

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
7 September 2001 (07.09.2001)

PCT

(10) International Publication Number
WO 01/65801 A2

(51) International Patent Classification⁷: **H04L 29/06**

Philip, James [GB/GB]; Elidir, Tomline Court, Nacton, Suffolk IP10 0HX (GB).

(21) International Application Number: PCT/GB01/00662

(74) Agent: **ROBINSON, Simon, Benjamin**; BT Group Legal Services, Intellectual Property Department, Holborn Centre, 8th floor, 120 Holborn, London EC1N 2TE (GB).

(22) International Filing Date: 16 February 2001 (16.02.2001)

(25) Filing Language: English

(81) Designated States (*national*): AU, CA, JP, US.

(26) Publication Language: English

(84) Designated States (*regional*): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

(30) Priority Data:
00301648.2 1 March 2000 (01.03.2000) EP

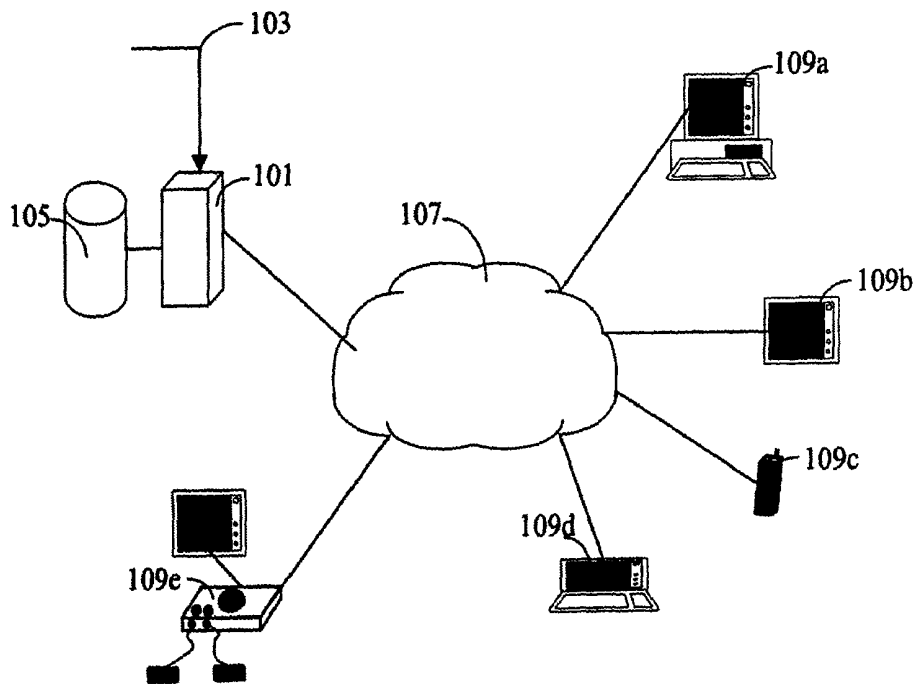
(71) Applicant (*for all designated States except US*): **BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY** [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

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

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): **POWER, Mark,**

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: DATA TRANSFER METHOD AND APPARATUS



(57) Abstract: A method and apparatus are disclosed for transferring data between computers over a network such as the internet or an intranet and is particularly useful for updating web pages with information particularly while being viewed by a user. The method and apparatus disclosed help to reduce the amount of data that is required to be downloaded between the server and client computers and also helps to reduce the flicker which can be experienced when internet page data is refreshed.



WO 01/65801 A2

Data Transfer Method and Apparatus

The present invention relates to the transfer of data between computers over a communications link and/or network. In particular the invention relates to the transfer
5 of data from a server computer to a client computer in a client/server environment, for example when a client computer is used in combination with a web browser to view web pages resident on the server computer.

One mechanism by which interaction in web pages is commonly provided is using a
10 software system called Java. Java can provide a system of programs that reside on the client computer and provide a secure execution environment (called the Java Virtual Machine (JVM)) on the client computer which is separate from the normal execution environment of the client computer (for example Windows). Most web browsers are compatible with Java, i.e. browsers have the functionality to provide a
15 JVM to run downloaded programs called Java Applets. Java Applets provide many types of functionality such as screen animation and other display features including on-screen forms and tables for data display, input and elicitation. For example, an Applet could be used to carry out periodic database queries and present the results of the query to the user. The database could hold share price information that is
20 frequently changing and requires presenting to a user at regular intervals.

However, one problem with the Java solution is that web pages that include Java Applets generally take longer to download than standard web pages. Furthermore, the JVM takes time to start-up before it can run an Applet and requires more
25 processing power than is needed to run the browser alone. These drawbacks become more significant when the client computer is of restricted processing power or is linked to the server computer by a low bandwidth network connection.

An alternative to the Java approach is to use Client Pull and/or Server Push CGI
30 techniques as described in detail in "CGI Programming on the World Wide Web" by Shishir Gundavaram (Published by O'Reilly & Associates, USA). The simpler of the two techniques is Client Pull which can, for example, be used to provide an updating mechanism for information on a web page by associating the page with a CGI script.

A CGI script is a program that is resident on the server that starts working in response to a request from a client computer for the associated web page. The CGI script can be used to carry out a database query and insert the results into the web page which is then downloaded to the requesting client.

5

The Client Pull and Server Push techniques can be combined with a browser feature referred to as frames which most browsers provide. This feature allows the browser window to be split into distinct areas each referred to as a frame and each having the properties of a normal browser window. The user interaction that occurs in one frame
10 can be independent of both the content and user interaction of other frames. Frames can be visible or invisible i.e. displayed or not displayed on the screen.

Web pages can also have a feature called a refresh tag which causes the browser displaying the page to repeatedly download the page at predetermined intervals
15 defined by the tag. Refresh tags enable information on a web page to be automatically updated (i.e. without user intervention) while being viewed. The process typically has the following steps:

- a) the server receives a page request (resulting from a user request);
- b) the server runs the CGI script associated with the requested page, perhaps
20 enters new data into the page as a result and then sends the updated page to the client that requested it along with a refresh tag;
- c) the client computer displays the received page and starts a timer corresponding to the refresh tag; and
- d) when the refresh time is up, the client automatically repeats its request to
25 the server for the page (or a different one) and the process starts again at step a).

