12-20-00

A/PROV

jc715 U.S. PTO 12/18/00

PROVISIONAL APPLICATION

jc784 U.S. PTO 60/256613 12/18/00

Attorney Docket No. 20689-000200US

"Express Mail" Label No. EL624000491US
Date of Deposit: December 18, 2000

ASSISTANT COMMISSIONER FOR PATENTS
BOX PROVISIONAL PATENT APPLICATION
Washington, D.C. 20231
Sir:

Transmitted herewith for filing is a provisional patent application under CFR 1.53(c) of:

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (City/State/Country)
Tuttle	Timothy		San Francisco, CA United States

Title: UPDATING DISPLAY AREAS USING DIGITAL NETWORK TRANSFER OF INFORMATION

Enclosed are:
[X] 16 pages of the specification (including description and claims).
[X] 5 sheet(s) of [ ] formal [X] informal drawing(s).
[ ] Abstract.
[ ] A verified statement to establish small entity status under 37 CFR 1.9 and CFR 1.27.
[ ] The invention was made by or under a contract with the following agency of the United States Government: _____ under Government contract number _____
[X] Attachments as listed in specification

Please charge Deposit Account No. 20-1430 as follows:

[X] Filing fee $ $150.00
[X] Any additional fees associated with this paper or during the pendency of this application.

2 extra copies of this sheet are enclosed.

Respectfully submitted,
TOWNSEND and TOWNSEND and CREW LLP

Charles J. Kulas
Reg. No. 35,809
Attorney for Applicant

Customer No. 20350

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Telephone: (415) 576-0200
Fax: (415) 576-0300

SF 1170966 v1

Attorney Docket No.: 020689-000200

PROVISIONAL PATENT APPLICATION

UPDATING DISPLAY AREAS USING DIGITAL NETWORK TRANSFER OF INFORMATION

Inventor(s):

Timothy Tuttle
3449 22nd Street, Apt. 8
San Francisco, CA 94110-2957
Citizen: U.S.A.

Assignee:

Bang Networks
808 Brannan Street
San Francisco, CA 94109

Entity: Small

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 415-576-0200

UPDATING DISPLAY AREAS USING DIGITAL NETWORK TRANSFER OF INFORMATION

BACKGROUND OF THE INVENTION

This invention relates in general to the transfer of information in digital networks and more specifically to a system using a digital network to transfer information to destination devices for the purpose of updating portions of displays associated with the devices.

Digital networks have become indispensable for transferring information in many aspects of life. Networks such as the Internet, corporate or campus intranets, local area networks (LANs), small-office/home-office (SOHO) networks, consumer electronics networks such as Firewire, Universal Synchronous Bus (USB), etc., are all examples of types of digital networks used in commerce, business, education and entertainment.

Typically, the information provided by these networks is presented to one or more human "users" in a well-known standardized format referred to as the World Wide Web (WWW) or, simply, "web." This format includes definitional languages such as hypertext markup language (HTML), extensible markup language (XML), Java, etc. Other associated standards govern the storage, addressing and transfer of web pages and referencing among web pages. The popularity of the web standard has resulted in millions of viewable web page documents that are compatible with each other and that can be viewed and manipulated in standard ways.

Web pages include displayable (or otherwise presentable) "objects" such as text, images, video, audio and hyperlinks. Hyperlinks allow a web page to reference places within the web page or to reference another web page. Multiple, related, web pages form a web page document that a user can navigate by using a keyboard, mouse, or other user input device. The web page standard has been adopted across many different viewing devices. Such devices include personal computers, handheld devices, cell phones, web-enabled televisions, etc. These devices are collectively referred to as user, or "client," devices or platforms. Wide support has resulted in tremendous popularity of the web.

However, a problem arises because the web page format is a "static" format based on a single "connection" to a web page source, or server. Once a page has been obtained, the connection to the web page server is closed so that a subsequent request for a page from the same server (or any server) requires another request for, and connection to, the server. This approach makes sense since web pages are typically requested by human selection and human selection is extremely slow when compared to computer server, digital network and other electronic device, speeds. Also, once a page is obtained by a user the user usually spends a considerable amount of time reading the page before requesting another page or making a selection.

In some applications, a more dynamic type of information updating is needed. For example, where a web page includes stock quotes it may be desirable to provide updated quotes every so often without requiring the user to request another web page or make a selection. Certain types of web page content, such as a song, radio show, phone conversation, text "chat," full-motion video, etc., require that information embedded in the web page be constantly updated, or streamed. A traditional, static, web page is ill-equipped to handle such updating.

Recent enhancements to web page definitional languages have introduced various mechanisms to improve a web page's ability to handle updating web page information. One approach uses so-called "server push" or "client pull" to allow a web page designer to specify that a web page is to be updated automatically – for example, every few seconds. The server push approach maintains a constant connection between the server and client so that the server can "push," or force, a transfer of a web page without requiring user selection. The client pull approach allows a request to automatically be made at the user's client device to request a page from the server independently of user selection. Although both of these approaches provide a web page author with a way to automatically update information, a drawback of these approaches is that the entire page must be refreshed. This results in a large overhead of additional information that must be transferred to achieve the update. Although recent refinements allow in-line images to be updated without updating the entire page, these refinements are still limiting and are helpful mostly in embedded streaming video window applications.

