction with the GUI: a single user's interaction being recorded against a particular session ID.

Typically, the GUI comprises a plurality of interconnected GUI pages, and the interaction data receiving means is arranged to receive interaction data describing user interaction events occurring within the GUI pages.

Preferably, the interaction recorder is arranged to record interaction data relating to multiple user interaction events for a plurality of GUIs, and further comprises a site ID generator arranged to generate a unique site ID for each of the plurality of GUIs.

Advantageously, the data store is arranged to link together stored session logs relating to the unique site ID. This offers the advantage that a user wishing to view results regarding their GUI can view all or some of the records which have been linked to the

unique site ID, and also ensures that other users cannot view the results relating to other GUIs associated with different site IDs not linked to their own account.

In a preferred embodiment, the interaction recorder comprises a pointer creation means which is arranged to create a pointer script, identifying the executable script stored at the interaction recorder wherein execution of the pointer script at the user terminal generates the first request.

Preferably, the pointer creation means is arranged to generate the pointer script by use of a template and the unique site ID. In this way, the central server advantageously need only store one version of the executable script, and the pointer script ensures the executable script is returned to the user terminal. Furthermore, the site ID within the pointer script ensures the interaction data is recorded in the data store in relation to the site ID.

In a preferred embodiment, the pointer creation means is arranged to generate the pointer script by additional use of a key element ID. This offers the advantage that meaningful results may be plotted for centred websites. For example, rather than plotting an absolute location of a clicks, the click is plotted relative to a known reference point e.g. a logo within the webpage.

Typically, the GUI comprises a website.

According to a second aspect of the invention there is provided a method for recording interaction data relating to multiple user interaction events with a GUI presented at a user terminal, the method comprising: transmitting an executable script to the user terminal in response to receiving a first request, receiving interaction data generated by execution of the script on the user terminal, the interaction data describing each user interaction event occurring with the GUI and including a set of co-ordinates specifying the location within the GUI of each user interaction event and an identifier identifying the GUI to which the interaction data relates, and storing the received interaction data, wherein the receiving and storing steps include receiving and storing a sequence of user interaction data relating to the GUI in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI is possible by viewing of the currently stored interaction data.

According to a further aspect of the invention there is provided a graphical user interface (GUI) interaction analyser for analysing the results of multiple user-interaction events with a GUI page presented at a user terminal, the analyser comprising: retrieval means for retrieving interaction data stored in a data store, the interaction data being stored in plurality of interaction records, each of which includes co-ordinates specifying the location within the GUI page of a user interaction event and an identifier identifying the GUI page to which the interaction data record relates, a graphical processor for processing the interaction data to generate a three-dimensional representation of the interaction data, two dimensions being determined by the location of the interaction event on the identified GUI page and the third dimension being determined by a further parameter relating to the interaction event occurring at a location of the identified GUI page, and map generation means for presenting a map showing the three-dimensional interaction data superimposed on a representation of the identified GUI page.

In the description the terms 'click on', 'clicked' or 'clicking' represent an interaction event with the GUI/web page.

The interaction analyser offers the advantages that user experience and usability of the web page is readily identifiable. Visual reports improve the ease with which the popularity or unpopularity of components within the web page can be identified. Furthermore, a non-technical audience can easily understand the graphic nature of the generated reports.

In a preferred embodiment, the retrieval means further comprises a filter arranged to compose a query request to retrieve the interaction data from the data store. This offers the advantage that meaningful reports may be obtained using a subset of the interaction data recorded.

Typically, the filter is arranged to receive from the user terminal at least one filter parameter selected from the group comprising: a start date, an end date, a start time, and end time, a screen resolution, a colour depth, a browser window size, a country, an organisation, a browser name, a browser version, and an operating system, and wherein the query request to effect interaction data retrieval includes at least one filter parameter. Advantageously, the present invention can provide highly accurate reports

because the reports may be generated on the basis of interaction data recorded in a similar/identical environment to the one in which the report is being viewed.

Advantageously, the graphical processor is arranged to receive the results of the request query and to effect generation of the three-dimensional representation of the interaction data.

Typically, the interaction data set includes one or more attributes selected from the group comprising: a session ID, session start date, a session start time, a browser name, a browser version, a screen resolution, a screen colour depth, a default font size, an IP address, and an operating system of the remote terminal, absolute positional co-ordinates, an event date, an event time, a target positional co-ordinate, a target ID, a horizontal scroll variable, a vertical scroll variable, a key element positional co-ordinate, a browser window size, a previous GUI page ID, an elapsed time variable, a session sequence number, a GUI sequence number, a current GUI page ID, and a URI of a selected element.

In a preferred embodiment, the graphical processor is arranged to determine the further parameter as the frequency of interaction events occurring at each location of the identified GUI page. This offers the advantage of showing not only the locations on the web page where users are clicking, but also the frequency/volume of the clicks in that location. This visual information immediately provides the ability to identify the true popularity or unpopularity of locations within the website.

In a further preferred embodiment, the graphical processor is arranged to determine the further parameter as the accuracy of interaction events occurring at an area location of the identified GUI. This offers the advantage of identifying every click on a web page, including those 'on target' (i.e. on elements such as hyperlinks, hyperlinked images, form fields, buttons, and elements with onclick handlers, etc) and those that are 'off target'. This is particularly useful in identifying 'near misses' and where visitors make clicks on elements, which they think are targets but which are not.

Typically, the area location represents an interactive button of the GUI.

Advantageously, the graphical processor is arranged to determine the accuracy of the interaction event by determining if the location of the interaction event corresponds to the area location of the interactive button.

Preferably, the graphical processor is arranged to represent graphically the accuracy of the interaction event by using at least two different colours.

In a preferred embodiment, the graphical processor is arranged to transform absolute positional co-ordinates of the interaction event into relative positional co-ordinates with respect to reference co-ordinates, and wherein the map generation means is arranged to generate the graphical representation of the interaction event at the relative positional co-ordinates.

In this preferred embodiment, the reference co-ordinates may be the co-ordinates of a key element. This offers the advantage that clicks can be plotted relative to a key element which has been specified for a centred website.

Preferably, the graphical processor is arranged to use the key element ID to locate the key element, within the identified GUI, and determine the positional co-ordinates of the key element, such that the relative positional co-ordinates of the interaction event can be determined.

In another preferred embodiment, the reference co-ordinates are the positional co-ordinates of a target element being identified by a unique target ID associated with the interaction data set. This offers the advantage that clicks can be plotted relative to the target element which has been selected, meaning that for dynamic websites, where selectable elements move position over time, clicks are always potted relative to the element selected rather than the absolute position on the page.

Preferably, the graphical processor is arranged use the unique target ID to locate the target element, within the identified GUI, and determine the positional co-ordinates of the target element, such that the relative positional co-ordinates of the interaction event can be determined.

Advantageously, the target ID is unique for each of a plurality of targets, and comprises a fingerprint of an associated target GUI element, wherein the graphical

processor comprises a comparator for comparing the fingerprint to a newly generated fingerprint representation of the target element of the representation of the identified GUI. This is advantageous because the element selected at the time of the interaction can be identified at the time the report is generated by virtue of the fact that the unique target ID generated for both is identical.

In a preferred embodiment, the graphical processor further comprises a scaling module arranged to reposition the positional co-ordinates of the location of the event as a function of a calculated difference between the screen resolution of the user terminal, whereupon the interaction event occurred, and the screen resolution of the user terminal whereupon the event is graphically represented. This offers the advantage of being able to account for differences in screen resolutions when plotting clicks recorded via screens of different resolutions.

Preferably, the scaling module is further arranged to reposition the positional co-ordinates of the location of the event as a function of a calculated difference between the default font size of the user terminal, whereupon the interaction event occurred, and the default font size of the user terminal whereupon the event is graphically represented. This offers the advantage of being able to account for differences in default font sizes when plotting clicks recorded via browsers which are configured differently in terms of default font size.

In a preferred embodiment, the map generation means is arranged to display the graphical representation of the interaction event centred at the received positional co-ordinates or the relative positional co-ordinates.

Advantageously, the map generation means is arranged to generate a display of the third dimension as colour-coded graphical representation of the frequency of the interaction events occurring at each location. Colour coding of the results is advantageous because the generated visual report is easily understandable.

Preferably, the map generation means is arranged to select a colour for the representation from a scale of colours indicating a range of interaction event frequencies.

Typically, there are a plurality of GUI pages and each page has a different scale.

Preferably, the scale is determined by the maximum and minimum frequencies of the three-dimensional interaction data for a given GUI page.

In a preferred embodiment, the map generation means is arranged to generate a display of the third dimension of the interaction data constantly with the two-dimensional interactive event location data.

Typically, the map generation means may be arranged to generate a display of the third dimension of the interactive data upon user selection of an element of the GUI to which the third dimension data relates.

Preferably, the map generation means is arranged to generate a display of the third dimension as a pop-up box which graphically represents statistical data regarding the frequency of the interaction events occurring at each location. This advantageously provides additional information which adds to the insight gathered through the visual report. For example, the pop-up box may contain statistical information regarding the volume of clicks made on a particular target associated with the location.

Optionally, the graphical processor is arranged to receive preferences regarding the statistical data which is displayed within the pop-up box.

In a preferred embodiment, the graphical processor is arranged to determine a value of the third dimension at the location of the interaction event and to assign a different value of the third dimension at a proximate location adjacent to the interaction event location.

Typically, the graphical processor is arranged to assign different values to a plurality of proximate locations surrounding the interaction event location, such that the effects of the third dimension are realised as a distribution over the proximate locations and interaction event location.

Preferably, the distribution forms a bell-shaped curve.

The graduated nature of the use of colour, to indicate the popularity of locations, offers the advantage of indicating how often one location is clicked in relation to other areas.

In a preferred embodiment, the map generation means further comprises export means arranged to convert the graphical representation into a portable image file, wherein upon user selection the portable image file is transmitted from the GUI interaction analyser to a viewing terminal such that it can be viewed or stored at the viewing terminal. This offers the advantage of enabling generated reports to be saved.

Typically, the GUI page comprises a page of a website.

According to a further aspect of the invention there is provided, a method of analysing the results of multiple user-interaction events with a GUI page presented at a user terminal, the method comprising retrieving interaction data stored in a data store; the interaction data being stored in plurality of interaction records, each of which includes co-ordinates specifying the location within the GUI page of a user interaction event and an identifier identifying the GUI page to which the interaction data record relates, processing the interaction data to generate a three-dimensional representation of the interaction data, two dimensions being determined by the location of the interaction event on the identified GUI page and the third dimension being determined by a further parameter relating to the interaction event occurring at a location of the identified GUI page, and generating a graphical map showing the three-dimensional interaction data superimposed on a representation of the identified GUI page.

According to another aspect of the invention there is provided a combination of a recorder and an analyser as discussed above.

It is an advantage of recording the results in real-time, i.e. as the interaction takes place, that reports can also be generated in real-time. This therefore, provides the ability that quick reactions and decisions can be made on the basis of the results/reports provided.

According to another aspect of the invention there is provided a capture device for capturing interaction data relating to multiple user interaction events with a GUI page presented at a user terminal, the device comprising receiving means for receiving an executable script from an interaction recording device; script execution means for executing the script concurrently with the operation of the GUI page at the user terminal, the script execution means generating interaction data describing each user interaction event occurring with the GUI page, the interaction data including a set of

co-ordinates specifying the location within the GUI page of each user interaction event and a unique identifier identifying the GUI page to which the interaction data relates; and transmission means for transmitting the user interaction data to the interaction recording device; wherein script execution means and the transmission means are arranged to generate and transmit a sequence of user interaction data relating to the GUI page in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI page is made possible from consideration of the transmitted information.

Advantageously, the interaction data is recorded at the user terminal, in the background, and has no unwanted affect on the user terminal.

In a preferred embodiment, the capture device further comprises request transmitting means arranged to transmit a first request to the interaction recording device, the first request requesting the executable script from the interaction recording device.

Typically, the script execution means is arranged to generate a request for a session ID and the transmission means is arranged to transmit the request for the session ID to the interaction recording device.

The receiving means is preferably arranged to receive the requested session ID, and store the session ID in a temporary file.

Advantageously, the script execution means is arranged to determine session data describing the environment in which the interaction events are recorded.

Optionally, the session data comprises one or more session attributes selected from the group comprising: a session start date, a session start time, a browser name, a browser version, a screen resolution, a screen colour depth, a default font size, an IP address, and an operating system of the remote terminal.

Typically, the session ID request includes one or more of the determined session attributes.

In a preferred embodiment, the script execution means is arranged to generate event data relating to a single user's interaction with the GUI page.

Preferably, the transmission means is arranged, for each interaction event, to transmit the received session ID together with the event data, specifying at least the positional co-ordinates of the interaction event, and the GUI page ID upon which the interaction event occurred.

Optionally, the event data transmitted to the interaction recording device includes one or more further event attributes and the one or more further event attributes may be selected from the group comprising: an event date, an event time, a target positional co-ordinate, a unique target ID, a horizontal pixel scroll variable, a vertical pixel scroll variable, a key element positional co-ordinate, a browser width variable, a browser height variable, an identifier for a previous GUI page, an elapsed time value, a session sequence number, a GUI sequence number, and an URI of a selected element.

In a preferred embodiment, the capture device may comprise means for identifying a selected target element, and means for generating a unique target ID from the description of the element.

Typically, the description comprises a mark-up language script describing the target element and the generating means is arranged to use a predetermined subset of the mark-up language script to generate the unique target ID. This offers the advantage of keeping the time to generate the target ID to a minimum because if the mark-up language script is very large (i.e. perhaps relating to a drop down select box containing a large number of options) the time to generate the target ID will be larger if the whole script is used.

Advantageously, the generating means is arranged to reorder the mark-up language script into a ranked ordered list. Re-ordering the script ensures the generating means generates the unique target ID in the same manner as the analyser to arrive at identical target IDs.

Preferably, the generating means comprising a hashing algorithm for creating a fingerprint from the mark-up language script, the fingerprint representing the unique target ID.

Typically, the executable script is a JavaScript program, and the GUI page may comprise a page of a website.

According to another aspect of the invention there is provided a method of capturing interaction data relating to multiple user interaction events with a GUI page presented at a user terminal, the device comprising: receiving an executable script from an interaction recording source, executing the script concurrently with the operation of the GUI page at the user terminal, the executing step including generating interaction data describing each user interaction event occurring with the GUI page, the interaction data including a set of co-ordinates specifying the location within the GUI page of each user interaction event and a unique identifier identifying the GUI page to which the interaction data relates, and transmitting the user interaction data to the interaction recording device, wherein the transmitting step and the generating step comprise and generating and transmitting a sequence of user interaction data relating to the GUI page in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI page is made possible from consideration of the transmitted information.

According to yet another aspect of the invention there is provided a mark-up language representation of a page of a website for display in a browser at a user terminal, the page comprising at least one element, the element including a mark-up language code segment defining a first content component and an executable script defining a website for provision of a second content component, wherein if the browser is unable to execute the script then the mark-up language code segment is displayed in the element. This offers the advantage that should a browser be unable to execute the executable script, the first content component is displayed such that a user is always presented with some content for the element rather than no content.

Brief Description of Drawings

Preferred embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic system diagram showing a central server (CS) embodying the present invention and connections between the central server via the Internet to remote terminals of website visitors and website servers;

Figure 2 is a flowchart showing an overview of the operation of an embodiment of the present invention shown in Figure 1;

Figure 3 is a block diagram of four main modules of the CS of Figure 1;

Figure 4 is a schematic block diagram of three main sections of the CS database shown in Figure 1 including a user account section and session log section shown in Figure 3 and a JavaScript code section;

Figure 5 is a schematic block diagram of an exemplary relational table illustrating the relationships between data stored in the CS database of Figure 4;

Figure 6 is schematic block diagram of an account manager shown in Figure 3;

Figures 7a to 7c are screenshots of a centred website, at increasing screen resolutions, showing the location of a click on a target with reference to the top left-hand corner of the web page and the top left-hand corner of a key element;

Figure 8 is a is a flowchart showing the method steps required for a "website configuration" stage of Figure 2;

Figure 9 is schematic block diagram of a real-time interaction recorder of Figure 3, Figure 10 is a detailed flowchart showing the steps of operation of a "set-up of interaction data recordal" stage and an "interaction data recordal process" stage of Figure 2;

Figure 11 is a schematic block diagram of a reporting module shown in Figure 3;

Figures 12a and 12b are schematic diagrams of Internet browser configurations with differing resolutions showing the position of clicks being reported, by the reporting module of Figure 11, with and without scaling;

Figure 13 is a screenshot exemplifying a heat map report generated by the reporting module, of Figure 5, in accordance with one embodiment of the present invention;

Figure 14 is a graphical image of a scale used as part of the heat map report shown in Figure 13;

Figure 15 is another screenshot showing a click map report generated by the reporting module of Figure 5;

Figure 16 is a screenshot showing a hover map report generated by the reporting module of Figure 5;

Figure 17 is a schematic diagram of an Internet browser configuration, similar to those in Figures 12a and 12b, showing the location of a click with reference to the top left-hand side of the browser window and the top left-hand side of a target element.

Figure 18 is a screenshot exemplifying a heat map report generated by the reporting module of Figure 5 when a follow targets filter option is selected;

Figure 19 is a screenshot exemplifying an A/B testing method which utilises a JavaScript pointer to facilitate A/B testing; and

Figure 20 is a screen shot exemplifying a click stream report generated by the reporting module of Figure 5.

Detailed Description of Preferred Embodiments

Described below is an exemplary single installation (central server) embodiment of the present invention. Alternative embodiments will be discussed thereafter.

