being played on CRT 81 as tracking is occurring and is subsequently sent on as output stream 89 from renderer filter 77 which is analogous to video stream 53 of Fig. 8. An annotation manager 85 within renderer 77 converts annotation data, input during annotation processes and the data relating to the tracked entities output from the tracking module , to metadata for more compact transmission in output stream 87. Stream 87 is a data stream containing information about the various annotations added by the author and the tracking co-ordinates of the tracked entities and is analogous to the annotation stream 62b of Figure 8. Such metadata conversion-data tables for compact transmission in output stream 87 may be stored elsewhere accessible to the CPU powering authoring station 61. User interface 83 provides considerable option and capability for entering commands to add image icons, animated graphics, following tracked objects or static or moving independently in the video in predefined manner, formatted text captions and so on.

In one embodiment, user interface 83 may pre-programmed by an author to supply the appropriate pre-selected annotations in a reactive fashion. That is, according to a specific time interval, a signal could initiate annotation inserts and so on. In other embodiments, an author may physically enter an annotation via pressing a pre-defined key on a keyboard and so on. There are many known methods for inserting annotations.

It will be apparent to one with skill in the art that other software module configurations may be used instead of those presented in this example without departing from the spirit and scope of the present invention. For example, similar functional modules may be provided to be compatible with alternate platforms such as UNIX or Macintosh.

It will also be apparent to one with skill in the art that the bulk of annotation in the form of inserted text, graphical icons, universal resource locators (URL's), interactive shapes, and so on will, in many embodiments, be at least partly associated with tracking coordinates of an image and therefore will depend on those frame by frame coordinates. For example, an interactive icon may follow a moving image entity and be visible by an end user as in case of advertisement logos for sponsors of

sportspersons in a sporting event. Text blocks and the like may take similar association. Hence, the specific content of annotations and insertion methods of such annotations may be pre-designed based on known facts about the video stream such as what image is to be tracked for what advertiser who has what URL's and so on. Execution of those annotations may be automatic according to a timed function as described above, or may be performed manually, perhaps using a macro or other designed input function.

In another embodiment, added functionality could be added to user interface 83 which allows for an author to adequately identify an image entity to be tracked so as to be enabled to place a tracking box such as box 29 of Fig. 5 over the entity at a maximally opportune instant during image motion. In this case, once the tracking box in activated, the software could be adapted to allow the author to manually track the object till such a time that the tracking box is placed more or less at the center of the object in the video. A synchronization module could be added in authoring server 63 and adapted to synchronize separate annotation streams before combining them and synchronizing them with the output video stream which is stream 53 in our example.

System for Synchronizing Data Streams Delivered Over Separate Networks

According to a preferred embodiment of the present invention, a unique synchronization system is provided and adapted to overcome unpredictable latency inherent in delivery of data-streams that are delivered over separate delivery media to end users. The method and apparatus provided and taught herein for this unique purpose is two-fold. Firstly, a video/data stream signature operation is executed after coordinate tracking and annotation operations are performed in an authoring station such as was described above with respect to authoring station 61 of Fig. 9. The signature streams are then sent to their respective broadcast and/or data-transmission systems to be sent to an end user. Secondly, a video/annotation stream capture and

synchronization operation, executed via software on customer premises equipment (CPE), must be executed at the user's end before a single combined stream may be viewed by the user.

Fig. 10 is a block diagram illustrating a signature application apparatus at the authoring end according to an embodiment of the present invention. A signature application module 91 is provided in this embodiment in the form of a software application module resident in an authoring server such as server 63 of Fig. 8. Module 91 is initiated in server 63 after tracking and annotation has been performed.

Separate data streams (video and annotation) are given frame-specific identification and marking so that they may latter be synchronized by using inserted data corresponding to the frame-specific identification.

A video stream 93 is shown entering signature module 91. Video stream 91 is analogous to stream 53 of Fig. 8. An annotation stream 95 is similarly illustrated as entering signature module 91. Annotation stream 95 is analogous to stream 55 of Fig. 8. Streams 95 and 93 are synchronous as they enter module 91. Synchronization has been achieved after image tracking and authoring in authoring server 63 of Fig. 8, as described in detail above.. Synchronization after separate broadcasting is much more complicated and is described in enabling detail below.

Referring back to Fig. 10, in this embodiment, a frame reader/counter module 97 is adapted to read video stream 93 and annotation stream 95 for the purpose of recording an association of annotation data to video-frame data using a serial count of each frame. Because annotation stream 55 of Fig. 8 was generated at the time of tracking an entity within video stream 53 of Fig. 8, each stream comprises a same number of frames constituting an entire stream length. Therefore, it is possible to count and associate individual frames in serial fashion. A number/time marker-generator module 99 generates code to represent frames in annotation stream 95 and also to represent time markers in video stream 93. Further binary numbers are generated for use in a pixel signature method described more fully below.

According to a preferred embodiment of the present invention, three separate signature methods, each method using one sequence of binary numbers described

above, are executed via signature module 91 in the course of it's function. Using three separate signatures insures that at least one of the applied signatures will successfully pass on to the end user's equipment. All three methods share a common goal, which is to record in one of two data streams to be later synchronized, at regular intervals, a marker, and information denoting which frame from the other of the two data streams should be displayed at the marker for the two streams to be properly synchronized.

In one of the three methods a number denoting frames in one of the two data streams is inserted into video blanking intervals (VBIs) of the other data stream. Although it is possible to insert such a synchronizing number in each VBI for the carrying stream, it is not necessary to do so for synchronizing purposes. Typically the synchronizing number need be inserted only once in several frames, and the fact of such a number appearing in a VBI can serve also as a marker; that is, the appearance of the number in a VBI can be taken to mean that the associated frame from the companion stream is to be displayed with the "next" frame in the carrying stream. The convention could also be applied to any frame follow g the "next" frame.

In a second method the identifying number is inserted in one or another of the horizontal blanking intervals (HBI) of a frame in the carrying stream. The particular HBI is known by convention, and more than one HBI may be used as a "belt-and-suspenders" approach. In this method the marker may be also by convention, such as the "next" frame, or some number of frames following the "next" frame.

A third method for synchronization signature according to an embodiment of the present invention involves altering pixel data in a manner to communicate a binary number to a system (described further below) at the user's end programmed to decode such a number from a carrying data stream. In this method, in the carrying data stream, the data stream values for an "agreed-upon" pixel are altered. For example, for one particular pixel in a frame, the R,G, and B values (or, in appropriate instances, the Y, U, and V values) may be arbitrarily set to zero to denote a zero bit in a binary signature, and in following frames the values for the same pixel may be set to maximum value (all 1's) to denote a binary 1 bit for the signature. In this manner, over

several frames, a binary number denoting a particular frame from the companion data stream may be inserted.

In this pixel alteration method, a marker is also needed. Again, the marker can be by convention (preferred), such as the third frame after the end of a decoded signature, or the same sort of coding may be used to insert a binary marker signature.

In the pixel insertion method, any pixel may be used by convention, but some may serve better than others. For example, in some instances jitter problems may make pixel identification relatively difficult. In a preferred embodiment, wherein a logo is used to identify a data stream, such as a network logo seen in the lower right of frames for some networks, a particular pixel in the logo may be used, which would serve to alleviate the jitter problem.

It will be apparent to the skilled artisan, giving the above teaching, that there will be a variety of ways pixel data may be altered providing a coding system for a synchronization signature. For example, the R, G, and B values may be altered differently by convention, providing three signature bits per pixel, and more than one pixel may be used; so a coded number of virtually any binary length may be provided with the data for a single frame in a video data stream.

In a preferred embodiment of the present invention, all three methods of stream signature, VBI, HBI, and pixel alteration are used. The reason for this is because it is possible that other systems downstream (toward broadcast, or in some rebroadcast) may use VBI's and HBI's to bear certain data, thus overwriting some or all data that may be inserted in blanking intervals via methods of the present invention. Similarly, a logo or other graphical alteration such as a commercial may be inserted into a video stream thus overriding a planned pixel alteration in a significant section of the video. By using all three methods at the authoring end survival of the synchronization information at the user's end is assured.

Referring back to Fig. 10, a frame writer and pixel command module 101, comprising sub-modules 101a, and 101b, uses previously generated data to insert time markers and binary numbers into frame data of at least one of the data streams (93 and 95), as well as causing alteration to one or more pixels over a series of frames to create

a serial transmission or physical marker that may be associated with frame numbers assigned to matching frames within annotation stream 95.

It will be apparent to the skilled artisan that either data stream may be the carrying stream. As a convention the primary video data stream is used as the carrying stream rather than the annotation stream.

In some embodiments, a natural screen change convention may be used for markers. For example, known software may be provided and adapted to detect screen changes wherein a majority of pixel values show significant alteration. These screen changes will happen randomly throughout the video and typically are spaced over a number of frames.

It will be apparent to one with skill in the art that module 91 may be programmed according to pre-determined criteria without departing from the spirit and scope of the present invention. Such criteria may vary according to factors such as density of annotation data in a particular annotation stream, normal frame rate of the video, whether or not it is known if there will be any further annotation before broadcasting, and so on. For example, a timing marker may be taken every 5th frame instead of every 10th frame. Screen-change marking may or may not be used. There are many variables that may be considered before applying the innovative signature methods of the present invention. Presenting the combined signatures insures that re-synchronization remains possible at the user's end as previously described.

Fig. 11 is a process flow chart illustrating logical steps for providing a synchronization signature at the authoring end according to an embodiment of the present invention. At step 103 the frames of the two streams are identified and monitored as necessary. The software may determine, for example, the scope (density) of annotation, the status of available VBI and HBI areas, the frequency of frames for time marking intervals, and so on. This step also includes counting frames for the purpose of generating annotation frame numbers for signature association purposes. In step 105, serial binary numbers are generated in separate sequences that may be used for time marking, physical marking, and frame association.

In step 107, annotation frame numbers are written into VBI and HBI areas associated with video frames as well as to the appropriate annotation frame headers. If a concerted pixel alteration method is pre-determined to be used as a marking scheme, then the pixel or pixels are selected, altered, and activated in step 109.

It will be apparent to one with skill in the art of video editing including knowledge of video-frame structure and the techniques for writing data into such video frames that there are many variations possible with regards to time marking and assigning identifying numbers to data frames wherein such numbers are also added to video frames. For example, differing frame intervals may be chosen as time markers, different bit structures may be used such as 16, 24, or 32 bit resolutions, and so on.

With reference to the stated objective of the present invention as previously described above, it was mentioned that the method of the present invention involves a second phase wherein separate data streams, marked via the conventions above, arrive at a user location after being sent via alternate mediums, such as one via cable broadcast, and one via a wide area network (WAN) delivery wherein, after receiving the streams, the user's equipment captures, re-synchronizes and combines the streams to be displayed for viewing as one annotated video stream. Such a CPE apparatus and method is provided and taught below.

Fig. 12 is a block diagram illustrating a data capture and synchronization system at the user's end according to an embodiment of the present invention. System 115 is provided and adapted to receive broadcast data-streams from varying sources and combine and synchronize the streams so the data from the two different streams may be integrally displayed as authored. System 115 has a central processing unit (CPU) 117 that has a cache memory and random access memory (RAM). System115 may be integrated with a computer or components thereof, a WEB TV or components thereof, or another type of receiving station capable of capturing and displaying broadcast video.

System 115 further comprises a signal receiving module 119, illustrated as connected to CPU 117 via bus structure 121. Bus structure 121 is the assumed connection to other illustrated modules within device 115 although an element number

does not accompany the additional connections. Module 119 is shown divided into sub-modules with each sub-module dedicated to capturing signals from a specific type of medium. In this case, there are six sub-modules that are labeled according to medium type. From top to bottom they are a modem, a satellite receiver, a TV receiver, a first optional input port (for plugging in a peripheral device), a second optional input port (for plugging in a peripheral device), and a cable receiver. The optional input ports may accept input from Video Cameras, DVD's, VCR's, and the like.

In this particular example, an annotation data stream 125 is illustrated as entering system 115 through a modem, as might be the case if an annotation data stream is sent to an end user via the Internet or other WAN. A video broadcast stream 127 is illustrated as entering system 115 through the sub-module comprising a cable receiver. Streams 125 and 127 are analogous to streams 95 and 93, respectively, as output from signature application module 91 of Fig. 10. Video stream 127 in this example is a live broadcast stream in digital form. Annotation stream 125 is delivered via a WAN which in a preferred embodiment will be the Internet. As such, stream 125 arrives as data packets which must be sorted as is well-known in the art.

System 115 further comprises a pipeline module 129 adapted to accept both streams 125 and 127 for the purpose of synchronization. Pipeline 129 is illustrated as having a time-begin mark of 0 and a time-end mark of T. The span of time allowed for buffering purposes may be almost any increment of time within reason. The inventors have has determined that a few seconds is adequate in most instances.

Video stream 127 flows trough pipeline 129 via a controllable buffer 133. Similarly annotation data stream 125 flows through pipeline 129 via controllable buffer 131. It is important to note here that either stream may arrive first to pipeline 129 and that neither stream has a predictable latency. The only constant factor between the two streams at this entry point are that they are both running at the same frame rate.

Innovative software is provided and adapted to read the time-marker and data-frame numbers in the carrying stream and to compare the indicated frame number for the opposite stream to the actual frame position relative to the carrying stream in the

pipeline. The system is adapted to adjust either data stream toward synchronization of the two streams. For example, CPU, through executing the software, may repeat frames in a pattern in either data stream to slow that stream relative to the opposite stream. The software in a preferred embodiment performs this calculation for every detected time marker in stream 127.

Buffering alteration parameters will depend upon the frequency of time markers and the extent of error detected in timing between the two data streams. For example, it is desired to produce what is termed in the art to be a *soft ramp* effect so that sudden movement or jumping of annotations related to video entities as viewed by a user does not noticeably occur. Similarly, latency factors are unpredictable regarding both streams during the entirety of their transmissions. Therefore, buffers 131 and 133 are utilized continually to synchronize streams 127 and 125 as they pass through pipeline 129. Synchronization error toward the end of pipeline 129 is small enough so that the signals may be combined via a signal combining module 135 before they are sent on as one stream into typically a video RAM of a display module 139.

A single annotated video-stream 137 is output from display module 139 to a suitable connected display monitor or screen. An input signal 141 represents user interaction with an entity in video stream 137 as it is displayed. Such a signal may trigger downloading of additional detailed information regarding the subject of interaction. Interaction signal 141 results from a mouse click or other input command such as may be initiated via a connected keyboard or the like.

It will be apparent to one with skill in the art that the architecture illustrated herein is but one example of a data stream capture and synchronization system or device that may be integrated with other equipment without departing from the spirit and scope of the present invention. In one embodiment, system 115 may be part of a computer station. In another embodiment, system 115 may be part of a set-top box used in conjunction with a TV. There are various possibilities. Moreover, there may be differing modular components installed in system 115. For example, instead of providing a dial-up modem, WAN connection may be via satellite and the modem may be wireless.

In one embodiment, a broadcast video stream without audio narration may be synchronized to a separately received audio stream. Furthermore, a prerecorded and authored video feed from a source connected to an optional input module may be synchronized with a previously stored and annotated data stream from a source connected to a second optional input module as long as the signature process was applied to both streams according to the embodiment of Fig. 10. Interaction with tracked entities and the like associated with the prerecorded streams may be sent to a participating Internet server or the like through the modem sub-module provided the system is on-line during viewing.

Fig. 13 is a Process flow chat illustrating logical steps for capturing and synchronizing separate video streams for user display and interaction according to an embodiment of the present invention. In step 143, separate data streams are captured and redirected into a synchronization pipeline such as pipeline 129 of Fig. 12. Time markers, and if applicable, screen-change markers are searched for and detected in step 145. In step 147, data-frame ID numbers are searched and compared to data-frame numbers inserted in marker frames of a video stream such as stream 127 of Fig. 12. The data may be inserted in VBI and HBI areas or as coded numbers added previously by pixel manipulation.

In step 149, a timing error is calculated with regards to data inserted in a marker frame in the video stream as matched to data in an annotation data-frame closest to the marker. The error will define an annotation frame as being n number of frame intervals ahead of or behind the target marker frame. In step 151, the stream determined to be running n number of frames ahead is buffered to reduce the error. In step 153, the process repeats (steps 145-151) for each successive marker in the video stream.

The process steps illustrated in this embodiment are intended to be exemplary only. The order and function of such process steps may vary according to differing embodiments. For example, in some embodiments wherein it may be known that no further annotation will be performed after signature operations, then only time marker intervals with VBI inserted data may be used. In another such instance, it may be

determined that only screen change marking and HBI inserted data will be used, and so on. In a preferred embodiment, the method and apparatus of the present invention is intended for a user or users that will receive the video data via broadcast, and the annotation data via a WAN, preferably the Internet. This is so that additional data obtained by a user through interaction with a tracked entity in the video may be personalized and specific to the user. In a case such as this a user would, perhaps, obtain a subscription to the service. In other embodiments, other broadcast and data delivery methods may be used.

Hypervideo and Scene Video Editor

In another aspect of the present invention, a video editor is provided for editing video streams and corresponding annotation streams and creating new video and synchronous annotation streams. The editor in a preferred embodiment comprises a software suite executable on a computer platform similar to the various platforms described above related to the coordinate tracking and annotating systems (authoring) of Fig. 1. The editor in some embodiment manipulates data streams in the well-known MPEG format, and in others in other formats. The format under which the editor performs is not limiting to the invention, and in various embodiments the system includes filters (translators) for converting data streams as need to perform its functions.

The Editor is termed by the inventors the HoTV!Studio, but will be referred to in this specification simply as the Editor. The Editor in various embodiments of the present invention may operate on computer platforms of various different types, such as, for example, a high-end PC having a connected high-resolution video monitor. As such platforms are very familiar to the skilled artisan, no drawings of such a platform are provided. Instead, descriptive drawings of displays provided in a user interface are used for describing preferred embodiments of the invention. It may be assumed that the editor platform includes typical apparatus of such a platform, such a one or more

pointer devices and a keyboard for user data entry. Platforms used as editors in embodiments of the invention may also be assumed to include or be connected to adequate data storage capacity to store a plurality of video data streams, and one or more ports for inducting such data streams.

In one embodiment of the invention, when an operator invokes the Editor, a main window 185 appears, as shown in Fig. 14. This window is a control window providing menus and icons for invoking various functions related to the Editor. It will be apparent to the skilled artisan that this control window may take a variety of different forms, and the form shown is but one such form. Window 185 in this embodiment has iconic selectors 186 for minimizing, maximizing and closing, as is well-known in the art, drop-down menus 198 providing menu-selectable functions, and a task-bar 193 with iconic selectors for the functions of the drop-down menus.

The File menu in this embodiment includes selections for *File; Tools; Window; Options; and Help.* Under *File* in this embodiment one can select to create a new video file, open an existing video file, or import a file while simultaneously transforming the file from any supported format, such as AVI, ASF, MPEG, and so on for video, or WAV and other formats for audio.

Under the *Tools* menu in this embodiment one may select functions for audio mixing, audio replacement, or for multiplexing audio and video. Under *Windows* one may select functions to determine how windows are presented, such as *Tile Horizontally; Tile Vertically; Cascade; and Close Stream.* Under Options one may select functions to set the image editor path, temporary volume path, temporary encoded files, and so forth. The *Help* menu provides functions for guiding a user through the functionality of the Editor application.

At the heart of the Editor is a display window with selectable functionality for playback and editing. Fig. 14 illustrates one such display window 187. When one creates a new file or selects a file from storage, a display window is opened for the file. An existing file will have a name and storage path, and for a new file the user will be provided a window (not shown) for naming the new file and designating the storage path. Such windows are well-known in the art. In the file window the name and path

is displayed in a title bar 197, which also displays a frame number and/or time of the video, in a display area 195.

A slider 194 is provided in this embodiment to indicate to a user the approximate position in the overall file for the frame displayed. In some embodiments one may drag the slider to force the display to a new frame position, and in some embodiments the slider is display only.

Play and editing functions are enabled in this embodiment by selectable icons in a taskbar 190. Included in the taskbar are functions for Play, Pause, Go-To-End, Go-To-Beginning, Fast Forward, and Fast Reverse. In addition there are selectable functions for marking (blocking) portions of a file for editing functions such as Cut, Copy, and Paste functions. It will be apparent to the skilled artisan that other functions may be included by appropriate icons and menus.

In a preferred embodiment of the invention a user can open and create multiple data streams, each of which will be represented by an editing window such as window 187. The user may also select the arrangement for the display of the windows. Fig. 14 shows a tiled arrangement 187, 188 of multiple windows, which may be manipulated by a user to accomplish editing of one or more of the data streams represented. A user can, for example, select portions of one stream, Copy the portion, and Paste the portion into another data stream. The user may also Cut selected portions and Paste, and mark position in a receiving data stream for a Paste to be done.

In addition to the traditional editing functions, there are also special effects that may be accomplished. With a Special Effects window (not shown), which a user may invoke from any one of the editing windows, the user may accomplish special effects on any selected frame in a data stream. The Special Effects Window provides a scrollable list of special effects that a user may choose for accomplishing the selected effect in a frame of a video stream. Among the special effects are Adding Text of a Bitmap to a frame, Merging Frames, Fade In or Out, and Alpha Merging frames. Other effects may be added to the list in other embodiments.

Given the editing functionality taught above, users are provided with a full-featured package with the Editor to create and edit video data streams and associated audio streams.

In yet another aspect of the present invention the video editor application is provided wherein visual thumbnails are widely used in editing functions. In HoTV!Studio window185, stored video files may be opened in separate windows within the editor window. Two such file windows 187 and 188 are shown in editor window 185. File window 187 is, in this example, a video clip titled backstreet.mpg, and file window 188 is for a video clip titled trailer1.mpg. The functionality of each of the file windows shown, and other such file windows, is the same. Therefore only file window will be described in detail here. It may be assumed that the others operate in a similar fashion.

File window 188 (and other such windows) has a display region 189 where the video file may be played and reviewed. A tool bar 190 has selectable icons (buttons) for controlling playback, and the button functions are intentionally crafted to resemble the familiar physical input keys on a video cassette recorder (VCR). There are, for example, from left to right, a button for Play, Pause, Stop, Rewind, and Fast Forward. The next button, shaped as an arrow is presently undefined. There are then two buttons 191 and 192 for marking a place in the video stream. Button 191 marks a beginning of a selection, and button 192 is used for marking the end of a selection. By use of these two buttons an editor may mark selections of the file for editing functions.

Editing functions in this embodiment are selectable from an editing window toolbar 193 which also displays selectable function buttons for such as file and view functions. The primary editing functions shown here are Cut (scissors), Copy (two files side-by-side), and Paste (a file on a clipboard). These are familiar icons to the skilled artisan. Functions may alternatively be selected from drop-down menus, whose titles show here, and, as is known in other Windows-based applications, the tool bars may be moved about as palettes for easy access by an editor.

In window 187 and other such windows a slide bar 194 is presented for indicating the position of the display in a video file. The slide bar is active. That is,

one may drag the slide bar with cursor control, and the video display will follow. In addition to the slide bar a time field 195 shows the relative position by time of the display in display region 189. Time field 195 in this example indicates the display is at 3 minutes and 14 seconds in an overall file time length of 3 minutes and 45 seconds. As a video file is played in region 189, the position time will be seen to change, and the overall time will not.

Fig. 15 is an illustration of an editing window 200 in a preferred embodiment of the present invention, illustrating additional functionality. Window 200 has the functionality described above for editing widow 187, and is further adapted to enable a user to manipulate entire video sequences by moving a single marker from one place in a sequence of markers into another. More detail regarding this innovative function is provided below.

Editing window 200 is adapted to work with stored digital files taken from a digital or analog video feed and, and is also adapted to simultaneously manipulate any associated annotation files that were rendered during authoring of the source feed wherein it was determined to send synchronous feeds to an end user via a separate network. Furthermore, window 200 is adapted to work on two or more separate video files wherein the final edited video contains sequences sourced from each separate file. However, in a preferred embodiment, window 200 is adapted to work with one video stream that is recorded and stored as a digital file.

Editing window 200 has a display window 202 for displaying a video stream during editing. Window 202 may also be used to view the entire video stream before, during, or after the editing process. A tool bar 204 displays various selectable functions adapted to control the video stream during the editing process, and also editing functions. For example, icons exemplifying various video-manipulation functions are presented in tool bar 204 and may be executed via selection and initiation by well-known pointer techniques, such as with a mouse device. The selectable functions are, in a preferred embodiment, the same as described above for editing window 185.

Editing window 200 in this embodiment has a first workspace 206 adapted for storing markers 206a-206d related to the video stream being edited. These markers are in a preferred embodiment are actually bitmap thumbnails. Thumbnails 206a-d are relatively low-resolution copies of an actual video frame. In this embodiment, thumbnails 206a-d are created by editor 185 to represent a first video frame or still of a sequence of frames that comprise a video-stream sequence. These markers are equivalent to thumbnails 196 in Fig. 14.

A software application known to the inventor and adapted to detect natural scene changes in a video stream is first used to identify and mark each beginning frame after a natural scene change. At each mark, a thumbnail such as thumbnails 206a-d is rendered and subsequently stored, and displayed in workspace area 206 as a sort of a book mark. In this way, all of the natural scene changes within the video data stream may be identified and made separable without the author having to review the video data. This process may be done at an accelerated video speed in order to save time. Once scene changes are marked as described above, the innovative software of the present invention provides the thumbnails and further capability regarding editing.

An exemplary order of presentation of thumbnails 206a-d as shown in workspace 206 is a serial order from the left to the right. It may be noted however, that other presentation orders or rules may be observed without departing from the spirit and scope of the present invention. Each thumbnail, such as thumbnail 206a is a first frame of all of the frames making up one video sequence beginning from a scene change (or from the first frame of the presentation, in the case of 206a). The inventors have noted that many notable highlights of a video begin with a scene change. It is to this logic that editor 185 implements scene change-detecting software. In actual practice, however, one video sequence beginning with a scene change and ending with a next scene change may contain more of or less of the particular subject of a desired highlight.

Thumbnails 206a-d are not limited in number, as it may be appreciated that one video may comprise numerous scene changes. Additional thumbnails stored in workspace 206 that are not immediately viewable may be rendered viewable by a scroll

bar technique such as is known in the art. Pagination techniques may also be employed with workspace 206 to house additional thumbnails on one serial page.

An object of the present invention in the instant embodiment is to allow quick editing of live and pre-recorded video feeds by integrating computer capabilities with video-editing processes. Therefore, capability is provided that allows traditional computer-based techniques such as opening more than one instance of editor 185, and tiling and cascading of editing windows. Other provided computer capabilities include, but are not limited to, drag and drop, copy and paste, embedding and linking, image enhancement, and so on. Such capabilities, although not illustrated in this embodiment, but described above, may be assumed to be present. Such described capabilities may be available through drop down menus, additional tool bars, keyboard options, etc.

By viewing the created thumbnails 206a-d, and others, an author/editor may quickly identify portions of the overall event that may be desirable to be included in an edited version. The editor may, for example, wish to create highlights for a halftime show from recorded footage of the first and second quarters of a football game.

The editor may select a portion (clip) of the overall video represented by the thumbnails by selecting a thumbnail, then optionally view that portion (clip) in window 202 at either normal or accelerated rate, by use of the soft buttons in toolbar 204. During this optional viewing, other functions may be done, such as annotating, tracking of objects etc. The editor may then cut, paste, delete, and so on, to pare the portion represented by the original thumbnail to a desired length and content, which may then be replaced in window 206 represented by the same or another thumbnail or identifier. In one method the new clip may be placed back in region 206, and the position where placed would insert that clip in serial order with the other portions represented by scene changes.

In another preferred embodiment, after paring a portion of an original video presentation by editing, the editor may "drop" a portion of the video in window 208, representing a new edited video presentation, wherein edited clips may be placed and arranged after selection and editing. The user may save edited clips to window 206 to

create a new video presentation. Such clips are represented in window 206 by thumbnails 206a-206h. Alternatively, the user can create an annotations-and-edit file, which could store just the "added" information, plus a file containing just the "clean video".

By repeating this process with other portions of the original video, the editor may create any desired reduced version having just the desired frames. In the second instance of the marker window, the editor may also further edit and re-arrange the newly created clips to finish and fine tune a new overall clip.

In an alternative embodiment new clips may be created by opening new instances of the editing interface 185 and dragging and dropping edited clips between instances. Also in an alternative embodiment voice annotation may be added to a new presentation.

Through further editing, frames may be deleted from a sequence and additional markers may be manually identified with a new thumbnail created thus dividing one sequence into multiple sequences. In a case such as this, each new sequence will begin with the created marker which appears as a new thumbnail in the workspace of the second editor.

In actual practice, thumbnails such as thumbnails 206a-d, are linked to the stored digital files that were created in the preliminary marking process to separate screen change sequences. As thumbnails are sorted and manipulated, stored files are thus sorted and manipulated accordingly, or alternatively, an editing file is created, which contains the new arrangement and can contain such things as annotation by itself or in a separate file. When played back on a window such as window 202 sequences appear in the newly manipulated order. Frame numbering may automatically be reassigned for the purpose of defining a new serial order of frames.

According to another embodiment of the present invention a more complex version of video editor 185 is provided and adapted to incorporate virtually all of previously described functions related to authoring such as image tracking, annotating, graphical overlay, signature processes and so on. This is accomplished via adding and integrating the various modules responsible for the added function and supplying the

input command capability to execute those functions. In this embodiment, the editing (cut and paste) operations performed on the video stream precede creation of a synchronous annotation stream (wherein tracking and annotating is performed) with the operation of providing a signature mark being a last operation before broadcast. In a preferred embodiment, however, editor 185 is an off-line editor and may stand alone from the previously mentioned conventions.

It will be apparent to one with skill in the art that editors may take an appearance other than the ones illustrated herein without departing from the spirit and scope of the present invention. Various other tool bars, drop down menus, separate work spaces (within one editor), and so on, may be present as desired. It will also be apparent to one with skill in the art that editors may be adapted to function as stand-alone editors that may be employed away from an authoring station without departing from the spirit and scope of the present invention. For example, editor 185 may be employed in a news studio wherein an arriving video feed is recorded digitally and edited before re-broadcast.

It must also be noted here that in a case of sending separate streams (one video and one annotation) via separate media, the signature process as described with reference to the section entitled **Method and Apparatus for Combining and Synchronizing Separately Sourced Video-Stream Data** must be applied after such editing with editor 185 as may be done in a news room or some other location that may be geographically separate from an authoring station.

Editor 185 may be used to edit one video stream, a video stream and a synchronous annotation stream, more than one separate video stream such as creating one sequence with highlights from several separate videos, and so on. Because editor 185 enables a user to do much editing without repeated viewing, it saves time and money.

Scene Authoring

In another aspect of the invention scene authoring is accomplished to provide thumbnails in a video presentation in much the same manner as is done for editors 185 and 200 of Figs. 14 and 15. In this aspect an automatic application finds scene changes as described above for the Editors, and thumbnails are provided comprising the first frame of each scene. An important difference is in the use of the thumbnails. In the editor applications described above the thumbnails may be selected and manipulated to manipulate the video clips they represent, allowing an editor to form new clips using all or portions of the individual video portions between scene changes. In the authoring aspect a video provided to an end user is provided with the main video displayed in a display window, as shown with window 187 in Fig. 14, and thumbnails, such as shown in Fig. 15 as a-h, are displayed proximate the video display window; in this case below the window.

Scene authoring comprises more than scene detection and presentation of thumbnails. In authoring, at the authoring station the author/editor may alter the strict division made by automatic scene detection, such as by deleting selected thumbnails and adding others, and by providing specific functionality for the thumbnails. For example, the author/editor may use editing functions as described above for combining video clips represented by thumbnails, by proscribing length of video between scene changes, and so on. The author may use any of the editing functions described, including text annotation and display of time stamps, overall frame numbers, and the like. Moreover, the thumbnails themselves may be edited in a number of ways, once they are produced and displayed. An author/editor, for example, may merge (group) two or more thumbnails, thereby merging the scenes the thumbnails represent. In this case the new clip will be represented by the thumbnail of one of the original clips. Grouped scenes may also be ungrouped into the constituent clips. One may also replace the thumbnail representing a clip or segment by a thumbnail derived from any other frame in the clip, or by a frame from another clip, or by any bitmap image

desired. Functions are also provided for selecting a saving thumbnails as separate static bitmap files.

In Fig. 15 time stamps are represented by element number 1203. In addition to time stamps and frame numbers (not shown) the author/editor may also do text annotation as described above. The text annotation in this case will most often associate with function for a thumbnail. For example, text annotation represented by element reads: "to see the last touchdown". A user may select this thumbnails to run a video clip of the last touchdown. This "last touchdown" annotation is a bare example of many annotations that may be used. In a preferred embodiment the thumbnails are enabled in authoring as hot-spots, akin to URLs, such that an end user/viewer may "click on" a thumbnail, sending an interactive signal to the providing server, which responds by providing the represented video clip.

Personalized and Interactive Ad System/Network

According to yet another aspect of the present invention, a system is provided for enabling personalized video advertisements and data to be authored and delivered, either separately from or as part of a main video data stream to end users based on selected profile information originally provided by the end user. In a subscriber environment, the system and network provides a vehicle through which the present invention may be practiced.

Fig. 16 is an overview of a video delivery network (VDN) 195 wherein personalized video advertising and delivery is practiced according to various embodiments of the present invention. VDN 195 in this example comprises three

traditionally separate networks represented herein as a satellite network 199, an Internet network 197, and a cable network 201. Satellite network 199 may be any user-subscribed satellite-based digital-television network such as are known in the art. A satellite relay system 219 is illustrated within network 199 and represents a means of signal transmission typical with such networks. Cable network 201 may be any cable-TV network such as are known in the art.

All three networks 197, 199, and 201 may be assumed to be networks through which content providers, defined for the purpose of this specification, as providers of video/audio content and delivery services, may deliver scheduled broadcast and on-demand video/audio programming to paying subscribers (end users). Various methods of delivery are possible with respect to content providers utilizing such networks.

VDN 195 also comprises, within the Internet domain, an Internet service provider (ISP) 203. An end-user premise 205 is coupled or connected to all three delivery systems in this example. User premise 205 has ability to receive video/audio programming from any one of networks 197, 199, and 201. A video display unit (VDU) 227 illustrated as part of user premise 205 is provided and adapted as a video/audio presentation system capable of playing content from any connected provider. VDU 227 may be a TV or a personal computer video display unit (PC/VDU) as are known in the art. In the system shown VDU 227 is a television.

A computerized video-capture and synchronization device 229, more commonly referred to as a set-top box, is provided and connected to VDU 227 for the purpose of capturing and synchronizing video and annotation streams from separate sources and preparing the content for suitable display on VDU 227. Set-top box 229 is analogous to video capture and synchronization device 115 of Fig. 12.

Set-top box 229 supports Internet (197) connectivity via an Internet access line 233 (typically a telephone line), which provides connection to ISP 203. Set-top box 229 in this example also supports cable-TV connectivity through access cable 235 to cable network 201. Box 229 further supports satellite-TV connectivity through a receiving dish 231 to satellite network 199. These connections are shown in their

simplest form, and actual connections including additional equipment (not shown) are well known in the art for user connections with the three networks illustrated.

ISP 203 includes in this embodiment a unique ad server 221 which executes a software suite 225 for coordinating and serving interactive and personalized ads to a user at premise 205. Integration of ad server 221 with an ISP is exemplary and convenient, and not limiting to the invention, as the ad server to be described in more detail below may be a stand-alone server, or may be hosted by another enterprise, such as a cable supplier.

Ad server 221, as is described in more detail below, may contact other servers and pull pre-stored video ads that are focused to certain user profiles. Ad server 221 may also be served such ads via other connected servers that may be maintained by content providers and others. Ad server 221 is further capable of sorting video ads and inserting or including WEB page addresses (URL'S) into such video ads.

In a preferred embodiment of the present invention, satellite network 199 and cable network 201 each maintain Internet-capable servers. These are server 217 illustrated within network 199 and server 215 illustrated within network 201. Server 217 is illustrated as having connection to a WEB server 207 within Internet network 197 via line 237. Likewise, server 215 is illustrated as having connection to a WEB server 211 within Internet network 197 via line 239. In this way, content providers utilizing networks 199 and 201 may also provide content through Internet access. Such content may be uploaded to WEB servers 207 and 211 respectively. Access lines 237 and 239 are two-way communication links providing for download of Internet-based content as well.

Web servers 207 (satellite-network server) and 211 (cable-network server) are connected to an Internet backbone 213. Also connected to backbone 213 is a WEB server 209 which is provided and adapted to provide WEB-based content such as authored videos, interactive videos, and annotation streams such as taught above in other sections of this specification. An innovative aspect of the present invention is the inter-connectivity as illustrated between networks 199, 197, and 201 wherein ISP 203 may exploit such inter-connectivity for the purpose of providing profiled video

advertisement from content providers operating in any of the three typically separate networks.

A content provider using satellite network 199 may broadcast a video to user 205 via satellite 219 and dish 231 to set-top box 229, and also may provide video advertisement (uploaded to WEB server 207 from server 217) for delivery through ISP 203 by way of ad server 221 over line 233 to set-top box 229. Similarly, a provider in network 201 may broadcast a video stream to user 205 via line 235 to set-top box 229, and also may provide video advertising uploaded to WEB server 211 from server 215 for delivery through ISP 203 via ad server 221 as described above. This interconnected and flexible system arrangement is provided and taught for reasons that will become more apparent in descriptions to follow.

According to a preferred embodiment of the present invention, ad server 221 provides all retrieval and coordination of ads with broadcast video and VOD streams. Via prior arrangement with an ISP such as ISP 203, content providers may store highly-profiled video advertisements at WEB servers such as servers 211 and 207, or at any connected repository accessible to ad server 221. Then, via push or pull techniques, such video advertising may be delivered in coordination with broadcast video or VOD streams through ISP 203 to user premise 205.

In an alternative embodiment of the invention a primary video data stream may be delivered by one network and advertisements by another, with integration still orchestrated by ad server 221. It was described above in the section entitled **System for Synchronizing Data Streams Delivered Over Separate Networks,** that annotated videos may be delivered with a main video stream delivered to a user via a cable network or such, while an annotation stream, associated with the main video, arrives via a separate network such as the Internet. VDN 195 provides for a very flexible system for delivering highly profiled video advertisements created by advertisers for a content provider that is delivering a normally scheduled broadcast of a main video to a user or users via a separate network such as via networks 199 or 201. In this case, the point where video data, annotated data, and video ads are combined and synchronized is at set-top box 229 which has added capability of combining data

steams received from different carriers, and inserting video advertisements into scheduled or tagged time-slots created in the main video.

In one embodiment, content providers may sell advertising slots to companies that provide products or services. Such scheduled time-slots may be inserted at pre-determined intervals in a broadcast video. Such an insertion technique is termed donut insertion. Donut insertion involves a provider supplying one or more pre-scheduled blank video segments or created time slots to an offered presentation for the purpose of enabling advertisers to provide video ads to be inserted therein. In this case the provider of the data stream into which ads will be inserted, by prior arrangement with the enterprise hosting the ad server, will inform the ad server enterprise in advance of the number, position, and duration of the blank segments. Control over the number, duration, etc., can be at either end, also by prior arrangement.

In a preferred (and simplified) embodiment all video feed and ad control is by server 221, via the user's Internet connection. In this example the user is shown connected to server 221 by a telephone line through modem bank 223, but this is exemplary only. The user may be connected to server 221 by a satellite link, by a telephone line, or by any means known in the art for delivering digital feed.

In the case of all feed and control through server 221, there are a number of possibilities within the scope of the invention. Video presentations may be, for example, broadcast or video-on-demand. If broadcast, the provider of the main video would provide blank intervals in the video stream in which ads may be inserted, and the number of ads and ad selection for each of the known blank intervals will be at the discretion of the enterprise hosting the ad server. The ads may be authored and stored at any convenient repository, and the ad server will recall the ads at the appropriate times and insert them in the appropriate slots.

The exact nature of the ads (degree of personalization) inserted by the ad server depends on the nature of the service. In some cases the choice of ads will be by the server, and based upon stored profiles of users logging in. In other cases, the choices will be based on preferences made by users interactively via the Internet back-channel. In practicing the invention advertisers may prepare a variety of video ads

targeting various user profiles, and the ad server will use stored profiles and user interactivity to select the appropriate ads to be inserted.

In the case of video-on demand (VOD) wherein a user orders a particular stored video presentation to be sent at a particular time, ads may be selected and inserted at any convenient time prior to sending to the user. Ads in this case may reach the maximum personalization because the video is unicast to just the selected client, and ads can be focused completely personally to the client.

There are a number of ways that the logistics may be handled in the case of VOD for ad authoring and serving. Ads can be authored in advance for individual subscribers, and personalized based on a user profile provided by the subscriber on subscription and periodically updated. Personalization will, in most cases, be based on such parameters as age group, sex, income groups and area. In a typical operation, ads will be prepared in advance, for example, for males, between 20 and 40, in professional occupations, over a threshold income, and living in urban areas. The hosting enterprise will sell time slots to advertisers in any convenient and reasonable combination, and the ads prepared can be stored at a data repository convenient to the ad server, or on a server elsewhere in the network.

When a subscriber orders a video presentation, the ad server notes the client ID, matches the ID with the user profile, consults a dynamic ad schedule (periodically updated), and determines the ads to be inserted. In this example the ad server controls, and pulls both the video presentation and the ads to be inserted from data storage or from other servers connected to the network (Internet), controlling the data streams at the ad server to start and stop each video stream at the appropriate times to place the ads, and so forth.

In an alternative embodiment the ad server does not insert ads into a video stream and send the result on to the client, but instead stores URLs (Internet addresses) for ads. When a subscriber orders a presentation the user profile is accessed based on the user ID, and the ad engine pulls the selected video stream from whatever server on the Internet provides that stream. The ad engine retrieves the

needed URLs for ads to be inserted, and inserts the URLs in the video stream as metadata by the techniques disclosed herein for such insertion.

In this embodiment the playback unit at the clients station (205) makes use of the inserted metadata to pull the relevant ad or ads from the appropriate destinations in the Internet. In many cases such ads may be provided with an annotation stream as well, so the ads themselves are interactive, and the user/client may interact with the ad or ads to accomplish a range of functions, such as displaying an advertisers page on the Internet, and accessing further product or service information, place orders and the like.

In a broadcast situation, as opposed to VOD, the logistics may differ in several respects. In this situation there are a number of options that the ad server may employ. In all cases, however, the ad server maintains control. Typically the ad server will follow a predetermined broadcast schedule, and store and organize URLs for the Internet-connected servers having the pertinent main video stream, ad streams, and annotation streams. The ad server, being the controlling entity, triggers other sources to broadcast as required by the schedule. For example, the ad server, via URL according to the broadcast schedule, triggers a video server to begin broadcasting a main video. At the same time the ad server maintains a compound profile of clients logged on to the server and to whom the broadcast is directed. Based on the profile and ad time sold, the ad server predetermines ads to be included in the broadcast.

According to the determined ad schedule, at appropriate times during the broadcast, the ad server triggers the video server to stop the main video stream, and triggers the same or a different server to begin broadcasting the appropriate ad stream. If there is an annotation stream associated with the ad stream that has to be triggered and broadcast as well. When the ad is finished the ad streams are stopped and the main video is triggered to begin again at the point it was stopped. The process continues according to the ad schedule. Again, ads may be selected and triggered according to profiles of users logged on. Also, in some embodiments, multiple ads may be streamed, and a profile set at the client may select ads from the ads streamed. Metadata may be inserted as well for full interactivity at the user station.

The embodiments of the invention relative to authoring and serving ads thus far described, are described within the framework of interactive television wherein the ad server, the video streams, the ads, and any annotation streams are all pushed or pulled from Internet-connected sources, and all sent to the user/client by Internet delivery. There are, however, many embodiments of the invention wherein video streams may be controlled and ads authored and server within the scope of the invention involving other networks than the Internet alone. Fig. 16 is meant to reflect this diversity and flexibility, by showing interconnected networks, including satellite network 199 and cable network 201. set-top box 229 is connected to all three networks and may receive digital data streams by any of the networks unilaterally or in combination. Note that all networks are interconnected through Internet connection. This allows control of content streaming from Internet-connected ad server 221.

Now, as an example, when a client at premise 205 decides to purchase a scheduled program (VOD) from a cable company, his profile may be communicated, by prior arrangement, to the cable company by ad server 221. Based on the provided profile, video ads may be selectively pulled or pushed from Internet-connected servers such as server 211 according to "best match to profile". Control remains with ad server 221, which may process its own ad schedule and perform many of the functions described above for the "Internet-only" embodiments. The same functionality may be provided through satellite network 199, and Interactive ads may be authored and served in several ways in cooperation with satellite based video providers. By the network interconnectivity shown, utilizing digital broadcast techniques, ads and annotation streams may be accessed and coordinated from virtually any source within the scope of the present invention.

In a preferred embodiment of the present invention, an ISP such as ISP 203 will be the dominant partner in a service arrangement with various content providers using various delivery networks. This will insure that user information such as may be included with a user's profile may be kept secure as his ISP will most likely retain his subscription over content providers. Content providers may be added or subtracted

from a user's capability based on user discretion. For example, user 205 may desire video services from cable network 201, but not satellite network 199, and so on.

In another embodiment of the present invention, wherein broadcasting of live events is practiced, each content provider may create interactive advertising via previously taught methods and additionally provide donut insertion intervals that may be filled by local ISP sponsored video ads.

Pre-authoring of either pre-stored or live broadcast videos may be performed at WEB-based facilities or content provider-based facilities. For example, a live football game may be broadcast from an onsite location wherein authoring is performed via an Internet-based authoring station before re-casting over various delivery networks.

A VDN network such as VDN 195 facilitates a flow of content in virtually any pre-conceived or planned direction. Such capability has not been offered or achieved in prior art. As previously described, a user may choose from which network to receive VOD content as long as such networks maintain a channel over which such content may be streamed and a working arrangement with the enterprise hosting the ad server. In this way, a user such as user 205 may add a wide variety of content providers to his service.

With regard to regularly scheduled broadcast channels such as are currently provided via cable companies and the like, advertisers already owning non-interactive or non-hyperlinked commercial slots may be solicited through consultation to restructure their advertisements based on statistics regarding supplied user profiles if there are enough equipped viewers to comprise a suitable advertising base wherein all individual profiles are known. Such combined profiles may be created for each program by comparing individual known profiles from a total number of viewers likely to be accessing programs via CPE of the present invention. Additionally such advertisers may be offered an opportunity to add certain interactive content such as URL's or the like to their video ads. In this case, ad server 221 may pull appropriate new adds based on the most updated statistical report showing current user-combined profile statistics. Under such an arrangement, ads created by different advertisers may be rotated in among different shows with profile matches retaining seniority. Charges

to advertisers may be based on a percentage of actual air time wherein the pricing for ads that better match a combined viewer profile, and thus are most often rotated in over the course of a broadcast day, is set according to actual air time after the fact.

In all aspects of the present invention, a controlling entity, such as the ad server described above, manages ad placement in video data streams, and, in preferred embodiments information from subscribers is used in selecting of ads to be included. Fig. 17 is a brief flow chart of steps in the process of the present invention. At step 241 a subscriber logs on to a service associated with an ad server enterprise. In various embodiments the user may log on to an ad management server, an ISP integrated with an ad management server, or to another service, such as a satellite or cable provider having an arrangement with the enterprise hosting an ad management server. At step 243 the ad server, being directly or indirectly notified, accesses stored user-profile data based on the user ID. At step 245 the ad server, executing its own control routines, accesses ads and optionally annotation streams, and controls integration of the ads and annotation streams with primary video stream, to accomplish personalization and interactivity according to embodiments of the invention, as described above. As also described above, the integration and interactivity of ads may be accomplished in a number of different ways

The above steps are intended to be exemplary of a wide variety of sequences which may occur under different circumstances. For example, content providers may have direct connection to ad servers instead of hosting additional network servers. There may be differences in process according to which network provides content, whether or not there are additional annotated streams that must be delivered, and so on. In a flexible VDN such as the one taught herein, there are many varying possibilities.

In preferred embodiments of the present invention the user premise 205 includes a set-top box or other computerized device capable of executing code. In many embodiments unique code is provided as a part of the service for adding functionality to the set-top box or other device used to receive and display video. Such added code comprises code, for example, to select among multiple ads that may

be served to the device, based upon user-defined profile data. The user may, for example, enter a profile including sex and age. Then in receiving and preparing for display of data streams provided, the set-top box or other device will select to display those ads targeted to the profile, and not other ads that may be sent simultaneously.. This facility can act as a selective filter in this respect.

Device 229 in various embodiments is also capable in many embodiments of synchronizing data streams received by distinct delivery paths, as described above in the section entitled: "System for Synchronizing Data Streams Delivered over Separate Networks. Main video streams, annotation streams, ad streams, and the like may therefore be delivered by separate and distinct delivery paths (different networks), and any latency may be ironed out at the user's device.

It will be apparent to one with skill in the art that a data capture and synchronization device such as device 229 may have yet additional features without departing from the spirit and scope of the present invention such as an input port for a personal computer for the purpose of inputting profile information to the unit. Moreover, multiple profiles may be kept on additional household members having different passwords and profile keys such that advertising may, in some embodiments, be specific to certain household members who purchase videos and otherwise interact with the service in embodiments of the invention.

Multiplexing Video Metadata and Use of Presentation Time Stamp

According to yet another aspect of the present invention, a method is provided for multiplexing separately-authored video metadata using a presentation time stamp (PTS) convention as a commonality in metadata packaging. A digital PTS is an integral part of substantially all digital video formats, and when dealing in digital video streams for purposes of this invention the digital PTS may be used. In the case of analog streams a PTS may be generated by any of several methods and integrated with the analog data stream for use in synchronizing annotation data with the analog stream. In the descriptions below the term PTS is meant in the more general sense covering

both digital and analog streams. Methods and apparatus for accomplishing the purposes of the invention are described in enabling detail below.

Fig. 18 is a block diagram illustrating a metadata authoring, multiplexing, and delivery system 247 according to a simplified preferred embodiment of the present invention. In practice the overall system may be considerably more diverse than shown in Fig. 18, but the simplified diagram of Fig. 18 serves to illustrate principles of the invention. System 247 illustrates three distinctly separate authoring functions in this example, each of which may comprise one or more authors and authoring stations. The fact of three authoring functions is exemplary, as there may be more than three, and of other sorts than those shown here.

The three authoring functions shown are a scene-change authoring function 251, a hyper-video authoring function 253, and an ad-authoring function 255. System 247 further encompasses a multiplexer function 259 for data merging, and a metadata transport mechanism 257 which allows various transport options for end-user delivery. Multiplexer 259 in many embodiments of the invention exists as a server with the unique multiplexing software described herein, ports for receiving data streams, both primary and annotation streams (metadata), and ports for serving the multiplexed result to either other stations in a network or directly to end users.

The delivery options include, but are not limited to, option 261 (Internet Streamed), option 263 (VBI Inserted), and option 265 (Internet Downloaded). In this aspect it is assumed that all data is delivered by a common carrier, in this example the well-known Internet network.

Multiplexed streams may be streamed over the Internet in various embodiments either multicast or WEBcast. In multicast the server streams the video, including metadata onto the Internet. In WEBcast the multiplexed data is streamed to subscribers logged onto the server. In the case of VBI inserted metadata, the stream is analog and the metadata is inserted in the video blanking intervals of the main video data stream. In this case the metadata will typically be compressed, because of the limited bandwidth aspect of VBI techniques. In the case of downloading over the

Internet data streams are stored at a WEB server, and the end user selects and downloads the video.

There are a number of options in delivery. Metadata may be inserted into digital video, for example, in a manner that equipment at the end user's station may access the metadata and display the enhancement provided by the metadata along with the primary video presentation. As has been described previously, there may be more data than an end user will use, and user characteristics may operate to select particular data, such as particular advertisements based on the end user's profile. Also, the metadata may be streamed separately from the primary video and coordinated at the user's end according to time placement information in the video streams. This coordination is at the heart of the present invention. In the case of multiplexed data streams stored for selective downloading by users, the stored stream may be a combined stream or one or more marked separate streams.

It is known to the inventors that digital video presentations typically contain a presentation time stamp (PTS) signature. This convention numbers each video frame of the video stream with a serial number. For example, the first video frame in a video is typically numbered 0 and the second frame 1, and so on. The inventors in several embodiments use this convention as a commonality for merging and coordinating metadata via a multiplexer such as multiplexer 259. Therefore when using the digital PTS for synchronization, analog video streams are converted to digital form before they are subjected to processing under system 247. In other cases a PT may be generated for an analog video stream, inserted in such as the VBI of the analog stream, and used for synchronization, in which case it is not necessary that the analog stream be converted to digital.

In a preferred embodiment, authored and merged metadata is transported across an Internet link to an end user having suitable CPE equipment adapted to decode and apply metadata so that an end user may interact with a main video through manipulation of displayed annotation as previously taught in this specification. This example is consistent with a subscriber situation wherein a user will order an annotated video from an Internet-based video server termed HoTV by the Inventors. However,

delivery of a main video by other means while obtaining metadata from the Internet is also possible as has been taught above.

Referring again to Fig. 18, a video stream 249 (main video) is provided, in this case from a satellite feed 269. This main stream can be from a stored source, another system like system 247, or may be a live feed. In some cases a timing coordination function 271 is provided to ensure that the main feed is coordinated in time with authoring stations and the multiplexer server. This function can take a number of forms, such as noting absolute time variation between stations and buffering the video feed accordingly. The video feed is provided to each authoring function 251, 253, and 255 as illustrated in this example. Authors at each station accomplish annotation during viewing of video 249. For example, scene authoring uses known SCDT techniques to mark scene changes for several purposes as described in the background section. Among these are marking for ad insertion and providing thumbnail markers as a selectable feature to an end user for allowing such as video review by portion. In this process the author may also merge scenes and edit markers. Hyper-video authoring function 253 provides object tracking coordinates, interactive regions based on such coordinates, creation of hot spots (hyper-links), and so on. Ad authoring function 255 provides metadata about ads such as which ads to pull and where they belong in the main video.

In addition to live authoring, there may be input to multiplexer 259 from one or more stored sources as represented by repository 267. This source may provide interactive ads previously prepared, and other sorts of metadata as well to become part of the ultimate feed to end users.

The present invention is not limited to the above example, as stream 249 may also be authored in serial order such as via station 251, then station 253 followed by station 255, rather than in parallel. Moreover, stream 249 may be a live stream or authoring may be performed on a pre-recorded stream. Authoring may be performed off-line or on-line. There may be more than one author at each represented function such as functions 251, 253, and 255 simultaneously working on one video stream such as stream 249. There are many possible variations in both distribution of authors,

assignment of authors, order of authoring, and so on. It should be clear as well that the entire process may be cascaded in a multitude of different ways. That is, an original process may provide, via a multiplexer server a data stream including a main video presentation and original metadata for any of the functions described, such as HoTV, ad insertion and the like. This result can go to another system where further authoring and multiplexing is done, and may be cascaded to any number of such processes, although in practice the cascading will be relatively limited to remain manageable.

In a preferred embodiment, authors working in function groups 251-253 tag created metadata with an appropriate frame number or numbers (PTS) which will indicate a frame or series of frames where resulting annotation will appear during playback of the video. This is automatic in most instances, but can also be performed manually. After authoring and tagging, the metadata progresses to multiplexing.

It is the purpose of multiplexer 259 to package all metadata for transport to an end user. Firstly, metadata is sorted and associated by PTS tag. For example, metadata from each of stations 251-253 is assembled or packaged according to a same PTS indicator or signature. Rules governing what type and how much metadata will be inserted into any one or a series of frames of a video are, of course, set-up before authoring.

After the metadata is packaged, multiplexer 259 may compress or encode the metadata according to any specific transport means. For example, if delivery will be over a high bandwidth, metadata may, optionally, not be compressed. Most often, however, bandwidth will be less than guaranteed, therefore compression of metadata will usually be performed by default. One or more known encoding techniques may be made available for differing scenarios.

Transport mechanism 257, as previously described, provides optional delivery methods illustrated via downward facing arrows 261, 263, and 265. Option 261, Internet streaming, may be performed with metadata being streamed independently while a main video arrives separately, or, a single muxed stream may be provided comprising the main video and the metadata. In option 261 the delivery of the

annotation stream may be either multicast or WEBcast. In the VBI case the metadata is typically compressed and inserted and the stream is broadcast. In another case, metadata could be downloaded independently over the Internet via option 265 and be disseminated and rendered when a user plays a previously downloaded video or a pre-recorded VOD presentation served from an Internet server. In still another case, metadata may be muxed in with a main video and then be made available for download via option 265. An analog video may be served with the metadata inserted in VBI as illustrated via option 263. Because of the wide range of architecture that may be supported in such a system, combining main and authored data at many points, it is necessary to do more than simply copy PTS from a main video stream and use it in placing metadata. Several synchronization techniques are also used. For example, at a point in such an architecture where two streams are brought together from different sources, in various embodiments of the invention the system may compare real time clocks of sources and buffer one or both streams from the separate sources to compensate for time differences. Another technique used in some embodiments is to rewrite the presentation time stamps of one stream based on the real time PTS of the other and the difference in time of the two streams. In all cases, video streams are preferably provided to separate authoring stations from a single source to avoid timing problems as much as possible.

A suitable de-multiplexing module (not shown) is provided and supported at the end user's location. Such an adaptation could be incorporated easily in a video capture and synchronization device such as device 115 of Fig. 12 which already supports annotation and main video stream buffering and synchronization as well as having dual capability of rendering analog or digital streams to an appropriate VDU. A de-multiplexing module decompresses, reads and buffers the metadata, makes use of PTS for synchronization purposes in the event that the metadata is not already muxed in to the main video, and renders the metadata for display on a suitable VDU over or alongside a main video.

It will be apparent to one with skill in the art of video authoring and broadcasting that the present invention may be practiced with live video or pre-

recorded videos without departing from the spirit and scope of the present invention. In the case of pre-recorded video, metadata is preferably sent to an end user slightly ahead of a main video to insure that all metadata is properly processed and rendered in time for presentation streaming.

If metadata associated with a pre-recorded video is to be streamed separately, metadata is preferably sent ahead of a main video and is buffered at the user's end, therefore, video delay is not required. If a video is authored live and streamed separately from the metadata, then the main video is buffered. This may cause a very slight delay in the video frame rate. Muxing a live metadata stream into a live video stream before delivery will ease buffering requirements that may be observed with separate streams.

It should also be apparent to one with skill in the art that the present invention is not limited to the domain of the Internet. For example, authoring may be performed in coordination with a broadcasting entity having suitable digital capability and connection to the Internet wherein authored metadata may be diverted to an end user via Internet connection such as via an ISP while main video content is delivered via normal cable path or perhaps via satellite. There are many possibilities.

It should further be apparent to one with skill in the art that the methods and apparatus of the present invention may be practiced in a variety of ways, and the elements may be implemented in a variety of hardware and software within the scope of the invention. Therefore, the present invention should be afforded the broadest scope and not be limited to the exemplary embodiments as taught herein. The present invention is limited only by the claims that follow.

What is claimed is:

1. An authoring system for interactive video, comprising:

a video feed providing a main video presentation stream;

two or more authoring stations coupled to the video feed providing authoring functions creating metadata for enhancing the main video stream; and

a multiplexer for coordinating authored metadata with the main video stream;

wherein the authoring stations note a presentation time stamp (PTS) of video frames or a synchronized system time and incorporate the PTS in the authored metadata for matching the metadata with the main video presentation stream.

2. The system of claim 1 further comprising a multiplexer for combining authored metadata with the main video data stream, and wherein the multiplexer places the metadata in relation to the main video data stream according to the PTS.

3. The system of claim 2 wherein the multiplexer receives multiple video streams as well as the authored metadata, and time clocks are monitored for separate stream sources and clocks are adjusted to compensate for real-time differences in sources.

4. The system of claim 3 wherein one or more of the stream sources is from a stored source.

5. The system of claim 3 wherein PTS values are rewritten in one or more streams to compensate for perceived time differences.

6. The system of claim 2 wherein the PTS-enhanced metadata is streamed over the Internet to an end user.

7. The system of claim 2 wherein the PTS-enhanced metadata is inserted into video blanking intervals (VBI) of an analog stream according to the PTS.

8. The system of claim 2 wherein the PTS-enhanced metadata is stored to be downloaded as needed by a user.

9. The system of claim 1 wherein the authoring stations include one or more of scene authoring, hyper-video authoring, and ad authoring stations.

10. The system of claim 1 wherein the main video presentation stream is an analog stream, and the analog stream is converted to a digital format before authoring and multiplexing.

11. The system of claim 1 wherein the main video presentation stream is an analog stream, and a presentation time stamp is generated and integrated with the analog stream.

12. The system of claim 2 further comprising a user system enhanced with software for rendering the main data stream and the authored metadata according to the PTS.

13. A method for coordinating authored video metadata with a main video data stream, comprising steps of:

(a) ensuring the main video data stream has a presentation time stamp (PTS);

(b) feeding the digital main video data stream to authoring stations;

(c) authoring matadata at the authoring stations; and

(d) marking the metadata with presentation time stamps (PTS) from the main digital video data stream.

14. The method of claim 13 further comprising a step for multiplexing authored metadata with the main video data stream, wherein the multiplexer places the metadata in relation to the main video data stream according to the PTS.

15. The method of claim 14 further comprising multiple sources of video fed to the multiplexer as well as metadata, and comprising a step for compensating for real-time differences between the multiple sources.

16. The method of claim 15 wherein, in the compensating step, presentation time stamps (PTS) are amended according to source time differences.

17. The method of claim 13 wherein the PTS-enhanced metadata is streamed over the Internet to an end user.

18. The method of claim 14 wherein the PTS-enhanced metadata is inserted into video blanking intervals (VBI) of an analog stream according to the PTS.

19. The method of claim 14 wherein the PTS-enhanced metadata is stored to be downloaded as needed by a user.

20. The method of claim 13 wherein the authoring stations include one or more of scene authoring, hyper-video authoring, and ad authoring stations.

21. The method of claim 13 wherein the main video presentation stream is an analog stream, and the analog stream is converted to a digital format before authoring and multiplexing.

22. The method of claim 13 further comprising a step for displaying the main video data stream and authored metadata according to PTS at a user's station

23. A video multiplexing system comprising:

inputs from video authoring stations;

an input for a main digital video data stream; and

an output to a video transport interface;

wherein the multiplexer notes presentation time stamps associated with authored metadata, and places the authored metadata relative to the main video data stream for transport to end users.

24. The multiplexing system of claim 23 comprising multiple video data stream inputs, and wherein one or more of the inputs is from a stored source.

25. The multiplexing system of claim 23 comprising multiple video data stream inputs from multiple sources, and wherein the multiplexer monitors real time clocks of the sources and uses the information to compensate one or both of the multiple streams.

26. The multiplexing system of claim 24 wherein the multiplexer compensates incoming streams by buffering one or more of the streams.

27. The system of claim 24 wherein the multiplexer compensates incoming streams by amending the presentation time stamps of one or more of the streams.