Another approach uses web page "frames" to present information in addition to already-displayed information in a web page. The information in a frame can be selectively updated without redundantly obtaining a copy of the information in a web page that is outside of the frame. However, the appearance of a frame usually obscures the

currently-viewed web page, or requires that the currently-viewed web page be moved and decreased in size. Thus, some of the information in the web page is obscured or removed from view. Also, frames have their own controls for allowing a user to move, resize or cancel the frame. Such controls and other features of frames take up valuable screen "real estate" or displayable area. This is especially crucial in small display devices, such as handheld computers.

Yet other approaches include using scripts on the client side, or using the HTML "Refresh" tag.

Scripts allow a web page author to include executable code as part of the web page. Such code is commonly written in Java and is interpreted by a web page display process such as a web page browser. Although this approach allows the author almost unlimited ability to control the presentation, updating, animation and other aspects of the web page, this approach is very labor intensive. Typically this task is relegated to a programmer who must work in association with a web page designer. Scripts must follow the tedious programming design cycle of designing the pages, writing code, testing the code and repeating the steps until the desired result is achieved. Each web page usually requires a custom script so that writing the scripts is done on a piece-by-piece basis. Scripts are rarely reusable for different pages without modification.

Finally, the HTML Refresh tag allows a web page author to designate an object on a web page as an updateable object.

SUMMARY OF THE INVENTION

A system for updating regions of a web page display.

In one embodiment the invention provides a method for designating areas of a web page for later updating. The method executed by a processor coupled to a user input device and a display. The method operating on a web page having a web page definition. The method comprising steps of displaying at least a portion of a web page to be modified on the display screen; accepting signals from the user input device to designate an area for later updating; accepting signals from the user input device to indicate the mode of later updating; and modifying the web page definition to include an indication of the area to be later updated and the mode of later updating.

In another embodiment, the invention provides a system for updating areas in a computer display. Where the computer display is coupled to a digital network. The system includes an update streamer coupled to the network for receiving update data from a data feed and for transferring the received update data to the network; and a client process for receiving the update data and for using the update data to update areas in the computer display.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates the system of the present invention;

Fig. 2A illustrates a web page display as would be displayed on a client platform;

Fig. 3 illustrates a preferred embodiment of the director console;

Fig. 4A shows a web page display that includes several live objects;

Fig. 4B shows a multiple function window in a voting mode; and

Fig. 4C illustrates a director console display.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

The present invention is incorporated in a system and products to be manufactured and deployed by Bang Networks. Details of the system can be found in documents filed with this specification as follows:

1. Monitoring of the Bang Network, Technical Specification (6 pages, revised 11/14/00);
2. Specification for Implementing Object Differentiation ("diff'ing") into the Data Pump (1 page);
3. Functional Specification for Gateway Fault Tolerance (6 pages, revised 11/06/00 by Joseph Benavidez);
4. Monitoring of the Bang Network, Design Proposal (8 pages, revised 11/14/00);
5. Bang Message Protocol (16 pages);
6. Network Safeguards, Possible Implementation Strategies (10 pages);
7. Network Safeguards, Technical Specification (4 pages, revised 12/18/00);
8. Proposal for Financial Data Stream (4 pages);

9. Bang Message Protocol (16 pages); and

10. Patent Addendum (12 pages).

Where a conflict arises between this specification and any of the above documents, the above documents are controlling.

System Overview

Fig. 1 illustrates the system of the present invention.

In Fig. 1, system 100 includes several components that can be functionally organized into developer, network and client components.

The developer components are used to create a web page definition that includes updateable areas, specify how the areas are to be updated, and provide a source for update information used to update the areas. Director's console 102 includes hardware and processes for creating web pages, or modifying existing web pages, with updateable areas. Update streamer 106 is used to handle receipt, storage and transfer of update information that is used to generate new displays for the updateable areas. Data sources, or "feeds," to the update streamer can be from another computer system, streaming media (e.g., digital audio or video), etc. A preferred embodiment of the invention uses a configuration applications programming interface (API) 104 to allow a director, or other person or process, to control update server 106, or to write software to control the update server.

Network components include Broadcast Gateway 110 for broadcasting update information to nodes and clients. Broadcast Gateway 110 also authenticates update information and verifies that the update information is transferred on time or on cue. Update information is transferred to nodes such as nodes 112, 114 and 116 for further distribution to client processes. The arrangement of broadcast gateway and nodes is repeated throughout a large network, such as the Internet, as needed to handle traffic demands. Such an arrangement is referred to as a "cluster." Other components in a cluster can include load balancer 118, log server 120 and WWW server 122.

Client components, such as client platform 128, client process 124 and web page display 126 are typically local to an end user viewing a web page. A web page has a definition that describes the contents and layout of the web page, and a displayable version of the web page that is generated from the web page definition. Any device that can use a web page definition to display the web page, or portion of the web page, can be used as the client platform 128 in conjunction with the system of the present invention. Such devices *include a*

personal computer, handheld computer, web-enabled television, cell phone or other device. For purposes of this application, both the web page definition, and its displayed form (or portion thereof) are referred to as the "web page". In cases where the distinction between a displayed web page and the web page definition is not explicitly made the distinction should be apparent from the context of use of the term "web page."

Client process 124 handles making requests for update information, receiving the update information and using the update information to change an area of web page display 126.

Each of the component categories of client, network and developer, is next discussed in turn.

Client Components

Fig. 2A illustrates a web page display as would be displayed on a client platform 128 of Fig. 1. The millions of web pages accessible from the Internet, or other networks, vary widely in their content. However, most web pages include one or more objects such as text, images, video, audio and hyperlinks. The present invention allows any one of these objects to be designated as an updateable, or "live," object, so that such an object can be selectively changed on a user's display. The mode of updating can be real-time streaming, by timed intervals, or based on an event either local to, or remote from, the user's client platform. Other updating modes are possible.