Figure 1 is an overview of an example environment 1 where the present embodiment operates. As shown, a central server (CS) 2, a plurality of website servers (WS) 4 and a plurality of website visitors 6 communicate with each other via the Internet 8. The central server 2 has access to a CS database 10 which stores a JavaScript program JSP. The website servers 4 each have access to a WS database 12 which stores an HTML file 14 for the website. Within the HTML file 14 is a pointer 16, which is a small executable code that is directed to (points to) the JSP.

The website servers 4 each host a website that the website visitor terminals 6 can remotely access, via an internet browser, using an Internet-enabled device (remote terminal), for example a personal computer, personal digital assistant (PDA), or a mobile telephone. Remote terminals accessing/browsing the website, download the HTML file 14 including the pointer 16, and this in turn enables the JSP to be downloaded to the remote terminal 6 (described in detail later).

Figure 2 shows a high-level overview 20 of the operation of the present embodiment. An administrator (user) of a website hosted by a website server, wishing to analyse how website visitor terminals 6 interact with and use the website, registers, at step 22, details of their website with the central server 2. The user configures, at step 24, their website for use with the embodiment of the present invention. The configuration step

24 includes amending the HTML file 14 of the website to include the pointer 16 to the JSP. When a website visitor views a web page of a registered website, the pointer 16 operates, at step 26, to set-up an interaction data recordal procedure/process by retrieving the JSP from the CS database 10. The JSP is downloaded to the remote terminal 6, and executes such that when visitors click on web pages of the website, interaction data, i.e. information regarding the viewing session, is captured, transmitted to and recorded, at step 28, in the CS database.

The interaction data comprises session data and click data. The session data includes details relating to the remote terminal 6 holding the viewing session, and the click data includes details relating to each visitor click (defined later) on the downloaded page of the website. The interaction data provides an insight into how website visitors interact with the website pages, and the interaction data is reported, at step 30, to the user. The interaction data is reported in a visual format for example as graphical heat maps (described later) that readily identify where visitors have clicked, including an indication of the popularity of certain selectable icons, such as hyperlinks, and the unpopularity of others.

Referring now to Figure 3, the composition of the central server is described. The central server 2 comprises four major components including a communications module 32 for handling communication requests to and from the central server 2, an account manager 34 for facilitating the set-up and management of user accounts, a real-time interaction recorder 36 for receiving and recording the interaction data in session logs, and a reporting module 38 for providing real-time reports to the users regarding the interaction data as required. As described above, the central server 2 has access to the CS database 10 with the account manager 34 accessing a user accounts section 40 and a JavaScript code section 42 in the CS database 10, and the real-time interaction recorder 36 and the reporting module 38 accessing the session logs 44.

An outline of the CS database 10 is shown in Figure 4. The CS database 10 stores data relating to user accounts, including login information and website configuration information in the user accounts section 40 of the CS database. Data relating to visitors' interaction with registered websites is stored in a session logs section 44 of the CS database 10, and the JSP and a template for the pointer are stored in the JavaScript code 42 section of the CS database 10.

The interaction data is stored in the CS database 10 using standard relational tables. Figure 5 is an example of a standard relational table 50 showing a high-level overview of the kinds of key relational information stored in the CS database 10. In the relational table 50 related data is linked. For example, when a visitor views a website, a session ID is assigned and each time the visitor clicks on a web page of the website, information relating to each click is linked to that web page and the relevant session ID. In practice, each user may register a plurality of websites and each website may contain a plurality of web pages. In addition, a plurality of visitors to the registered websites will result in a plurality of recorded sessions, each having a unique session ID, and within each recorded session, each click results in a record of information relating to that click.

Relational tables 50 are preferable for real-time reporting since the information can be retrieved from the relational table 50 very quickly. However, it is to be appreciated that there are other methods of storing and retrieving information in and from the database 10 which could be used with the present embodiment. For example, multi-dimensional cubes may be used. A multi-dimensional cube is similar to a relational table and is in fact built upon several specially designed relational tables. The way they differ is that tables are restricted to two dimensions (rows and columns) but cubes can have as many dimensions as required. An example of a simple cube would be one with three dimensions representing sales orders. The three dimensions could represent the store the order was placed in, the item sold and the date. Every cell in the cube is then identified by a unique store/item/date combination. The value of a cell would be the number of orders placed in that store for that item on that date. You can also get the number of orders placed in a store on a particular date by omitting the item value from your query.

As mentioned previously with reference to Figure 3, the account manager accesses the user accounts section 40 and JavaScript code section 42 of the CS database 10 and the real-time interaction recorder 36 and the reporting module 38 access the session logs 44 section of the CS database10. Although, as described above, it is clear that the data stored in the user accounts section 40 is related and linked to data stored in the session log section 44.

Communication requests to the central server 2 are handled by the communications module 32. Similarly, all data being transmitted from the central server 2 is also sent via the communications module 32, which is connected to each of the account manager 34, real-time interaction recorder 36, and reporting module 38.

The account manager 34 is arranged to handle the tasks of user account creation, dealing with requests to register websites, payment transactions, and facilitating access to configuration material required by the users to configure their websites for monitoring.

The real-time interaction recorder 36 is arranged to assign a new session ID each time a remote terminal 6 accesses a registered website, and is further arranged to record a plurality of sets of data including information relating to the remote terminal's browser and information relating to each time the visitor uses a mouse or other device to make a selection (click) from the options displayed on the website's web pages. The real-time interaction recorder 36 is arranged to store the received data within the relational tables 50 of the CS database 10.

The reporting module 38 is arranged to receive a request for a report from the remote terminal (viewing terminal) of a user connectable to the central server 2 via the Internet 8, process the request in order to retrieve the relevant data from the CS database 10 and format the results into visual reports to be transmitted to and displayed to the user. This is carried out in real-time and on demand from the user.

Each of the above modules are discussed in further detail below.

From Figure 6 it can be seen that, the account manager 34 comprises an account creation module 60 for handling the set-up of accounts for new users, an e-mail generator 62 for creating and sending an e-mail to the new users, a user login module 64 for restricting access to the system to users with valid login details, a website initialisation module 66 for handling user website registration requests, a payment handler 68 for handling payment transactions, a pointer generator 70 for generating the pointer, and a plan manager 72 for the configuration of user requirements.

The pointer generator 70 accesses the JS code section 42 of the CS database 10, and the account creation module 60, user login module 64, website initialisation module

66, payment handler 68 and plan manager 72 access the user accounts section 40 of the CS database 10.

The account creation module 60 is arranged to receive administrator/user details, including username, password and e-mail address information, which are recorded in the CS database 10 in order to enable users to login to the central server 2 to generate and retrieve desired reports regarding the website usage/interaction results. The account creation information is received by the central server 2 via online forms hosted by a web-serving element of the central server 2.

Upon registration of a user, the e-mail generator 62 sends an e-mail to the user's e-mail address, via the communications module 32. The e-mail confirms to the user their username and password details and provides a link to a webpage hosted by the web-serving element of the central server 2, where the user can login to their account in order to proceed with initialising their website. It is to be appreciated that it is not essential that an e-mail with this information is sent to the user and that the user may be directed to the login webpage after they have entered their details without receiving an e-mail.

The user login module 64 is arranged to authenticate the identity of each user at login, in accordance with their username and password details, and to restrict access to the reports which the user has the appropriate permission rights to access, i.e. only reports relating to the websites associated with their account. The user login module 64 is also arranged to permit the user to update their account details as necessary, for example, to add additional websites to be monitored.

The payment handler 68 is arranged to receive and process payment details entered by the user. Payment details are recorded securely using standard payment methods.

The plan manager 72 is arranged to present the user with sets of options, known as plans, relating to the type and level of monitoring/reporting they require. This advantageously enables the user to tailor the system to their individual needs. The different options/plans may be in terms of time, click, or visitor quotas. For example, a user may require their website to be monitored for a month, or until a specified number of clicks are recorded against the website, or a number of visitors have visited the website. It is to be appreciated that this list of options is not limiting and a user

may be presented with other options relating to measurable visitor interaction parameters.

When a user is logged in, has selected a plan to meet their requirements and effected appropriate payment, the user is able to initialise/register their website such that monitoring of the website can take place. The website initialisation module 66 is arranged to handle user requests to register their website of interest. The request from the user's remote terminal (not shown) comprises, at least, data relating to the URL of the website to be monitored.

In one embodiment, the system can account for whether the website to be registered is a centred website, and as such the registration request also comprises data relating to an ID of a main HTML element (known as a key element) of the website. Typically, this is a main logo of the website, which is usually located at the same X/Y co-ordinates on every web page of the website, although it can be any main user-specified element.

Typically, websites have a sufficiently low resolution in order to allow their web pages to be rendered on most screens, i.e. those with high or low resolutions. Where a screen has a higher resolution than the web page to be rendered, the web page does not fill the whole browser window and 'blank' space fills the remaining area. For a left justified website, the blank space is found to the right hand side of the web page. Centred websites are websites which are rendered centrally within the browser window, regardless of the resolution of the web pages. Figures 7 (a), (b) and (c) each show a screenshot 78 of a centred website being displayed at different (increasingly larger) screen resolutions.

Identifying a key element (in the case of Figure 7 the key element is the BOX UK logo) within the registration request permits the X/Y co-ordinates of clicks to be recorded (and displayed) relative to the co-ordinates of the key element of the page, rather than the absolute X/Y co-ordinates of the click as measured from the top-left corner of the browser window. In other words, the X/Y co-ordinates of the key element provide a reference point X from which the position of the click can be measured and plotted. As shown in Figures 7 (a), (b), and (c), the absolute X/Y screen co-ordinates of the click, represented by the diagonal lines A, are all different.

Whereas the X/Y co-ordinates relative to the main element, represented by the vertical lines B, are the same in Figures 7 (a), (b), and (c). In other words, the difference between the X/Y co-ordinates of the key element and the clicked object remains constant for different screen resolutions.

The website initialisation module 66 stores the URL and key element information in the CS database 10 in relation to the user's account information. The website initialisation module 66 also converts the website URL information in the request into a unique site ID, which is also stored in the CS database 10 in relation to the user's account information. The website initialisation module 66 passes the site ID and main element/ key element ID to the pointer generator 70 in order for it to be integrated into a section of JavaScript pointer code 16. The pointer generator 70 retrieves a template of JavaScript code which is supplemented with the site ID and key element ID.

The pointer 16 therefore is in itself a JavaScript code and is displayed in a text form to the user, together with instructions, such that the user can complete the website configuration step 24 of Figure 2. As mentioned above, the pointer 16 is a pointer to the JSP that is stored, in the present embodiment, at the central server 2, in the CS database 10. Each time a visitor requests a web page that includes the pointer 16, the visitor's remote terminal sends a JSP request to the central server 2 to access the JSP. The communications module 32 handles the request and returns the JSP to the remote terminal 6, where it is executed in order to facilitate recordal of the interaction data.

Figure 8 details the method steps 80 which a user must follow in order to configure their website to be monitored by the present embodiment of the invention. If not already logged to the website hosted by the central server 2, the user logs in, at step 82, and selects, at step 84, a link to access the pointer text. The user copies, at step 86, the displayed pointer text and enters, at step 88 the pointer text into the body section of the HTML file 14 or relevant template file of the appropriate website to be monitored (i.e. the website whose URL has been entered by the user during the registration process). The user can carry this operation out using standard 'copy' and 'paste' functions.

The user then saves and uploads the website, at step 90, to the associated website server 4, where the website can be accessed by visitors. Thereafter, the clicks of those

visitors are recorded along with other interaction information such that records of the website usage and interaction can be reported back to the user/administrator as required. This process is described in detail later.

An example of an HTML file 14, which does not include the pointer 16 is shown below:

```
<html>
 <body>
       Hello
 </body>
</html>
```

An HTML file including the pointer inserted is shown below:

```
<html>
  <body>
        Hello                    {the main body of the HTML file is found here}
      <script type="text/javascript"
      src="http://j.clickdensity.com/cr.js"></script>
      <script type="text/javascript">
      //<![CDATA[
      var clickdensity_siteID = 103;
       var clickdensity_keyElement = 'kvalue';
      //]]>
      </script>
 </body>
</html>
```

The pointer 16 comprises configuration information, in the form of the site ID which identifies the website being monitored, and the key element identifying the reference point on the web page, and is XHTML compliant.

In a preferred embodiment, the pointer text is entered just above the end body tag (i.e. just before </body> near the bottom of the HTML file). This is preferable because execution of the pointer code causes a window.onload event handler to initialise

recording of the click data. However, other JavaScript code present in the HTML file 14 of the webpage may also use this event handler and as such may result in the pointer code associated with the present embodiment to be overwritten. Advantageously, pasting the pointer text at the end of the body element ensures that the pointer is executed last and so cannot be overwritten.

When the JSP is downloaded, to the visitor's remote terminal 6, and executed, the real-time interaction recorder 36 is arranged to record the received interaction data from the remote terminal 6, including data associated with the remote terminal 6 and browser, and click data. As shown in Figure 9, the real-time interaction recorder 36 comprises a session ID assignor 92, session data recorder 94, and click data recorder 96. Each of the session ID assignor 92, session data recorder 94, and click data recorder 96 communicate with the communications module 32 and access the session logs section 44 of the CS database 10.

Execution of the JSP at the remote terminal 6 results in an initial GET request being sent to the central server 2. The initial GET request is passed via the communications module 32 to the session ID assignor 92 which is arranged to generate a new session ID.

The session ID assignor 92 is further arranged to return the session ID to the remote terminal 6 making the initial GET request. The session ID returned to the remote terminal 6 is stored in a temporary file (not shown), known as a cookie, at the remote terminal 6, and is used each time visitor click data is sent to the central server 2 in subsequent GET requests which are transmitted in response to each click.

The initial GET request also comprises the session data (known also as session environment data) relating to the remote terminal 6 making the request. The session data comprises the date and time at which the session commenced, the browser name and version for the browser being used to view the website, the resolution and colour depth of the remote terminal's screen, the default font size that the browser is using to display text on the screen, an IP address of the remote terminal 6 making the request, and an operating system of the remote terminal 6. A person skilled in the art will appreciate that not all of the above segments of information are essential for

implementing the present embodiment and also that additional segments of information may be included in the session data.

The session data of the initial GET request is passed to the session data recorder 94, which is arranged to store the session data in the CS database 10 in relation to the assigned session ID.

Once a browsing session has commenced, and until the visitor leaves or navigates away from the website being monitored, visitor interaction data or click data for every click made on the web page is recorded. It is to be appreciated that the term 'click' represents any user selection of, or interaction with, icons/elements on the web page, and this is not limited to mouse clicks. For example, the present embodiment is equally applicable to touch screens or panels.

For each click made by a visitor, the click data sent to the central server 2 comprises: an X and Y co-ordinate for the click, a date and time for the click, an X and Y (positional) co-ordinate of the target (the element the visitor clicked on), a Unique ID for the target, a variable for the horizontal and vertical pixels the user has scrolled prior to making the click, an X and Y co-ordinate for the key element (the element of the web page that is used as a point of reference), a variable for the width and height of the browser window, a record for the path a visitor has taken to arrive at the current page of the website, a measure for the time that has lapsed since the last click in the session, or the page load in the case of the first click on any page, a sequence number for the click within the session, a sequence number for the click on that page (first click=0, second click=1), a URI of the page on which the click occurred (i.e. page ID), and information about the element that was clicked (e.g. the URI of the element, the link reference, the element type).

The unique ID of the selected element/target is generated at the remote terminal 6. It is constructed by applying a hashing algorithm to a portion of the HTML for the target.

In a preferred embodiment, the hashing algorithm is applied to the first 200 characters of the HTML for the target, the attributes of which have been re-ordered. The hashing result becomes the unique identifier and forms part of the click data.

The reason for only taking the first 200 characters is to prevent performance problems when determining the click data to be sent to the central server. For example, if the target is a <select> drop down box with hundreds of options, it could take several minutes to perform the hashing algorithm on the complete HTML element for the target.

The attributes are re-ordered because different browsers store the attributes within the HTML element (in memory) in different orders. Therefore, in order to produce the same unique ID for each target, regardless of the browser, the attributes must be ordered in the same way (for example alphabetically) each time the unique ID is to be determined.

Figure 10 shows a flowchart 100 of the method steps required in order to carry out the data recording step, step 28, of Figure 2.

At the remote terminal 6 a visitor opens a web browser, such as Internet Explorer, and navigates to a website of interest. The remote terminal 6 sends, at step 102, a request for the desired web page. The website server 4 receives the request, and returns, at step 104, the related HTML file 14, which includes the pointer 16. The remote terminal 6 loads the web page and executes, at step 106, the pointer code 16, which causes the JSP request to be sent, at step 108, to the central server 2. The central server 2 retrieves and returns the JSP, at step 110, and the remote terminal 6 executes, at step 112, the JSP. As a result of executing the JSP, the remote terminal 6 sends, at step 114, the initial GET request, which includes the session data, to the central server 2. The central server 2 assigns, at step 116, a new session ID, stores, at step 118, the session data in the CS database 10 in relation to the session ID, and returns, at step 120, the session ID to the remote terminal 6, which stores, at step 122, the session ID in a temporary file/cookie.

Whether the visitor has clicked on the web page is determined, at step 124. If the answer is no, the remote terminal 6 waits for a click to be made, and if the answer is yes, the remote terminal 6 sends, at step 126, a GET request including the click data to the central server 2. The central server 2 receives the GET request and stores, at step 128, the click data in the CS database 10 in relation to the session ID. In this case, the

GET request is not used to pull information from the central server 2 but rather to provide raw result data to the central server 2.