The second of the techniques, Server Push, works in a similar manner to Client pull except that the server outputs a multiple part message to the client, each part of which can contain a page with information updated using a CGI script. Again, a timer
30 can be used to start the CGI script or alternatively the script can be run only when the data to be displayed in the page has been updated. The main difference is that the network connection between the client and sever in a Server Push mechanism is kept open for the duration of the client/server interaction for that page.

One problem with the Client Pull and Server Push mechanisms noted above is that each time the page being viewed is updated, the page is redisplayed and therefore the page (or screen) is seen by the user to flicker. This makes studying the page for
5 any length of time uncomfortable for a user. Furthermore, if the refresh rate for the page is high (e.g. every few seconds) then the flicker effect is uncomfortable even for limited viewing of the page. In order to reduce the flicker frequency the refresh rate of the page can be reduced but in situations where changes in the data presented by the page have to be reacted to promptly this may not be feasible.

10

According to the present invention there is provided a method of updating data displayed on an apparatus from a data source over a communications link or network, said data comprising a graphical element and one or more updatable data elements, the graphical element being arranged to have one or more updatable areas
15 corresponding to the or each updatable data element(s) for displaying the or each updatable data element, said method comprising the steps of:

- in response to a request from the apparatus to the data source, transmitting the graphical element from the data source to the apparatus for display of the graphical element;
- 20 transmitting the or each data element to the apparatus;
- entering the or each data element in the corresponding updatable area(s) of the graphical element;
- transmitting one or more updated data elements to the apparatus; and
- entering the or each of the updated data elements in the corresponding
25 updatable area(s) of the graphical element for display.

Splitting a page to be displayed into two pages, one being static and displayed the other being dynamic and not displayed means that only the changing data, i.e. the dynamic page, has to be refreshed. Also, because only the data within the static
30 page is redisplayed, the redisplay of the whole page is avoided. These features alleviate problems of the prior art by reducing flicker, reducing the amount of data that has to be downloaded and/or reducing the amount of processing required at the client and/or server computers.

Embodiments of the invention are described below with reference the accompanying figures in which:

figure 1 is a schematic diagram showing a plurality of client devices
5 connected to a server device via a network such as the internet;

figures 2a and 2b are diagrammatic representations of the pages and data stored on the sever device of figure 1 containing parts to be displayed on the client device of figure 1;

figure 3 is a process diagram of the processing and display of the pages of
10 figure 2 according to a first embodiment of the invention;

figure 4 is a process diagram of the processing and display of the pages of figure 2 according to a second embodiment of the invention; and

figure 5 is a diagrammatic representation of pages and data stored on the sever device of figure 1 containing parts to be displayed on the client device of figure
15 1 in accordance with a third embodiment of the present invention;

With reference to figure 1, a server computer 101 is fed with data via a data feed 103 and is arranged to process the data and store the results in a database 105. The data feed 103 provides, for example, test data produced automatically from a
20 telecommunications network or financial information such as exchange rates between a plurality of currencies. In either case the data is such that it changes relatively frequently.

The sever 101 is also connected to a network 107 such as the internet or an
25 intranet, to which network 107 a plurality of client devices 109 are also connected such as a personal computer 109a, a digital television 109b, a mobile telephone 109c, a personal digital assistant 109d or a games console 109e. The server computer 101 is arranged with suitable server software to communicate over the network 107 with each of the client devices 109 and each client device 109 is
30 arranged with corresponding software so as to be able to communicate with the server computer 101 over the network 107.

The server 101, in this embodiment is an IBM RS/6000 J50 computer with server software installed called Hyper Text Transport Protocol Daemon (Httpd) from Apache Software Inc or alternatively Oracle Web Server from Oracle Corporation. Each of the client devices 109 is installed with a suitable web browser. In the case for the personal computer 109 the web browser could be Netscape from Netscape Inc or Explorer from Microsoft Corp.. Others of the client devices 109 are installed with suitable types of client software arranged to provide network access and browsing which depend on the particular device and method of connection to the network 107.

10 The Httpd or Oracle Web Server (OWS) software when running on the server computer 101 is capable of sending files to the client devices 109 and to run programs on the server 101 in response to a request from a browser running on one of the client devices 109. If the server software receives a request for a plain file such as an HTML file (Hyper Text Markup Language – a format commonly used for web pages), then the server software looks up the file in the server 101 directory system and sends the file across the network 107 to the requesting browser. Such an HTML file can be referred to as a static page.

The server 101 may also receive a request to run a process using server application software such as a query on the database 105 via a server database application. In this case an interface mechanism called the Common Gateway Interface (CGI) is used to manage communication between the Httpd or OWS software and the server application software. The CGI mechanism is arranged to take parameters from the incoming process request, pass them to the server application software and to return the results of the process request to the requesting browser. The results returned to the browser can be referred to as a dynamic page because the content of the page can be changed from time to time.

In the present embodiment, the server 101 uses the CGI mechanism to interface with a database application which processes incoming data from the data feed 103 and stores the results in the database 105. In this embodiment the data relates to the status(es) of alarms which are used to alert engineers to faults occurring in a telecommunications network.

With reference to figure 2, the data in the database 105 is presented to the user using two pages – a static page 201 and a dynamic page 203. The static page 201 is set out as an HTML form 202 containing a plurality of labelled boxes 205 each of which is blank. A form is a standard HTML function which is commonly used for
5 obtaining input i.e. used to allow a user to enter data for transmission back to the server 101. The static page 201 also contains a script 209 which, in this embodiment, is written using the JavaScript language (as will be appreciated by those skilled in the art JavaScript is distinct from Java). The form 202 is the part of
10 the static page 201 that is actually displayed on the client while the script 209 is not intended for display and so is normally invisible.

The dynamic page 203 comprises no visible part i.e. the page is not displayed on the browser. Instead, the dynamic page 203 comprises data structure 207 having a
15 plurality of elements that correspond to the boxes 205 of the form 202 in the static page 201. The data structure 207 in this embodiment is a JavaScript array.

When a browser downloads the static page 201 it firstly displays the form 202 and then runs the script 209. The script 209 is arranged to load the dynamic page 203
20 into an invisible frame, extract the data from its data structure 207 and insert the data into the appropriated parts of the form 202. The script 209 includes a refresh tag 211 that at predetermined intervals causes a request for a refreshed dynamic page 203 to be sent to the server 101. The server 101 has a CGI script (not shown) which is arranged, in response to a request from the browser, to access the database
25 105 to obtain the latest alarm status data and to populate the array 207 in the dynamic page 203 with that fresh data. These processes are explained in further detail below with reference to figures 3 and 4.