As is well known in the art, the hypertext markup language (HTML) definition of Fig. 2A can be generated by a human web-page author. The definition is stored on a server computer, or other device, generally coupled to a network such as the Internet, a campus or corporate intranet, local-area network, etc. The definition is then available for loading into a client platform for display.

The web-page definition includes "tags" or other mechanisms that are used to define the content and layout of the web page. Tags are used in pairs with a first tag used to show the start of an object and a second tag used to show the end of the object. The start tag is a text word or symbol surrounded by angle brackets such as <FONT>. The ending tag is typically the same text word or symbol in angle brackets but preceded with a forward slash such as </FONT>.

A preferred embodiment of the invention uses a unique tag to designate an object as a live object that can be updated according to the methods of the invention. Table I,

below, shows the type of tag used to designate an object as a live object where the web page definition is written in HTML.

<DIV class=Live> Live object goes here. </DIV>

TABLE I

In Table I, the tag "<DIV class=Live>" is the start tag for the live object. In the example, the live object is merely the text "Live object goes here." However, it should be apparent that any type of object supported by the HTML language definition such as images, audio, hypertext, etc., can be used. The ending tag "</DIV>" marks the end of the object.

A similar approach can be used in an extensible markup language (XML) definition as shown in Table II.

<LIVE> Live object goes here. </LIVE>

TABLE II

In Fig. 2A, web page display 202 is displayed on hardware display device 204. A typical display device can be a monitor, or cathode ray tube; liquid crystal display, plasma display, television screen, etc. Typically the display is driven by a processing device such as an integrated circuit central processing unit (CPU) chip. The web page is usually displayed within a browser window such as 206 which, in turn, can operate from within an operating system window (not shown). Each of these windows may require a portion of the screen display to provide accessible controls to the user. Additionally, the device, itself, may reserve, or use, a portion of the display for its own purposes, such as to provide the user with a way to check battery life in the device. In a preferred embodiment, the invention only acts on the web page display and not the other portions of the display. However, other embodiments may make use of these other portions of the display such as by using the client

process to communicate with the browser, operating system, or device firmware to allow updating of display areas controlled by these processes or the device.

In Fig. 2A, web page display 202 includes displayable objects. Examples of displayable objects are title 210, image 212, icon 214, text 216 (which potentially includes many sub-objects such as paragraphs, words and characters), table 218 and hyperlink 220. Typically, such a web page is static. That is, the objects will not significantly change unless the user requests a different web page. In many cases the entire web page will not fit within a display area devoted to the web page display. The user can scroll horizontally and vertically, as is known in the art, to bring hidden portions of the web page into view.

The browser handles displaying objects as they come into view. In a preferred embodiment, the client process continually receives all update information for all objects that are live whether the specific object is visible or not. Other embodiments can use a client process that works with the browser to inhibit the transfer of update information for objects that are on the current page but which are not visible.

Note that many more objects exist in a web page than the displayable objects. Page properties (e.g., columnar layout, background), text characteristics (e.g., font, size, blinking) and other properties are specified in the web page definition but are not, themselves, visible in the web page display. These properties can be seen in the web page definition. The present invention allows updating of such properties in a similar manner to updating the displayable objects. The ability to update properties has distinct advantages. For example, text can be made larger, or be made to flash, if properties are changed. Such a change can bring a user's attention to text that has been updated – such as a stock quote where the stock price has jumped by a sudden amount.

Note that although the present invention is discussed with respect to particular web page definition languages and specific mechanisms within those languages, that any manner of designating live objects in any suitable web page definition language is adaptable for use with the present invention. Additionally, the present invention can be applied to visual, or other media, presentations that are not based on web pages. For example, it is possible to apply aspects of the invention to displays generated by applications such as spreadsheet programs, word processors, web browsers, etc. Aspects of the present invention can be applied to application programs or operating systems in general, or even to "dumb" terminals or consumer electronic devices that act only to display received information apart from using the web page protocols. For example, approaches for the network delivery of update information, the director's console interface and associated *components for*

designating objects or areas for update; and the APIs used to enable efficient update information delivery and control, are each susceptible for use in various applications. In general, aspects of the present invention can be applied to any type of information used to create a display or other presentation to an end us

In Fig. 1, Client process 124 typically resides on, and is executed by, the client platform. Client process 124 receives update information from the digital network and uses the update information to update objects or areas on the display device (or other media presentation device). In a preferred embodiment, client process 124 is an ActiveX component, applet or plug-in to a web browser such as Netscape's Navigator or Microsoft's Internet Explorer. One embodiment also uses secured transfer of information such as via encryption in the browser's or operating system's Secure Socket Layer (SSL) or via digital signatures that can be supported in manners that are known in the art.

Another approach to client process 124 includes using an HTML hidden frame to execute a Meta Refresh tag. In general, any manner of identifying an identified object or area from the update information, and subsequently using the update information to update the display is suitable for use with the present invention. A web page that is updated with information according to the present invention is referred to as a "live" page.

Using the approach of the preferred embodiment, the client process is received at a client platform as part of the web page definition . The client process is provided with the address of a server (discussed below) to obtain update information for the live page. The client process is also informed of the objects in the current page that will be updated and the mode of updating – that is, whether to request updating every few seconds, upon user input, etc. Note that other embodiments may not require a "request" from the client process but can, instead, send the information without a client process request.