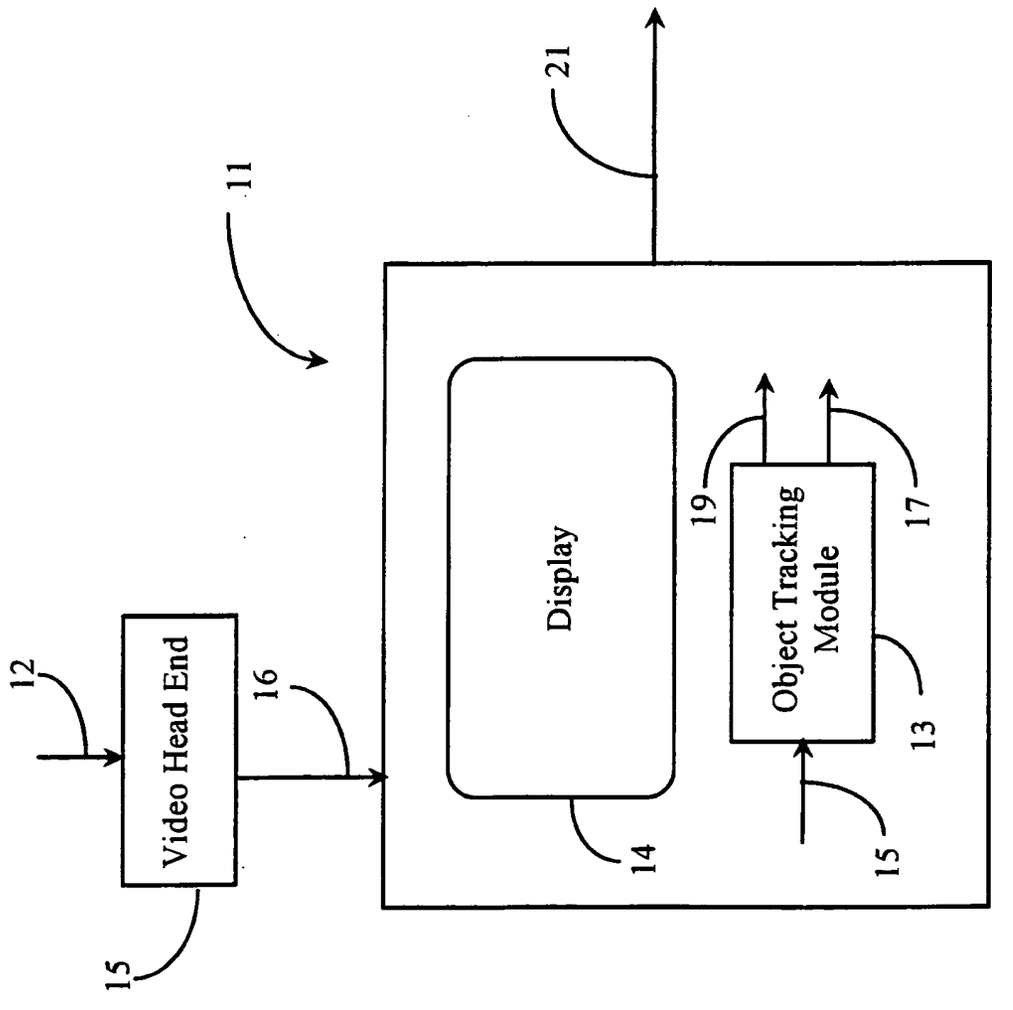

21
11
Display
19
17
Object Tracking Module
13
12
Video Head End
16
14
15
15

Fig. 1

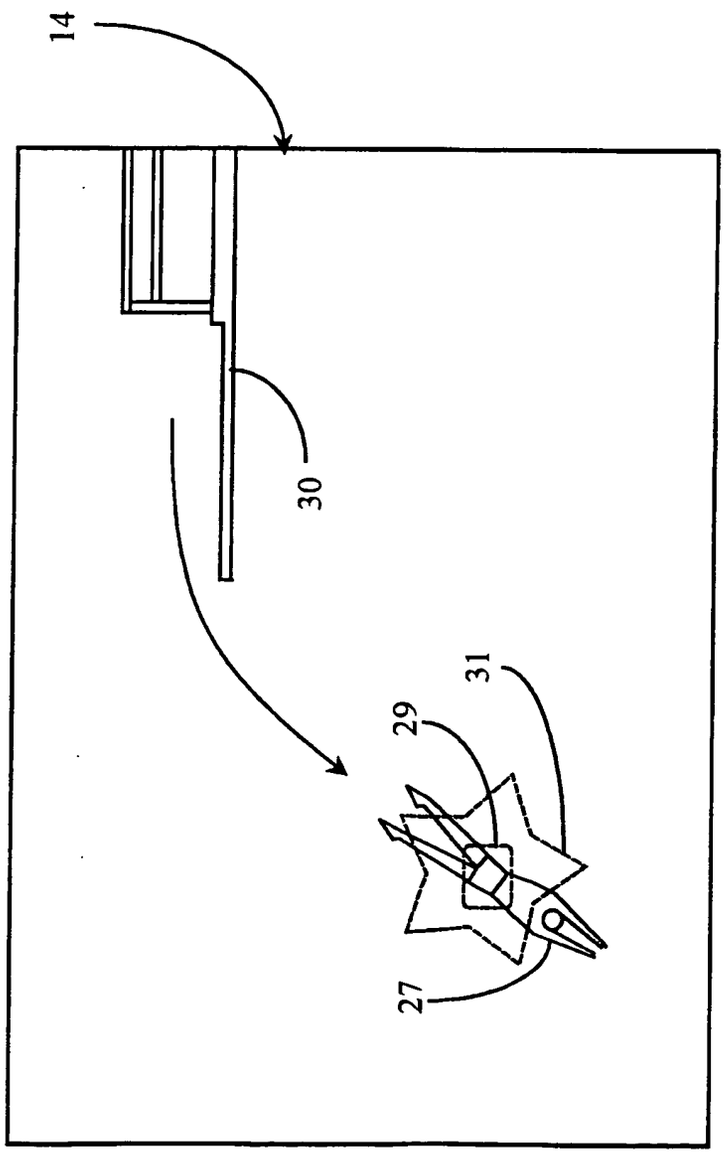


Fig. 2

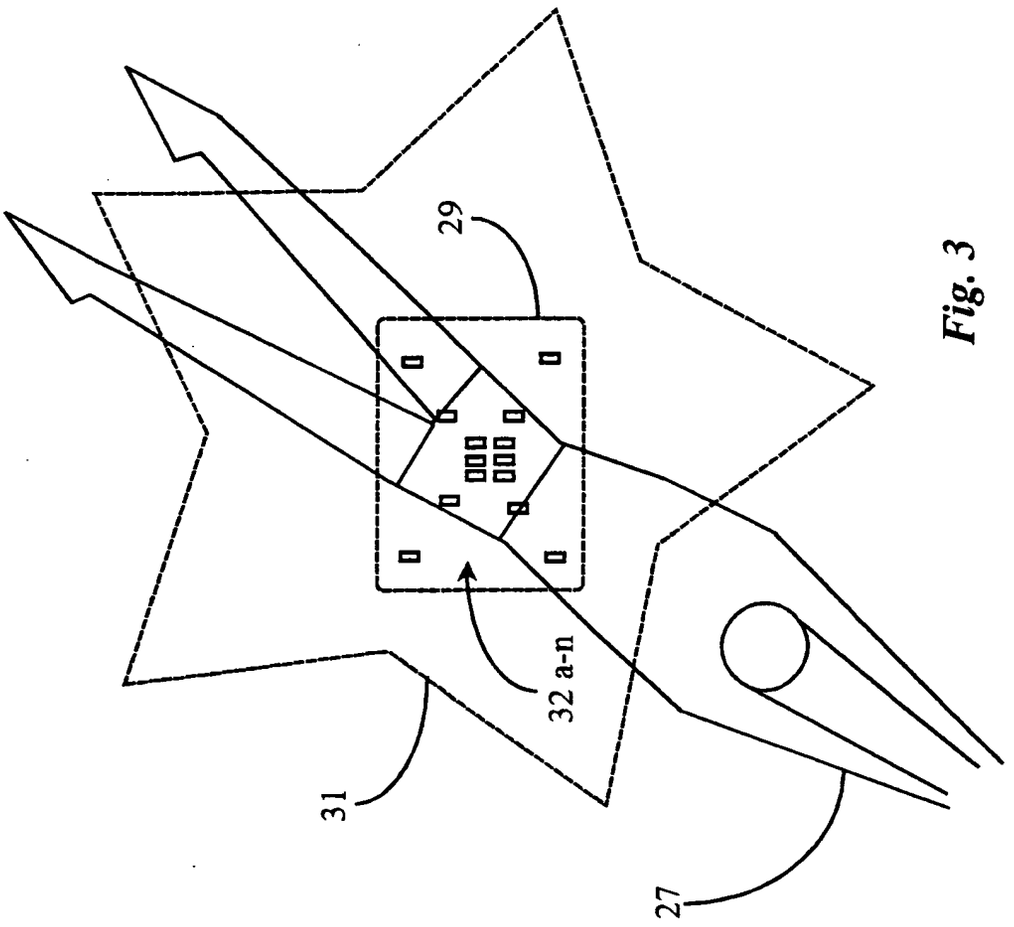
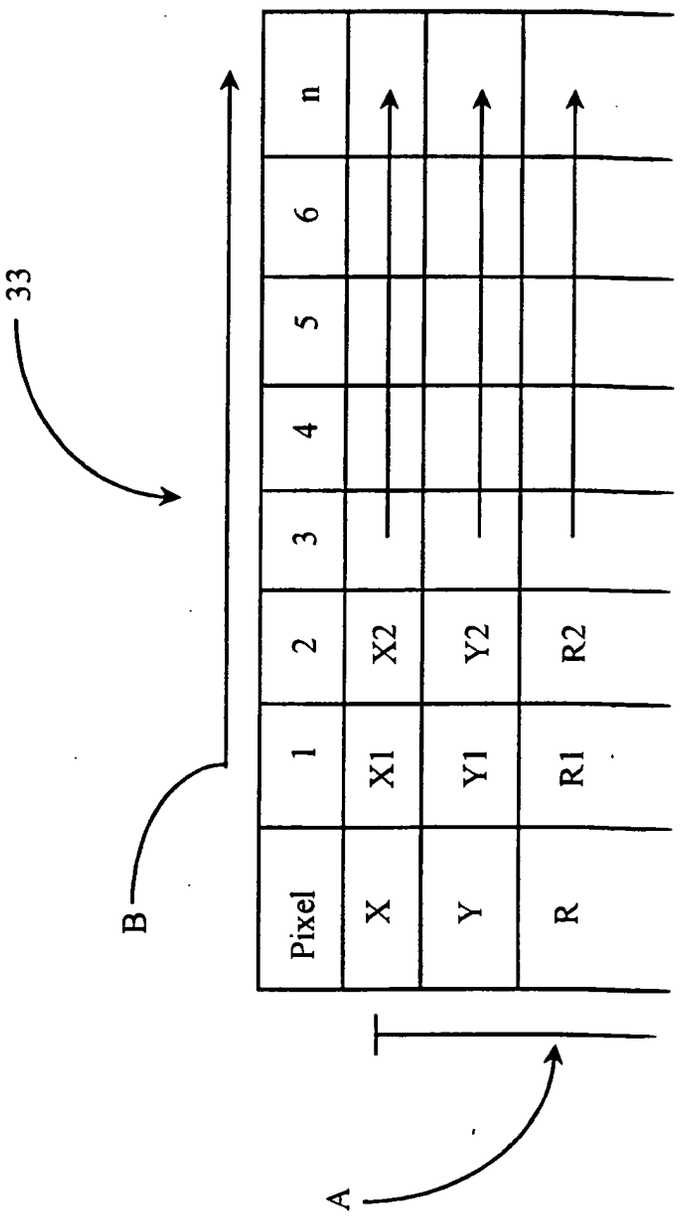


Fig. 3


33
B
A
Pixel
X
Y
R
1
X1
Y1
R1
2
X2
Y2
R2
3
4
5
6
n

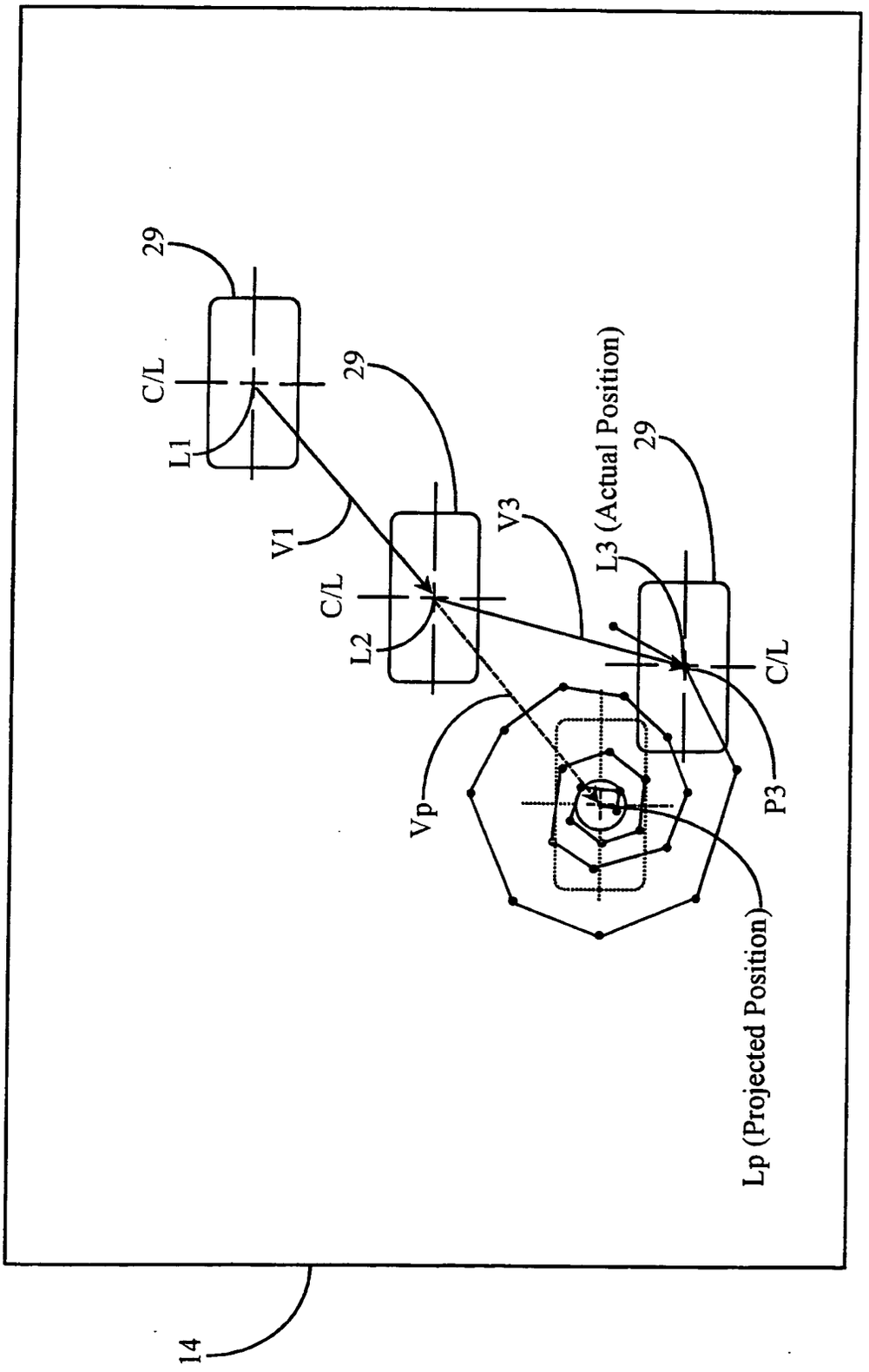

29
C/L
L1
V1
29
C/L
L2
V3
L3 (Actual Position)
29
C/L
Vp
P3
Lp (Projected Position)
14

Fig. 5

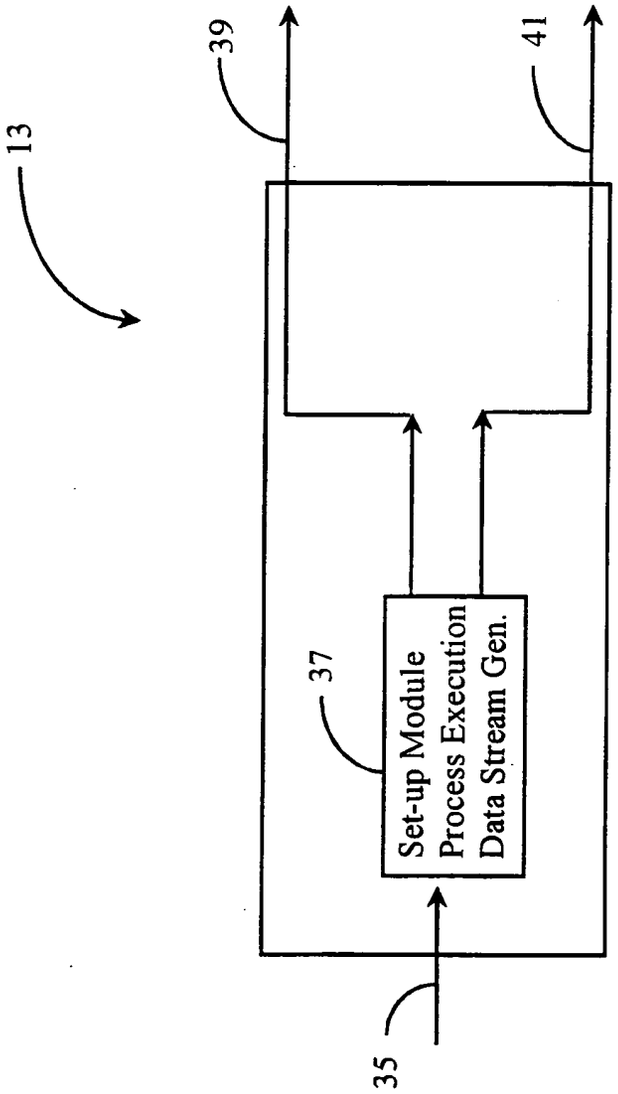

13
39
41
37
Set-up Module
Process Execution
Data Stream Gen.
35

Fig. 6

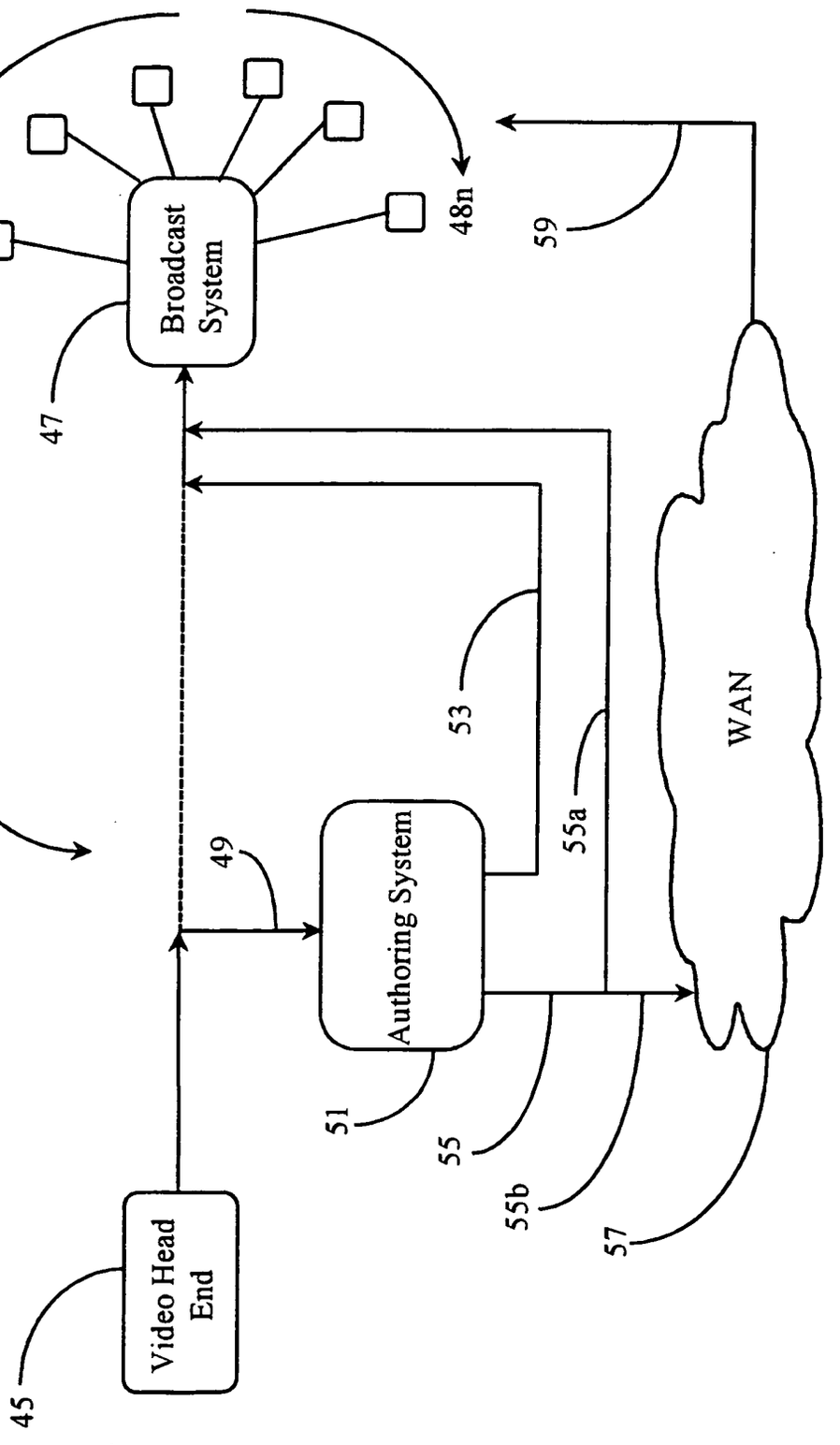

48a
48n
Broadcast System
47
59
43
49
53
55a
WAN
Authoring System
51
55
55b
57
Video Head End
45

Fig. 7

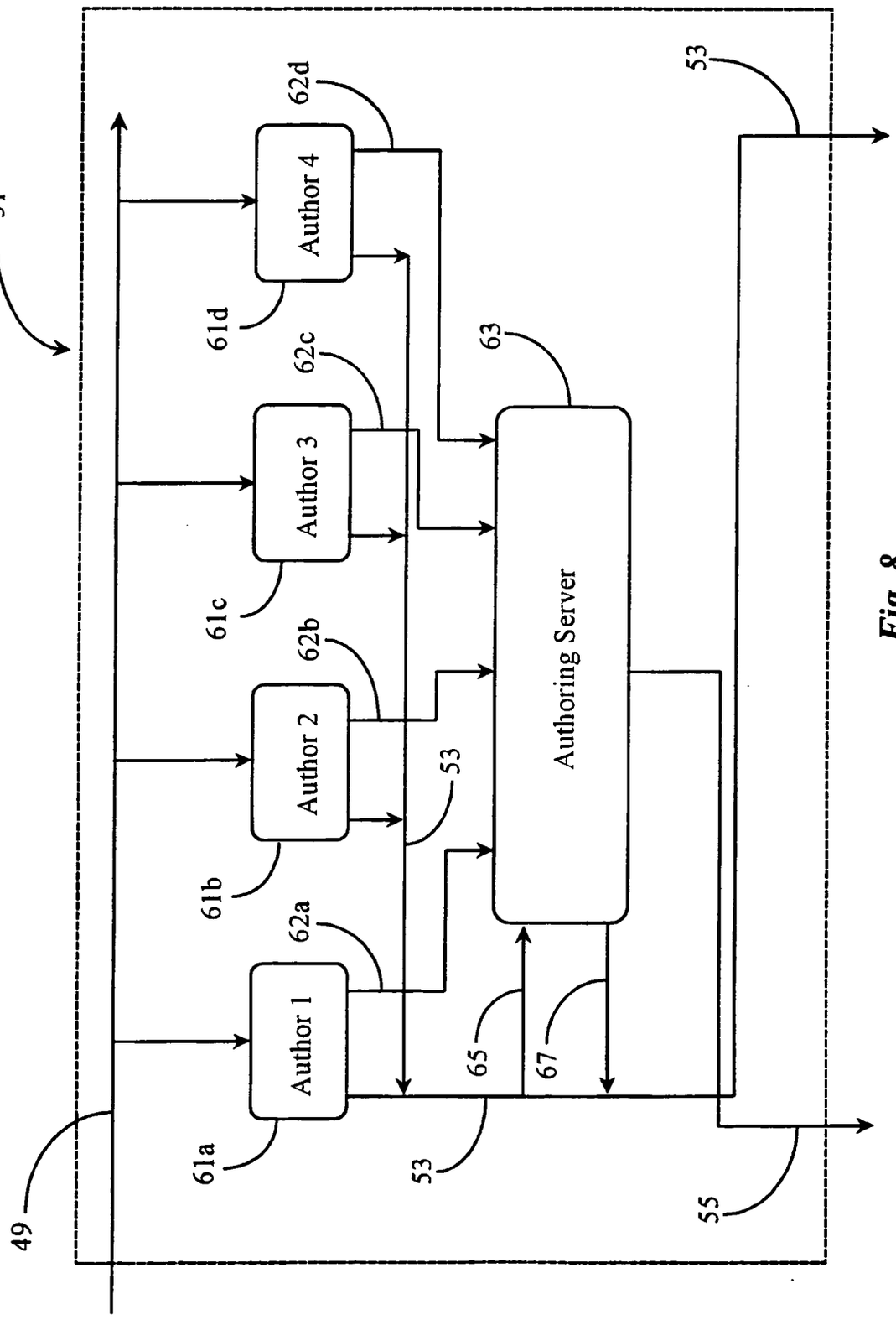

51
Author 4
62d
61d
Author 3
62c
61c
Author 2
62b
61b
Author 1
62a
61a
63
Authoring Server
53
53
53
65
67
49
55

Fig. 8

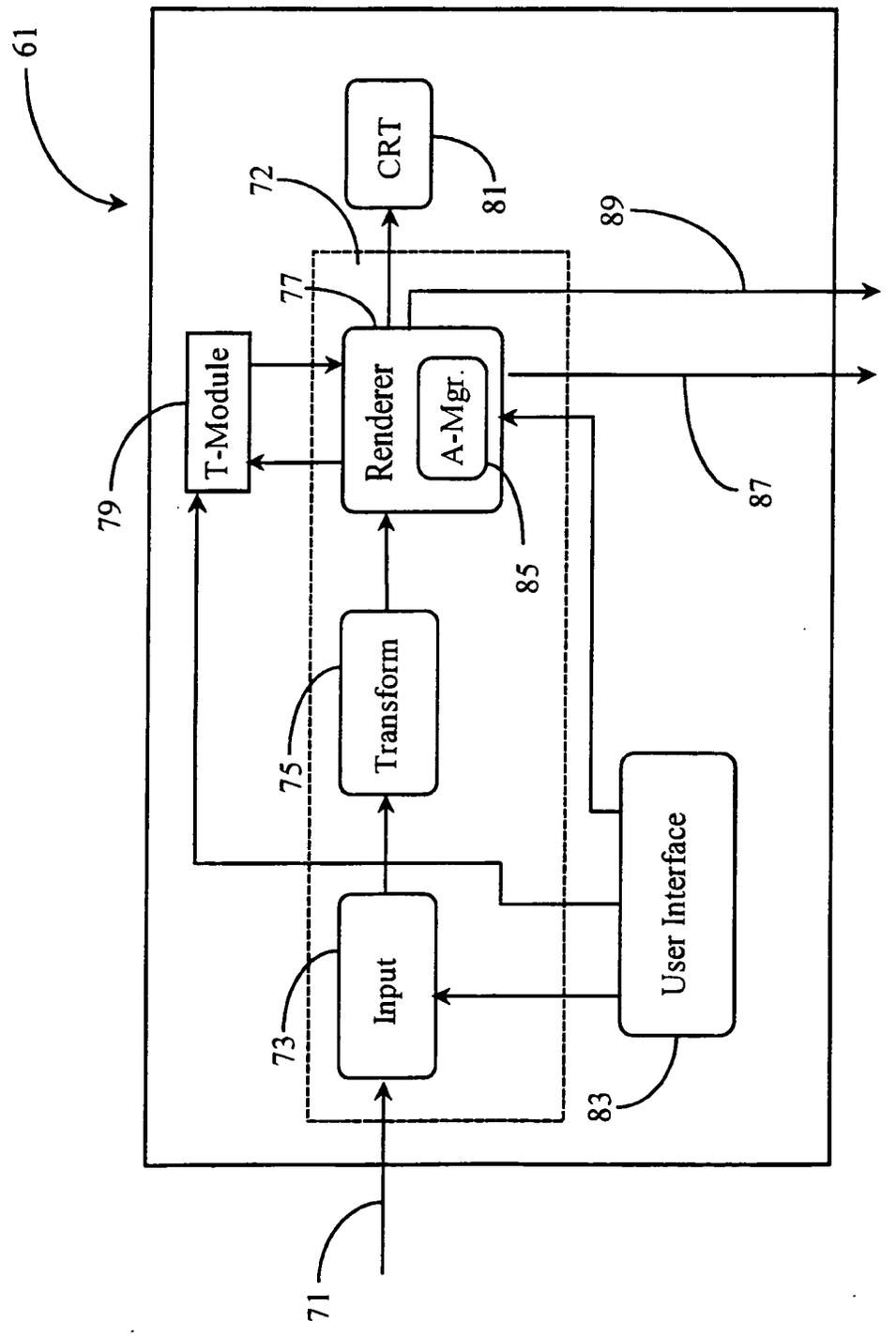

61
CRT
81
72
89
77
T-Module
Renderer
A-Mgr.
79
87
85
Transform
75
User Interface
Input
73
83
71

Fig. 9

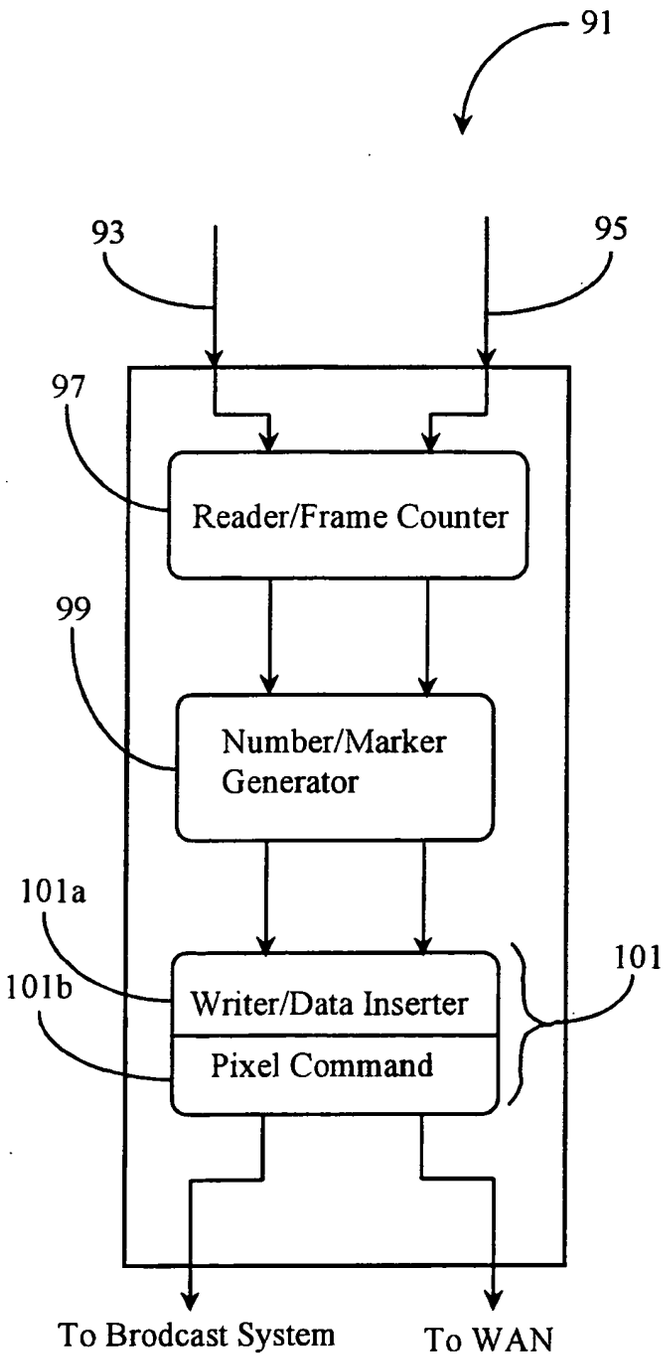

91
93
95
97
Reader/Frame Counter
99
Number/Marker Generator
101a
101b
Writer/Data Inserter
Pixel Command
101
To Brodcast System
To WAN

Fig. 10

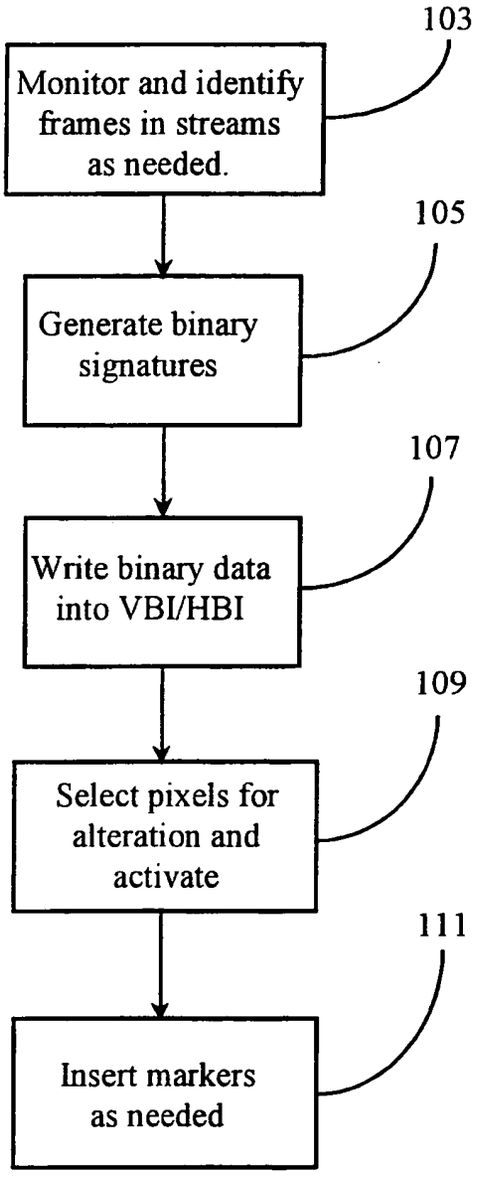

103
Monitor and identify frames in streams as needed.
105
Generate binary signatures
107
Write binary data into VBI/HBI
109
Select pixels for alteration and activate
111
Insert markers as needed

Fig. 11

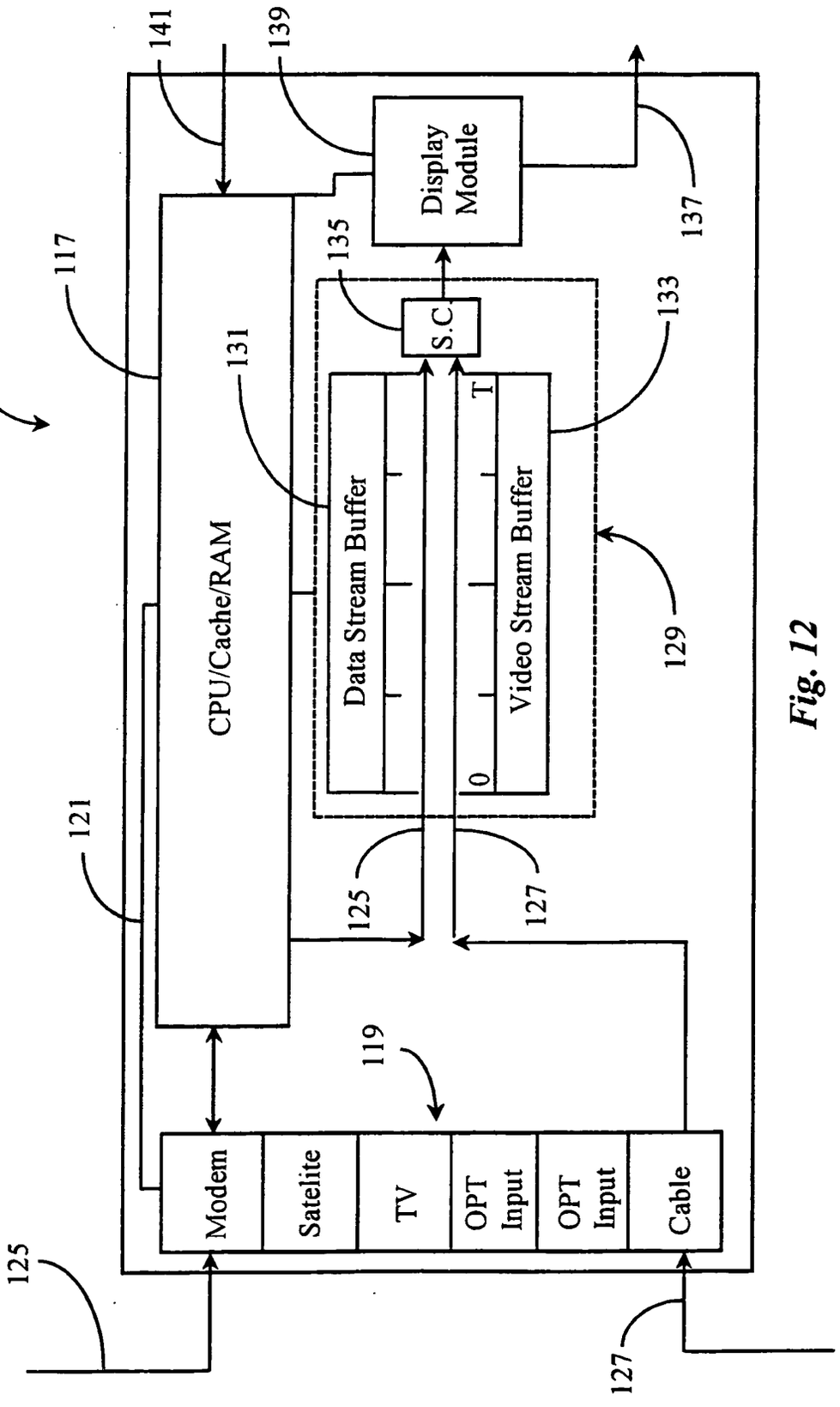

115
117
141
139
Display Module
137
135
S.C
133
131
Data Stream Buffer
T
0
Video Stream Buffer
129
CPU/Cache/RAM
121
125
127
119
Modem
Satelite
TV
OPT Input
OPT Input
Cable
125
127

Fig. 12

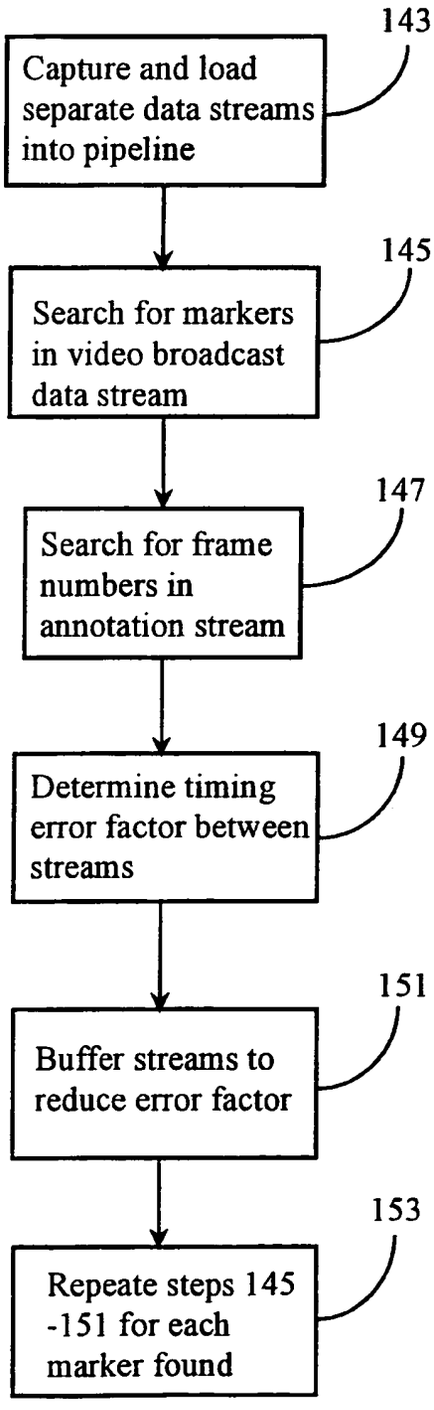

143
Capture and load separate data streams into pipeline
145
Search for markers in video broadcast data stream
147
Search for frame numbers in annotation stream
149
Determine timing error factor between streams
151
Buffer streams to reduce error factor
153
Repeate steps 145 -151 for each marker found

Fig. 13

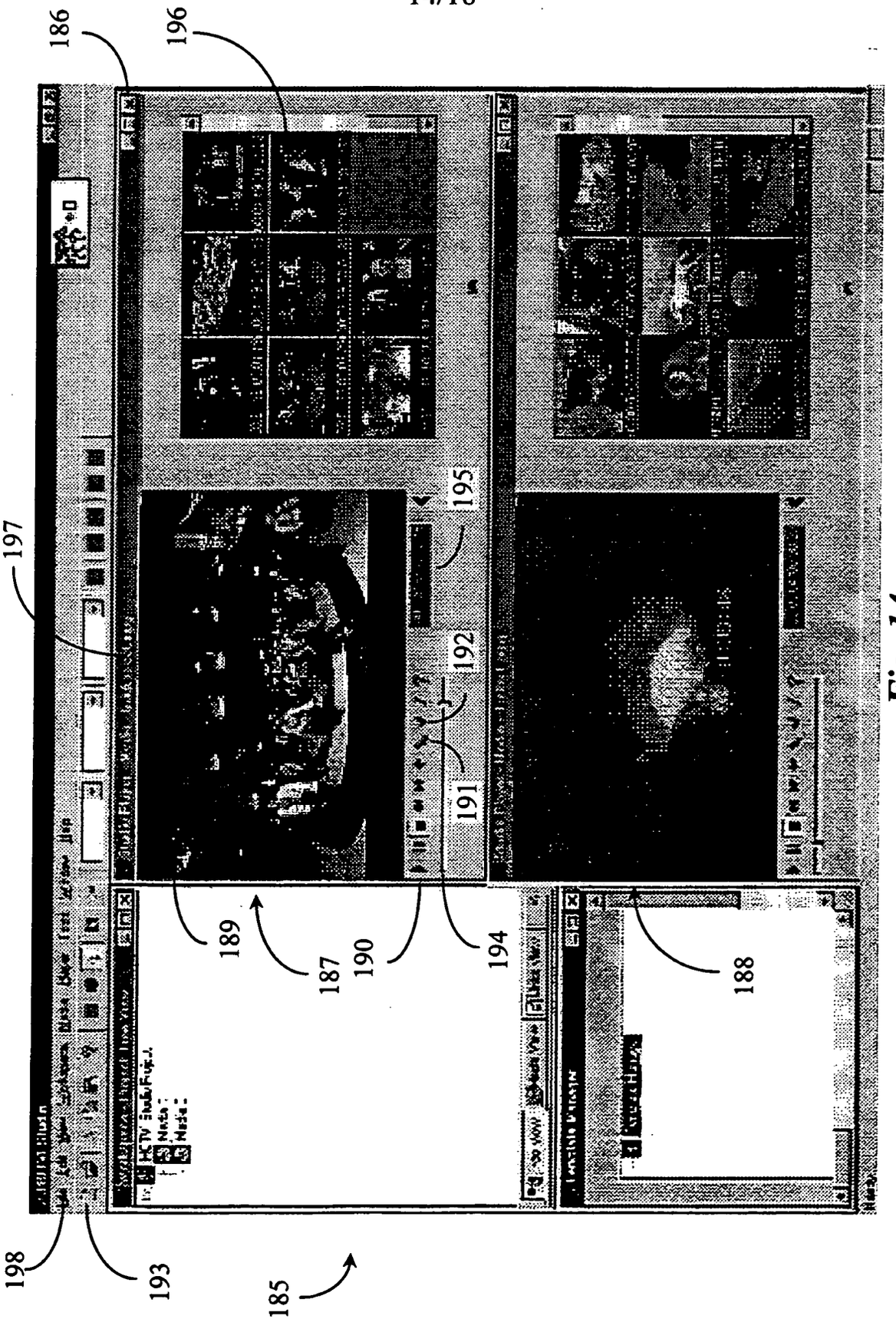


Fig. 14

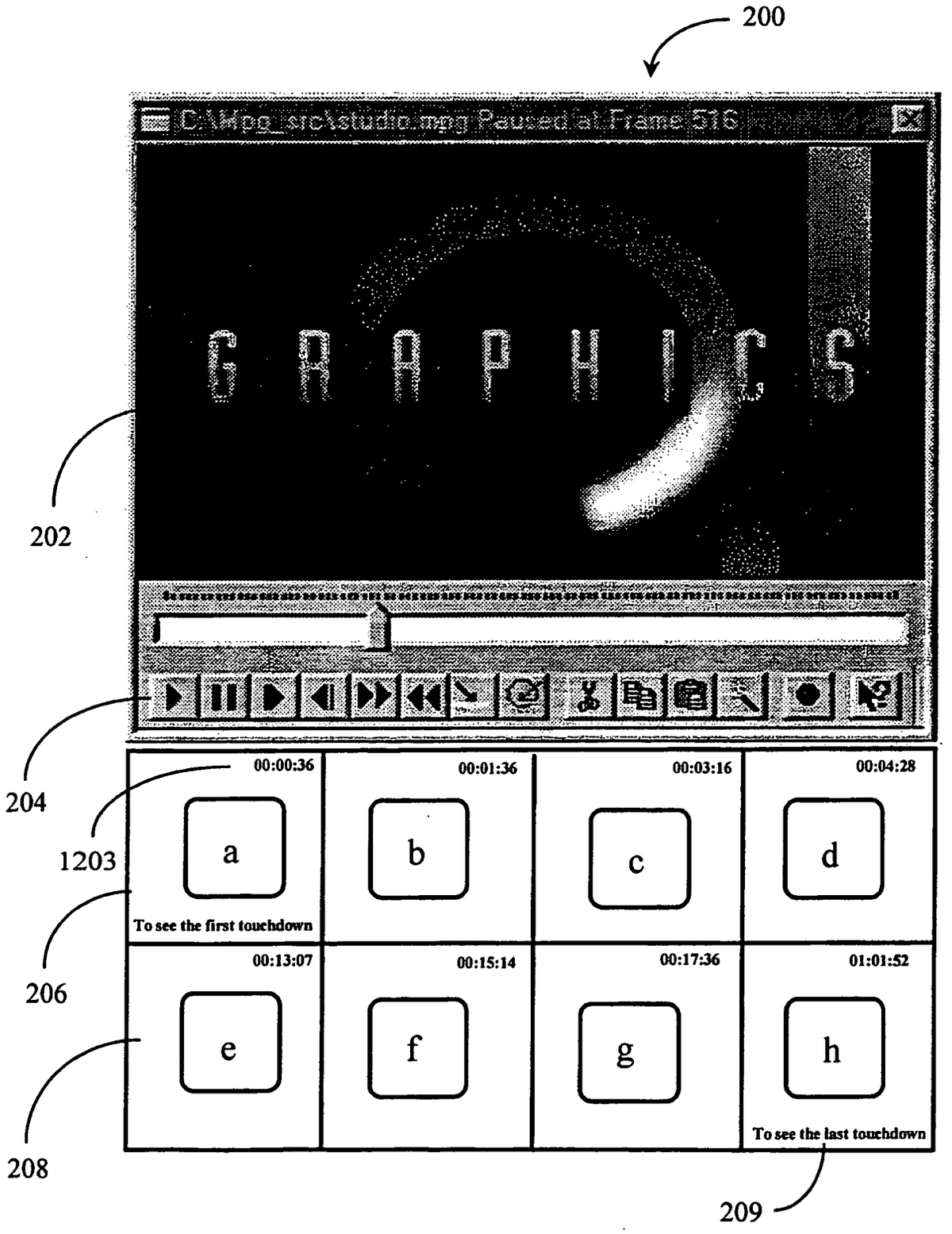

200
C:\Mpg_src\studio.mpg Paused at Frame 516
GRAPHICS
202
204
1203
206
208
209
00:00:36
a
To see the first touchdown
00:01:36
b
00:03:16
c
00:04:28
d
00:13:07
e
00:15:14
f
00:17:36
g
01:01:52
h
To see the last touchdown

Fig. 15

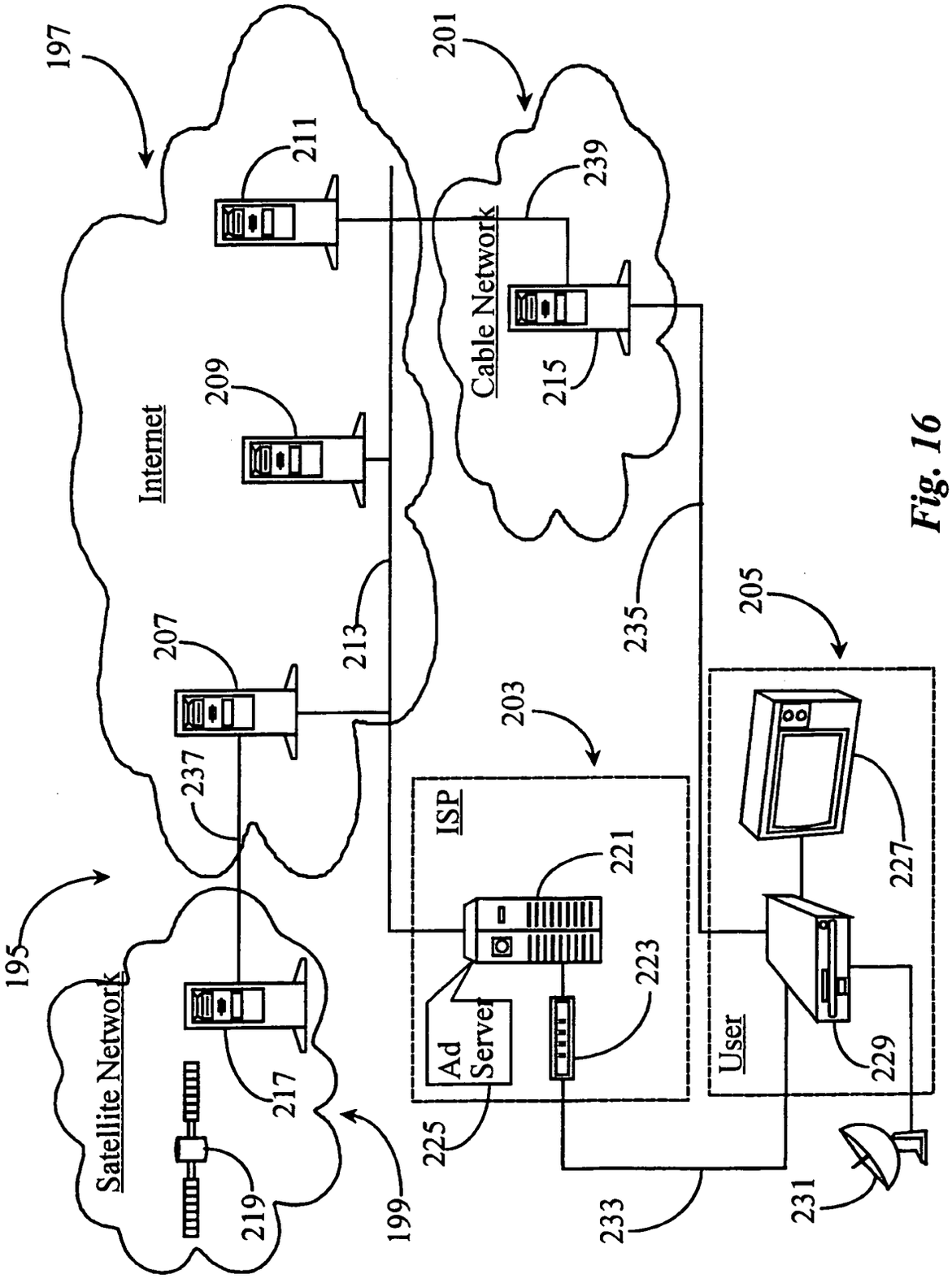

197
201
211
239
Internet
Cable Network
209
215
207
213
235
205
203
237
ISP
221
227
195
Satellite Network
217
223
Ad Server
User
225
229
219
199
233
231

Fig. 16

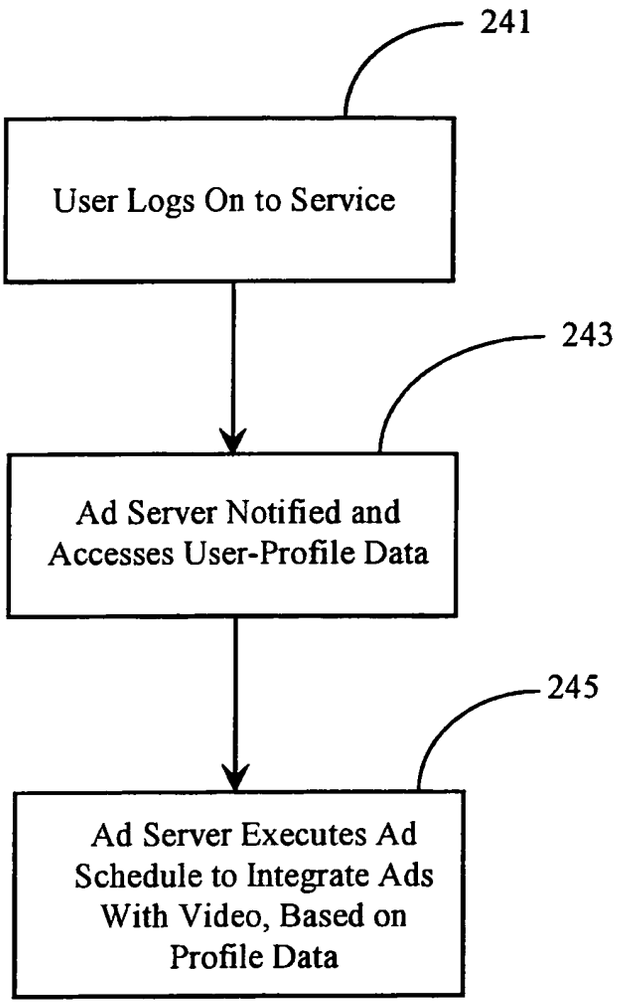

241
User Logs On to Service
243
Ad Server Notified and Accesses User-Profile Data
245
Ad Server Executes Ad Schedule to Integrate Ads With Video, Based on Profile Data

Fig. 17

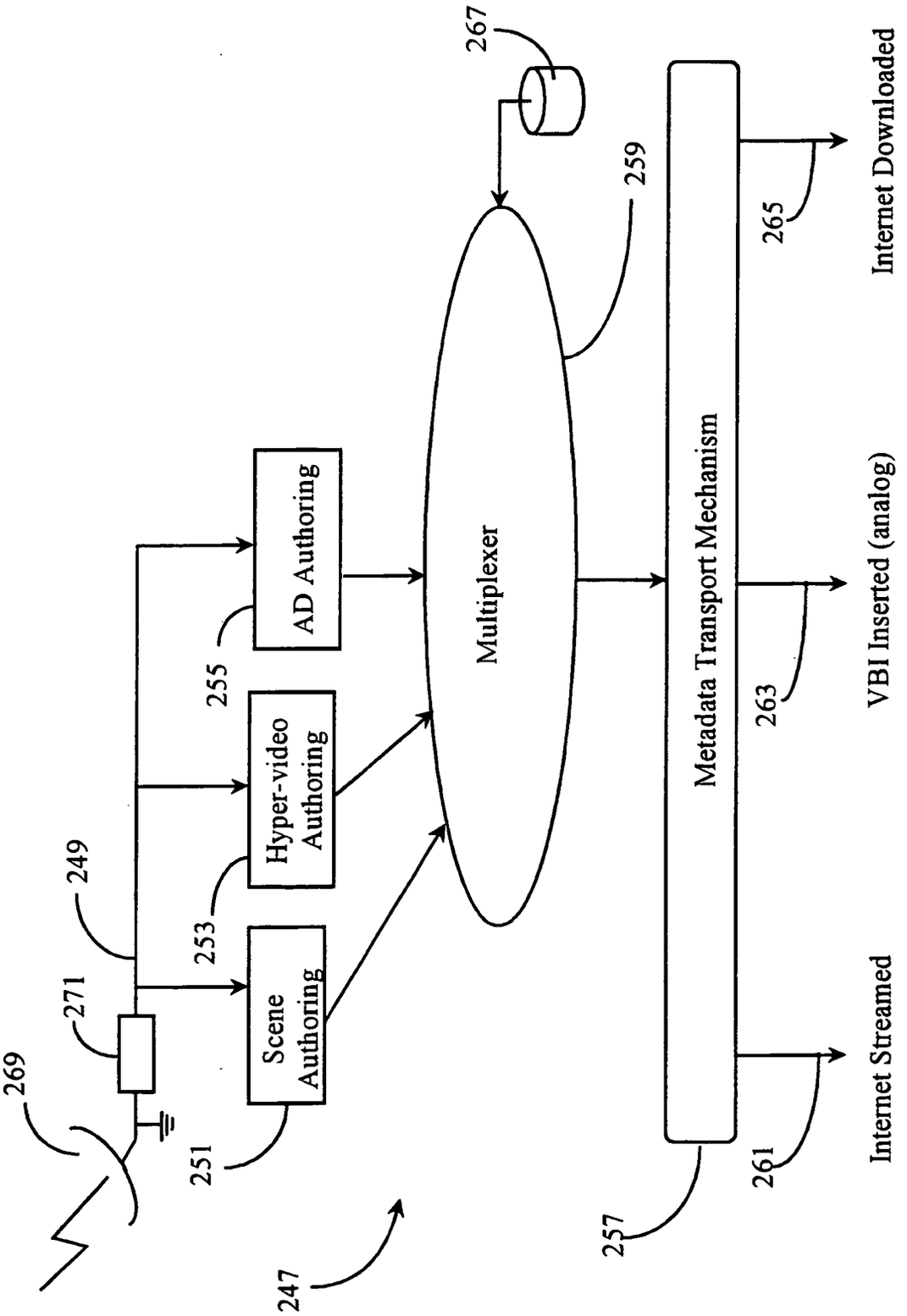

269
271
249
253
251
247
Scene Authoring
Hyper-video Authoring
255
AD Authoring
Multiplexer
267
259
Metadata Transport Mechanism
257
265
Internet Downloaded
263
VBI Inserted (analog)
261
Internet Streamed

Fig. 18

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/01699

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/167; H04N 7/10, 7/14; 1/14, 1/00
US CL :709/217; 348/12, 13, 6, 10; 455/3.1, 3.2, 5,1, 6.1, 6.2, 6.3

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/217; 348/12, 13, 6, 10; 455/3.1, 3.2, 5,1, 6.1, 6.2, 6.3

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

STN - data, commercial, Insertion, time stamp, PTS, video

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 5,894,328 A (TAHARA et al) 13 April 1999, col. 8, lines 40-67, col. 9 lines 1-10, col. 10 lines 56-67, col. 11 lines 7-14	1-27
A, P	US 6,002,393 A (HITE et al) 14 December 1999, ALL	1-27
A, P	US 6,034,746 A (DESAI et al) 07 March 2000, ALL	1-27
A	US 5,424,770 A (SCHMELZER et al) 13 June 1995, ALL	1-27
A	US 5,027,400 A (BAJI et al) 25 June 1991, ALL	1-27
A, P	US 5,917,830 A (CHEN et al) 29 June 1999, ALL	1-27

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"B"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
23 APRIL 2000	18 MAY 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Vivek Srivastava Telephone No. (703) 305 - 4038

Form PCT/ISA/210 (second sheet) (July 1998) ★

(12) UK Patent Application (19) GB (11) 2 350 213 (13) A

(43) Date of A Publication 22.11.2000

(21) Application No 9911739.2

(22) Date of Filing 21.05.1999

(51) INT CL[7]
G06F 17/30

(52) UK CL (Edition R)
G4A AMU AUDB

(56) Documents Cited
WO 97/46955 A1
http://libsun1.jr2.ox.ac.uk/training/netscape.html
HCLU,"Netscape Navigator", 1996

(58) Field of Search
UK CL (Edition R) G4A AMU AUDB
INT CL[7] G06F 17/30
ONLINE: COMPUTER EPODOC JAPIO WPI INTERNET

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Gordon Douglas Hutchison
Noel Rooney

(74) Agent and/or Address for Service
P Waldner
IBM United Kingdom Limited, Intellectual Property Department, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(54) Abstract Title
Web page downloading

(57) A method of using a "markup language", such as HTML, browser, in which a page is loaded into display memory, the user selects multiple URLs from those displayed on the page, and the browser preloads pages corresponding to the URLs into a cache without necessarily displaying them, so the user may continue to view the original page. Before preloading a given page, it may update a navigation list with the corresponding URL, and a user may select an entry in this list, directly or via "forward" and "back" GUI buttons, to display immediately.

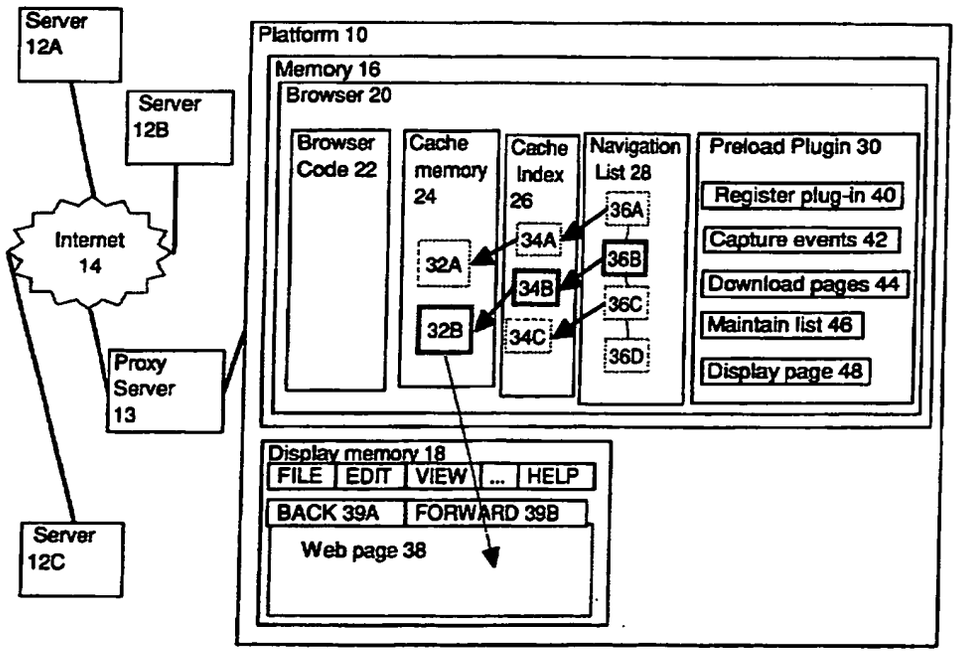


Figure 1

GB 2 350 213 A

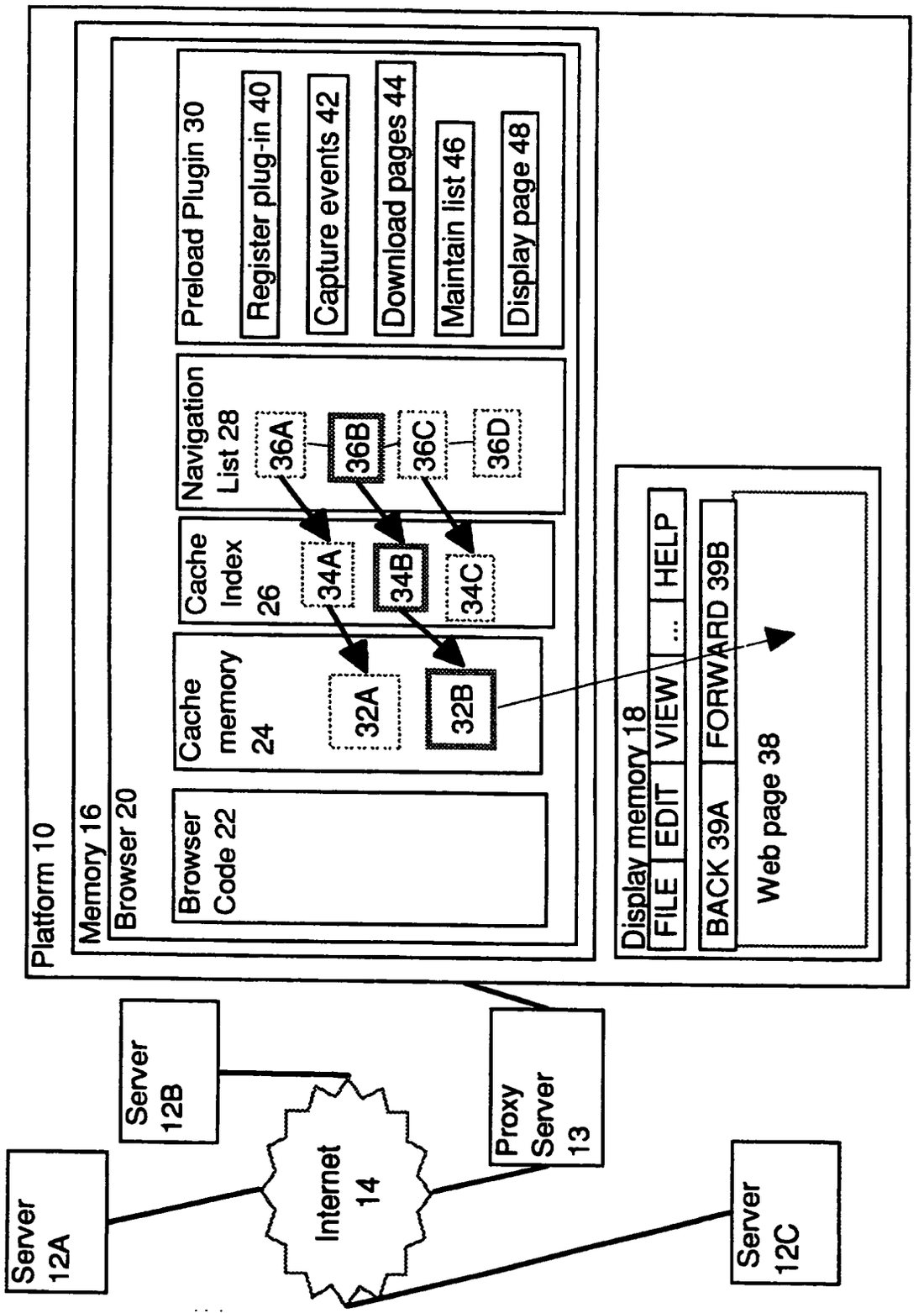

Platform 10
Memory 16
Browser 20
Browser Code 22
Cache memory 24
32A
32B
Cache Index 26
34A
34B
34C
Navigation List 28
36A
36B
36C
36D
Preload Plugin 30
Register plug-in 40
Capture events 42
Download pages 44
Maintain list 46
Display page 48
Display memory 18
FILE
EDIT
VIEW
...
HELP
BACK 39A
FORWARD 39B
Web page 38
Server 12A
Server 12B
Server 12C
Internet 14
Proxy Server 13

Figure 1

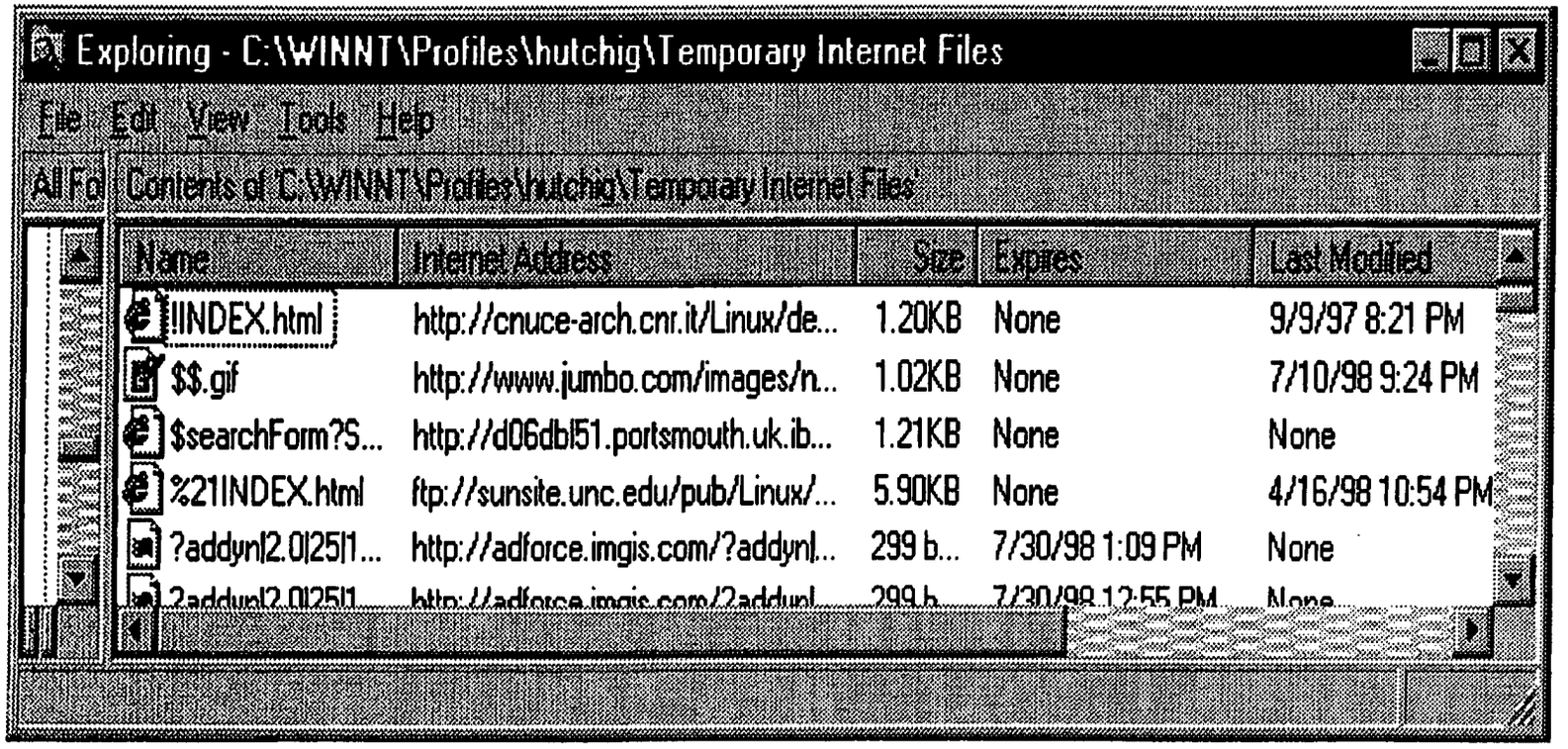

Exploring - C:\WINNT\Profiles\hutchig\Temporary Internet Files
File Edit View Tools Help
All Fo
Contents of 'C:\WINNT\Profiles\hutchig\Temporary Internet Files'
Name
Internet Address
Size
Expires
Last Modified
!INDEX.html
http://cnuce-arch.cnr.it/Linux/de...
1.20KB
None
9/9/97 8:21 PM
$$.gif
http://www.jumbo.com/images/n...
1.02KB
None
7/10/98 9:24 PM
$searchForm?S...
http://d06db151.portsmouth.uk.ib...
1.21KB
None
None
%21INDEX.html
ftp://sunsite.unc.edu/pub/Linux/...
5.90KB
None
4/16/98 10:54 PM
?addynl2.0|25|1...
http://adforce.imgis.com/?addynl...
299 b...
7/30/98 1:09 PM
None

Figure 2

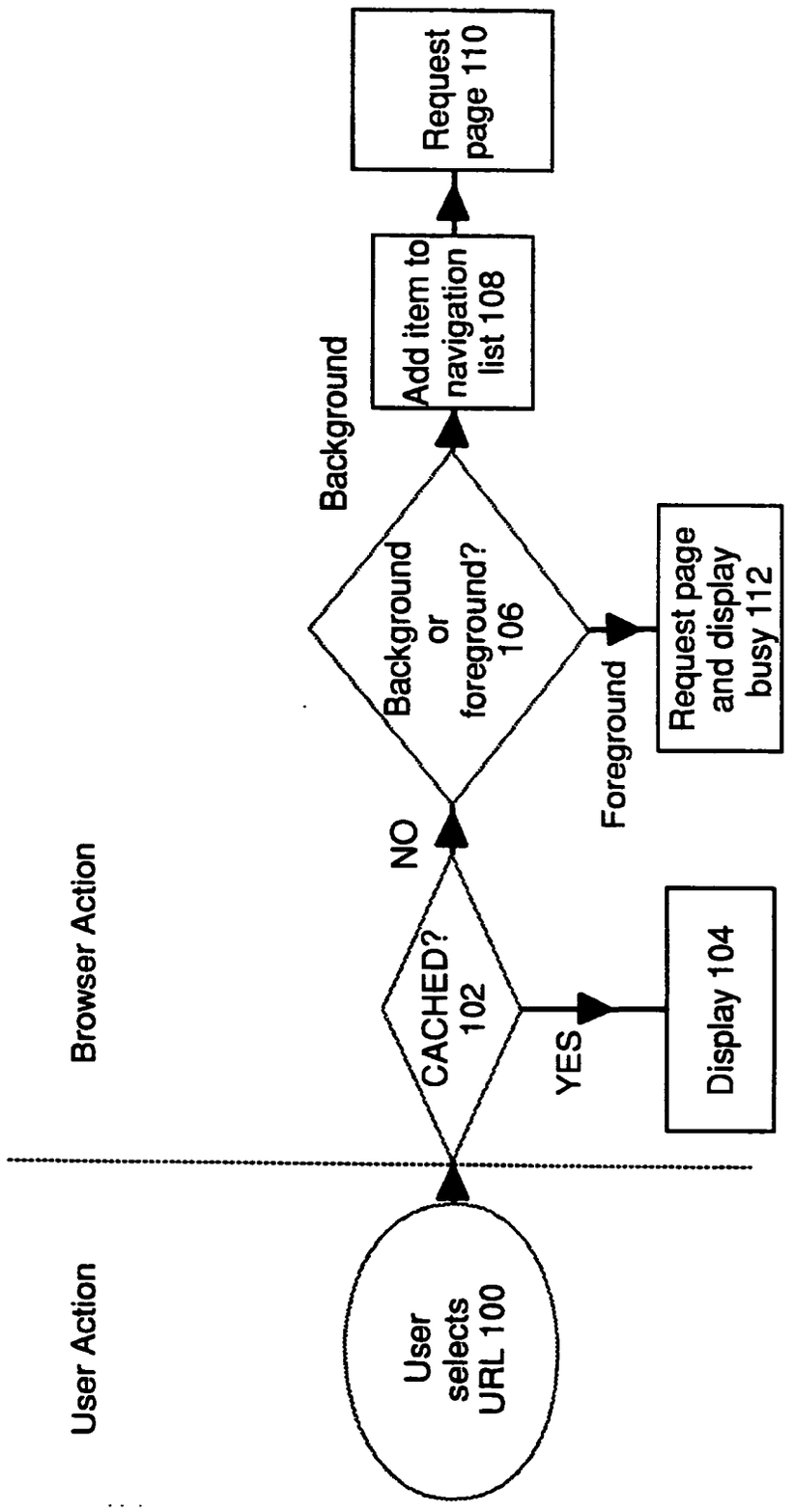

User Action
Browser Action
User selects URL 100
CACHED? 102
YES
Display 104
NO
Background or foreground? 106
Foreground
Request page and display busy 112
Background
Add item to navigation list 108
Request page 110

Figure 3A

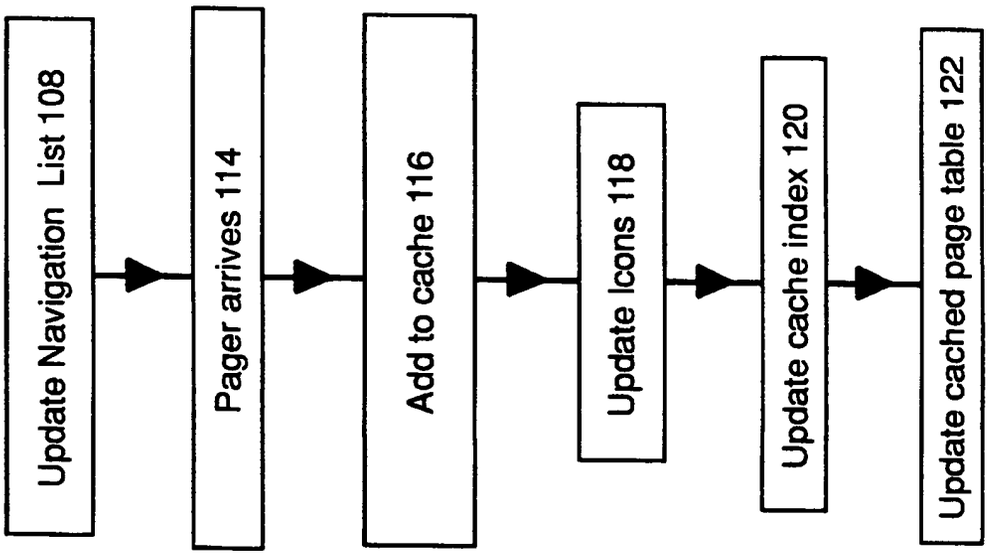

Update Navigation List 108
Pager arrives 114
Add to cache 116
Update Icons 118
Update cache index 120
Update cached page table 122

Figure 3B

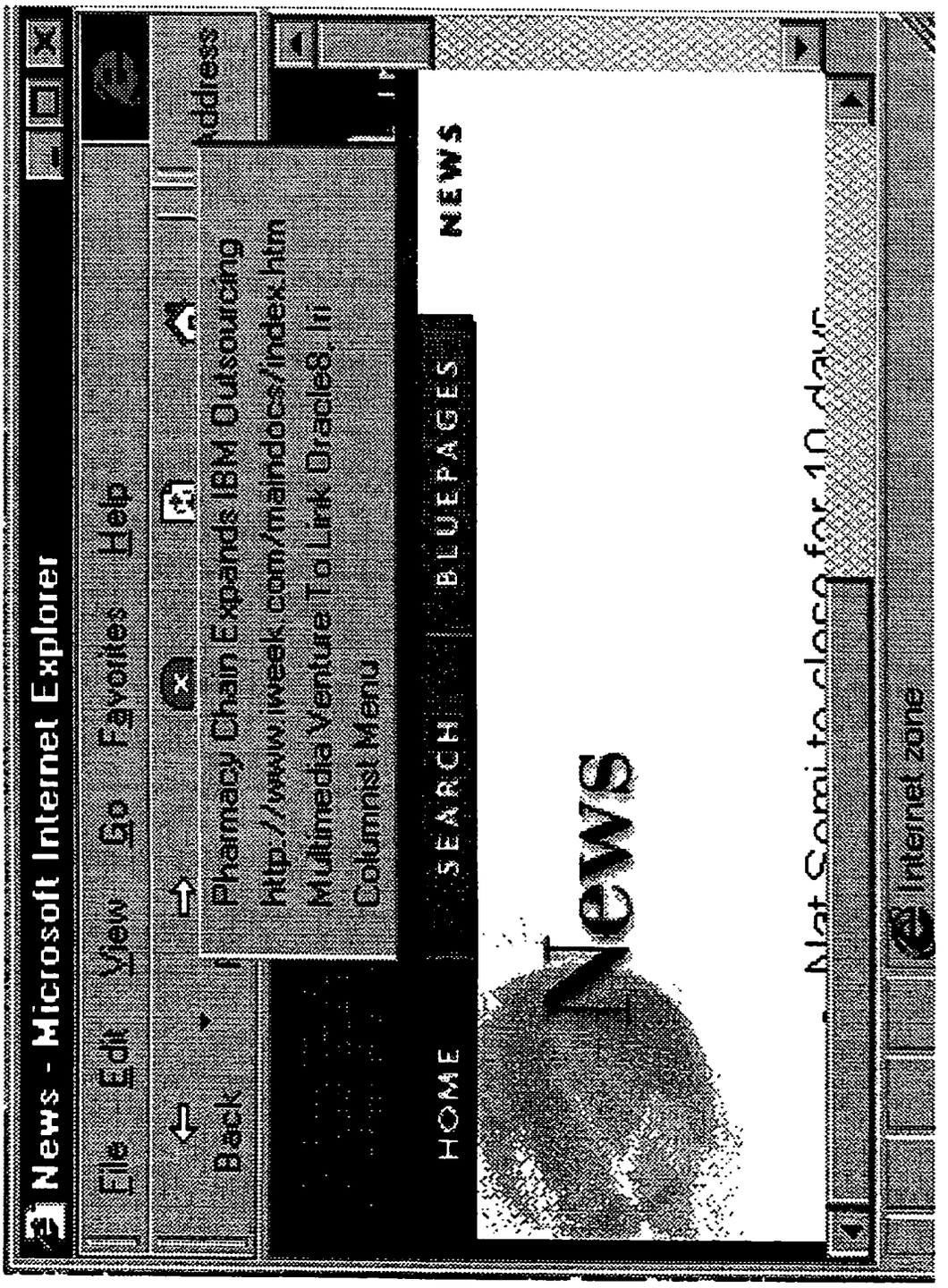

News - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back
Address
Pharmacy Chain Expands IBM Outsourcing
http://www.iweek.com/maindocs/index.htm
Multimedia Venture To Link Oracle8, Iii
Columnist Manu
HOME
SEARCH
BLUEPAGES
NEWS
News
Internet zone

Figure 4

WEB PAGE DOWNLOADING

FIELD OF INVENTION

This invention relates to the Internet world wide web page downloading and in particular relates to browser prefetching of web pages.