Figure 3 shows a flow diagram of the processing in accordance with the first
30 embodiment of the pages 201, 203 by the client 109 and the server 101 in response to a client 109 user's request to view the alarm status page 201, 203. At step 301, in response to the user making the request (by entering the URL (Universal Resource Locator) of the page 201, 203 or using a hyperlink), the client opens up a link via the

network 107 to the server 101. At step 303 the client 109 sends the request for the alarm status page 201,203. In response to the request, the server, at step 305, runs the CGI script associated with the alarm status page 201, 203 and at step 307 receives the fresh data from the database 105. The fresh data is inserted into the data structure 207 of the dynamic page 203 at step 309.

At step 313, if the request in step 303 was for the whole page (i.e. included the static page 201) then the processing moves on to step 315 and sends the static page 201 to the client 109. At step 317, the client 109 displays the static page 201 to the user in a browser window. Next, at step 319, the server sends the dynamic page 203 to the client which, under the control of the script 209, extracts the refreshed data from the dynamic page 203 and inserts it into the corresponding fields 205 of the displayed static page 201.

At step 323, the client then closes the link over the network 107 to the server 101 and at step 325 sets a timer corresponding to the refresh tag 211. Unless the user views a different page while the timer is running then when the allotted time has elapsed the client 109 checks that the alarm status page 201, 203 is still being viewed and if so, at step 327, opens a link to the server 101 and requests a refreshed dynamic page 203.

In response to the request the server re-runs the CGI script at step 305 and carries out steps 307 and 309 as noted above. At step 313, the fact that the request is a refresh request results in the processing moving to step 319 and the sending of the refreshed dynamic page 203 to the client. At step 321, the client processes the new refreshed data from the dynamic page as noted above and continues to steps 323, 325 and 327. The process of refreshing the data in the static page 201 with data from the dynamic page 203 will continue until the user chooses a different page to view via the browser on the client 109.

30

Figure 4 shows a flow diagram of the processing in accordance with the second embodiment of the invention of the pages 201, 203 by the client 109 and the server 101 in response to a client 109 user's request to view the alarm status page 201,

203. At step 401, in response to the user making a request, the client 109 opens up a link via the network 107 to the server 101. At step 403 the client 109 sends the request for the alarm status page 201,203. In response to the request, the server, at step 405, runs the CGI script associated with the alarm status page 201, 203 and at
5 step 407 receives the fresh data from the database 105. The fresh data is inserted into the dynamic page 203 at step 409.

At step 413, if the request in step 403 was for the whole page (i.e. included the static page 201) then the processing moves on to step 415 and sends the static
10 page 201 to the client 109. At step 417, the client 109 displays the static page 201 to the user in a browser window. Next, at step 419, the server sends the dynamic page 203 to the client which, under the control of the script 209, extracts the data from the dynamic page 203 and inserts the data into the appropriate fields 205 in the form 202 of the static page 201.

15

At step 423 the server 101 monitors the data for the dynamic page 203 for any updates. When an update to the data is detected the processing moves to step 405 and the server re-runs the CGI script and carries out steps 407 and 409 as described above. In other words, rather than being driven by a refresh timer, the refreshing of
20 the dynamic page 203 in this embodiment is driven by the data to be displayed.

At step 413, the fact that only the dynamic page 203 is being updated results in the processing moving to step 419 and the sending of the refreshed dynamic page 203 to the client 109. At step 421, the client 109 extracts the refreshed data from the
25 dynamic page 203 and inserts it into the appropriate fields 205 of the static page 201 and the server 101 continues to steps 423 and 425 as described above. The process of refreshing the dynamic page 203 will continue until the user chooses a different page to view via the browser on the client 109.

30 Figure 5 shows a third embodiment of the invention in which, a static page 501 comprises a form 502 which is the same as the form 202 described above with reference to figure 2. In addition, the static page 501 comprises a set of symbols 503 that provide a schematic representation of elements in the network from which

the alarms being monitored are derived. Each symbol has two states each of which indicate to the viewer the status of the network element that a given symbol represents e.g. a red symbol for an element with an alarm condition and a black symbol for an element that is operating normally. The static page 501 also includes
5 a dialog box 505 that is used to provide the viewer with text messages describing, for example, the progress of a task such as a test routine being carried out remotely in the network.

The static page 501 also includes a script 509 a refresh tag 511 and data 513. The
10 script 509 functions in substantially the same manner as the script 209 described above with reference to figures 2 and 3 or 4 (the differences will be described below). The data 513 comprises two parts, the first being a set of images each of which represent one of the two possible states of each of the symbols 503. The second part is sets of text messages, each set forming an ordered sequence arranged
15 to provide the dialog for the dialog box 505. The set of text messages are designed to provide dialog for all of the tasks that a user would need to be informed about. As with the previous embodiments, the script, 509 is not displayed and in a similar manner, in this embodiment, the data 513 is stored by the browser until needed as described below.

20

The dynamic page 515 comprises a data structure 517 which, in addition to the data 516 for the form 502 also holds instructions 517a relating to the symbols 503 and instructions 517b relating to the text messages for the dialog box 505. The instructions 517a provide an indication of which of the two possible representations
25 of each symbol (indicating one of the two states of the network element that the symbol is representing) should be displayed. The instructions 517b provide an indication of which of the sequence of messages should next be displayed in the dialog box 505.

30 In this embodiment, the script 509 differs from the script described with reference to figure 3 in that, as well as extracting data 516 from the database 105 that will eventually be displayed in the form 502, the instructions 517a and 517b are also extracted and inserted into the dynamic page 515. When the browser receives the

data 516, it deals with it in the same manner as for the first and second embodiments above. When the browser receives the instructions 517a, 517b it applies the instructions to the symbols 503 and the dialog box 505 respectively. For example, if one of the instructions 517a indicates that one of the symbols 503
5 should change state (e.g. from black to red – indicating a fault at the corresponding network element) then the script accesses the stored images 513 from the static page 501, obtains the alternative image in accordance with the instruction 517a and displays it in the static page 501 in place of the previous image. Similarly, if one of the instructions 517b indicates that the next message in the sequence of messages
10 being displayed in the dialog box 505 should be displayed then the script 509 will obtain the next message from the stored sets of messages 513 and displays the message in the dialog box 505.