The client process scans the tags in the web page definition and keeps a list of live objects in the page. When it is time for updating each object, each object is identified using the Document Object Module as is known in the art. Thus, for each object to be updated, the client process receives a string that includes the object name and string identifier (ID) to use to identify the object in the list of live objects. The update information is provided in association with the object name and string ID.

As mentioned above, although the invention is discussed primarily with respect to updating objects within a web page, other embodiments of the invention can allow updating of arbitrary area portions of the displayed web page. For example, a 2" by 2" square area can be designated as an updateable area. Such an area can be devoted to a single

object type such as a video window. Alternatively, the area can encompass multiple objects of different types, or portions of an object or objects.

For example, Fig. 2A shows updated information in area 222. The live area includes an image, two icons and a portion of text. Other elements of the area can be modified also, such as the background color, so that changes can be more readily be brought to the attention of the user – or for other purposes.

Network Components:

The network components act to efficiently distribute update information to the potentially millions of live web pages that can be active at any one time on a network such as the Internet. The design of such a downstream distribution is critical so that the network is not overloaded and so that update information is available on-time, as needed, by each client process.

The network components are organized as "clusters." Each cluster is designed to handle a geographic region of client processes. Naturally, given the amorphous design of the Internet, such divisioning is approximate, at best. However, the changing of allocation of clusters to regions, addition of clusters to regions, etc., can be an ongoing process as traffic needs are assessed and bottlenecks identified. The use of clusters also minimizes the number of replications of live data which, e.g., in the case of video information, can greatly reduce the amount of overall Internet traffic due to live updating.

Each cluster is provided with a broadcast gateway such as broadcast gateway 110 of Fig. 1. The broadcast gateway acts to monitor the receipt of update information for the cluster. The broadcast gateway makes requests of upstream delivery components such as update streamer 106, discussed below. The broadcast gateway also authenticates the update information and broadcasts the update information to nodes within the cluster.

Nodes within the cluster; such as 112, 114 and 116; keep connections with their respective client processes open so that the transfer of live information is more efficient. Each node is typically a server computer with multiple processes, or threads, executing to keep track of many different client process destinations for information received from the broadcast gateway. Each node is aware of objects, or areas, on each page of their associated client processes that may require updating. The nodes maintain a hash table of client processes and objects. Update information that arrives from the broadcast gateway includes an ID that is used to lookup the specific object or area in the hash table. If the object is

present in the table then the update information is sent along to the respective client process. In a preferred embodiment, an open channel is maintained for each client process.

Director's Console and Update Streaming

Fig. 1 shows director console 102 which is used to create, or modify, a web page definition so that one or more objects or areas in the web page display are updateable.

A preferred embodiment of the director console is illustrated in Fig. 3.

Fig. 3 shows web page display area 250 for displaying a web page being authored or modified. In Fig. 3, the web page of Fig. 2A is shown in display area 250. The author selects an object or area by, e.g., pointing and clicking with a mouse to select an object, or by clicking and dragging to define an area. Control windows, such as control window 252, allow the author to designate the type of updating, the source of update information and other features.

Types of updating include IMMEDIATE, DELAYED, INTERVAL, USER INITIATED, and EVENT. Any manner of updating is possible. Update sources include predefined channels.

A preferred embodiment of the invention uses channels to organize the delivery and distribution of update information. Each channel is associated with a content source and update type. A typical channel may be text information such as stock quotes, streaming news commentary, winning lottery numbers, etc. Each channel is associated with a subscriber. A subscriber can be an individual or company that desires live updateable content on a web page. A primary company is an entity that provides the facilities for including live updateable content to the subscriber's web page.

In a first revenue model, a primary company such as Bang Networks provides the downstream infrastructure of network and client components, director's console tools and update streamer servers and APIs (discussed below). Each subscriber can pay the primary company for the use of such infrastructure and authoring tools. Payment can be based on the amount of update information data that is transferred per time period (e.g., per month, day, etc.).

Other revenue models are possible as where the primary company includes advertising in the updateable information so that other sponsor companies who benefit from the advertising make payments to the primary company in place of, or in addition to, the payments made by the subscriber companies. Such advertising can be displayed

intermittently in space that is designated as live display space in the subscribers' web pages – or can be displayed anywhere else on the display device.

Still other revenue models are possible. For example, the software discussed herein can be licensed. The client process can be licensed to web browser manufacturers, operating system manufacturers, platform or device manufacturers, etc. The APIs, discussed below, can be licensed or sold to authors. Royalties can be collected on the number of API installations, for version updates, etc.

Fig. 1 shows update streamer 106 used to receive update information from an external source, to store information for later transfer or to originate information. For example, a subscriber can provide a data feed in the form of a video program to the update streamer. In this case, update streamer 106 represents an entire facility for receiving, digitizing, compressing, storing (even for very brief amounts of time) and sending the information. Another example is where a subscriber uploads text information to be used to update objects. The text information can be in a file in a directory in update streamer 106. The text information is then accessed according to other properties of a channel using the information such as the frequency of updating, how much text to transfer for each update, etc.

Update streamer 106 sends update information, as needed, to broadcast gateway 110, discussed above.

Either the primary company or a subscriber can control many aspects of the update information transfers and streaming. Control is provided by means of APIs which allow customization of the updating mechanisms.

Next, Figs. 4A-C are discussed to illustrate use of the director's console and APIs to produce a web page with live objects.