BACKGROUND OF INVENTION

The Internet is a large number of computers connected in an ad hoc fashion forming a world wide network. It was originally conceived by the US government to allow a scaleable network of computers to talk to each other and share information and has the added effect of being resilient to malfunction of single computers in the network. This means a lower risk of loosing total communication between computers in the event of an intermediate computer failing because the network may be connected in other ways. The communication language of the Internet is the double layer Transmission Control Protocol/Internet Protocol (TCP/IP). TCP controls the way that data is broken down into small packets for transmission over the Internet and controls the way the packets are subsequently built up in the correct order. The Internet protocol controls the addressing of the packets so that they are transmitted to and received by the correct computer.

The Internet's best known medium is the world wide web (WWW) which is very broadly a set of protocols allowing transfer of multimedia data between computers. One of these protocols is Hypertext Transfer Protocol (HTTP) which allows pages of data in hypertext mark up language (HTML) to be transferred. The application which sits on a computer to make this transfer possible is an Internet browser such as Microsoft Internet Explorer or Netscape Navigator.

A very common mode of operation in web browsers is the following. A user enters criteria into a search engine prompt or goes to an index/contents page of link. The user is then presented with list of available web Uniform Resource Locators (URLs) a subset of which he/she wishes to browse and selects one location for download. The user then waits for the page to load from the Internet server to the proxy server and to his machine before viewing the page. After finding something of interest or not the user moves back to the index/list and makes a next selection, waits for download etc. This process repeats until the user has read or collected the relevant material. In this mode of operation it is necessary to repeatedly wait, sometimes for excessive periods, for the pages to download. One solution is to launch a new browser to request a download from the selection and return to the original browser to

launch further browsers for each desired URL. This is a messy and irritating way to operate as one must keep track of where one is in the desired list and has no visual clue as to when the pages have been downloaded to the multiple windows and which to which window on the screen a particular page will appear at.

GotIt from Ahead Software Inc., Redmond, WA (www.goahead.com) preloads web pages into proxy cache. GotIt is a www proxy gateway to the Internet for providing a portal and location for fire walls, security policy and web page caching as standard. For example most www proxies are likely to fetch www.yahoo.com many times each hour and the advantage of storing it and refreshing it say only every 10 minutes is clear. GotIt allows preloads of pages linked to the current page. The proxy server will download from the servers in the background all the pages that are linked to from a page currently being viewed it then stores these on the disk of the machine running the proxy.

However in the GotIt solution the proxy has no interaction with the browser interface and visa versa during the background downloading process. This has two disadvantages: 1) The user's model is not improved since the process of moving to proxy pages remains inflexible. 2) The GotIt! proxy has no way of knowing which pages the user will actually hit so it downloads all of them that are linked to. It undoubtedly filters out these links for it's own use during the action of serving the linking page to the user. Depending on the performance of serving machine and the nature of the linking page, this could actually slow down performance if, for example, the user is only interested in the 3,4, 6 and 9th links as all 1 to 12 links will be fetched. GotIt's pre-fetch model blindly prefetches all links on a page. This would result in too many prefetched pages for the average browser set up and platform as sites now commonly have many navigation links at their top levels.

SUMMARY OF INVENTION

According to one aspect of the invention there is provided a method of browsing an HTML page on a TCP/IP network comprising: loading a first HTML page into browser memory from a network server; loading said memory into display memory for displaying said the page; providing means for selecting multiple URLs on the HTML page; and preloading pages corresponding to user selected URLs into browser memory without loading them into display memory or affecting the ability to select further URLs. The solution allows multiple requests by the user and results in a much reduced start-to-finish time for a user who is using the net in a 'query, list, browse' mode (as is commonly done). The user need not look at every page downloaded and available from cache memory. As most of the pages will be 'instant' from the cache, page providers may benefit too by

receiving more 'hits' as users are much more likely to view a page if they can see that it will be displayed 'instantly' from the browser cache.

Advantageously, this aspect of the invention further provides a step of on selection of a URL and before preloading the corresponding page, updating a navigation list with the URL. This allows a user, having finished making multiple selections from the first page, to instruct the browser to display pages from the navigation list even though the pages may not be fully loaded into memory. More advantageously the method further comprises updating the navigation list with information as to whether a URL on the list is fully loaded. In which case the user can maximise the time spent in viewing only fully loaded pages.

Advantageously, on selection of a URL and before preloading the corresponding page into cache memory, there is provided a further step of updating a cache index file with the location of the page in cache. This allows a user, having finished making multiple selections from the first page, to instruct the browser to display pages from the navigation list even though the pages may not be fully loaded into memory.

Advantageously pages corresponding to selected URLs are not displayed during preloading so that the user may view the original page until finished checking all the URLs. After the user is finished checking he may move on to one of the selected web pages.

One beneficial method for navigating the selected URLs is to build a list of page references including that of the first page. A drop down list may be provided for selecting a page reference in the list so that the corresponding page can be displayed. Alternatively or as well as 'forward' and 'back' GUI buttons may be provided for selecting the previous or next page reference in the list and displaying the corresponding page.

The user is already used to using control-mouse-left-click (for copy-drag actions in Windows) and the above solves the problem in an intuitive way. As the model suggests, making use of the 'forward-arrow' for navigation through the pages downloaded in the background is a familiar navigation method for the user. The browser cache will end up in exactly the same state it would if he/she had interactively downloaded the pages one at a time and then used 'back-arrow' to return to the index page. Moreover the user is not faced with having to continually jump back to the top level and has much less 'waiting time' overall as the downloads occur in parallel. Furthermore the user has something active to do/read while downloads are going on the background the operation will be even more efficient/enjoyable.

Preferably a page reference is added to the list when its corresponding URL is selected from the first page so that the list is order sequentially in a logical order. Alternatively a page reference maybe added to the list when its corresponding page is substantially loaded into browser memory so that if a user moves forward in the list there is greater chance that the page will have loaded into the browser. This solution provides for a much more easily navigable model for doing parallel downloads over the current possible mechanisms. The solution is coherent with the current user model of path walking with marks for places visited. The solution is applicable to a 'usage scenario' employed many times every day in a highly competitive area and is easily detectable.

Browser memory comprises cache memory. When a preloaded page in proxy memory is required for display on a browser, the browser sends a TCP/IP request to the proxy for a page stored in proxy memory. The page is copied into both cache memory and display memory for display on the browser. It is advantageous to preload selected pages directly into cache memory from an TCP/IP network so that a quicker internal request from the browser to the cache memory renders the displayed page to the user in a shorter time as opposed to a slower TCP/IP request from the browser to the proxy.

According to another aspect of the invention there is provided a system for browsing an HTML page on a TCP/IP network comprising: means for loading a first HTML page into browser memory from a network server; means for loading said memory into display memory for displaying said page; means for selecting multiple URLs on the HTML page; and means for preloading pages corresponding to selected URLs into browser memory without loading them into display memory.

According to another aspect of the invention there is provided a method of processing a URL database comprising: loading the HTML result of a URL search into browser memory; loading the HTML results into display memory and displaying a list of located URLs; providing means for selecting multiple URLs from the list; and preloading web pages corresponding to selected URLs without displaying said web pages.

According to a further aspect of the invention there is provided a computer program product comprising a computer usable medium having computer readable program code means embodied in the medium for processing work items in a data processing system, the program code means comprising: code means for causing the data processing system to load a first HTML page into browser memory from an network server; code means for causing the data processing system to load said first HTML page into display memory for displaying said page; code means for causing the data

processing system to provide means for selecting multiple URLs on the HTML page; and code means for causing the data processing system to preload pages corresponding to selected URLs into browser memory without loading them into display memory.

According to a further aspect of the invention there is provided a computer program product comprising a computer usable medium having a computer readable code page embodied in the medium for processing work items in a data processing system, the code page comprising: code means for instructing a code page browser to display one or more links to other codes pages; and code means for, on selection of one or more of the displayed links, instructing a code page browser to load one or more of the code page into browser memory whilst instructing the code page browser to retain the current code page display of said one or more links.

Once a user has completed the process of viewing all his/her desired pages by initiating each download interactively, all the previous pages are easily selectable for reviewing using the 'back' and 'forward' features of the browser. As these work in conjunction with the browser cache one can quickly move backward through the list (and subsequently forward again). If it were possible to reverse this process how much easier say Yahoo or patent searches would be.

Although the invention is described in terms of TCP/IP networks it could equally apply to other types of packet networks. Furthermore although the invention is described in terms of HTML pages it could equally apply to other types of mark up language data containing URLs. Although the invention is described in terms of URLs it could equally apply to other network data references.

BRIEF DESCRIPTION OF DRAWINGS

In order to promote a fuller understanding of this and other aspects of the present invention, an embodiment will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic representation of a computer system of the embodiment;

Figure 2 is a screen dump of a cache memory;

Figure 3A and 3B represent the procedures of the embodiment; and

Figure 4 is a screen dump illustrating one of the advantages of the embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Referring to Figure 1 there is represented an operational platform 10 connected to a plurality of servers 12A, 12B, 12C, via a proxy server 13 and an Internet network 14. Platform 10 is an Intel Pentium II 450Mhz processor based computer, having 128 MB 100 MHz SDRAM (reference number 16 on Figure 1), 17 GB ATA-33 hard drive (also represented as reference number 16 on Figure 1), 19" colour SVGA monitor, 16 MB video memory (reference number 18 on Figure 1), 17/40 x CD-ROM drive, 3 PCI, 1 ISA and 1 AGP expansion slots and Microsoft Windows 98 operating system. The platform 10 has a TCP/IP browser 20 such as Netscape Navigator. The servers 12,13 have similar specifications to platform 10 but run under Microsoft Windows NT operating system. Although the example platform is described in detail, any processor and operating system combination capable of running a TCP/IP browser is suitable. The example browser need not be Netscape Navigator but any browser capable of receiving mark up code pages from a network, for instance Microsoft Internet Explorer.

Browser 20 is the client application which is used to view pages of information sent by the servers. Five components of the browser are of interest in the present embodiment. The browser code 22; the cache memory 24; the cache index 26; and the navigation list 28 are known components of browser 20. In this embodiment the invention is centred on a browser plug-in 30 so that no adaptation of the browser code is necessary and the invention may be incorporated into any present browser set up by adding the component to the browser.

In another embodiment the browser code 22 may be modified with necessary functionality so that no plug-in to the shipped browser is necessary.

In a further embodiment the methods of the invention may take the form of code (such as Javascript) within the web page data instructing the browser to carry out the steps of the invention.

Proxy server 13 is the gateway to the Internet for the platform 10 and potentially many other platforms. When web page information is downloaded from servers on the Internet to the browser it is stored in the proxy server 13 as well as being transferred to the cache memory 24 of the browser 20. If any other browser using the proxy server 13 as a gateway requests the same page then the proxy server 13 can transfer it from its own memory without incurring the delay by transferring it again from the Internet. The proxy also has some security functionality for limiting what information is sent and received. Internal web pages can not be sent through 'fire wall' security measures in the proxy server.

Browser code 22 contains the functionality of the browser including the TCP/IP interface code and the mark up language interpreter (both not

shown). Cache memory 24 stores several pages 32A and 32B including main pages and related files of information from the Internet servers. The proxy server 13 has a similar cache memory for Internet information but stores more information than a browser cache memory so that it may supply the other connected browsers. Cache index 26 contains the groups 34A,B,C of names and references of the files that make up web pages 32A and 32B. Navigation list 28 is a sequential list of web pages visited in the browser session with associated URLs. Both Netscape Navigator and Microsoft Internet Explorer have cache memories 24 where HTML, GIFs, MP3 etc. files are cached in a hard disk directory with a cache index 26. With Internet Explorer these are held at C:\winnt\profiles\<user>\Temporary Internet Files\index.dat and in Netscape the equivalent file is, for example, C:\Program Files\Netscape\Users\<user>\Cache\fat.db. For Internet Explorer, these can be seen using Windows Explorer as shown in Figure 2.

The navigation list contains the URLs 36A,B,C,D visited including the present web page 36B (indicated by the bold outline), those previously visited 36C,D (indicated by the dotted outlines) and the page 36A to be visited in the future (indicated by the dotted line). The cache index 26 holds file references 34A for the present web page and perhaps those of the previous web page e.g. 34B and 34C but not the web page references of 36D which have been removed due to limited space for illustration only. The actual embodiment has extended cache memory 24 and holds from two to ten or more complete web pages. The cache memory 24 holds the web page data 32A and 32B as referenced by the cache index 34A anbd 34B respectively and some or all of previous web pages, for example those referenced by pages 36C and 36D. The mark up code interpreter in the browser code 22 uses the mark up files 32A as indexed by cache index 34A and pointed to by list item 36A to represent a web page 38 in graphical form in the display memory 18 which is displayed to the user on the monitor.

Plug-in 30 uses a defined way of extending the functionality of the well-known browser described above without having to alter the underlying browser or requiring a recompilation of it. It works by a convention of entry points (or functions) that the browser can call within the plug-in and vice versa. The functions are called at significant points in the processing of a downloaded page and different page 'types' can be associated with different plugins. It is implemented as a dynamically loaded library and thus has access to the underlying computer's file system and other facilities (and so could manipulate the browser cache), it also has the ability to intercept mouse and keyboard actions from the user and have visibility of the browser's URL fetching via the defined browser to plug-in interface. Plugins conventionally deal with new file types (such as the well known VRML plug in that displays virtual reality

markup language and interacts with the user to control the display). However plug-in 30 deals with existing file types such as .HTML with slightly different processing than is conventionally used.

To implement the embodiment we need to achieve the following functional units as identified in Figure 1: means 40 to register a plug-in that is called for 'standard' HTML and HTM web pages; means 42 to capture mouse or keyboard events to control the page downloads; means 44 to request that pages are downloaded without being displayed and furthermore that these pages can be placed in the Netscape cache; means 46 to maintain a list of pages that have been so placed in the cache to enable them to be displayed in response to the user selecting the 'forward' and 'reverse' option in the user interface; and means 48 to trigger Netscape to display a particular page that is stored in the Netscape cache.

Each of these five parts of the design is implemented using the Netscape plug-in interface and will be discussed in turn.

1) Registering a plug-in that is called for 'standard' HTML and HTM web pages.

When Communicator starts up, it checks for plug-in modules in the plugins directory for the platform:

MS Windows: plugins subdirectory, in the same directory as the Communicator application.

Mac OS: Plug-ins folder. A Mac OS plug-in can reside in a different directory if one installs a Macintosh alias that links to the plug-in in the Plug-ins folder.

UNIX: usr/local/lib/Netscape/plugins or $HOME/.Netscape/plugins. If a different directory is wanted, the NPX_PLUG-IN_PATH environment variable is set to its filepath, for example, $HOME/yourplugins:/ usr/local/lib/Netscape/plugins. Communicator searches any directory that this variable specifies. The local user location, if it exists, overrides the network location.

On all platforms, the plugins default subdirectory or folder is in the same directory as the Communicator application. Users can install plugins in this directory manually, by using either an installation program or Communicator's JAR Manager facility. The plugins currently installed can be viewed under Netscape by selecting the 'help' menu option and the "About Plug-ins" menu item. This gives a list of installed plugs ins currently registered with the Netscape Browser and

identifies the 'default' plug in that is run for all file types not registered with another plug in:

Netscape Default Plug-in
File name: C:\PROGRA~1\Netscape\COMMUN~1\Program\plugins\npnul32.DLL

Default Plug-in

Mime Type	Description	Suffixes	Enabled
*	Netscape Default Plug-in	*	Yes

There is usually no other plug in that is registered for the file type .HTM or .HTML as this type being handled by the default services of Netscape. The plug in architecture does not prevent this happening and allows a plug-in that is activated when a file of type .HTM or .HTML is loaded. A dynamic link library (DLL) file is developed that conforms to the plug-in conventions as documented in the "Netscape Communicator Plug-in Guide" with a number of known function entry points. The DLL file is placed in the appropriate directory on platform 10. When it starts up, Communicator checks for plug-in modules in the plug-in directory for the platform and registers them. It determines which plugins are installed and which types they support through a combination of user preferences that are private to Communicator and the contents of the plugins directory. When Netscape is initialising it sweeps this directory and for each plug-in DLL that it finds it calls the NPP_Initialise function entry point in the DLL file. It also examines the DLL's version information.

On Windows, the plugin's directory is located in the same directory as the Communicator application and has a 8.3 filename beginning with NP and ending with .DLL. The Windows version information for the plug-in DLL determines the MIME types, file extensions, file open template, plug-in name, and description. In the MIME types and file extensions strings, multiple types and extensions are separated by the "|" character, for example: video/quicktime|audio/aiff|image/jpeg

The version stamp of the plug-in DLL contains the following lines so that Communicator recognises the plug-in,:

File Extents: for file extensions

MIME Type: for MIME types

Language: for language in use

In order to display an example of this use Windows Explorer to get to the C:\Program Files\Netscape\ Communicator\Program\Plugins> directory

and right click on one of the np*.DLL files - use the pop-up menu to view the DLL's properties. Up will come a notebook, click on the Version Page Tab -> You will see fields called "File Extents" and "MIME Type" these fields are decoded by Netscape using the convention above.

By creating a Netscape plug in using the instructions in the plug-in SDK and setting the values in the plug-in DLL's properties information to indicate that this plug is for file extensions of .HTM and .HTML we will cause Netscape to communicate with this plug in when files of this type are loaded.

2) Capturing mouse or keyboard events to control mouse or keyboard events to control web page downloads.

In order to capture mouse events that are driven from the HTML page the plug-in 30 operates as a 'full page' plug-in. A full page plug-in is visible but not part of an HTML page. The server looks for the media (MIME) type registered by a plug-in, based on the file extension, and starts sending the file to the browser. Communicator looks up the MIME type and loads the plug-in if it finds a plug-in registered to that type. This type of plug-in completely fills the Communicator page and are commonly used for document viewers, such as Adobe Acrobat. Plug-in 30 receives mouse events that take place while the mouse pointer is over it's own display window. The following events are visible to the plug-in

WM_PAINT
WM_LBUTTONDOWN
WM_LBUTTONUP
WM_LBUTTONDBLCLK
WM_RBUTTONDOWN
WM_RBUTTONUP
WM_RBUTTONDBLCLK
WM_MBUTTONDOWN
WM_MBUTTONUP
WM_MBUTTONDBLCLK
WM_MOUSEMOVE
WM_KEYUP
WM_KEYDOWN
WM_SETCURSOR
WM_SETFOCUS
WM_KILLFOCUS

These events will be passed to the plug-in via the "int16 NPP_HandleEvent(NPP instance, NPEvent *event);" function entry point defined in the plugin's DLL. In the embodiment the user holds down the control key to indicate that a background download of a URL. This causes

a WM_KEYDOWN event to take place for the control key. Plug-in 30 maintains a flag that represents whether the control key is down or up.

3) Requesting pages for download in cache memory without displaying

When the download of a new page is requested and a plug-in is registered for that page type then browser 20 will create a new 'stream' that represents the stream of data in the web page. To tell the plug-in when a new stream is created, browser 20 calls a NPP_NewStream method. This method also determines which mode it should use to send data to the plug-in via a return parameter in the function call.

The NPP_NewStream method has the following syntax:

```
NPError NPP_NewStream(NPP instance, NPMIMEType type,  NPStream *stream,
NPBool seekable, uint16* stype);
```

The instance parameter refers to the plug-in instance receiving the stream; the type parameter represents the stream's MIME type. The stream parameter is a pointer to the new stream, which is valid until the stream is destroyed. The seekable parameter specifies whether the stream is seekable (true) or not (false). Seekable streams support random access (for example, local files or HTTP servers that support byte-range requests). The plug-in can set the type output parameter type to various transmission modes and the embodiment chooses the mode based on whether the control key is down or not. If the control key is currently down (background download is requested) the type NP_ASFILEONLY is selected. The plug-in gets full random access to the data using platform-specific file operations. Browser 20 saves stream data to a local file in the cache, and, when the stream is complete and delivers the path of the file to the plug-in through a call to NPP_StreamAsFile. The URL will not be launched in a browser window or replace the contents of the current window. This mode will fetch the page to the Netscape cache without causing it to be displayed on a window. If the control key is not held down then the type of the fetch will be NP_NORMAL which will allow for the default behaviour of the page download.

4) Maintaining a list of pages placed in the cache to enable display in response to the user selecting the 'forward' and 'reverse' option in the user.

Plug-in 30 has visibility of all HTML pages requested, those downloaded in the background and their cache filenames. By monitoring the requests, downloads and file names the plug-in can maintain a sequential list of web pages and URLs in the order they were requested.

5) Triggering the display of a particular page.

As Netscape will have populated its cache for those pages subject to a NP_ASFILEONLY type of download plug-in 30 responds to the chosen user interface action (e.g. WM_KEYDOWN on < Control > and < -> >) to make the call back into the browser to display the next page using the NPN_GetURL function (defined below).

```
NPError NPN_GetURL(NPP instance,
                   const char* url,
                   const char* target);
```

NPN_GetURL is used to request browser 20 to load a URL into a particular HTML window or frame for display, or to deliver the data of that URL to the plug-in instance in a new stream. When the browser attempts to display the URL it will check it's cache as usual and because the page has already been downloaded to the cache it will find it there.

Operation of the plug-in

Using the plug-in 30 to preload selected web pages, the user may enhance present web searches as now described.

A) user enters search criteria.

B) user is presented with list of results.

C) using the plug-in selection mechanism (for instance holding the control key while selecting an item) the user can select a number of links while the browser view remains focused on the list of results and the browser stays 'live' to accept further requests.

D) as the requested URLs arrive at the browser the 'forward-arrow' of the browser stops being greyed out, indicating that there is a page present in the cache memory 24 'forward' from the present web page.

E) The user can continue making selections or make use of the forward arrow once finished. The 'forward-arrow' will grey whenever the user is at the latest requested page present in the cache and ungrey as any new 'background upload' pages arrive.

Referring to Figure 3A a more detailed description of the operation of the plug-in is given. On start up of the platform the operating system, browser and plug-in are loaded into operational memory from disk and initialised. In the course of Internet browsing the user is presented with an HTML screen containing multiple URLs from for example a search or a bookmark list. The user selects a URL (step 100). The browser checks whether the web page is already in cache memory (step 102) and displays the page if present (step 104). If not so present the plug-in checks for a background select (or preload select) (step 106), adds the URL (step 108) to the navigation list 28 and requests preloading of the page (step

110) into cache memory 24. If the user selects a foreground load (at step 106) then the browser proceeds as known by requesting the page at the same time as it is displayed (step 112).

After a URL is loaded (see Figure 3A) into the navigation list without having completely loaded the web page information into the cache memory it is possible to move from page to page by selecting 'Backwards' or 'Forwards' icons or options. In the present embodiment the cache memory is of a size to allow several pages of web page information, it is finite and web page information for previous links will eventually be overwritten. Hence 'navigation' through the list may require reloading of pages.

Referring to Figure 3B the process of web page information arriving at a web browser is depicted. Once a URL is pre-selected the navigation list is directly updated with the URL (108) and the web page information is received by the browser (step 114). It is then saved to the cache memory (step 116). Once all the information for a page is received icons are updated (step 118). Forward button 39B on the browser is normally grey indicating that there is no forward page to move to, when the page is pre-selected it is ungreyed to indicate that there is a forward page to move to. It may be ungreyed when the web page is fully loaded to indicate a fully loaded page or alternatively when the page is preselected to indicated a new page forward. The cache index is updated (step 120) as the web page information is added to the cache. An additional feature to the embodiment, a cached page table, is updated (step 122) with details of complete cached pages. This allows the user to jump ahead to fully received pages in the navigation list immediately from a pull down list of the pages with fully received pages indicated (for example by highlighting). The cache page table is a flag set in the navigation list 28 (e.g. by hitting the forward button 31A).

The following is an illustration of prefetching according to the embodiment. A user is browsing page 38 (see Figure 1) which web information is displayed from the data indicated by 32B (indicated in Figure 1 by darker lines) in cache memory 24. The web data files 32B are referenced in cache index reference 34B and this index is indicated by URL 36B (all indicated by darker lines in Figure 1) in the navigation list. The user may pre-select a new URL (step 100) and the URL 36A is added to the navigation list (step 108) and the web page information requested (step 110) and loaded into cache at 32A. An indication of when the web page is fully loaded is made by ungreying the Forward button 39B but the web information 32A is not displayed until the user selects the URL 36A in the navigation list 28.

The known browser feature that allows one to right-click on the forward and back arrows to see what pages are in the ring would be clearly useful. The browser could easily use this to show files requested and not yet downloaded in a manner that fits naturally with the current behaviour. This option is displayed in Figure 4 (the user right clicked on the Forward button of Internet Explorer). When one right clicks on the 'forward' or 'back' buttons, one is presented with a list of sites one has visited. These sites are already cached and will display immediately. In the envisioned embodiment of this idea the forward button would display a list of sites logically 'forward' in the users navigation path, these would include sites as today and sites requested with the background cache file download mechanism that have already arrived and are available in highlighted (or darker) text and sites that have been requested for background cache download that have not yet fully arrived in greyed out text.

In summary, this specification relates to the Internet world wide web page downloading and in particular relates to browser prefetching of web pages. There is disclosed a method and system of browsing an HTML page on the Internet comprising preloading pages corresponding to selected URLs into browser cache memory without loading them into display memory or effecting the ability to select further URLs. The method loads a first HTML page into browser cache memory from a network server via a proxy server and the Internet and loads said memory into display memory for displaying said the page. A user may then select multiple URLs on the HTML page without causing them to be displayed but causing then to be loaded as selected or on mass into cache memory not display memory or proxy.

Now that the invention has been described by way of a preferred embodiment, various modifications and improvements will occur to those person skilled in the art. Therefore it should be understood that the preferred embodiment has been provided as an example and not as a limitation.

CLAIMS

1. A method of browsing a "markup language" browser page from a network comprising:

loading a first "markup language" page into cache memory from a network server;

loading said page into display memory for displaying said the page;

providing means for user selecting multiple URLs on the "markup language" page; and

preloading pages corresponding to selected URLs into browser cache memory from at least one network server.

2. A method according to claim 1 further comprising, on selection of a URL and before preloading the corresponding page, updating a navigation list with the URL.

3. A method according to claim 2 further comprises updating the navigation list with information as to whether a URL on the list is fully loaded.

4. A method according to claim 1,2 or 3 further comprising, on selection of a URL and before preloading the corresponding page into cache memory, updating a cache index file with the location of the page in cache.

5. A method according any of claims 1 to 4 wherein pages corresponding to selected URLs are not displayed during preloading so that the user may view the original page until finished checking all the URLs.

6. A method according to claim 2 further comprising providing a drop down navigation list for selecting a page reference in the list so that the corresponding page can be displayed.

7. A method according to claim 6 further comprising providing 'forward' and 'back' GUI buttons for selecting the previous or next page reference in the list and displaying the corresponding page.

8. A method according to any one of claim 1 to 7 further comprising preloading selected pages directly into cache memory from an TCP/IP network

9. A system for browsing an HTML page on a TCP/IP network comprising:

means for loading a first HTML page into browser memory from a network server;

means for loading said memory into display memory for displaying said page;

means for selecting multiple URLs on the HTML page; and

means for preloading pages corresponding to selected URLs into browser memory without loading them into display memory.

10. A computer program product comprising a computer usable medium having computer readable program code means embodied in the medium for processing work items in a data processing system, the program code means comprising:

code means for causing the data processing system to load a first HTML page into browser memory from an network server;

code means for causing the data processing system to load said first HTML page into display memory for displaying said page;

code means for causing the data processing system to provide means for selecting multiple URLs on the HTML page; and

code means for causing the data processing system to preload pages corresponding to selected URLs into browser memory without loading them into display memory.

The Patent Office

INVESTOR IN PEOPLE

Application No:	GB 9911739.2	**Examiner:**	Julyan Elbro
Claims searched:	1 to 10	**Date of search:**	21 January 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4A (AMU, AUDB)

Int Cl (Ed.7): G06F 17/30

Other: ONLINE: COMPUTER EPODOC JAPIO WPI Selected Internet sites

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X, Y	WO 97/46955 A1 AT&T see abstract and fig. 3.	X: 1, 4, 5, 8 to 10 Y: 2, 3, 6, 7
Y	HCLU, "Netscape Navigator", published 1996, available from http://libsun1.jr2.ox.ac.uk/training/netscape.html see especially the section "Back and Forward".	2, 3, 6, 7

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 December 2000 (21.12.2000)

PCT

(10) International Publication Number
WO 00/77664 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US00/15938

(22) International Filing Date: 8 June 2000 (08.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/330,904 11 June 1999 (11.06.1999) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/330,904 (CON)
Filed on 11 June 1999 (11.06.1999)

(71) Applicant *(for all designated States except US)*: **LIBERATE TECHNOLOGIES LLC** [US/US]; 2 Circle Star Way, San Carlos, CA 94070-6200 (US).

(71) Applicant *(for US only)*: **BOUCHER, Antoine** [CA/CA]; 32-270 North Centre Road, London, Ontario N6G 5E2 (CA).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **MCRAE, Paul, E.** [CA/CA]; 141 Chesham Place, London, Ontario N6G 3T7 (CA). **SCHEYEN, Peter, G., N.** [CA/CA]; 20 Inverary Road, London, Ontario N6G 3L6 (CA).

(74) Agent: **BRUCKNER, John, J.**; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, 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: METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

(57) Abstract: Various embodiments of the invention provide increased speed and decreased computer processing for playing and navigating multimedia content by using two types of data objects for displaying the multimedia content. The data object type includes rendered multimedia content data. The second data object type provides semantic content corresponding to the rendered multimedia content. The storage medium in which these two types of data objects are contained is referred to as a rendered cache. The semantic content can include locations, sizes, shapes, and target universal resource identifiers of hyperlinks, multimedia element timing, and other content play instructions. The very fast play of content stored in the rendered cache is due to the elimination of the steps of laying out the content, rendering the content, and generating the semantic representation of the content. These steps are required each time the content is played after retrieval from a conventional cache. The only steps required for playing content from the rendered cache are to read the rendered content, read the semantic content, restore the semantic representation, and play the content. A traditional web browser visiting a web site that resides in a rendered cache provides an almost instantaneous display of the web site. The caching mechanism provided by various embodiments of the invention is independent of content file format and the stored semantic content file format. As long as a client application, such as a content browser, can recognize and play the multimedia content and recognize and interpret the semantic content, the application can realize the benefits provided by the rendered cache.

WO 00/77664 A2

METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates generally to improvements in computer systems. More particularly, the invention relates to methods, apparatus, and systems storing multimedia content such as audio, text, image, and graphical content in a cache directory.

Discussion of the Related Art

Prior art graphics processing storage medium, sometimes called a cache system, is known to those skilled in the art. For example, a conventional caching system is typically composed of a small fast storage device that contains a "snapshot" of information originally received from a larger, slower source. The snapshot is considered by the particular implementation to be the most relevant information to the processing occurring during the current time period.

In the context of Internet content, a "cache" is a file, database, directory, or set of directories disposed in a computer file system. The cache stores content that has been previously retrieved generated or otherwise produced. Internet browsers and editors use cache directories to store content. The cached content is used in place of remote content whenever possible in order to decrease retrieval latencies. Therefore, many web browsers and text editors save Internet and other text and graphical content in a cache directory in order to reduce access times. This content is usually stored in its original form [for example, hypertext markup language (HTML) and accompanying images].

A problem with this technology has been that to view content based on the image data stored in the cache typically requires layout and rendering of the data. If the data upon which content is based does not change, the process of rendering need only occur once to a display buffer. When information is

changed, the information must be re-rendered to reflect the desired change. For complex graphics scenes re-rendering can require massive processing for only incremental changes in the scene or particular graphic. The layout and rendering processes are time consuming and require processor resources. Therefore, what is required is solution that provides play of multimedia content more efficiently in terms of time and processor resources.

Heretofore, the requirements of timely and processor efficient play of multimedia content have not been fully met. What is needed is a solution that simultaneously addresses these requirements. The invention is directed to meeting these requirements, among others.

SUMMARY OF THE INVENTION

A primary goal of the invention is to provide timely and processor efficient display of multimedia content. In accordance with these goals, there is a particular need for a storage medium that includes multimedia content and the semantic content of the multimedia content. A storage medium including both the multimedia content and the semantic content is referred to herein as a rendered cache.

For various embodiments of the invention, the semantic content can include locations, sizes, shapes, and target universal resource identifiers of hyperlinks, multimedia element timing, and other content play instructions. The very fast play of content stored in the rendered cache is due to the elimination of the steps of laying out the content, rendering the content, and generating the semantic representation of the content. These steps are required each time the content is played after retrieval from a conventional cache. The only steps required for playing content from the rendered cache are to read the rendered content, read the semantic content, restore the semantic representation, and play the content.

A traditional web browser visiting a web site that resides in a rendered cache provides an almost instantaneous display of the web site. The caching mechanism provided by various embodiments of the invention is independent of

content file format and the stored semantic content file format. As long as a client application, such as a content browser, can recognize and play the multimedia content and recognize and interpret the semantic content, the application can realize the benefits provided by the rendered cache. Thus, it is possible to simultaneously satisfy the above-discussed requirements of timely and processor efficient display of multimedia content, which, in the case of the prior art, are not simultaneously satisfied.

A first aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for storing multimedia data. The method for storing multimedia data comprises detecting multimedia content, generating a semantic representation of a rendered representation of the multimedia content from the play instructions, storing the rendered representation in a storage medium, and storing data corresponding to the semantic representation in the storage medium. The multimedia content includes play instructions and at least one multimedia element. The at least one multimedia element includes at least one of graphical images, audio, text, and full motion video. The play instructions include at least one of timing of the multimedia content and ordering of the multimedia content. The semantic representation describes at least one of characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation.

A second aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for storing multimedia data. The method for storing multimedia data comprises detecting multimedia content including layout instructions, and laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content. The method also includes rendering the multimedia content according to the rendering instructions to produce the rendered representation, storing the rendered representation in a storage medium, and storing data corresponding to the semantic representation in the storage medium.

A third aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for retrieving multimedia data. The method for retrieving multimedia data comprises processing resources of a first computer of the at least one computer detecting a request for requested multimedia content, and processing resources coupled with the first computer determining whether data corresponding to the requested multimedia content is disposed in a storage medium. The storage medium is coupled with the first computer and includes rendered representations of multimedia content and semantic content. Embodiments according to the third aspect of the invention also include responding to a determination that data corresponding to the requested multimedia content are disposed in the storage medium by retrieving a rendered representation of the requested multimedia content; and retrieving semantic content corresponding to the requested multimedia content.

A fourth aspect of the invention is implemented in an embodiment that is based on a rendered cache comprising a storage medium, and an indexing mechanism adapted to store and retrieve a rendered representation of the multimedia content formatted for rapid play and semantic content of the multimedia content.

A fifth aspect of the invention is implemented in an embodiment that is based on a client. The client comprises processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations, and processing resources adapted to respond to detecting the rendered representation of the multimedia content and the semantic content by playing at least a portion of the rendered representation according to the semantic content.

A sixth aspect of the invention is implemented in an embodiment that is based on a system for using multimedia content. The system comprises web crawler processing resources adapted to access the multimedia content from source data storage, rendering processing resources, and a rendered cache as described above as the fourth aspect of the invention. The rendering processing

resources are adapted to generate a semantic representation of a rendered representation of the multimedia content, and format the semantic representation as semantic content, and render the multimedia content into the rendered representation, the rendered representation is formatted for rapid play.

A seventh aspect of the invention is implemented in an embodiment that is based on a system for accessing multimedia content. The system for accessing multimedia comprises a rendered cache as described above as the fourth aspect of the invention, and rendering processing resources adapted to convert the multimedia content into the rendered representation, the rendered representation is formatted for rapid play, and create a graphical representation of the multimedia content.

An eighth aspect of the invention is implemented in a method for playing multimedia content. The method comprises retrieving a rendered representation of the multimedia content from a storage medium, and retrieving semantic content of the rendered representation from the storage medium. The method includes browser processing resources reading the rendered representation and the semantic content, and the browser processing resources restoring a semantic representation based on the semantic content. The method includes the browser processing resources transmitting an active portion of the rendered representation to a client, and transmitting an active portion of the semantic content corresponding to the active portion of the rendered representation to the client. The active portion of the rendered representation is one of a portion of the rendered representation presently being played, and a portion of the rendered representation to be played rapidly after transmitting. The method also includes client processing resources detecting the active portion of the rendered representation and the active portion of the semantic content, and the client processing resources playing the active portion of the rendered representation.

BRIEF DESCRIPTION OF THE DRAWINGS

These, and other, goals and aspects of the invention will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. Various embodiments of the invention are illustrated in the drawings accompanying and forming a part of this specification, wherein like reference characters (if they occur in more than one view) designate the same parts. It should be noted that the features illustrated in the drawings are not necessarily drawn to scale.

FIG. 1 illustrates a schematic block diagram of a conventional method for retrieving and playing multimedia content, appropriately labeled "PRIOR ART".

FIG. 2 illustrates a schematic block diagram of a process overview for retrieving and playing multimedia content using a rendered cache, representing an embodiment of the invention.

FIG. 3 illustrates a schematic block diagram including render process details, representing an embodiment of the invention.

FIG. 4 illustrates a schematic block diagram including play process details, representing an embodiment of the invention.

FIGS. 5A-5B illustrate screen shots of portions of a Toronto Exchange Internet page, representing an embodiment of the invention.

FIG. 6A illustrates the timing of play of different multimedia elements for an example of multimedia content that does not require layout, representing an embodiment of the invention.

FIGS. 6B-6D illustrate different images included in the multimedia content not including layout example representing an embodiment of the invention.

FIG. 7 illustrates a communications system including a rendered cache, representing an embodiment of the invention.

DESCRIPTION OF PREFERRED EMBODIMENTS

The invention and the various features and advantageous details thereof are explained more fully with reference to the non-limiting embodiments that

are illustrated in the accompanying drawings and detailed in the following description of preferred embodiments. Descriptions of well-known components and processing techniques are omitted so as not to unnecessarily obscure the invention in detail. It should be understood, however, that the following description, while indicating preferred embodiments of the invention and numerous specific details thereof, is given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the invention without departing from the spirit thereof, and the invention includes all such modifications.

Methods, apparatus and systems are described for storing multimedia content using a process for caching fully rendered documents in a way that significantly increases content viewing speeds, navigation in a hyperlink document, while decreasing processing requirements.

Definitions

The following terms are used in the description of various embodiments of the invention provided herein.

Content: Text and graphical information that require a layout and/or rendering process in order to be viewed on a computer, television or other display device. Other terms for content include web-page, document, Internet content, hypertext markup language (HTML), eXtensible Markup Language (XML), and Television Markup Language (TVML). Content can also include non-graphical information such as audio.

Content Browser: A computer program designed to retrieve, display or navigate content. Examples include Internet web browsers, HTML/XML/Standard Generalized Markup Language (SGML) editors, word processors, and Internet web proxies.

HTML: The de facto Internet content standard. HTML includes a set of markup rules that describe the layout of Internet content. Browsers use this markup to layout and render the HTML for viewing computer monitors, televisions, or other displays.

Markup: Notation used to describe the syntactic and semantic features

of a content document.

Multimedia Content: Multimedia elements used for playing a presentation for a user. The multimedia elements can include graphical images (including rendered HTML), audio, text, and full motion video.

Navigation: The process of selecting an indexing indication, such as a URI in the form of a hyperlink, from displayed content to access further content.

Presentation: Content that references at least one multimedia element. Presentations include play instructions that can be used to define the timing, order, and position of the multimedia plays. The play instructions can include the size, shape and target of all hyperlinks, information on interactive elements (like HTML forms), and Meta values.

Render: The process of generating a graphical representation of data that can be viewed on a display. For example, web browsers render HTML pages into graphical images that can be viewed on a computer monitor or television. Also the process of generating or converting multimedia data (images, audio, text, full motion video) into a format that can be played.

Rendered Cache: Various embodiments of the invention use the concept of a rendered cache to mean a cache of content that is not only generated (or retrieved) from a multimedia content data source, such as the Internet; but also is rendered and ready for rapid play. The rendered cache can include two types of objects: multimedia content and semantic content. The multimedia content stored in the rendered cache is content that has been rendered and is ready for very quick display. Semantic content includes a description of the semantic features or representation of the rendered content. Examples of semantic features include the location, size, shape and target of hyperlinks, the timing, location, and size of animated graphics interchange format (GIF) frames, the size and relative location of HTML frames, information on HTML forms, HTML meta values, presentation play timing, and other play instructions. A more detailed description of the rendered cache is provided in the Process Description section below.

Semantic Representation: A description of the characteristics,

attributes, logical structure, and features of multimedia elements (or objects) that form a rendered representation of multimedia content, or a portion thereof. The data can also describe the relationships between different multimedia elements within a particular presentation portion, and the way various elements of the multimedia content are accessed and manipulated. The semantic representation is typically generated during the layout process and is structured such that the semantic representation can be saved as formatted and indexed semantic content in a file or database, and rapidly restored from the semantic content. The semantic content can be stored along with the multimedia content or as one or more separate indexed files. The semantic representation is independent of the format of the stored semantic content. The Document Object Model (DOM) is one type of semantic representation and is adapted for use with HTML and XML documents.

TVML: Some embodiments of the invention (including the VirtualModem™ presentation system provided by Interactive Channel, Inc. located in London, Ontario, Canada) use an XML language called television markup language (TVML) to describe multimedia content. TVML includes markup to describe how to play multimedia content. The multimedia content can include text (including HTML), graphical images, audio, text, and full-motion video. TVML can include markup to describe when each multimedia component should be played relative to the other multimedia components.

URI: A Universal Resource Identifier (or URI) is an Internet standard term for all types of names and addresses that refer to content. The term URI encompasses terms such as filename, hyperlink, and Universal Resource Locator (URL).

VMML: An XML markup language (called VMML - VirtualModem™ Markup Language) used to store semantic representations of rendered multimedia content by various embodiments of the invention, such as the VirtualModem™ presentation system.

XML: A markup language used to describe other markup languages, such as HTML and TVML.

Process Description

Various embodiments of the invention include methods, implemented in at least one computer, for storing and retrieving multimedia data. These methods navigate and play multimedia content with increased speed and decreased computer processing by using different types of data objects to represent the multimedia data. A first data object type includes pre-rendered multimedia content data. A second data object type includes a semantic representation of the pre-rendered multimedia content. These data object types can be stored as separate files or can be contained in the same file.

Prior art methods for retrieving and playing multimedia content are represented by Figure 1, which includes a traditional cache 110. After detecting a request to play multimedia content (at step 120), retrieving processing resources, such as those disposed in a web browser, retrieve the corresponding multimedia content data. A traditional web browser, such as Netscape Navigator, Netscape Communicator, or Microsoft® Internet Explorer, when coupled with a traditional cache 110, then performs the steps described below in response to each and every play request 120.

After retrieving the content (e.g., the HTML content description), the content is read (step 130) from either a traditional cache 110, an Internet 105, or another content data source. Processing resources disposed in a computer can layout the content (step 140), e.g., according to the content's HTML description. During the layout 140, the processing resources generate rendering instructions 140A and derive a semantic representation 140B of the multimedia content. Note that for some embodiments, layout 140 is not required. For these embodiments, the semantic representation 140B can be generated from play instructions, as shown in Figure 3 (at step 315).

Content browsers can use the semantic representation 140B to determine location, size, shape and targets of hyperlinks; and content play instructions. The semantic representation 140B can also be used to describe other interactive presentation elements, e.g., HTML forms. The semantic features corresponding

to the depicted graphical representation generated for play when using traditional content browsers coupled with traditional caches 110 persist only as long as the content is being viewed. Because the semantic features must be present whenever the multimedia content is played, and because traditional caches 110 store the multimedia content in a non-rendered original form, traditional browsers must re-render the graphical representations each time a user requests the content, as shown in Figure 1.

A render engine then renders the multimedia content (at step 150) according to the rendering instructions 140A to form rendered content 160 (otherwise referred to herein as the rendered representation of the multimedia content). Finally, a multimedia play engine uses both the rendered content 160 and the semantic representation 140B to play the rendered content (at step 170). For multimedia content including images, the playing 170 includes displaying the rendered image on a user screen according to the semantic representation 140B. The rendered content 160 is also referred to herein as a rendered representation of the multimedia content.

In prior art implementations, playing 170 occurs after layout 140 (or other process in which the semantic representation 140B is generated), and rendering 150 have been completed. Rendering 150 and generation of the semantic representation 140B require a relatively long time between when the play request 120 is received and when multimedia content is played 170 compared to the time required to play content using various method embodiments of the invention. Also, greater data processing is required for rendering 150 and generation of the semantic representation 140B for the multimedia content than the processing required for play 170 using various method embodiments of the invention.

The details of retrieving (step 260), rendering 150, and playing 170 multimedia content for some embodiments of the invention are illustrated in Figs 2 through 4. Fig. 2 provides an overview of the retrieving 260, and the playing 170 processes for multimedia content, e.g., HTML content, using a rendered cache 201. The methods can be implemented in at least one computer

having one or more programs for retrieving and playing multimedia content. The benefits of using the rendered cache 201 for subsequent access to the same multimedia content are also described below.

The rendered cache 201 includes not only rendered content 160 (which can include image data) but also some means of reconstructing the semantic representation 140B of the multimedia data. The reconstruction of the semantic representation 140B can be done using proprietary image formats or separate files that describe the semantic features. This semantic representation 140B can include locations, sizes, and destinations of hyperlinks, descriptions of animations or other dynamic content, and other "meta" information. Meta information can include tagging, refresh (client pull replacement), Meta lists, and platform for Internet content selection (PICS) association labels.

Some embodiments of the invention (including VirtualModem™ interactive presentation systems provided by Interactive Channel Technologies, Inc. located in London, Ontario, Canada) use an XML language called VMML to store the semantic content. The VMML semantic content can include markup to represent the following semantic features of the rendered content 160:

1. Location, size, shape, and target indices (such as URI) of hyperlinks,
2. Size and relative location of HTML frames in the rendered image,
3. Size, location, and timing of animated GIFs,
4. Size, location, and type of HTML form elements,
5. Timing of multimedia content elements, and
6. Other play 170 instructions.

Proper use of content from a rendered cache 201 eliminates the steps of generating a semantic representation 140B, layout 140 (when needed), and rendering 150. On the other hand traditional web browsers using traditional caching mechanisms must perform these steps before playing 170 the content. Eliminating these steps reduces the time and use of processing resources required for playing 170 the multimedia content.

A rendered cache 201 can include of two types of data objects: multimedia content and semantic content. The content can be stored 320 in any

format (i.e., the caching mechanism is independent of file format). Typically, the layout 140 and/or rendering 150 processing resources format the semantic presentation 140B for storage in the rendered cache 201 as semantic content. Alternatively, the layout 140 and/or rendering 150 processing resources can transfer the semantic presentation 140B to rendered cache 201 server processing resources which then format the semantic presentation into semantic content to be stored in properly indexed files for retrieval 260. For some embodiments of the invention, content browsers (and/or other client applications using content from the rendered cache 201) can include processing resources, such as a program, for detecting the format of the rendered content 160 and for viewing multimedia content.

When a request for content is received (step 210) the content browser can determine (step 220) whether a rendered representation of the content already exists in the rendered cache 201. The browser can also determine (step 230) whether the content in the rendered cache 201 is outdated. The content request received at step 210 can be provided to the browser using a file target index, such as a Universal Resource Indicator (URI).

Once it has been determined whether an updated rendered representation of the requested data already exists in the rendered cache 201, a first and simpler processing path indicated in Figure 2 can be followed. Because the rendered cache 201 contains valid rendered content corresponding to the request, browser engine processing resources can simply read the semantic content and the rendered content 160, restore the semantic representation (step 240), and then play 170 the rendered content corresponding to the requested content.

The process proceeds along a second path if the server-based system (or other processing resources coupled with the rendered cache 201) has determined that the requested content is not in the rendered cache 201, or that the content stored in the rendered cache is outdated. Along the second path, the browser submits a request to retrieve the content from an updated source (e.g., the Internet 105) and retrieves the content (step 260). After the updated content has

been stored 320 in the rendered cache 201 (as shown in Figure 3), the process continues along the first method path as long as the stored content does not become out of date. The first method path, as shown in Figure 2, includes reading the semantic content and the rendered content 160, and restoring the semantic representation (step 240), to play 170 the rendered content for each request -

Figure 3 illustrates a more detailed depiction of the layout 140 and storing 320 processes. After retrieving 260 the requested multimedia content with layout instructions and/or play instructions, the computer determines whether layout 140 is required for the multimedia content (step 310). The semantic representation 140B of the semantic features is generated during the layout 140 process, or generated from play instructions (step 315) when no layout is required.

After rendering 150, the rendered content 160 is stored 320 in the rendered cache 201. Similarly, after construction of the semantic representation 140B, the semantic representation is formatted as semantic content and also stored 320 in the rendered cache 201.

If the rendered cache 201 stored only the resulting rendered content 160, the description of the hyperlinks, display instructions and other semantic content would be lost. The semantic content can take the form of flat text files, XML or other structured files, or other proprietary formats. Some embodiments of the invention format the semantic content according to an XML language called VirtualModem™ Markup Language (VMML) to represent the semantic features of HTML pages and TVML presentations. The rendered content 160 and semantic content can be stored in a traditional cache, a database, a file system or other storage media. The underlying file system can be used to store the content in a directory and file hierarchy that represents the rendered cache 201.

The rendered content 160 stored 320 in the rendered cache 201 can include images, audio, text, full motion video, animations, etc. The content is stored in the rendered cache 201 regardless of its format [i.e. the rendered cache

201 can store binary large objects (blobs) or format-independent objects]. The format in which the semantic content is stored is independent of the rendered cache 201 mechanism. The content browsers and other client applications that access the rendered content 160 stored in the rendered cache 201 include processing resources adapted to recognize the format and interpret the semantic content appropriately.

According to some embodiments of the invention, content browsers and other client applications include processing resources to recognize and play 170 the rendered content 160 after the corresponding format-independent objects are retrieved 260 from the rendered cache 201. Some embodiments of the invention, including various VirtualModem™ presentation systems, can render HTML pages into a proprietary image format, called a fat macroblock (FMB), that is suitable for display on televisions. FMB's are described in greater detail by United States patent application serial number 09/287,235, entitled "System and Methods for Preparing Multimedia Data Using Digital Video Data Compression", filed April 6, 1999, having inventors Antoine Boucher, Paul E. McRae, and Tong Qiu, the entire contents of which are hereby incorporated herein by reference as if fully set forth herein.

In the case where the content is not missing but is outdated, the entire content can be retrieved 260, or just the outdated portions can be retrieved. By retrieving 260 only outdated portions some savings can be gained in the rendering 150 step by eliminating the need for a full rendering. For example, perhaps only an animated image on an HTML page has changed in the requested content. The rendering system can detect this situation and render 150 only the new animation rather than the entire page.

Once the needed portion of the request content has been retrieved 260, the content is rendered 150 before it is played 170. The retrieved content is handed to a rendering system that typically performs the following actions:

1. Laying out 140 of the content according to the appropriate rules (e.g., HTML rules)

2. Rendering 150 the content according to the rendering instructions

140A, thereby producing presentation data (e.g., for an MPEG image formatted as an FMB- or set of images for HTML frames) that represent the fully rendered representation of the content (e.g., the HTML page). The page may also have other graphical elements created for such things as animated GIFs.

3. Generating 315 a semantic representation 140B of the semantic features. Generally, the layout engine or the render engine creates the semantic representation 140B from the layout 140 or play instructions. For an HTML page, the semantic representation 140B can include the location, size, shape, and target of all HTML anchors (links to other HTML pages), the timing, location, and size of animated GIF frames, the size and relative location of HTML frames, information on HTML forms that can be accessed from the page, and HTML meta values.

4. Storing 320 the rendered content 160 [e.g., MPEG image(s)] in the rendered cache 201 using an appropriate index, e.g., a URI. The semantic content is also stored 320 in the rendered cache 201 using an appropriate index. In some embodiments, the semantic content can be stored 320 in an XML-based format so that it can be easily parsed and restored (e.g., in step 240) in the future. After the rendering system is finished, the rendered content 160 can be provided to the user by simply reading and restoring 240 and playing 170 the content.

The "format" of the semantic representation 140B is determined by the engine that generates the semantic representation (e.g. Netscape Communicator and Microsoft ® Internet Explorer use the DOM). This internal semantic representation 140B is then stored as a physical entity (semantic content) in the rendered cache 201. The format of semantic content is adapted for the browser engine that reads the semantic content for play 170. The format of the semantic content is sufficiently detailed for the browser engine to create its own semantic representation 140B. The semantic representation 140B in the browser engine can be the same internal format that the layout/render engine uses or the semantic representation in browser can have a different format.

As shown in Figure 4, when a request is received for content already in

the rendered cache 201 the rendering system process can be skipped entirely. The following simple steps are all that is involved to play 170 content already in the rendered cache 201.

1. Read the semantic content and the rendered content 160, and restore the semantic representation 140B from the semantic content stored in the rendered cache 201, e.g., the VMML description.

2. Play 170 the rendered content 160 on the user's screen according to this semantic representation 140B.

Some multimedia content, such as an HTML web page, does not fit entirely on a user's screen at once. For such partial page displays, the browser can use the semantic representation 140B to determine which portion of the page should be displayed, and for some embodiments which subset of the hypertext links are selectable on the page portion. An example of this scrolling is described below in the "HTML Page with Layout" example below.

Retrieving Content from the Rendered Cache

When a content browser, or other client application, requests a target index, such as a URI, the rendered cache 201 mechanism first looks in the rendered cache for a rendered representation of the content. The caching mechanism provides a means to search and retrieve this content based on the content's indexing indication. Examples of cache retrieval mechanisms include database queries, simple index files, file system directory structures, or traditional browser caches.

If the rendered content 160 can be found in the rendered cache 201, the content will be displayed very quickly. The semantic representation 140B of the rendered content 160 will be restored using the semantic content stored in the rendered cache 201 (i.e. the semantic features need not be computed again before the rendered content is played). For example, some embodiments restore the semantic representation 140B of a rendered HTML page by reading the VMML formatted semantic content.

If the content browser cannot locate a rendered representation of the

multimedia content in the rendered cache 201, or the browser determines that the content is out of date, then the content can be retrieved 260 (either from a traditional cache 110, from the Internet 105, or from another content source) and rendered 150. The retrieval 260 and rendering 150 results in at least one new rendered cache 201 entry that can be used the next time the multimedia content is accessed.

A system that uses a rendered cache 201 will, after determining that no rendered representation is in the cache, perform the same steps as described above. That is, the HTML source will be read and the page laid out 140. The resulting rendering instructions 140A are followed but rather than displaying the page (or, alternatively, in addition to displaying) the page the rendering will be stored as a graphical image in the rendered cache 201. The semantic content (describing the location, size, and target URI of the single hyperlink on the image) is also stored in the rendered cache 201. The next time and every subsequent time the browser receives a request to view this URI, the browser simply reads the semantic content and the rendered content 160, restores the semantic representation 140B, and displays the rendered content. Thus, the use of the rendered cache 201 saves the cost of processing for layout 140, generation 315 of the semantic representation 140B, and rendering 150B. For more complicated HTML pages this savings can be substantial.

Examples

Specific embodiments of the invention are further described by the following, non-limiting examples which will serve to illustrate in some detail various features of significance. The examples are intended merely to facilitate an understanding of ways in which the invention may be practiced and to further enable those of skill in the art to practice the invention. Accordingly, the examples should not be construed as limiting the scope of the invention.

An HTML Page with Layout using a Distributed Server-Based Content

System

Some embodiments of the invention provide storage 320, retrieval 260 and/or play 170 of HTML pages. One embodiment of the invention is represented by the Toronto Stock Exchange (TSE) HTML homepage illustrated by Figures 5A and 5B.

For this embodiment, the "content browser" can be broken up into a distributed server-based content preparation and viewing system. The viewing system can include a display device, e.g., a television, and a digital set-top box (such as a General Instruments DCT-2000).

For some HTML page embodiments, the set-top box has neither the processing nor the storage resources needed to render 150 or cache content. The set-top box typically does have the capability to decode and play MPEG images and Dolby AC-3 audio, and some limited graphics capabilities in order to do text and simple graphical overlays. For these embodiments, all access to rendering 150 processing resources and content stored in the rendered cache 201 is done at the server. These embodiments are described in greater detail in the "System" section below.

In other embodiments, the set-top box, or other addressable processing equipment, can have processing resources and storage medium capable of rendering 150 and caching the content. In response to the server-based system receiving a request to view some content with the URI http://www.tse.com/ and determining that the content is either not in the rendered cache or is outdated, the server system browser requests retrieval of the TSE web page and any graphical elements the TSE web page references.

Once the web page and graphics have been retrieved 260 (either from a traditional cache 110 or from the Internet 105) the browser requests that the content be laid out 140 and rendered 150. The rendering system creates an MPEG representation (in FMB format) of the rendered web page. Because MPEG is the only image format the GI DCT-2000 recognizes, we use MPEG in this example. The rendering system can also generate other FMB files representing animated GIF frames, if animated GIFs were referenced in the

HTML page. The rendering system also creates a semantic representation 140B of the page including the location, shape, size, and target of all hyperlinks; location, size, and timing of animated GIF frames; HTML form information; and HTML meta information.

The FMB files are stored 320 in the rendered cache 201 using the URI of the HTML page ("www.tse.com") as an index. The semantic content is also stored 320 in the rendered cache 201 using the URI as an index. The semantic content is stored in an XML format called VMML. For distributed systems embodiments, e.g., the VirtualModem™ system, the internal semantic representation 140B for the layout/render engine is different than the semantic representation 140B for the browser engine (although these semantic representations 140B are conceptually equal). The stored semantic content (in the form of VMML for VirtualModemTM) is detailed enough to allow for "information transfer" so that two different semantic representations 140B can be used.

Once the rendered content 160 (FMBs) and semantic content (VMML) are stored 320 in the rendered cache 201, the browser can then read and restore the semantic representation 140B based on the VMML file. Using this semantic content the web page can be displayed.

The first screen capture (Fig. 5A) of the TSE homepage shows the top portion of the page. The rectangular highlight box 510 in the top left corner indicates that the user can select the first hyperlink for viewing. Users can press arrow keys on their remote control to move from one link to another link on the page. The browser provides enough information for the set-top box to draw the highlight box 510 and to navigate the page from link to link using the arrow keys.

Eventually the user may scroll past the bottom of the screen. The set-top will then inform the server-based browser that a scroll is required and the browser will then determine from the semantic content which new portion of the rendered MPEG should be visible and which new subset of the hyperlinks is now selectable.

The second screen capture (Fig. 5B) illustrates the TSE homepage after a scroll down. The user can continue to view the same page scrolling around and viewing the content in the fashion described above. However, once a link is selected the browser is informed of the corresponding new URI request and the retrieval 260 (or read and restore 240) process are initiated again after the browser receives a play request 120.

An HTML Page with Layout using a Self-Contained Content System

Some embodiments do not use the distributed server-based content preparation and viewing system described in the above example (HTML Page with Layout). Instead, these embodiments are self-contained content systems with layout 140, rendering 150, and play 170 processes all combined in a single computer program. Netscape Communicator and Microsoft ® Internet Explorer are examples of such embodiments.