As will be understood by those skilled in the art, the instructions 517a, 517b that are
15 inserted into the dynamic page 515 can be a full set that define the state of the symbols 503 and the text in the dialog box 505. Alternatively, after the initial download i.e. for subsequent refreshing of the data in the static page 501, the instructions 517a, 517b could be a minimal set i.e. only include instructions relating to symbols that have changed status or dialog that has moves to the next message in
20 the sequence. In this manner, the amount of processing required to produce the dynamic page 525 and to process the instructions is reduced.

Many browsers support a feature referred to as frames. This means that the browser window can be split into distinct areas each referred to as a frame and each having
25 the properties of a normal browser window. The user interaction that occurs in one frame can be independent of both the content and user interaction of other frames. In the above embodiment, instead of displaying the static page 501 as a single page it could be split into a plurality of frames. For example, the symbols 503 could be displayed in one frame, the dialog box 505 displayed in another frame and the form
30 502 displayed in a further frame.

As will be understood by those skilled in the art, the static page 501 could be arranged without the symbols 503 or without the dialog box 505. Alternatively, the static page 501 could be arranged without the form 502.

5 As an alternative additional feature in the first embodiment, the server 101 is provided with a second refresh timer which controls the refreshing of the dynamic page 203. The second timer has a shorter time period than the first timer on the client 109. The first timer on the client 109 is used to only to stimulate the upload of the dynamic page 203 (as the refreshing of the dynamic page is now controlled by
10 the second server side timer). This feature can be used to ensure that the server is not running the CGI script more often than the database 105 is updated or that the CGI script is run so frequently that the processing power of the server 101 is not used efficiently.

15 In the first and second embodiments described above, the refresh tag is associated with the static page 203. As an alternative, the tag could be associated with the dynamic page 201 and each time the dynamic page 203 is refreshed the tag 211 could be varied under the control of the client 109 or the server 101 so as to be dependant on the bandwidth/transmission rate available across the network 107. The
20 bandwidth available may depend on the traffic load on the network 107 or the mode of connection between the client and the network e.g. PSTN, ISDN, ADSL, GSM (or other mobile system). As a further alternative, the refresh tag 211 could be dependent on the data that is inserted into the dynamic page or vary with respect to time in accordance with a predetermined function. The tag could be set by and under
25 the control of the user.

As will be understood by those skilled in the art, the static and dynamic pages described above could be stored and created separately on separate servers. Furthermore, the static and dynamic pages could be uploaded to an intermediate
30 server arranged to assemble the page into its display form and then allow client apparatus to access the page and display it.

As will be understood by those skilled in the art, any or all of the software used to implement the invention can be contained on various transmission and/or storage mediums such as a floppy disc, CD-ROM, or magnetic tape so that the program can be loaded onto one or more general purpose computers or could be downloaded over
5 a computer network using a suitable transmission medium.

Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise", "comprising" and the like are to be construed in an inclusive as opposed to an exclusive or exhaustive sense; that is to say, in the sense
10 of "including, but not limited to".

CLAIMS

1. A method of updating data displayed on an apparatus from a data source over a communications link or network, said data comprising a graphical element and
5 one or more updatable data elements, the graphical element being arranged to have one or more updatable areas corresponding to the or each updatable data element(s) for displaying the or each updatable data element, said method comprising the steps of:
- in response to a request from the apparatus to the data source, transmitting
10 the graphical element from the data source to the apparatus for display of the graphical element;
- transmitting the or each data element to the apparatus;
- entering the or each data element in the corresponding updatable area(s) of the graphical element;
- 15 transmitting one or more updated data elements to the apparatus; and
- entering the or each of the updated data elements in the corresponding updatable area(s) of the graphical element for display.
2. A method for acquiring data from a data source over a communications link
20 or network, said data comprising a graphical element and one or more updatable data elements, the graphical element being arranged to have one or more updatable areas corresponding to the or each updatable data element(s) for displaying the or each updatable data element, said method comprising the steps of:
- requesting the data from the data source;
- 25 receiving the graphical element and the or each elements from the data source;
- entering the or each of the data elements in the corresponding updatable area(s) of the graphical element for display;
- acquiring one or more updated data elements from the data source; and
- 30 entering the or each of the updated data elements in the corresponding updatable area(s) of the graphical element for display.

3. A method for providing data from a data source to a destination apparatus over a communications link or network, said data comprising a graphical element and one or more updatable data elements, the graphical element being arranged to have one or more updatable areas corresponding to the or each updatable data element(s) for display of the or each updatable data element, said method comprising the steps of:

acquiring the data elements from a data source;

transmitting the graphical element and the or each data element to the destination apparatus for entry in the corresponding updatable area(s) of the graphical element for display; and

acquiring one or more updated data elements for transmission to the destination apparatus for entry in the corresponding updatable area(s) of the graphical element for display.

4. A system comprising a first apparatus for storing data and a second apparatus for displaying data said apparatuses being interconnected by a communications link or network, wherein said data comprises a graphical element and one or more updatable data elements, the graphical element being arranged to have one or more updatable areas corresponding to the or each updatable data element(s) for displaying the or each updatable data element, said first apparatus comprising:

means for acquiring the or each data elements from a data source and for updating the data elements periodically;

means operable to transmit the graphical element to the second apparatus;

means operable to transmit the or each data elements to the second apparatus; and

said second apparatus comprising:

means for entering the or each of the data elements in the corresponding updatable area(s) of the graphical element for display;

means operable to acquire from the first apparatus one or more updated data elements to the second apparatus; and

means operable to enter the or each of the updated data elements in the corresponding updatable area(s) of the graphical element for display.

5. Apparatus for acquiring data from a data source over a communications link or network, said data comprising a graphical element and one or more updatable data elements, the graphical element being arranged to have one or more updatable areas
5 corresponding to the or each updatable data element(s) for displaying the or each updatable data element, said apparatus comprising:

requesting means operable to request the data from the data source;

receiving means for receiving the graphical element and the or each data element from the data source;

10 processing means operable to enter the or each of the data elements in the corresponding updatable area(s) of the graphical element for display;

refresh means operable to acquire one or more updated data elements from the data source; and

said processing means being further operable to enter the or each of the
15 updated data elements in the corresponding updatable area(s) of the graphical element for display.