Fig. 4A shows a web page display that includes several live objects, or areas. The large window at 300 is the main picture view of a video feed such as a sports event. Multiple function window 302 has different functions selectable by the user, as discussed below. Windows 304, 306 and 308 include live video, audio and text, respectively. Each of these windows includes live objects selectable, or controllable, by the user. For example, the user can select one of the video feeds represented by the miniature, or "thumbnail," icons in video window 304. The thumbnail images are updated intermittently to indicate the camera angle and subject area of the event. Once selected, large window 300, is used to display the selected feed. Similarly, different audio and text presentations can be selected in windows 306 and 308, respectively.

Multiple function window 302 performs different functions such as allowing the user to select replays of intervals of past feeds (such as a replay of an "out" in a baseball game), voting on trivia or entertaining topics, purchasing goods, responding to surveys, etc. In Fig. 4A, multiple function window 302 is shown in the replay mode where the user can select a thumbnail representing a video portion that can be replayed.

Fig. 4B shows multiple function window 302 in a voting mode. This mode allows the user to answer a question, or vote on a topic. In Fig. 4B, the topic for voting is the question "Who is your choice for the game's Most Valuable Player?". The user can select from one of four players' names. The user can also view real-time results of the voting so far in the chart at the right of multiple function window 302.

The creation of a vote question, or topic, is done via the director console. Fig. 4C illustrates director console display 340 including web page display 342 and control panel 344. Web page display 342 is a display of the basic web page presented to a user such as in Fig. 4A or Fig. 4B. Control panel 344 is a control mechanism for entering questions and answer choices. A human author, or director, can enter a question at 346 and can enter possible choices at 348. Selecting preview button 350 will update web page display 342 so that the director can check the presentation before going live. Selecting broadcast button 352 causes the voting window update information (i.e., the new question and choices) to be sent to client computers that are displaying the web page of Figs. 4A and 4B. Note that the update information will not be displayed at a user's display unless, or until, the user selects the voting mode of multiple function window 302.

WHAT IS CLAIMED IS:

1. A method executed by a processor for designating areas of a web page for later updating, the method executed by a processor coupled to a user input device and a display, the method operating on a web page having a web page definition, the method comprising:

displaying at least a portion of a web page to be modified on the display screen;

accepting signals from the user input device to designate an area for later updating;

accepting signals from the user input device to indicate the mode of later updating; and

modifying the web page definition to include an indication of the area to be later updated and the mode of later updating.

2 The method of claim 1, further comprising:

uploading the web page definition to a server computer for access by a user.

3. The method of claim 1, wherein the web page definition is hyper-text markup language.

4. The method of claim 1, wherein the step of accepting signals from the user interface to indicate the mode of later updating includes

indicating that the later updating will occur at specified intervals.

upon event at the client.

upon event at a server.

upon event triggered at the console

upon other event.

5. A method for providing areas in web pages to be updated, the method using a digital network including a client system for receiving a web page definition, the method comprising

storing an executable process in the client system;

storing a web page definition in the client system;

detecting a predetermined tag in the web page definition, wherein the predetermined tag indicates an area in a displayed web page to be updated;

receiving update information from the network; and

using the update information to update the area in the displayed web page.

6. A system for updating areas in a computer display, the computer display coupled to a digital network, the system comprising

an update streamer coupled to the network for receiving update data from a data feed and for transferring the received update data to the network;

a client process for receiving the update data and for using the update data to update areas in the computer display.

UPDATING DISPLAY AREAS USING DIGITAL NETWORK TRANSFER OF INFORMATION

ABSTRACT OF THE DISCLOSURE

A system for updating regions of a web page display. In one embodiment the invention provides a method for designating areas of a web page for later updating. The method executed by a processor coupled to a user input device and a display. The method operating on a web page having a web page definition. The method comprising steps of displaying at least a portion of a web page to be modified on the display screen; accepting signals from the user input device to designate an area for later updating; accepting signals from the user input device to indicate the mode of later updating; and modifying the web page definition to include an indication of the area to be later updated and the mode of later updating. In another embodiment, the invention provides a system for updating areas in a computer display. Where the computer display is coupled to a digital network. The system includes an update streamer coupled to the network for receiving update data from a data feed and for transferring the received update data to the network; and a client process for receiving the update data and for using the update data to update areas in the computer display.