Such "traditional" browsers can also use the invention to reduce retrieval 260 and playing 170 time, and decrease processor usage. Traditional web browsers have long used caching technologies to minimize the need to use slower content retrieval 260 methods such as network access. These browsers store the original retrieved content in a cache database. When a request to view content is received, the browser searches the traditional cache 110. If the content not in the cache then the browser retrieves the content from an alternate source (such as the Internet 105). Visiting web sites that reside in a rendered cache 201 results in almost instantaneous display of the web site content rather than the usual delay (due to the cost of layout, rendering and creation of semantic context) that is normally seen.

Whether or not the content was found in the traditional cache 110, the content is then read and laid out 140 according to the rules of HTML. Laying out 140 produces rendering instructions 140A and a semantic representation 140B of the content. The page is then rendered 150 to a graphical format (typically a bitmap) and played 170 according to the semantic representation 140B. These steps are performed each and every time the content is requested.

For the self-contained embodiments of the invention, when a request for content is received the browser will search in the rendered cache 201 to determine whether a rendered representation of the content is available. If the content is not in the rendered cache 201, or if the rendered content is found to be outdated, then the content must be requested from an alternate source (such as the Internet 105, or a traditional cache 110). Once the content is received it will go through the same layout 140, rendering 150, and generation 315 of the semantic representation 140B steps as these browsers do now.

The difference is that once the rendering 150 and generation 315 of the semantic representation 140B is complete the rendered content 160 and the semantic representation 140B are stored in the rendered cache 201.

Once the content is stored in the rendered cache 201, then each time the browser receives a request for this content, the browser simply reads and restores 240 the semantic representation 140B and plays 170 the rendered content 160 according to this semantic representation. The format of the rendered content and semantic representation are entirely up to the browser. It is recommended that the rendered content be stored in a "native format". That is, a format that the browser can immediately recognize and does not have to convert to a recognized format. It is also recommended that the format for the semantic representation 140B be rich enough to cover all the various semantic elements that HTML can describe. VMML is a good example of such a format. For self-contained systems the format of the internal semantic representation 140B is likely to be the same for both the layout 140/render 150 and browser portions of the program.

Another related embodiment that could benefit from the invention is what is commonly referred to as a "web proxy". A web proxy is a computer program that retrieves content on behalf of content browsers. Various embodiments of the invention enable the web proxy to only retrieve 260 content from the Internet 105 for the first request, while all future requests for the content from browsers using the proxy use the locally cached version.

Note that in either the distributed or self-contained scenarios, the task of

converting to and from the stored semantic content format is up to the relevant engines (layout 140/render 150 engine for storing and browser engine for retrieval 260.

An alternate scenario could involve the layout/render engine transferring the semantic representation (through some communications medium) to a "rendered cache server" that converts the representation into semantic content. This server would also receive request to retrieve content from the cache and would read the semantic content, convert it to an appropriate internal representation and then transfer this representation. In this case the task of converting to and from semantic content is entirely up to the "rendered cache server". In
practice, this approach is less flexible than alternative approaches.

In the case where the web proxy and the content browsers all have access to the same storage or have access to a fast internal communications network the web proxy could perform the layout 140, rendering 150, and generation 315 of semantic representation 140B steps on behalf of the content browsers. In such a scenario, when a content browser receives a request for content, the content browser can either look directly in the rendered cache 201 or query the web proxy for the rendered content 160. The browser can then simply read and restore 240 the semantic content and display the rendered content 160 accordingly. This use of the web proxy allow for the use of very small and efficient web browser implementations since all the resources for layout 140, rendering 150, and generation 315 of the semantic representation 140B are external of the browser.

An intelligent web proxy can pre-render the content that it downloads in order to offset the rendering cost in browsers. This approach is especially beneficial in situations where client computing resources are limited. A key application of this approach is in the emerging market of set-top devices and other network computers. These devices typically have tightly constrained resources and do not presently provide true web browsing. The use of a rendered cache proxy would offload the process of layout 140, rendering 150,

and generation of the semantic representation 140B.

In some embodiments, word processing programs can store 320 rendered documents in a rendered cache 201 for faster loading and previewing. Using the rendered cache 201 for storing 320 word processing documents also enables programs other than the word processor to preview the content without using proprietary plug-ins or libraries.

Multimedia Content with Play Instructions

Some embodiments of the invention provide storage 320, retrieval 260 and/or play 170 of multimedia content. The multimedia content can include images, audio, text, graphics, and full motion video, all of which can be timed to play at different moments. This multimedia content can have a means of referencing other multimedia content in a manner similar to HTML hyperlinks. Some embodiments of the invention, including the VirtualModem™ system from Interactive Channel, use an XML language called TVML to represent the play instructions of a multimedia presentation. TVML can include markup to represent the following play instructions of the multimedia content:

1. Timing of multimedia content playing;
2. Order of multimedia content playing;
3. Size and location of multimedia content; and
4. Location, size, shape, and target URI (or other index) of hyperlinks.

One embodiment of the invention is represented by the News Menu TVML presentation illustrated by Figs. 6A through 6D. Fig. 6A illustrates a timeline representing how the News Menu TVML presentation should be played. Figs. 6B through 6D show the images that make up the News Menu TVML presentation. As in the previously described embodiment (HTML with layout) the "content browser" can be broken up into a distributed server-based content preparation and viewing system.

The server-based system can receive a request to view some content with the URI http://www.virtualmodem.com/news.tvml and then determined that the content is either not in the rendered cache 201 or is outdated. The

browser can respond to this circumstance by submitting a request to retrieve the TVML presentation and any multimedia elements referenced by the presentation. Once the presentation and its multimedia elements have been retrieved 260 (either from a traditional cache 110 or from the Internet 105), the browser requests that the content be rendered 150. In this case, layout 140 is unnecessary and the rendering 150 can be limited to converting the multimedia content into a format that the set-top recognizes. In the case of the GI DCT-2000 images and full motion video are converted to MPEG formatted data and audio is converted to Dolby AC-3 formatted data.

The rendering system can also generate 315 a semantic representation 140B of the page from the TVML play instructions. The semantic representation 140B can include context such as the relative play times and order of the multimedia content; the location, shape, size, and target of all hyperlinks; and TVML meta information. The rendered content 160 can be stored 320 in the rendered cache 201 using the URI of the presentation ("www.virtualmodem.com/news.tvml") as an index. The appropriately formatted semantic content based on the semantic representation 140B is also stored in the rendered cache 201 using the URI as an index. For some embodiments of the invention, the semantic content is stored in a VMML format.

Once the rendered content 160 and semantic content (VMML) are stored in the rendered cache 201, the browser can read and restore 240 the semantic representation 140B from the VMML file in which the semantic content is disposed. Using this semantic representation 140B the presentation can be displayed.

Figure 6A shows the start time and duration that each image of the presentation should be played, and illustrates the start time and duration of the accompanying audio. The presentation plays 170 from t_0 to t_3.

Figures 6B through 6D show each of the images used in the for the News Menu TVML presentation. The first image 650 of the presentation, shown in Fig. 6B, includes a single circle with a small diameter around the top

of the transmitter to indicate that a signal is being sent from a transmitter. As shown by the first time line 610, the first image 650 is shown from t_0 to t_1.

The second image 660 of the presentation, shown in FIG. 6C, includes three circles around the top of the transmitter to indicate that the signal will be received by the user sooner than when the first image 650 was displayed. As shown by the second time line 620, the second image 660 is shown from t_1 to t_2.

The third image 670 of the presentation, shown in FIG. 6D, includes a first hyperlink that is enclosed by a rectangular highlight box 510 to indicate that the first hyperlink "World News Update" is presently available for selection. As shown by the second time line 630, the third image 660 is shown from t_2 to t_3. A user can press arrow keys disposed on the user's remote control devices to move from link to link in the third image 670. The browser provides enough information for the set-top box to draw this rectangle and to navigate using the arrow keys from link to link. If the multimedia content is larger than the physical screen then it becomes possible to scroll in the same manner as described in the "HTML with layout" example. As shown by the fourth time 640, the accompanying audio plays 170 for the entire duration of the News Menu TVML presentation.

In the News Menu TVML presentation example only the third image 670 of the presentation contains hyperlinks. However, in other embodiments of the invention, any of the earlier images can also contain hyperlinks. The browser can update the client (set-top box or other addressable processing equipment) whenever the semantic representation 140B (e.g. hyperlink information or image display duration) changes.

The presentation can play 170 until all multimedia objects have been played. The user can continue to view the last image of the presentation in the same manner as for HTML pages. The user can also manipulate the remote control VCR functions to rewind, fast-forward, or pause the presentation. However, once a hyperlink is selected the browser will be informed of the new URI request and the content retrieval process will start again with a request for

content.

Systems for Storing, Retrieving and Playing Multimedia Content

Some embodiments of the invention include systems for storing 320, retrieving 260 and playing 170 multimedia content using a rendered cache 201. Listed below are the key elements of a system that can implement various embodiments of the invention. Previous descriptions and examples mentioned in the "Distributed Server-Based Content System" section have illustrated the use of the invention in a distributed server-based system. In such a system the various complimentary components, such as those listed below, are typically found in separately running processors that can reside in a single computer or in multiple connected computers. Some embodiments, such as the VirtualModem™ system can include the following components:

Web crawler processing resources adapted to access multimedia content from source data storage. The multimedia data can include HTML and TVML content. The source data storage can include at least one of the Internet 105 and a web proxy cache.

Rendering processing resources adapted to generate semantic representation 140B of, and render 150 multimedia data, and can format the semantic representation as semantic content. In some embodiments, a rendering program can also be adapted to layout 140 the multimedia data.

Multimedia playing processing resources, such as an audio/video terminal server (AVTS), adapted to play multimedia content. Such play can include displaying images and playing audio and full motion video. Some embodiments of an AVTS are described in greater detail in United States patent application serial number 09/255,052, entitled "System and Method for Interactive Distribution of Selectable Presentations," filed February 22, 1999, and having inventors: Antoine Boucher, James Lee Fischer, and Allan E. Lodberg, the entire contents of which are hereby incorporated herein by reference as if fully set forth herein.

Browser processing resources adapted to interpret the semantic content

and control when and how the multimedia content should be played. The browser processing resources can act as the "control center" for the entire process. The browser processing resources can communicate with the web crawler, rendering, and the multimedia playing processing resources and coordinate the interactions of each of these.

A second group of embodiments discussed above in the "Self-Contained Content System" section contain all the required rendered cache 201 complimentary components in a single program. Netscape Communicator and Microsoft ® Internet Explorer are both examples of such single programs. Both of these browsers include programs adapted to retrieve 260 content from the Internet 105 or a proxy. Netscape Communicator and Microsoft ® Internet Explorer also have layout 140, rendering 150, and semantic representation 140B generating capabilities. They both have the ability to display the rendered content 160 to a computer monitor display and they both can interpret the semantic representation 140B.

FIG. 7 illustrates the components and features configured in a system for accessing multimedia content using a rendered cache 700 representing one embodiment of the invention. The system for accessing multimedia content using a rendered cache 700 includes the components and features described below, including: access to source content 710, at least one layout engine 720, at least one render engine 730, a rendered cache 201, at least one multimedia play engine 750, at least one browser engine 760, and a display 770. These components can be combined together to form one or more computer programs that implement the storing 320, retrieving 260 and playing 170 methods described above.

Source content is content that is not yet rendered. The source content can include HTML, XML, images, audio, text, and full motion video. Access to source content 710 can be through an Intranet, the Internet 105, a web proxy, or on local storage. Connections adapted to provide such access can be through any carrier capable of providing sufficient bandwidth for practical retrieval 260 the content, such as: digital subscriber line (DSL), cable modem, T-1, T-2, T-3,

OC-1 through OC-256, fiber distributed data interface (FDDI), E1 through E5, Ethernet, fast Ethernet, and Gigabit Ethernet. Access to source content 710 can also include processing resources adapted to use standard Internet protocols such as TCP/IP and HTTP, and to read files from a file system. The component providing access to source content 710 includes processing resources for retrieving the source content, such as the content fetch 715 resources shown in FIG. 7.

The system for accessing multimedia content using a rendered cache 700 can include layout processing resources, such as a layout engine 720, adapted to derive rendering instructions 140A from a content definition (e.g., HTML). The layout engine 720 can also derive a semantic representation 140B of the features of the content from the layout 140, or from the play 170 instructions. Netscape Communicator and Microsoft ® Internet Explorer both contain processing resources to perform HTML layout 140 as part of their overall functionality. Stand-alone layout engines 720 include Spyglass Device Mosaic, NGLayout from Mozilla, and Chimera. In some embodiments, processing resources other than the layout processing resources can be adapted to generate 315 the semantic representation 140B from play 170 instructions.

The system for accessing multimedia content using a rendered cache 700 can include rendering processing resources, such as a render engine 730, adapted to create a graphical representation of content that has been laid out 140 by the layout engine 720. The rendering engine 730 can also have the capability of converting content that does not require layout 140 into a form that is ready for rapid play 170.

Many layout engines 720 also include a render engine 730. Systems that have layout engines 720 that do not include a rendering 150 capability have a separate render engine 730 and typically specify the interface that a rendering engine must have (e.g., this is how Spyglass Device Mosaic works). Both Netscape Communicator and Microsoft ® Internet Explorer include rendering engines as part of their overall functionality. Both of these browsers render 150 the content into a bitmap that can be displayed on a computer monitor display

770. Some embodiments, such as the VirtualModem™ system, use their own custom render engine 730 that renders the content to MPEG files stored in FMB format.

The rendered cache 201 provides access to an indexed storage mechanism. The rendered cache 201 stores both the rendered content 160 and the semantic content so that these data objects can be easily retrieved 260 at a later time. The rendered cache 201 includes as indexing mechanism that can take a variety of forms including database queries, index files, file system directories.

The format of the rendered content 160 is independent of the storage mechanism. A format that requires little or no conversion at play 170 time (i.e. a "native format") provides greater time and processing savings using the rendered cache 201.

The format of the semantic content is also independent of the storage mechanism. The semantic content format used in the system for accessing multimedia content using a rendered cache 700 that fully captures all the semantic features of the rendered content 160 provides enhanced play results. The semantic content format can avoid unneeded complexity to ensure that the processing and time required to restore the semantic representation 140B are less than that required to layout 140 and re-render the content.

The system for accessing multimedia content using a rendered cache 700 includes multimedia play processing resources, such as a multimedia play engine 750, adapted to play the rendered content 160 on a display 770 device. The multimedia play engine 750 can read the rendered content 160 directly from the rendered cache 201 indexed storage mechanism, read the rendered content from memory, or otherwise receive the rendered content from an external source. Netscape Communicator and Microsoft ® Internet Explorer both contain, as part of their overall functionality, processing resources to display multimedia content to a computer display 770. Some embodiments, including the VirtualModem™ system, include a separate program that is part of the overall distributed system, called the AVTS, that is adapted to play multimedia

content to APEs.

The system for accessing multimedia content using a rendered cache 700 also includes a browser engine 760 adapted to interpret the semantic representation 140B of the rendered content 160 being played 170. The browser engine 760 can read the semantic content directly from the rendered cache 201 indexed storage mechanism, or interpret the rendered content 160 from memory, or otherwise receive the semantic content from an external source.

The browser engine 760 can be adapted to interpret the semantic features from the semantic content. In some embodiments, the browser engine 760 is adapted to control navigation of hyperlinks (i.e. determining from user input which content should be displayed next). The browser engine 760 also can determine which portions of the rendered content 160 should be played 170, and which corresponding portions of the semantic representation 140B are active (e.g., when scrolling an image).

The browser engine 760 can be included in commercially available software such as Netscape Communicator, Microsoft ® Internet Explorer, or any other browser engine that is adapted to perform the functions described above. Netscape Communicator and Microsoft ® Internet Explorer both contain, as a part of their functionality, processing resources adapted to interpret a semantic representation 140B [or Document Object Model (DOM) as both call it]. Both of these browsers use the DOM to determine which links are currently visible (and which others are scrolled off the screen), animated GIF timing and location, information about HTML forms and other HTML features. Some embodiments, such as the VirtualModem™ system, include a browser program that coordinates the retrieving 260 of content, layout 140 and rendering 150 of content, and playing 170 of rendered content. These browser embodiments can also contain processing resources for reading semantic content from the rendered cache 201 and restoring the semantic representation 140B.

The above engines (layout, render, play, and browser) are all at least loosely coupled. That is, they need not be part of the same program but there needs to be some form of communication between them all. This

communication can take a variety of forms including inter-process communication (such as shared memory, pipes, or messaging protocols), or shared files. Some embodiments, such as the VirtualModem™ system, use a communications protocol built on a user data protocol (UDP) to communicate between the various engines. Netscape Communicator and Microsoft ® Internet Explorer include all the engine components in the same program.

There is no requirement that any of the above system components be directly tied together (i.e. included in the same program). However, there are advantages to tightly coupling certain components. For example, it is more efficient to couple the layout engine 720 and the render engine 730 in the same program. In such a scenario the rendering instructions 140A resulting from layout 140 process can be used directly by the rendering engine 730 component. If the layout engine 720 and the render engine are separate programs, then some intermediate form of rendering instructions (e.g. either a file or data passed over a network) would have to be used.

Only the component responsible for accessing source content 710 needs to include processing resources to access the communications carrier and the underlying communications protocol. It is not required that the other engine components have these processing resources.

The layout engine 720 and the render engine 730 has access to the rendered cache 201 storage mechanism since they read the rendered content 160 and the semantic content.

The multimedia play engine 750 has access to at least the rendered content 160 portion of the rendered cache 201 storage mechanism. The browser engine 760 has access to at least the semantic content portion of the rendered cache 201 storage mechanism. Both the multimedia play engine 750 and the browser engine 760 can have full access to the entire rendered cache 201 storage mechanism but at minimum they have access to their respective content.

Splitting access to the rendered content 160 and the semantic content allows for efficient distribution of the multimedia play engine 750 and browser engine 760.

Formatting the Semantic Content

Some embodiments of the invention use an eXtensible markup language (XML) language to format and store 320 semantic content in the rendered cache 201. Embodiments including the VirtualModem™ system use a markup language called VMML to format and store 320 semantic content in the rendered cache 201.

VMML contains elements to describe the semantic features of both HTML and TVML. TVML is another XML language originally based on synchronized multimedia integration language (SMIL) from the World Wide Web Consortium at http://www.w3.org/. The descriptive elements include:

1. Multimedia elements - The <img>, <audio>, <video>, and <text> elements are used to describe fully rendered multimedia objects. The <screen> element is used to describe fully rendered HTML. Each of these elements can include an optional start time using the "begin" attribute.

2. Aggregation elements - The <par> and <seq> elements are used to describe how the multimedia elements are played. Elements inside a <par> are played in parallel. The start times of multimedia elements in a <par> are relative to the beginning of the <par>. Elements inside a <seq> are played sequentially. The start times of multimedia elements in a <seq> are relative to the end of the previous element. Both the <par> and <seq> elements can define optional start times using the "begin" attribute.

For example, the following <par> element contains an <audio> and two <img> elements which are played in parallel (i.e. at the same time). The display of the second image is delayed by 5 seconds.

```
<par>
    <audio src="voice-over.ac3"/>
    <img  src="first-screen.fmb"/>
    <img  src="second-screen.fmb" begin="5.0s"/>
</par>
```

3. HTML elements - The <screen> element is used as a container for all the semantic information concerning a rendered HTML page. Elements allowed in a <screen> element include:

<frame> - contains attributes for defining the FMB (the rendered frame), size, and location relative to other frames of the HTML page;

<anchor> - each <frame> element can contain a list of <anchor> elements which describe the location, size, shape, and target of HTML hyperlinks;

<form> - each frame can contain form elements which fully describe HTML forms;

<animation> - <frame> elements can contain animation elements that describe the timing, size and location of animated GIFs.

4. Non-display elements - The <title> and <meta> elements describe non-audiovisual features of the content. Examples of <meta> information include HTML refreshes, and expire metas.

5. Anchors - Information about non-HTML hyperlinks is also described in VMML <anchor> elements.

6. Applets - The <applet> element instructs the browser to run other applications.

VMML contains other minor elements and a wide variety of attributes but the above list describes the major features. VMML is capable of describing all the various features of TVML and HTML in sufficient detail that the semantic representation 140B can be reconstructed after reading the semantic content from the rendered cache 201. The reconstruction of the semantic representation 140B includes simple tokenization (i.e. text parsing) using freely available tools such as sgml-lex (available from http://www.w3.org/). The parsing process is much faster and uses far fewer processor resources than the processes of layout 140 and rendering 150.

The term coupled, as used herein, is defined as connected, although not necessarily directly, and not necessarily mechanically. The term substantially, as used herein, is defined as approximately (e.g., preferably within 10% of, more preferably within 1% of, most preferably within 0.1% of).

Advantages of the Invention

A rendered cache 201 coupled with multimedia content render, play, and browser processing resources, representing an embodiment of the invention, can be cost effective and advantageous for at least the following reasons. The rendered cache 201 enables the play 170 of multimedia content in less time and using less data processing because the steps of layout and rendering are eliminated.

All the disclosed embodiments of the invention described herein can be realized and practiced without undue experimentation. Although the best mode of carrying out the invention contemplated by the inventors is disclosed above, practice of the invention is not limited thereto. Accordingly, it will be appreciated by those skilled in the art that the invention may be practiced otherwise than as specifically described herein.

For example, although the rendered cache 201 described herein can be a physically separate module, it will be manifest that the rendered cache 201 can be integrated into the apparatus with which it is associated. Furthermore, all the disclosed elements and features of each disclosed embodiment can be combined with, or substituted for, the disclosed elements and features of every other disclosed embodiment except where such elements or features are mutually exclusive.

It will be manifest that various additions, modifications and rearrangements of the features of the invention may be made without deviating from the spirit and scope of the underlying inventive concept. It is intended that the scope of the invention as defined by the appended claims and their equivalents cover all such additions, modifications, and rearrangements. The appended claims are not to be interpreted as including means-plus-function limitations, unless such a limitation is explicitly recited in a given claim using the phrase "means-for." Expedient embodiments of the invention are differentiated by the appended sub-claims.

CLAIMS

What is claimed is:

1. A method, implemented in at least one computer, for storing multimedia data, comprising:

detecting multimedia content, the multimedia content includes play instructions and at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

generating a semantic representation of a rendered representation of the multimedia content from the play instructions, the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation;

storing the rendered representation of the multimedia content in a storage medium; and

storing data corresponding to the semantic representation in the storage medium.

2. The method for storing multimedia data of claim 1, wherein the semantic representation includes at least one of:

a location of a hyperlink;

a size of the hyperlink;

a shape of the hyperlink;

a target index of the hyperlink;

a size of a portion of the multimedia content disposed in a rendered image;

a relative location of the portion of the multimedia content disposed in a rendered image;

sizes of animated formatted graphics;

locations of the animated formatted graphics;

timing of the animated formatted graphics;

browser commands for addressable processing equipment; and

data corresponding to the play instructions.

3. The method for storing multimedia data of claim 1, wherein:

one of the at least one computer includes rendering processing resources adapted to generate the semantic representation; and

storing the data corresponding to the semantic representation includes:

formatting the semantic representation into semantic content to enable rapid restoring of the semantic representation by a content browser; and

storing the semantic content.

4. The method for storing multimedia data of claim 3, wherein:

storing the semantic content includes the rendering processing resources transferring a semantic content file to the storage medium;

storing the rendered representation includes the rendering processing resources transferring at least one rendered representation file to the storage medium; and

the method includes indexing the semantic content file and the at least one rendered representation file based on at least one universal resource identifier, to form at least one indexed file.

5. The method for storing multimedia data of claim 4, including arranging the indexed files to enable a client application to retrieve the rendered representation and the semantic content, the client application includes the content browser.

6. The method for storing multimedia data of claim 1, including converting the multimedia content into the rendered representation of the multimedia content.

7. A method, implemented in at least one computer, for storing multimedia data, comprising:

detecting multimedia content including layout instructions, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

rendering the multimedia content according to the rendering instructions to produce the rendered representation;

storing the rendered representation in a storage medium; and

storing data corresponding to the semantic representation in the storage medium.

8. The method for storing multimedia data of claim 7, including:

detecting play instructions, and

wherein laying out includes incorporating the play instructions into the semantic representation, and the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content.

9. A method, implemented in at least one computer, for retrieving multimedia data, comprising:

processing resources of a first computer of the at least one computer detecting a request for requested multimedia content;

processing resources coupled with the first computer determining whether data corresponding to the requested multimedia content is disposed in a storage medium, the storage medium is coupled with the first computer, the

storage medium includes rendered representations of multimedia content and semantic content, the semantic content includes data corresponding to semantic representations derived from one of: play instructions for the rendered content, and layout of the multimedia content, the semantic representations describe at least one of: characteristics of the rendered representations, and relationships between different multimedia elements disposed in the rendered representations; and

responding to a determination that data corresponding to the requested multimedia content are disposed in the storage medium by:

retrieving a rendered representation of the requested multimedia content; and

retrieving semantic content corresponding to the requested multimedia content.

10. The method for retrieving multimedia data of claim 9, including restoring the semantic representation for the requested multimedia content using the semantic content corresponding to the requested multimedia content.

11. The method for retrieving multimedia data of claim 9, wherein determining whether data corresponding to the requested multimedia content are disposed in the storage medium includes searching the storage medium using a retrieval mechanism adapted to search and retrieve content based on an index corresponding to the requested multimedia content.

12. The method for retrieving multimedia data of claim 11, wherein a semantic representation corresponding to the requested multimedia content includes play instructions for the requested multimedia content, and the method includes playing the requested multimedia content according to the play instructions.

13. The method for retrieving multimedia data of claim 11,

including: responding to a determination that data corresponding to the requested multimedia content are not disposed in the storage medium by:

storing the data corresponding to the requested multimedia content in the storage medium;

retrieving the rendered representation of the requested multimedia content; and

retrieving the semantic content corresponding to requested multimedia content.

14. The method for retrieving multimedia data of claim 13 including restoring a semantic representation for the rendered representation of the requested multimedia content using the semantic content corresponding to requested multimedia content.

15. The method for retrieving multimedia data of claim 9 including responding to a determination that data corresponding to the requested multimedia content is not disposed in the storage medium by:

storing data corresponding to the requested multimedia content in the storage medium;

retrieving the rendered representation of the requested multimedia content; and

retrieving the semantic content corresponding to requested multimedia content.

16. The method for retrieving multimedia data of claim 10, wherein:

data disposed in the storage medium includes at least one file;

the at least on file includes the rendered representation of the requested multimedia content, and the semantic content corresponding to the requested multimedia content;

restoring the semantic representation is performed by a client application using the semantic content corresponding to requested multimedia content; and

the method includes:

a client computer requesting the requested multimedia content; and

the client application recognizing and playing the requested multimedia content from at least a portion of the at least one file.

17. The method for retrieving multimedia data of claim 9, wherein responsive to determining that data corresponding to the requested multimedia content are disposed in the storage medium, determining whether the data corresponding to requested multimedia content disposed in the storage medium require updating.

18. The method for retrieving multimedia data of claim 9, including, prior to retrieving the rendered representation of the requested multimedia content:

processing resources coupled with the first computer determining whether the data corresponding to the requested multimedia content disposed in the storage medium require updating:

responsive to a determination that the data corresponding to the requested multimedia content disposed in the storage medium require updating:

storing an updated version of the data corresponding to the requested multimedia content in the storage medium;

retrieving at least a portion of an updated version of the rendered representation of the requested multimedia content;

retrieving at least a portion of an updated version of the semantic content corresponding to the requested multimedia content; and

restoring a semantic representation for the requested multimedia content using the at least a portion of the updated version of the semantic content.

19. The method for retrieving multimedia data of claim 18, wherein

determining whether the data corresponding to the requested multimedia content disposed in the storage medium require updating includes at least one of:

comparing an expiry date tag for the data corresponding to the requested multimedia content disposed in the storage medium with a date corresponding to the updated version of the data corresponding to the requested multimedia content; and

comparing the data corresponding to the requested multimedia content disposed in the storage medium with a version of data corresponding to the requested multimedia content disposed in a different storage medium.

20. The method for retrieving multimedia data of claim 9, including, prior to retrieving the rendered representation of the requested multimedia content:

processing resources coupled with the first computer determining whether the data corresponding to the requested multimedia content disposed in the storage medium require updating; and

responsive to a determination that the data corresponding to the requested multimedia content disposed in the storage medium require updating:

storing an updated version of the data corresponding to the requested multimedia content in the storage medium including an updated version of the rendered representation of the requested multimedia content, and an updated version of the semantic content corresponding to the updated version of the rendered representation;

retrieving the updated version of the rendered representation of the requested multimedia content;

retrieving the updated version of the semantic content corresponding to the updated version of the rendered representation; and

restoring the semantic representation for the requested multimedia content corresponding to the updated version of the rendered representation using the updated version of the semantic content.

21. The method for retrieving multimedia data of claim 9, wherein:

the request includes an index corresponding to the requested multimedia content; and

determining whether data corresponding to the requested multimedia content is disposed in a storage medium includes searching the storage medium using a retrieval mechanism adapted to search and retrieve content based on the index corresponding to the requested multimedia content.

22. The method for retrieving multimedia data of claim 21, wherein the retrieval mechanism includes using at least one of:

a database query,

index files, and

a file system directory structure.

23. A rendered cache comprising:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

a rendered representation of the multimedia content formatted for rapid play, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation.

24. The rendered cache of claim 23, wherein the semantic content includes data corresponding to at least one of:

a location of a hyperlink;

a size of the hyperlink;

a shape of the hyperlink;

a target index of the hyperlink;

a size of a portion of the multimedia content disposed in a rendered image;

a relative location of the portion of the multimedia content disposed in a rendered image

sizes of animated formatted graphics;

locations of animated formatted graphics;

timing of animated formatted graphics;

browser commands for addressable processing equipment; and

data corresponding to play instructions.

25. The rendered cache of claim 24, wherein the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content.

26. The rendered cache of claim 23, wherein the indexing mechanism includes processing resources for converting the rendered content address into a unique index, the unique index is adapted to store and retrieve the rendered content and semantic content.

27. A client comprising:

processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation; and

processing resources adapted to respond to detecting the rendered representation of the multimedia content and the semantic content by playing at least a portion of the rendered representation according to the semantic content.

28. The client of claim 27, wherein the semantic content includes data corresponding to hyperlinks.

29. The client of claim 27, wherein:

the client includes a set-top box;

only a portion of the multimedia content, and only a portion of the semantic content are received by the processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations; and

the semantic content includes data corresponding to scroll commands.

30. A system for using multimedia content comprising:

web crawler processing resources adapted to access the multimedia content from source data storage, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

rendering processing resources adapted to:

generate a semantic representation of a rendered representation of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

format the semantic representation as semantic content; and

render the multimedia content into the rendered representation, the rendered representation is formatted for rapid play; and

a rendered cache including:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

the rendered representation of the multimedia content; and

the semantic content of the multimedia content;

31. The system for using multimedia content of claim 30 including:

a browser adapted to:

interpret the semantic content; and

control play of the rendered representation of the multimedia content, and

multimedia playing processing resources adapted to play the rendered representation of the multimedia content.

32. The system for using multimedia content of claim 30, wherein:

source data storage includes at least one of the Internet and a web proxy cache; and

the rendering processing resources are adapted to layout the multimedia content.

33. The system for using multimedia content of claim 31, wherein the browser processing resources:

communicate with the web crawler processing resources, the rendering processing resources, and the multimedia playing processing resources; and

are adapted to control the interactions of at least one of:

the web crawler processing resources;

the rendering processing resources, and

the multimedia playing processing resources.

34. A system for accessing multimedia content comprising:

a rendered cache including:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

a rendered representation of the multimedia content formatted for rapid play, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation; and

rendering processing resources adapted to:

convert the multimedia content into the rendered representation, the rendered representation is formatted for rapid play; and

create a graphical representation of the multimedia content.

35. The system for accessing multimedia content of claim 34 including: multimedia play processing resources adapted to:

read the rendered representation; and

play the multimedia content on a display; and

browser processing resources adapted to interpret the semantic content.

36. The system for accessing multimedia content of claim 34 including:

layout processing resources adapted to:

derive rendering instructions from a content definition;

lay out the multimedia content; and

generate a semantic representation of the multimedia content from lay out of the multimedia content; and

wherein the rendering processing resources use the rendering instructions to create the graphical representation.

37. The system for accessing multimedia content of claim 35, wherein the browser processing resources are adapted to control play of the multimedia content.

38. A method for playing multimedia content, comprising:

retrieving a rendered representation of the multimedia content from a

storage medium;

retrieving semantic content of the rendered representation from the storage medium, the semantic content including data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation;

browser processing resources reading the rendered representation and the semantic content;

the browser processing resources restoring a semantic representation based on the semantic content;

the browser processing resources transmitting:

an active portion of the rendered representation, the active portion of the rendered representation is one of: a portion of the rendered representation presently being played, and a portion of the rendered representation to be played rapidly after transmitting; and

an active portion of the semantic content corresponding to the active portion of the rendered representation;

client processing resources detecting the active portion of the rendered representation and the active portion of the semantic content; and

the client processing resources playing the active portion of the rendered representation.

39. The method for playing multimedia content of claim 38, wherein:

the client includes a set-top box;

the multimedia content includes an image having at least one hyperlink;

the semantic representation including at least one of:

a location of at least one hyperlink;

a size of at least one hyperlink;

a shape of at least one hyperlink; and

a target index of at least one hyperlink.

40. An apparatus for storing multimedia data, comprising:

means for detecting multimedia content including layout instructions, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

means for laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

means for rendering the multimedia content according to the rendering instructions to produce the rendered representation;

means for storing the rendered representation; and

means for storing data corresponding to the semantic.

41. A rendered cache comprising:

means for storing data corresponding to multimedia content, the data including:

a rendered representation of the multimedia content formatted for rapid play, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation

indexing means for storing and retrieving:

the rendered representation; and

the semantic content.

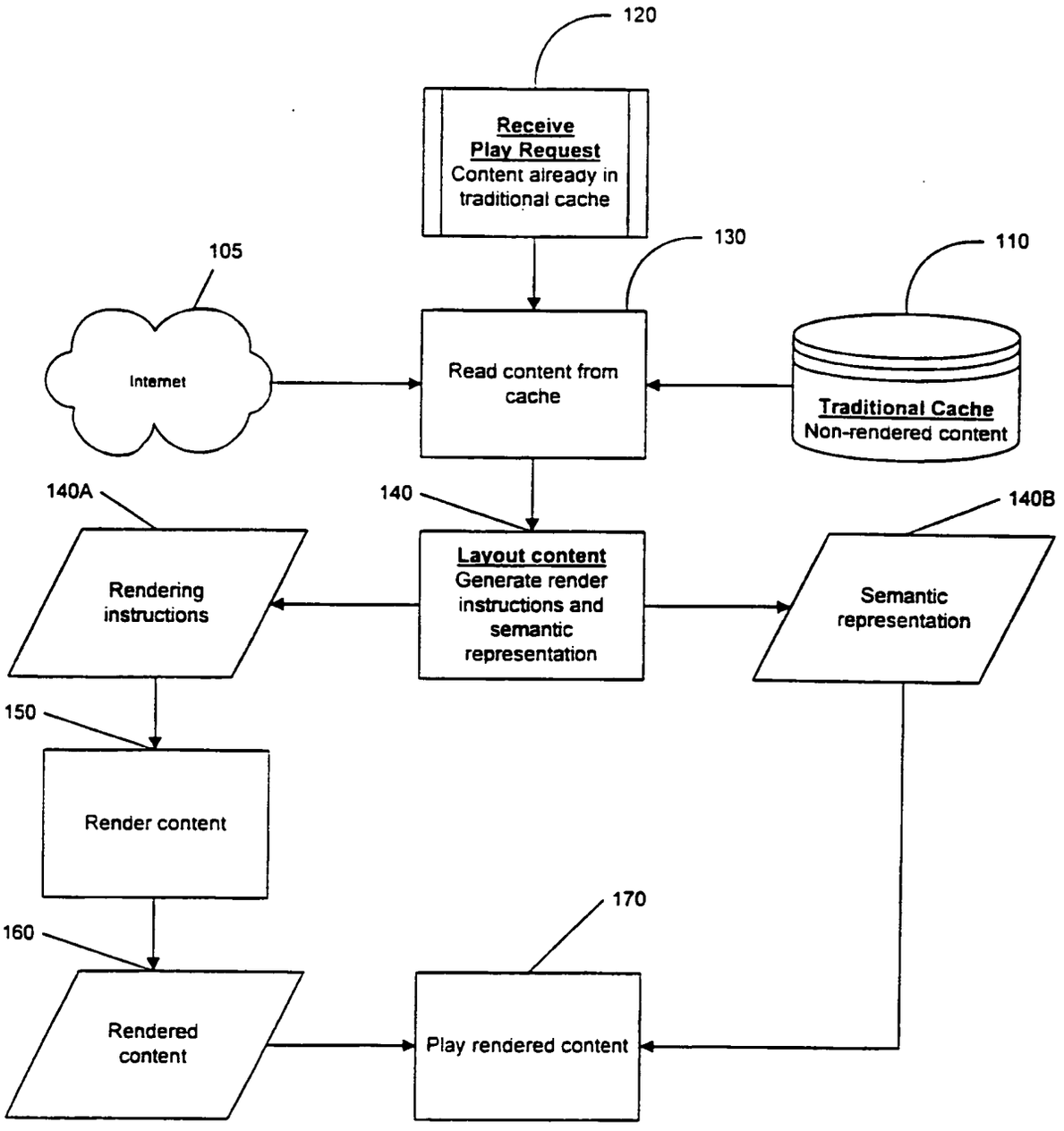


Fig. 1: Prior Art Process Details

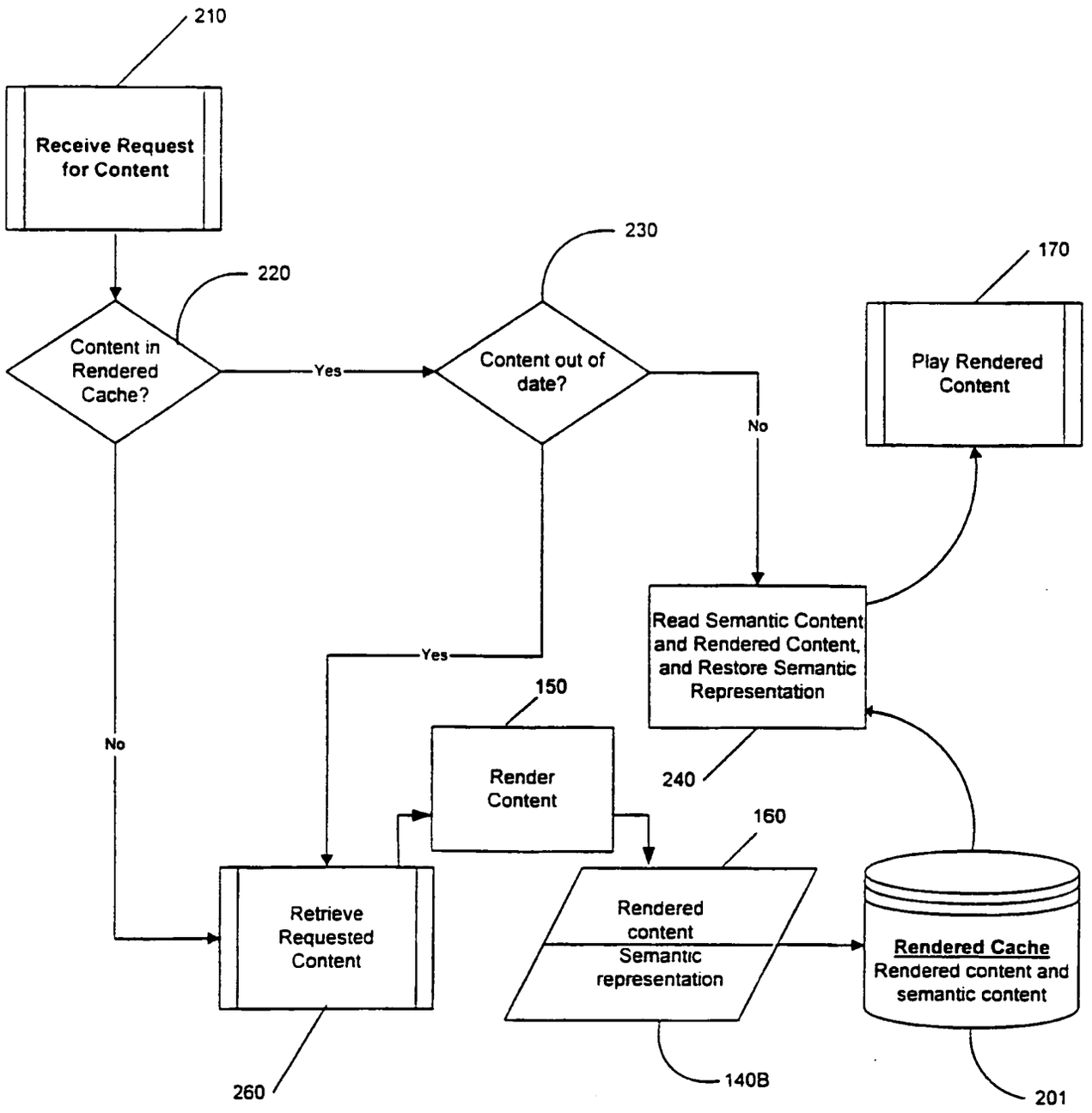


Fig. 2: Rendered Cache Process Overview

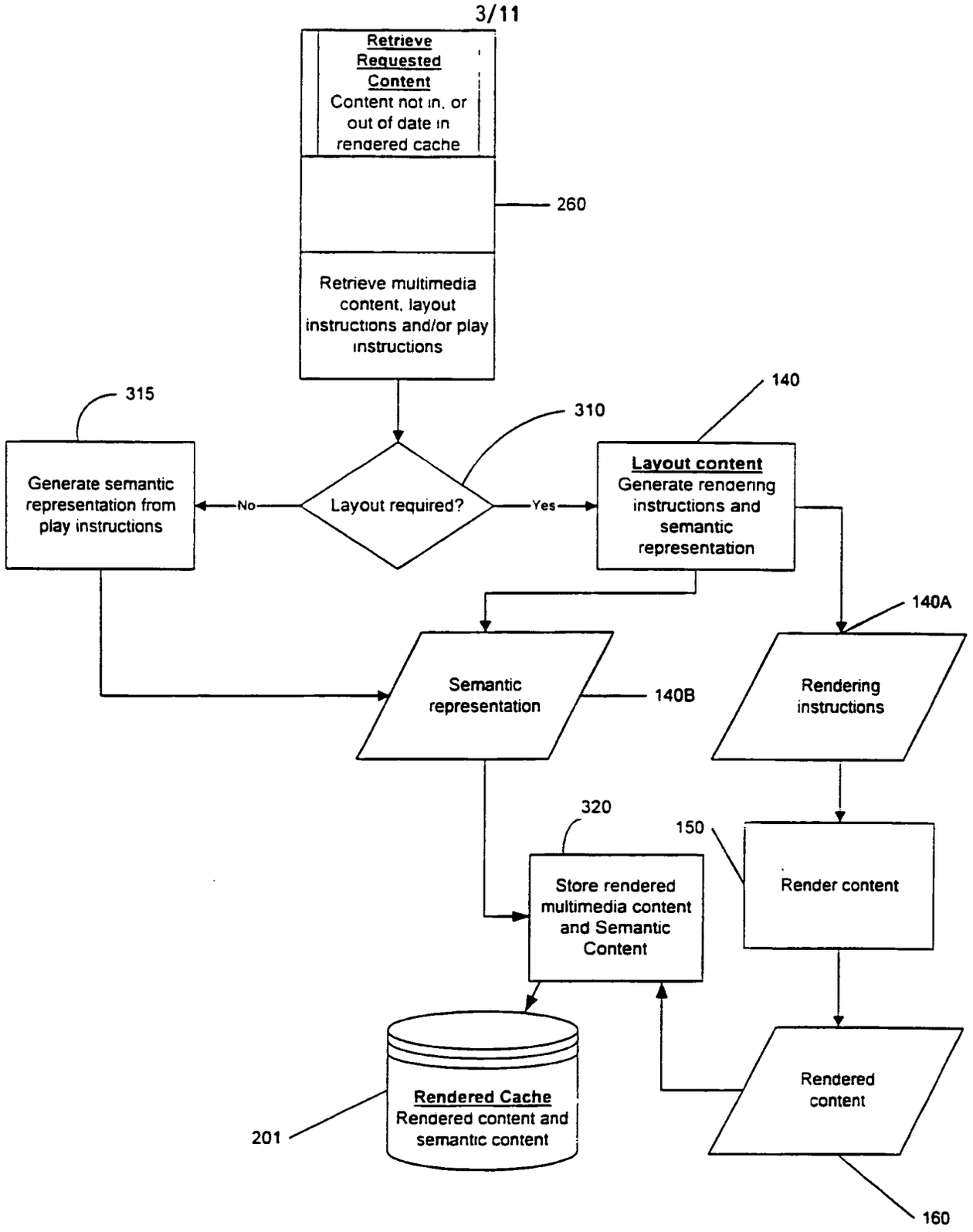


Fig. 3: Render Process Details

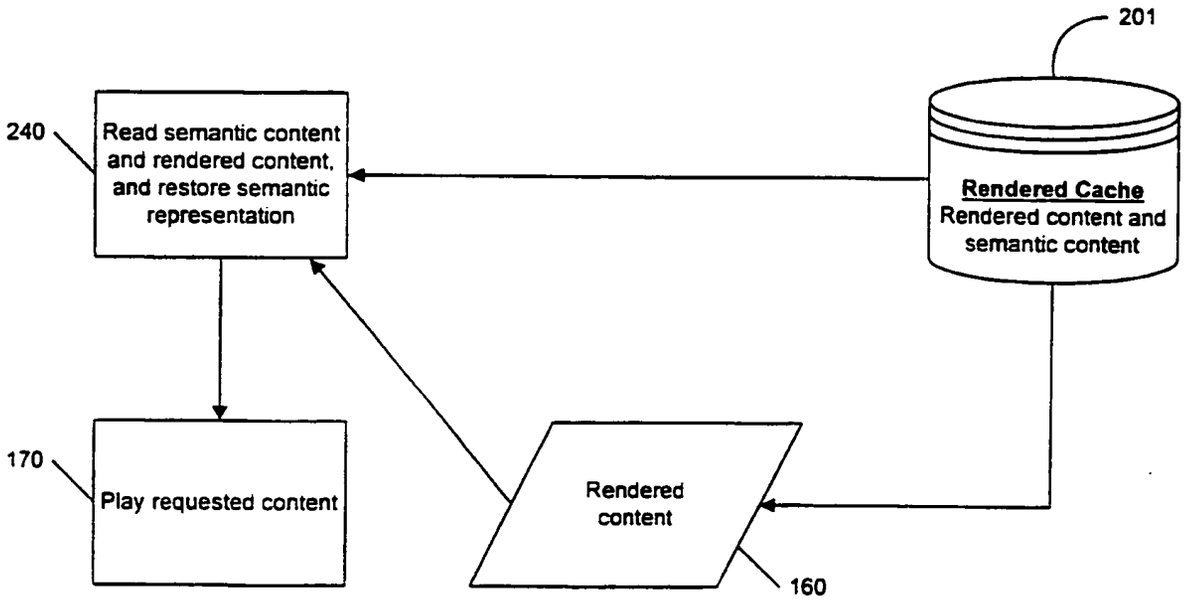


Fig. 4: Play Process Details

510

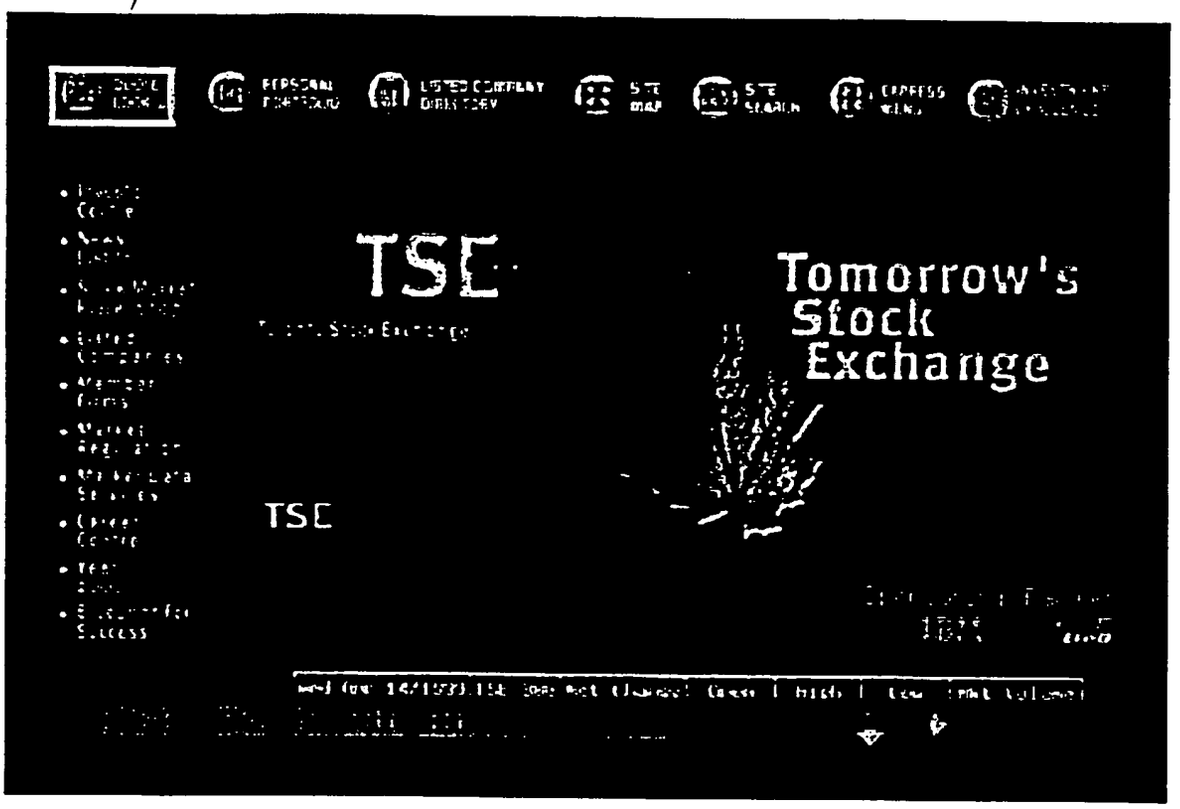


FIGURE 5A

510

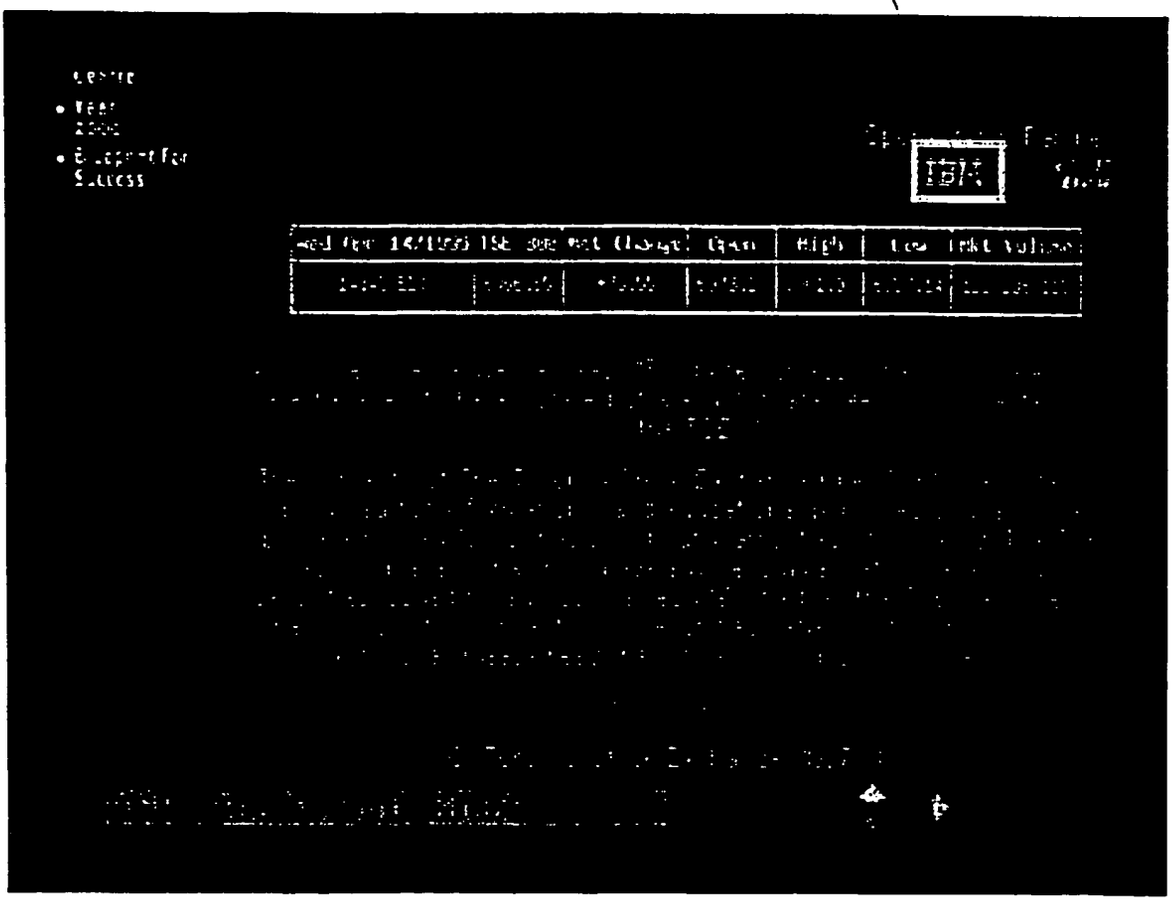


FIGURE 5B

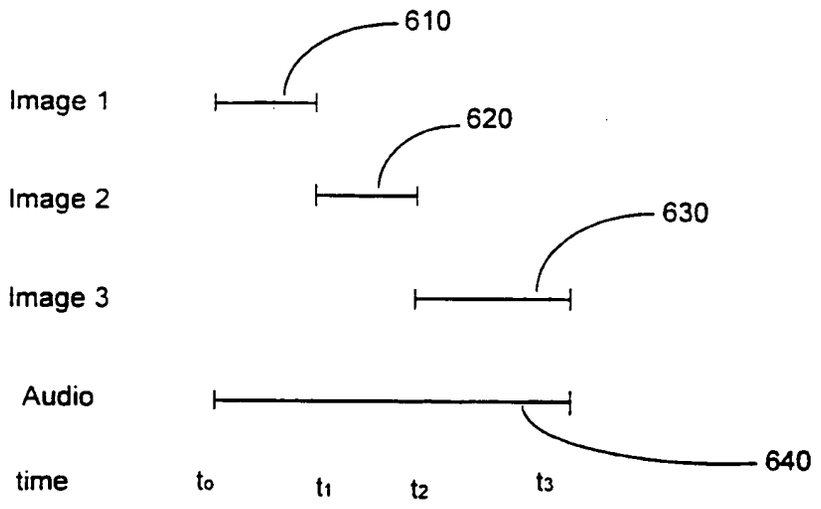


Fig 6A: Presentation timeline

650

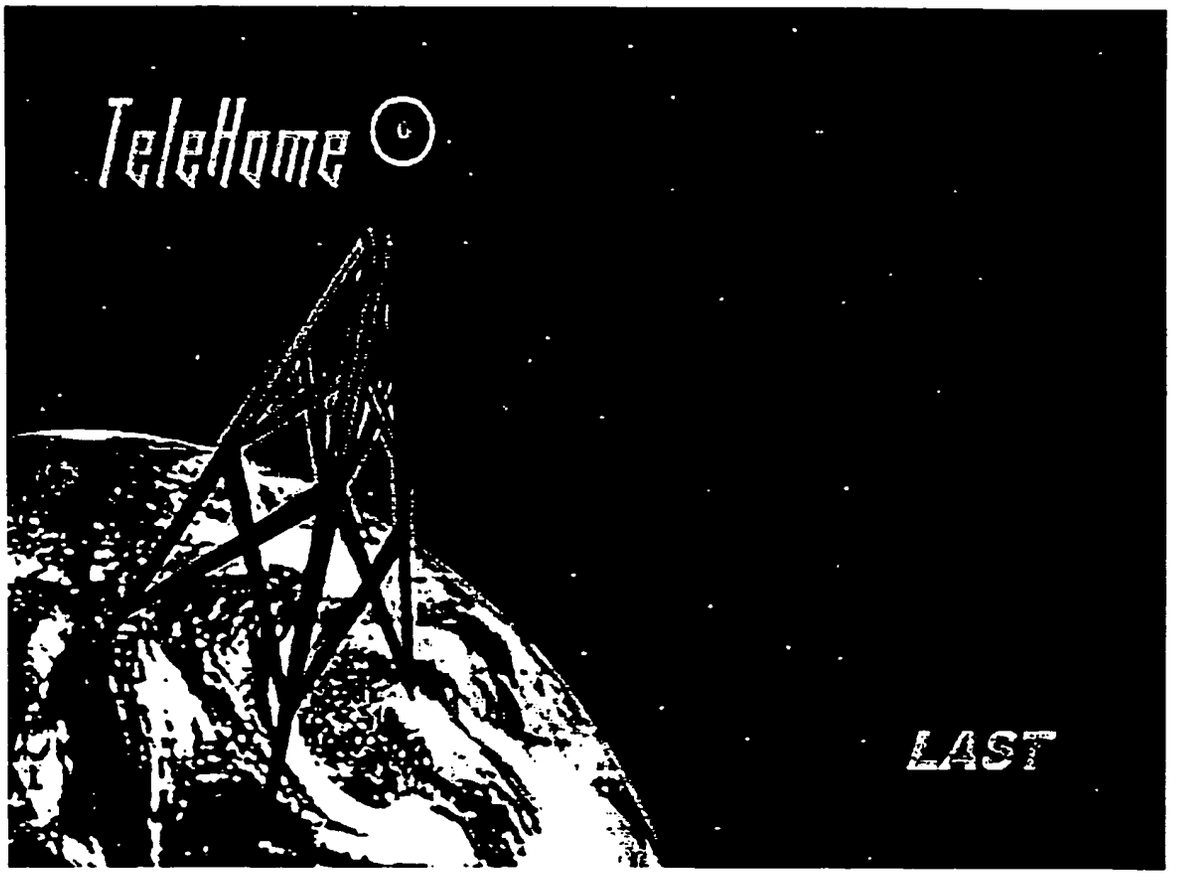


FIGURE 6B

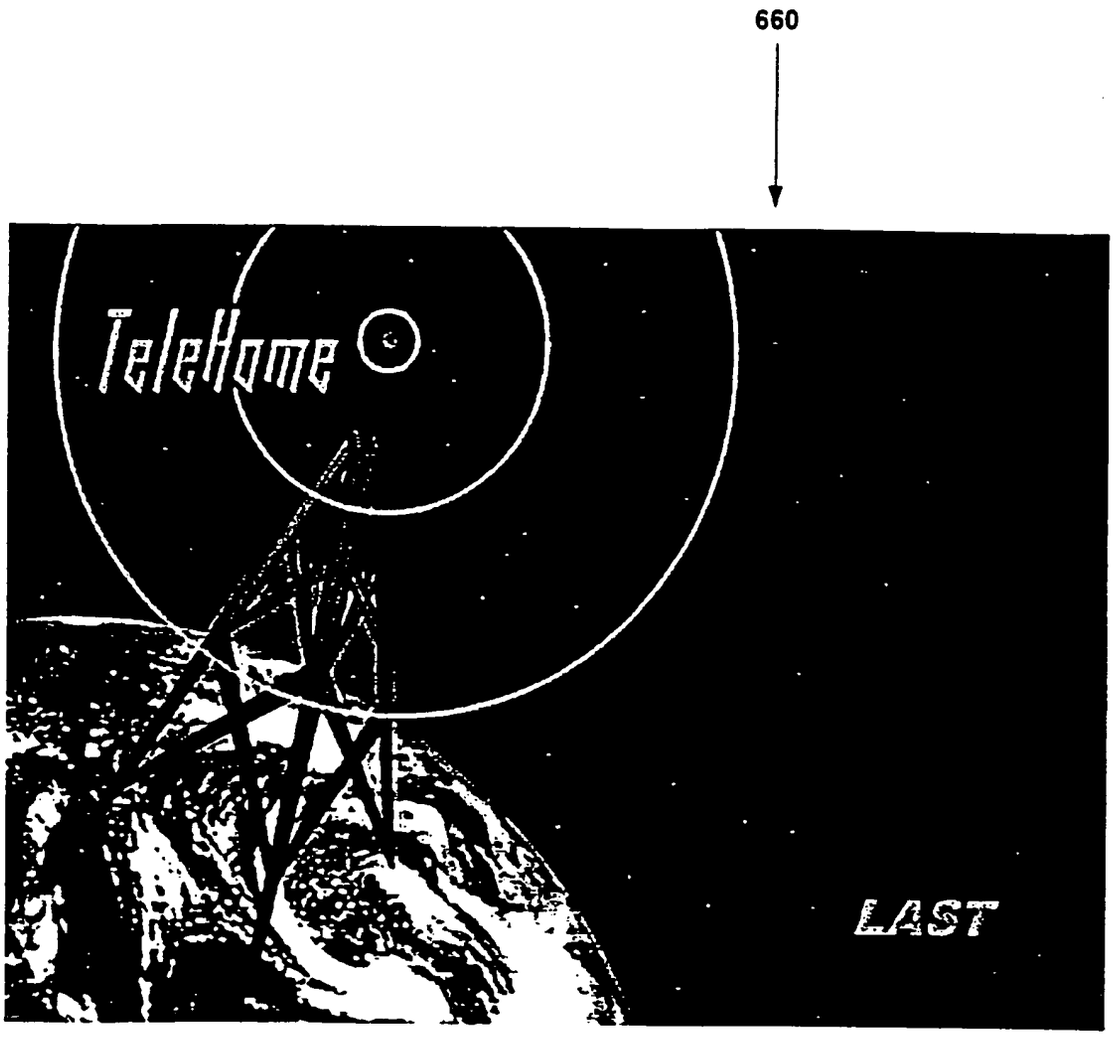


FIGURE 6C

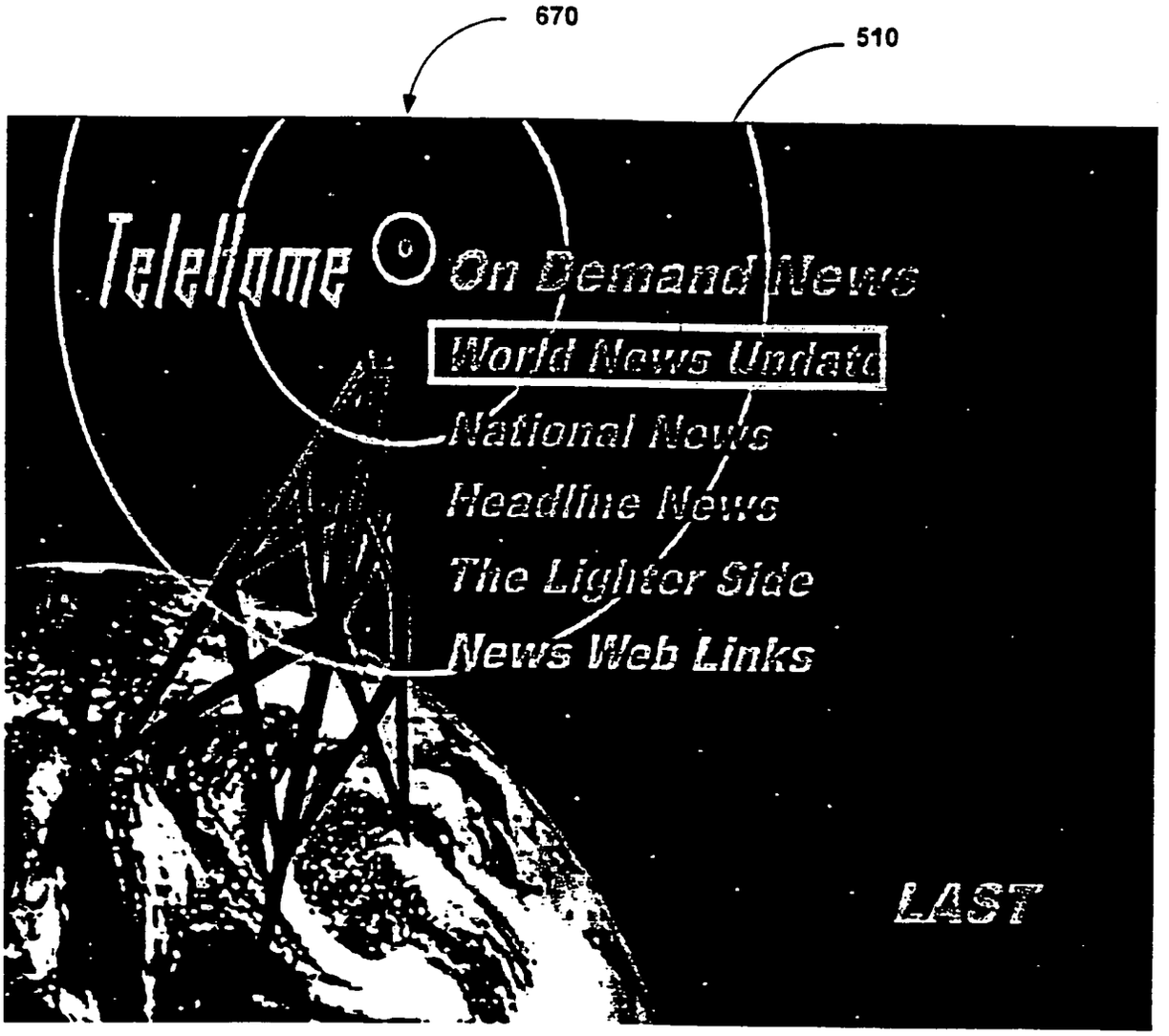

670
510
Telehome
On Demand News
World News Updata
National News
Headline News
The Lighter Side
News Web Links
LAST

FIGURE 6D

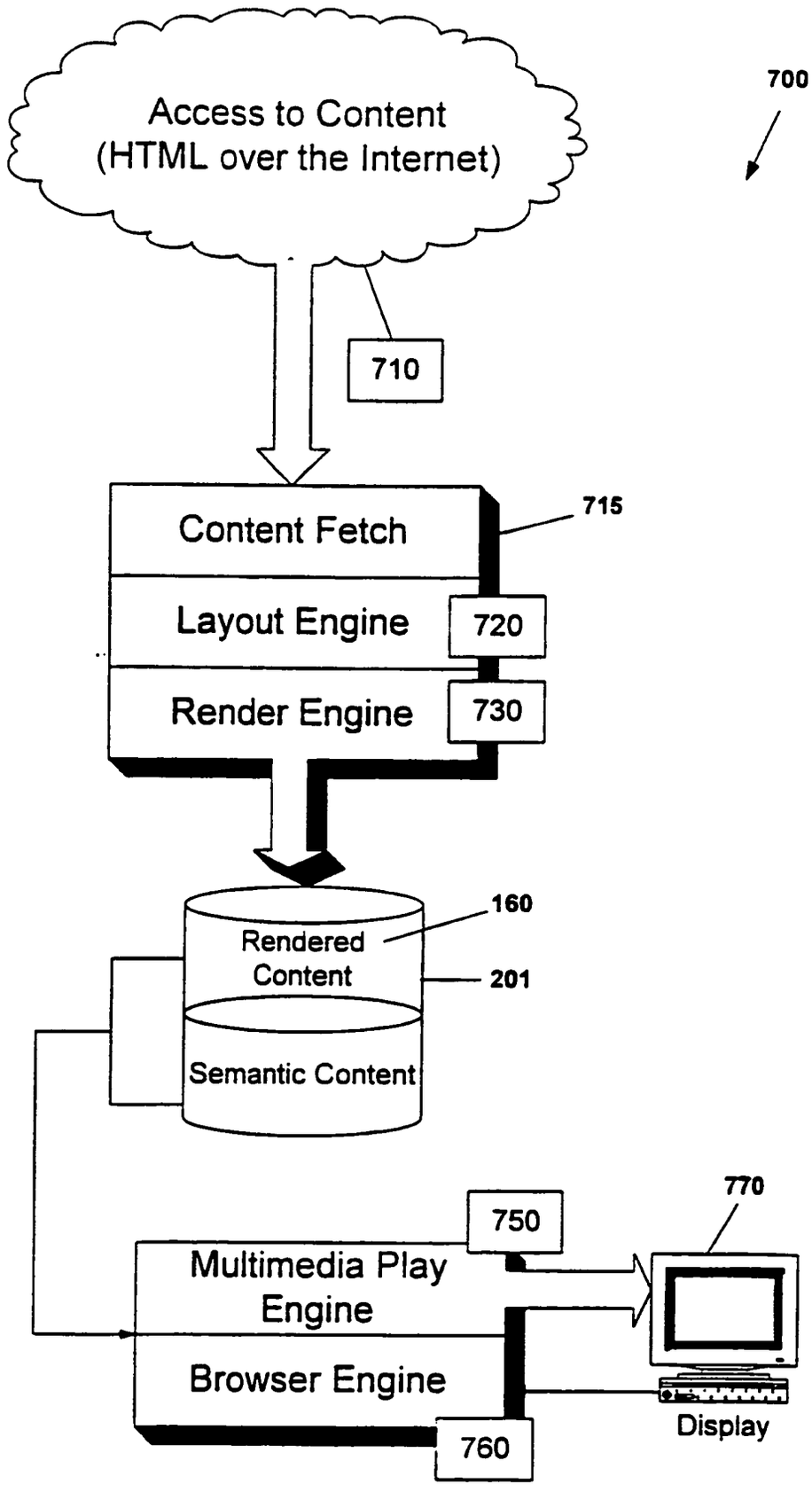

Access to Content
(HTML over the Internet)
700
710
Content Fetch
715
Layout Engine
720
Render Engine
730
160
Rendered
Content
201
Semantic Content
770
750
Multimedia Play
Engine
Browser Engine
760
Display

FIGURE 7

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 August 2002 (22.08.2002)

PCT

(10) International Publication Number
WO 02/065252 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US02/04684

(22) International Filing Date: 15 February 2002 (15.02.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/269,593 15 February 2001 (15.02.2001) US
10/077,284 14 February 2002 (14.02.2002) US

(71) Applicant: ACTV, INC. [US/US]; 10th Floor, 233 Park Avenue South, New York, NY 10003 (US).