The reason for recording session and click data is to enable the reporting module 38 to generate reports regarding visitor interaction with the website and web pages. The table below summaries the use of each of the data types in the generation of the reports.

Session Data	
Date/Time	To enable the number of 'visits' per day, hour, etc to be calculated.
Browser name and version	To allow reports to be filtered by browser (e.g. only plot clicks from IE 7 visitors)
Screen Resolution	To allow reports to be filtered by screen resolution (e.g. only plot clicks from visitors with 600x480 screen resolutions)
IP Address	To allow for filtering data by IP address (e.g. to exclude data from visitors within a certain organisation), or to filter data by 'country' of visitor (using an IP to Country conversion table)
Operating System	To allow the reports to be filtered by operating system (e.g. only plot clicks from visitors with Apple OSX)
Font Size	To allow the reports to be filtered by font size (e.g. only plot clicks from visitors who have a medium font-size setting in their browser)
Click Data	
Date/Time	To allow the reports to be filtered by date (e.g. only plot clicks from Saturday and Sunday)
Co-ordinates of click	To allow the click co-ordinates to be calculated relative to top of screen and/or key element and/or target element.
Co-ordinates of target (element clicked on)	To allow the click co-ordinates to be recalculated based on the targets co-ordinates.
Unique ID of the target	To enable the click co-ordinates to be calculated relative to the target when the web page is dynamic. This information also enables statistics relating to the target to be formulated.
Number of horizontal and vertical pixels	To enable reports to be filtered in relation to visitor scrolling patters (e.g. "show only clicks from people who

scrolled	scrolled vertically").
Co-ordinates of the "Key element"	To allow the relative position of a click to be recorded when a website is centred.
Width and Height of browser window	One reason this information is recorded is to enabling filtering of results to show click-through rates when visitors view web pages in full, or when their view of the web page is reduced due to the browser size being reduced.
URL/Path/Query String of the page	A click is recorded against the web page on which it is made; this is required to allows users to view reports for each web page.
Time since the last click on the page	To enable the reports to be filtered in relation to click timing (e.g. "only show clicks from people who clicked within 3 seconds").
Sequence number of the click on the page	To enable 'noise' removal (i.e. removal of clicks from visitors who click randomly all over the page, be it because they wish to upset the statistics or otherwise). Filtering using clicks on each page enables the removal of click data from any visitors who have a large sequence number for the current web page.
Information about the element that was clicked on	In order to generate reports relating to visitors path through the website.

As shown in Figure 11, the reporting module 38 comprises a result filter 130 for determining which results are required for the report, a raw data locator 132 for retrieving the required raw interaction data from the CS database 10, a report generator 134 for generating the reports, and an export module 136 for enabling the reports to be exported in a format suitable for saving.

It is preferable to provide reports in a visual format which relates to the web page on which the click occurred. Typically, the reports are displayed to the user in an image which overlays the web page in question. The types of reports which are suitable includes heat maps that graphically show the volume and location of clicks on the web page, click maps that graphically show the actual position of clicks and whether they were on or off target, hover maps which graphically provide usage data/statistics for individual elements on the web page, click stream reports that show a visitors path

through the website and page statistic summaries which provide summary statistics for the page being reported.

Filtering of the interaction data results enables the user to be presented with appropriate, accurate and meaningful information. Any of the required reports can be based on a filtered set of interaction data results. The clicks that are to be included in a report are filtered depending on the user choices relating to: a start date and time, end date and time, day(s) of the week, time periods (morning, afternoon, or evening) browser name and version, time taken to click, default font size, the page the visitor navigated from, the page the visitor navigated to, and screen resolution. It is to be appreciated that this list of filter parameters is not exhaustive, and other filter parameters may be desirable.

A primary reason for filtering the results is related to the number of variations between the different Internet browsers and the browser being used by the user (website administrator). Different browsers render web pages slightly differently. In addition, the same browsers may be configured differently in terms of screen resolution and colour depth, and default text size. As a result, the positions of clicks that are made via browsers different to that through which the reports are being viewed may be inaccurate to the actual (absolute) position of the click made. Filtering the results, so that the user only sees those clicks which were made with browsers which match their own (i.e. the browser name and version match and are configured with the same resolution and default font size), removes the any potential inaccuracies from the reports.

Temporal filtering is also useful, and permits analysis of information stored in relation to when the clicks occurred (namely the time between clicks) to help understand a visitor's interaction with a given web page over time. This may be particularly helpful when looking for where (i.e. the element clicked on) a user first clicks when presented with a web page, how long it takes for them to click a certain button for example and in general how they interact with the rest of the options on the web page over time. The results can be colour-coded for ease of understanding in a similar manner to the heat map (described below).

In addition, the content of web pages vary with time and the number of visitors to a page may also vary with time. To provide added insight it is possible to filter the results to show the clicks made between particular dates, or at particular times on particular days of the week.

In another embodiment, filtering of the results enables the system to account for the variations between the browser through which the report is being viewed and the browsers through which the clicks were made. By accounting for the variations, the analysis tool amends the position of the clicks in the reporting browser to more accurately reflect where the click was made in the visitor's browser.

One way in which the variations can be accounted for is through the use of scaling. As shown in Figure 12, the report generator 134 of the central server comprises a scaling module 138 which is responsible for amending the actual X/Y co-ordinates of a click to an equivalent (relative) X/Y co-ordinate of where the click would have been had the visitor been using the same browser as that of the user viewing the report.

The scaling module 138 first filters the interaction data results into groups depending on resolution and default font size, and then treats/scales the X/Y co-ordinate positions in each group in the same manner, thereby accounting for the variations.

A person skilled in the art will appreciate that the scaling module must determine which browser the user is using to view the report and also the configuration (i.e. the resolution and default font size) of the browser in question.

Figures 12 (a) and (b) show an example of how a click's position may be scaled. Figure 12 (a) shows a screen layout at a first resolution 140, and Figure 12 (b) shows a screen layout at a second resolution 142, higher than the first resolution. In this example, the second resolution is twice the size of the first resolution, i.e. the number of pixels (squares) in Figure 12 (b) is twice that of Figure 12 (a).

A click at location A in Figure 12 (a) is on target for a selectable link 144. However, if that click was viewed by a user using a resolution like that of Figure 12 (b), without taking account of the change in resolution, it would appear that the click missed the target at location B. However, if the X/Y co-ordinates of the actual click (made using a browser with the resolution of Figure 12 (a)) are scaled up by a factor of two then

the click appears to have been made at the relative 'correct' location C. Of course, scaling is also possible where the website is centred. The clicks are plotted with reference to the key element in that case, although, the scaling is carried out in the same manner.

A person skilled in the art will appreciate that the above scaling technique can be used to make adjustments for clicks made in browsers using a different default font than that of the browser being used to display the results to the user.

Filtering using the abovementioned filter parameters is performed by querying the relational tables 50 stored in the CS database 10 with a SELECT query which is composed in relation to the results required.

An alternative method for performing the filtering requires the use of a multi-dimensional cube. Which has been described previously.

The result filter is arranged to compose a suitable SELECT query on the basis of the results required for reporting, and the raw data locator 132 is arranged to execute the SELECT queries in order to retrieve the raw data from the CS database 10.

Shown below are first and second SELECT queries that return the same data. The first query is an SQL query that gets the data from the tables. The second is a Multidimensional Expressions (MDX) query that gets the data from the multi-dimensional cube.

First SELECT query

```
SELECT count(Click.ClickId), Click.ClickX, Click.ClickY, Browser.Name,
Platform.Name, Page.Path, Site.Name
FROM ClickDensity.dbo.Click, Clickdensity.dbo.Session,
ClickDensity.dbo.Browser,
        ClickDensity.dbo.Platform, ClickDensity.dbo.Page,
ClickDensity.dbo.Site
WHERE Click.SessionId = Session.SessionId AND
    Browser.BrowserId = Session.BrowserID AND
    Platform.PlatformId = Session.PlatformId AND
```

```
    Page.PageId = Click.PageId AND
    Site.SiteId = Session.SiteId
GROUP BY clickx, clicky, browserid, platformid, pageid, siteid
```

Second SELECT query

```
SELECT { [Measures].[Click Count] } ON COLUMNS,
  { [Click].[Click X].Children, [Click].[Click Y].Children } ON COLUMNS,
  { [Browser].[Browser Name].Children } ON PAGES,
  { [Platform].[Platform Name].Children } ON SECTIONS,
  { [Page].[Page Path].Children } ON PAGES,
  { [Site].[Site Name].Children } ON SITES
FROM ClickDensity
```

Databases configured for OLAP allow for complex analytical and ad-hoc queries with a rapid execution time. The retrieved raw interaction data is passed to the report generator 134 which converts the data into the required graphical/visual representation.

An example heat map 150 is shown in Figure 13. As shown, heat maps 150 offer a visual representation of the locations within a webpage, which are clicked most often. For each click on the webpage, a coloured marker 152, 160 appears in the heat map report image giving it a three-dimensional information presentation ability. A plurality of clicks at one co-ordinate location results in a graduated change in colour. Thereby, showing not only the locations which are clicked most often but an indication as to how often one location is clicked in relation to other areas. A scale 154 is provided on the website showing the graduated range of colours, for example blue 156 through to red 158. In this example, a location, which is clicked once, shows a blue marker 152, and a location which has been clicked a large number of times shows a red marker 160. An example scale 154 is shown in Figure 14. This scale is shown as a scale of white 162 through darker shades of grey to black 164, where the white end 162 of the scale represents blue and the black end 164 of the scale represents red.

In one embodiment, the heat map 150 is generated by plotting the clicks on a web page recorded in the database on a bitmap image. In another embodiment, the heat map is generated independently of the image of the web page and is simply overlaid on top of the actual web page during the reporting process.

Each click is represented with a circle of colour showing the greatest "heat" in the middle at the pixel where the click was made. The colours used are adjusted so that the "hottest" part of the resulting image is the point where there were the most clicks on the web page. A single click is shown in Figure 13 at point 152.

The heat maps 150 are created using a bell curve. The pixel where the click actually occurred is awarded 5 points, then the pixels immediately either side get 3 points, then the ones next to that get 1, as shown below (each number represents one pixel).

```
      1
    3 3 3
  1 3 5 3 1
    3 3 3
      1
```

Pixels which have been clicked on several times accumulate in value. For example, if the pixel represented above by the number 5 has been clicked on twice, its value increases to 10, and the remaining values surrounding the 5 are also doubled. The value of each pixel is then represented by a colour. The colour red is used for pixels that have the maximum value and the colour blue is used for pixels that have the minimum value. All other values, between the maximum and minimum are given different colours of the spectrum, in accordance with the scale 154. This provides a smooth gradient for colours within the heat maps 150. Pixels that have no value are not assigned a colour and are remain transparent, such that when the heat map is overlaid on top of the website, only the colour of the website at that pixel is shown.

Click maps 170, as shown in Figure 15, are very similar to heat maps 150 but are a simplified version showing a small marker 172, 174 for each click on the page. The graphical representation of a click (a cross) is drawn in either green 174 or red 172. Green clicks 174 are on 'clickable' elements 176, 184 (hyperlinks, hyperlinked images, form fields, buttons, elements with onclick handlers, etc), and red clicks 172 are on non-clickable elements. Click maps 170 do not visualise the concentration of clicks as accurately as heat maps. However, they can be used to identify 'close misses' 178, i.e. show where visitors are clicking indicating that they think there is a link at that location, and enables analysis of the exact location of clicks.

As shown in Figure 16, a hover map 180 shows a pop-up box 182 which contains statistics about an element, in this case, a 'try the demo' button 184. When the user, having selected hover map 180, positions the mouse over a selectable element 184, the pop-up box 182 appears. The report generator 134 prepares the necessary reporting information (including relating the information about each element to the unique ID for that element) and transmits it to the user's remote terminal. Thereafter, a mouse-over event handler ensures that the relevant pop-up box 182 is displayed in response to a mouse-over for that element, the unique ID of which is determined, as described above.

When a user wishes to view a report, they are shown in the first instance the home page of their website. Initially, an option to navigate through the website is selected 190. Other options for viewing a heat map 192, click map 194 or hover map 196 are presented to the user. Upon selection of one of these options a request is sent to the central server 2, and the communications module 32 will pass the request to the reporting module 38 to process the request as described above.

If a user wishes to view a report for another page of the website, they select the navigate button 190 when in Figures 13, 15, and 16, which turns off the reporting feature of the browser and allows the user to navigate through the website to find a further page of interest.

The user is also presented with filter options 200 as detailed above, and the user can select to update the image as necessary depending on the filter parameters selected. The user also has the option of changing the transparency of the heat or click maps. This is achieved by adjusting the contrast, between the report image overlaid on top of the web page, and the web page itself, using standard image processing techniques.

A further filter option, a follow target option 210, provides a visual representation of the location (X/Y co-ordinates) of each click in relation to the X/Y co-ordinated of the selected element/target. Without follow targets selected, the clicks are displayed at the exact (absolute) X/Y co-ordinates recorded, relative to either the top corner of the page or the key element, if specified. However, as detailed above, when a visitor clicks on a clickable target, the X/Y co-ordinates of the clickable element are also recorded in addition to the unique ID of the element. As a result, when follow targets

is selected, the clicks are plotted relative to the selected element's co-ordinates. This enables the clicks to be repositioned on dynamic pages to line up with the elements/targets they are associated with.

As shown in Figure 17, a click 220 is made 55 pixels from the top of the page and 85 pixels from the left. Due to the fact the click 220 is on a clickable element 222, the X/Y co-ordinates of the top left pixel of the clickable element can be determined because a unique ID of the selected clickable element is sent with the click data to the central server. As a result, it is possible to plot the click in relation to the clickable element (i.e. in this case the click is located 17 pixels from the top of the element and 30 pixels from the left of the element). At the time of rendering the result, the report generator 134 will ascertain from the web page (being displayed underneath the report) current X/Y co-ordinates for the selected element 222. As such, it is possible to display the clicks 220 with reference to the selected element 222 regardless of how the element changes 222 its position on the web page over time.

Since the filter for follow targets requires a record of the selected element's X/Y co-ordinates to be made, or a record of the unique ID of the selected element, the follow targets filter being selected has the result that any clicks on non-clickable elements will be filtered out, i.e. X/Y co-ordinates of an element can only be recorded when the click is on an element. Figure 18 shows a screen shot 230 of the results when follow targets 210 is selected, and no "off-target" clicks are shown.

As shown in Figure 18, it is possible to select to view the results as an export map 234 (i.e. normal web page format) or a JPEG format 236. This facilitates local storing of result maps at a remote terminal of the user since the resulting JPEG may be saved. The export module 136 is arranged to convert the report image in export map mode into a JPEG file, which can subsequently be displayed to a user using a suitable program such as Microsoft Photo Editor or Paint.

In another embodiment, the present invention can be used to facilitate A/B testing, A/B tests allow website owners to test the relative merits of two different versions of web page content (e.g. an advert, link, content) which is to be provided in one location with real visitors. By examining the 'success' (number of clicks, sales, page view times) of each version, they can then optimise their page by choosing the most successful version.

Most existing A/B testing suites allow you to define both versions (i.e. content A and B) on a remote A/B testing server (which delivers the tests). A JavaScript pointer is included in the HTML file for the web page, and when the web page is downloaded from the website server, the JavaScript pointer makes a request to the A/B testing server to retrieve and returns either content A or B, for inclusion in the web page. Figure 19 (prior art) shows a web page 240 which includes a JavaScript pointer 242 to the A/B testing server (www.ab.com) where the content A and B is stored.

The main disadvantage with this prior art method is that browsers that have JavaScript disabled (because of the users privacy concerns), or are unable to use/execute JavaScript (e.g. basic mobile browsers, accessible text-to-speech browsers, other non-PC devices), the JavaScript will not get processed, and hence no content will be delivered. This results in a blank portion of the screen being displayed

In terms of the real-time recording of the interaction data, any browsers which are not able to execute JavaScript will not facilitate recording of any interaction data, but more importantly the visitor to the web page is unable to see either content A or B and the web page does not deliver its intended message. This is a disadvantage that deters website administrators from making use of A/B testing when evaluating the content of their website.

However, the A/B testing of the present embodiment overcomes this problem by ensuring that content A (default content) is defined in the HTML file for the web page and that 50% of the time that content is replaced by content B (alternative content), from an A/B testing server, through the use of a portion of JavaScript code. In this way, if the browser is not capable of executing JavaScript code, then the visitor is always presented with the default content, and therefore this avoids the situation with known methods of A/B testing where no content is delivered.

In the present embodiment, the website administrator defines content for version B on the A/B testing server, and specifies an ID for an existing HTML element on the page to replace (i.e. the *existing* element is the default content A). JavaScript code, including, a pointer to the content for version B, is included within the downloaded HTML file. After download, the JavaScript code executes such that a request is made to the A/B testing server to replace content A within the HTML file. The A/B testing

server alternates delivering or not delivering the content B to ensure that 50% of the web page downloads result in content A being replaced with content B. Advantageously, when JavaScript execution is not available, content A will be displayed by default, since there is no requirement for it to be dynamically delivered.

Another report which the present embodiment is able to produce includes a click stream report 250. A click stream report is a visualisation of one or more website visitors' path through the website. The click data relating to the URI a visitor navigated from to arrive and the web page where a current click occurred (i.e. the click resulting in the click data being sent), and the URI of the element the visitor clicked on (i.e. where the visitor navigated to) is recorded for each click and can be collated to provide meaningful reports. An example click stream report 250 is shown in Figure 20.

In order to generate a click stream report 250, the report generator collates the information regarding how visitors navigate through the website. Alternatively, the click stream report may be based on a single visitor's path through the website.