6. Apparatus for providing data from a data source to a destination apparatus over a communications link or network, said data comprising a graphical element and
20 one or more updatable data elements, the graphical element being arranged to have one or more updatable areas corresponding to the or each updatable data element(s) for display of the or each updatable data element, said apparatus comprising:

acquiring means for acquiring the data elements from a data source;

transmitting means operable to transmit the graphical element and the or
25 each data element to the destination apparatus for entry in the corresponding updatable area(s) of the graphical element for display; and

the acquiring means being operable to acquire one or more updated data elements for transmitting by said transmitting means to the destination apparatus for entry in the corresponding updatable area(s) of the graphical element for display.

30

7. A method of updating data displayed on an apparatus from a data source over a communications link or network, said data comprising:

a graphical element having one or more updatable areas for displaying a graphical sub-elements;

one or more data sets comprising two or more alternative graphical sub-elements; and

5 one or more sub-element instructions identifying the alternative graphical sub-elements from the or each data set to be entered into an area of the graphical element for display,

said method comprising the steps of:

10 in response to a request from the apparatus to the data source, transmitting the graphical element and the or each data set from the data source to the apparatus for display of the graphical element;

transmitting the or each sub-element instructions to the apparatus;

15 in accordance with the or each sub-element instruction entering the appropriate graphical sub-element in the corresponding updatable area(s) of the graphical element;

transmitting one or more replacement sub-element instructions to the apparatus; and

20 in accordance with the or each replacement sub-element instruction, entering the or each appropriate graphical sub-element in the corresponding updatable area(s) of the graphical element for display.

8. A method for acquiring data from a data source over a communications link or network, said data comprising:

25 a graphical element having one or more updatable areas for displaying a graphical sub-elements;

one or more data sets comprising two or more alternative graphical sub-elements; and

30 one or more sub-element instructions identifying the alternative graphical sub-elements from the or each data set to be entered into an area of the graphical element for display,

said method comprising the steps of:

requesting the data from the data source;

receiving the graphical element, the or each data set and the or each graphical sub-element instruction from the data source;

in accordance with the or each graphical sub-element instruction, entering the or each graphical sub-element in the corresponding updatable area(s) of the graphical element for display;

acquiring one or more replacement graphical sub-element from the data source; and

in accordance with the or each replacement graphical sub-element instruction, entering the or each appropriate graphical sub-elements in the corresponding updatable area(s) of the graphical element for display.

9. A method for providing data from a data source to a destination apparatus over a communications link or network, said data comprising:

a graphical element having one or more updatable areas for displaying a graphical sub-elements;

one or more data sets comprising two or more alternative graphical sub-elements; and

one or more sub-element instructions identifying the alternative graphical sub-elements from the or each data set to be entered into an area of the graphical element for display,

said method comprising the steps of:

determining the or each sub-element instruction and transmitting said sub-element instructions to the destination apparatus;

transmitting the graphical element and the or each data set to the destination apparatus for entry in the corresponding updatable area(s) of the graphical element for display in accordance with the or each sub-element instruction; and

determining one or more replacement sub-element instructions for transmission to the destination apparatus for enabling replacement of the appropriate graphical sub-elements in the corresponding updatable area(s) of the graphical element for display.

10. A system comprising a first apparatus for storing data and a second apparatus for displaying data, said apparatuses being interconnected by a communications link or network, said data comprising:

a graphical element having one or more updatable areas for displaying a
5 graphical sub-elements;

one or more data sets comprising two or more alternative graphical sub-elements; and

one or more sub-element instructions identifying the alternative graphical sub-elements from the or each data set to be entered into an area of the graphical
10 element for display,

said first apparatus comprising:

means operable in response to a request from the second apparatus , to transmit the graphical element and the or each data set to the second apparatus for display of the graphical element; and

15 means operable to transmit the or each sub-element instructions to the second apparatus,

said second apparatus comprising;

means operable in accordance with the or each sub-element instruction to enter the appropriate graphical sub-element in the corresponding updatable area(s) of
20 the graphical element;

means operable to acquire one or more updated sub-element instructions;

and

means operable in accordance with the or each updated sub-element instructions, to enter the or each appropriate graphical sub-element in the
25 corresponding updatable area(s) of the graphical element for display.

11. An apparatus for acquiring data from a data source over a communications link or network, said data comprising:

a graphical element having one or more updatable areas for displaying a
30 graphical sub-elements;

one or more data sets comprising two or more alternative graphical sub-elements; and

one or more sub-element instructions identifying the alternative graphical sub-elements from the or each data set to be entered into an area of the graphical element for display,

said apparatus comprising:

5 requesting means operable to request the data from the data source;

receiving means for receiving the graphical element, the or each data set and the or each graphical sub-element instruction from the data source;

processing means operable in accordance with the or each sub-element instruction, to enter the or each appropriate graphical sub-element in the
10 corresponding updatable area(s) of the graphical element for display; and

update means operable to acquire one or more updated sub-element instructions from the data source,

said processing means being further operable to enter the or each graphical sub-elements in accordance with the or each updated sub-element instruction in the
15 corresponding updatable area(s) of the graphical element for display.

12. An apparatus for providing data from a data source to a destination apparatus over a communications link or network, said data comprising:

a graphical element having one or more updatable areas for displaying a
20 graphical sub-elements;

one or more data sets comprising two or more alternative graphical sub-elements; and

one or more sub-element instructions identifying the alternative graphical sub-elements from the or each data set to be entered into an area of the graphical
25 element for display,

said apparatus comprising:

acquiring means for acquiring data determining the or each graphical sub-elements instruction from the data source; and

transmitting means operable to transmit the graphical element and the or
30 each data set to the destination apparatus for entry in the corresponding updatable area(s) of the graphical element for display,

the acquiring means being further operable to acquire one or more updated sub-element instructions for transmission to the destination apparatus for enabling

entry of the appropriate graphical sub-elements in the corresponding updatable area(s) of the graphical element for display.

13. A computer program or suite of computer programs for use with one or more
5 computers to carry out the method as set out in any one of claims 1 to 3 and 7 to 9
or to provide any of the apparatus as set out in any one of claims 4 to 6 or 10 to 12.