(72) Inventor: HARRINGTON, Jeffrey, Michael; 442 6th Street, Brooklyn, NY 11215 (US).

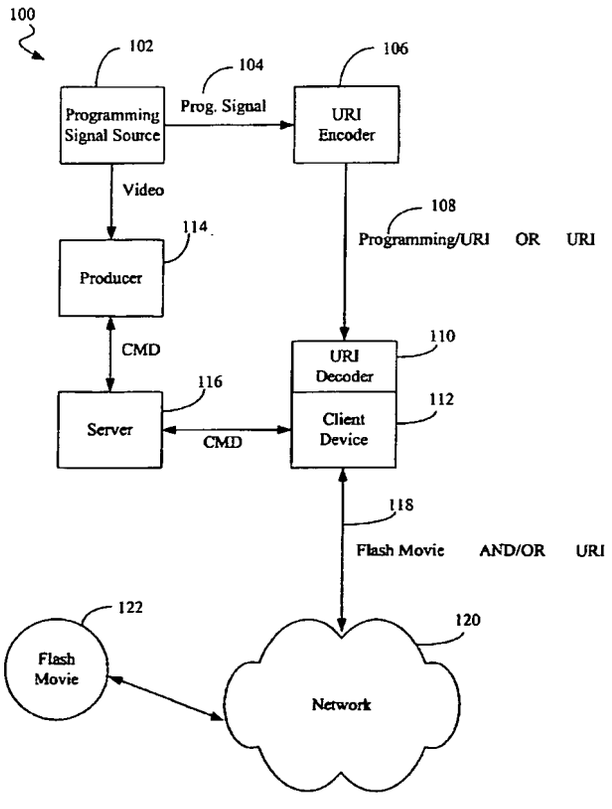
(74) Agents: OSBORNE, Thomas, J., Jr. et al.; DORSEY & WHITNEY LLP, Suite 4700, 370 17th Street, Denver, CO 80202 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR SERVER-SIDE CONTROL OF A FLASH PRESENTATION



(57) Abstract: A system and method for controlling a Flash presentation on a client device through the use of a command received from a server is disclosed. The Flash presentation may be further synchronized to a programming signal by receiving a Universal Resource Indicator ("URI") specifying the location of a Flash movie that relates to the programming signal; retrieving the Flash movie from the location; loading the Flash movie onto the client device including a Flash player; establishing a connection between the client device and a server; and receiving a command from the server to direct the Flash movie on the client device. A Flash presentation may also be used to present a real-time data feed on a client device under the control of a server. The server receives a real-time data feed and generates a command directed to a Flash movie on the client device. The command directs the presentation of the Flash movie.

WO 02/065252 A2

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

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.

SYSTEM AND METHOD FOR SERVER-SIDE CONTROL OF A FLASH PRESENTATION

of which the following is a specification.

REFERENCE TO RELATED APPLICATIONS

This application claims priority to United States Application Serial No. _____ entitled "SYSTEM AND METHOD FOR SERVER-SIDE CONTROL OF A FLASH PRESENTATION" and filed on February 14, 2002 by Jeffrey M. Harrington, which is incorporated by reference in its entirety. This application further claims priority to United States Provisional Application No. 60/269,593 entitled "SYSTEM AND METHOD FOR SERVER-SIDE CONTROL OF A FLASH PRESENTATION" and filed on February 15, 2001 by Jeffrey M. Harrington, which is incorporated herein by reference in its entirety. This application is further related to the following applications, which are incorporated herein by reference in their entirety: U.S. Patent application serial number 09/396,693 of Craig D. Ullman, Michael R. Abato, Jeffrey M. Harrington, and Carl R. Duda, entitled "ENHANCED VIDEO PROGRAMMING SYSTEM AND METHOD FOR PROVIDING A DISTRIBUTED COMMUNITY NETWORK," and filed on September 15, 1999 (hereafter, the "DCN application"); U.S. Patent application serial number 09/472,385 of Craig Ullman, Jack D. Hidary, and Nova T. Spivack entitled "ENHANCED VIDEO PROGRAMMING SYSTEM AND METHOD INCORPORATING AND DISPLAYING RETRIEVED INTEGRATED INTERNET INFORMATION SEGMENTS," and filed December 23, 1999; and U.S. provisional patent application of Michael R. Abato, entitled "A SYSTEM AND METHOD FOR PRESENTING CONTENT RELATED TO A TEMPORAL EVENT TO A USER VIA A VIRTUAL STAGE," and filed on February 15, 2001 (hereafter, the "STAGE" application").

FIELD OF THE INVENTION

The present invention relates generally to a Flash movie playing on a client device. In more particularity, the present invention relates to server-side control of a Flash movie

playing on a client device. In addition, the present invention relates to synchronizing programming with a Flash movie playing on a client device.

BACKGROUND OF THE INVENTION

Today, the capabilities of computers to provide massive amounts of educational and entertainment information have exploded with the Internet. The Internet has the power to transform society through unprecedented levels of information flow between members. Currently, on-line systems offer a variety of different services to users, including news feeds, electronic databases (either searchable by the user directly on the on-line system, or downloadable to the user's own computer), private message services, electronic newsletters, real time games for play by several users at the same time, and job placement services, to name a few. However, today, most on-line communications occur merely through text. This currently stands in great contrast to the audio/visual presentation of the alternative electronic medium, television. However, it is expected that as multi-media's incessant growth continues, audio/visual programs will proliferate and text will become less and less dominant in the on-line environment. Even though these programs will be introduced, the Internet will remain essentially user unfriendly due to its very massiveness, organization, and randomness. Simply stated, there is no order or direction in the Internet. Specific pieces of information are many times hard to find, and harder yet, is the ability to put that piece of information into a meaningful context.

Television, on the other hand, has been criticized for being a passive medium--"chewing gum for the eyes," as Fred Allen once observed. Television has always been something you watched, not something you do. Many social critics believe that the passivity television depends on has seeped into our entire culture, turning a nation of citizens into a nation of viewers. While interactive television systems have increased the level of user interaction, and, thus, provided greater learning and entertainment opportunities, vast information resources such as databases are inaccessible from such a medium.

Recent innovations in combining Internet content with television and other audio and/or video programming signals have been described in various patents and publications, for example, United States Patent Number 5,778,181, which issued on July 7, 1998 to Jack D. Hidary, et al., and also in United States Patent Number 5,774,664, which issued on June 30, 1998 to Jack D. Hidary, et al. (hereinafter, collectively the "Hidary patents"), and also U.S. Patent Number 6,018,768, which issued on Jan. 25, 2000 to Craig Ulman et al. The contents of each of these patents are herein incorporated by reference in their entirety. As is now well known in the art, these patents describe innovative systems and processes for combining the user-friendly visual experience of television programming signals, and other time based events or signals, with information resources located on the Internet which relate to the programming signal (hereinafter, the "Enhanced Content"). Since segments in a programming signal are generally presented in a sequence to a client based upon a reference to a known event (for example, the amount of time remaining in a football game is based upon the kick-off, or the amount of time remaining in a recorded movie is based upon when the playback of the movie is started and not when it was actually filmed), such programming signals shall herein be regarded as applying to any signal, show, or sequence of events, whether pre-recorded or live, which are defined or based upon a temporal relationship (hereinafter, the "Temporal Signal"). Such Temporal Signals may include live events (for example, a cut-away by a television broadcaster to a then breaking news event), pre-recorded events, and combinations of live and pre-recorded events.

Recently, various approaches have been implemented for providing client-side and server-side systems capable of providing Enhanced Content related to a Temporal Signal. As is well known in the art, such approaches generally require a client to download (commonly from an Internet based Web site) and then install a proprietary plug-in or software, which configures the client's system as a specific application. Another approach utilizes a client system's Web browser, and a downloaded program which configures the client system to retrieve Enhanced Content over a specific type of communications link, for a specific type of

client device based upon the reception of a Temporal Signal and an address identifying a provider of Enhanced Content related thereto. Regardless of the specific methodology, today's client systems commonly must download and install an application program to receive and present Enhanced Content program segments, which relate to a given Temporal Signal.

Further, since the Internet has innumerable sites, which a client may or may not find using a search engine, producers of Temporal Signals often identify a location providing Enhanced Content (for example, an Internet site) by presenting a Uniform Resource Identifier (URI), which includes Uniform Resource Locators ("URL"), or similar address in the video or audio signal presenting the Temporal Signal. Once the site is identified by the client and/or the client's system, the approach then commonly requires the client system to register the client with the provider of the Enhanced Content.

Following registration, the client then may actually need to select a program or segment for which the client desires to receive the Enhanced Content (since Enhanced Content for multiple programs may be accessible from a single Internet site). Once selected, the client side system then often downloads and installs a browser plug-in, Java applet, Java script application, Shockwave™ component, or similar program code, which configures the client device for connecting with a persistent socket to a server to receive the specific Enhanced Content. A persistent socket, for example, may be implemented via, but is not limited to, a TCP/IP socket, any sort of communication protocol that implements persistence, or an application layer that implements persistence. At this point, the client system is then ready to connect to the provider of the Enhanced Content, satisfy any pre-requisites (for example, providing a password, sign-on, or user profile information), and receive the Enhanced Content.

As such, the approaches commonly utilized today to receive Enhanced Content generally require a client to first identify the location of a provider of Enhanced Content, register the client with the provider, download a program which configures the client system,

install the program, connect to a site providing Enhanced Content related to a specific Temporal Signal, and then satisfy any pre-requisites prior to receiving the Enhanced Content (for example, providing user profile information). In short, these approaches require so much time and effort to configure the client side system and access the Enhanced Content that many clients are discouraged from utilizing such systems.

What is needed is a means to reduce and minimize the amount of time and effort required by a client to receive automatically, or upon request, Enhanced Content related to a Temporal Signal. What is needed is a wider, richer, quicker, and more efficient system and process for receiving and processing audio/visual and textual database elements into an organized unique interactive, educational, entertainment experience.

Macromedia Flash™ technology includes a powerful animation application, which may substantially replace the hypertext mark-up language ("HTML") as the application of choice for Web site developers. A programmer using Flash can create interactive Web sites with sophisticated animation and sound, requiring low bandwidth and small file sizes. The visual presentation of a Web site using Flash is referred to as a Flash movie, which provides a window for capturing and displaying information, similar to an HTML page. Flash movies, unlike HTML pages, stay loaded in a Web browser, or any device with a Flash plug-in. In a most basic implementation, the Flash movie includes a series of vector graphic images that are animated by changing their parameters in keyframes along a timeline, conceptually similar to the way in which animation in a cartoon is achieved.

For a more sophisticated site, Flash also includes functionality to create interactive movies, where the visitor to the Web site may use a keyboard or a mouse to jump to different parts of the movie, enter information on forms, and perform other interactive operations. Flash movies may run from start to finish, or a viewer of the Flash movie may direct the Flash movie to change state. Flash also provides for layering, which allows movies and functionality to be overlaid. For example, a Flash movie illustrating human anatomy might have a first layer with graphics of a skeleton and a second layer with graphics of the muscular

system overlaid on the graphic of the skeleton. The movie might include a third layer with text describing the various anatomical features. As one can imagine, layering can be used to enhance the functionality and visual appeal of a Web site.

What is needed is a system and method for synchronizing the visual experience of TV with the dynamic capabilities of Flash movies on a client device. What is further needed is a system and method for server-side control of a Flash movie playing on a client device.

SUMMARY OF THE INVENTION

Systems consistent with the present invention provide a system and method for relating Temporal Signals (which appear, for example, on a television broadcast, a VHS or Beta tape, CD-ROM, DVD, CD, memory stick, or other medium) with a Flash movie on a client device (accessible, for example, via the Internet). Such systems do not require lengthy downloads, specific client devices or operating systems, specific data formats or similar constraints in order to implement the features and functions identified herein. Preferably, such a system is implemented on a client device capable of hosting a Web browser. As such, the present invention is described preferably in the context of a client device using a Web browser for supporting its operations.

Further, the present invention creates a new, efficient, dynamic, diverse and powerful educational and entertainment medium. The system allows consumers to receive more information in a more efficient manner than either television or the Internet alone and over prior systems and processes utilized to present Enhanced Content related to a Temporal Signal. Instead of requiring client systems to execute lengthy, and sometimes problematic, downloads, which often require the user to perform an installation of new software on the client system prior to receiving an Enhanced Content segment, the present invention streamlines such processes by providing server-side control of a Flash movie playing on a client device. By utilizing server-side control, the present invention minimizes the amount of client-side software that needs to be downloaded. As such, the Enhanced Content segments are available for immediate use without requiring a lengthy download or installation phase.

In such an embodiment, the Flash movie is preferably played using any standard Web browser that has a Flash plug-in, which is estimated to be currently installed on over 96% of all Web compatible computers. As such, by using the new systems and processes of the present invention, consumers not only can see a news report on television, but they can also be pushed pertinent information which will be displayed on the client device. The act of viewing a program has now become a more engaging, enriching experience, because Enhanced Content can now be obtained almost instantaneously without any lengthy downloads and installations, initialization routines, or constraints upon compatible systems or sources.

The present invention can also create a more intimate relationship between the client and the program. For example, in an educational environment, a student (the client) might be solving problems or performing virtual experiments on an Internet site that a teacher is discussing in an educational television program. The client is an active participant in the process, rather than a passive observer. Unlike previous systems, the present invention enables the student and the teacher to visit the classroom via any device capable of playing a Flash movie and connecting to a server, including the ever more increasingly popular wireless devices such as Personal Data Assistants ("PDA") and wireless communications devices. Such capabilities are possible with the present invention because the invention provides for pushing commands from the server to the client device (regardless of the device's specific configuration and/or capabilities above a minimum threshold as defined in terms of providing a Web browser or a comparable presentation mechanism and some accessible memory) to control the Flash movie relating to the temporal event. The server can contain sophisticated program logic, which would otherwise need to be downloaded and installed on the client device, to allow for more complex presentation options. In an educational setting where a student answers a question wrong, for example, the server may re-push material that was not understood or may even push a more detailed explanation for easier understanding by the student. Thus, this allows for more sophisticated user

experiences without the necessity of performing lengthy, and sometimes problematic, downloads and installations of the application software.

Another advantage of the system is that it changes the nature of advertising by making its application delivery more adaptable to the viewers. By keeping the complex programming logic on the server-side of the system, advertising can be more easily created and delivered in a targeted and individualized manner while allowing the client side content delivery to be instantly available. Branching logics and interactive sales presentations can be delivered without lengthy downloads. Since additional information can be now given to consumers automatically and without large downloads, advertising can now be more interactive, responsive, and substantive. Such real-time responsiveness allows customers to make more informed choices and spontaneous choices. Now, the act of purchasing a product seen on television can be streamlined -- the consumer can be given the choice of buying the product instantly using the two-way capabilities of the system. For example, the processes of the present invention enable an Enhanced Content provider to push a command to the Flash movie playing on the client device, and the command can seamlessly display a button for purchasing the product at a newly displayed price. The button includes the functionality necessary to purchase the product from a remote location.

In addition, users can take advantage of the two-way capabilities of the Internet to respond to polls, to send e-mail, or to link to additional sites. For example, a viewer watching a television financial news program, through the system of the present invention, can receive a real-time data feed that will be displayed, perhaps as a layer on the Flash movie.

The present invention includes a method for synchronizing programming with a Flash movie on a client device, which includes receiving programming on the client device. The programming contains a Uniform Resource Indicator ("URI"), such as a Uniform Resource Locator ("URL"), functional push, an object push, and a software enhancement push (i.e., the software on the client side can be dynamically enhanced without user intervention). The programming may be video programming, audio programming, or other Temporal Signals, as

defined herein. According to one aspect of the present invention, the URI specifies a location from where a Flash movie can be obtained, wherein the Flash movie relates to the content of the programming. The Flash movie is then retrieved from the location and loaded on the client device.

In one aspect of the present invention, the client device includes a Flash player, a Web browser having Flash movie playing capabilities, such as a Flash plug-in, or another application having Flash movie playing capabilities, such as an e-mail client. A connection is established between the client device and a server, such as a Web server, a DCN server, or a database server. To facilitate the connection, the client device preferably includes a receiver software layer, such as an ActiveX control, and a bridge layer, preferably running in a browser window with scripting capabilities such as JavaScript or VBScript. The receiver software layer, for example, may be located within a frame containing an active component capable of establishing a persistent socket or may utilize Flash 5 XMLSocket capabilities. The bridge layer can be used to communicate between the active socket component, i.e., the receiver layer, and the Flash movie, such as via a "LiveConnect" interface. The client device receives a command from the server via the receiver layer and passes it through the bridge layer to the Flash movie. Any client-side processing or logical operations are performed, and the push command, the data, or the software command is sent to the client device directing the presentation of the Flash movie.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram of a first system according to the present invention for synchronizing a Flash movie on a client device with a programming signal, and for server-side control of the Flash movie;

Figure 2 is a block diagram illustrating an exemplary client device according to the present invention;

Figure 3 is a flow diagram illustrating a method according to the present invention for synchronizing a Flash movie on a client device with a programming signal, and for server-side control of the Flash movie;

Figure 4 is a diagram of a second system according to the present invention for server-side control of a Flash movie, the second system including a producer sending commands to a Flash movie on a client device, the commands relating to a Temporal Signal;

Figure 5 is a flow diagram illustrating a method according to the present invention for server-side control of a Flash movie;

Figure 5a is a flow diagram illustrating a method according to the present invention for loading a Flash movie on a client device;

Figure 5b is a flow diagram illustrating a method according to the present invention for transmitting a command from a producer to a server;

Figure 6 is a diagram of a third system according to the present invention for server-side control of a Flash movie relating to a programming signal;

Figure 7 is a diagram of a fourth system according to the present invention including a first client device with a communication link to a second client device;

Figure 8 is a diagram of a fifth system according to the present invention for broadcasting a real-time data command correlating to a real-time data feed to at least one client device having a Flash movie;

Figure 9 is a diagram of a sixth exemplary system according to the present invention for server-side control of a Flash movie playing on a client device, wherein the client pushes commands to the server, which are then broadcast to a plurality of client devices; and

Figure 10 is a diagram of a seventh exemplary system according to the present invention for server-side control of a Flash movie playing on a client device, wherein the client pushes commands to the server, which are then sent to one other client device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A first exemplary system consistent with the present invention preferably combines the rich visual capabilities of video with the dynamic capabilities of Macromedia Flash™ technology to provide a synchronized experience, and to provide a means for server-side control of the Flash movie. The first system preferably refers to video programming; however, the systems and methods described herein are equally applicable to any programming signal including, for example, audio, streaming video, streaming audio, holographic images, virtual reality signals, and any other type of Temporal Signals.

Referring to Figure 1, an embodiment of the first system 100 of the present invention for synchronizing a Flash movie 122 on a client device with a programming signal 104 via server-side control of the Flash movie 122 is illustrated. This embodiment allows a client device 112 to receive a programming signal 104, such as a video programming signal, with an embedded Uniform Resource Identifier URI, collectively the combined signal 108. An URI, for example, may identify to the client device 112 an address location on a network 120 where a Flash movie 122 is located. The client device 112, whether automatically (for example, a push) or upon client direction (for example, a pull), retrieves Flash movies 122 from the address location. The client device 112 may be a personal computer, a set-top box, a digital TV, a Web tablet, a PDA, a wireless device, or any other device with a connection to a network and the ability to run a Web browser with a Flash player. A Flash player on a personal computer or other client device, for example, may be used as a screen saver for taking over a display screen if there are no user inputs such as keystrokes or mouse movements for a specified duration. Thus, the Flash movie may provide animations, graphics, text, and the like on the display. Alternatively, the client device 112 may be any device capable of running a stand-alone Flash player and communicating with a network 120.

In the embodiment illustrated in Figure 1, the client device 112 is also preferably connected to either a cable and/or broadcast television connection or to a local VCR or other video source, and receives a programming signal by that connection. The programming

signal 104 can then be processed for presentation, such as for display on a screen of the client device 112 using any conventional PC card capable of displaying NTSC signals on a screen, such as a WinTV card, and/or played over a speaker of the client device 112 using any conventional PC audio card. Alternatively, the client device 112 may be run in parallel with a second client device for viewing video programming, such as a television or for listening to audio programming, such as a radio.

The programming signal is preferably distributed to viewers in their homes from a centralized location, e.g., the programming signal source 102, and is created according to any conventional means known in the art. After the programming signal is created, an URI or a plurality of URI(s) are embedded into the programming signal 104 via the URI encoder 106. In one embodiment, for example, the URI can be embedded into the Vertical Blank Interval ("VBI") of the video programming by the URI encoder 106. In this embodiment, the URI is preferably encoded into eight fields of line 21 of the VBI. Line 21 is the line associated with close captioning, among other things. However, the URI could also be embedded in other fields of the VBI, in the horizontal portion of the video, as part of the audio channel, in any subcarrier to the video, or if digital, in one or more of the data fields. In a video programming embodiment, the particular information in line 21 is not part of the visual part of the program, is not perceptible to the human eye, and, thus, is ideal to send data information to the users. While the bandwidth capacity of line 21 is limited, because the system transmits only the URI(s), and not full Flash movies, there is more than enough capacity. Furthermore, no additional hardware is necessary at the client device to implement the elements of the present invention. Thus, the present invention has the additional advantages of being very efficient and takes advantage of conventional hardware.

Although Figure 1 shows the programming signal with an embedded URI transmitted over the same line, the URI(s) alternatively can be transmitted independently of the programming signal on the same data channel or over a different data channel, or via the network itself. In this embodiment, the URI(s) can be forwarded to the remote sites either

prior to initiation or during the transmission of the programming signal 104. In one embodiment, the URI(s) have associated time stamps, which indicate to the subscriber platforms (e.g., the client device 112) when, during the programming signal 104, to fetch and play Flash movie(s), which can be obtained via the network 120 or other device identified by an address specified by the particular URI(s). As shown in Figure 1, a Flash movie 122 is illustrated as being associated with the address provided in the URI. The Flash movie 122 may reside on a server or any other device that may be identified by an address or similar designator and from which the Flash movie may be obtained. For example, when the Flash movie 122 associated with a given URI is provided on a CD or DVD, the URI may refer to a location on such computer readable medium at which the data of interest (i.e., the Flash movie) is stored. In such an embodiment, the DVD player effectively operates as a server by providing the Flash movie 122 to the client device 112. In the preferred embodiment for this system 100, however the URI suitably identifies a server accessible via the network 120, for example, the Internet. Alternatively, as discussed in more detail below, the user can select when to call the particular Flash movie(s) for display with the programming signal 104.

Once the programming signal 104 is created, it can be transmitted to user sites over any transmission means, including broadcast, cable, satellite, or Internet, and may reside on servers, such as video or audio servers. Furthermore, the programming signal 104, one or more URI(s), and/or the combined signal 108 can be encoded on a storage medium, such as a VHS tape, a Beta tape, an audio tape, a digital audio tape ("DAT"), DVD, CD, CD-ROM, CD-R, CD-RW, or other storage medium. Also, digital recording mechanisms and devices, such as a TiVO® unit, may be utilized to record and/or play back any recordings of the programming signal 104, the URI(s), and/or the combined signal 108.

The programming signals 104 and/or the combined signal 108 may also be communicated as a live or prerecorded signal to the client device 112. Such communications may be pre-set (for example, based upon a network broadcast schedule), may be real-time (for example, when a breaking news event occurs), and/or may be on-demand. For example,

the programming signal 104, the URI(s), and/or the combined signal 108 may reside on audio and/or video servers until requested by a client (for example, as video on demand).

In the embodiment illustrated in Figure 1, for example, a local URI decoder 110 receives the combined signal 108 including the programming signal 104 with the encoded URI(s). The local URI decoder 110 extracts the URI(s), preferably embedded in the VBI, with the use of any conventional decoder device. The URI decoder 110 may include a stand-alone unit, include hardware associated with the client device 112, such as a card that is connected to the client device, and/or a software application running on the client device 112. Alternatively, the URI decoder may be located at a server connected with the network. When the URI decoder receives the combined signal 108, it strips out the URI(s), such as from line 21 of the VBI, and delivers the URI(s) independently to a server. The URI is then subsequently delivered via the network 120 to the client device 112. Simultaneously, the programming signal 104 is broadcast over conventional broadcast or cable transmission means to the client device 112.

When the client device 112 receives the URI, the Flash movie 122 residing at the URI is loaded into the memory of the client device 112. Preferably, the client device 112 automatically establishes a communications link with a server located at the URI, accesses the URI, and loads the Flash movie 122 (i.e., a configuration often referred to as a 'push"). Alternatively, the client device 112 may be configured to establish the communications link with the server upon the receipt of an URI and a command from a client directing the client device 112 to retrieve the Flash movie 122 (i.e., a configuration often referred to as a "pull"). As such, the client device 112 may be configured to be "pushed" or to "pull" Flash movies identified by an URI provided in conjunction with a programming signal 104.

The URI decoder 110 of Figure 1 preferably routes the URI(s) to a device or a portion of the client device 112 configured to receive URI(s), such as a Web browser on a personal computer, a set-top box, a digital TV, a wireless device, a gaming console, a wireless telephone, a PDA, or any other device capable of presenting a Flash movie. Since the URI(s)

identify Flash movies 122, which require Flash capabilities, preferably the Web browser 136 (shown in Figure 2) includes a Flash™ player. However, the client device 112 may also be configured with a stand-alone Flash™ player (i.e., a Flash™ player that operates independent of or in conjunction with a Web browser in order to present a Flash movie). Currently, any Flash™ equipped Web browser (for example, a Microsoft® Internet Explorer® or Netscape® Navigator™ browser) is capable of presenting a Flash movie without modification to the Flash movie or to the Web browser. As shown in Figure 2, for the first embodiment of the present invention, a Flash™ compatible Web browser 136 is used to present the Flash movie 122.

However, it is anticipated that as Flash Movies become more ubiquitous, devices will be provided for presenting Flash Movies without requiring or utilizing the full capabilities of a Web browser. As such, the client device 112 preferably may be configured to provide a platform for receiving URI(s) and presenting Flash movies 122 in conjunction with or separate from the reception and presentation of a programming signal 104. Such a client device 112 may not require or utilize the full capabilities of a Web browser operating on a personal computer or similar device. Thus, it is to be appreciated that for the system 100 shown in Figure 1, the client device 112 provides the capabilities of identifying, locating, retrieving and presenting Flash movies in conjunction with a programming signal by utilizing URI(s) or other schemes for identifying local and/or remotely located Flash movies.

For the embodiment shown in Figure 1, the client device 112 suitably communicates with a server 126 and provides any URI(s) received from the URI decoder 110 to the network 120 over a suitable communications link 118. In Figure 1, a single bi-directional communications link is commonly adequate for facilitating communications between the client device 112 and the network 120. However, in certain embodiments, wherein high speed communications are utilized, parallel and/or numerous communications links may be utilized. Further, the communications links shown in Figure 1, and throughout the Figures, are provided for illustrative purposes only and are not to be construed as depicting an actual

or preferred hardwire configuration. As is commonly appreciated, Web browsers commonly connect with a server, associated with a particular URI, via the Internet, a Local Area Network ("LAN"), a wired network, a wireless network, a combination wired and wireless network and/or a proprietary system providing a non-standard extension such as a Distributed Community Network ("DCN"). For a description of a DCN, see United States Patent application serial number 09/396693, which was filed on September 15, 1999 in the name of inventors Craig Ullman et al., and is entitled "Enhanced Video Programming System and Method for Providing a Distributed Community Network", the contents of which are herein incorporated by reference in their entirety.

In the embodiment illustrated in Figure 1, a producer 114 is connected with a server 116, which is connected to the client device 112. In this embodiment, the server 116 is shown separate from the network 120. The server 116, however, may also be a server residing on the network 120. Connected with the client device 112, via the server 116, the producer 114 may direct the Flash movie 122 to relate with the programming signal 104, and additionally, or alternatively, direct the Flash movie 122 to relate to other Temporal Signals. For example, the producer, using a command line interface, may issue a command to the Flash movie 122 residing on the client device 112. The command will be sent to the server 116, which in turn will send the command to the client device 112.

An exemplary client device is illustrated in Figure 2. The client device 112 includes a Flash master movie 130, a receiver 132, and a bridge layer 134, running in a browser window 136 with scripting capabilities such as JavaScript or VBScript. The receiver 132 is connected to a server. Preferably, the receiver 132 is implemented as an ActiveX control or a Java Applet, and facilitates communication between the Flash movie, e.g., the Flash movie 122 shown in Figure 1, and the server. The server may be a Web server, a DCN server, or any other type of server. The receiver communicates with the bridge layer 134, which in turn communicates with the Flash movie 130. The bridge layer 143 may, for example, run in the browser window with scripting capabilities such as, but not limited to, JavaScript or

VBScript (hereinafter, for simplicity, the terms "bridge layer" and "JavaScript layer" are used interchangeably although one skilled in the art would readily appreciate that the bridge layer may be implemented using JavaScript, VBScript, or other known scripting capabilities).

A first method consistent with the present invention relates a Flash presentation with a programming signal. Referring to Figure 3, a preferred method for relating a Flash presentation with a programming signal, such as a video programming signal, is illustrated. In operation 300, a user of the client device preferably launches a Web browser having a Flash player. The client device, for example, may be a personal computer, a set-top box, a wireless device, or any other device with a connection to a network and the ability to run a Web browser. Alternatively, the Web browser may include a Flash plug-in, which facilitates the execution of a Flash movie on the client device. Alternatively, the client device may be capable of running a stand-alone Flash Player, such as a Sony PS2 Game Console™, and communicating with the network. A preferred embodiment of the client device is illustrated in Figure 2.

In operation 310, the client device receives a programming signal having an embedded URI, which preferably directs the Web browser to a network location, such as a Web site, with a Flash movie relating to the programming signal. Preferably, as discussed hereinbefore, the URI may be embedded in the first 21 lines of the VBI. Alternatively, the URI may be sent independently of the programming signal, such as via a server. The delivery of the URI for the movie that relates to the programming signal is a means for synchronizing a Temporal Signal, such as a programming signal, with a Flash movie.

In operation 320, the client device is connected with the Web page corresponding to the URI, the Web page having the master movie. In operation 330, the Flash movie is loaded on the client device. In one embodiment, the URI is accessed and the Flash movie is downloaded automatically by the client device when the URI is received. Alternatively, the user may manually enter the URI into the browser, and connect with the Web page, and download the master movie. Preferably, the downloaded Flash movie is a master movie.

As used herein, the master movie is preferably a Flash movie having the core functionality relating to the programming signal. For example, a master movie designated to relate with a live television broadcast of a football game, may include touchdown graphics, and text corresponding to key players on the teams. Non-core functionality, which may be pushed to the master movie by the producer according to the present invention, might include a sudden death overtime graphic. In an alternative example, such as with the system 400 of Figure 4, the Flash movie 412 is generally a stand-alone presentation, which may be controlled by a producer 404 or controlled by a server-side playlist, script, application, or other functionality on the server according to the present invention. As is well known in the art, Macromedia ActionScript™ is an object-oriented scripting language that allows a user to define a set of instructions that run when a triggering event occurs. There are various events that trigger a script including the viewer clicking on a button or the movie reaching a certain point along its timeline. For example, a button may be displayed on the Flash movie that is associated with a script that jumps to an URI, such as a URL, and fetches a document, conceptually similar to an HTML hyperlink, and when the viewer clicks on the button the script is executed and the document is fetched. In another example, a movie may be stopped when it reaches a certain point along its timeline such as when a Web site's introductory graphics have concluded.

In operation 340a (Figure 3), a programming event is started, and in operation 340b, the master movie is started, preferably contemporaneously with the programming event, and the events are synchronized, and or related to one another through server-side control of the Flash movie on the client device according to one embodiment of the present invention.

According to the present invention, the Flash movie running on the client device may be synchronized with the programming signal through server-side control. In addition, according to the present invention the Flash movie on the client device may be directed by the producer, or directed by server-side controls. The client device, such as the client device illustrated in Figure 2, is connected with the server.

One example consistent with the present invention for server-side control of the Flash movie on the client device includes having a playlist resident on the server, the playlist being for the Flash movie being presented on the client device. The playlist includes a timeline, and at least one command related to the timeline. The playlist may be played from the server, which will issue the command at the appropriate time to the Flash movie playing on the client device. To synchronize the Flash movie on the client device with the programming signal, the playlist timeline and associated commands are related to the programming signal. For example, consider a live video broadcast of a football game. The playlist can include prescripted commands that direct a pre-show Flash movie, which may be downloaded with the master movie or fetched by the master movie, to execute along with the preshow portion of the programming signal. The playlist, for example, may include a set of time/push pairs, such as the example shown in Table 1. In this example, the playlist includes two URI pushes and a functional push. Thus, at times 1:00 and 2:05, URI pushes are provided to the application, and the application may retrieve Enhanced Content from the locations indicated by the URI(s) at the respective times. Then, at time 3:10, a functional push directs the application to execute a ShowData function with the parameters 32/23/13. In a live video broadcast of a football game, for example, the first and second URI pushes may direct the application to retrieve and play prerecorded pregame shows for each of the teams involved. The functional push may then direct the application to display the parameters 32/23/13, which may be live or prerecorded data related to the football game, e.g., scores, player statistics, and team statistics, or unrelated to the football game, e.g., stock tickers, advertisements, and breaking news updates. One benefit of having the playlist on the server is that commands can be issued or broadcast to a plurality of client devices at the same time, and can be sent in temporal relationship with the programming signal. In this way, the Flash medium becomes a synchronized mass media mechanism akin to traditional mass media such as television or radio.

Table 1

Time	Push
1:00	http://www.hypertv.com/push1.swf
2:05	/push2.swf
3:10	{command: ShowData (33/23/13)}

A second example consistent with the present invention for server-side control of the Flash movie on the client device includes the producer issuing a command, or commands, to the Flash movie playing on the client device. This functionality allows the producer to direct the Flash movie as it relates to the programming signal or to other Temporal Signals. In one embodiment, such as shown in Figure 1, using a command line interface ("CLI"), the producer 114 can send a command to the server 116, which in turn will issue the command to the client device 112. The Flash movie on the client device, preferably via a persistent socket connection maintained by the receiver 132 (shown in Figure 2) and the programming resources maintained by the JavaScript layer 134, will execute the command. For example, if the football game goes into sudden death overtime, the producer can send a command to the Flash movie 130 on the client device 112 to play a sudden death graphic, which was not part of the playlist on the server for the football game master movie, perhaps because it is unexpected or uncertain. In another example, the producer 114 (shown in Figure 1) can create a new Flash movie and save it at a location identified by an URI, and send a command to the Flash movie 130 on the client device to connect to the URI, load the new Flash movie and present it.

Flash also includes functionality referred to as ActionScript "methods," which can be called from the browser, to control a movie in the Flash Player from Web browser scripting languages such as JavaScript and VBScript. Such a method is generally a predefined Flash function that can be called from a host environment, which is any device capable of running a Flash movie (such as a Web browser or a stand-alone Flash player), to the Flash movie. An exemplary method is "GotoFrame," which starts playing the Flash movie at the specified frame. A second exemplary method is "LoadMovie," which loads an external movie from a

specified URI. In one example, the producer can send new software methods to the Flash movie on the client device by commanding the download of a new Flash movie. The new Flash movie, for example, may include new software functionality with or without any additional visual or audio components. Alternatively, the playlist could be sent and loaded on the client device from the server. With the playlist resident on the client-side, the system could prefetch Flash content to use for later playback with or without a network connection.

A third example consistent with the present invention for server-side control of the Flash movie on the client device 112 includes pushing a command to jump to a movie clip index, i.e., functionality is pushed to the client device 112. The object, or other functionality, at the frame location or index, executes when the location is pushed. For example, the producer 114 may want certain text located at a specified URI to be displayed on a layer of the Flash movie. The producer will push a command to jump to the index through the JavaScript layer 134 (shown in Figure 2), which will direct the Flash movie to the index having a getURL command, which will access the URL, get the appropriate text, and display it in a window, such as the text layer. In this example, the getURL command was prescripted and the producer 114 commanded the movie to jump to the index location for the command, wherein the functionality for accessing the URL was located.

In a fourth example consistent with the present invention for server-side control of the Flash movie on the client device, the producer 114 may push new functionality, such as a new JavaScript function expressed as a string, to the JavaScript layer 134 on the client device 112. This is especially useful for unforeseeable or uncertain events, and for non-core functionality. For example, there may be a breaking news event, and the producer 114 generates a Flash movie including text discussing the event and places it at a specified URI. The movie may include additional functionality, such as to change text as events change. The JavaScript would be able to access the new functionality in the new movie, and accordingly set the appropriate variables in the movie. The producer 114 may send the Movieclip.LoadMovie method to the client device 112, whereby the JavaScript layer 134 will

communicate with the Flash movie using, for example, MovieClip.SetVariable or MovieClip.LoadMovie methods. The newly-loaded Flash movie could incorporate and execute the new functions as a new ActionScript function.

Figs. 1, 4, 6, and 7 each illustrate a single client device, however, the present invention is equally applicable to any number of client devices. Accordingly, in one example of the present invention a single playlist running on a server, or a plurality of servers, can broadcast playlist commands to any number of client devices connected with the server, and thereby synchronously direct the Flash movies playing on the respective client devices. Moreover, the producer can issue commands to any number of subscribers, and the subscribers preferably all playing the same Flash movie on their respective client devices will contemporaneously receive the commands issued by the producer, and the Flash movie playing on their client machine will contemporaneously react to the commands. Alternatively, the subscribers may be playing different Flash movies, and are all pushed commands relating to a Temporal Signal, such as a breaking news event, which will be displayed on the same portion of their respective client devices. Flash playback systems synchronized in such a manner could become a mass medium akin to traditional mass media such as television and radio.

A second exemplary system 400 consistent with the present invention provides for server-side control of a Flash movie 412 running on a client device. Referring to Figure 4, an embodiment of the second system 400 of the present invention is illustrated. In this embodiment, a client device 416 is connected to a network 410, such as a wireless network, an intranet, an extranet, or the Internet. Preferably, the Flash movie 412 resides at a site on the network accessible via an URI entered into a browser running on the client device 416, and the Flash movie 412 is loaded on the client device 416. Alternatively, the Flash movie 412 may be loaded from a CD-Rom, a floppy disk, or from any memory element connected to the client device. Preferably, the Flash movie 412 loaded on the client device 416 is a

master movie having core functionality as discussed herein. The present invention, however, works equally well with Flash movies 412 having any degree of functionality.

A producer 404 is also connected with the network. To cause the Flash movie 412 to be reactive to Temporal Signals 402, the producer may push playlist commands to the Flash movie 412, and may push new functionality, such as a new Flash movie, to the client device 416. Preferably, the Flash movie 412 includes an ActionScript script. Accordingly, the producer 404 may push methods to a JavaScript layer 134 (shown in Figure 2) running on the browser 136, which will expand the functionality of the JavaScript layer 134. For example, a client may be playing a Flash movie corresponding to a music video, when an unrelated Temporal Signal, such as a team winning a sporting event occurs. The producer 404 can create a Flash movie 406 relaying the Temporal Signal, such as a Flash movie including a layer with a graphic displaying the winning team and a layer with the score of the game. The producer 404 can push a command, using a server, to the Flash movie 412 on the client device 416, instructing the Flash movie 412 to fetch and play the new Flash movie 406 for the winning team on the client device 416. The enhanced JavaScript layer would allow for completely new presentation logic to be added to the Flash movie dynamically. In the beginning of the presentation, there might not be a presentation resource to show team scores. After the enhancement through the pushing of additional code (e.g., both JavaScript and Flash), however, the Flash movie would have this new capability.

A method consistent with the present invention provides for server-side control of a Flash movie playing on a client device. Additionally, a method consistent with the present invention provides for server-side control of a Flash movie playing on a client device responsive to a Temporal Signal, or responsive to commands from a producer. Referring to Figure 5, a method for server-side control of a Flash movie is illustrated.

In operation 500, a Flash movie is loaded on a client device. Preferably, the client device is any device with a network connection or a connection to a provider of a Flash movie, and the ability to play a Flash movie, such as a personal computer, a set-top box, a

wireless device, a Web tablet, a PDA, and the like. Preferably, the client device includes a Web browser with a Flash player.

Referring to Figure 5a, the preferred operations for loading a Flash movie on a client device are illustrated. In operation 502, the user launches a Web browser having a Flash plug-in, such as Microsoft Internet Explorer™, on the client device. In operation 504, the Web browser is connected with a network location, such as a Web site, having a Flash movie. This is preferably done by the user. Alternatively, consistent with Figure 1, an URI for the Web site having the Flash movie may be delivered along with a Temporal Signal. Preferably, the URI may be in the first 21 lines of the vertical blanking interval of a video programming signal. Alternatively, the URI may be received directly from a Web server, or the user may enter the URI into the browser manually. For example, the user may be watching an event on TV, which indicates that an interactive Flash presentation for the TV event is available at a certain Web page, and the user connects their client device to the appropriate URI for the Web page.

In operation 506, the Flash movie is loaded on the client device. The Flash movie may include core functionality, or may simply provide a vehicle for communication with the server, as discussed below, in which case the functionality will be pushed to the Flash movie from the server responsive to commands by the producer. In operation 508, the Flash movie is played.

Referring again to Figure 5, in operation 510, a connection between the client device and a server is established. Preferably, the client device includes a receiver (e.g., ActiveX, a Java Applet, or a Web server connection) and a bridge layer (e.g., a JavaScript or VBScript layer). In an alternative embodiment, for version Flash 5.0 and higher, the ActionScript object XMLSocket can be used, which allow a continuous connection with a server to be established. The Flash movie may be a stand-alone application commonly called a Projector. However, the user may desire to have the Flash movie reactive to Temporal Signals, such as breaking news stories, or stock prices. Alternatively, the Flash movie may provide core

functionality, and the producer may push new functionality to the user based on the characteristics of the user, such as user profile information. For example, the producer may learn that the user is a 30 year old male, living in Boulder, Colorado. Based on this demographic, the producer may push a Flash movie advertising high-end mountain bikes to the user.

Referring again to Figure 5, in operation 520, the producer sends a command, directing some functionality of the Flash movie, to the server. Referring to Figure 5b, a preferred method of producer control of the Flash movie is illustrated. In operation 522, the producer identifies a Temporal Signal, such as a breaking news story, or identifies a characteristic of the user. In operation 524, the producer sends a control signal to the server responsive to the Temporal Signal, or the characteristic of the user. For example, the producer could send a command to the server, perhaps using a CLI, that directs the master movie to display a Web page with a breaking news story.

In operation 530, the server transmits the control signal to the Flash movie using the connection between the client device and the server. Preferably, the control signal is sent using the continuous connection between the client device and the server established with the receiver and JavaScript embodiment. The control signal causes some functionality in the playlist of the Flash movie to be executed. In one embodiment, the control signal is a command to jump to an index in the timeline of the Flash movie, and causes the functionality at the index to execute. For example, the index may include a getURL command that fetches a document located at the URL, and displays the document in a browser window. In another embodiment, the control signal corresponds to new functionality, which is inserted in the Flash movie as a new movie with additional functionality, or in a layer between the Flash movie and the JavaScript Layer within the browser.

A third exemplary system 600 consistent with the present invention combines programming with the dynamic capabilities of Flash movies to provide a synchronized experience. The third system 600 also provides for server-side control of the Flash movie

622 whereby a producer 618, or server-side playlist, application, object, or script is capable of controlling the Flash movie. Referring to Figure 6, an embodiment of the present invention is illustrated that allows a client device 610 to receive a programming signal 604 with an embedded URI that directs the client device to address locations on the network to retrieve a Flash movie 622 located at the address, collectively, the combined signal 608. Alternatively, the Flash movie 612 may be downloaded directly to the client device from a CD-ROM, a floppy, or from a memory device connected with the client device. As discussed above, the Flash movie may include an ActionScript script. In this embodiment, a producer 618 is connected with the client device via a network 616, such as the Internet, an extranet, or wireless network, and the producer 618 directs the Flash movie 612 to synchronize the Flash movie with a programming event, and/or to relate or synchronize to other Temporal Signals.

A fourth exemplary system 700 for providing server-side control of a Flash movie on a client device, consistent with the present invention, is illustrated in Figure 7. In this embodiment, a first client device 710 receives a programming signal 704 from the programming signal source 702. The URI encoder 706 preferably embeds an URI for a Flash movie relating to the programming signal, collectively, the combined signal 708. Similarly to the first system discussed with reference to Figure 1, an URI encoder 706 encodes the URI into the programming signal 704. The first client device 710, e.g., a digital TV, set-top box, or a personal computer, extracts the URI from the combined signal 708. The second client device 712, e.g., a PDA such as a Palm™ device, a Web tablet, or a lap-top computer, has a communication link with the first client device. The communication link may be hard wired connection such as a serial, Universal Serial Bus ("USB"), parallel, or other hard wired connection, or may be through a network such as a Bluetooth™ wireless network, the Internet, an extranet, or an intranet. After the first client device 710 extracts the URI, it is sent to the second client device 712 over the communication link. Preferably, the second client device 712 is connected to a network 714, which may also provide the communication link with the first client device 710. When the second client device 712 receives the URI, the

Flash movie 718 residing at the URI is loaded on the second client device 712. The second client device 712, via the network 714, is connected with the producer 716 and/or a server. Accordingly, through the various methods discussed herein, the Flash movie 718 residing on the second client device 712 can be controlled from the producer 716 and/or a server. For example, a playlist on a server can broadcast playlist commands to all client devices subscribing to the enhanced content.

A fifth exemplary system 800 for providing server-side control of a Flash movie on a client device 806, consistent with the present invention, is illustrated in Figure 8. In this embodiment, a Flash movie is playing on a client device 806, and a data feed 802 streams into a server 804 that parses the data feed 802 and generates commands derived from the data feed 802, which are broadcast to all subscribers to the broadcast, and the data is incorporated into the Flash movie playing on the client device 806. The data feed 802 is sent to a server 804, such as a Web server or DCN server. The server 804 encapsulates the data with the appropriate command to incorporate the data into a Flash movie playing on the client device 806. The command and associated data is then broadcast to all subscribers to the data feed 802. The client device 806 preferably includes the functionality as discussed with reference to the exemplary client device illustrated in Figure 2. Accordingly, the receiver 132 receives the command and associated data. If the Flash movie is contained within a browser, for example, the command and associated data is relayed to the JavaScript layer 134, which communicates the command and associated data to the Flash movie 130, and the data is presented. Alternatively, the command and associated data may be relayed directly to the presentation layer if the Flash movie contains the receiver and presentation functionality, such as a Flash 5 movie utilizing XMLSocket functionality.

In one embodiment, for example, a user may subscribe to a stock-ticker data feed. Unlike HTML, the present invention allows the stock-ticker to continuously update, without a refresh. The stock-ticker data feed streams into the server, and the server incorporates the

data with a command. For example, the command may instruct the Flash movie to display stock prices in the upper left hand corner of the browser window.

A system 800 may control the presentation of a data on the client device in a Flash 5 movie using XMLSocket as the receiver layer. The data, for example, may be a clock that is updated every second under the control of a server. One exemplary Java application that may be compiled and installed as a server for controlling the display of a Flash clock movie on a single client device is listed below. In this example, the clock is supplied by a Java Function call.

```
import java.awt.event.*;
import java.util.*;
import java.awt.*;
import java.io.*;
import java.net.*;

/**
 *
 * ClockServer
 * Example Server for System and Method for Server-Side Control of a Flash
 * Presentation
 * Will handle one example Flash client at a time.
 *
 * Usage: java -cp ./ ClockServer port
 *
 * @author  Jeff Harrington
 */

public class ClockServer implements Runnable{
      ServerSocket clockServer;
      Socket clientSocket;
      Thread clockThread;
      PrintWriter out;

      public ClockServer(int port) {
            startClockServer(port);
      }

      private void startClockServer(int port) {
            System.out.println("Starting the clock server");
            if (clockThread == null) {
                  clockThread = new Thread(this, "Clock");
                  clockThread.start();
            }
            try {
                  clockServer = new ServerSocket(port);
                  System.out.println("ClockServer running port: " + port);
                  while(true) {
                        clientSocket = clockServer.accept();
                        out = new
PrintWriter(clientSocket.getOutputStream(), true);
                  }
            } catch(IOException ex) {
                  ex.printStackTrace();
                  System.exit(0);
```

```
            } catch(Exception ex) {
                  ex.printStackTrace();
                  System.exit(0);
                  }
            }

      public void run() {
            while (true) {
                  try {
                        sendTime();
                        clockThread.sleep(1000);
                  } catch(java.lang.InterruptedException ex) {
                        ex.printStackTrace();
                        System.exit(0);
                  }
            }
      }

      public synchronized void sendTime() {
            Calendar calendar = Calendar.getInstance();
            String time = "<TIME VALUE=\"";
            time += calendar.get(Calendar.HOUR_OF_DAY)
                  + ":" + calendar.get(Calendar.MINUTE)
                  + ":" + calendar.get(Calendar.SECOND);
            time += "\" />";
            time += '\0';
            try {
                        System.out.println("TIME = "+time);
                        if (out != null) {
                              out.print(time);
                              out.flush();
                        }
            } catch(Exception ex) {
                  ex.printStackTrace();
            }
      }

      public static void main(String args[]) {
            if(args.length == 1) {
                  ClockServer clockServer = new
ClockServer(Integer.parseInt(args[0]));
            } else {
                  System.out.println("Usage: java -cp ./ ClockServer
port");
            }
      }
}
```

Further, the following ActionScript code may be compiled into a Flash 5 movie of the client device for receiving the updated clock data from the server-side application using an XMLSocket receiver layer.

```
// Flash 5 ActionScript to be included into the example Flash clock display
// movie.
mySocket = new XMLSocket();
mySocket.onConnect = handleConnect;
mySocket.onXML = handleIncoming;
mySocket.connect("localhost", 1024);

function handleConnect() {
```

```
        trace("CONNECTED!");
}

function handleIncoming(message) {
        var time = message.firstChild;
        timeDisplay = time.attributes.value;
}

        // The Flash requires a Single Line Dynamic Text area with
        //variable name timeDisplay where the time will be displayed.
```

The server-side application listed above runs continuously and will accept one Flash movie client. The Flash movie connects to the server at startup and begins receiving an XML update expressing the time every second. The Flash movie displays the time in the text area, timeDisplay, under the control of the server-side application.

Figure 9 illustrates a sixth exemplary system 900 consistent with the present invention, which provides for server-side control of a Flash movie playing on a client device 902, wherein the client device 902 pushes commands to the server 904, which are then broadcast to a plurality of client devices (i.e., the client in effect becomes the producer). This system is useful, for example, in chat systems and multi-player games. Consider a multiplayer Flash movie game of the word game Scrabble™; according to the present invention, whenever a player makes a move by placing a new word on the board displayed on the screen of the client device 902, perhaps by selecting and dragging letters, a command is issued corresponding to the move and is sent to the server 904. The command corresponding to the move is then sent to all of the client devices, e.g., client devices 906, 908, and 910, connected to the server 904. The command is received and preferably sent through the JavaScript layer 134 to the Flash movie 130 on the client devices 906, 908, and 910.

In a chat system consistent with the present invention, the client may push commands to the server, such as: Ignore (allows a member of a chat room to filter out unwanted chat messages from specific chat room members), Whisper (allows a member of a chat room to speak to one other particular chat room member without other chat room members seeing the chat), and ChangeChatRoom (allows a member to change chat rooms). These commands can change the client state, or the server state with regard to the users interface to the chat room,

which are in addition to basic chat functionality such as sending text, and having the text broadcast to all members of the chat room.

In addition to the immediate availability of the application, a further advantage to multiplayer games, chat systems, and similar systems, is that by establishing a connection between the client device and the server, such as, but not limited to, through the receiver and the JavaScript layer on the client device or by using native Flash 5 ActionScript XMLSocket connections, whenever a command is generated in response to a user action, the command is sent to the server, and through server-side controls the command is broadcast and immediately displayed in the Flash movie on each player's client device.

Figure 10 illustrates a seventh exemplary system 1000 consistent with the present invention, which provides for server-side control of a Flash movie playing on a client device, wherein the client pushes 1002 commands to the server 1004, which are then sent to one other client device 1006. This exemplary system is useful, for example, in two-player games and instant messaging systems. Consider a two player Flash movie game of tic-tac-toe; according to the present invention, whenever a player makes a move by putting an X or an O on the tic-tac-toe grid displayed on the screen of the client device 1002, perhaps by using a drawing tool, for example drawing an X on the screen of a Palm™ PDA client device using the Graffiti™ application, a command is issued corresponding to the X and is sent to the server 1004. The command corresponding to the move is then sent to the client device 1006 of the second player. The command is received and preferably sent through the JavaScript layer to the Flash movie on the second player's client device 1006.

One particular advantage of the present invention for two-player games, instant messaging services, and similar systems, is that by establishing a connection between the first client device and the server (i.e., preferably through the receiver and the JavaScript layer on the client device) whenever a command is generated by the first client device 1002 in response to some user action, the command is immediately sent to the server 1004, and through server-side controls the command can control the Flash movie playing on the second

client device 1006 of the second participant, e.g., the move is sent by the second game player or the message is received.

The client device illustrated herein preferably includes an input mechanism for generating a command, which provides for two-way interactivity. In a client device such as a personal computer, the input mechanism is generally a keyboard or a mouse, which can be used to perhaps click on a button in the Flash movie, which will generate a command that will be sent to all players in a multiplayer game, as discussed above. Alternatively, the input mechanism can include other user input mechanisms or signal generating mechanisms wherein the output from the mechanism generates a command, which can be sent to the server, and then through server-side controls sent to the client device(s). For example, an exemplary signal generating mechanism includes a sensor, which could generate a signal corresponding to some characteristic, such as the temperature, the temperature output signal from the sensor could be converted into a command, that according to the present invention would be sent through a server to the client device(s) receiving the sensor data, perhaps through a subscription. In another example, an exemplary user input mechanism includes a virtual reality suit, which could generate signals corresponding to the movements of a person wearing the suit, the movement output signals could be converted into a series of commands, that according to the present invention would be sent through a server to the client device(s). A Flash movie playing on the client device could, for example, generate a series of Flash movie animations derived from the movement commands and display the Flash movie animations on the client device. Accordingly, a person with the second client device in a geographically remote area, could, according to the present invention, load a Flash movie on the second client device, establish a connection with the server, and receive the movement commands from the server to display Flash movie animations derived from the movement commands from the first client device, i.e., users with client devices connected to the server could view the movements of the person wearing the virtual reality suit. This would be useful in any number of practical uses, including: teaching, wherein students could remotely

view and interact with a teacher; gaming, wherein games could be taken to a new level by virtually physically interacting with other players; and viewing sporting events, where the movements of the field could be viewed on the client device and the viewer could interact through perhaps a chat service.

While the present invention has been described in relation to specific systems, hardware, devices, software, platforms, configurations, process routines, and a preferred embodiments, it is to be appreciated that the present invention is not limited to any specific embodiments, process, systems, devices, signal formats, data formats, and/or configurations. As such, the present invention may be considered to cover any and all subject matter, as specified in the attached claims.

CLAIMS

1. A method for synchronizing a programming signal with a Flash movie on a client device, the method comprising:

receiving a programming signal on a client device,

receiving an URI, wherein the URI specifies a location in a network from where a Flash movie which relates to the programming signal can be obtained;

retrieving the Flash movie from the location;

loading the Flash movie on the client device, the client device including a Flash player; and

receiving a command at the client device from the server, the command directing the Flash movie on the client device.

2. The method of claim 1, wherein the programming signal includes at least one of a video signal, an audio signal, a streaming video signal, and a streaming audio signal.

3. The method of claim 1, wherein the URI is included as at least one of received with the programming signal, embedded in the programming signal, and embedded in a vertical blanking interval of the programming signal.

4. The method of claim 1, wherein the network includes at least one of a publicly accessible network, a privately accessible network, a distributed community network, a wireless network, an extranet, an Internet, and an intranet.

5. The method of claim 1, wherein the client device includes a Web browser having a Flash plug-in.

6. The method of claim 5, wherein the Web browser includes a receiver in communication with a bridge layer, the bridge layer for transmitting the command to the Flash movie.

7. The method of claim 6, wherein the receiver includes at least one of a receiver applet, an ActiveX control, a Java applet, and a persistent socket function of a Flash movie.

8. The method of claim 1, wherein the command is received through the playback of a playlist residing on a server.

9. The method of claim 1 wherein the command is generated by a producer connected with the network.

10. The method of claim 1, wherein the Flash player includes at least one of an email client capable of displaying Flash movies, a Flash projector, a Flash plug-in with persistent socket capabilities, a Flash projector with persistent socket capabilities, and a Flash projector used as a screen saver.

11. The method of claim 1, wherein the command is received via a persistent socket.

12. A program resident on a memory device accessible by a client device for synchronizing a programming signal with a Flash movie on the client device, the program comprising:

a first program component resident on a memory device for receiving a programming signal;

a second program component resident on the memory device for receiving a URI, wherein the URI specifies a location on a network from where a Flash movie that relates to the programming signal can be obtained;

a third program component resident on the memory device for retrieving the Flash movie from the location;

a fourth program component resident on the memory device for loading the Flash movie on the client device, the client device including a Flash player; and

a fifth program component resident on the memory device for receiving a command at the client device from the server, the command directing the Flash movie on the client device.

13. The program of claim 12, wherein the program is an applet.

14. The program of claim 12, wherein the client device includes a Web browser having a Flash plug-in.

15. The program of claim 14, wherein the Web browser includes a receiver in communication with a bridge layer, the bridge layer for transmitting the command to the Flash movie.

16. The program of claim 15, wherein the receiver includes at least one of a receiver applet, an ActiveX control, a Java applet, and a persistent socket function of a Flash movie.

17. The program of claim 12, wherein the command is received from the playback of a playlist on a server.

18. The program of claim 12, wherein the command is generated by a producer connected with the network.

19. The program of claim 12, wherein the Flash player includes at least one of an email client capable of displaying Flash movies, a Flash projector, a Flash plug-in with persistent socket capabilities, a Flash projector with persistent socket capabilities, and a Flash projector used as a screen saver.

20. A system for presenting a programming signal and a related Flash movie, the system comprising:

a first means for receiving the programming signal;

a second means for receiving one or more URIs, wherein the URI specifies a location on a network for the Flash movie;

a means for decoding, connected to the second means for receiving the URI to determine the location on the network for the Flash movie;

a means, connected with the decoding means, for sending message requests to the location on the network for the Flash movie and for retrieving the Flash movie residing at the network location;

a means, connected with the means for sending message requests, for playing the Flash movie;

a presentation means, connected to the first and second receiving means, for presenting the programming with the Flash movie; and

a means for receiving a control signal from a server, the control signal controlling the Flash movie.

21. The system of claim 20, wherein the programming signal contains a video signal and an audio signal.

22. An apparatus for presenting a programming signal and a related Flash movie, the system comprising:

a decoder for receiving at least one URI, decoding the at least one URI and determining a location corresponding to the URI; and

at least one presentation device for presenting a programming signal, retrieving a Flash movie from the location and presenting the Flash movie;

whereupon receiving a programming signal and at least one URI, the decoder decodes the URI to determine the location, and the at least one presentation device retrieves the Flash movie from the location, presents the Flash movie, and receives at least one command providing direction to the presentation of the Flash movie.

23. The apparatus of claim 22, wherein the at least one presentation device includes a first presentation device for presenting the programming signal and a second presentation device for presenting the Flash movie.

24. The apparatus of claim 22, wherein the presentation device presents the programming signal on a first layer and the Flash movie on a second layer.

25. The apparatus of claim 22, wherein the presentation device presents the programming signal on a first window and the Flash movie on a second window.

26. The apparatus of claim 22, wherein the URI is received as at least one of contemporaneously with the programming signal, prior to the programming signal, separately from the programming signal, embedded in the programming signal, and embedded in a vertical blanking interval of the programming signal.

27. A memory for storing data utilized to synchronize a programming signal with a Flash movie on a client device, the memory comprising:

a data structure stored in the memory, the data structure including information used by the application program and including:

a first data object utilized to receive a programming signal;

a second data object utilized to receive an URI, wherein the URI specifies a location in a network from where a Flash movie relating to the programming signal can be obtained;

a third data object utilized to retrieve the Flash movie from the location;

a fourth data object utilized to load the Flash movie on the client device, the client device including a Flash player; and

a fifth data object utilized to receive a command at the client device from the server, the command directing the Flash movie on the client device.

28. A method for controlling a Flash movie by a server, the method comprising:

identifying a Flash movie; and

sending a command from the server to a client device, wherein the command controls the presentation of the Flash movie.

29. The method of claim 28, wherein the command is sent via the playback of a playlist residing on the server.

30. The method of claim 28, wherein the command is generated by a producer connected with the network.

31. The method of claim 28, wherein the command is generated live.

32. The method of claim 28, wherein the command is received via a command line interface.

33. A method for synchronizing a programming signal with a Flash movie on a client device, the method comprising:

receiving a programming signal on a client device,

receiving an URI, wherein the URI specifies a location in a network from where a Flash movie which relates to the programming signal can be obtained;

retrieving the Flash movie from the location;

loading the Flash movie on the client device, the client device including a Flash player;

downloading a playlist from a server;

playing the playlist on the client device, wherein the playlist controls the presentation of the Flash movie on the client device.

34. A method for controlling a Flash movie by a playlist, the method comprising:

identifying a Flash movie;

downloading a playlist onto a client device from a server; and

playing the playlist on the client device, wherein the playlist controls the presentation of the Flash movie.

35. A method for providing a real-time data feed to a client device having a Flash movie, the method comprising:

receiving a real-time data feed at the server;

generating a command at a server, the command directed to a Flash movie on the client device, and the command responsive to the real-time data feed; and

sending the command to the client device;

wherein the command sent to the client device directs the Flash movie playing on the client device.

36. The method of claim 35, wherein the server is accessible via a communications link further comprising at least one of a network, an intranet, an extranet, the Internet, a distributed community network, a publicly accessible network, a privately accessible network, a wireless network, and a stand-alone configuration separate from a network.

37. The method of claim 35, wherein the command is sent via a persistent socket.

38. The method of claim 35, wherein the real-time data feed includes at least one of a stock ticker, a sports ticker, a news ticker, an advertising ticker, and a current event ticker.

39. A computer-readable data transmission medium containing a data structure configured to provide a real-time data feed to a client device having a Flash movie, the computer-readable transmission medium comprising:

a first portion receiving a real-time data feed at the server;

a second portion generating a command at the server, the command directed to a Flash movie on the client device, and the command responsive to the real-time data feed; and

a third portion sending the command to the client device;

wherein the command sent to the client device directs the Flash movie.

40. A computer readable medium providing a data structure configured to provide a real-time data feed to a client device having a Flash movie by:

receiving a real-time data feed at the server;

generating a command at a server, the command directed to a Flash movie on a client device and responsive to the real-time data feed; and

sending the command to the client device;

wherein the command sent to the client device directs the Flash movie.

41. A signal embodied in a transmission medium for controlling the presentation of a Flash movie on a client device, comprising:

a first program code segment providing an abstraction of a first receiver for receiving a programming signal;

a second program code segment providing an abstraction of a second receiver for receiving an URI, the URI specifying a location on a network of a Flash movie;

a third program code segment providing an abstraction for retrieving the Flash movie from the location;

a fourth program code segment providing an abstraction for loading the Flash movie on the client device, wherein the client device includes a Flash player;

a fifth program code segment providing an abstraction for receiving a command from a server, the command directing the presentation of the Flash movie on the client device.

42. The signal embodied in a transmission medium of claim 41, wherein the URI is included as at least one of received with the programming signal, embedded in the programming signal, and embedded in a vertical blanking interval of the programming signal.

43. The signal embodied in a transmission medium of claim 41, wherein the client device includes a Web browser having a Flash plug-in.

44. The signal embodied in a transmission medium of claim 41, wherein the command is received through the playback of a playlist residing on a server.

45. The signal embodied in a transmission medium of claim 41 wherein the command is generated by a producer connected with the network.

46. The signal embodied in a transmission medium of claim 41, wherein the Flash player includes at least one of an email client capable of displaying Flash movies, a Flash projector, a Flash plug-in with persistent socket capabilities, a Flash projector with persistent socket capabilities, and a Flash projector used as a screen saver.