SF 1132834 v2

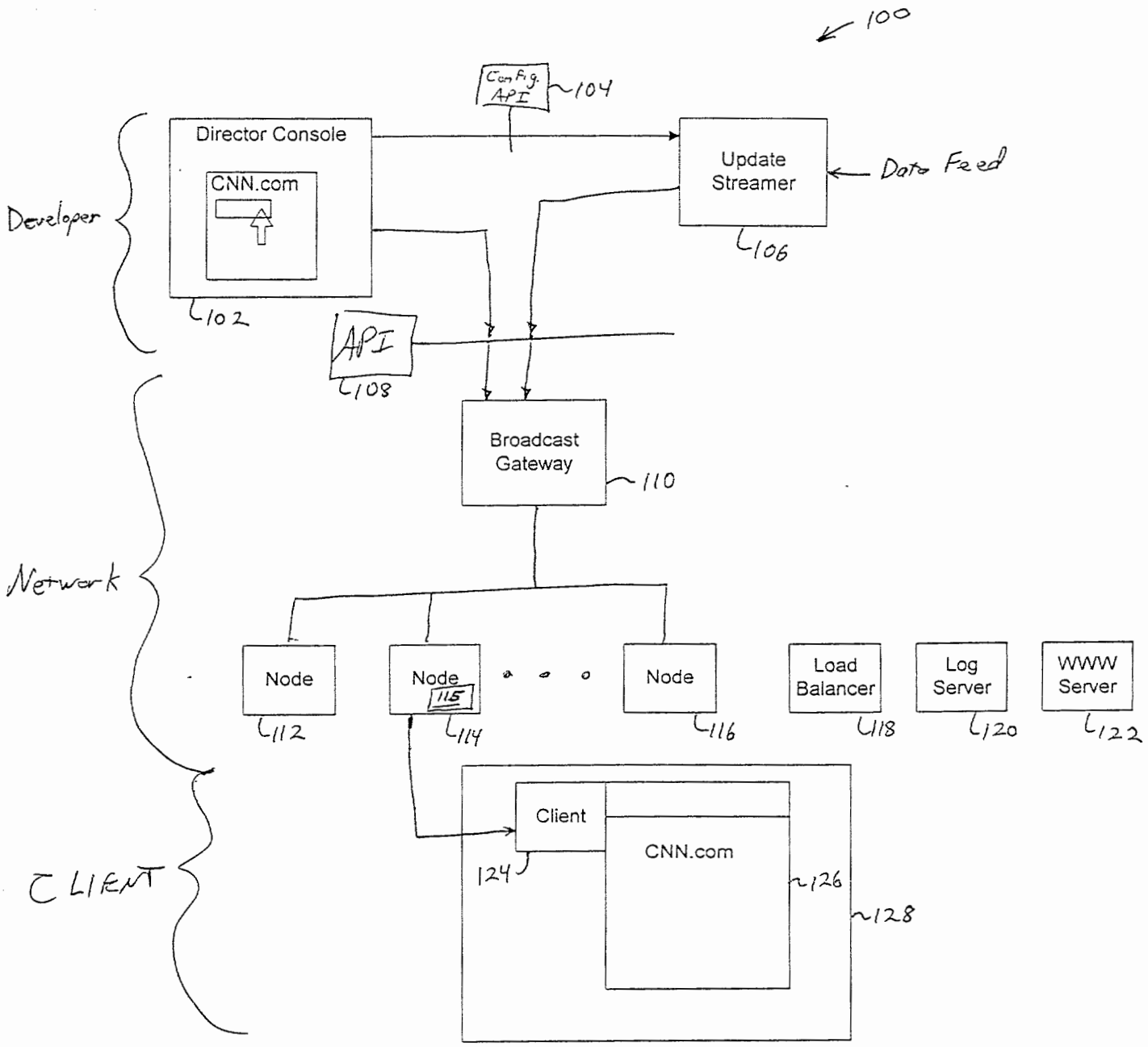

100
Config. API
104
Director Console
CNN.com
102
Update Streamer
106
Data Feed
Developer
API
108
Broadcast Gateway
110
Network
Node
112
Node
115
114
Node
116
Load Balancer
118
Log Server
120
WWW Server
122
Client
124
CNN.com
126
128
CLIENT

Fig. 1

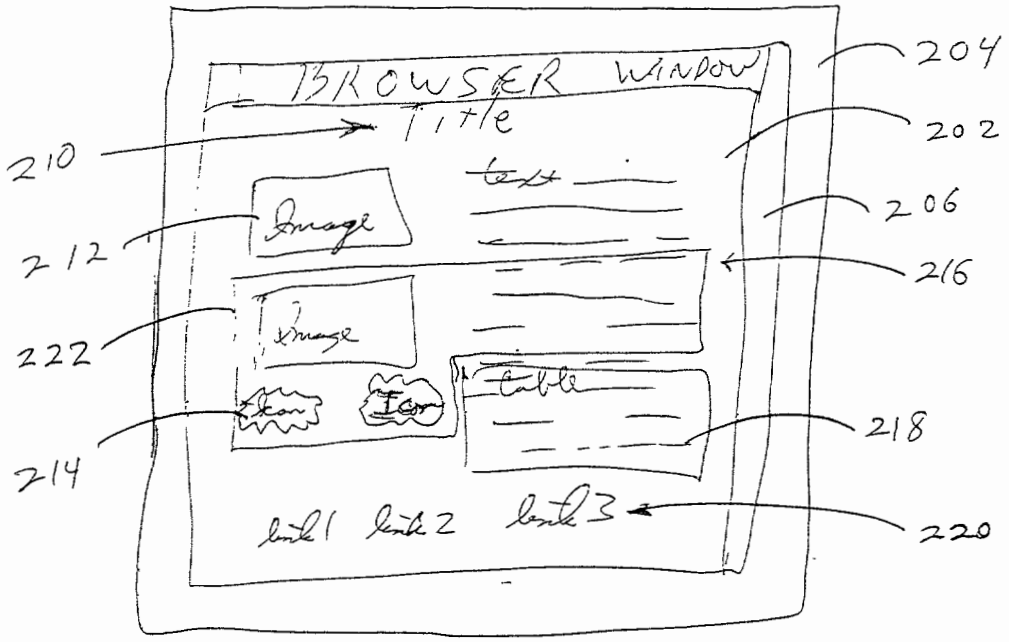

BROWSER WINDOW
Title
text
Image
Image
Icon
Icon
table
link1 link2 link3
204
202
206
216
218
220
210
212
222
214