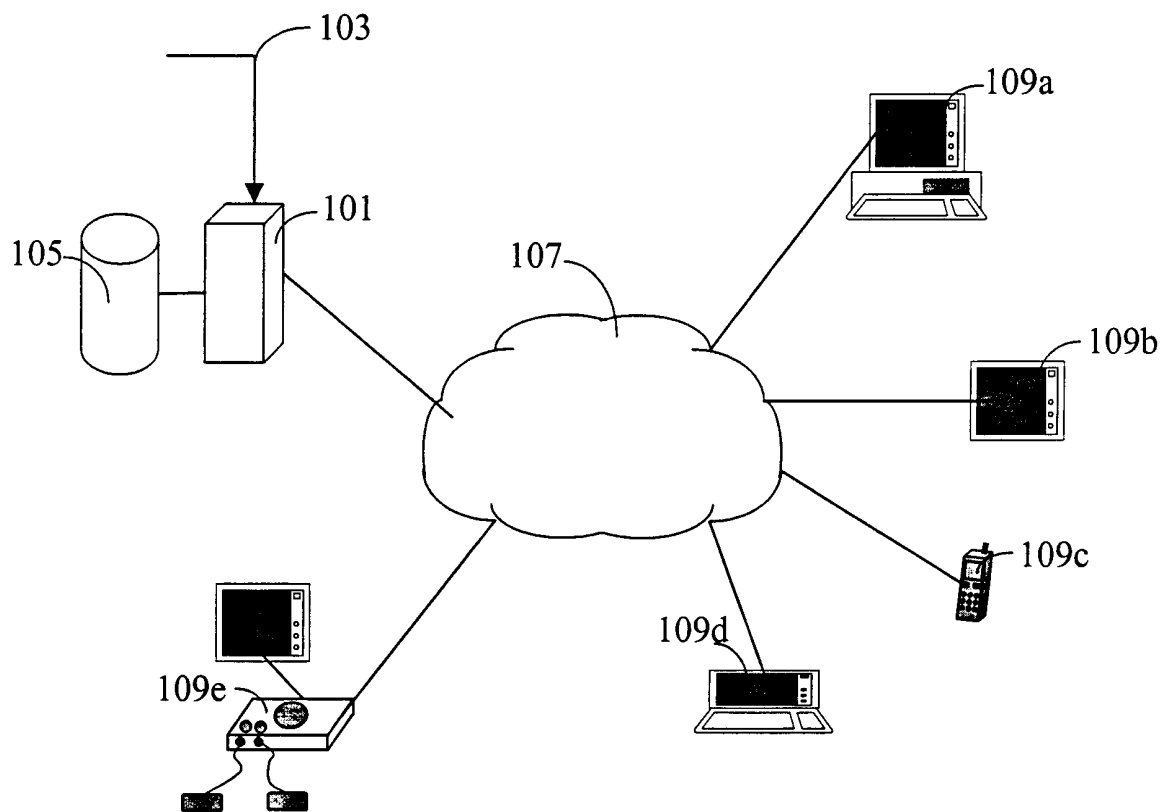


Figure 1

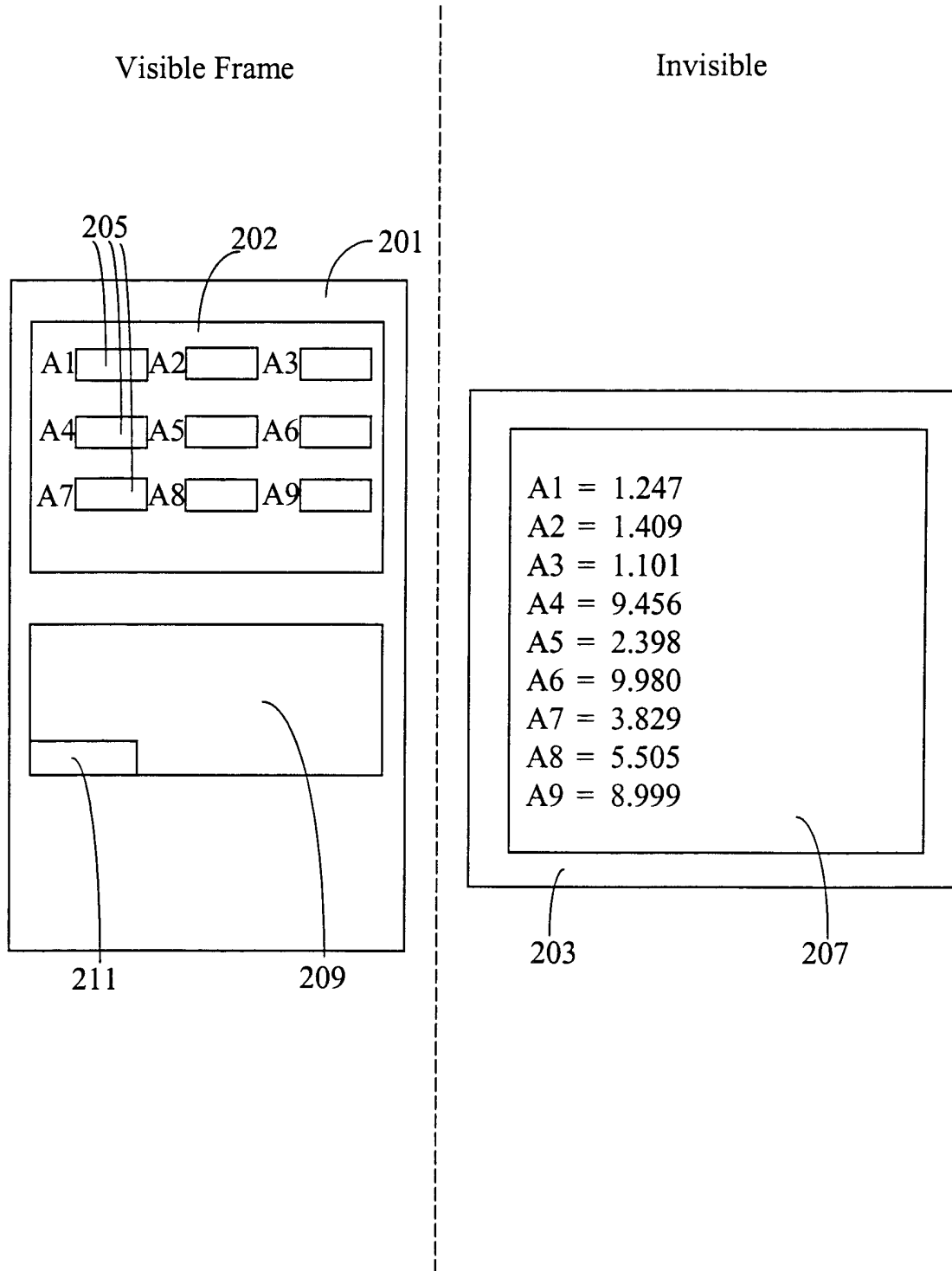