47. The signal embodied in a transmission medium of claim 1, wherein the command is received via a persistent socket.

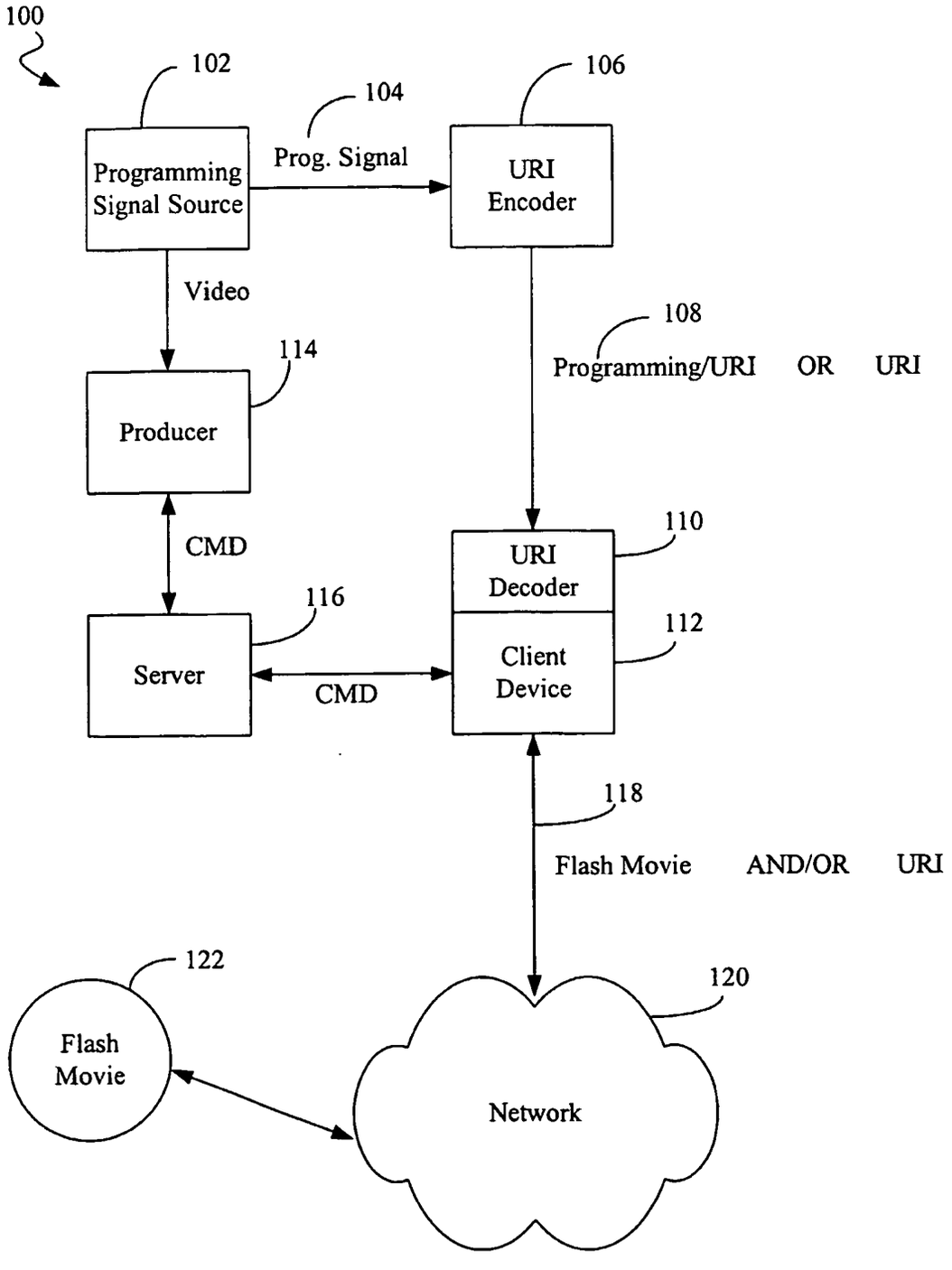

100
102
104
106
Programming Signal Source
Prog. Signal
URI Encoder
Video
108
Programming/URI OR URI
114
Producer
CMD
110
URI Decoder
116
112
Server
CMD
Client Device
118
Flash Movie AND/OR URI
122
120
Flash Movie
Network

Fig. 1

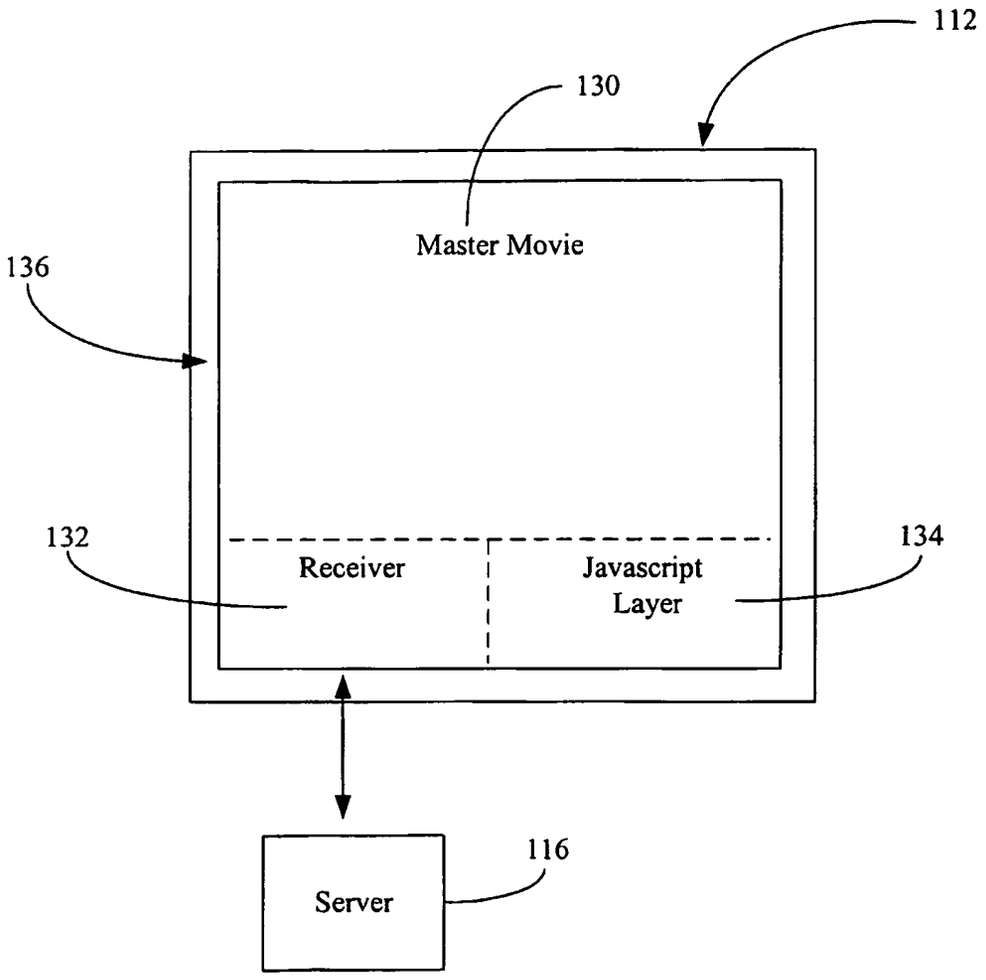

112
130
Master Movie
136
132
Receiver
Javascript
Layer
134
116
Server

Fig. 2

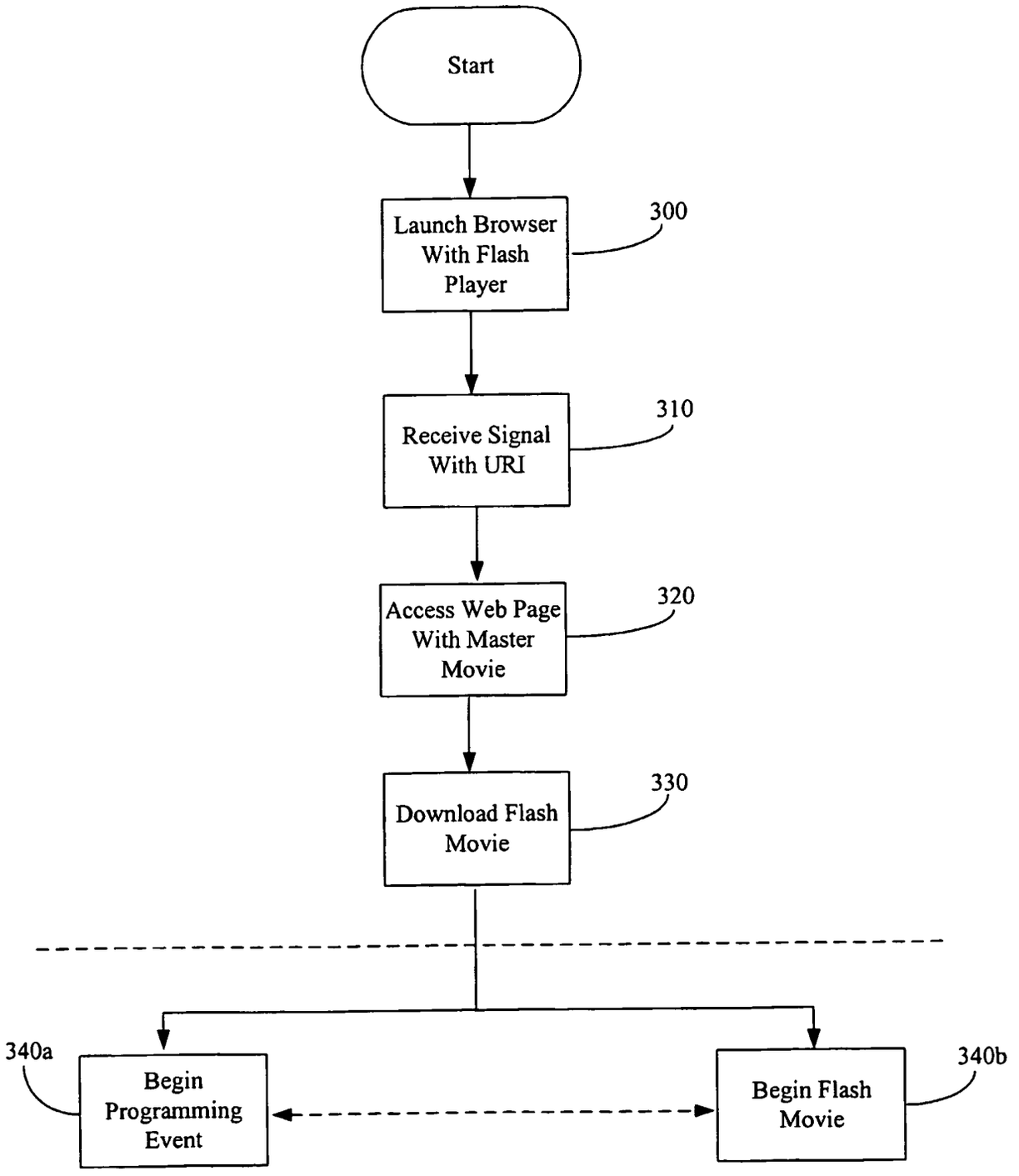

Start
Launch Browser With Flash Player
300
Receive Signal With URI
310
Access Web Page With Master Movie
320
Download Flash Movie
330
340a
Begin Programming Event
Begin Flash Movie
340b

Fig. 3

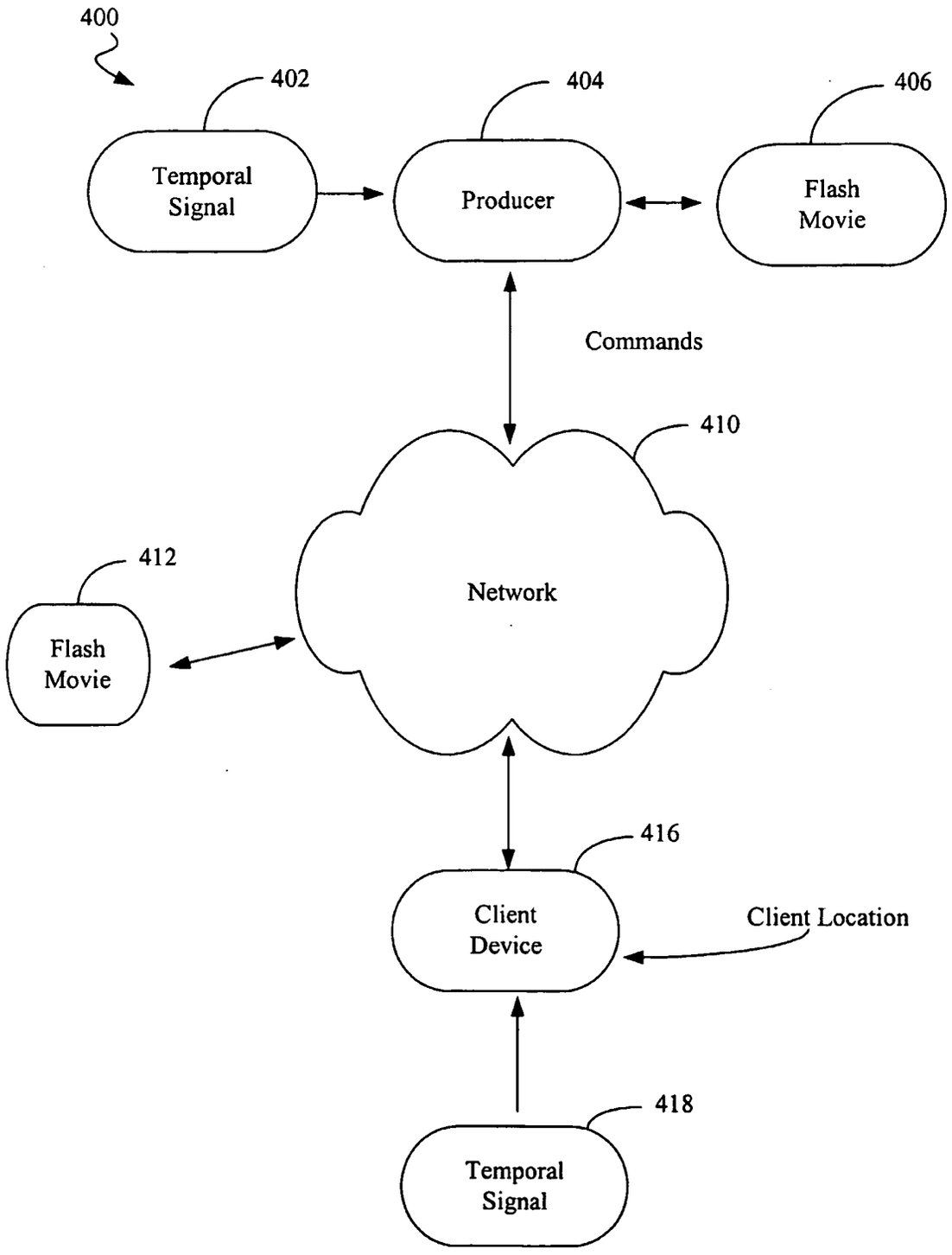

400
402
Temporal Signal
404
Producer
406
Flash Movie
Commands
410
Network
412
Flash Movie
416
Client Device
Client Location
418
Temporal Signal

Fig. 4

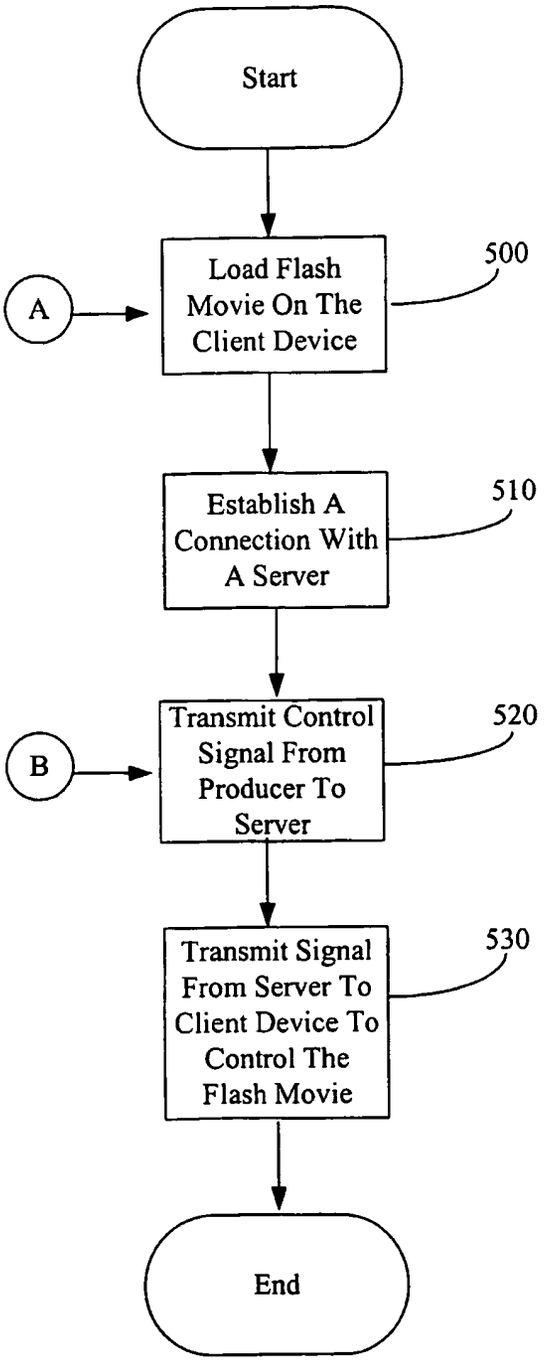

Start
A
Load Flash Movie On The Client Device
500
Establish A Connection With A Server
510
B
Transmit Control Signal From Producer To Server
520
Transmit Signal From Server To Client Device To Control The Flash Movie
530
End

Fig. 5

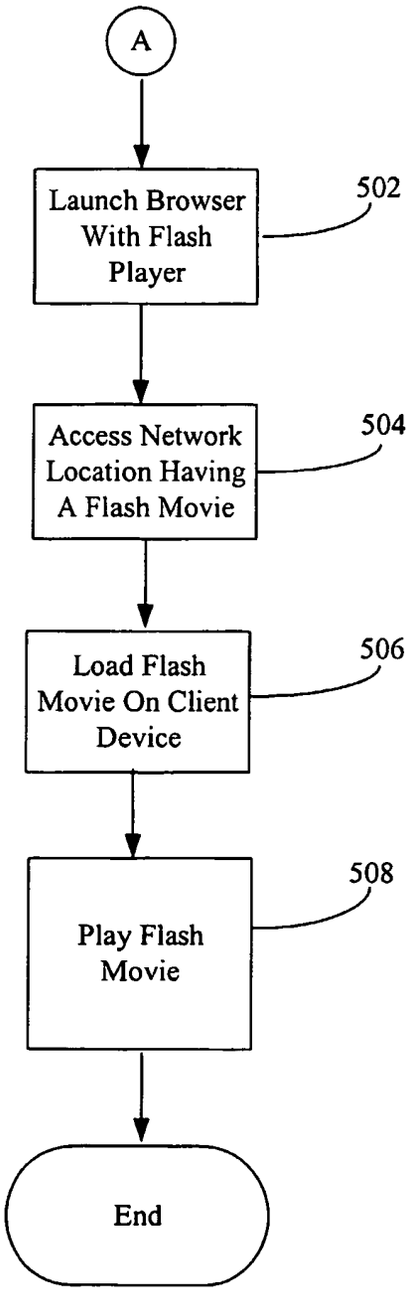

A
Launch Browser With Flash Player
502
Access Network Location Having A Flash Movie
504
Load Flash Movie On Client Device
506
Play Flash Movie
508
End

Fig. 5a

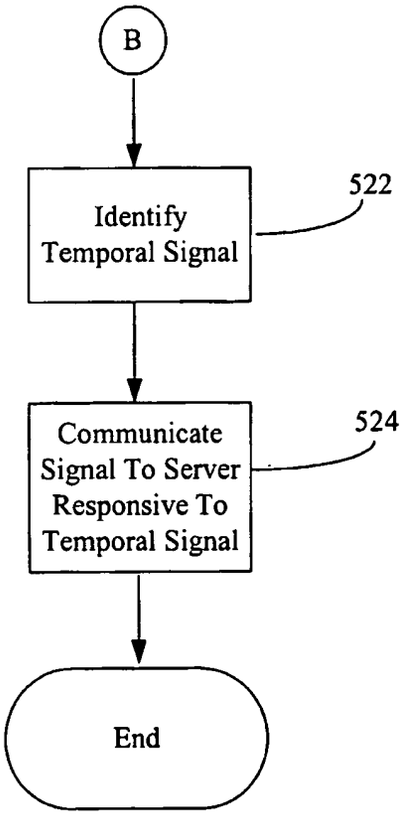

B
Identify Temporal Signal
522
Communicate Signal To Server Responsive To Temporal Signal
524
End

Fig. 5b

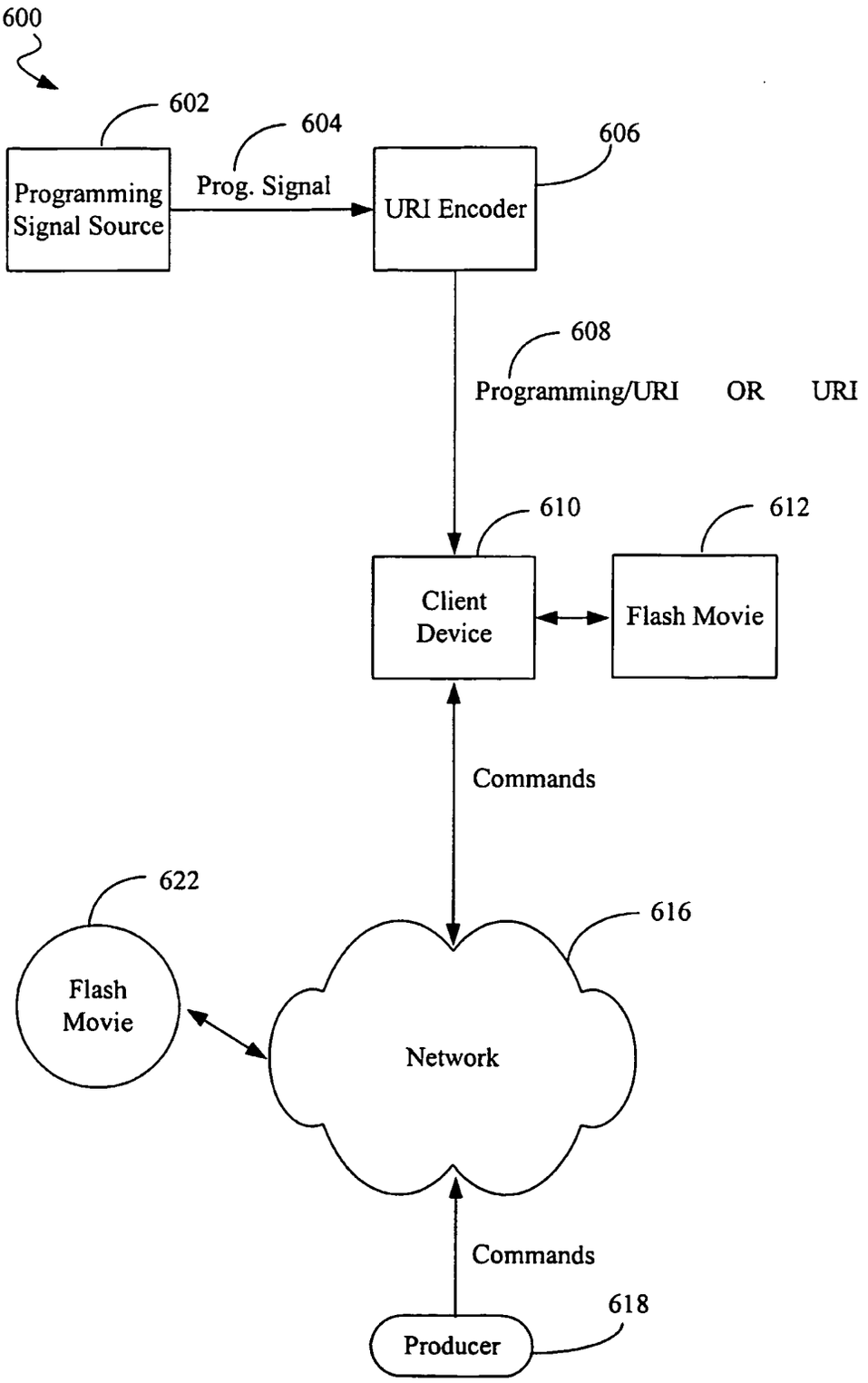

600
602
604
606
Programming Signal Source
Prog. Signal
URI Encoder
608
Programming/URI OR URI
610
612
Client Device
Flash Movie
Commands
622
616
Flash Movie
Network
Commands
618
Producer

Fig. 6

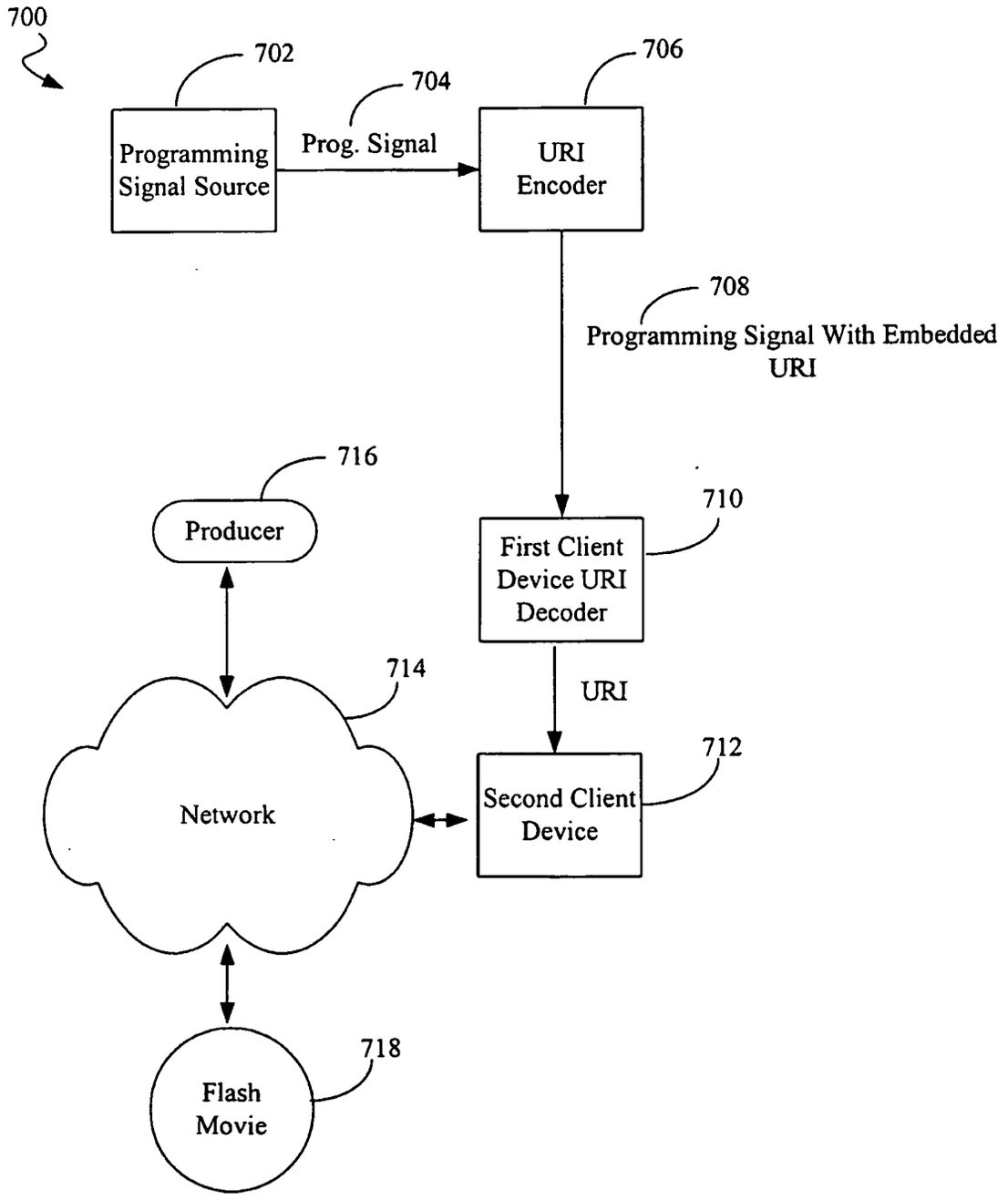

700
702
704
706
Programming Signal Source
Prog. Signal
URI Encoder
708
Programming Signal With Embedded URI
716
Producer
710
First Client Device URI Decoder
714
URI
712
Network
Second Client Device
718
Flash Movie

Fig. 7

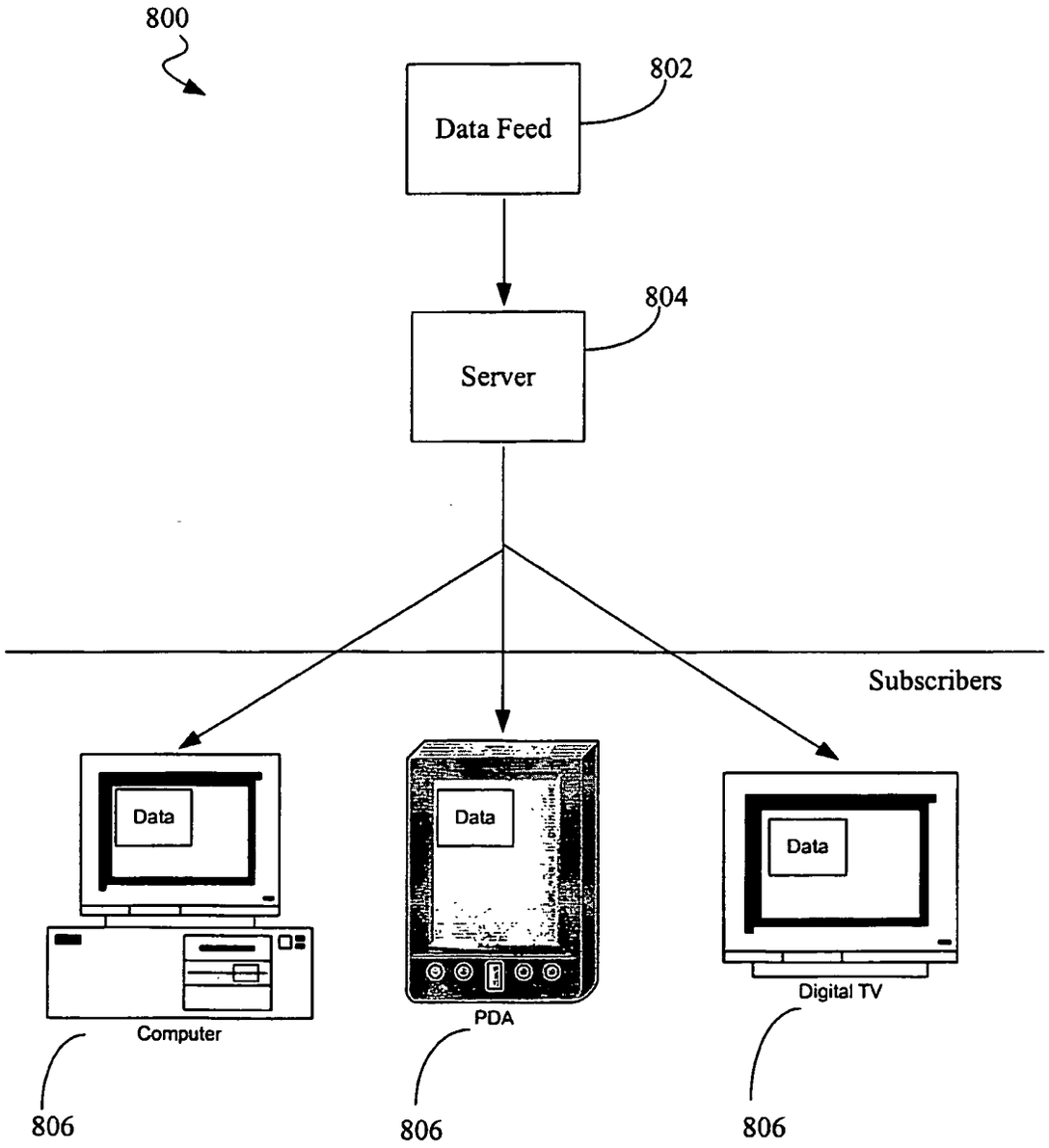

800
802
Data Feed
804
Server
Subscribers
Data
Computer
Data
PDA
Data
Digital TV
806
806
806

Fig. 8

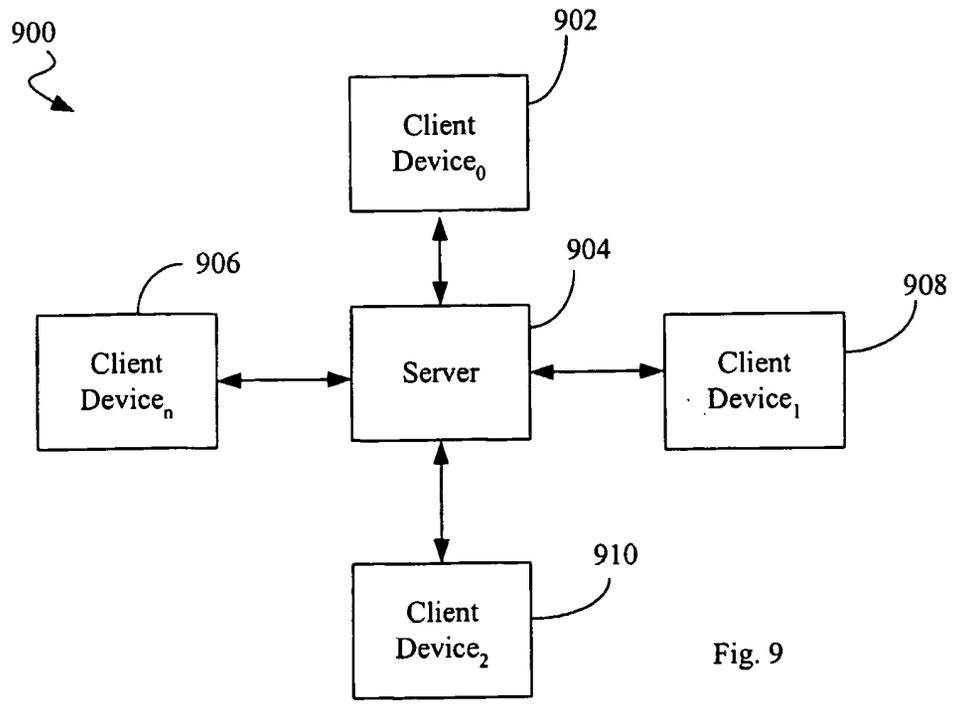

900
902
Client Device$_0$
904
906
908
Client Device$_n$
Server
Client Device$_1$
910
Client Device$_2$

Fig. 9

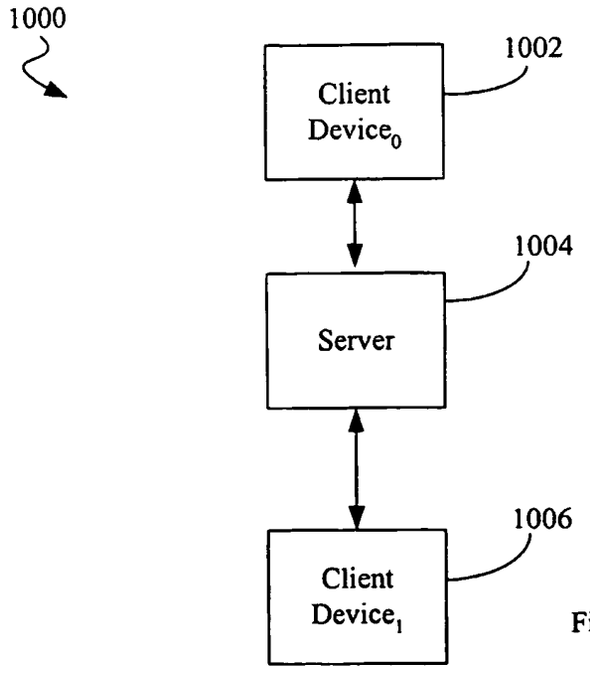

1000
1002
Client Device$_0$
1004
Server
1006
Client Device$_1$

Fig. 10

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 August 2002 (22.08.2002)

PCT

(10) International Publication Number
WO 02/065318 A2

(51) International Patent Classification[7]: G06F 15/16

(21) International Application Number: PCT/US02/04683

(22) International Filing Date: 15 February 2002 (15.02.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/269,592	15 February 2001 (15.02.2001)	US
Not furnished	14 February 2002 (14.02.2002)	US

(71) Applicant: ACTV, INC. [US/US]; 10th Floor, 233 Park Avenue South, New York, NY 10003 (US).

(72) Inventor: ABATO, Michael, Raymond; 64 Prescott Avenue, White Plains, NY 10605 (US).

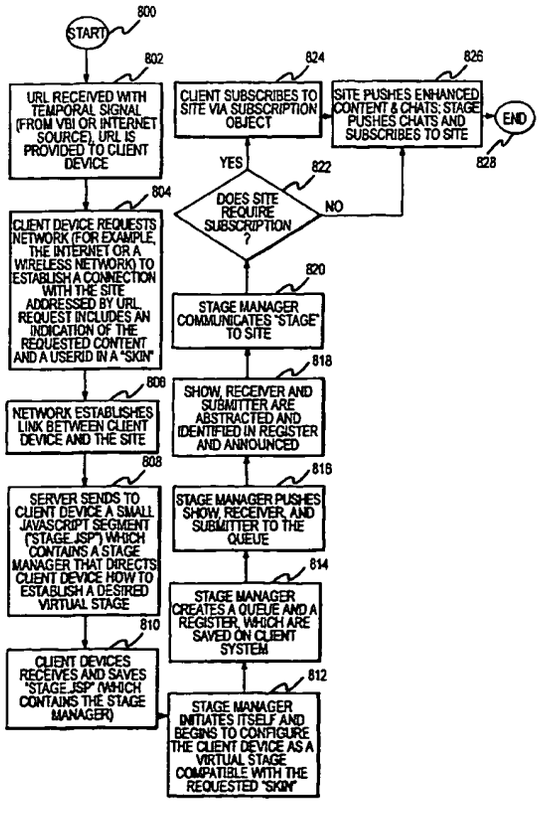
(74) Agents: KENNEDY, John, T. et al.; DORSEY & WHITNEY LLP, Suite 4700, 370 17th Street, Denver, CO 80202 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A SYSTEM AND PROCESS FOR CREATING A VIRTUAL STAGE AND PRESENTING ENHANCED CONTENT VIA THE VIRTUAL STAGE



(57) Abstract: A system and process for creating a Virtual Stage on a client device and presenting Enhanced Content on the Virtual Stage is provided. The Virtual Stage is preferably implemented on a Browser or similarly equipped presentation device. The Virtual Stage enables any presentation device to receive Enhanced Content from any provider regardless of the capabilities of the Browser or client device, the data format of the Enhanced Content, and/or the communications medium utilized to communicate the Enhanced Content to the Browser or client device. The Virtual Stage suitably includes an abstracted Show Object which provides a framework for presenting the Enhanced Content and an abstracted Receiver Object which provides a receiver for communicating with the Enhanced Content provider and receiving the Enhanced Content. Alternative embodiments may also include multiple abstracted Receiver Objects and/or abstracted Subscribers, which facilitates communications with subscription service provider systems, including, but not limited to, chat service systems.

WO 02/065318 A2

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

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.

SPECIFICATION

A SYSTEM AND PROCESS FOR CREATING A VIRTUAL STAGE AND PRESENTING ENHANCED CONTENT VIA THE VIRTUAL STAGE

of which the following is a specification.

REFERENCE TO RELATED APPLICATIONS

This application claims priority to and incorporates by reference, in its entirety, the U.S. application filed on February 14, 2002, in the name of inventor Michael R. Abato and entitled "A System and Process for Creating a Virtual Stage and Presenting Enhanced Content Via the Virtual Stage," as further identified by attorney docket number 10016.02. This application claims priority to and incorporates by reference, in its entirety, U.S. provisional application serial number 60/269592, by Michael R. Abato, entitled "A SYSTEM AND METHOD FOR PRESENTING CONTENT RELATED TO A TEMPORAL EVENT TO A USER VIA A VIRTUAL STAGE" which was filed on February 15, 2001. The present application is also related to the following applications, which are incorporated, in their entirety, herein by reference: U.S. application serial number 09/396,693 of Craig D. Ullman, Michael R. Abato, Jeffrey M. Harrington, and Carl R. Duda, entitled "ENHANCED VIDEO PROGRAMMING SYSTEM AND METHOD FOR PROVIDING A DISTRIBUTED COMMUNITY NETWORK," and filed on September 15, 1999 (hereafter, the "DCN application"); and U.S. provisional

application serial number 60/269,593 of Jeffrey M. Harrington, entitled "A SYSTEM AND METHOD FOR SERVER SIDE CONTROL OF FLASH," and filed on February 15, 2001 (hereinafter, the "FLASH application").

BACKGROUND OF THE INVENTION

Today, the capabilities of computers to provide massive amounts of educational and entertainment information has exploded with the Internet. The Internet has the power to transform society through unprecedented levels of information flow between members. Currently, on-line systems offer a variety of different services to users, including news feeds, electronic databases (either searchable by the user directly on the on-line system, or downloadable to the user's own computer), private message services, electronic newsletters, real-time games for play by several users at the same time, and job placement services, to name a few. However, today, most on-line communications occur merely through text. This currently stands in great contrast to the audio/visual presentation of the alternative electronic medium, television. However, it is expected that as multi-media's incessant growth continues, audio/visual programs will proliferate and text will become less and less dominant in the on-line environment. Even though these programs will be introduced, the Internet will remain essentially user unfriendly due to its very massiveness, organization and randomness. Simply stated, there is no order or direction in the Internet. Specific pieces of information are many times hard to find, and even harder is the ability to put that piece of information into a meaningful context.

Television, on the other hand, has been criticized for being a passive medium– "chewing gum for the eyes," as Fred Allen once observed. Television has always been something you watched, not something you do. Many social critics believe that the passivity television depends on has seeped into our entire culture, turning a nation of citizens into a nation of viewers. While interactive television systems have increased the level of user interaction and provided greater learning and entertainment opportunities, vast information resources, such as databases, are inaccessible from such a medium.

Recent innovations in combining Internet content with television and other audio and/or video programming signals have been described in various patents and publications, for example, United States Patent Number 5,778,181, which issued on July 7, 1998 to Jack D. Hidary, et al., United States Patent Number 6,018,768, issued

on January 25, 2000 to Craig Ullman and Jack Hidary et al., and also in United States Patent Number 5,774,664, which issued on June 30, 1998 to Jack D. Hidary, et al. (hereinafter, collectively the "Hidary patents"). The contents of these patents are herein incorporated by reference, in their entirety. As is now well known in the art, these patents describe innovative systems and processes for combining the user-friendly visual experience of television programming signals, and other time based events or signals, with information resources located on the Internet which relate to the programming signal (hereinafter, the "Enhanced Content"). Since segments in a programming signal are generally presented in a sequence to a client based upon a reference to a known event (for example, the amount of time remaining in a football game is based upon the kick-off, or the amount of time remaining in a recorded movie is based upon when the playback of the movie is started and not when it was actually filmed), such programming signals shall herein be regarded as applying to any signal, show, or sequence of events, whether pre-recorded or live, which are defined or based upon a temporal relationship (hereinafter, the "Temporal Signal"). Such Temporal Signals may include live events (for example, a cut-away by a television broadcaster to a then breaking news event), pre-recorded events, and combinations of live and pre-recorded events.

Recently, various approaches have been implemented for providing client side and server side systems capable of providing Enhanced Content related to a Temporal Signal. As is well known in the art, such approaches generally require a client to download (commonly from an Internet based Web site) and then install a Java applet which configures the client's system as a specific application. Another approach utilizes a client system, such as a Web Browser or an equivalent system including, but not limited to, a Flash player or an XML browser (hereinafter, collectively, a "Browser"), and a downloaded program which configures the client system to retrieve Enhanced Content over a specific type of communications link, for a specific type of client device, based upon the reception of a Temporal Signal and an address identifying a provider of Enhanced Content related thereto. Regardless of the specific methodology utilized, today's client systems commonly must execute a large download to receive and present Enhanced Content program segments which relate to a given Temporal Signal.

Further, since the Internet has innumerable sites, which a client may or may not find using a search engine, producers of Temporal Signals often identify a location providing Enhanced Content (for example, an Internet site) by presenting, in

the video or audio portion of a Temporal Signal, a Uniform Resource Identifier (URI), as defined in the RFC 2396 which includes, for example, a tangible Web asset, a Uniform Resource Locator, a Uniform Resource Name, a functional push and an object push (hereinafter, collectively referred to as an "URI"). Once the site is identified by the client and/or the client's device or system (hereinafter, "client device" and "client system" are utilized interchangeably), the client system commonly registers the client with the provider of the Enhanced Content.

Following registration, the client often needs to identify a program or segment for which the client desires to receive the related Enhanced Content (since Enhanced Content for multiple programs may be accessible from a single Internet site). Once selected, the client system then often downloads and installs a Java applet, or similar program code, which configures the client device to receive the related Enhanced Content. At this point, the client system is then ready to connect to the provider of the Enhanced Content, satisfy any pre-requisites (for example, providing a password, sign-on, or user profile information), and receive the Enhanced Content.

As such, the approaches commonly utilized today to receive Enhanced Content generally require a client to first identify the location of a provider of Enhanced Content, register the client with the provider, download a program which configures the client system, installs the program, connects to a site providing Enhanced Content related to a specific Temporal Signal, and then satisfies any pre-requisites prior to receiving the Enhanced Content (for example, providing user profile information). In short, these approaches require so much time and effort to configure the client system and access a provider of the Enhanced Content that many clients are discouraged from utilizing such an approach.

What is needed is a system and process which reduces and minimizes the amount of time and effort required by a client device to automatically, or upon request, receive Enhanced Content related to a Temporal Signal. What is needed is a wider, richer, quicker, and more efficient system and process for receiving and processing audio/visual and textual database elements into an organized unique interactive, educational, entertainment experience.

SUMMARY OF THE INVENTION

Systems consistent with the present invention provide a system and process for combining Temporal Signals (which appear, for example, on a television broadcast, a VHS or Beta tape, CD-ROM, DVD, CD, or other medium) with Enhanced Content (accessible, for example, via the Internet) without requiring

lengthy downloads, specific client devices or operating systems, specific data formats, or similar constraints. By abstracting the concept of a Virtual Stage which contains at least a Show Object and a Receiver Object, the present invention allows Enhanced Content to be received and processed by virtually any system regardless of configuration, device, hardware, software, communications links utilized, or practically any other factor. Preferably, such a system is implemented on a client device capable of hosting a Browser. As such, the present invention is described in the context of using a Browser for supporting its operations.

Further, the present invention creates a new, efficient, dynamic, diverse and powerful educational and entertainment medium. The system allows consumers to receive more information in a more efficient manner than either television or the Internet alone and over prior systems and processes utilized to present Enhanced Content related to a Temporal Signal. Instead of requiring client systems to execute lengthy, and sometimes problematic, downloads prior to receiving an Enhanced Content segment, the present invention streamlines such processes by removing the determination of a type of Receiver Object utilized by a client device from the initialization processes. As such, by using the new systems and processes, consumers not only can see a news report on television, but they can also read pertinent information about the report, as well as explore related information about the story regardless of the device, type of Browser or platform utilized by the client system, and/or the source of the Enhanced Content. The act of viewing a program has now become a more engaging, enriching experience, because Enhanced Content can now be obtained almost instantaneously without any lengthy downloads, initialization routines, or constraints upon compatible systems or sources.

The systems and processes of the present invention can also create a more intimate relationship between the client and the program. For example, in an educational environment, a student (the client) might be solving problems or performing virtual experiments on an Internet site that a teacher is discussing in an educational television program. Similarly, the consumer might be solving problems that the fictional characters in a television program must solve. In both cases, the consumer is an active participant in the process, rather than a passive observer. Unlike previous systems, the present invention enables the student and the teacher to visit the classroom via any client device, including the ever more increasingly popular wireless devices such as personal data assistants and wireless communications devices. Such capabilities are possible with the present invention, because the

invention provides for the creation of an abstraction of a Receiver Object that allows a client device (regardless of the device's specific configuration and/or capabilities) to receive Enhanced Content without requiring those elements responsible for presenting the Enhanced Content to know from where and/or how such content was obtained. Instead of the entire Browser or a specific application software being concerned with when and how Enhanced Content segments are received, in the present invention, only an abstracted Receiver Object (i.e., an Application Program Interface (API)) is so concerned. This enables the Show Object (i.e., another API that is responsible for actually formatting and presenting the Enhanced Content to the client) to function without concern as to the origin of the Enhanced Content segment(s) to be presented. Thus, the abstracted Show Object is much like a car engine in that the engine doesn't care where or how it gets the fuel it needs, it is merely concerned with utilizing the fuel made available to it as efficiently as possible while maximizing the performance of the car.

Another advantage of the system is that it changes the nature of advertising. Since additional information can be now given, via the present invention, to consumers automatically and without large downloads. The system enables the advertising to be more interactive, responsive, and substantive. Such real-time responsiveness allows customers to make more informed and/or spontaneous choices. Now, the act of purchasing a product seen on television or listened to via radio or other sound system can be streamlined -- the consumer can be given the choice of buying the product instantly using the two-way capabilities of the system. For example, the processes of the present invention enable an Enhanced Content provider to quickly establish a virtual store on whatever device the client is using. Preferably, the client device includes a Web browser, but a Web browser is not required to implement the present invention. Thus, all that is needed to establish such a virtual store, is the configuring of a Virtual Stage and then the pushing or pulling (for example, in FLASH) of those store elements needed (racks, clothes, cashiers, etc.) to/by an abstracted Receiver Object which, as directed by the Stage Manager, provides the Enhanced Content (for example, in a FLASH format) to an abstracted Show Object for presentation to the client.

In addition, users can take advantage of the two-way capabilities of the Internet to respond to polls, to send e-mail or to link to additional sites. For example, a viewer watching a television news program, through the system of the invention, can receive a stream of Web pages (i.e., Enhanced Content) which provide additional,

specific information relating to the news content. Such information might also be received as a chat message (provided in any format), which the client device suitably processes via the Receiver Object API, the Show Object API and a Subscriber Object API (e.g., configured to process chat messages) and suitably presents, via the Virtual Stage, to the client.

Further, video programming and corresponding Enhanced Content (which may include static or dynamic Internet pages) can be viewed on a personal computer equipped with a television card and a Browser. The Virtual Stage approach, which utilizes a Show Object API and a Receiver Object API in the preferred configuration in conjunction with a Browser to create a Virtual Stage, enables any suitably configured client device to receive a Temporal Signal and Enhanced Content via a Virtual Stage, regardless of the device or the communications medium utilized.

By simplifying those systems and processes needed to marry the appeal of Temporal Signals (for example, a video signal) with the two-way data transfer capabilities of the Internet, a powerful new medium is created which allows producers of Temporal Signals and creators of Enhanced Content to combine their mediums and content without concern as to where, when or how such Enhanced Content will be received and/or processed by a client device. Such capabilities will allow advertisers to extend their brand identity and differentiate their program offerings to the millions of people who may not have a Web enabled television or a home personal computer but are equipped with a wireless Personal Data Assistant (PDA), telephone, pager, or similar device which can be configured to support a Virtual Stage. In addition to providing significant and immediate benefits to broadcasters and advertisers, the system will also present educational programmers with a way to more effectively use Internet resources in a classroom which is not bound by physical and/or temporal constraints.

Additionally, just as a Receiver Object API, a Show Object API, and other elements (for example, a Subscriber Object API) are utilized to create a Virtual Stage, an abstraction of a location providing Enhanced Content (i.e., a Virtual Site API) is also provided for by the extension of the client side processes to the server side. The Virtual Site API makes possible the creation of playlists (which identify those Enhanced Content segments that are to be synchronized with or are somehow related to a Temporal Signal). Further, the Virtual Site facilitates the creation of playlists without concern as to how, when, or where an Enhanced Content segment will be communicated to the client device. The Virtual Site allows programmers and other

Enhanced Content creators to focus merely upon the creation of programs and related Enhanced Content. The Virtual Site also enables a producer to access Enhanced Content segments, regardless of the origin, and then reconfigure such segments into a format supported by a particular communications medium and/or client device. Thus, when a Virtual Site is utilized, two different classifications of personnel may be utilized to create and present Enhanced Content related to a Temporal Signal. These classifications are: programmers/content creators, who create the Enhanced Content segments based upon a Virtual Site; and producers, who control the Virtual Site such that the Enhanced Content, regardless of origin or format, is appropriately configured and transmitted to a client device over a given communications link.

For example, a programmer creates a playlist for a game show question and answer segment (i.e., the Enhanced Content segment) based upon an abstraction of a transmitting site, i.e., a Virtual Site. The game show segment is then provided to the producer who determines (in advance of the Temporal Signal or on a real-time basis) that client devices will/are request(ing) access to the Enhanced Content segment via the Internet, a direct cable modem link, and/or a wireless link and that the segment will be presented to the client via a PDA. Instead of having to create a playlist for each of the above types of communications links, the programmer can create one playlist of segments which are directed towards the Virtual Site. The segments are then converted by the producer, as necessary, into a format compatible with each type of communications link and/or device over which requests for the Enhanced Content segment are received.

Thus, the present invention provides abstractions of a Virtual Stage and/or a Virtual Site. The Virtual Stage enable clients to receive Enhanced Content segments, preferably via a Browser provided on their device, without experiencing lengthy downloads and initialization sequences. Similarly, the Virtual Site enables programmers and creators of Enhanced Content to create segments without concern as to the specific communications links or client devices utilized to receive the segments.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram of a prior art system design showing the receipt and decoding of video signals at the subscriber location.

Figure 2 is a diagram showing an alternative prior art system embodiment to achieve the integration of the Enhanced Content with the Temporal Signal (as represented by a video signal) by decoding a URI at a server site and then transmitting the URIs to the subscriber client devices via the Internet.

Figure 3 is a flow diagram of the basic software design utilized in the prior art to provide Enhanced Content related to a Temporal Signal to a client devices.

Figure 4 is a diagram showing another prior art system which provides for the direct transmission of URIs over the Internet to the client devices at a broadcaster's entered time without encoding the URIs into the VBI.

Figure 5 is a diagram of another prior art system which utilizes a digital cable box as an element of a client device for presenting Enhanced Content related to a Temporal Signal.

Figure 6 is a diagram of another prior art system which utilizes a digital television as an element of a client system for presenting Enhanced Content related to a Temporal Signal.

Figure 7 is a diagram of a prior art distributed communications server embodiment for providing Enhanced Content related to a Temporal Signal to a client device.

Figure 8 is a flow diagram illustrating the process by which a Virtual Stage and Enhanced Content is presented to a client via a client device for a preferred embodiment of the present invention.

DETAILED DESCRIPTION

As discussed above, the present invention includes a system and process for providing Enhanced Content related to a Temporal Signal to a client device without requiring lengthy downloads and/or timely initialization sequences. The system and process may be implemented on any client device or system capable of receiving Enhanced Content related to a Temporal Signal. For purposes of illustration, such a system and process is described in relation to a system and process configured for communicating Enhanced Content related to a Temporal Signal as disclosed and discussed in the previously identified Hidary patents, the DCN application and also the FLASH application, the text of some of which are explicitly incorporated herein (for purposes of clarity) and otherwise incorporated by reference.

As discussed in the previously mentioned references, a system for combining the rich visual capabilities of video with the vast resources of the Internet is shown in Figure 1. As shown, such a system is preferably a computer based system which receives a Temporal Signal (for example, a video program) along with embedded URIs, which direct the client's device 16 (for example, a personal computer or other device capable of receiving Enhanced Content) to address locations, or Web sites, on the Internet 20 to retrieve at least one Web page associated with the Enhanced

Content that relates to the Temporal Signal. The particular Temporal Signal may include, but is not limited to, audio, video, textual, graphic, and virtual segments, and may be delivered in any format, for example, analog, digital or digitally compressed formats (e.g., MPEG 2, MPEG 4 and MPEG 7) via any transmission means, including satellite, cable, wire, television broadcast, wireless network, or via the Web.

The Temporal Signal is preferably created at a centralized location (i.e., content creation 4 as shown in Figure 1) for distribution to clients at any location where a client device 16 can receive a Temporal Signal (for example, in their homes, car or via wireless device). Creation of the Temporal Signal is accomplished according to any conventional means known in the art. After a Temporal Signal is created, URIs identifying Enhanced Content related to the Temporal Signal are embedded into the Vertical Blank Interval (VBI) of the video programming segment of the Temporal Signal by, for example, the URI encoder 8, as shown in Figure 1. In this embodiment, the URIs are encoded onto eight fields of line 21 of the VBI. Line 21 is the line associated with close captioning, among other things. However, the URIs could also be embedded in other fields of the VBI, in the horizontal portion of the video, as part of the audio channel, in any sub-carrier to the video or, if digital, in one of the data fields.

Although Figure 1 shows the video with URIs over the same transmission line, the URIs can be sent to a client device 16 independently of the Temporal Signal (and independent of the video programming segments) on a separate data channel. In this embodiment, the URIs can be forwarded to the remote sites either prior to initiation or during the program. Preferably, the URIs have associated time stamps which indicate to the subscriber stations when, during the Temporal Signal, to present the particular Enhanced Content addressed or associated with the Temporal Signal to the client device. Alternatively, client can select when to call the particular Enhanced Content for presentation with the Temporal Signal (for example, a presentation of a Web page, a FLASH page, or any other segment at a specific time point during a video program).

Once the Temporal Signal is created, it can be transmitted to client devices 16 over any transmission means including broadcast, cable, satellite, wireless, or Internet. Further, the Temporal Signal may reside on video servers, be presented live, or even contain, in whole or in part, pre-recorded signals, for example, those provided on a VHS or Beta tape, DVD, CD, memory stick, or other medium.

Preferably, each receiver station comprises any Intel x86 machine (preferably a 486 processor, Pentium processor, etc.), an Apple Computer, UNIX or any other

type of generic purpose workstation or standard computer workstation. Those skilled in the art appreciate that as the miniaturization of electronic computing devices (specifically computer workstations) continues, such devices may become available in various configurations of size, shape and capabilities. Thus, the client devices 16 capable of utilizing the present invention are not to be construed as being limited to any specific embodiment of a computing device. As such, any device capable of presenting the Enhanced Content may be utilized to provide the features and functions of the present invention including, but not limited to, a personal computer, a computer workstation, a wireless personal computer, a PDA, a wireless communications device, a main frame computer, and any other device capable of receiving Enhanced Content and/or a Temporal Signal.

In the embodiment shown in Figure 1, the client device 16 is preferably connected to either a cable and/or broadcast television connection or to a local VCR or other video source. At each client site, the client device 16 preferably receives the Temporal Signal by a cable connection. The video/audio program can then be processed for display on a video screen using known in the art systems. For example, in a personal computer embodiment, any conventional PC card capable of displaying NTSC signals on a computer monitor, such as a WinTV card may be used. In addition to the cable connection, however, an Internet 20 connection is also provided concurrently with the cable connection.

The Internet 20 connection can be via any systems, devices, methods, or mediums capable of establishing communications between a client device and an Internet server including, but not limited to, high-speed lines, radio frequency signals conventional modems or by way of a two-way cable carrying the video or audio programming. The client device 16 has Internet access via any of the current ASCII software mechanisms. In one embodiment, at each subscriber home, an associated local URI decoder 12 receives the cable video television program, as shown in Figure 1. The local URI decoder 12 extracts the URIs, preferably embedded in the VBI of a video portion of the Temporal Signal, with the use of any conventional VBI decoder device. The URI decoder 12 may be either a stand-alone unit or a card which is implemented by the client device 16.

In another embodiment shown in Figure 2, the URIs are encoded into the video portion of a Temporal Signal in the same manner as described above. Again, the URIs are preferably encoded onto eight fields of line 21 of the VBI, but may also be sent independently of the video. In this embodiment, the URI decoder 24 is

located at the server site, as opposed to the client location. When the decoder 24 receives the video program signal, it strips out the URI codes on line 21 of the VBI and delivers these codes independently to an Internet server 28. The URI code is then subsequently delivered over the Internet 20 to the client device 16. Simultaneously, the Temporal Signal is broadcast over conventional broadcast or cable transmission means 36 to the client's device 16.

Another embodiment of a compatible system is shown in Figure 4. This system does not depend on, or even use, the VBI. In this embodiment, the system provides an online service over the Internet 20. This service is in the form of an Internet Web site 62 that provides a client-interface to a database 78 and to one or more associated data servers 90. Member-accounts are provided to TV broadcasters 66 who sign up to use the system in conjunction with their broadcasts. Each member broadcaster will enter the service at their computer 70 through Browser software 74 using their member account by entering various identification and password information. Once within their account, the member will be provided with a graphical user interface for pre-scheduling Enhanced Content (which may include URIs) for transmission to clients 118 over a direct Internet connection 94 at particular times of the day. The same client interface, or a variation on it, can be used by broadcasters for live transmission 82 of URIs (or Enhanced Content) to clients at the same time as a broadcast 86.

Other embodiments of a compatible system may be configured that do not depend on, or even use, the VBI, or the Internet for establishing a connection between the client device 16 and the location providing the Enhanced Content. For example, one such system may directly provide Enhanced Content, for example, via a wireless network, over a telephone network, a satellite network, or even a local area network. Thus, it is to be appreciated that the present invention is not limited to a system which utilizes a television video signal, VBIs, URIs, the Internet or any other device or system to provide Enhanced Content related to a Temporal Signal. Further, the present invention may be configured such that the URIs (or other triggers) are embedded in static media (for example, a CDROM). In such an embodiment, the static media, versus a propagated programming signal, pushes the triggers/URIs which are then utilized in accordance with the systems and processes for creating a Virtual Stage and presenting Enhanced Content on a Virtual Stage, as discussed in greater herein.

It is further appreciated that various embodiments can also be used which enable personalization in the form of unique series of Enhanced Content specific to each client's unique profile, which are directly sent over a network (for example, the Internet) to each client's device. This can be achieved from the broadcaster to each individual client, or to particular collections of clients. To accomplish personalization, the service may send a different stream of URIs or even a different stream of Enhanced Content to each client's device. The stream of Enhanced Content (or URIs identifying locations for such content) sent would depend on a client profile stored in the database or at the client device 16. As is well known in the art, a client profile may be built on demand or over time for each client based on criteria such as the location of the client, choices the client makes while receiving a Temporal Signal and/or Enhanced Content, or choices the broadcaster 66 makes during a broadcast 86, or automatic choices made by an algorithm (such as a filter) residing on the service 62. Personalization enables each client to receive Enhanced Content and/or URIs which are uniquely relevant to their interests, demographics, history, or behavior in the system.

Once the URIs have reached the client device 16, system operation is similar for all of the embodiments diagramed in Figures 1, 2, and 4. In the preferred embodiment, a Browser 98 is installed on the client device 16. The Browser 98 allows the client device 16 to retrieve the Web pages 102 or other Enhanced Content, since it is platform independent, and thus, enables efficient and flexible transfer of programs, images, etc., over the Internet or other networks to the client device. Therefore, it is to be appreciated that the process works in and/or with systems that push URIs to a client device and those that do not (i.e., a system wherein the first operation requires the client device to establish a communications link with a location providing Virtual Stage creation elements). In short, the system is capable of presenting Enhanced Content on a client device regardless of the particular systems or methods utilized to create a synchronization between the Enhanced Content and the Temporal Signal. As such, the system is independent of the communications medium, the format, the source, or any other element of the Enhanced Content and/or the Temporal Signal. However, in the preferred embodiment, the Browser is the preferred platform upon which a Virtual Stage is created, as is explained in greater detail below.

Further, in those embodiments in which a URI is embedded in a VBI or other segment of the Temporal Signal, specialized software is provided in addition to the

Browser which enables the client device 16 to extract the URIs from the Temporal Signal. The specialized software acts as an interface between the video programming and the various Internet functions provided by the system. The specialized software retrieves URIs from the video program (embodiment of Figure 1) or directly from the Internet connection (embodiments of Figures 2 and 4), interprets these URIs and directs the Browser 98 to retrieve the particular relevant Web pages 102. In certain embodiments, wherein multiple URIs are sent in conjunction with the Temporal Signal, the specialized software also synchronizes the retrieved Web pages to the Temporal Signal for presentation to the client via the client's device 16, as shown in Figures 3 and 4 and explained in more detail below.

Further, one embodiment of the specialized software also has the capability to detect identical URIs sent directly after one another which causes the Browser not to fetch URIs in these particular cases. As shown in Figure 3, once the URI code is received at the client device 16 (operation 38), the specialized software interprets the URI and determines whether the particular URI has been received previously (operation 42). If it has already been received, the next received URI is interpreted for determination of prior receipt. If the particular URI has not been previously detected, the software checks for misspellings and any other errors. If errors exist, the specialized software corrects the particular errors (operation 46). Once again, it is determined whether the URI has been previously detected. If it has, the next URI is accessed (operation 38). If the URI has not been detected, the specific URI is added to the URI list (operation 54). The specific URI is then sent to the Browser 98, which is preferably a JAVA enabled Browser. Upon receipt of the URI, the Browser 98 accesses the Web site address 122 (Figure 4) indicated by the URI and retrieves the cited Web page 102 via the Internet or other network connection (operation 58). At this point, i.e., when the Browser establishes a connection with a Site providing Enhanced Content for a Virtual Stage, the present invention suitably establishes a Virtual Stage on the client device 16 utilizing the process discussed herein below.

Regardless of whether the URIs, addressing a Web page providing the Enhanced Content related to a Temporal Signal, are accessed after extraction from a VBI or otherwise obtained, for the preferred embodiment, the process suitably entails establishing a Virtual Stage on the client device's Browser before any Enhanced Content is received and presented to the client. As discussed previously, the Virtual Stage provides a platform (a Stage) upon which Enhanced Content may be presented regardless of the source of the Enhanced Content, the communications link utilized,

the format of the Enhanced Content, the type of Temporal Signal, and the type of client device.

While the Virtual Stage is discussed in the context of being created on, and/or via a Browser, it is to be appreciated that such Virtual Stage may also be created on other platforms or applications (either generic or specialized) as desired. However, a Browser is the preferred platform upon which the Virtual Stage is built, because of the widespread availability of Browsers and the non-device and/or signal specific characteristics of such Browsers.

The process by which a Virtual Stage is established on the client's Browser is illustrated in Figure 8. As shown, the process begins upon the reception by the client device 16 of a URI, or similar address, identifying a provider of Enhanced Content related to a Temporal Signal. Preferably, the extraction of the URI, as necessary, from a VBI or other segment of a Temporal Signal has already occurred and has been provided to the Browser for establishing a connection with the Site addressed by the URI (operation 802). The Browser then suitably requests a network to establish a communications link with an online information provider site identified by the URI or other address (hereinafter, the "Site"). Upon establishing a link with the Site, the Browser (either automatically or in response to a query issued by the Site) communicates a userid and an indication of either the URI extracted from the site, the Temporal Signal to which the URI relates, and/or any other incidence of Enhanced Content of which the client device desires to receive (hereafter, a "Skin") (Operation 804). As such, the Skin preferably communicates sufficient information which enables the Site to determine: (1) which Enhanced Content the client desires to receive, and (2) the identity of the client device (and/or the client).

It is to be appreciated, that the Browser may be configured to send the Skin when initially requesting the establishment of a communications link with the Site or upon reception of a query from the Site. Further, a userid is preferably communicated by the Browser to the Site thereby providing an identification of a client to a provider of Enhanced Content. Such identification is desirable, for example, in an educational environment to identify individual students and teachers (as discussed in greater below), to facilitate user profiling and the targeting of content to such user profiles, and for any purposes which utilize, rely or depend upon a userid. Additionally, while the Skin preferably contains two elements, additional elements may also be provided, for example, an indication of the network or other communications system utilized to

reach the Site, the type of client device being used by the client, the location of the client device, and/or the processing capabilities of the client device.

At this point, a communications link is established between the client device and the Site (Operation 806). The communications link may be configured from any single or combination of system, links, devices, paths, or mediums, including, for example, the Internet, intranets, extranet, Ethernet, dial-up network, regional network, wireless networks, cable links, satellite links, wireless links, broadcast networks, hubs, routers, transmitters, and receivers. As discussed previously, the Virtual Stage is device, system, communications link, and content non-specific and preferably may be utilized on any and all devices.

Once the communications link is established between the Site and the client device, the process of creating the Virtual Stage continues with the Site sending a java script program (for example, under a filename such as "stage.jsp") to the client device. The java script preferably contains those elements necessary to initialize the Virtual Stage and a Stage Manager (Operation 808), which "oversees" the creation and use of the Virtual Stage.

More specifically, upon receipt of the "stage.jsp" (hereinafter, the "Routine"), the client device automatically initializes the Routine. However, as desired, the Routine may also be configured to initialize at a later time and/or upon receipt of a command from the client. Once initiated, the Routine effectively detects the production (i.e., the presentation of the Enhanced Content related to a Temporal Signal) specific platform utilized by the client device. For example, in a first presentation, a Flash platform may be utilized and, in a second presentation, a textual platform may be utilized by the same client device. The Routine also detects those production specific settings (for example, the identification of the Show) and requests additional pages (which may include additional program codes) from the server based upon the detected information. Basically, the Routine determines which implementation (i.e., which API) of the Show Object, Receiver Object, Subscriber Object, and any other Object (for example, a chat filter, and the "splash" screen presented on the client device while the Virtual Stage is configured) are needed to present the Enhanced Content. The Routine then requests and receives those APIs and other Objects and elements needed from the server (if it does not already possess them).