Fig 2A

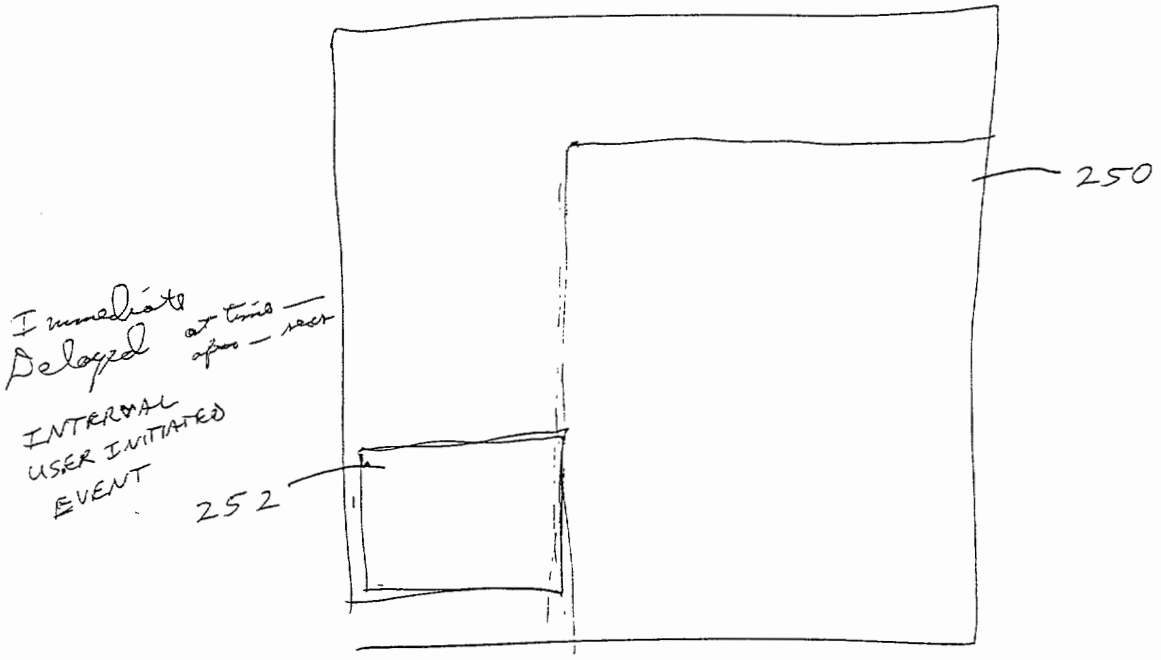

250
Immediate
Delayed
at time
INTERVAL
USER INITIATED
EVENT
252

Fig 3

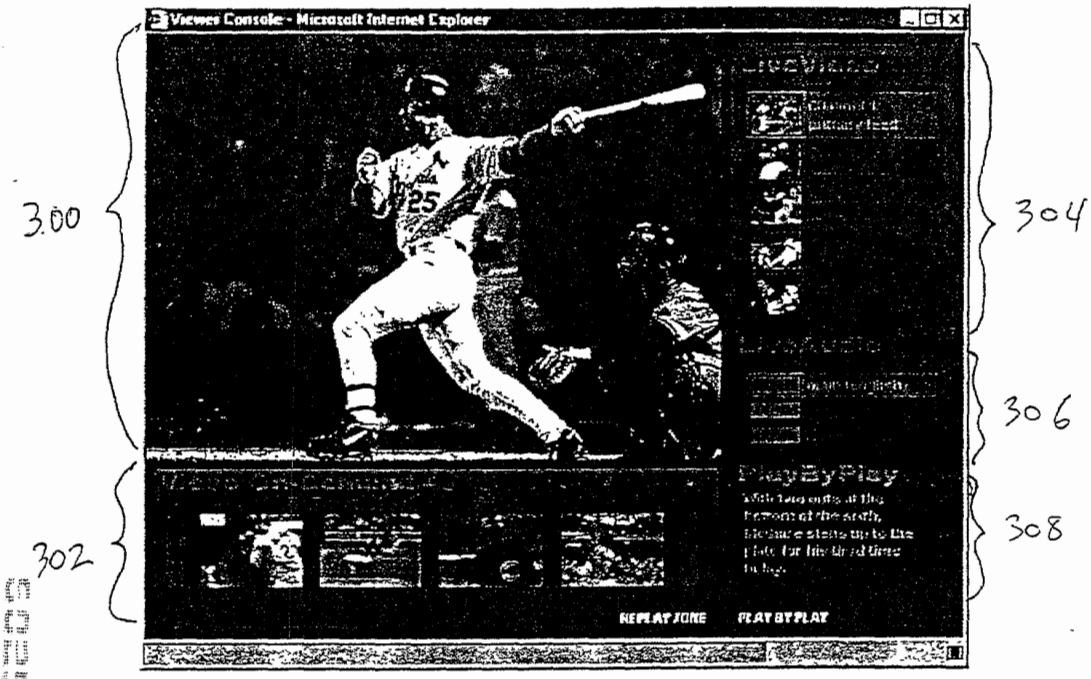

Viewer Console - Microsoft Internet Explorer
300
302
304
306
308
PlayByPlay
REPLAY ZONE
PLAY BY PLAY