It is to be appreciated that click stream reports may be based on any collection of results. For example, the report generator may, for each session ID, plot the path through the website (perhaps only the first four clicks). Having done that for each session ID, the report generator can order the different paths in terms of their popularity, and provide a click stream report similar to that in Figure 20 for the most popular paths through the website. The click stream report may also include statistical details, for example, regarding the percentage of people who took the most popular path. The click stream report may visualise the popularity of different paths through the website. As shown in Figure 20, a wide arrow 252 may represent the number of visitors choosing the path from a first web page 254 to a second webpage 256, and a thin arrow 258 may represent a less popular route.

Advantageously, the system provides a user with the ability to view every single click that occurs on their website. Known web analytics software have not previously offered this functionality. Furthermore, because the reports are presented graphically, the user does not require technical knowledge to be able to analyse the clicks. In addition, analysis of every visitor is possible, down to their individual route through

the site and the actual locations that they clicked. It is, of course, highly advantageous for website administrators/users to identify the elements on the website that are of interest to visitors and those that are not.

Filtering of the reported data is advantageous because it permits the user to view information which is accurate and meaningful. Furthermore, the ability to view real-time reports enables users to react extremely quickly if the desired results are not being achieved.

Whilst the above mainly concerns an embodiment of the invention in which the JavaScript code is held on a central web server, it is to be appreciated that it is possible to vary the method of installation. For example, an embodiment of the present invention may be installed on the user's web servers, whereby the code and associated data are held on the same server as the web site itself. In addition, integration of the system with any graphical user interface, such as a stand-alone software package, is possible and this would not necessarily need to be web based.

Having described particular preferred embodiments of the present invention, it is to be appreciated that the embodiments in question are exemplary only, and that variations and modifications, such as those that will occur to those possessed of the appropriate knowledge and skills, may be made without departure from the spirit and scope of the invention as set forth in the appended claims. For example, it will be readily apparent to those skilled in the art the code responsible for initialising and executing the recordal if the interaction data need not be JavaScript, as other types of script may be suitable. In addition, interaction data need not relate solely to mouse clicks, and could include data from touch screens, wherein interaction with the touch screen may be using a pointing tool or a user's finger. Furthermore, the 'click' need not solely relate to a single pixel. Especially, in terms of touch screens a range or group of consecutive pixels may be used.

Claims:

1. An interaction recorder for recording interaction data relating to multiple user interaction events with a GUI presented at a user terminal, the recorder comprising:

script transmitting means for transmitting an executable script to the user terminal in response to a first request;

interaction data receiving means for receiving interaction data generated by execution of the script on the user terminal, the interaction data describing each user interaction event occurring with the GUI and including a set of co-ordinates specifying the location within the GUI of each user interaction event and an identifier identifying the GUI to which the interaction data relates; and

a data store arranged to store the received interaction data;

wherein interaction data receiving means and the data store are arranged to receive and store a sequence of user interaction data relating to the GUI in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI is possible by viewing of the currently stored interaction data.

2. An interaction recorder as claimed in Claim 1, wherein the data store is arranged to store the interaction data in a plurality of session logs wherein each session log comprises data relating to a single user's interaction with a GUI over a period of time.

3. An interaction recorder as claimed in Claim 2, wherein the interaction data includes session data relating to session attributes describing the environment in which the interaction events are recorded.

4. An interaction recorder as claimed in Claim 3, wherein the session data comprises one or more attributes selected from the group comprising: a session start date, a session start time, a browser name, a browser version, a screen resolution, a screen colour depth, a default font size, an IP address, and an operating system of the remote terminal.

5. An interaction recorder as claimed in any of Claims 2 to 4, wherein the interaction data includes event data relating to the single user's interaction with the GUI.

6. An interaction recorder as claimed in Claim 5, wherein the event data comprises one or more attributes selected from the group comprising: positional co-ordinates, an event date, an event time, a target positional co-ordinate, a target ID, a horizontal scroll variable, a vertical scroll variable, a key element positional co-ordinate, a browser window size, a previous GUI page ID, an elapsed time variable, a session sequence number, a GUI sequence number, a current GUI page ID, and a URI of a selected element.

7. An interaction recorder as claimed in any preceding claim, further comprising a session ID generator arranged to generate and transmit a session ID to the user terminal in response to a script generated request.

8. An interaction recorder as claimed in Claim 7, wherein the data store is arranged to store the interaction data using the session ID, the interaction data being stored in a relational table.

9. An interaction recorder as claimed in any preceding claim, wherein the GUI comprises a plurality of interconnected GUI pages, and the interaction data receiving means is arranged to receive interaction data describing user interaction events occurring within the GUI pages.

10. An interaction recorder as claimed in any preceding claim, wherein the interaction recorder is arranged to record interaction data relating to multiple user interaction events for a plurality of GUIs, and further comprises a site ID generator arranged to generate a unique site ID for each of the plurality of GUIs.

11. An interaction recorder as claimed in Claim 10 as dependent upon Claim 2, wherein the data store is arranged to link together stored session logs relating to the unique site ID.

12. An interaction recorder as claimed in any preceding claim, further comprises a pointer creation means arranged to create a pointer script, identifying the executable script stored at the interaction recorder wherein execution of the pointer script at the user terminal generates the first request.

13. An interaction recorder as claimed in Claim 12, wherein the pointer creation means is arranged to generate the pointer script by use of a template and the unique site ID.

14. An interaction recorder as claimed in Claim 13, wherein the pointer creation means is arranged to generate the pointer script by additional use of a key element ID.

15. An interaction recorder as claimed in any preceding claim, wherein the GUI comprises a website.

16. A method for recording interaction data relating to multiple user interaction events with a GUI presented at a user terminal, the method comprising:

transmitting an executable script to the user terminal in response to receiving a first request;

receiving interaction data generated by execution of the script on the user terminal, the interaction data describing each user interaction event occurring with the GUI and including a set of co-ordinates specifying the location within the GUI of each user interaction event and an identifier identifying the GUI to which the interaction data relates; and

storing the received interaction data;

wherein the receiving and storing steps include receiving and storing a sequence of user interaction data relating to the GUI in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI is possible by viewing of the currently stored interaction data.

17. A graphical user interface (GUI) interaction analyser for analysing the results of multiple user-interaction events with a GUI page presented at a user terminal, the analyser comprising:

retrieval means for retrieving interaction data stored in a data store; the interaction data being stored in plurality of interaction records, each of which includes co-ordinates specifying the location within the GUI page of a user interaction event and an identifier identifying the GUI page to which the interaction data record relates;

a graphical processor for processing the interaction data to generate a three-dimensional representation of the interaction data, two dimensions being determined by the location of the interaction event on the identified GUI page and the third dimension being determined by a further parameter relating to the interaction event occurring at a location of the identified GUI page;

and map generation means for presenting a map showing the three-dimensional interaction data superimposed on a representation of the identified GUI page.

18. An analyser as claimed in Claim 17, wherein the retrieval means further comprises a filter arranged to compose a query request to retrieve the interaction data from the data store.

19. An analyser as claimed in Claim 18, wherein the filter is arranged to receive from the user terminal at least one filter parameter selected from the group comprising: a start date, an end date, a start time, and end time, a screen resolution, a colour depth, a browser window size, a country, an organisation, a browser name, a browser version, and an operating system, and wherein the query request to effect interaction data retrieval includes at least one filter parameter.

20. An analyser as claimed in Claim 18 or 19, wherein the graphical processor is arranged to receive the results of the request query and to effect generation of the three-dimensional representation of the interaction data.

21. An analyser as claimed in any of Claims 17 to 20, wherein the interaction data set includes one or more attributes selected from the group comprising: a session ID, session start date, a session start time, a browser name, a browser version, a screen resolution, a screen colour depth, a default font size, an IP address, and an operating system of the remote terminal, absolute positional co-ordinates, an event date, an event time, a target positional co-ordinate, a target ID, a horizontal scroll variable, a

vertical scroll variable, a key element positional co-ordinate, a browser window size, a previous GUI page ID, an elapsed time variable, a session sequence number, a GUI sequence number, a current GUI page ID, and a URI of a selected element.

22. An analyser as claimed in any of Claims 17 to 21, wherein the graphical processor is arranged to determine the further parameter as the frequency of interaction events occurring at each location of the identified GUI page.

23. An analyser as claimed in any of Claims 17 to 21, wherein the graphical processor is arranged to determine the further parameter as the accuracy of interaction events occurring at an area location of the identified GUI.

24. An analyser as claimed in Claim 23, wherein the area location represents an interactive button of the GUI.

25. An analyser as claimed in Claim 23 or 24, wherein the graphical processor is arranged to determine the accuracy of the interaction event by determining if the location of the interaction event corresponds to the area location of the interactive button.

26. An analyser as claimed in Claim 25, wherein the graphical processor is arranged to represent graphically the accuracy of the interaction event by using at least two different colours.

27. An analyser as claimed in any one of Claims 17 to 26, wherein the graphical processor is arranged to transform absolute positional co-ordinates of the interaction event into relative positional co-ordinates with respect to reference co-ordinates, and wherein the map generation means is arranged to generate the graphical representation of the interaction event at the relative positional co-ordinates.

28. An analyser as claimed in Claim 27, wherein the reference co-ordinates are the co-ordinates of a key element.

29. An analyser as claimed in Claim 28, wherein the graphical processor is arranged use the key element ID to locate the key element, within the identified GUI, and determine the positional co-ordinates of the key element, such that the relative positional co-ordinates of the interaction event can be determined.

30. An analyser as claimed in Claim 27, wherein the reference co-ordinates are the positional co-ordinates of a target element being identified by a unique target ID associated with the interaction data set.

31. An analyser as claimed in Claim 30, wherein the graphical processor is arranged to use the unique target ID to locate the target element, within the identified GUI, and determine the positional co-ordinates of the target element, such that the relative positional co-ordinates of the interaction event can be determined.

32. An analyser as claimed in Claim 30 or 31, wherein the target ID is unique for each of a plurality of targets, and comprises a fingerprint of an associated target GUI element, wherein the graphical processor comprises a comparator for comparing the fingerprint to a newly generated fingerprint representation of the target element of the representation of the identified GUI.

33. An analyser as claimed in any one of Claims 17 to 32, wherein the graphical processor further comprises a scaling module arranged to reposition the positional co-ordinates of the location of the event as a function of a calculated difference between the screen resolution of the user terminal, whereupon the interaction event occurred, and the screen resolution of the user terminal whereupon the event is graphically represented.

34. An analyser as claimed in Claim 33, wherein the scaling module is further arranged to reposition the positional co-ordinates of the location of the event as a function of a calculated difference between the default font size of the user terminal, whereupon the interaction event occurred, and the default font size of the user terminal whereupon the event is graphically represented.

35. An analyser as claimed in any one of Claims 17 to 34, wherein the map generation means is arranged to display the graphical representation of the interaction event centred at the received positional co-ordinates or the relative positional co-ordinates.

36. An analyser as claimed any one of Claims 17 to 35, wherein the map generation means is arranged to generate a display of the third dimension as colour-coded graphical representation of the frequency of the interaction events occurring at each location.

37. An analyser as claimed in Claim 36, wherein the map generation means is arranged to select a colour for the representation from a scale of colours indicating a range of interaction event frequencies.

38. An analyser as claimed in Claim 37, wherein there are a plurality of GUI pages and each page has a different scale.

39. An analyser as claimed in Claim 37 or 38, wherein the scale is determined by the maximum and minimum frequencies of the three-dimensional interaction data for a given GUI page.

40. An analyser as claimed in any one of Claims 17 to 39, wherein the map generation means is arranged to generate a display of the third dimension of the interaction data constantly with the two-dimensional interactive event location data.

41. An analyser as claimed in any one of Claims 17 to 39, wherein the map generation means is arranged to generate a display of the third dimension of the interactive data upon user selection of an element of the GUI to which the third dimension data relates.

42. An analyser as claimed in Claim 41, wherein the map generation means is arranged to generate a display of the third dimension as a pop-up box which graphically represents statistical data regarding the frequency of the interaction events occurring at each location.

43. An analyser as claimed in Claim 42, wherein the graphical processor is arranged to receive preferences regarding the statistical data which is displayed within the pop-up box.

44. An analyser as claimed in any one of Claims 17 to 43, wherein the graphical processor is arranged to determine a value of the third dimension at the location of the interaction event and to assign a different value of the third dimension at a proximate location adjacent to the interaction event location.

45. An analyser as claimed in Claim 44, wherein the graphical processor is arranged to assign different values to a plurality of proximate locations surrounding the interaction event location, such that the effects of the third dimension are realised as distribution over the proximate locations and interaction event location.

46. An analyser as claimed in Claim 45, wherein the distribution forms a bell-shaped curve.

47. An analyser as claimed in any one of Claims 17 to 46, wherein the map generation means further comprises export means arranged to convert the graphical representation into a portable image file, wherein upon user selection the portable image file is transmitted from the GUI interaction analyser to a viewing terminal such that it can be viewed or stored at the viewing terminal.

48. An analyser as claimed in any one of Claims 17 to 47, wherein the GUI page comprises a page of a website.

49. A method of analysing the results of multiple user-interaction events with a GUI page presented at a user terminal, the method comprising:

retrieving interaction data stored in a data store; the interaction data being stored in plurality of interaction records, each of which includes co-ordinates specifying the location within the GUI page of a user interaction event and an identifier identifying the GUI page to which the interaction data record relates;

processing the interaction data to generate a three-dimensional representation of the interaction data, two dimensions being determined by the location of the interaction event on the identified GUI page and the third dimension being determined by a further parameter relating to the interaction event occurring at a location of the identified GUI page; and

generating a graphical map showing the three-dimensional interaction data superimposed on a representation of the identified GUI page.

50. A combination of a recorder as claimed in Claims 1 to 15, and an analyser as claimed in Claims 17 to 48.

51. A capture device for capturing interaction data relating to multiple user interaction events with a GUI page presented at a user terminal, the device comprising:

receiving means for receiving an executable script from an interaction recording device;

script execution means for executing the script concurrently with the operation of the GUI page at the user terminal, the script execution means generating interaction data describing each user interaction event occurring with the GUI page, the interaction data including a set of co-ordinates specifying the location within the GUI page of each user interaction event and a unique identifier identifying the GUI page to which the interaction data relates; and

transmission means for transmitting the user interaction data to the interaction recording device;

wherein script execution means and the transmission means are arranged to generate and transmit a sequence of user interaction data relating to the GUI page in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI page is made possible from consideration of the transmitted information.

52. A capture device as claimed in Claim 51, further comprising request transmitting means arranged to transmit a first request to the interaction recording device, the first request requesting the executable script from the interaction recording device.

53. A capture device as claimed in Claim 51 or 52, wherein the script execution means is arranged to generate a request for a session ID and the transmission means is arranged to transmit the request for the session ID to the interaction recording device.

54. A capture device as claimed in Claim 53, wherein the receiving means is further arranged to receive the requested session ID, and store the session ID in a temporary file.

55. A capture device as claimed in Claim 53 or 54, wherein the script execution means is arranged to determine session data describing the environment in which the interaction events are recorded.

56. A capture device as claimed in Claim 55, wherein the session data comprises one or more session attributes selected from the group comprising: a session start date, a session start time, a browser name, a browser version, a screen resolution, a screen colour depth, a default font size, an IP address, and an operating system of the remote terminal.

57. A capture device as claimed in any one of Claims 51 to 56, wherein the session ID request includes one or more of the determined session attributes.

58. A capture device as claimed in any one of Claims 51 to 57, wherein the script execution means is arranged to generate event data relating to a single user's interaction with the GUI page.

59. A capture device as claimed in any one of Claims 58, wherein the transmission means is arranged, for each interaction event, to transmit the received session ID together with the event data, specifying at least the positional co-ordinates of the interaction event, and the GUI page ID upon which the interaction event occurred.

60. A capture device as claimed in Claim 59, wherein the event data transmitted to the interaction recording device includes one or more further event attributes.

61. A capture device as claimed in Claim 60, wherein the one or more further event attributes is selected from the group comprising: an event date, an event time, a target positional co-ordinate, a unique target ID, a horizontal pixel scroll variable, a vertical pixel scroll variable, a key element positional co-ordinate, a browser width variable, a browser height variable, an identifier for a previous GUI page, an elapsed time value, a session sequence number, a GUI sequence number, and an URI of a selected element.

62. A capture device as claimed in any of Claims 50 to 61, further comprising means for identifying a selected target element, and means for generating a unique target ID from the description of the element.

63. A capture device as claimed in Claim 62, wherein the description comprises a mark-up language script describing the target element and the generating means is arranged to use a predetermined subset of the mark-up language script to generate the unique target ID.

64. A capture device as claimed in Claim 63, wherein the generating means is arranged to reorder the mark-up language script into a ranked ordered list.

65. A capture device as claimed in Claim 63 or 64, wherein the generating means comprising a hashing algorithm for creating a fingerprint from the mark-up language script, the fingerprint representing the unique target ID.

66. A capture device as claimed in any one of Claims 17 to 65, wherein the executable script is a JavaScript program.

67. A capture device as claimed in any one of Claims 17 to 66, wherein the GUI page comprises a page of a website.

68. A method of capturing interaction data relating to multiple user interaction events with a GUI page presented at a user terminal, the device comprising:

receiving an executable script from an interaction recording source;

executing the script concurrently with the operation of the GUI page at the user terminal, the executing step including generating interaction data describing each user interaction event occurring with the GUI page, the interaction data including a set of co-ordinates specifying the location within the GUI page of each user interaction event and a unique identifier identifying the GUI page to which the interaction data relates; and

transmitting the user interaction data to the interaction recording device;

wherein the transmitting step and the generating step comprise and generating and transmitting a sequence of user interaction data relating to the GUI page in real-time as the interaction events occur, such that real-time graphical reporting of the user interaction with the GUI page is made possible from consideration of the transmitted information.