The Show Object, Receiver Object, and Subscriber Object APIs may communicate with each other via structural function calls, object messages, an

abstract event model, or other invocation pattern, all of which are equivalent for purposes of the present invention. Further, the Show, Receiver Object and Subscriber Object received from the Site are, in their most basic form, expectations on the availability of certain functions and behaviors. For each specified function/behavior, the Virtual Stage actually implements at least one on the client device. Thus, when the Virtual Stage is configured, at least one Show Object implementation and at least one Receiver Object implementation (in addition to other implementations) is provided. Further, each implementation of an Object (for example, a Show Objector a Receiver Object) is independent of the implementation of any other Object. Thus, a mix and match relationship is created by the Virtual Stage based upon the platform (i.e., is it an Internet Explorer Browser or a Netscape Browser), the client device (i.e., a personal computer versus a PDA), and the Enhanced Content presentation. Such mixing and matching is limited only by those implementations viable on a given platform, client device, and configuration of the presentation, as commonly determined by the producer of such presentation. Such a configuration is analogous to a video signal coming into a home via a cable box, routed through a VCR and then to a television. The video signal is the same regardless of the cable box receiving it. The cable box decodes the signal regardless of what, if any, VCR is connected to it, and the VCR will record and play back the signal regardless of whether a television is attached and/or turned on. Each element performs its respective job regardless of the precise configuration of the other element. However, the signal passing from one device to another ultimately must be compatible with both devices, and thus some non-discriminatory dependencies exist between the devices, provided they fall within a given set of parameters.

For the Virtual Stage, the Temporal Signal embeds the URI regardless of where they point (i.e., which data file is addressed by the URI) or how they are decoded. The Receiver Object (whether in Java, javascript, ActiveX, or another format) decodes the URI and calls the API of the Show Object associated with the type of Enhanced Content to be presented. Whichever Show Object is instantiated renders the URIs, including the chat, rooms, URIs, and actions, as appropriate for the implementation. As such, a similar set of dependencies are created between the various Objects (for example, the Show Object and the Receiver Object), while similar levels of non-discrimination are created between the Objects.

Further, the "stage.jsp" program elements and the Stage Manager are saved in the client device or elsewhere in an appropriate storage location, for example, RAM,

EPROM, a hard disc drive, a memory stick, or a memory card (Operation 810). At this point, or at a later point (as desired by the specific implementation of the Virtual Stage), the client device directs the Stage Manager to initiate the creation of the Virtual Stage (Operation 812).

As mentioned previously, the Virtual Stage includes at least a Show Object and a Receiver Object. However, before those Objects can be created a "floor" and an organizational framework for the Virtual Stage is created. Thus, at this point of the initialization routine, the Stage Manager begins to create those elements necessary to configure the client device to operate as a Virtual Stage. As is appreciated by those skilled in the art, such elements are generally stored in memory locations as data variables and abstractions of objects commonly found in a stage. For example, a brick and mortar stage commonly contains at least one stage or presentation area on which the theatrical show is actually presented to the viewing audience and also includes hidden areas where the technical assistance necessary to "put-on" the show is located. Such hidden areas often include stage hands who, for example, call actors for appearances on the stage, and otherwise control the lighting, audio, props, the interface with the audience (when available), and various other aspects of the presentation. Directing the theatrical presentation is often a stage manager who oversees all of the functions of both the presentation itself and the stage hands.

For the Virtual Stage, similar elements are created, including a Show Object (i.e., the publicly viewable presentation platform) and a Control Area (i.e., the hidden areas from which the stage hands control the presentation). More specifically, the Control Area contains and controls those necessary interfaces between the Show Object and the external environment (i.e., everywhere except for the Show Object area presented on the client device).

However, before the Virtual Stage can be created, the Stage Manager requires two Objects to first exist, a Queue and a Register. Both of these are preferably initially created by the Stage Manager (Operation 814). The Queue provides a listing of tasks to be completed by the Stage Manager in order to configure the client device for the presentation of the Enhanced Content on the Virtual Stage. The Queue is suitably utilized both during the configuration of the Virtual Stage and during the presentation of the Enhanced Content, as necessary.

Often tasks on a queue (as used in the computer sense) are completed in the order in which they are entered into the queue. However, the Queue is preferably configured to emulate and utilize well known threading techniques from other

programming languages including "C" and Java. Examples of such threading techniques include, but are not limited to, wait and notify blocking, mutex locks, semaphores, and synchronization. As such, the Queue preferably contains two fields: a pre-condition field (true or false), and a firing field (i.e., the particular action to be initiated when the pre-condition is true).

For the Virtual Stage, the Queue preferably utilizes at least one threading technique to determine when to cycle through its listing and trigger applicable events, tasks, and/or Objects. As is commonly appreciated, the wait and notify technique is more efficient than using a polling technique, as such the Virtual Stage preferably utilizes a wait and notify scheme. However, utilizing this and similar techniques creates a dependency for the Queue upon other Objects (for example, the Register) to notify the Queue of state changes and, thus, for specific applications, other threading techniques may be utilized.

The other element the Stage Manager initially creates is the Register. The Register is basically a table identifying the existence of an element and how it can be reached. In the brick and mortar stage, a register might include a listing of stage locations (which may be hidden or in view) and how to get to such locations (for example, a stage location situated on a raised lighting assembly might include directions on how to find the assembly). In the Virtual Stage, the Register preferably contains two columns, an Object column and a Key column. The Object column identifies a particular Object and the Key column identifies where the particular Object is located (for example, a RAM pointer or java script pointer which identifies where the Object can be found in a memory or storage device). The Register enables the various Objects of the Virtual Stage to create dependencies to each other, such that when a first Object is referenced by a second Object, they may suitably locate each other.

Additionally, the Queue and the Register preferably work in conjunction to initialize the Virtual Stage. For example, when an Object is added to the Register, the Register suitably "announces" the Object and thereby signals the Queue to survey those tasks on its listing and perform any task which was waiting, for example, upon the "announcement" of the given Object.

At this point in the Routine, the Register preferably contains only one row containing a reference to the Stage Manager itself and where the Stage Manager is located.

Once the Queue and the Register have been created, the initialization process, which is suitably customized on a per-production basis continues with the Stage Manager calling a push method from the initialization routine stored in the memory location identified in the Register as the location of the Stage Manager. Or, in other words, the Stage Manager instructs itself to perform a specific task (a push method) which is co-located with itself. The push routine directs the Stage Manager to create a framework for the stage and the hidden Control Area off the stage. Additionally, the push routine loads into the Queue an instantiation (a time specific representation) of a Show, a Receiver Object and, optionally, a Subscriber. As other elements are needed for a specific Virtual Stage implementation, such elements may also be pushed into the Queue at this time or a later time.

In the Virtual Stage, the Show Object provides instructions to the client device on how to utilize its presentation space (i.e., the stage platform) to present the Enhanced Content. To provide such instructions, as mentioned previously, the Show Object is preferably configured as an API which knows how to present to the client, via the Browser, pushes, chats, and rooms. However, before such elements may be presented on the Show Object to the client, the Stage Manager must first subdivide the stage platform into an appropriate number and configuration of sectors, or frames, as needed (if any).

In a brick and mortar stage, for example, risers are situated in various portions and locations, such as "stage left" or "stage right" to divide the main stage into segments. Additionally, the brick and mortar stage might also contain initially only one frame (i.e., a stage with no risers, props and/or furniture), upon which additional elements are added or subtracted as needed. Additionally, it is appreciated in such an environment, that sub-frames are dependent upon the main frames upon which they are built. For example, a riser can not be extended until the main stage exists.

Similarly, for the Virtual Stage, a sub-frame can not exist until the frames upon which it depends exist. As such, a hierarchical relationship exists between the various frames and sub-frames which may exist for a Show. Often such a relationship is a cascading relationship. For example, suppose a frame definition for Virtual Stage existed as follows:

element "A" is identified as "A.html" and is situated in the top (the root) of the tree and is always available;

element "B" is identified as "B.html" and located in section "X" of element "A";

element "C" is identified as "C.html" and located in section "Y" of element "A"; and

element "D" is identified as "D.html" and located in section "Z" of element "C". When a push targeted at section "W" of element "D" arrives, the following cascade of pushes occurs:

"D" is not available yet, so "D.html" is pushed into "Z" of element "C";

since "C" is not yet available, "C.html" is pushed in to "Y" of element "A";

since "A" is not yet available, "A.html" is pushed into top.

The result being that none of the pushes occur because none of their respective pre-conditions have been satisfied. However, when "A.html" completes, it announces itself to the Queue, firing the push of "C.html". When "C.html" completes, it announces itself to the Queue firing the push of "D.html". When "D.html" completes, it announces itself to the Queue firing the push of "W" into "D" - the original request, which is then completed. All of these actions commonly occur in a client imperceptible moment. Thus, dependencies and cross-dependencies may exist between the various Objects which constitute the Virtual Stage.

Additionally, utilizing this configuration, the system correctly handles competition for the same resources (or stage locations) by utilizing a last in time, last in right system, wherein the last Object to announce itself, unannounces any other Object competing for the same resources. In this manner, the Virtual Stage does not have to concern itself with deleting unneeded Objects or elements as their existence is unannounced once a competing Object or element acquires a given resource. However, in alternative embodiments, an "unannounced" system or approach may also be utilized.

The Virtual Stage also enables the framing to be defined initially and then may vary as the presentation proceeds. To accomplish such framing, the client device preferably divides its presentation space (e.g., the video screen's area) into a given number of frames and sub-frames. When an HTML platform is to be utilized, the client device may, for example, create three frames, one frame in which chat messages occur, a frame in which advertisements are presented, and a frame in which on-line content is presented. Similarly, when a FLASH platform is to be utilized, the client device may create only one frame on which the FLASH segments are suitably overlaid. The number of frames may increase or decrease. For example, a client may decide that they do not wish to receive chat content. In such an event, the Stage

Manager may suitably reconfigure the frames, or may leave the chat frame ("room") open for the client to later re-enter.

For the preferred embodiment, a single frame stage is preferably utilized and upon which FLASH presentations are suitably overlaid, as discussed in the FLASH application. An instantiation of such a Show Object on a single main frame stage might include a push as follows: "stage/show/flash/ie.main". This push directs the client device to establish on an Internet Explorer (IE) configured Browser a stage which includes a main frame on which a Show Object composed of Flash elements is to be presented. Other instantiations of a Show Object frame push might include, for example, "stage/show/html/ie.main", in which Hyper Text Markup Language (HTML) based Enhanced Content is presented on the main portion an IE Browser.

Similarly, instantiations of a Show Object may be utilized for other Enhanced Content formats including: FLASH content ("stage/show/flash/ie.main"), XML content ("stage/show/xml/ie.main"), DHTML, Java and other formats, and/or for other types of Browsers (such as a Netscape® Browser or AOL® Browser) or client devices. It is to be appreciated that as the presentation area of a client device is dissected, additional instantiations of a Show Object sub-element may also be provided to the Queue.

Regardless of the number of frames and sub-frames created on the Virtual Stage by the Stage Manager at initialization or later in the presentation, once a frame is created, each frame ultimately is matched with a Show Object element, an instantiation of which is suitably listed in the Queue. The Show Object element is that element which identifies to a Receiver Object where it is located and how Enhanced Content is to be presented. A Show Object element may be configured for processing content, chats, and/or other types of Enhanced Content.

Further, as mentioned previously, after the announcement of an Object, the Queue suitably cycles through its listing searching for any Objects (actions, or other elements) to trigger, which depend upon the given announcement. For example, a Show Object action may require a sub-frame off a main frame to be provided on the Register and announced before the Show Object element on the Queue is triggered. That is, the sub-frame can not exist on the Register until the main frame exists. Likewise, the Show Object can not exist until both the main frame and the sub-frame exist on the Register. Thus, as the Queue completes its various configurations of Objects, such Objects are suitably identified in the Register.

Once a first Object is identified in the Register, any other Object requesting or depending upon the first Object may obtain a reference location from the Register (until the Object is unannounced). In short, the Register provides pointers to where other Objects are located such that mutual dependencies between Objects may be created, as dictated by the Stage Manager. Since mutual dependencies are created, as necessary, between various Objects utilized by the Virtual Stage, a single Object does not depend upon how a specific Object is implemented and instead merely depends upon the Object's functionality (for example, retrieving Enhanced Content from a site regardless of how accomplished).

For a brick and mortar stage, similar dependencies often exist. For example, an actor in a play depends upon a lighting technician (commonly located in a back-stage, hidden area) to control the lighting of the stage to enhance the mood and environment in which the actor is trying to present. The actor generally is not concerned with how such lighting is generated and instead merely depends upon the lighting to accomplish the scene. However, when the function (the lighting) ceases to operate, for whatever reason, the actor can no longer perform the scene. Thus, dependencies exist between the actor and the stage hand that are based upon the functions performed by each and not on the specific details of how such functions are provided or implemented.

In addition to pushing the Show Object (i.e., how the presentation area of the client device is to be utilized), the Stage Manager also pushes a Receiver Object and (optionally) a Subscriber Object to the Queue. The Receiver Object and the Subscriber Object are similar to the lighting technician (stage hand) of the brick and mortar stage in that they provide hidden functions which are often crucial to the success of the presentation by the Show. Both the Receiver Object and the Subscriber Object are provided for in a hidden Control frame (i.e., a "backstage" area of the Virtual Stage that is not presented to the client) (Operation 818). As with the Show, which requires a frame to be constructed before the Show Object itself could be constructed, the Receiver Object and/or the Subscriber Object also require a Control Area to be Registered before they can be configured and instantiated by the Stage Manager into abstractions identified by the Register. Further, the Control area itself is preferably identified as a hidden frame on the Browser that is further broken into two hidden sub-frames, a receiver and a subscriber. The process of creating the Control Area mirrors that of creating the main frame and other sub-frames. However, unlike the main frame and its sub-frames, the Control Area is preferably never modified or

deleted during a presentation as its existence is crucial to the success of the show, even though it is hidden. Deleting the Control Area is comparable to pulling the power on a brick and mortar theatrical presentation, in that the presentation (i.e., the show) would immediately grind to a halt.

Similarly, just as a brick and mortar show may be restarted after a power or other interruption, the Virtual Stage may also be restarted or continued (when non-volatile memory is utilized to contain the various Objects utilized by the Virtual Stage). More specifically, since the Stage Manager itself is identified in the Register, the Show Object may be restarted by the Stage Manager re-configuring and re-instantiating the Control Area, the Receiver Object, the Show Object, and the Subscriber, and other elements, as necessary, in the Queue and the Register.

Once the Control Area is appropriately configured, the process of creating a Virtual Stage preferably continues with creating the Receiver Object. The Receiver Object is preferably the Object which creates and maintains a down channel with the Site (which may or may not be a Virtual Site). More specifically, the Receiver Object identifies itself as being that Object which will handle pushes, chats, rooms, and subscription announcements sent by the Site to the client device. Pushes, chats and rooms are herein described in abstract terms. The Receiver Object is preferably configurable such that it may receive pushes, chat, etc. regardless of their format (i.e., TCP/IP, UDP, etc.). For example, "hypervbi.html" might be one instantiation of a Receiver Object which knows how to receive Enhanced Content provided in a VBI. Similarly, "hyperweb.html" might be an instantiation of a Receiver Object which knows how to receive Enhanced Content provided over a Web connection. Based upon the configuration of the client device and/or the configuration of the Site (or the Virtual Site), the Stage Manager appropriately configures a Receiver Object that is compatible with the Site. As mentioned previously, the configuration of the Receiver Object, the Subscriber, and other elements is determined at the time of initializing the Routine.

The Receiver Object also identifies itself as being the Object which will call those routines provided in the Stage Manager and the Enhanced Content provided by the server to configure the client device to receive pushes. These pushes may be provided via various communications links and in various formats including, for example, over an Internet connection, a television broadcast connection, or a wireless connection. In this regard, the Receiver Object mirrors the Show Object in that the Receiver Object knows how to receive the pushes which are then rendered and

presented to the client by the know-how of the Show. Since the Receiver Object and the Show Object are abstractions and APIs (as discussed previously), it is to be appreciated that they may be implemented on any device, utilizing any configuration. Thus, it is to be appreciated that the configuration of the Receiver Object vis-à-vis the Site, may occur at the Receiver Object, the Site and/or somewhere in-between (for example, a network hub).

The Subscriber Object is the other Object utilized to control the basic operation of the Virtual Stage. The Subscriber Object basically is an abstraction of a device which knows how to send chats and subscribes from the client device to the Site. As provided for the Receiver Object and the Show, the Subscriber Object is also identified in the Register to enable the Show Object to know where the element capable of sending subscribes and chats is located. The Subscriber Object also accesses those elements provided in the Stage Manager and/or the client device to make possible such chats and subscribes.

Up to this point, the Stage Manager has merely told the client device how to present information received from the Receiver Object, how to establish a connection with the provider of the Enhanced Content to be presented to the client via the Show, and how to subscribe to chat rooms and send, receive and present chat messages. However, while the preferred method includes an abstraction of only a Show, a Receiver Object and a Subscriber, it is to be appreciated that various other abstractions of Objects may be specified by the Routine and implemented by the Stage Manager. For example, an alternative embodiment of a Stage Manager may abstract two Receiver Objects, one for receiving Internet based content and one for receiving wireless content, both of which might be suitably presented on a single Show. Regardless of the number and/or type of abstracted Objects utilized, the present invention is preferably configured with, as a minimum, a Show Object and a Receiver Object. Further, the foregoing actions occur in today's processors so fast, that a client device is generally configured and ready to receive Enhanced Content without incurring any human perceptible delay.

Once the Show Object and the Receiver Object have been configured and identified in the Register (i.e., announced to the Stage Manager and the Queue), the process continues with the Stage Manager communicating a Stage (which includes an identification of the Receiver Object and the Show) to the Site (Operation 820). The Stage contains an identification that the Virtual Stage that is ready to receive the Enhanced Content. In the brick and mortar example, the communication of the

existence of the Virtual Stage corresponds to a curtain call, wherein the actors and others (i.e., the Site) are informed that it is show time. Shortly thereafter, the various elements of the show (i.e., the Enhanced Content) are then presented to the viewing audience (i.e., the client). Importantly, the Stage and the Receiver Object are basically oblivious to the source of the Enhanced Content as they are mere abstractions, which are capable of receiving the Enhanced Content.

As such, the preferred embodiment of the process enables one to provide code for configuring the client device at either the server Site or the client device. The present invention facilitates the removal, to the server, of many of the device configuration actions thereby reducing the amount of memory, downloads, and routines needed to be used and/or executed by the client device to configure a Virtual Stage. However, it is to be appreciated that placing the device configuration codes on the Site results in less scalability, whereas placing the device configuration codes on the client device increases memory utilization. Thus, a trade-off exists between the higher scalability and the amount of memory needed on a client device to provide a given level of scalability.

Once the Stage is announced to the Site, the Site may then commence sending the desired Enhanced Content. However, often many Sites require a client to subscribe to the Site. The Subscriber object handles requests for subscriptions, which are suitably received by the Receiver Object, passed to the Show Object for presentation to the client, and then to the Subscriber Object (which may automatically generate or be manually configured to generate the subscription) and then send the subscription to the Site (Steps 822 and 824).

At this point, the client device is configured and ready to receive Enhanced Content that may or may not relate to a specific Temporal Signal. The Receiver Object receives the Enhanced Content pushes and provides such pushes to the Show Object for presentation to the client via, for example, a Browser. These pushes may come in any format including, but not limited to, URIs, content pushes, frame set pushes (which reconfigure the framework for the presentation area of the Browser), functional pushes (which often include both commands and data), and RAID pushes (as discussed in the above referenced FLASH application). Since these various types of pushes provide varying levels of commands and content, the Stage Manager is preferably configured to process functional pushes and RAID pushes, while the Show Object handles frame pushes and content pushes. Further, such pushes are carefully and precisely determined when pertaining to the hidden control area, the Receiver

Object and/or the Subscribe Object, as a mis-configuration of such Objects/pushes may wreck havoc with the client devices' ability to receive and process Enhanced Content.

Thus, the Virtual Stage is preferably implemented on a Browser and is not implemented as a stand-alone application. However, the Virtual Stage may be implemented as a stand-alone application, as desired, with appropriate trade-offs in the necessary java script or Java applet or similar files to be downloaded and stored on the client device. In another embodiment, the Virtual Stage suitably reduces the need for large downloads and time consuming initialization routines and thereby enables the Enhanced Content to be quickly presented to a client via virtually any compatible device by providing such capabilities at the Site (or the Virtual Site).

Further, since the client device utilizes an abstraction of a Receiver Object in order to establish a communications link with a Site and receive Enhanced Content, it is to be appreciated that such abstractions may also occur at the Site itself. For example, a Site may be abstracted such that a single playlist for an Enhanced Content presentation may be generated, which is then abstracted by the Site, as necessary, into the appropriate data formats requested by the Receiver Object. Thus, a producer of Enhanced Content is capable of producing a single rendering of a playlist which may be communicated to various client devices over various implementations of a Site including, for example, an Internet site, a wireless site, an intranet site, and a broadcast site.

Additionally, it is anticipated, for example, that a television game show program presenting Enhanced Content to a multitude of clients may frequently encounter sudden bursts of responses (subscribes or chats) from multitudes of clients attempting to answer a game show question. Current server systems are extremely inefficient and often incapable of processing such sudden and dramatic subscribes because they often are configured to process a transaction completely before returning to a given process and then format the return values with the result. In contrast, a Site implementing the present invention preferably utilizes a Capture module, which is preferably located at the Site (or an independent location designated for receiving client subscribes). The Capture module provides an upstream data capture module which does not provide an immediate return value. Instead, the Capture module returns to the client before a transaction is completed. The Capture module schedules each client's subscribe/response for processing. Thus, the Capture module enables a Client to send up various pieces of information without delay. Further, the upstream

interaction occurs on a queue hidden frame such that multiple upstream requests can be pending at any time and from multiple clients. This configuration allows the Show Object to continue processing Enhanced Content segments even if the upstream channel is blocked or unavailable for an extended period of time.

An exemplary implementation of the Virtual Stage can best be understood with reference to an example. A client can begin watching a musical video featuring a new band. As the video is received by the client device 16, URIs are either being received with the video signal or are being received directly via the Internet 20 or another data channel, and are being interpreted by the specialized software 106. Upon the first instance of receiving an indication that Enhanced Content is available, the Browser 98 retrieves a particular Web page 102 from an Internet 20 Web site that is identified manually or automatically. Upon establishing a connection with the Web site, the Browser receives the "stage.jsp" program and begins installing the Virtual Stage.

Once the Virtual Stage has been established and its existence notified to a Site providing the Enhanced Content, the client device 16 may then begin receiving Enhanced Content, regardless of the source of such content. In a frame based example, such Enhanced Content is provided via HTML based Web pages which are suitably displayed on the video screen in a pre-determined frame format, at particular times. In the preferred Flash implementation, the Enhanced Content is presented as overlays to a master movie. Thus, for example, while the viewer is watching the music video, biographical information on the band can also be displayed adjacent to the video window, or in Flash, as an overlay on a full frame or partial frame window. The Enhanced Content could also include an upcoming concert schedule, or even audio clips of the band's music may be downloaded from the Internet.

As another example, a client could be watching a program relating to financial news. While the narrator is shown discussing high tech stocks, Enhanced Content corresponding to detailed financial performance information on high tech stocks, environment and characteristics can be displayed with the video on the Virtual Stage. Such Enhanced Content can be presented regardless of the source of the information or the type of device utilized by the client. However, preferably a Browser or application program is utilized on the client device to control the presentation space. If the personalization features are included, Enhanced Content associated with a particular client's stock can be fetched and displayed on the Virtual Stage with the Temporal Signal. When the narrator in the Temporal Signal switches to a discussion

on the weekly performance of the Dow Jones, Enhanced Content presenting related financial performance information can be simultaneously displayed. Thus, it is evident that the present invention profoundly enriches the viewing and/or learning experience and the ease of providing and configuring such an experience.

It is understood that there can exist alternative embodiments for use with the present invention. For example, the client can view the Temporal Signal using a television set or other display monitor in conjunction with the display screen of a client device (for example, a personal computer) presenting the Virtual Stage. In this embodiment, the relevant Enhanced Content is shown on the client device while the Temporal Signal is displayed on the television monitor. In this alternative embodiment, a cable set top box receives the Temporal Signal, for example, a television program, from the multi-channel cable. The client device also receives the Temporal Signal from the multi-channel cable and extracts the URIs, embedded in the VBI of the video signal or directly transmitted over the Internet. The specialized software, as needed, extracts the URIs and retrieves the particular Enhanced Content as described above. The Virtual Stage is then created on the client device and the Enhanced Content related to the Temporal Signal is presented to the client. It is understood that a hyperlink may exist on the Site that will allow the client device to automatically load the specialized software needed to extract URIs from a Temporal Signal, the Virtual Stage and call up the specific television channel referenced in a Site and providing Enhanced Content related to a Temporal Signal.

For example, someone browsing the Internet may come upon a major television network's Web site. They scroll to an interesting story then click on a hyperlink, which provides the software necessary to create a Virtual Stage on their device (if one has already not been created). The Virtual Stage suitably includes a television frame in which the Temporal Signal is presented and another frame (or an Overlay in the Flash environment) in which the Enhanced Content related to the Temporal Signal (or even to a second, non-displayed, Temporal Signal) is displayed. For example, a signal providing a first baseball game's box score may suitably overlay a signal providing the video portion of a second baseball game in the television frame.

Furthermore, instead of receiving Temporal Signals from a transmission means, the Temporal Signals can also be addressed directly from the client site if the Temporal Signal, with or without embedded URIs, is stored on a VHS, Beta, DVD, CD, memory stick or other storage medium. In this embodiment, the client device

and/or television are connected to a VCR, DVD, CD player or other appropriate device.

Figures 5 and 6 show two alternative embodiments for use with the present invention. For example, the client can view the interactive program using a television set 18 or other display monitor in conjunction with a digital cable box 140, as shown in Figure 5. In this embodiment, the digital cable box 140 performs the functions of the client device 16 shown in Figures 1, 2 and 4. In the embodiment shown in Figure 5, the specialized software is stored in memory in the digital cable box 140. In one embodiment, the digital cable box 140 includes two tuners, thus allowing both the Enhanced Content, preferably using the Browser, and the Temporal Signal (for example, a video program) to be simultaneously viewed on the same screen. The screen may be configured as one Virtual Stage, or even split in two, wherein a first portion of the screen is configured as a television monitor and a second portion of the screen is configured as the Virtual Stage. If the Temporal Signal and Enhanced Content, however, are carried on one channel, then only one tuner may be necessary.

The specialized software retrieves URIs from the received video program, directly from the Internet connection 20 or via a separate data channel, interprets these URIs and directs the Browser to access the particular relevant Site at which the java script code script elements needed to establish the Virtual Stage are provided. Once the Virtual Stage is established, the digital cable box receives the Enhanced Content that is synchronized to the Temporal Signal for presentation to the client via the television 18, as shown in Figure 5. In this embodiment, the relevant Enhanced Content is preferably shown in one frame of the television 18 while the video elements of the Temporal Signal are displayed in another frame. Alternatively, the Enhanced Content can replace the Temporal Signal on the display, as desired.

Further, in this alternative embodiment, the digital cable set top box 140 receives the television program from the multi-channel cable. The URIs identifying the location at which the Virtual Stage and/or the Enhanced Content are located is preferably encoded into the digital program channel using MPEG 1, MPEG 2, MPEG 4, MPEG 7 or any other video compression scheme. Alternatively, the URIs can be transmitted to the digital cable boxes 140 from an Internet server 148. The digital cable box 140 decodes the URIs from the digital video signal or directly transmitted over the Internet 20. The specialized software decodes the URIs and retrieves the particular Virtual Stage information and/or Enhanced Content as described above.

The Enhanced Content is then preferably synchronized with the particular Temporal Signal and presented to the client on the Virtual Stage.

As with all the embodiments described above, instead of receiving the Temporal Signal from a transmission means, the Temporal Signal can also be obtained directly from a local source 144 if the Temporal Signal, with or without embedded URIs, is stored on a VHS, Beta, DVD, CD, or memory stick, or other recording/storage medium. In this embodiment, the digital cable box 140 is connected to a device capable of communicating the Temporal Signal from a recording/storage medium to a client device, in, for example, the digital cable box 140. System operation then continues, as above, with the Browser contacting the Site, establishing the Virtual Stage and receiving the Enhanced Content related to the Temporal Signal.

Figure 6 discloses an embodiment wherein a digital TV 152 is the remote reception unit. In this embodiment, the digital TV 152 performs the functions of the client device, shown in Figures 1, 2 and 4, and the digital cable box 140 shown in Figure 5. In the embodiment shown in Figure 6, a processor and memory are incorporated into the digital TV 152. Further, the specialized software and a Browser are implemented into memory in the digital TV 152. All of the functions described above with reference to the other embodiments are performed in a similar manner by the digital TV 152 embodiment.

Although the digital cable box/TV 140, 18 and digital TV 152, shown in Figures 5 and 6, are incorporated into the embodiment of Figure 1, they also could be substituted for the client device 16 shown in Figures 2 and 4.

For example, when the Temporal Signal includes a video portion, the client can view the video and the Enhanced Content on one screen (in two windows), in whole or in part, via the Virtual Stage. Similarly, the Temporal Signal can be viewed on one display screen and the Enhanced Content on a separate display monitor. Alternatively, the client can access the video or Enhanced Content separately. Thus, the client can branch from video to Enhanced Content and vice versa.

The present invention is well-suited to the education environment. In this embodiment, students and teachers access one or more Web servers. The software components include instructor and student user software, authoring software and database assessment software. In one such embodiment, an instructor uses content creation software on a suitable device, for example, a personal computer, to easily integrate into their curriculum current information published on the Web, through an

easy to use interface, such as one provided on a Virtual Stage (which in this embodiment is recognized as a "Virtual Classroom"). The instructor creates a lesson (i.e., a Show), the lesson comprising a listing of Web pages, text notes and questions. Further, the Web sites and questions are set forth in a predetermined order and can be assigned times. Preferably, the URIs identifying the Web site and time stamps are sent automatically to the desktop of each student assigned to the Virtual Classroom, either during playback of a pre-recorded program or during a live event.

In the educational embodiment, the Virtual Classroom provides a classroom (much like a real classroom) at which a lesson (the Show) may be presented. Preferably the Virtual Classroom is analogous to an actual classroom in that students may view the classroom (i.e., the Virtual Classroom) regardless of their perspective (i.e., their client device), the time of day, and/or their location. Further, via a Virtual Classroom, the students may interact with the instructor (the Stage Manager) of the lesson. For example, via a chat interface, video images of themselves posing a question to the class may be presented to the Virtual Classroom for all participants to view. As discussed previously, the lesson (i.e., the Show) provides an API through which the instructor may push content, provide and/or allow chats, and establish various rooms in which students may be presented with the assignments, experiments, and various educational materials and/or activities. Since the lesson may be pre-recorded and/or live, each student, via their client device, may be part of the Virtual Classroom from a remote location and regardless of the type of device utilized by the student, provided certain minimum requirements are met.

At each of the student devices, the educational program presented on the Virtual Classroom as a lesson is produced/directed by the instructor via pushes from a Site (i.e., a teacher) to the Stage Manager, via the Receiver Object. In other words, the Site (i.e., the teacher) provides the structure for the educational experience. At predetermined times as dictated by the Site, the Browser fetches and displays Enhanced Content segments on the Virtual Classroom. Such segments may be obtained from the Site or an alternative site, as directed by the instructor. Because the lesson can be set up in this manner at predetermined times, the entire lesson can be prerecorded and stored in a Web database for later access by the students. For example, a student having difficulty completing a homework assignment could access the lesson from home and re-learn those elements they missed in class (virtual or real).

A significant advantage of an embodiment for educational applications is that the students and the instructor can be located anywhere, as long as they can be all connected to the lesson via a Virtual Classroom. Similarly, a virtual tutoring environment may also be created. Further, because a teacher is essentially controlling the educational experience, the lesson can take almost any desired form on the Virtual Classroom as presented on the student devices.

Unlike conventional distance learning systems, systems consistent with the present invention are more powerful by allowing the instructor to freely and conveniently exercise almost any type of testing strategy without concern as to the type of devices and/or location of students to the lesson. As such, students connected via wireless devices, computer workstations, or other devices may be suitably connected to the Virtual Classroom. The instructor can test students using a combination of the chat dialogue feature and Enhanced Content. For example, multiple choice questions and short answer questions can appear in the chat section. Essay questions, requiring longer answers, become Enhanced Content frames. As mentioned above, students can perform virtual experiments on-line. Once the instructor's device (for example, a personal computer) receives student answers, student scoring can be presented to the instructor in any format including tables, charts, diagrams, bar graphs, etc. The instructor, thus, can analyze the results and has the capability of providing real-time feedback to the students.

Students can also receive individualized feedback via branched interactive audio, video and/or graphics responses. For example, a student device may branch to a particular audio response contained in a segment of Enhanced Content, preferably prerecorded in the instructor's own voice, based on the student response to a multiple choice question. In this embodiment, a plurality of potential audio responses are made available at the student's device according to any one of the methodologies set forth in, for example, U.S. Patent No. 5,537,141, entitled DISTANCE LEARNING SYSTEM, which is herein incorporated by reference in its entirety. Alternatively, personalized video, audio and graphics segments can be delivered and displayed to the student based on a student answer or personal profile in the manner set forth in U.S. Patent No. 5,724,091, entitled COMPRESSED DIGITAL DATA INTERACTIVE PROGRAM SYSTEM, which is herein incorporated by reference in its entirety.

In another embodiment of the present invention, a system is described that is capable of handling the education requirements of several schools utilizing multiple

Virtual Classrooms in an efficiently designed network. The system shown in Figure 7 solves the problems inherent in attempting to service large numbers of users, the most obvious obstacles being the issues of load, performance, and varying types of student devices. In this embodiment shown in Figure 7, communications servers 180 distribute and route messages across a LAN, WAN and the Internet. Referring to Figure 7, in the center of the diagram is the Group Database server. Surrounding the database server are several Com Servers 180, each serving an area 192. Surrounding each Com Server 180 are squares representing student devices 188, which are suitably configured with a Virtual Classroom permitting various types of devices to communicate to a Com Server 180, regardless of the communications path(s) utilized. The Communication Servers 180 are organized in node relationships with one another.

Each node is responsible for serving an Area 192. An Area 192 is defined as a virtual location serviced by a single Communications Server 180 (or "Com Server"). An Area 192 may be a single school, an office, or may consist of several actual physical or virtual locations. The defining characteristic of an Area 192 is that messages sent from one member of an Area 192 to another need not be routed outside of the servicing Com Server 180. An Area member is analogous to the frequently used term "student" (or "client").

The Distributed Communication System of Figure 7 permits the dynamic addition of Communication Servers 180 within a group with little or no administrative tasks as well as the addition of groups within an overall communications network. A Communication Server group consists of several defined virtual Areas 192 (preferably, consisting of no more the 250 client devices each), each Area 192 serviced by a single Com Server 180. This system enables client devices of one Area 192, or group to easily communicate with client devices in another Area 192 or group without any configuration changes by using Virtual Classrooms on each student device that are system and configuration independent. As such, a student in one Area 192 (for example, the United States and using a CMDA equipped wireless phone) need not be concerned that their messages are compatible with another student in another Area 192 (for example, a student in Europe using a GSM equipped wireless phone) because the Virtual Classroom on each student's device enables the client to receive messages without regard to the format, provided a communications path between the two can be established which provides a translation service from one communications signal format to another.

Further, service of very large numbers of students has required large expensive servers and networks. As the user base increases, performance suffers and hardware must be upgraded to service the demand.

The Distributed Communication System of the present invention allows the same, relatively inexpensive machines to serve an ever-increasing user base because each student device (preferably configured with a Browser) may utilize a Virtual Classroom that is platform and communications path independent. The technique by which this will be accomplished will be through the routing of messages from one server to another when necessary until it reaches a student device with a Browser and a Virtual Classroom.

The method essentially follows the same core pattern as IP routing and DNS lookups. If a message is for a member not belonging to the current Area 192 or group, the message may be routed through the Distributed Communication System until its destination, or someone who knows the destination and can deliver the message, is found. The destination may be cached so subsequent messages for that student or group of students may be more efficiently delivered, and/or until a Virtual Classroom is established on the specific student's device.

Referring again to Figure 7, if a message is posted by student "A" and is intended only for the students of group 1 the message preferably never leaves the Area 1 Com Server. However, if the message is intended for students of Area 1 and Area 2, the Area 1 Com server forwards the message to the group database server 184. The message may be broadcast to the students of Area 1 and tagged in the database 184 as belonging to Area 2. The message may then be routed to Area 2 and broadcast to Area 2 students. With this technique any student can potentially send a message to any other student via the Virtual Classroom or even during a Virtual Chat session (i.e., a session in which a Virtual Stage or Virtual Classroom is primarily configured as a chat room or a similar room for providing video and/or audio communications and interactions between one or more clients). If the Area Com server 180 does not recognize the destination, the message is forwarded up the line. Each Com server 180 does not need to know about any other server 180. Messages are routed until they are delivered. If undeliverable, the original sender is notified. Since each Com server 180 may also be equipped with a Virtual Classroom, the need for Com servers 180 to share a mutual operating platform may be significantly reduced.

New Areas 192 can be added on the fly. When a new Com server 188 is added to the network, it registers itself with the database application, establishes Virtual Classrooms on its associated student devices, and provides an indication of the type of signals and devices it prefers to receive. Henceforth, any message destined for the new Area 192 can be routed properly without altering the other Area Servers 180. This method and system works for global messages or for student to student messages. Furthermore, new Groups may also be dynamically added. Once added, each new Group Database Server 184 registers itself with the existing database servers 184. This distribution of load permits nearly unlimited expansion with existing software and hardware. Each server manages a finite number of students, cumulatively serving a growing community, for example, a large university.

Students need not be informed as to the particular Com Server 180 they should connect to. Students are directed to a single URI at which those routines necessary to establish the Virtual Stage/Classroom/Chat room is provided. The selection of the server for user connection is determined by load balancing software. In this manner, the network may appear to be a global network of Servers or simply a local classroom.

The unique aspects of this architecture, using database servers as routing gateways, using techniques resembling IP routing and DNS lookup, and utilizing Virtual Classrooms, where appropriate, to increase interoperability of systems and devices enables this system to serve with minimum administration and configuration and with lower end, cost-effective hardware.

While the present invention has been described in relation to specific systems, hardware, devices, software, platforms, configurations, process routines, and various preferred embodiments, it is to be appreciated that the present invention is not limited to any specific embodiments, process, systems, devices, signal formats, data formats, and/or configurations. As such, the present invention may be considered to cover any and all subject matter, as specified in the attached claims.

CLAIMS

1. A computer readable medium providing program code segments for creating a Virtual Stage on a client device, wherein the Virtual Stage provides a platform by which at least one segment of Enhanced Content can be presented to a client via a presentation device, comprising:

a first program code segment providing an abstraction of a Receiver Object, wherein the Receiver Object provides at least one instruction which configures the client device to receive at least one segment of Enhanced Content from an Enhanced Content provider; and

a second program code segment providing an abstraction of a Show, wherein the Show Object provides at least one instruction which configures the client device to present the at least one segment of Enhanced Content;

whereupon establishment of a communications link between the client device and the Enhanced Content provider, the Receiver Object and Show Object configured client device receives the at least one segment of Enhanced Content and presents the received Enhanced Content segments via a presentation device.

2. The computer readable medium of claim 1, wherein the Virtual Stage further comprises a Virtual Classroom.

3. The computer readable medium of claim 1, wherein the Virtual Stage further comprises a Virtual Chat session.

4. The computer readable medium of claim 1, wherein the client device comprises at least one of a personal computer, a personal data assistant, a Web tablet, a wireless communications device, a computer workstation, a gaming console, a set-top box, an Internet equipped television, a digital television, a Browser, a cable box, and a device capable of presenting Enhanced Content to a client.

5. The computer readable medium of claim 1, wherein the presentation device further comprises at least of a television, a video display system, an audio system, a virtual reality system, a gaming system, slow motion video presentation system, a still-frame presentation system, a motion picture presentation system, and a home theater system.

6. The computer readable medium of claim 1, wherein at least one of the first program code segment and the second program code segment provides at least one

instruction to configure the client device to receive or present at least one segment of Enhanced Content received in at least one of a hyper-text mark-up language file format, a Flash file format, a dhtml file format, a Java file format, an xml file format, a text file format, a graphic file format, a video file format, and a sound file format.

7. The computer readable medium of claim 1, further comprising:

a third program code segment providing an abstraction of a Subscriber, wherein the Subscriber Object provides at least one instruction which configures the client device to subscribe to at least one subscription.

8. The computer readable medium of claim 7, wherein the at least one instruction configures the client device to subscribe to at least one chat message service.

9. The computer readable medium of claim 1, wherein the computer readable medium is located with at least one of a network server, the client device, the Enhanced Content provider, and a provider of a temporal signal to which the Enhanced Content relates.

10. The computer readable medium of claim 1, further comprising:

a fourth program code segment providing an abstraction of a Control Area, wherein the abstracted Control Area includes at least one Object utilized by the client device to establish interfaces between the client device and at least one Enhanced Content provider.

11. The computer readable medium of claim 1, further comprising:

a fifth program code segment providing a Stage Manager, wherein the Stage Manager controls the creation and operation of the Virtual Stage.

12. The computer readable medium of claim 11, wherein the Stage Manager further comprises at least one instruction for creating a Queue and a Register; wherein the Queue includes a listing of at least one task utilized to configure the client device to present the at least one segment of Enhanced Content and the Register includes an identification of at least one Object and how each of the at least one Object can be contacted.

13. The computer readable medium of claim 12, wherein at least one of the abstracted Show Object and the abstracted Receiver Object are identified as an Object on the Register.

14. The computer readable medium of claim 12, wherein the Register further comprises a table having an Object column and a key column, wherein the Object column identifies a particular Object, and the key column identifies where the particular Object is located.

15. The computer readable medium of claim 14, wherein the Object is located in a memory device that is co-located with at least one of a network server, an Enhanced Content provider, the client device and a provider of a Temporal Signal related to the Enhanced Content.

16. The computer readable medium of claim 12, wherein at least one Object in the Register has a co-dependency with at least one additional Object in the Register.

17. The computer readable medium of claim 16, whereupon adding an Object to the Register, the Register announces the Object to the Queue, and whereupon receiving the announcement, the Queue surveys any listed tasks and directs the Stage Manager to perform those tasks which are awaiting the announcement prior to being executed.

18. The computer readable medium of claim 17, wherein the Stage Manager further comprises an instruction which provides that an Object which is announced last unannounces any other Object competing for a same resource on the client device.

19. A computer readable medium providing program code segments for creating a Virtual Stage on a client device, wherein the Virtual Stage provides a platform by which at least one segment of Enhanced Content can be presented to a client via a presentation device, comprising:

a first program code segment providing an abstraction of a Receiver Object, wherein the Receiver Object provides at least one instruction which configures the client device to receive at least one segment of Enhanced Content from an Enhanced Content provider;

a second program code segment providing an abstraction of a Show, wherein the Show Object provides at least one instruction which configures the client device to present the at least one segment of Enhanced Content; and

a third program code segment providing an abstraction of a Subscriber, wherein the Subscriber Object provides at least one instruction which configures the client device to subscribe to receive the at least one segment of Enhanced Content

from the Enhanced Content provider;

whereupon establishment of a communications link between the client device and the Enhanced Content provider, the Receiver Object and Show Object configured client device receives the at least one segment of Enhanced Content, presents the received Enhanced Content segments in accordance with the configuration instructions specified by the Show, and subscribes to at least one of a subscription and a chat room in accordance with the instructions specified by the Subscriber.

20. The computer readable medium of claim 19, wherein the client device further comprises at least one of a personal computer, a personal data assistant, a Web tablet, a wireless communications device, a computer workstation, a gaming console, a set-top box, an Internet equipped television, a digital television, a Browser, a cable box, and a device capable of presenting Enhanced Content to a client.

21. The computer readable medium of claim 19, wherein the presentation device further comprises at least of a television, a video display system, an audio system, a virtual reality system, a gaming system, slow motion video presentation system, a still-frame presentation system, a motion picture presentation system, and a home theater system.

22. The computer readable medium of claim 19, wherein at least one of the first program code segment and the second program code segment provides at least one instruction to configure the client device to receive or present at least one segment of Enhanced Content received in at least one of a hyper-text mark-up language file format, a Flash file format, a dhtml file format, a Java file format, an xml file format, a text file format, a graphic file format, a video file format, and a sound file format.

23. The computer readable medium of claim 19, wherein the Subscriber Object further provides at least one instruction which configures the client device to subscribe to at least one chat message service.

24. The computer readable medium of claim 19, further comprising:

a fourth program code segment providing an abstraction of a Control Area, wherein the abstracted Control Area includes at least one Object utilized by the client device to establish interfaces between the Show Object and the at least one Enhanced Content provider.

25. The computer readable medium of claim 19, wherein the computer readable medium is located with at least one of a network server, the client device, the Enhanced Content provider, and a provider of a Temporal Signal to which the Enhanced Content relates.

26. A computer readable medium providing program code segments for creating a Virtual Stage on a client device, wherein the Virtual Stage provides a platform by which at least one segment of Enhanced Content can be presented to a client via a presentation device, comprising:

a first program code segment providing an abstraction of a Receiver Object, wherein the Receiver Object provides at least one instruction which configures the client device to receive at least one segment of Enhanced Content from an Enhanced Content provider;

a second program code segment providing an abstraction of a Show, wherein the Show Object provides at least one instruction which configures the client device to present the at least one segment of Enhanced Content;

a third program code segment providing an abstraction of a Subscriber, wherein the Subscriber Object provides at least one instruction which configures the client device to subscribe to at least one subscription service;

a fourth program code segment providing an abstraction of a Control Area, wherein the abstracted Control Area contains those elements utilized by the client device to establish interfaces between the client device and at least one Enhanced Content provider; and

a fifth program code segment providing a Stage Manager, wherein the Stage Manager provides at least one instruction which configures the client device to create and operate the Virtual Stage;

whereupon establishment of a communications link between the client device and the Enhanced Content provider, the Receiver Object, Show Object and Subscriber Object configured client device receives the at least one segment of Enhanced Content, presents the received Enhanced Content segments in accordance with the configuration instructions specified by the Show, and subscribes to at least one of a subscription and a chat room in accordance with the instructions specified by the Subscriber.

27. The computer readable medium of claim 26, wherein the Stage Manager provides at least one instruction which controls an order in which an abstraction is created on the client device.

28. The computer readable medium of claim 27, wherein the Stage Manager further comprises at least one instruction for creating a Queue and a Register; wherein the Queue includes a listing of at least one task utilized to configure the client device to present the at least one segment of Enhanced Content and the Register includes an identification of at least one Object and where the at least one Object is located.

29. The computer readable medium of claim 28, whereupon adding an Object to the Register, the Register announces the Object to the Queue, and whereupon receiving the announcement, the Queue surveys any listed tasks and directs the Stage Manager to perform those tasks which are awaiting the announcement prior to being executed.

30. The computer readable medium of claim 29, wherein the Stage Manager provides at least one instruction which controls an order in which the abstraction of the Show, the Control Area, and the Receiver Object are created on the client device.

31. A system for presenting Enhanced Content related to a Temporal Signal to a client via a client device on a Virtual Stage comprising:

a receiver for receiving a Temporal Signal, wherein the Temporal Signal includes at least one URI embedded into the Temporal Signal, the URI providing an address for a Site providing Enhanced Content related to the Temporal Signal;

a decoder, connected to the receiver, for extracting the URI from the Temporal Signal and outputting the URI;

a client device, connected to the decoder, the client device further comprising:

a Browser; and

a storage device;

whereupon receipt of the URI from the decoder, the Browser establishes a connection with the Site and receives from the site a program code which configures the client device as a Virtual Stage by initializing and saving, in the storage device, cross-dependent abstractions of a Show Object and a Receiver Object, wherein the Receiver Object and the Show Object collectively enable the Browser to receive and present the Enhanced Content from any source and via any communications link utilized to communicate the Enhanced Content to the client device.

32. The system of claim 31, wherein the client device further comprises at least one of a personal computer, a personal data assistant, a Web tablet, a wireless communications device, a computer workstation, a gaming console, a set-top box, an Internet equipped television, a digital television, a Browser, a cable box, and a device capable of presenting Enhanced Content to a client.

33. The system of claim 31, wherein the client device further comprises a presentation device, the presentation device further comprising at least of a television, a video display system, an audio system, a virtual reality system, a gaming system, slow motion video presentation system, a still-frame presentation system, a motion picture presentation system, and a home theater system.

34. The system of claim 31, wherein the client device includes at least one instruction to configure the Browser to present at least one segment of Enhanced Content received in at least one of a hyper-text mark-up language file format, a Flash file format, a dhtml file format, a Java file format, an xml file format, a text file format, a graphic file format, a video file format, and a sound file format.

35. A Virtual Stage provided in at least one of a computer readable medium and a propagated signal, for enabling a client device to receive and present at least one segment of Enhanced Content related to a Temporal Signal irrespective of a communications medium utilized to transmit the at least one segment of Enhanced Content from an Enhanced Content provider to the client device, comprising:

a Show Object providing an abstraction of a presentation area in which the Enhanced Content is presented; and

a Receiver Object providing an abstraction of a device capable of receiving the Enhanced Content and providing the received Enhanced Content to the Show Object for presentation to a client via the client device.

36. A memory for storing instructions utilized to configure a presentation space on a client device as a Virtual Stage upon which at least one segment of Enhanced Content may be presented, comprising:

a first data structure stored in a memory, the first data structure including instructions for controlling a presentation space on a client device; and

a second data structure stored in a memory, the second data structure including instructions for configuring the presentation space as a Virtual Stage, further comprising:

an abstracted Stage Manager data Object, wherein the abstracted Stage Manager controls the creation and operation of the Virtual Stage;

a plurality of abstracted Objects, each of said abstracted Objects being utilized to create the Virtual Stage;

an abstracted Queue, in communication with the abstracted Stage Manager, wherein the abstracted Queue comprises a list of tasks to be accomplished and Objects to be abstracted, under the direction of the Stage Manager, in creating and operating the Virtual Stage;

an abstracted Register, in communication with the abstracted Stage Manager and the abstracted Queue, wherein the abstracted Register includes an identification of at least one of the plurality of abstracted Objects and where each abstracted Object is located.

37. The memory of claim 36, wherein the first data structure includes instructions for configuring the presentation space as a Browser.

38. The memory of claim 36, wherein the memory is co-located with at least one of the client device, a network server, a provider of the Enhanced Content, and a provider of a Temporal Signal to which the Enhanced Content relates.

39. The memory of claim 36, wherein at least one of the plurality of abstracted Objects includes an abstracted Receiver Object, wherein the abstracted Receiver Object provides at least one instruction which configures the Browser to receive at least one segment of Enhanced Content.

40. The memory of claim 36, wherein at least one of the plurality of abstracted Objects includes an abstracted Show, wherein the abstracted Show Object provides at least one instruction which configures the Browser to present at least one segment of Enhanced Content.

41. The memory of claim 36, wherein at least one of the plurality of abstracted Objects includes an abstracted Subscriber, wherein the abstracted Subscriber Object provides at least one instruction which configures the Browser to subscribe to at least one subscription service.

42. The memory of claim 41, wherein the at least one subscription service include a chat message service.

43. A process for creating a Virtual Stage on a client device comprising:

abstracting a Show Object, wherein the Show Object provides an abstraction of a platform upon which at least one segment of Enhanced Content provided by a Site may be presented; and

abstracting at least one Receiver Object, in communication with the Show Object, wherein the at least one Receiver Object provides an abstraction of a receiver for receiving from the Site at least one segment of Enhanced Content and providing the received Enhanced Content to the Show Object for presentation on the client device.

44. The process of claim 43, further comprising:

abstracting at least one Subscriber Object, in communication with the Receiver Object, wherein the at least one Subscriber Object provides at least one instruction with configures the client device to subscribe to at least one subscription.

45. The process of claim 44, wherein the at least one subscription includes a subscription to at least one chat message service.

46. A process for creating a Virtual Stage on a client device comprising:

identifying a main frame in the Register, wherein the main frame provides an abstracted framework upon which at least one segment of Enhanced Content may be presented;

identifying a control frame in the Register, wherein the control frame provides an abstracted framework for at least one Object used to control a presentation of Enhanced Content;

abstracting a Show Object, hosted by the main frame, wherein the Show Object provides an abstraction of a platform, upon the abstracted main frame, on which at least one segment of Enhanced Content provided by a Site may be presented; and

abstracting at least one Receiver Object, hosted by the control frame and in communication with the Show Object, wherein the at least one Receiver Object provides an abstraction of a receiver for receiving from the Site at least one segment of Enhanced Content and providing the received Enhanced Content to the Show Object.

47. The process of claim 46, further comprising:

establishing a Queue in a computer readable medium, wherein the Queue provides an indication of a condition and an event that is to be performed when the

condition is satisfied;

establishing a Register in a computer readable medium, wherein the Register provides an indication of at least one Object and where each of the at least one Object is located.

48. A process for identifying to an online information provider a type of Enhanced Content to present upon a Virtual Stage comprising:

establishing a communications link between a client device, upon which a Virtual Stage is to present Enhanced Content, and an online information provider; and

communicating a Skin from the client device to the online information provider;

wherein the Skin communicates information indicative of which Enhanced Content the Virtual Stage is to present and an identity of the client device upon which the Virtual Stage is to be established such that the Enhanced Content is compatible with and may be presented on the Virtual Stage hosted by the client device.

49. The process of claim 48, wherein the communications link between the client device and the online information provider is established over at least one of an Internet connection, a wireless link, an intranet connection, a satellite link, a cable link, a dial-up network, a regional network, a broadcast network, a multi-cast network, a simulcast network, and a wired link.

50. The process of claim 49, wherein the client device further comprises at least one of a personal computer, a personal data assistant, a Web tablet, a wireless communications device, a computer workstation, a gaming console, a set-top box, an Internet equipped television, a digital television, a Browser, a cable box, and a device capable of presenting Enhanced Content to a client.

51. An apparatus for executing an application program and being in communication with a database used by the application program to render a Virtual Stage in order to present at least one segment of Enhanced Content on the Virtual Stage comprising:

a processor for processing an application program which configures a presentation space, on a presentation device connected to the processor, for presenting at least one segment of Enhanced Content; and

a memory for storing at least one instruction utilized by the application

program to configure the presentation space as a Virtual Stage and further comprising:

an abstracted Stage Manager, wherein the abstracted Stage Manager controls the creation and operation of the Virtual Stage;

a plurality of abstracted Objects, each of said abstracted Objects being utilized to create the Virtual Stage;

an abstracted Queue, in communication with the abstracted Stage Manager, wherein the abstracted Queue comprises a list of tasks to be accomplished and Objects to be abstracted, under the direction of the Stage Manager, in creating and operating the Virtual Stage;

an abstracted Register, in communication with the abstracted Stage Manager and the abstracted Queue, wherein the abstracted Register includes an identification of at least one Object and where the abstracted Object is located;

an abstracted Receiver Object providing at least one instruction which configures the presentation space to receive at least one segment of Enhanced Content;

an abstracted Show Object providing at least one instruction which configures the presentation space to present at least one segment of Enhanced Content.

52. The apparatus of claim 51, wherein the presentation space is controlled by a Browser.

53. The apparatus of claim 51, wherein the memory is co-located with the processor.

54. The apparatus of claim 51, wherein the memory is remotely located with respect to the processor.

55. A signal embodied in a transmission medium for rendering a client device as a Virtual Stage upon which at least one segment of Enhanced Content may be presented to a client, comprising:

a first program code segment providing an abstraction of a Receiver Object, wherein the Receiver Object provides at least one instruction which configures the client device to receive at least one segment of Enhanced Content from an Enhanced Content provider;

a second program code segment providing an abstraction of a Show, wherein the Show Object provides at least one instruction which configures the client device

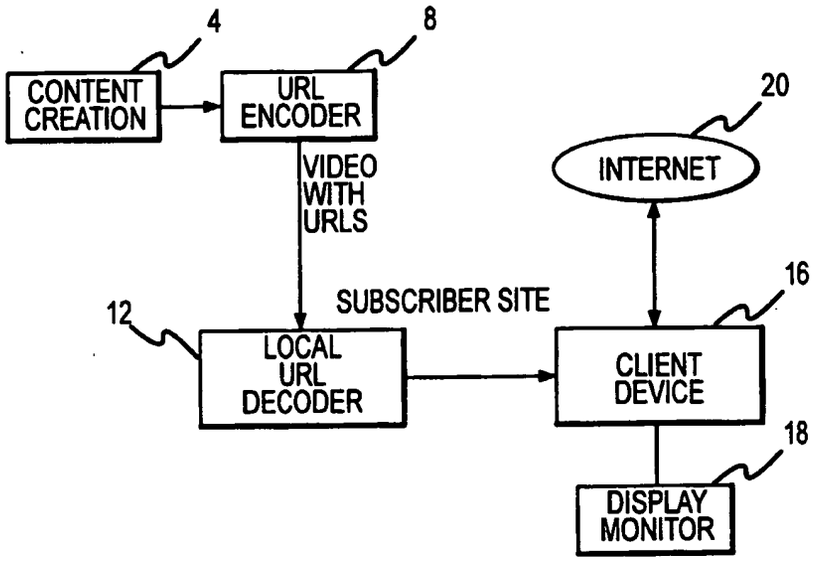
to present the at least one segment of Enhanced Content;

a third program code segment providing an abstraction of a Subscriber, wherein the Subscriber Object provides at least one instruction which configures the client device to subscribe to at least one subscription service;

a fourth program code segment providing an abstraction of a Control Area, wherein the abstracted Control Area contains those elements utilized by the client device to establish interfaces between the client device and at least one Enhanced Content provider; and

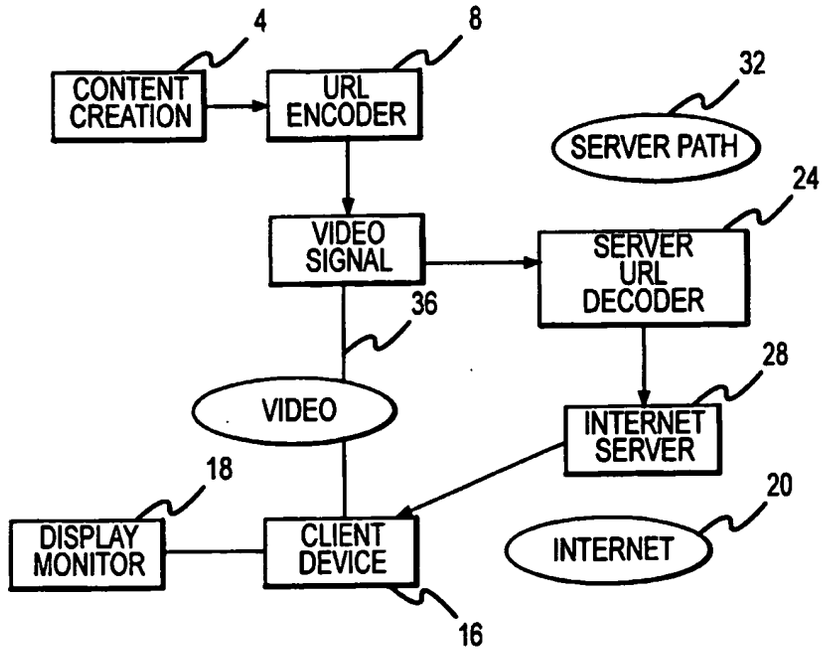
a fifth program code segment providing an abstracted Stage Manager, wherein the abstracted Stage Manager provides at least one instruction which configures the client device to create and operate the Virtual Stage;

whereupon establishment of a communications link between the client device and the Enhanced Content provider, the Receiver Object, Show Object and Subscriber Object configured client device receives the at least one segment of Enhanced Content, presents the received Enhanced Content segments in accordance with the configuration instructions specified by the Show, and subscribes to at least one of a subscription and a chat room in accordance with the instructions specified by the Subscriber.