Figure 2

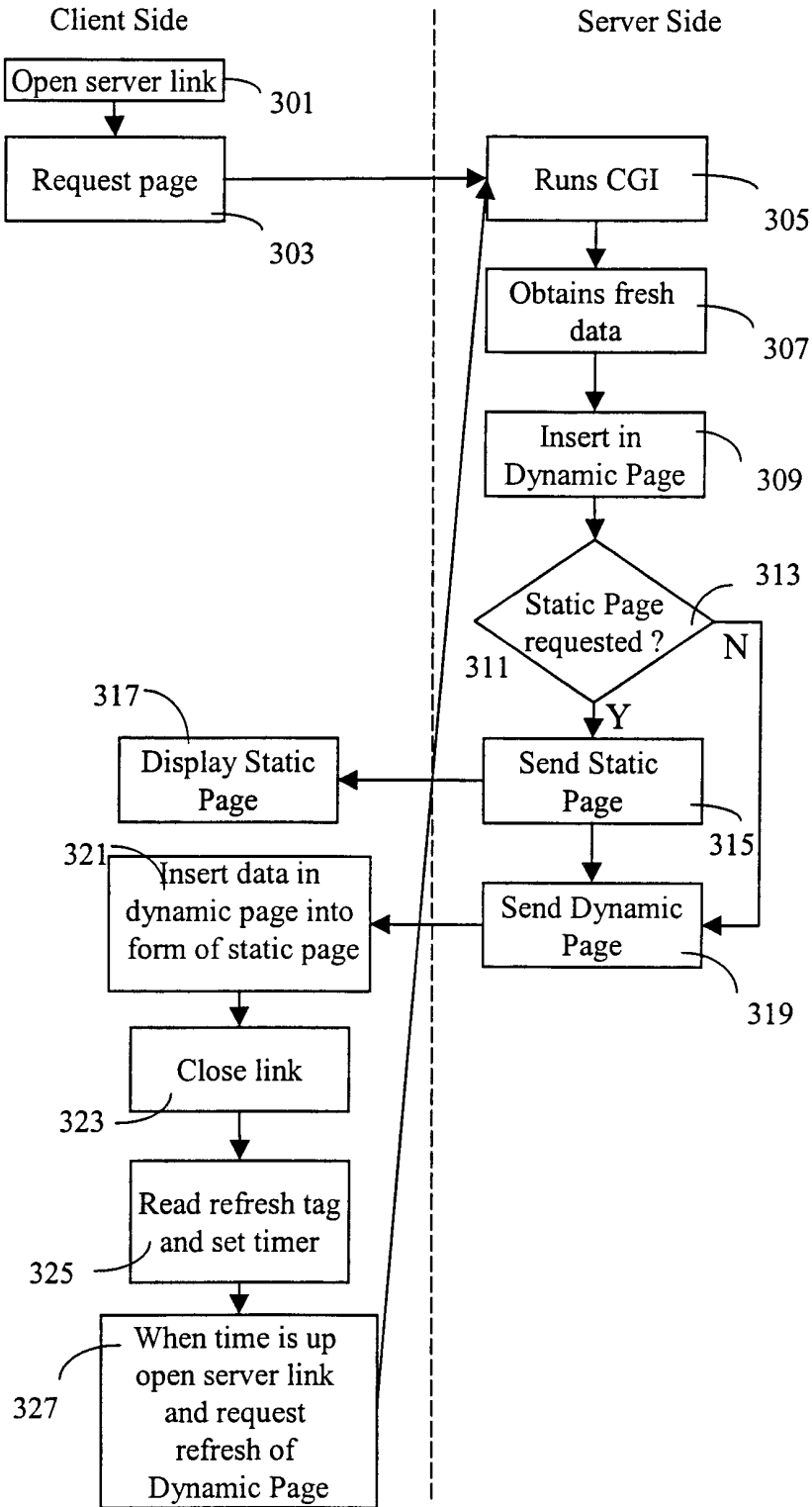


Figure 3

4/5