69. A mark-up language representation of a page of a website for display in a browser at a user terminal, the page comprising at least one element, the element including a mark-up language code segment defining a first content component and an executable script defining a website for provision of a second content component, wherein if the browser is unable to execute the script then the mark-up language code segment is displayed in the element.

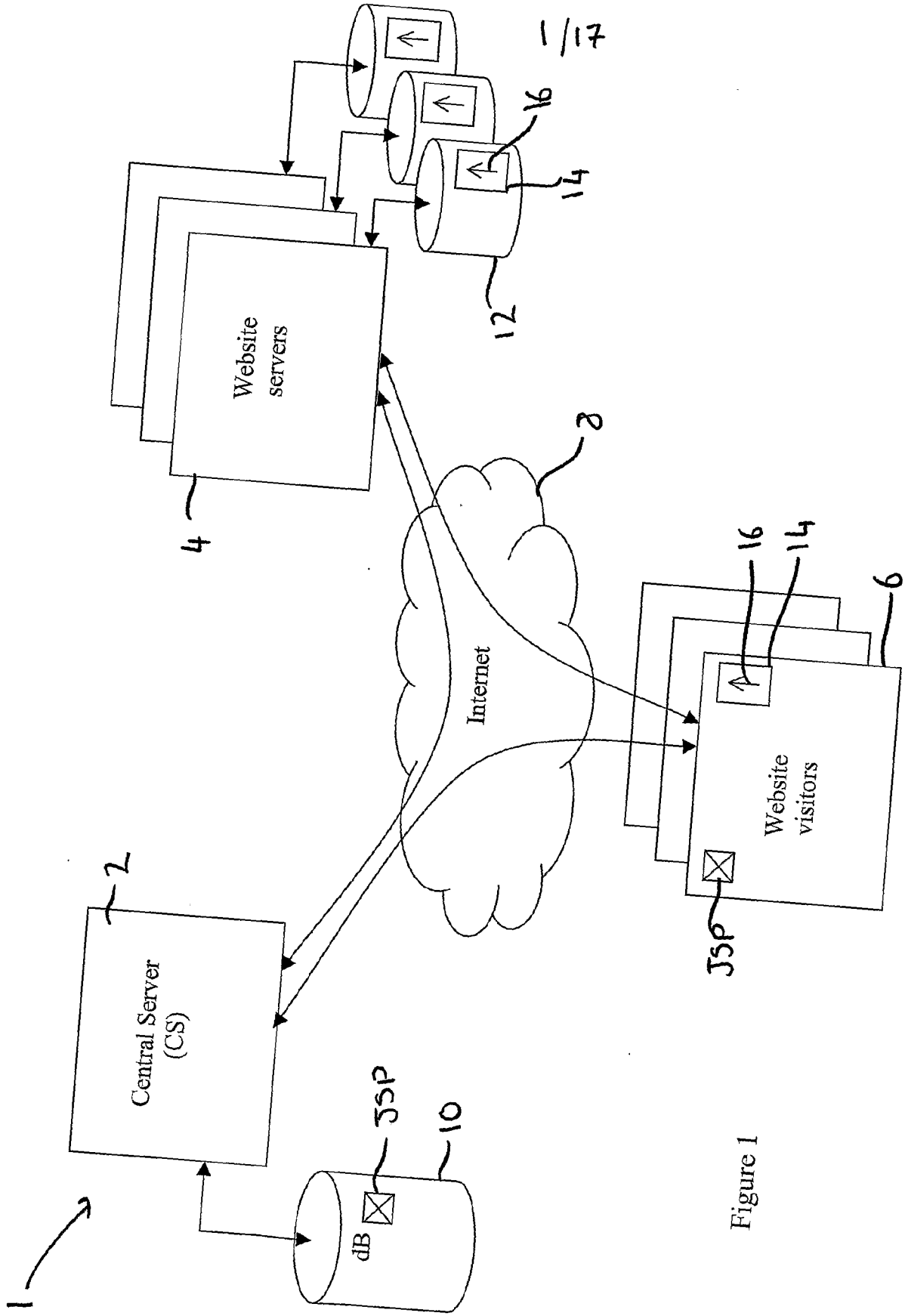

1/17
16
14
12
Website servers
4
8
Internet
16
14
6
Website visitors
JSP
2
Central Server (CS)
JSP
10
dB
1

Figure 1

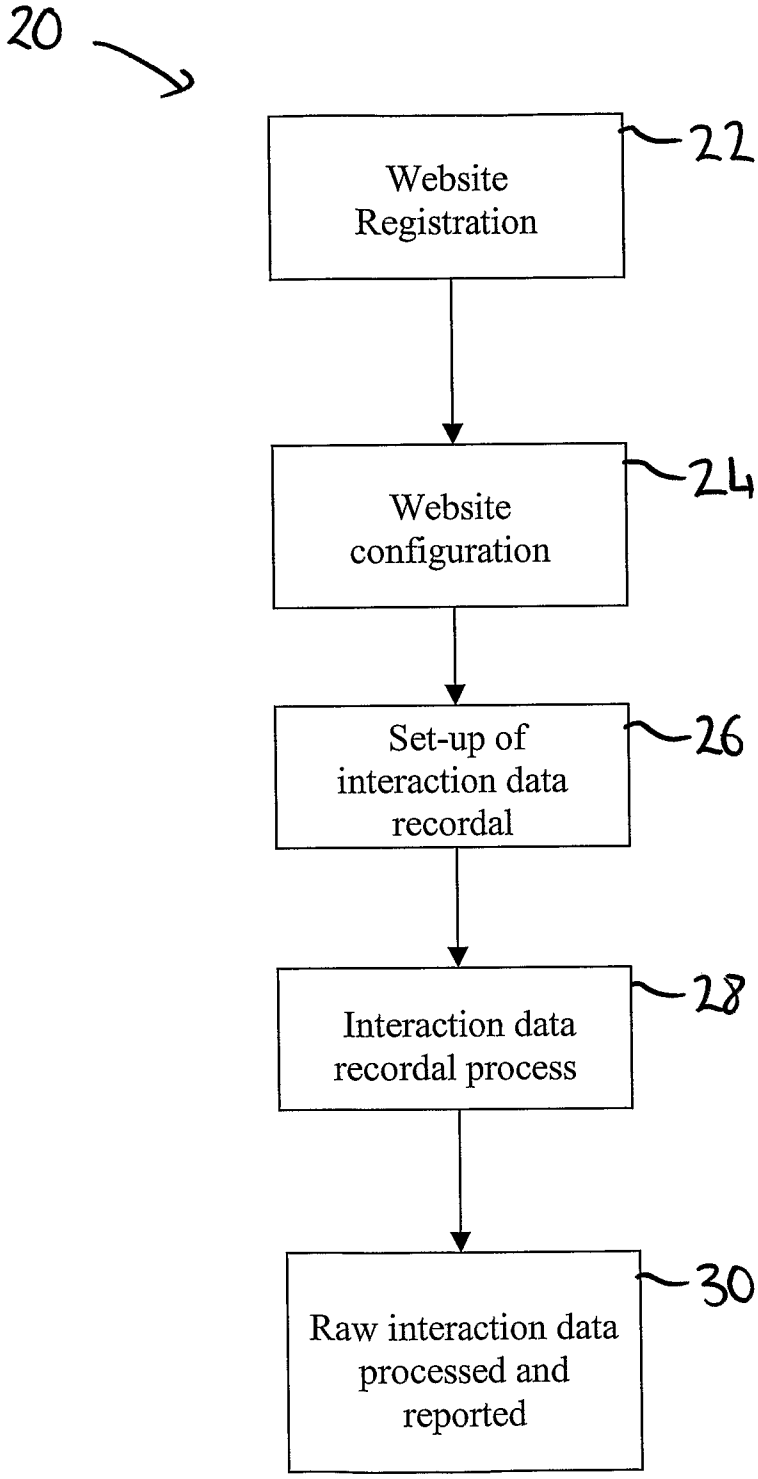

20
Website Registration
22
Website configuration
24
Set-up of interaction data recordal
26
Interaction data recordal process
28
Raw interaction data processed and reported
30

Figure 2

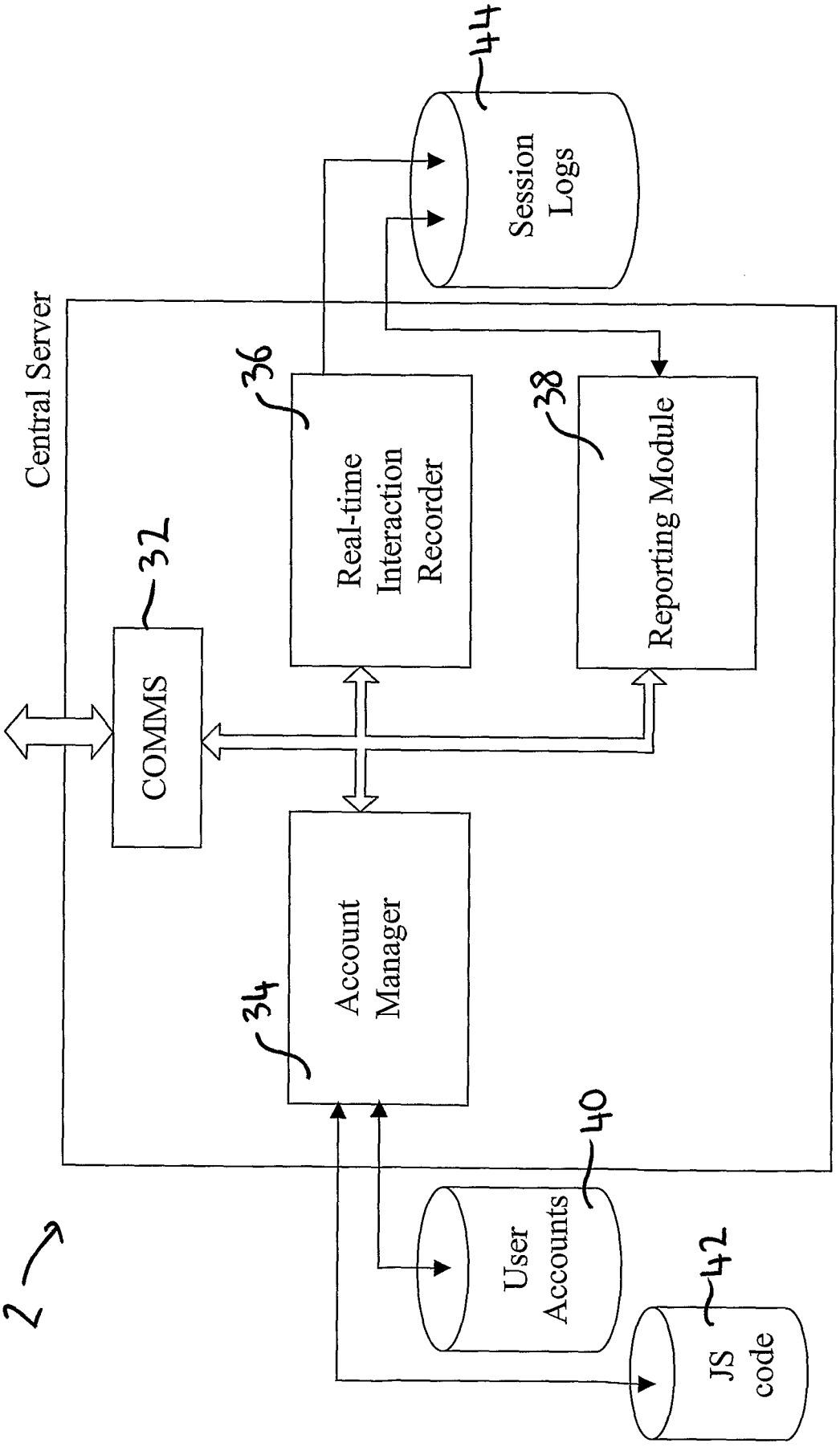

Central Server
COMMS
32
Real-time Interaction Recorder
36
Reporting Module
38
Account Manager
34
Session Logs
44
User Accounts
40
JS code
42
2