Fig. 4A

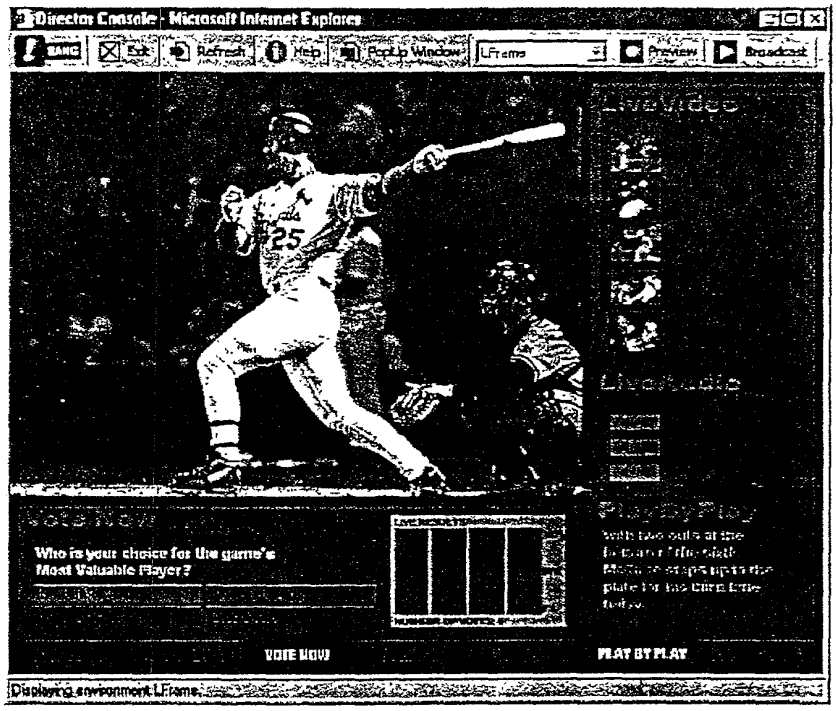

Director Console - Microsoft Internet Explorer
Exit
Refresh
Help
Pop-Up Window
LFrame
Preview
Broadcast
Who is your choice for the game's Most Valuable Player?
VOTE NOW!
PLAY BY PLAY
Displaying environment LFrame

Fig. 4B

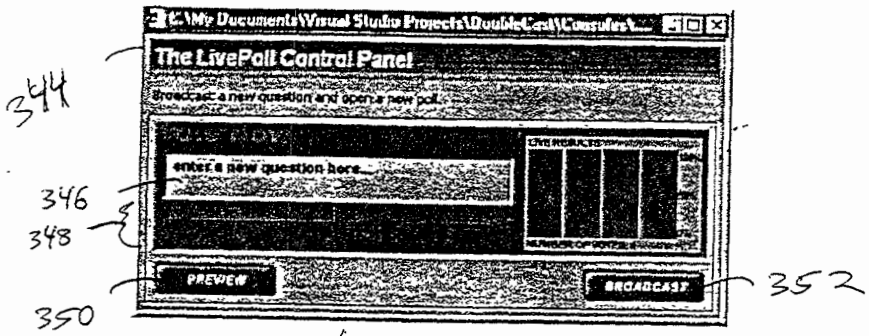
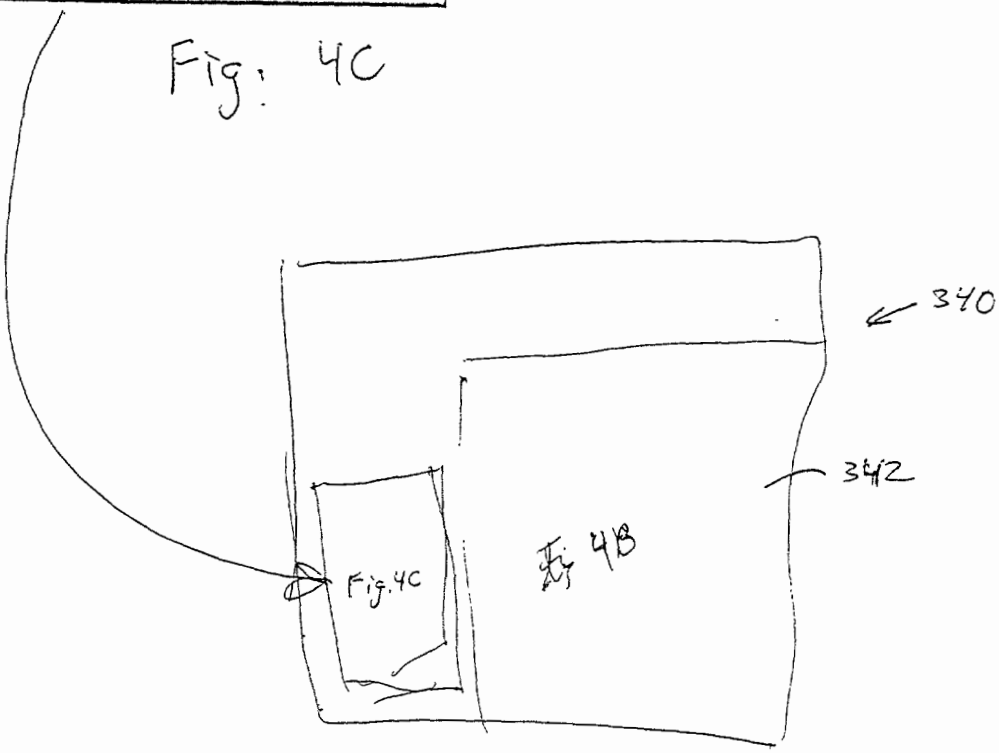

The LivePoll Control Panel
enter a new question here...
PREVIEW
BROADCAST
344
346
348
350
352

Fig. 4C


340
342
Fig.4C
Fig 4B