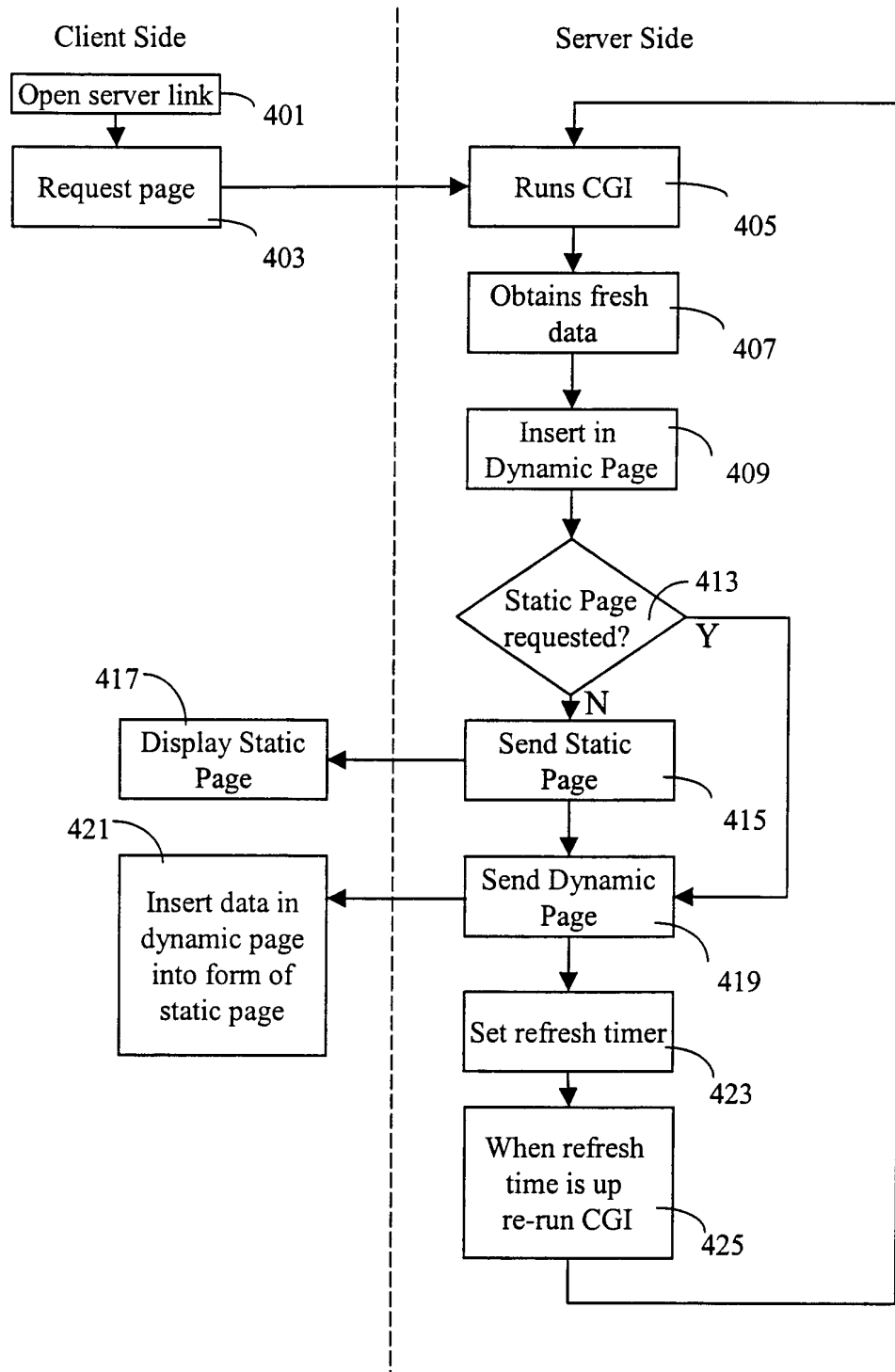


Figure 4

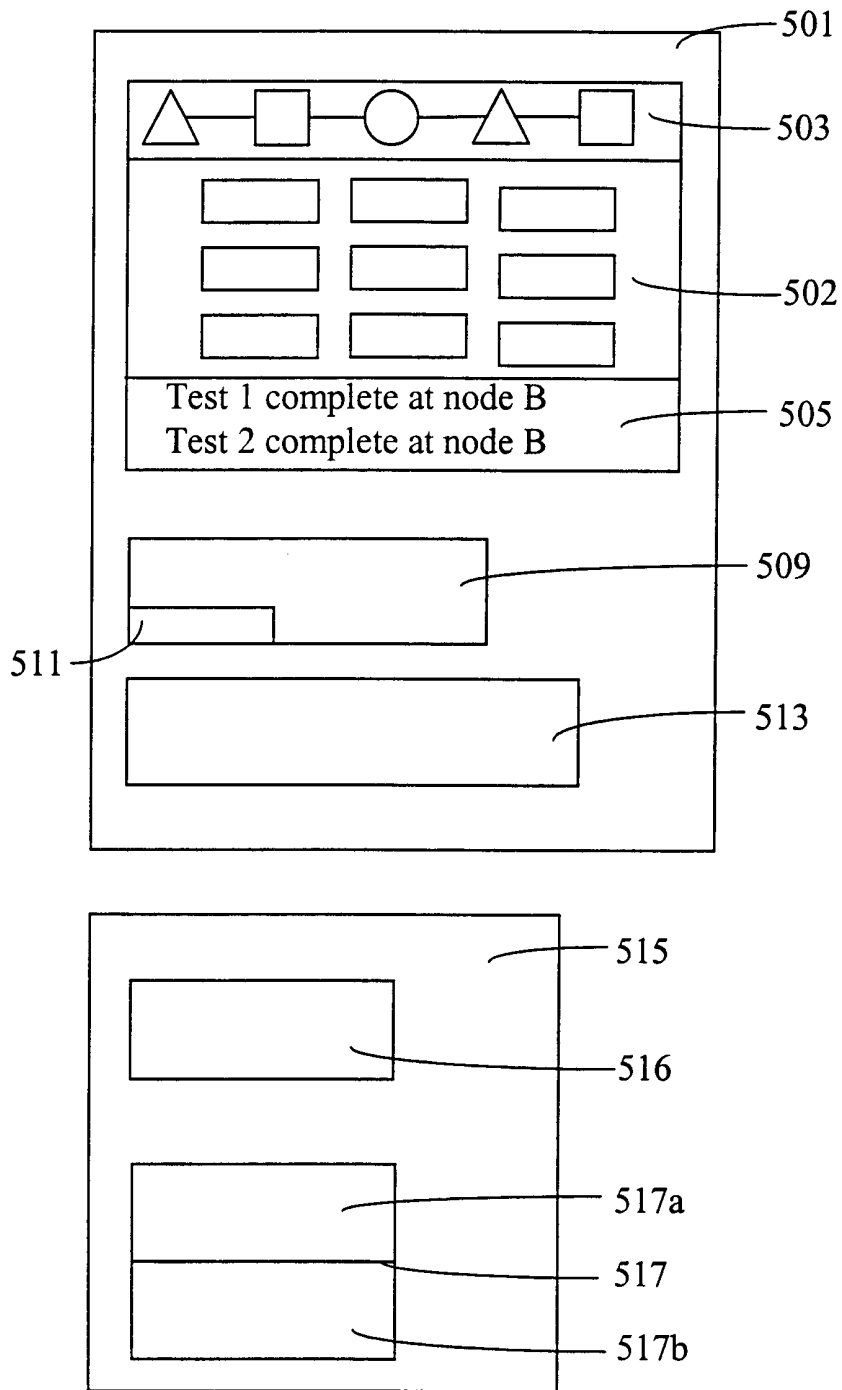


Figure 5