Figure 3

4/17

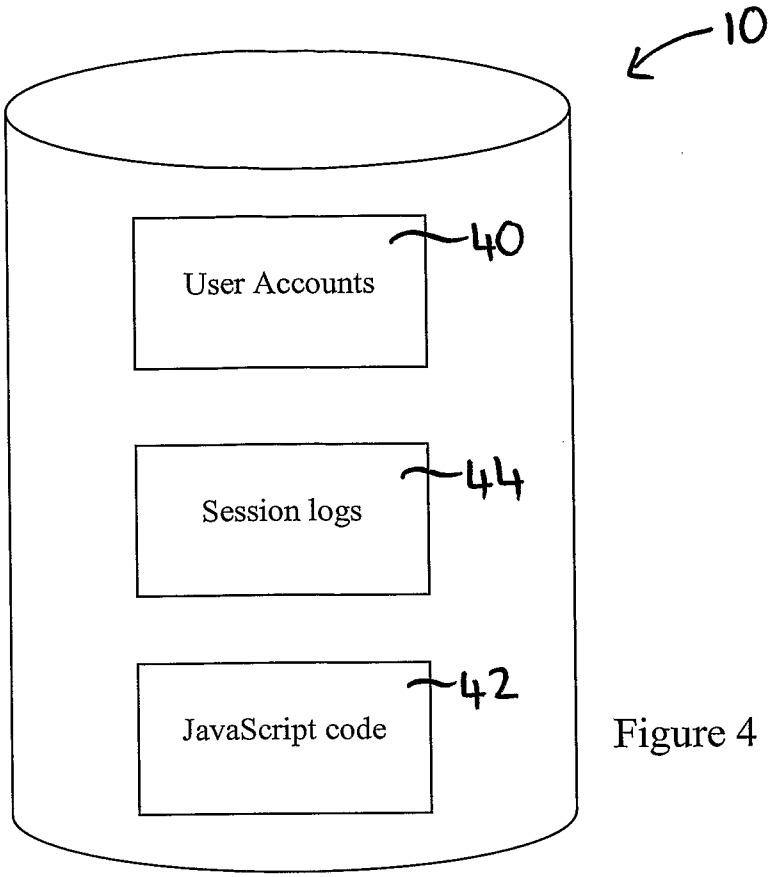


Figure 4

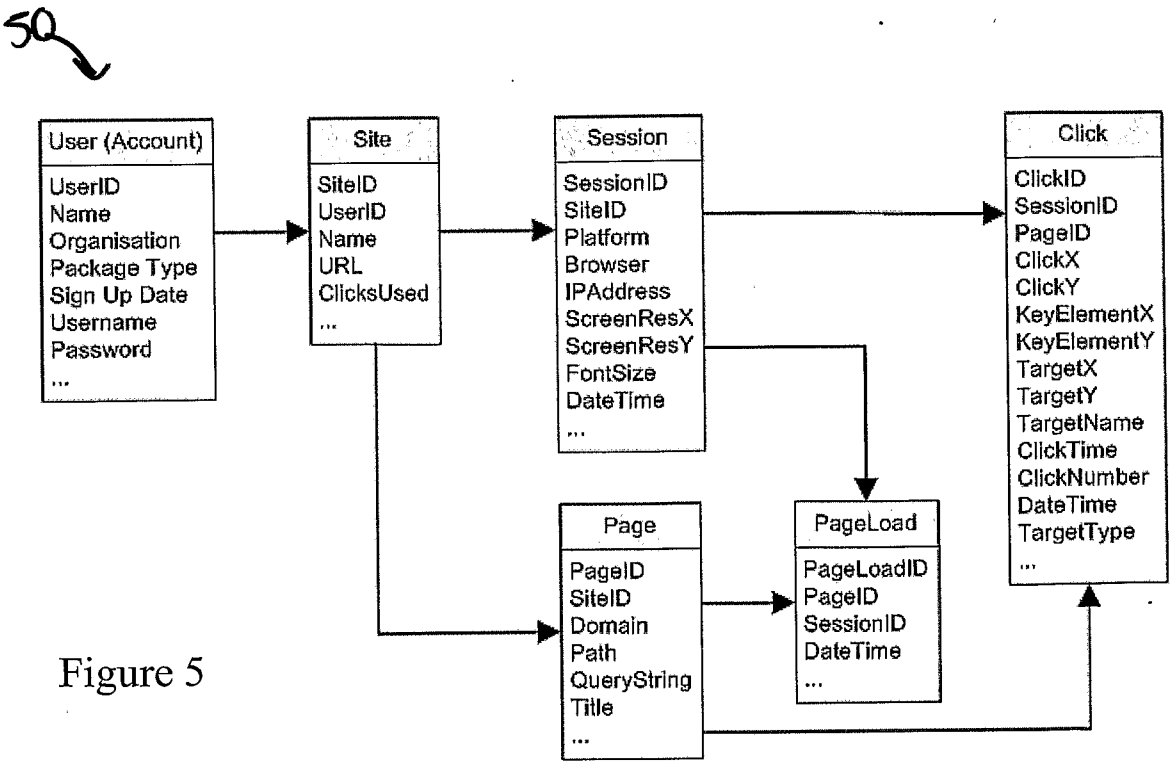


Figure 5

5/17

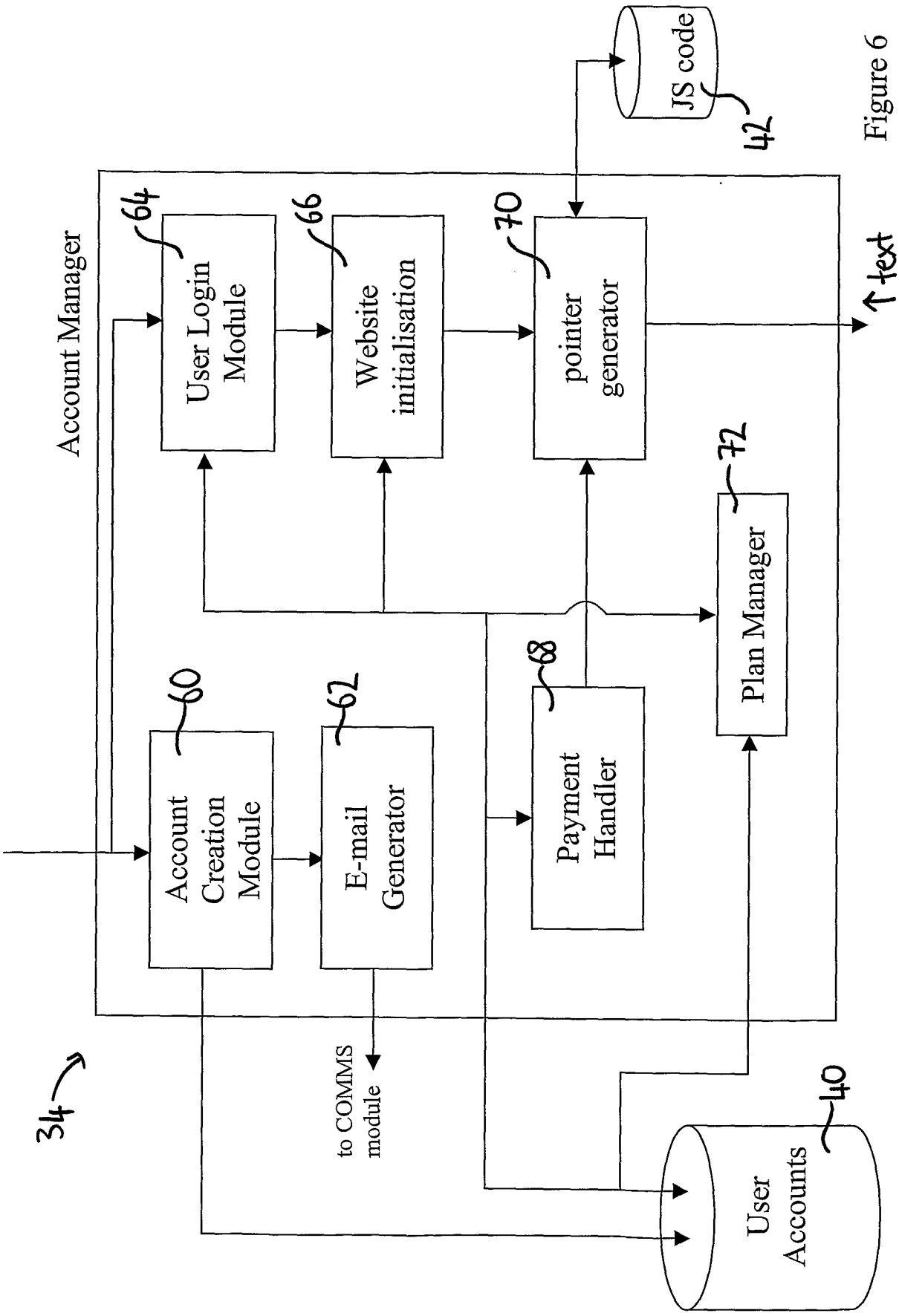

JS code
42
Account Manager
64
User Login Module
66
Website initialisation
70
pointer generator
text
72
Plan Manager
60
Account Creation Module
62
E-mail Generator
68
Payment Handler
to COMMS module
34
User Accounts
40

Figure 6

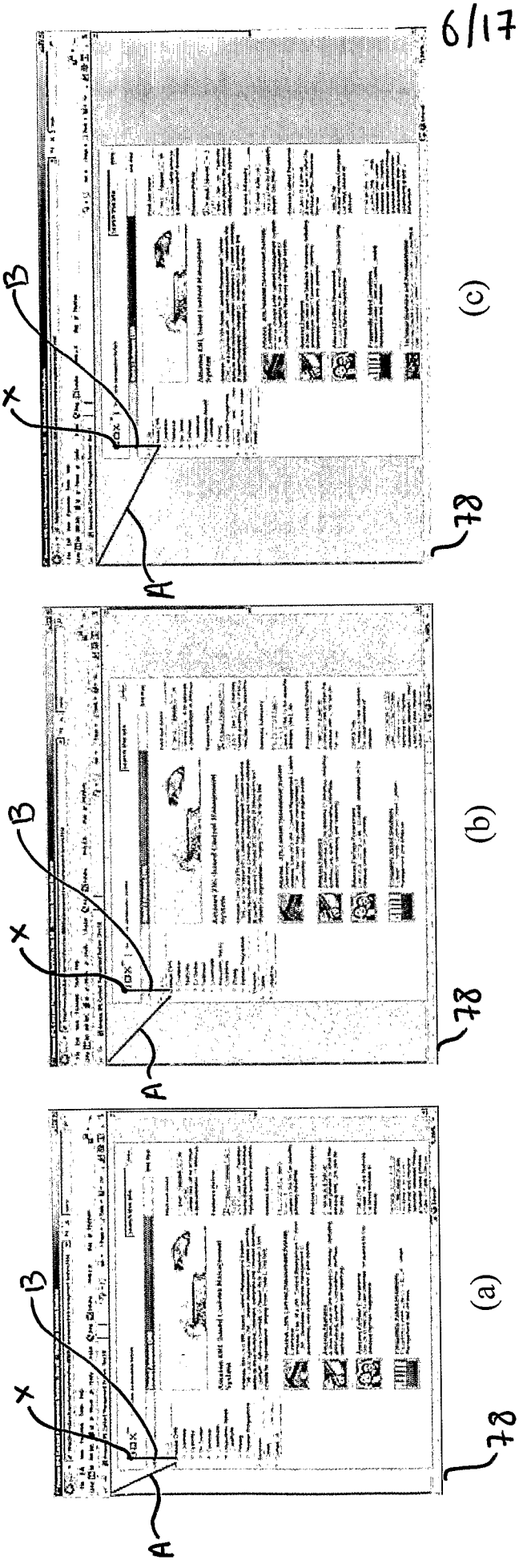


Figure 7

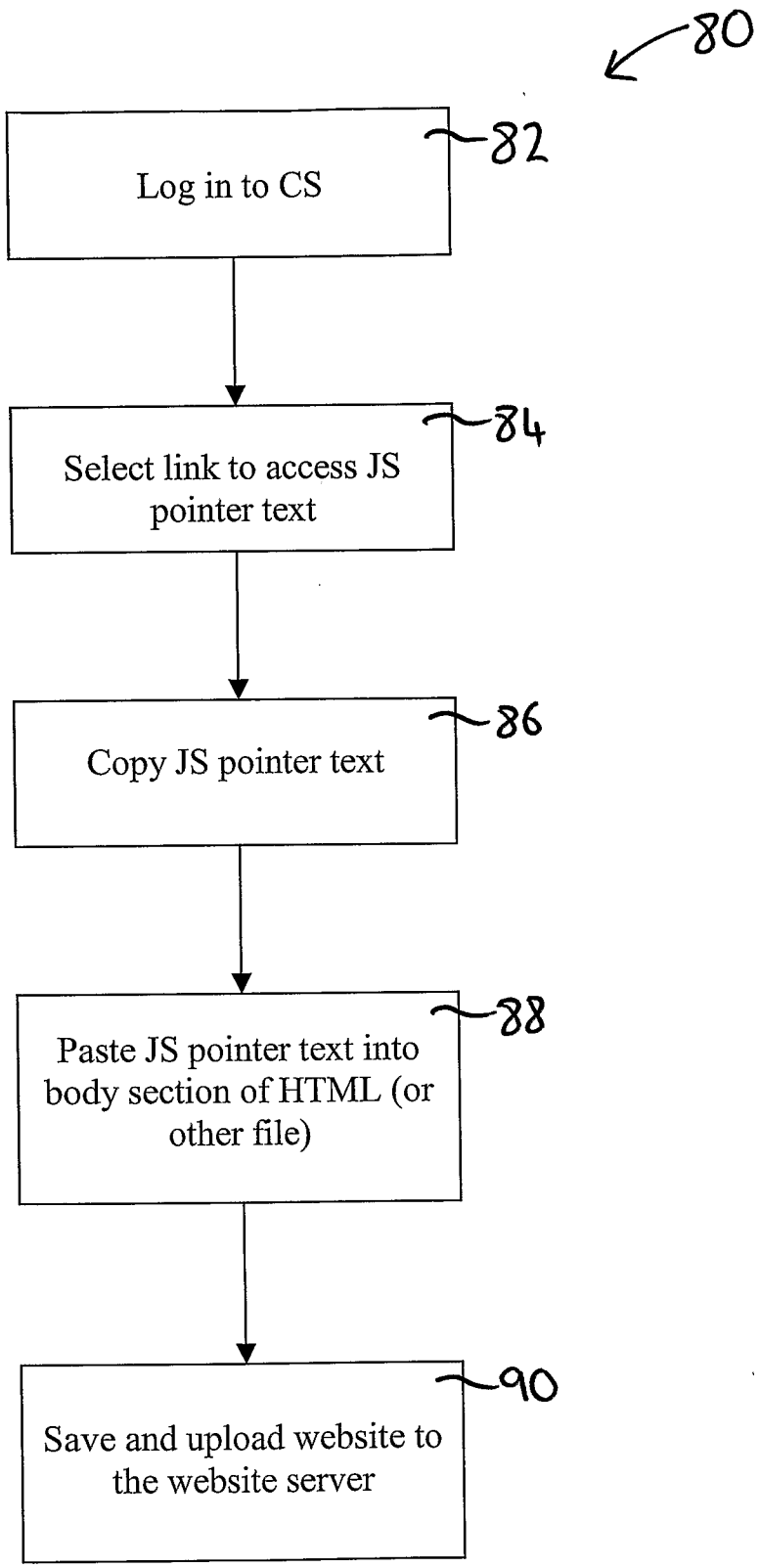

80
Log in to CS
82
Select link to access JS pointer text
84
Copy JS pointer text
86
Paste JS pointer text into body section of HTML (or other file)
88
Save and upload website to the website server
90

Figure 8

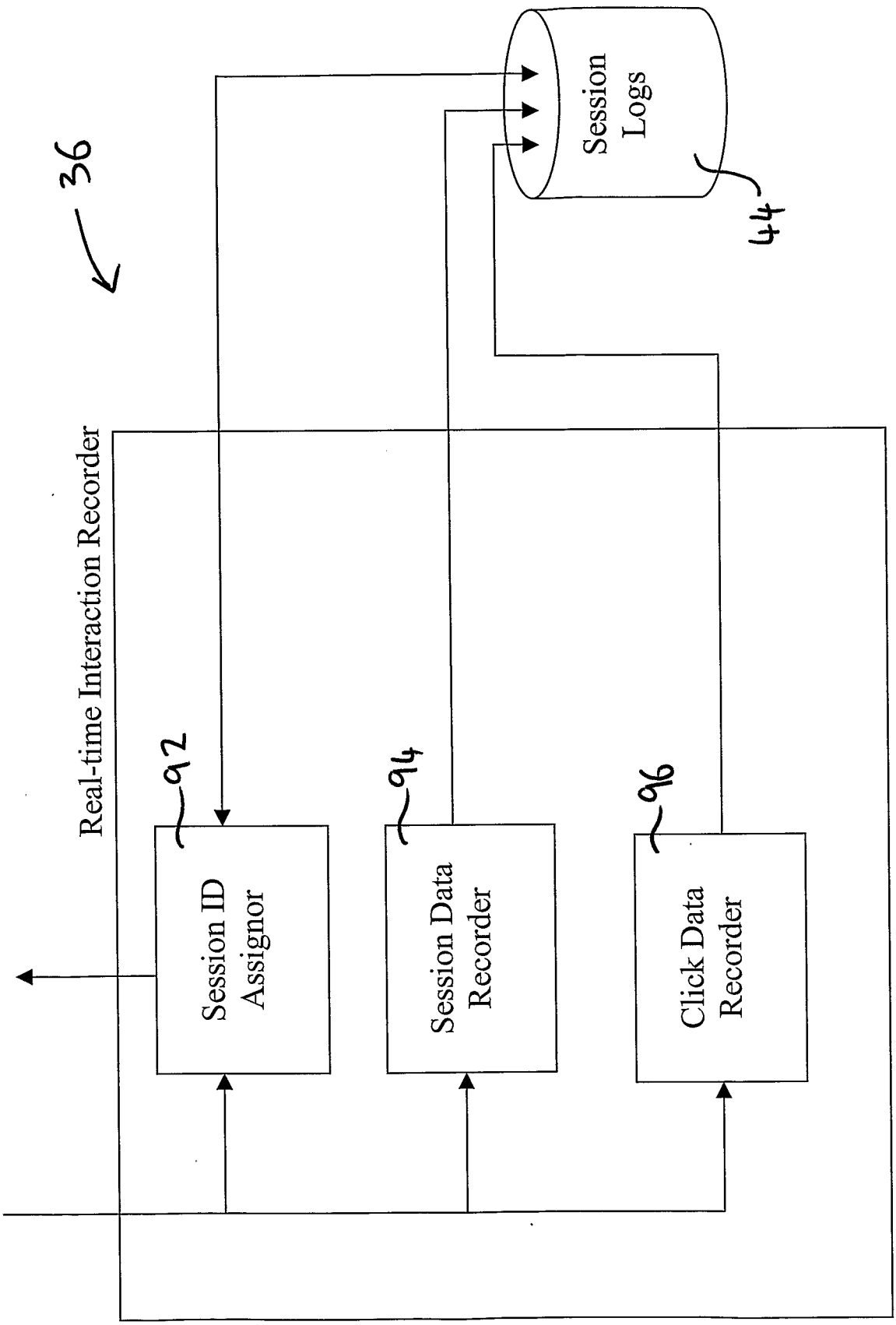

36
Session Logs
44
Real-time Interaction Recorder
92
Session ID Assignor
94
Session Data Recorder
96
Click Data Recorder

Figure 9

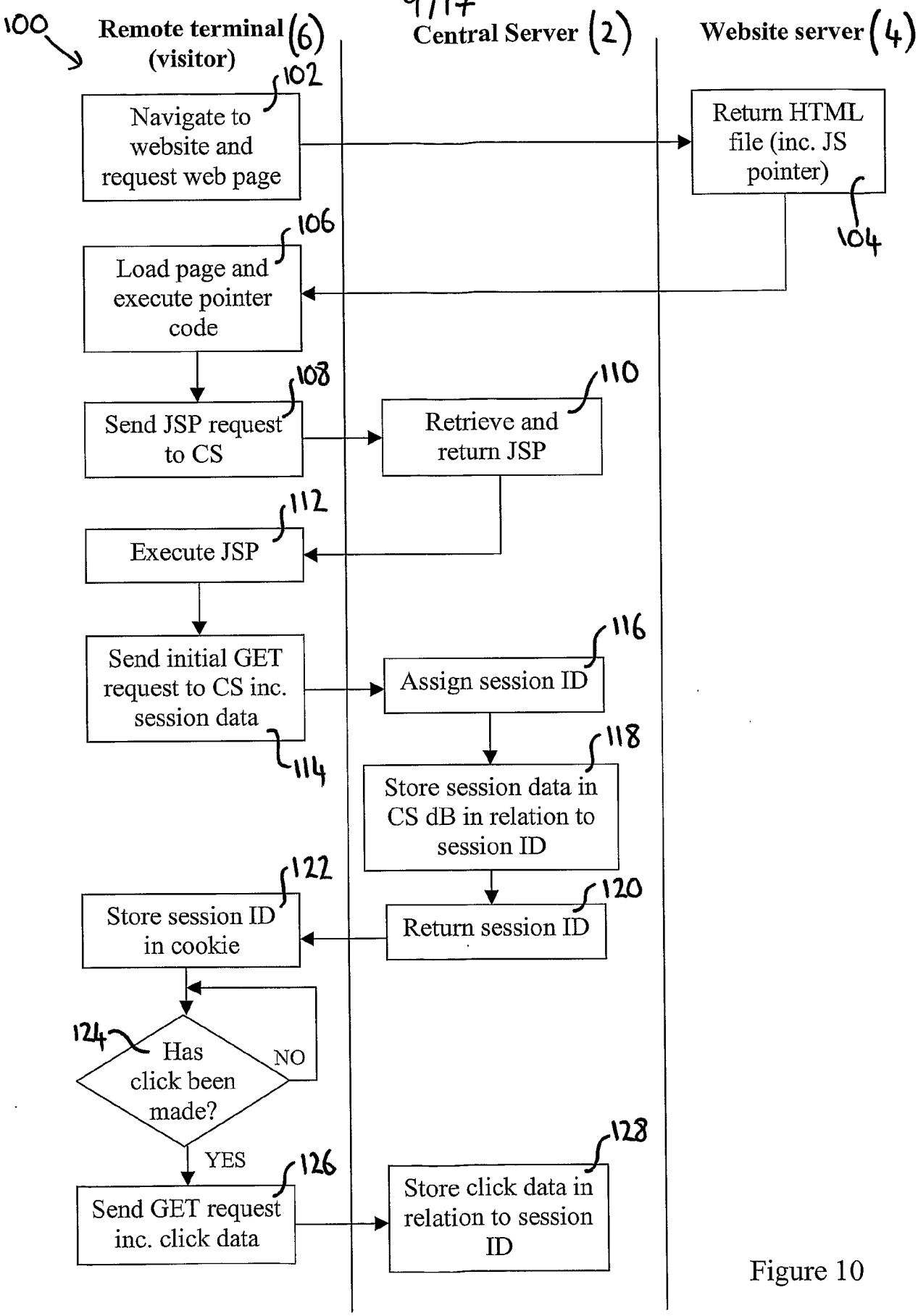

100
Remote terminal (6) (visitor)
Central Server (2)
Website server (4)
102
Navigate to website and request web page
Return HTML file (inc. JS pointer)
104
106
Load page and execute pointer code
108
Send JSP request to CS
110
Retrieve and return JSP
112
Execute JSP
Send initial GET request to CS inc. session data
114
116
Assign session ID
118
Store session data in CS dB in relation to session ID
120
Return session ID
122
Store session ID in cookie
124
Has click been made?
NO
YES
126
Send GET request inc. click data
128
Store click data in relation to session ID

Figure 10

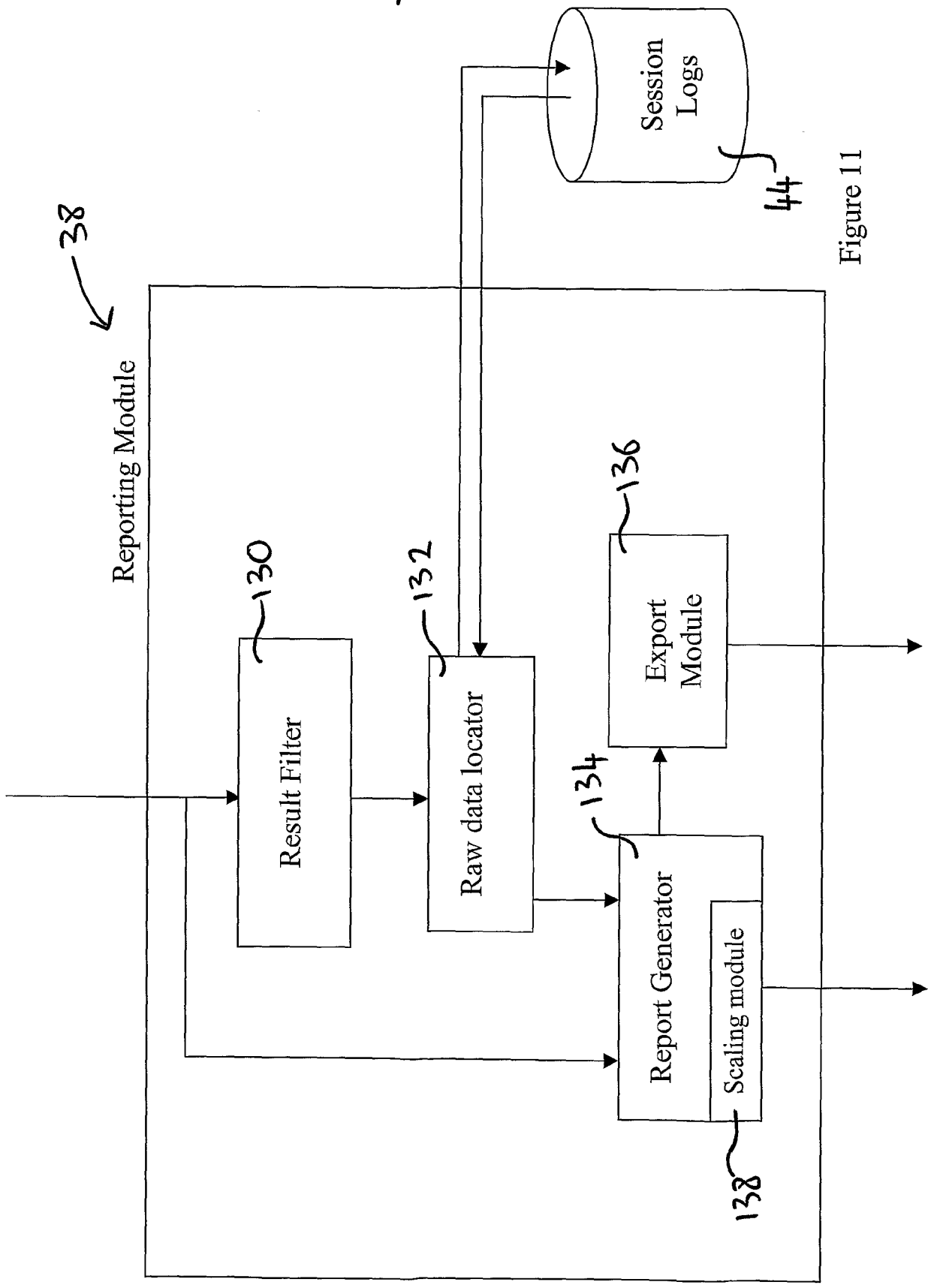
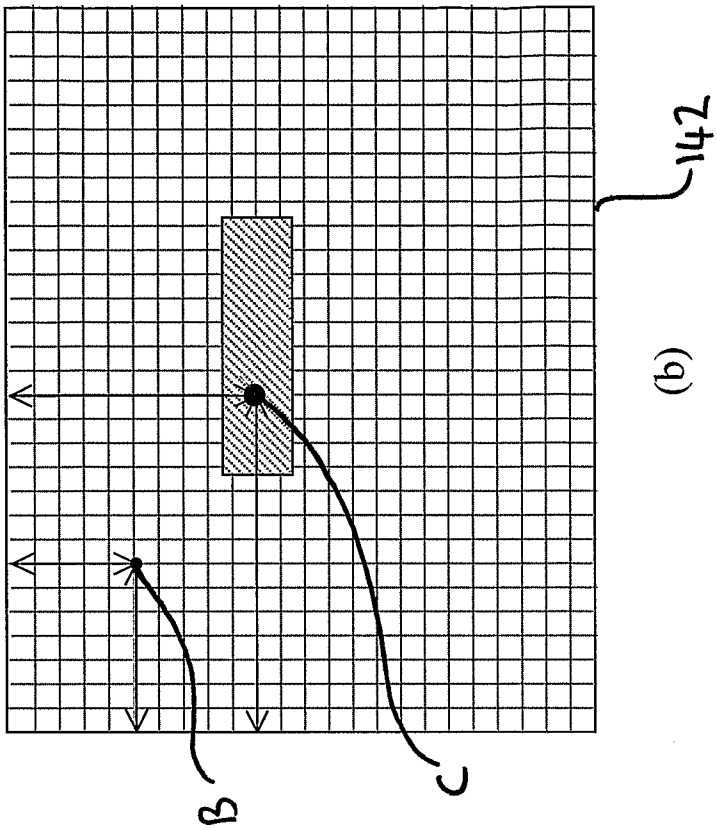

38
Reporting Module
Session Logs
44
130
Result Filter
132
Raw data locator
134
Report Generator
Scaling module
136
Export Module
138

Figure 11



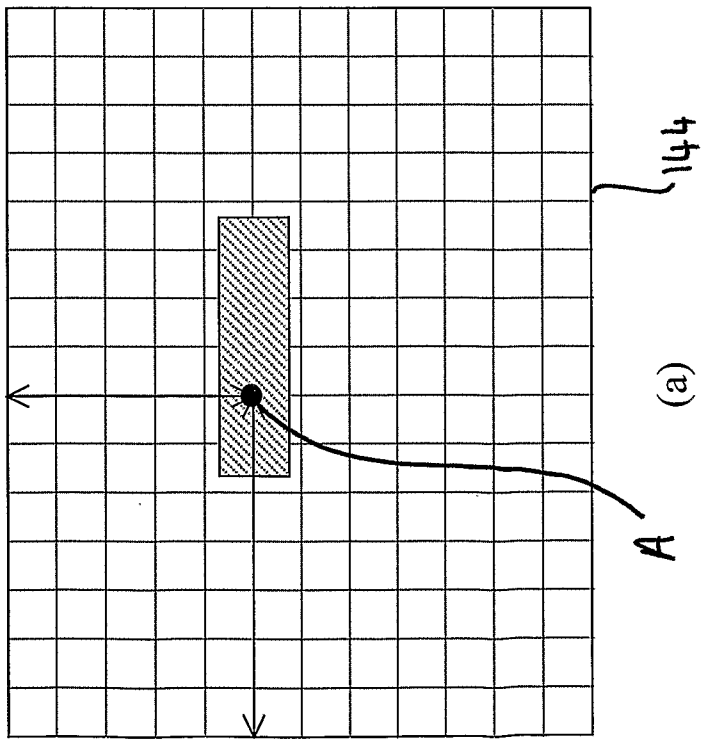


Figure 12

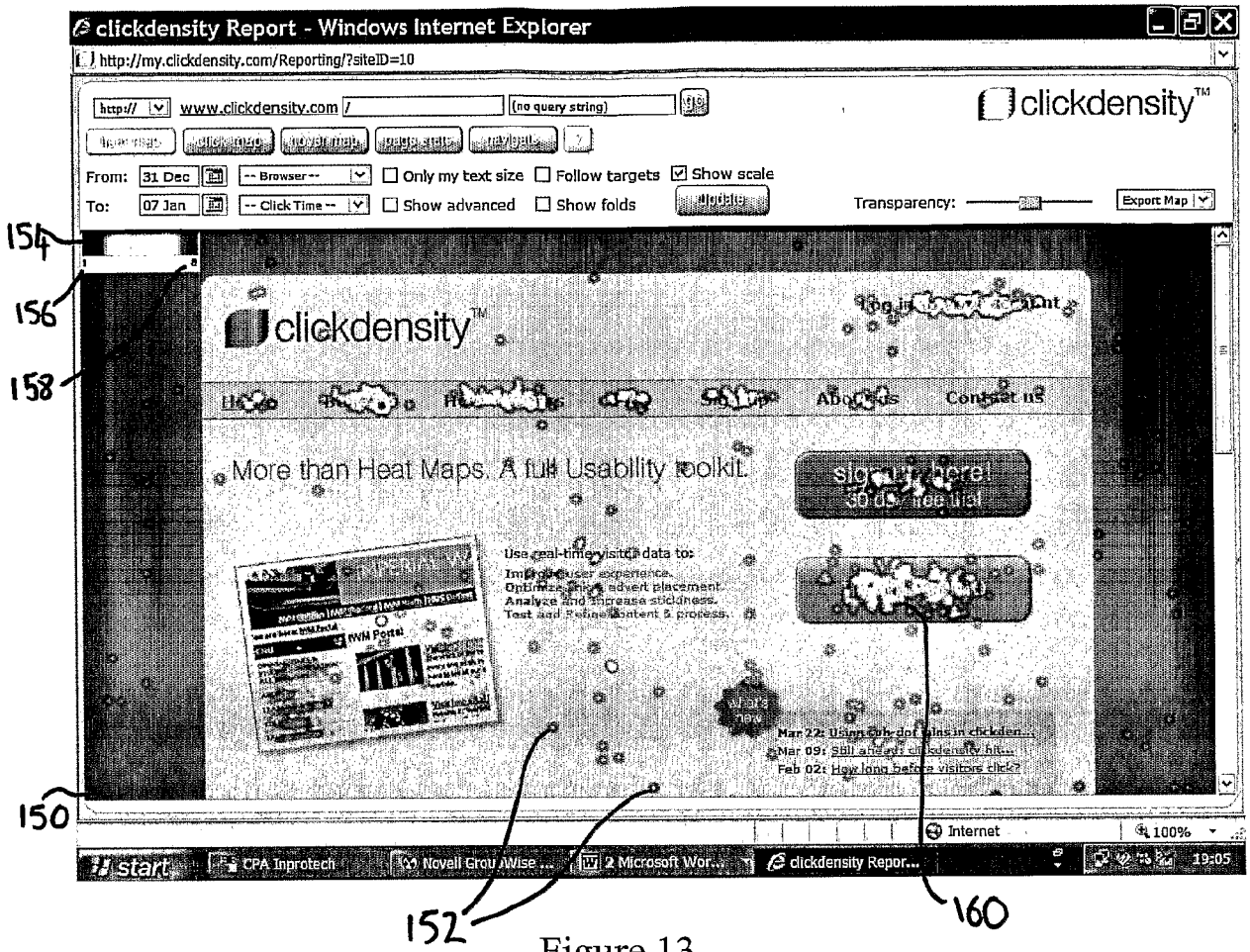


Figure 13

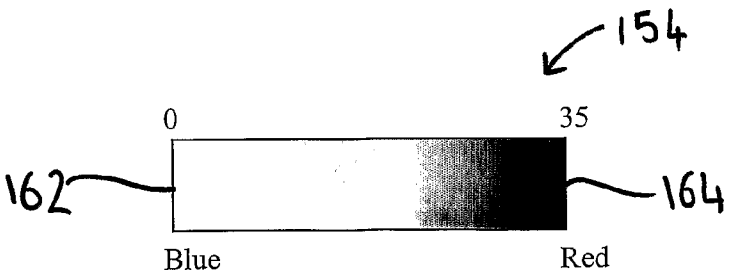


Figure 14

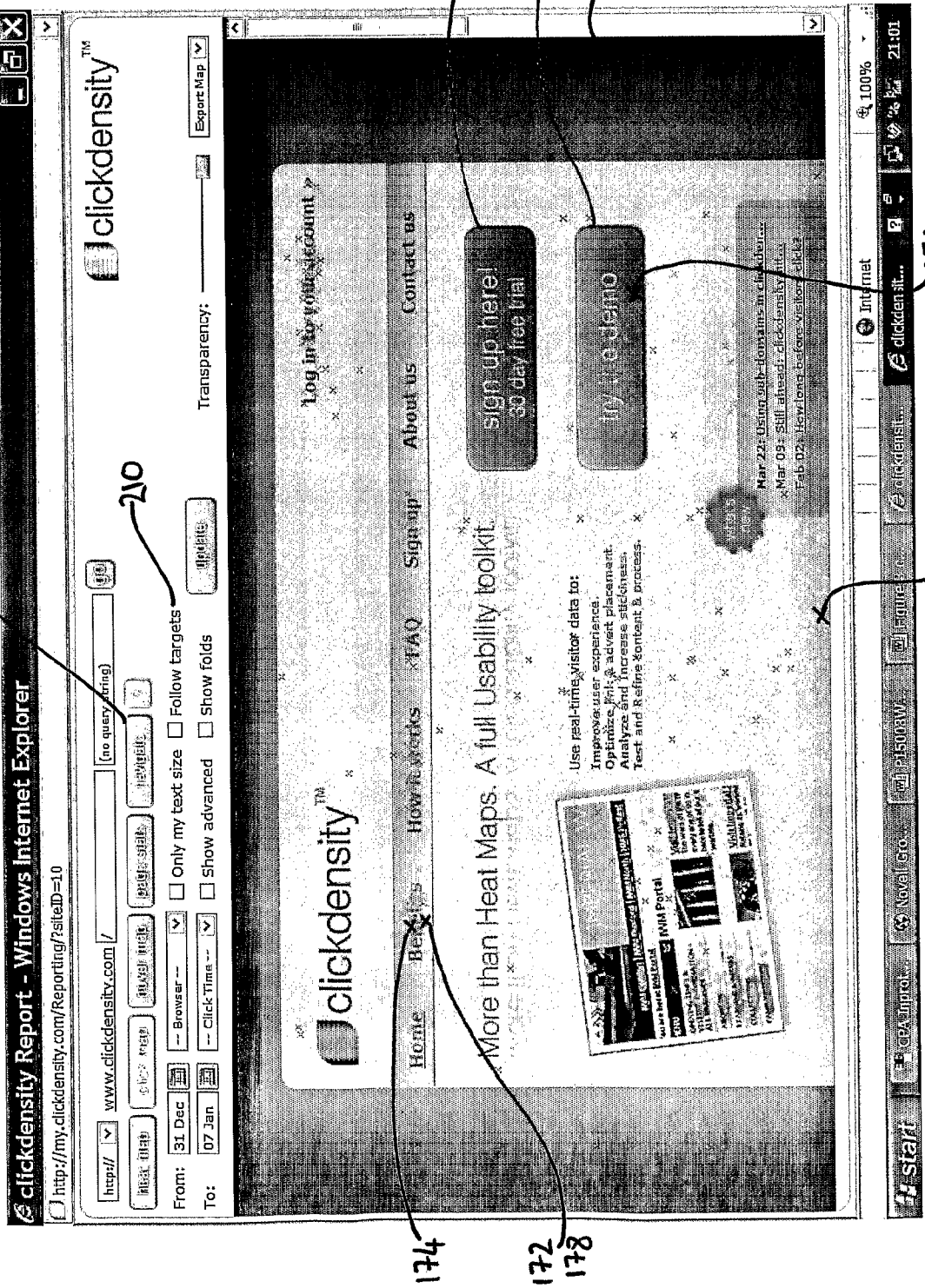

clickdensity Report - Windows Internet Explorer
http://my.clickdensity.com/Reporting/?siteID=10
clickdensity
www.clickdensity.com
Only my text size
Follow targets
Show advanced
Show folds
Transparency:
Export Map
From: 31 Dec
To: 07 Jan
-- Browser --
-- Click-Time --
Update
Log in to your account
Home
How it Works
FAQ
Sign up
About us
Contact us
More than Heat Maps. A full Usability toolkit.
sign up here!
30 day free trial
Use real-time visitor data to:
Improve user experience.
Optimise link & advert placement.
Analyze and increase stickiness.
Test and refine content & process.
Internet
100%
21:01
190
210
176
184
170
174
172
178
172
174

Figure 15

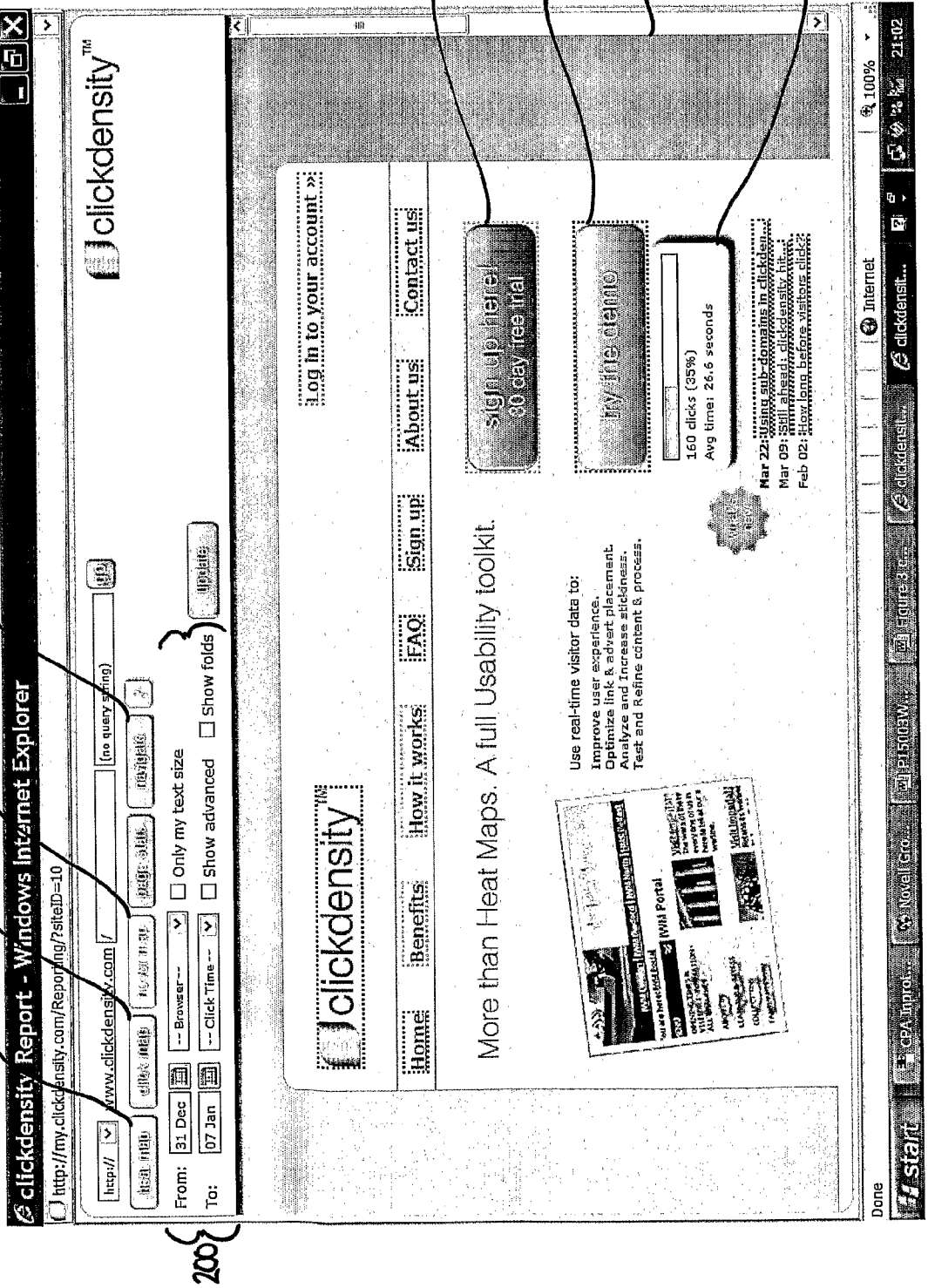

clickdensity Report - Windows Internet Explorer
http://my.clickdensity.com/Reporting/?siteID=10
clickdensity™
www.clickdensity.com
(no query string)
From: 31 Dec
To: 07 Jan
-- Browser --
-- Click Time --
Only my text size
Show advanced
Show folds
Update
clickdensity™
Log in to your account »
Home
Benefits
How it works
FAQ
Sign up
About us
Contact us
More than Heat Maps. A full Usability toolkit.
Use real-time visitor data to:
Improve user experience.
Optimize link & advert placement.
Analyze and increase stickiness.
Test and Refine content & process.
Sign up here!
30 day free trial
Try the demo
160 clicks (35%)
Avg time: 26.6 seconds
Mar 22: Using sub-domains in clickden...
Mar 09: Still ahead: clickdensity hit...
Feb 02: How long before visitors click?
Done
Internet
100%
start
21:02
192
194
196
190
176
184
180
182
200

Figure 16

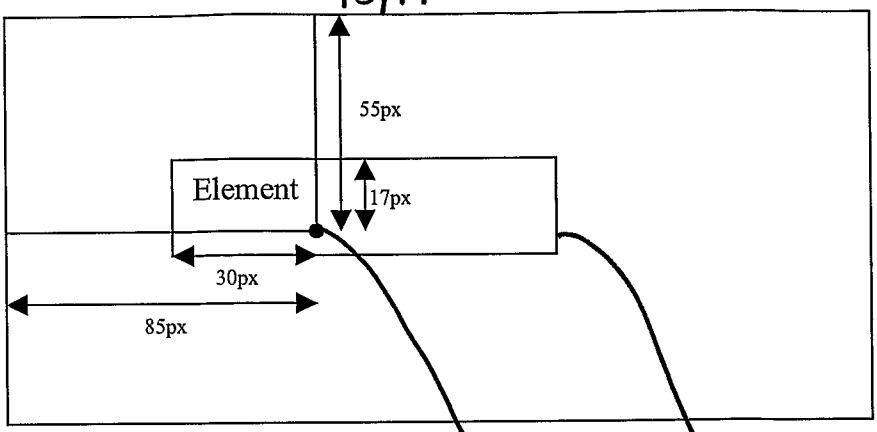


Figure 17

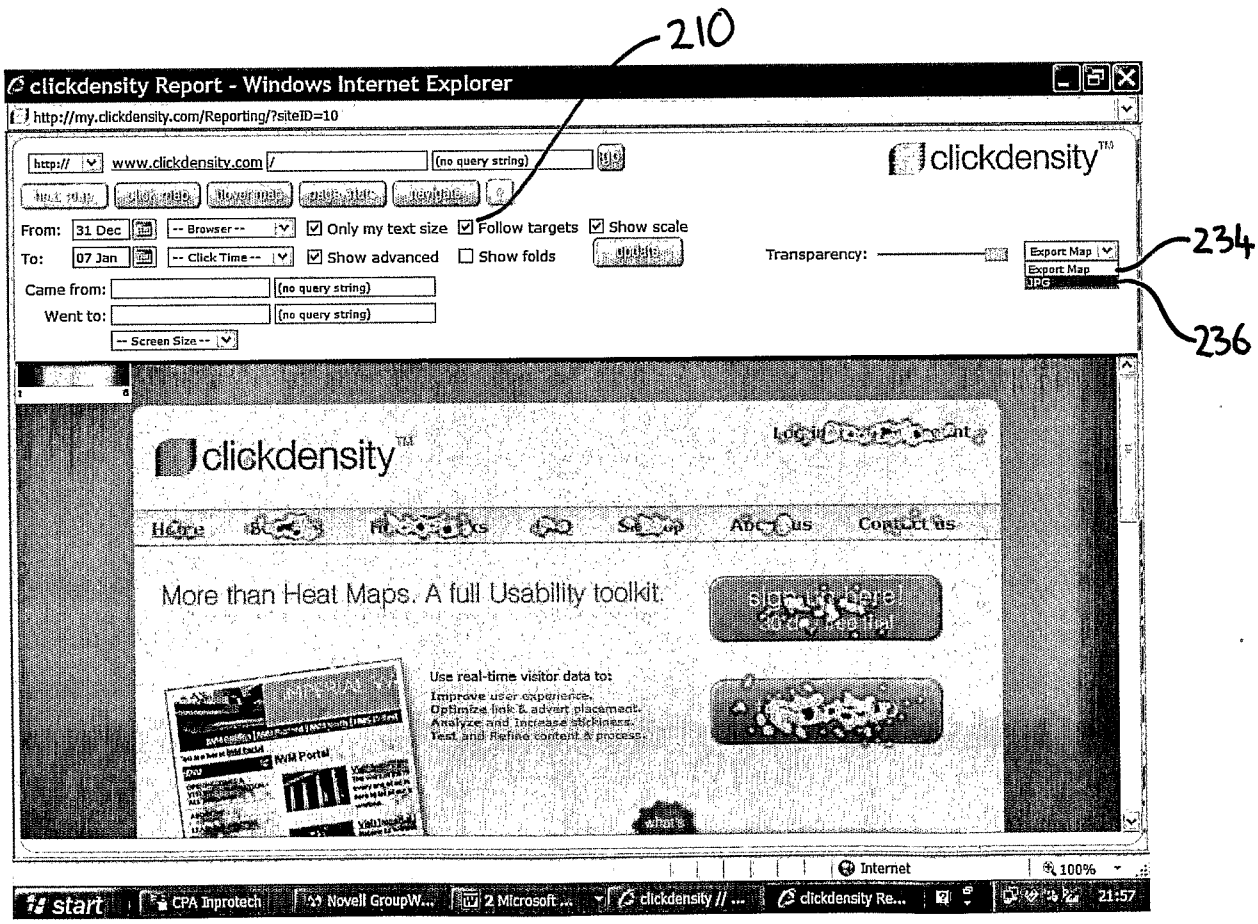


Figure 18

16/17

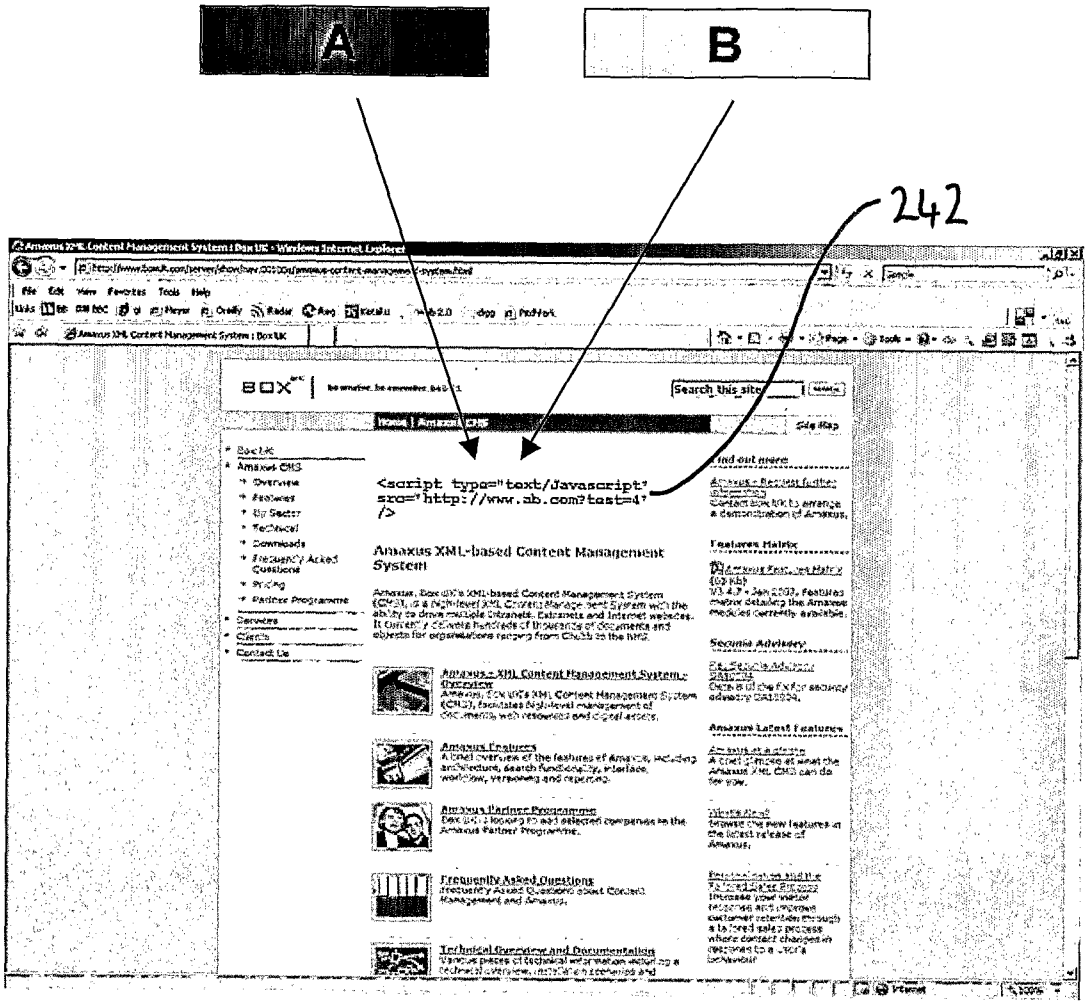


Figure 19

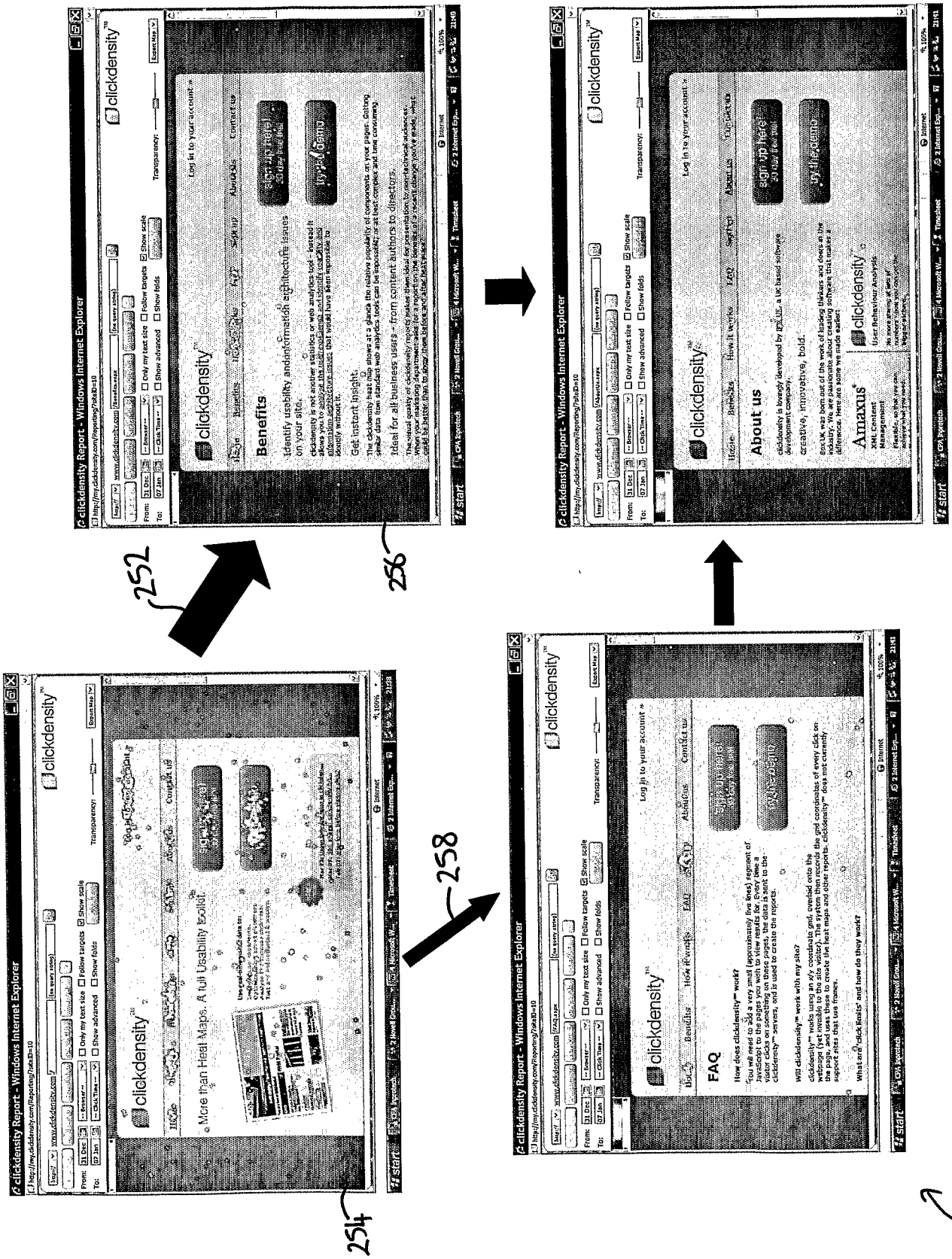

clickdensity Report - Windows Internet Explorer
clickdensity
More than Heat Maps. A full Usability toolkit.
Benefits
Identify usability and information architecture issues on your site.
Get instant insight.
Ideal for all business users - from content authors to directors.
FAQ
How does clickdensity™ work?
Will clickdensity™ work with my site?
What are 'click trails' and how do they work?
About us
clickdensity is lovingly developed by ipx Ltd., a uk based software development company.
creative, innovative, bold.
Amaxus
XML Content Management
User Behaviour Analysis
sign up here!
try the demo
252
254
256
258
250

Figure 20