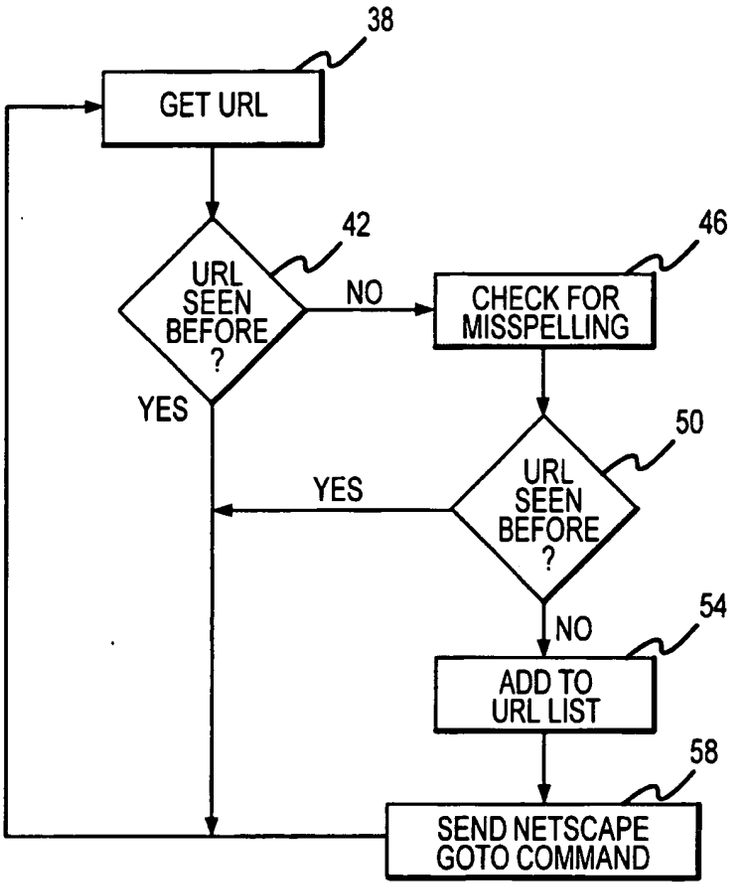
(PRIOR ART)

FIG.1



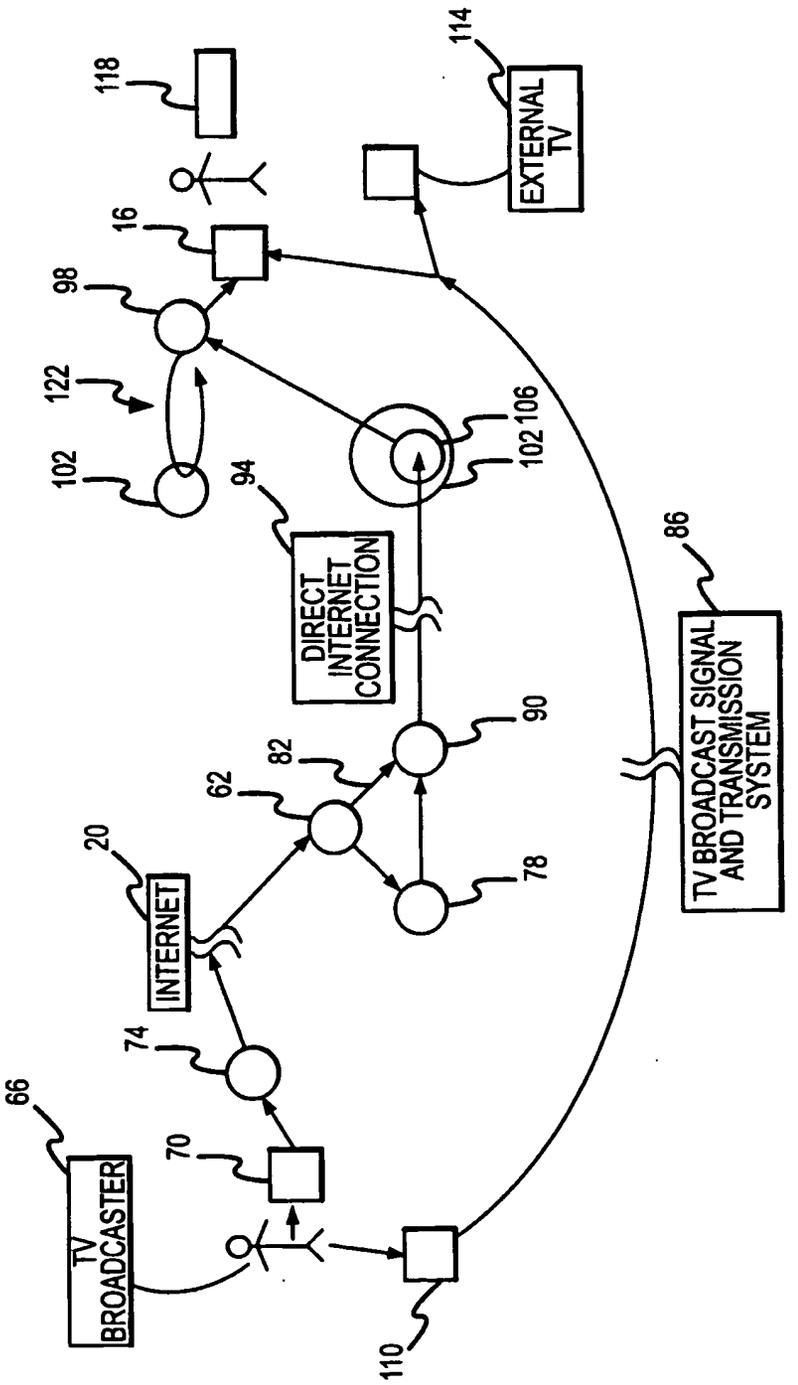
(PRIOR ART)

FIG.2



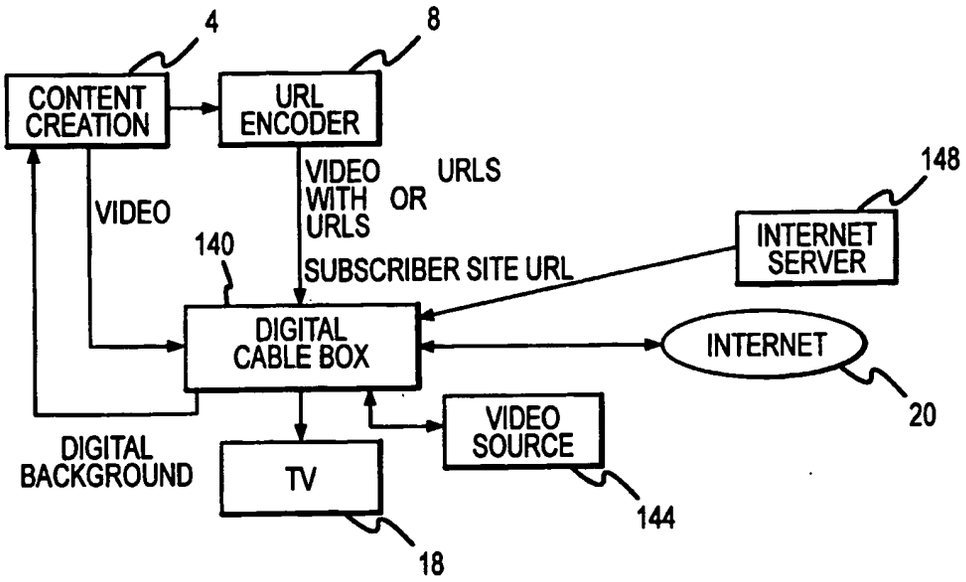
(PRIOR ART)

FIG.3



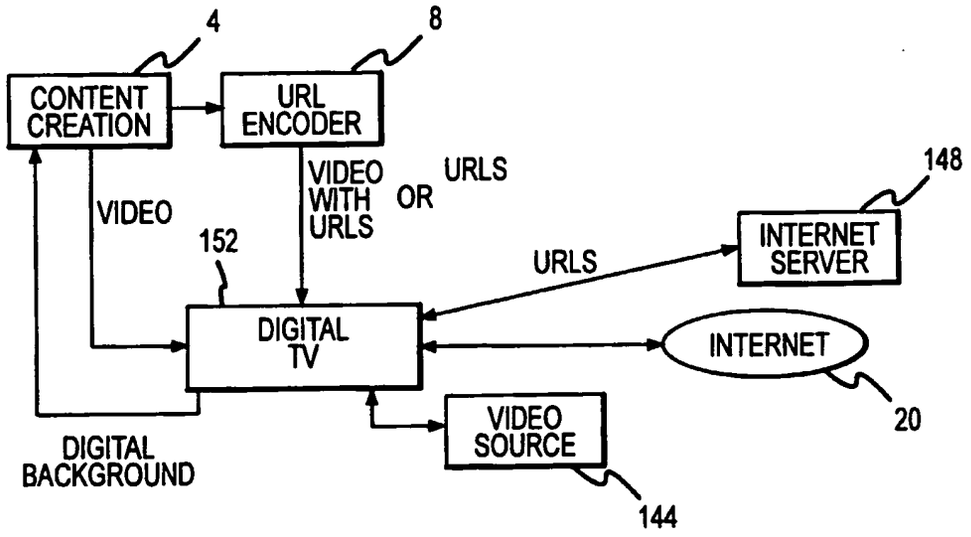
(PRIOR ART)

FIG.4



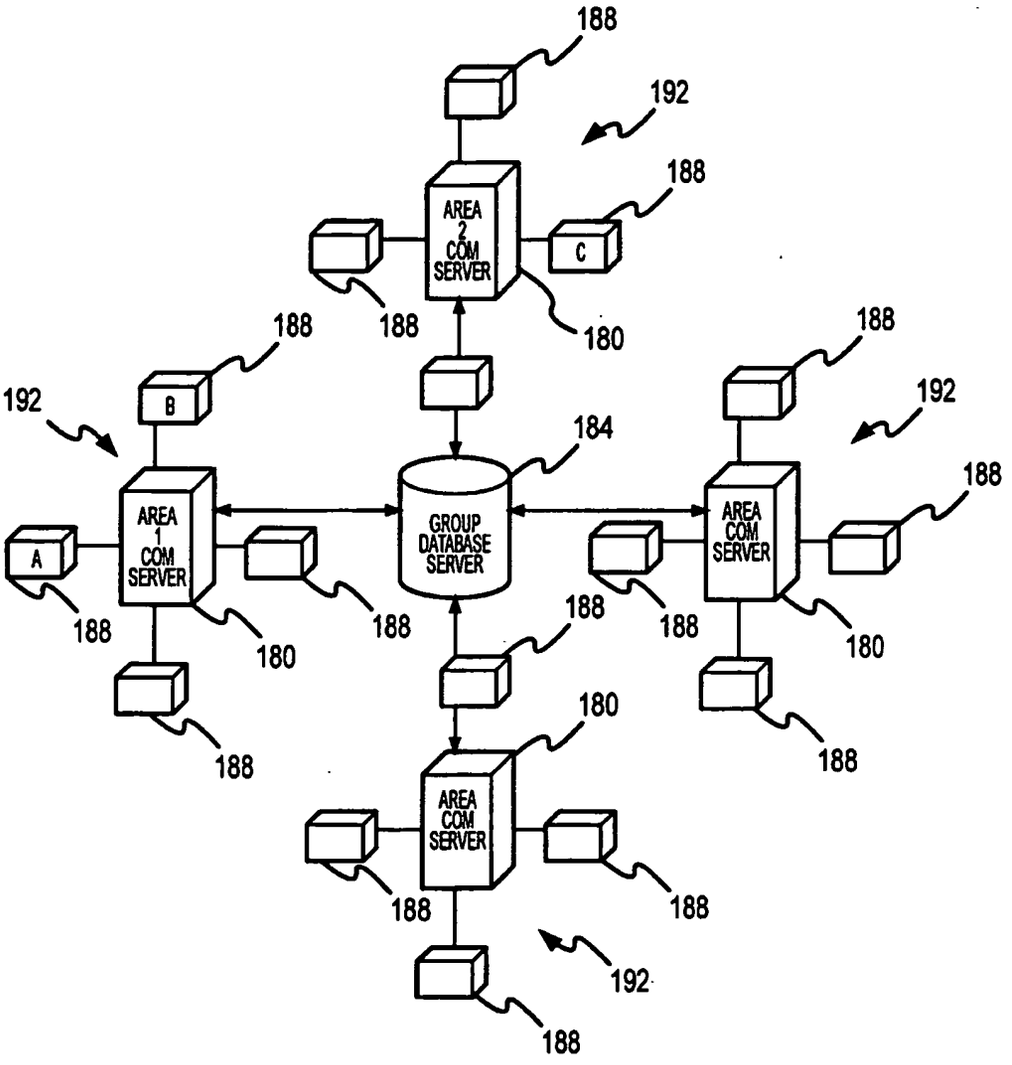
(PRIOR ART)

FIG.5



(PRIOR ART)

FIG.6



(PRIOR ART)

FIG.7

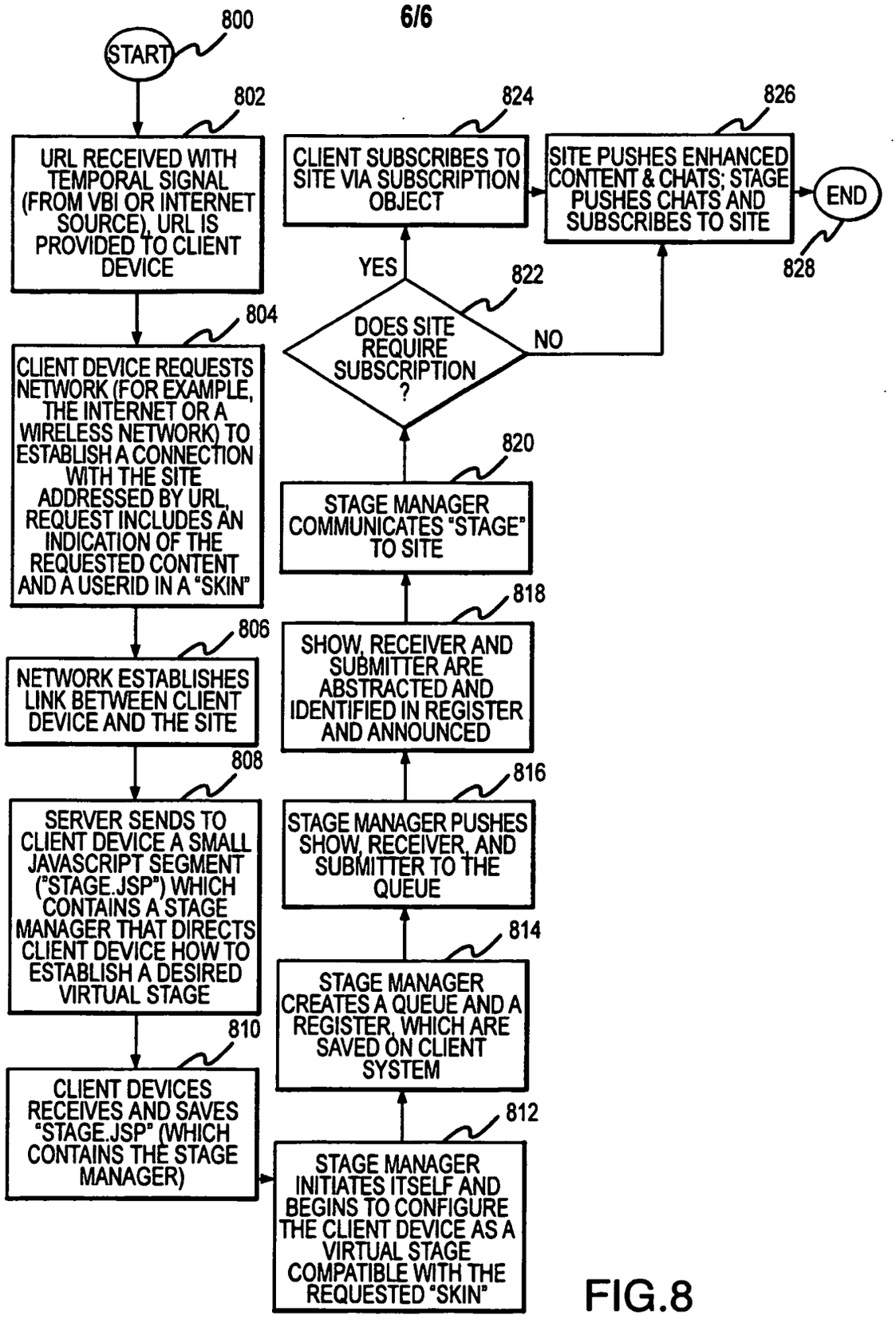

800
START
802
URL RECEIVED WITH TEMPORAL SIGNAL (FROM VBI OR INTERNET SOURCE), URL IS PROVIDED TO CLIENT DEVICE
804
CLIENT DEVICE REQUESTS NETWORK (FOR EXAMPLE, THE INTERNET OR A WIRELESS NETWORK) TO ESTABLISH A CONNECTION WITH THE SITE ADDRESSED BY URL, REQUEST INCLUDES AN INDICATION OF THE REQUESTED CONTENT AND A USERID IN A "SKIN"
806
NETWORK ESTABLISHES LINK BETWEEN CLIENT DEVICE AND THE SITE
808
SERVER SENDS TO CLIENT DEVICE A SMALL JAVASCRIPT SEGMENT ("STAGE.JSP") WHICH CONTAINS A STAGE MANAGER THAT DIRECTS CLIENT DEVICE HOW TO ESTABLISH A DESIRED VIRTUAL STAGE
810
CLIENT DEVICES RECEIVES AND SAVES "STAGE.JSP" (WHICH CONTAINS THE STAGE MANAGER)
812
STAGE MANAGER INITIATES ITSELF AND BEGINS TO CONFIGURE THE CLIENT DEVICE AS A VIRTUAL STAGE COMPATIBLE WITH THE REQUESTED "SKIN"
814
STAGE MANAGER CREATES A QUEUE AND A REGISTER, WHICH ARE SAVED ON CLIENT SYSTEM
816
STAGE MANAGER PUSHES SHOW, RECEIVER, AND SUBMITTER TO THE QUEUE
818
SHOW, RECEIVER AND SUBMITTER ARE ABSTRACTED AND IDENTIFIED IN REGISTER AND ANNOUNCED
820
STAGE MANAGER COMMUNICATES "STAGE" TO SITE
822
DOES SITE REQUIRE SUBSCRIPTION ?
YES
NO
824
CLIENT SUBSCRIBES TO SITE VIA SUBSCRIPTION OBJECT
826
SITE PUSHES ENHANCED CONTENT & CHATS; STAGE PUSHES CHATS AND SUBSCRIBES TO SITE
828
END

FIG.8

IFW

Atty. Docket No. 559442600201

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Craig ULLMAN et al.

Serial No.: 10/299,335

Filing Date: November 18, 2002

For: ENHANCED VIDEO PROGRAMMING SYSTEM AND METHOD FOR INCORPORATING AND DISPLAYING RETRIEVED INTEGRATED INTERNET INFORMATION SEGMENTS

Examiner: Not yet assigned

Group Art Unit: 2611

Confirmation No. 6304

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

Pursuant to 37 C.F.R. §1.97 and § 1.98 and MPEP 2001.06(c), applicant submits for consideration in the above-identified application the following disclosure. All of the documents identified below are believed to be of record already. If an additional copy of any document is required, the Examiner is invited to contact applicant.

These documents were cited by defendants in the matter of ACTV, Inc. and HyperTV Networks Inc. v. The Walt Disney Co., ABC, Inc. and ESPN, Inc in U.S. District Court for the Southern District of New York (Civil Action No. 00 Civ. 9622 JSR) as relevant to the validity of one or more claims in U.S. Patent Nos. 5,778,181 (Ser. No. 615,143), 5,774,664 (Ser. No. 622,474) and 6,018,768 (U.S. Ser. No. 09/109,945). Applicant would appreciate the Examiner initialing and returning the Form PTO/SB/08a/b, indicating that the information has been considered and made of record herein.

va-163168

In the above-referenced legal proceeding, defendants also alleged inequitable conduct in the prosecution of the '181, '664 and '768 patents. The '768 patent is a continuation-in-part of the '181 patent and is related to the '664 patent.

a) Defendants alleged intentional failure to disclose U.S. Patent No. 5,479,268 to Young et al. in prosecution of the above-referenced patents. Patent-owner showed that the '268 patent is cumulative with U.S. Patent No. 5,589,892 to Knee et al. which was cited by the Examiner during prosecution of the '181 patent. Patent-owner also showed that the same examiner responsible for examining the '181, '664 and '768 patents issued a search report in a PCT application claiming priority to the '181 and '664 patents in which the '268 was cited as being of "particular relevance."

b) Defendants alleged misrepresentation in the "Summary of the Invention" sections of the above-referenced patents of the prior art Intercast system as a one-way analog instead of a two-way digital system. Patent-owner showed that the Intercast product literature distinguishes the Intercast system from two-way systems and that each of the three co-inventors of the Intercast system testified in depositions that the Intercast system is a one-way analog system.

c) Defendants alleged intentional failure to disclose that the subject matter of claims 19 and 20 of the '768 patent was on sale as of July 6, 1997, more than one year prior to its filing. Patent-owner showed that the Examiner found that claim 19 was fully supported by earlier disclosures including application Serial No. 613,144, filed on March 8, 1996, and in the application that matured into the '181 patent. Patent-owner showed that the subject matter of claim 20 was not complete and ready for patenting before July 7, 1997 and that the claims of the '768 patent were supported by the '181 patent.

d) Defendants alleged intentional failure to disclose U.S. Patent No. 5,761,606 to Wolzien in prosecution of the '768 patent. Patent-owner showed that the '606 patent was cited by the Examiner during prosecution of the application that issued as the '768 patent.

e) Defendants alleged intentional failure to disclose U.S. Patent No. 5,818,441 to Throckmorton et al. in prosecution of the '768 patent. Patent-owner showed that there was no evidence that the inventors or the prosecuting attorney had reviewed the '441 patent until after the '768 patent had already issued. Patent-owner also showed that the '441 patent was cumulative with the Intercast system and the '606 patent to Wolzien.

f) Defendants alleged intentional misrepresentation in prosecution of the '768 patent by the assertion that ACTV had a one hundred percent ownership interest in the '181 and '664 patents. Patent-owner showed that the assignments made in the '181 and '664 patents reflect that the inventors of the '181 and '664 patents had sold, assigned and transferred their entire right, title and interest in those patents to ACTV.

The above-referenced case was settled and no decision on the merits was reached with respect to these contentions. The relevant pleadings were filed under seal pursuant to a protective order and are not publicly available. Applicant asserts that the above description of the allegations made and responses plead is accurate and complete.

The information contained in this Supplemental Information Disclosure Statement under 37 C.F.R. § 1.97 and § 1.98 is not to be construed as a representation that: (i) a complete search has been made; (ii) additional information material to the examination of this application does not exist; (iii) the information, protocols, results and the like reported by third parties are accurate or enabling; (iv) the above information constitutes prior art to the subject invention; (v) applicant admits to the inequitable conduct allegations in whole or in part.

In the unlikely event that the transmittal form is separated from this document and the Patent Office determines that an extension and/or other relief (such as payment of a fee under 37 C.F.R. §

1.17 (p)) is required, applicant petitions for any required relief including extensions of time and authorizes the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to Deposit Account No. 03-1952 referencing 559442600207.

Dated: May 17, 2006

Respectfully submitted,

By _____
James M. Denaro
Registration No.: 54,063
MORRISON & FOERSTER LLP
1650 Tysons Blvd, Suite 300
McLean, Virginia 22102
(703) 760-7700

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/299,335
Filing Date	November 18, 2002
First Named Inventor	Craig ULLMAN
Art Unit	2611
Examiner Name	Not yet assigned
Attorney Docket Number	559442600201

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1.	5,479,268	12/26/1995	Young et al.	
	2.	5,724,103	3/3/1998	Batchelor	
	3.	5,761,606	6/2/1998	Wolzien	
	4.	5,782,692	6/21/1998	Stelovsky	
	5.	5,818,441	10/6/1998	Throckmorton et al.	
	6.	5,905,865	5/18/1999	Palmer et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]

*EXAMINER: Initial if information considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	7.	Press Release, Intercast Industry Group, *Leaders in PC, Broadcast and Cable Industries Announce Formation of Industry Group to Promote New Digital Medium for the Home PC,* Business Wire, Oct. 23, 1995.	
	8.	Per Einar Dybvik and Hakon W. Lie, *Combining WWW/Mosaic with Realtime Multimedia Conferencing in Distance Education,* the Second International WWW Conference '94, Mosaic and the Web, Advance Proceedings, Vol. 1, October 17-20, 1994, at 423.	
	9.	Tak K. Woo and Michael J. Rees, *A Synchronous Collaboration Tool for the World-Wide Web,* The Second International WWW Conference '94, Mosaic and the Web, Advance Proceedings, Vol. 1, October 17-20, 1994, at 315.	
	10.	Vinay Kumar, Jay Glicksman and Glenn A. Kramer, *A SHAREd Web to Support Design Teams*, Third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Morgantown, West Virginia, April 17-19, at 178.	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1]Applicant's unique citation designation number (optional). [2]Applicant is to place a check mark here if English language Translation is attached.

Examiner Signature		Date Considered	

va- 163173

IFW

PATENT
Docket No. 559442600201

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of:	
Craig ULLMAN et al.	Examiner: Not yet assigned
Serial No.: 10/299,335	Group Art Unit: Not yet assigned
Filing Date: November 18, 2002	Confirmation No.: 6304
For: ENHANCED VIDEO PROGRAMMING SYSTEM AND METHOD FOR INCORPORATING AND DISPLAYING RETRIEVED INTEGRATED INTERNET INFORMATION SEGMENTS	

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

Pursuant to 37 C.F.R. § 1.97 and § 1.98, Applicants submit for consideration in the above-identified application the documents listed on the attached Form PTO/SB/08a/b. Copies of the foreign documents and non-patent literature are also submitted herewith. The Examiner is requested to make these documents of record.

This Supplemental Information Disclosure Statement is submitted before mailing of a first Office Action on the merits; accordingly, no fee is required.

Applicants would appreciate the Examiner initialing and returning the Form PTO/SB/08a/b, indicating that the information has been considered and made of record herein.

va-166710

The information contained in this Supplemental Information Disclosure Statement under 37 C.F.R. § 1.97 and § 1.98 is not to be construed as a representation that: (i) a complete search has been made; (ii) additional information material to the examination of this application does not exist; (iii) the information, protocols, results and the like reported by third parties are accurate or enabling; or (iv) the above information constitutes prior art to the subject invention.

In the event that the transmittal form is separated from this document and the Patent Office determines that an extension and/or other relief (such as payment of a fee under 37 C.F.R. § 1.17 (p)) is required, Applicants petition for any required relief including extensions of time and authorize the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to **Deposit Account No. 03-1952** referencing 559442600201.

Dated: June 23, 2006

Respectfully submitted,

By: _____

Jonathan Bockman
Registration No. 45,640
Morrison & Foerster LLP
1650 Tysons Boulevard
Suite 300
McLean, Virginia 22102
Telephone: (703) 760-7769
Facsimile: (703) 760-7777

va-166710

ALTERNATIVE TO PTO/SB/08a/b (07-05)

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/299,335
Filing Date	November 18, 2002
First Named Inventor	Craig ULLMAN
Art Unit	Not yet assigned
Examiner Name	Not yet assigned
Attorney Docket Number	559442600201

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1.	4,602,279	7-22-1986	Freeman et al.	
	2.	6,006,256	12-21-1999	Zdepski et al.	
	3.	6,076,072	6-13-2000	Libman	
	4.	6,263,505	7-1-2001	Walker et al.	
	5.	6,630,963	10-7-2003	Billmaier	
	6.	2005-0097622	5-5-2005	Zigmond et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
	7.	WO-98/03016-A1	1-22-1998			
	8.	EP-0915621	5-12-1999			
	9.	WO-01/58159	8-9-2001			

*EXAMINER: Initial if information considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	10.	Papadimitriou, C. H. et al. "Information Catching for Delivery of Personalized Video Programs on Home Entertainment Channels," Multimedia Computing and Systems, May 15, 1994, Proceedings of the Int'l Conference on Boston, MA, pp. 214-223	
	11.	Ramanathan, S. et al. "Architectures for Personalized Multimedia," IEEE Multimedia, March 21, 1994, pp. 37-46	
	12.	Venkat, Rangan et al. "Designing an On-Demand Multimedia Service," IEEE Communications Magazine, IEEE Service Center, July 1992, pp. 56-64	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1]Applicant's unique citation designation number (optional). [2]Applicant is to place a check mark here if English language Translation is attached.

Examiner Signature		Date Considered	

va- 166692

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/16 — A1

(11) International Publication Number: WO 98/03016

(43) International Publication Date: 22 January 1998 (22.01.98)

(21) International Application Number: PCT/US97/12001

(22) International Filing Date: 10 July 1997 (10.07.97)

(30) Priority Data:
60/021,644 12 July 1996 (12.07.96) US

(71) Applicant *(for all designated States except US)*: INTERACTIVE PICTURES CORPORATION [US/US]; Suite 100, 1009 Commerce Park, Oak Ridge, TN 37830 (US).

(71)(72) Applicants and Inventors: MARTIN, H., Lee [US/US]; 11615 South Monticello Drive, Knoxville, TN 37760 (US). GRANTHAM, H., Craig [US/US]; 717 East Broadway, Jefferson City, TN 37760 (US).

(74) Agents: POTENZA, Joseph, M. et al.; Banner & Witcoff, Ltd., Suite 1100, 1001 G Street, N.W., Washington, DC 20001 (US).

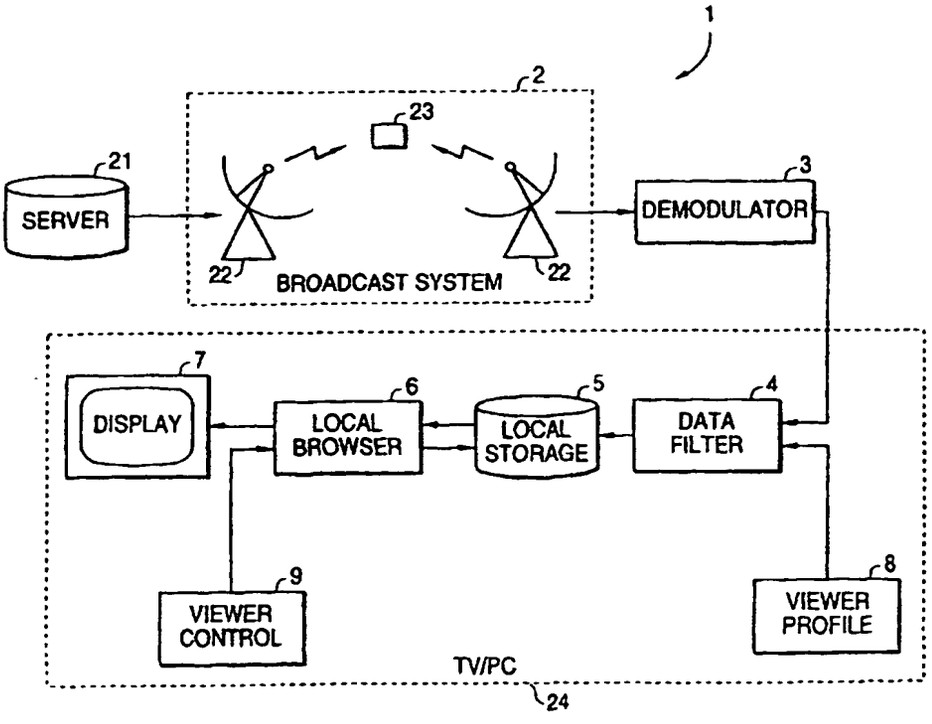
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: VIEWER PROFILE OF BROADCAST DATA AND BROWSER

(57) Abstract

A method and apparatus for broadcast information distribution of digital data and images with selective capture and interaction with those images without any need for bidirectional communications. The apparatus receives a continuous transmission of data broadcast through common unidirectional methods typically associated with television (free space transmission, satellite transmission, cable transmission) (21, 22, 23), the information is collected based on criteria established by the viewer (8) and is then stored (5) for later interactive review (6, 7) by the viewer. The method and apparatus can be used in a system where massive amounts of interactive data are to be distributed, but the infrastructure of the communications network is such that bidirectional communications are not feasible. The collection of the information is done in a manner as to capture only the portion of information that is of interest to the viewer, based on his/her interests. In such a way, a data network with interactive images can provide comprehensive, interactive images for real estate, apartment, automobile, and employment listings.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

VIEWER PROFILE OF BROADCAST DATA AND BROWSER

TECHNICAL FIELD

The invention relates to a method and apparatus for processing broadcast data streams and, in particular, to a method and apparatus for allowing the selective capture of digital data and then allowing interactivity with the captured data at the convenience of the viewer.

BACKGROUND OF THE INVENTION

The Internet has grown rapidly in availability and use in recent years. This growth has resulted in a number of data transmission and distribution problems. Conventionally, the communications infrastructure has been developed either for low-bandwidth, bidirectional communications (telephone) or for high-bandwidth, unidirectional transmission (television). Recently, this conventional infrastructure has been adapted to accommodate Internet and Intranet communications requirements. As a result, systems have been constrained at the distribution point due to server performance and capacity limitations, and at the destination point due to modem reception rate limitations. Additionally, these systems may also be constrained across "backbone" communication links as the bandwidth requirements for satisfying the exploding Internet user base increase beyond the available infrastructure. It is anticipated that these problem will be exasperated due to the advent of cable modems, asynchronous transmission modems, and other devices.

Attempts at alleviating some of the above mentioned problems include the use of intelligent "agents" to provide for local data selection. Intelligent agents are programmed by a user to search the available sources of on-line information to identify data which meets a predetermined criteria and to present this data to the user. The agents move throughout a network collecting data and return data which has the best correlation to the selection criteria. However, although intelligent agents reduce the local response time for an individual user, the local agents increase the load on the network "backbone" structure.

In a similar, but more simplified approach, PointCast has developed an information network that allows the viewer to complete a profile on his/her interests, and each time the viewer attaches to the PointCast Internet site, the pertinent information of interest is sent to the viewer. The agent becomes a list of desired information that is sent by the server to the viewer whenever the viewer logs onto the system.

Outside of the Internet arena, Direct broadcast satellites use encrypted codes to allow only the target units to receive the transmission information.

Additionally, Intercast systems that broadcast digital data during the vertical blanking interval of the television signal also exist to provide services, such as closed-captioning and other information for viewers. The feasibility of broadcast of digital information in a region, nationally, or internationally is established, and the capture of such data with an appropriately equipped personal computer is also proven in the art of broadcast television.

Conventionally, the Internet and its supporting infrastructure have a number of limitations such as bandwidth constraints and server constraints. Since communications in the

Internet are point-to-point, every user must be connected to a host in a bidirectional manner that is uniquely responsive. This makes the network model look much like a switched circuit telephone system which limits the bandwidth in current configurations.

Additionally, servers are constrained by the need to respond to requests by individual users and the Internet backbone resources are taxed by continually active intelligent agents roaming the Internet. Accordingly, there is a need for rapid access to a selected portion of a wide array of information stored in centralized databases (e.g., the Internet or other Intranets), without the need for bidirectional communications.

SUMMARY OF THE INVENTION

Objects of the invention include providing an apparatus and a method for remedying the above problems by eliminating the need for communications back to the server from the local terminal device and the need for intelligent agents which continuously query the server.

One aspect of the present invention is a local terminal device such as a television or personal computer which accepts all information broadcast from the server but retains in local storage only the information of interest to the viewer. Once this information is captured, the viewer can interact with the information locally to browse the Internet (e.g., via hypertext or other html links) in the manner desired. By using broadcast means to distribute digital data, massive amounts of data can blanket the country from a single server and available infrastructure. The invention does not require bidirectional communications, so the resulting network can resemble a television broadcast providing data at a much higher rate.

Additionally, the local terminal device can browse through the captured data much more efficiently without the bandwidth constraints and associated delay of present on-line information sources. Since most on-line information sources (such as the Internet and other on-line information sources) typically provide much more information than the user has an interest in receiving, the present system allows the user to have highspeed and up-to-date access for information which is of interest.

The invention allows for the distribution of the processing burden from the server to each of the local terminals by utilizing a profile of the viewer's interest to filter broadcast data and thereby select the data that is to be captured and stored locally. Thus, the invention allows for the economical availability of massive, up-to-date data possible on a wide scale to the general populace.

In accordance with the present invention, there is provided an information system that allows selective capture of broadcast digital information, based on the local desires of the viewer, without any need for communications back to the source. The invention includes a means for receiving a broadcast digital stream of data, means for automatically capturing that data based on specific criteria set forth by the viewer, means for saving the captured data to a local storage media for later review, and means for navigating the captured data under the control of a viewer after the data that has been stored locally.

The content captured for the viewer is preferably a subset of the entire stream of data that is received by the viewer's unit, based upon the current interests and needs of the viewer. More particularly, the viewer's unit captures all data of interest, as specified by the viewer,

without the need for a back channel communications link to the originating source of the data. In this manner, the processing load on the central server is minimized in the need for a return communications path is eliminated. The captured data can be interacted upon by the viewer as if he/she were on-line, but the interactivity is with data that has recently been captured to a local storage media, rather than to the on-line server. In this way, millions of viewer units can receive the same broadcast data stream (similar to television signal distribution), but can selectively interact with the information of their interests (similar to on-line Internet) from a region of storage on their local storage media (most preferably, a hard disk). Thereafter, a conventional browser may provide the navigation means to explore the captured data.

The result is personalized data extracted individually from a massive stream of data that is readily accessible without a change in the communications distribution infrastructure presently available. The invention has wide applicability, but may be particularly applicable for updating specialized data bases where the on-line subscriber requires a up-to-date information which is a subset of the total data base contained on the server.

Although the invention has been described, in general, in the "Summary Of The Invention" section, it should be noted that the invention includes any of the components, functions, and/or steps described, claimed, and/or shown herein when used in any combination or subcombination. Accordingly, there are any number of alternate combinations for defining the invention which combine one or more elements from the existing claims and/or from the specification in various combinations or subcombinations.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a partial flow and partial block diagram of aspects of the present invention;

Fig. 2 shows a flow diagram of an example of the data extraction analysis (filtering) in accordance with aspects of the present invention;

Fig. 3 shows a block diagram of local terminal device in accordance with aspects of the present invention; and

Fig. 4 shows a block diagram of a physical arrangement of aspects of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Referring to Fig. 1, a broadcast data retrieval system 1 may include a server 21 which outputs a stream of, for example, all of the data available on a network (e.g., the Internet or a particular Intranet) and/or another database of information in a continuous manner for transmission via a broadcast system 2. The broadcast system 2 may be variously configured to include any suitable broadcast medium, but most preferably includes one or more earth stations 22 and at least one communication satellite 23. Of course, those skilled in the art will appreciate that other broadcast techniques (e.g., free space transmission, satellite transmission, and/or cable transmission) are also suitable to practice aspects of the present invention.

In preferred embodiments, the data broadcast by broadcast system 2 is demodulated by demodulator 3 to convert the data from the broadcast system, e.g., radio frequency waves, into

digital date suitable for processing by a local processing device 24, e.g., a TV and/or a PC 24. Viewed functionally, the data from demodulator 3 is first input into a data filter 4. Data filter 4 analyzes the broadcast data stream received from demodulator 3 and filters out undesirable data based on a viewer's profile 8. The desired information is then saved in local storage 5, e.g., a conventional optical or magnetic disk. The user profile information is preselected by a viewer. Alternatively, and less preferably, the user profile information for a particular local processing device may be supplied by an information provider using any suitable method. After the data review process has begun, the viewer may browse the locally stored data in server 5 via local browser 6, viewer control 9 and display 7. Thus, the local processing device 24 preferably stores a subset of the information broadcast from server 21, which may thereafter by viewed by a viewer.

Fig. 3 shows the main components of the apparatus when it is used as a local browser of information. In Fig. 3, the local processing device 24 may include a local storage 5, a demodulator 3, central processor 12, display 7, and viewer control 9. A viewer inputs data requests through the viewer control 9, e.g., a remote control, a mouse and/or a keyboard. The central processor (e.g., a central processor of a PC and/or a TV) interprets the viewer controls to search the local disk 5 for information pertaining to the desires of the viewer. The resulting data in the form of text, graphics, images, and/or video are made available to the user via suitable output mechanism such as display monitor 7.

Fig. 4 shows a block diagram of the components of an exemplary broadcast data retrieval system 1 arranged in block diagram form. The components similar to the components

of Fig. 1 have the same reference numerals and have already been described above. In the embodiment shown in Fig. 4, it should be noted that the demodulator 3 may be part of or separate from the local processing device 24. For example, the local processing device 24 may be variously configured to include demodulator 3, local storage 5, display 7, user interface 9, CPU 26 and/or memory 27. Additionally, each of the components of the local processing device 24 may be coupled together in any suitable configuration such as by using one or more system buses 28. Further, server 21 preferably includes a modulator for modulating digital data transmitted via broadcast system 2. One or more software programs for implementing the digital filtering, user interface, and browsing functions are preferably stored in the local storage 5.

In operation, the application software is loaded in memory 27 for operation with CPU 26 and demodulator 3. In the most preferred embodiments, the filtering software operates continuously on the digital data received by demodulator 3. Alternatively, the local processing device 24 interrupts the monitoring of digital data when accessed by a user in order to better allocate local processing resources to the user's application programs such as the browser. When accessed by a user, the browsing software and/or software for capturing a viewer preferences is also preferably loaded from local storage 5 into memory 27 and processed by CPU 26.

Fig. 2 shows a functional flow diagram of one example of aspects of the present invention. For example, an incoming digital data stream 10 received via demodulator 3 contains a large amount of information. Local extraction 11 via data filter 4 in conjunction

with the viewer's profile 8 operates to extract data from the digital data stream. In the exemplary embodiment illustrated in Fig. 2, local data filter 11 selects data representative of homes in the price range of 100 to 130 thousand dollars. The selected data is thereafter stored in local storage 5 for later review. Data which does not match the requirements specified by the viewer's profile may be discarded (i.e., simply not retained in local storage 5). The result is that an extensive data stream 10 can be significantly and intelligently reduced and captured for local review and interaction.

Other examples include updating specialized data bases which are now only updated on a quarterly basis via one or more CD ROMs. For example, numerous database service providers distribute quarterly updates of certain database by sending out new CD ROMs to their subscribers. The present system could be utilized to provide monthly, daily, or even hourly updates via broadcast system 2.

Still other examples may include a national used car network in digital form. For example, a data base of used cars including photos, pricing, make, model, and year, could be implemented about all of the vehicles presently available nationwide. One local viewer may not have a present interest in a car, therefore, none of the car data would be captured. Another local viewer might be looking for a late model Chevrolet, and only late model Chevrolets would be captured to the viewer's local disk for later viewing. Yet another viewer might be looking for any car costing less than $1,000, and all matches would be captured off the broadcast data for the viewer to look at conveniently. Networks for automobiles, real estate, rental properties, job, or classified advertisements of any type could be established.

In this manner, broadcast distribution of digital data and images can be accomplished with selective capture and interaction with the digital data and images without the need for bidirectional communications. The apparatus receives a continuous transmission of data (e.g., unidirectional data) broadcast through common unidirectional methods typically associated with television (e.g., free space transmission, satellite transmission, cable transmission). The information is collected based on criteria established by the viewer, and is then stored for later interactive review by the viewer. Embodiments of the invention can be used in a system where massive amounts of interactive data are to be distributed using a communications network infrastructure where bidirectional communications are not feasible. The collection of the information is done in a manner as to capture only the portion of information that is of interest to the viewer, based on his/her interests. In this manner, a data network with interactive images can provide comprehensive, interactive images for real estate, apartment, automobile, and employment listings, and sundry of speciality catalogues such as clothing, sporting goods, plants etc.

While exemplary broadcast data retrieval systems embodying the present invention are shown, it will be understood, of course, that the invention is not limited to these embodiments. Modifications may be made by those skilled in the art, particularly in light of the foregoing teachings. It is, therefore, intended that the appended claims cover any such modifications which incorporate the features of this invention or encompass the true spirit and scope of the invention.

WHAT IS CLAIMED IS:

1. A broadcast retrieval system including:

a server including data;

a unidirectional broadcast system means for broadcasting the data as a digital data stream;

demodulator means for demodulating the digital data stream;

means for capturing a viewer interest profile;

local processor means for selectively capturing a subset of the data in the digital data stream using a filtering means for retaining data based on the viewer interest profile;

means for storing the subset of the data in a local storage; and

means for browsing and viewing a portion of the subset of the data stored in the local storage based on input from a viewer.

2. The broadcast retrieval system of claim 1 wherein the local processor does not include a return path to the server.

3 The broadcast retrieval system of claim 1 wherein the digital data includes text and graphics.

4. The broadcast retrieval system of claim 1 wherein the digital data includes video.

5. The broadcast retrieval system of claim 1 wherein the digital data includes hypertext links.

6. An apparatus including:

a demodulator for receiving a unidirectional digital data stream;

a user interface for inputting data from a user;

a local storage device for storing data and a viewer interest profile;

a display for displaying data; and

a central processor coupled to the demodulator, the user interface, the local storage device, and the display for selectively storing in the local storage device a subset of the unidirectional digital data stream responsive to the viewer interest profile and for browsing the subset of the data stored in the local storage responsive signals from the user interface.

7. The apparatus of claim 6 wherein the unidirectional digital data stream includes text and graphics.

8. The apparatus of claim 7 wherein the unidirectional digital data stream includes video.

9. The apparatus of claim 7 wherein the unidirectional digital data stream includes hypertext links.

10. An article of manufacture including a disk storing software, the software including a program for selectively storing a subset of a unidirectional digital data stream responsive to a viewer interest profile and for selectively browsing the subset of the unidirectional digital data stream.

11. A method comprising:

broadcasting a unidirectional digital data stream;

demodulating the unidirectional digital data stream;

inputting a viewer interest profile;

selectively storing in a local storage device a subset of the data in the unidirectional digital data stream responsive to the viewer interest profile; and

thereafter, browsing the subset of the data in the local storage responsive to commands received from a user.

12. The method of claim 11 wherein browsing the data includes browsing hypertext data.

FIG. 1

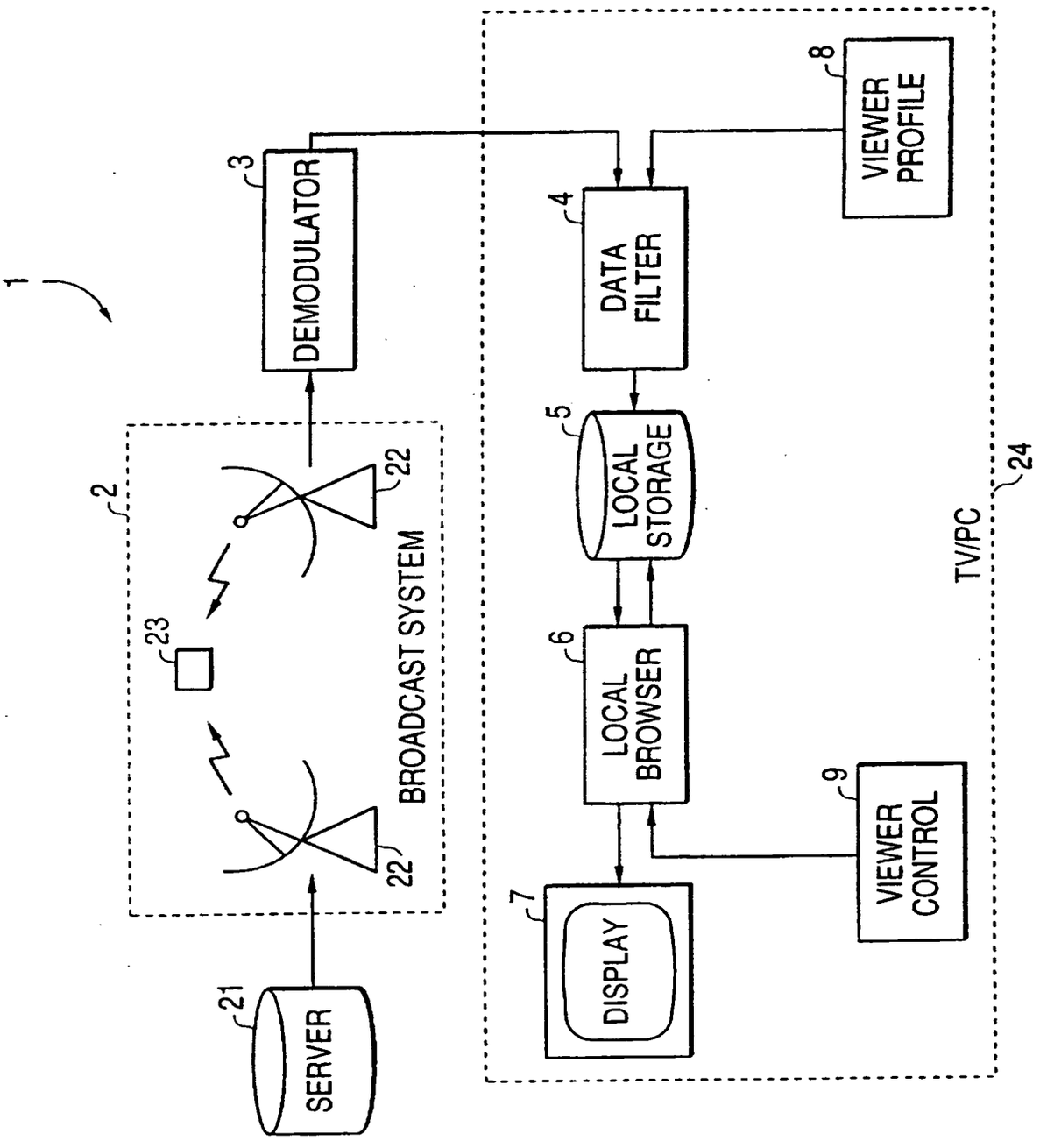


FIG. 2

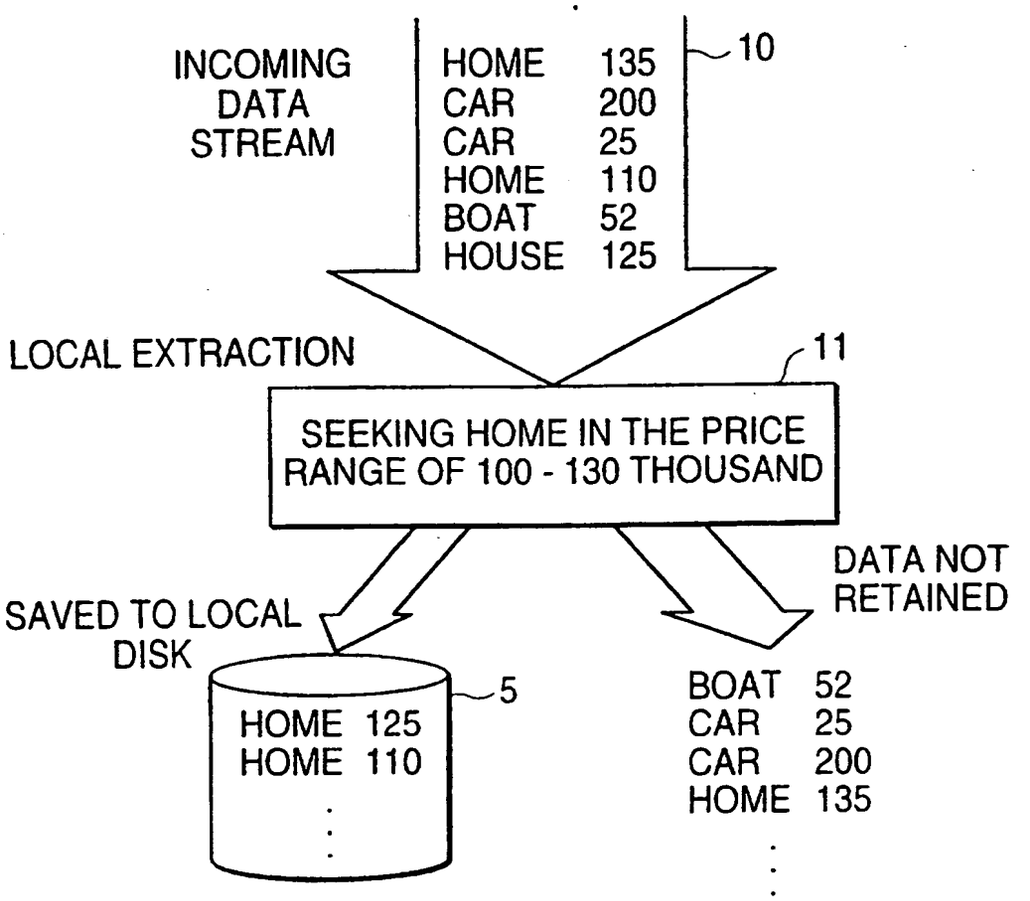


FIG. 3

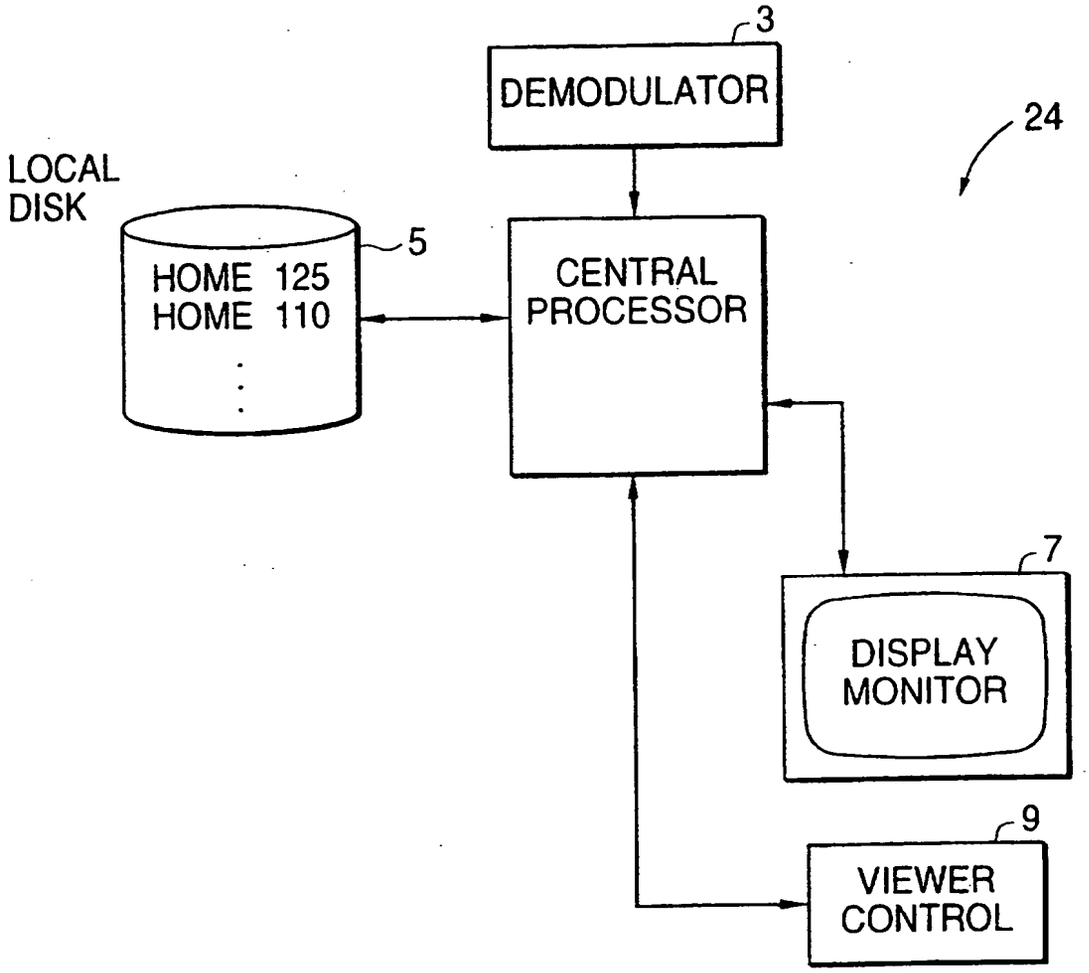
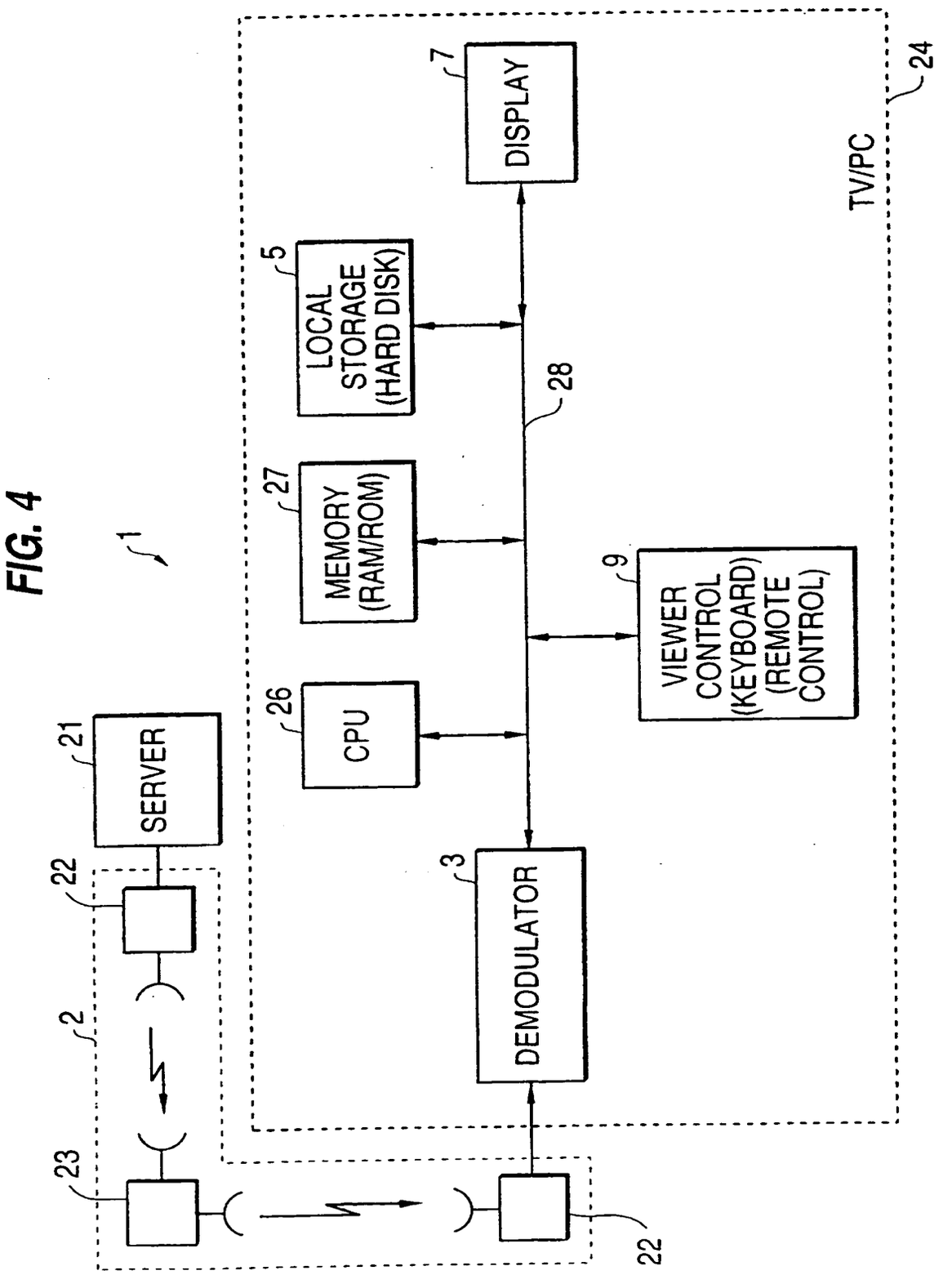


FIG. 4



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/12001

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :H04N 7/16
US CL :348/1; 455/2
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 348/1, 2, 3, 6, 7, 10, 12, 13; 455/2, 4.2, 5.1, 6.1, 6.2, 6.3

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 5,469,206 A (STRUBBE ET AL.) 21 November 1995, see entire document	6-12 --- 1-5
Y	US 5,446,490 A (BLAHUT ET AL.) 29 August 1995, see figure 1, col. 3, line 21 - col. 4, line 10	1-5
X	US 5,465,113 A (GILBOY) 07 November 1995, see entire document	6-12
X	US 5,481,296 A (CRAGUN ET AL.) 02 January 1996, see entire document	6-12
X	US 5,410,344 A (GRAVES ET AL.) 25 April 1995, see entire document	6-12

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 27 SEPTEMBER 1997	Date of mailing of the international search report 29 OCT 1997
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer CHRISTOPHER GRANT Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/12001

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,038,211 A (HALLENBECK) 06 August 1991, see entire document	6-12

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 915 621 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
12.05.1999 Bulletin 1999/19

(51) Int Cl.[6]: **H04N 7/173**

(21) Application number: **98308794.1**

(22) Date of filing: **27.10.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **06.11.1997 US 965633**

(71) Applicant: **LUCENT TECHNOLOGIES INC.**
Murray Hill, New Jersey 07974-0636 (US)

(72) Inventors:
- **Bendinelli, Samuel J.**
  Montgomery, New Jersey 08540 (US)
- **Krishnaswamy, Venkatesh**
  Holmdel, New Jersey 07733 (US)

(74) Representative:
Watts, Christopher Malcolm Kelway, Dr. et al
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex, IG8 0TU (GB)

(54) Synchronized presentation of television programming and internet content

(57) Uniform Resource Locators (URLs) or other network information identifiers are transmitted with television signals in order to permit web content to be displayed in synchronization with television programming. In an illustrative embodiment, URLs are embedded in a closed caption portion of a transmitted television signal, and delimited from the closed caption text using predetermined delimiting characters. A decoder extracts the URLs from the television signal, and supplies the URLs to a retrieval device which automatically retrieves corresponding web pages or other similar information over a network. The retrieved web pages are then displayed to a viewer in synchronization with related programming in the television signal. The retrieval device may be a set-top box associated with a television set that displays both a retrieved web page and the corresponding television picture portion of the television signal. Alternatively, the retrieval device may be a computer which retrieves and displays a web page, while the corresponding television picture is displayed on a television set.

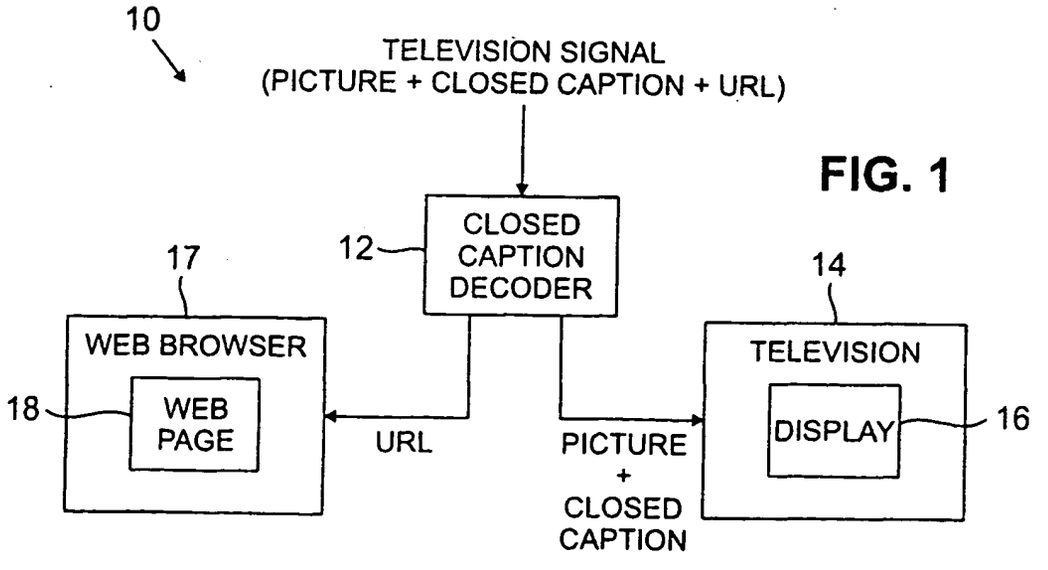


FIG. 1

EP 0 915 621 A2

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

[0001] The present invention relates generally to techniques for integrating television and computer systems, and more particularly to techniques for presenting World Wide Web content or other computer network information to viewers in conjunction with television programming.

Background of the Invention

[0002] It is not uncommon for television programming transmitted to viewers to include Internet web addresses or other information that the viewers can use to access additional information related to the programming. For example, television advertisements often include Uniform Resource Locators (URLs) which identify the web site of the advertising company. Similarly, television news programs and televised events often display URLs which identify web sites corresponding to the respective programs and events. Under current practice, a viewer desiring to access a URL displayed in a television program must either write down the URL or attempt to remember it. The viewer then must enter the URL in a browser program on their personal computer in order to access the corresponding web site. This can be inconvenient and frustrating for the viewer.

[0003] Although systems have been developed recently which allow viewers to access web sites from a television using a television set-top box, such systems typically operate in a manner similar to web browsers on personal computers. As a result, such systems are generally unable to extract a URL from a televised program such that the viewer need not reenter it to access the corresponding web site. Furthermore, such systems are unable to present web content to a viewer in synchronization with television programming which includes the corresponding URLs. For example, even a viewer which has a computer and a television in close proximity must continually enter the televised URLs in order to view a sequence of web content related to the television programming.

Summary of the Invention

[0004] The invention utilizes URLs or other network information identifiers transmitted with television signals in order to permit web content to be displayed in synchronization with related television programming. In an illustrative embodiment, URLs are embedded in a closed caption portion of a transmitted television signal, and delimited from the closed caption text using predetermined delimiting characters. A decoder extracts the URLs from the television signal, and supplies the URLs to a retrieval device which automatically retrieves corresponding web pages or other similar information over a network. The picture portion of the television signal is supplied to a conventional television set, and the retrieved web pages are displayed to a viewer in synchronization with the television programming. A viewer therefore no longer needs to manually enter televised URLs into a web browser or similar network access program. Instead, web content corresponding to the embedded URLs is automatically retrieved and displayed to the viewer as the television programming progresses. For example, as the television programming moves from one advertisement or program to another, the embedded URL will change, such that a new web site is accessed and the corresponding web page displayed to the viewer when the new advertisement or program is displayed on the television set. If a given displayed web page is of particular interest to a viewer, the viewer can select that web page for continuous display, in which case the synchronized presentation of web content is disabled until the viewer elects to resume it.

[0005] In one embodiment of the invention, the retrieval device may be a set-top box associated with the television set, such that the television set displays both a retrieved web page and the television picture portion of the television signal. The set-top box, which may include the above-noted decoder, establishes a connection over the network with a server that delivers a web page for display on the television set with the picture portion of the television signal. The retrieved web page may be displayed on the television set simultaneously with the picture portion of the television signal, using a picture-in-picture (PIP) type arrangement, or a viewer may be permitted to select between display of a retrieved web page and display of the picture portion of the television signal. In another embodiment of the invention, the retrieval device may be a computer which retrieves and displays a web page specified by an embedded URL, while the corresponding picture portion of the television signal is displayed on the television set. In yet another embodiment of the invention, the retrieval device may be a television tuner card within a computer. A television picture generated by the tuner card is then displayed in one window on the computer monitor, while a web page specified by a URL embedded in the television signal is displayed in another window on the monitor.

Brief Description of the Drawings

[0006]

FIG. 1 illustrates an exemplary embodiment of the invention in which a closed caption decoder is used to extract a Uniform Resource Locator (URL) or other network information identifier transmitted in a closed caption portion of a television signal;

FIG. 2 is a flow diagram illustrating a URL detection and display process in accordance with the invention;

FIG. 3 illustrates an embodiment of the invention in which a set-top box is used to display web content on a television display based on a URL or other network information identifier extracted from a television signal; and

FIG. 4 illustrates an embodiment of the invention in which a computer is used to display web content based on a URL or other network information identifier extracted from a television signal.

Detailed Description of the Invention

[0007] The invention will be described below in conjunction with a number of illustrative embodiments for displaying web content in conjunction with related television programming. The invention is not limited to use in these embodiments, but is instead more generally applicable to any system in which it is desirable to transmit, in conjunction with television programming, Uniform Resource Locators (URLs) or other identifiers which specify web sites or other sources of computer network information. The term "network information identifier" as used herein should be understood to include URLs as well as any other type of network information identifier. The term "web" as used herein is intended to include not only the World Wide Web portion of the Internet, but also other portions of the Internet as well as other types of computer networks including intranets and extranets. The term "web site" should be understood to include any source of information which can be specified by a URL or other type of network information identifier. The term "television signal" as used herein is intended to include analog video signals in accordance with NTSC, PAL or other standard formats, digital video signals including MPEG-encoded signals, high definition television (HDTV) signals, and any other type of image signal which may be supplied to a receiver via broadcast airwaves, cable network, satellite network, computer network or any other type of signal distribution mechanism. The term "automatically" as used herein in conjunction with retrieval of information over a network includes the retrieval of such information without the need for further intervention from a viewer.

[0008] FIG. 1 shows a system 10 for implementing presentation of web content in conjunction with related television programming in accordance with the invention. The system 10 includes a closed caption decoder 12 and a television 14. A television signal transmitted from a broadcast transmitter includes a video or "picture" portion, a closed caption text stream, and one or more URLs. The URLs are transmitted so as to coincide with the television programming represented by the picture portion of the signal. For example, when the programming corresponds to an advertisement, the corresponding transmitted URL may identify a web site of the company or product being advertised. Similarly, news programming may be transmitted with URLs identifying one or more web sites which describe various news items in greater detail, and music video programming may be transmitted with URLs identifying web sites of the recording company, artist or studio associated with the music video. Numerous other relationships between the transmitted programming and URLs are of course possible. The television signal may be received in the system 10 in a conventional manner via an antenna, cable network, satellite receiver or other type of conventional signal reception device, and is applied to an input of the closed caption decoder 12 as shown.

[0009] The URLs may be transmitted as strings embedded in the closed caption text stream of the television signal. A given URL string transmitted in this manner may be delimited by a known character, or set of characters, that appears in the closed caption text stream before and after the URL string. For example, brackets or quotes may be used as delimiting characters for a URL string. The decoder 12 includes appropriate hardware or software logic functions for extracting URLs from the closed caption stream. In the exemplary system 10 of FIG. 1, the picture and closed caption text stream are supplied to the television 14 and displayed on the display 16 in a conventional manner. It should be noted that an extracted URL may also be supplied from the decoder 12 to the television 14, such that it is displayed to a viewer along with the programming to which it corresponds. Such a display of the URL on the television 14 will generally be in an area of the display 16 separate from that used for the closed caption text. It should also be emphasized that although the FIG. 1 embodiment illustrates the invention in conjunction with the use of closed caption text, the invention does not require the use of closed caption text.

[0010] The extracted URLs from the decoder 12 are applied as inputs to a web browser program 17. The web browser program 17 uses a given extracted URL to establish a connection over a network for accessing the web site identified by the extracted URL. A web page 18 associated with the web site identified by the extracted URL is then displayed by the web browser 17. As will be described in greater detail in conjunction with FIGS. 3 and 4 below, the web browser 17 may be implemented using a set-top box connected to the television 14, such that web page 18 is displayed in a portion of the television display 16, or may be implemented using a computer with an Internet connection or other suitable network connection, such that the web page 18 is displayed on a display screen of the computer. The web pages associated with the URLs extracted from the television signal may be displayed in synchronization with the corresponding television programming, such that the displayed web page changes each time a different URL is extracted from the closed caption text stream by decoder 12. As noted previously, the embodiment of FIG. 1 does not require the use of a closed caption decoder. Alternative embodiments may use any other type of decoder which is capable of extracting a URL from a transmitted television signal.

[0011] FIG. 2 is a flow diagram illustrating the operation of the system 10 of FIG. 1 in greater detail. Step 20 indicates that URLs are broadcast as strings embedded in a closed caption text stream associated with a television program. In step 22, the closed caption decoder 12 of FIG. 1 extracts a current URL from a television signal applied thereto. The current URL refers generally to the latest URL received by the decoder 12. In step 24, a determination is made as to whether or not the current URL extracted in step 22 corresponds to a "blank" URL. A blank URL in accordance with the invention is a specified URL or other suitable code which indicates that the associated URL will not be displayed or used to access a web site. When a blank URL is received, step 25 indicates that the URL-based part of the display, for example, the web page 18 displayed by the web browser program 17, is cleared, and the FIG. 2 process then waits until the next URL is received. As noted previously, the URL itself may also be displayed along with the television picture on display 16 of television 14. In such an implementation, clearing the URL-based part of the display will also clear the URL from the television display. Upon receipt of the next URL, the process will return to step 22 to extract that URL as the current URL.

[0012] If the URL extracted in step 22 is not a blank URL, that URL is used to access a web page which is displayed in step 26. The web page corresponding to the current URL may be displayed continuously until the next URL is received in the decoder 12. In step 27, a determination is made as to whether the next URL has been received. If the next URL has not yet been received, the display based on the previously-extracted URL continues in step 26 until the next URL is received. If the next URL has been received, a determination is made in step 28 as to whether the viewer has selected the current URL displayed in step 26 for continued display. The viewer may select this continued display mode for a given displayed URL if that URL is of particular interest to the viewer. This allows the viewer to turn off the ongoing display of transmitted URLs in order to focus on the URL of particular interest.

[0013] If the viewer has selected the current displayed URL for continued display, step 29 indicates that this selected URL, as well as its associated web content, is displayed to the viewer until the viewer elects to resume display of additional URLs in synchronization with the displayed television programming. If it is determined in step 28 that the viewer has not selected the current URL displayed in step 26 for continued display, the process returns to step 22. The decoder 12 then extracts the next URL from the closed caption text stream this extracted URL becomes the current URL, and the displayed web content is updated in accordance with steps 24, 25 and 26. A viewer is thus able to view web content related to television programming on the television 14 without ever having to enter a URL into a web browser program. As the television programming progresses, the transmitted URLs change, and the displayed web content also changes accordingly, without any input required from the viewer. As previously noted, a viewer can elect to continue the display of any transmitted URL and its corresponding web content if that URL is of particular interest to the viewer.

[0014] FIGS. 3 and 4 illustrate two exemplary implementations of a system such as that described in conjunction with FIGS. 1 and 2. FIG. 3 shows a system 30 in which a broadcast transmitter 32 embeds URLs in television signals transmitted to viewers. As noted above, such a television signal may be received via an antenna, a cable network, a satellite receiver or other suitable receiving equipment. The set-top box 34 receives the television signal and the URLs embedded therein, and utilizes an internal decoder to extract the embedded URLs. This internal decoder may, but need not, be a closed caption decoder such as decoder 12. Any decoder suitable for detecting a transmitted URL or other transmitted network information identifier may be used. The set-top box 34 in this embodiment includes an internal telephone line modem, a cable modem or other similar device suitable for establishing a network connection with a network 40. The network 40 may be, for example, the Internet or any other computer, cable or satellite network as well as various portions or combinations thereof. The set-top box 34 may also implement a web browser program or other suitable network access software. The set-top box 34 uses the web browser program and the internal modem to establish a connection via network 40 to a server 42 corresponding to a given extracted URL. The server 42 then delivers a web page specified by the extracted URL to the set-top box 34 over the network 40.

[0015] The set-top box 34 delivers the picture portion of the television signal to a television 36, for display in a main area of the television display 38. The set-top box 34 also delivers the extracted URL, and the corresponding web page retrieved from server 42, to the television 36. In this embodiment, the television display 38 includes a picture-in-picture (PIP) area 44 which is used to display the retrieved web page simultaneously with the picture portion. The set-top box 34 or television 36 may also be configured to allow a viewer to switch between a variety of different types of displays, such as a display of only the picture portion, a display of only the web page, or a display in which the picture portion is in the PIP area 44 and the retrieved web page is in a main area of the display. Numerous other types of displays are also possible. The web browser software implemented by set-top box 34 may be configured to allow a viewer to visit a variety of web sites other than that corresponding to the current URL. For example, the set-top box 34 may be configured to store a list of recently-extracted URLs, such that a viewer can select display of a given web page from either the current URL or from some other recently-extracted URL. Viewer commands and other control information may be entered in a conventional manner using a remote control, keyboard,

mouse or similar input device.

[0016] FIG. 4 shows a system 50 in which URLs are embedded in a television signal transmitted from broadcast transmitter 32 as previously described. A decoder 52 receives the television signal, extracts an embedded URL, and delivers it over a serial connection or other suitable connection to a computer 54. The computer 54 runs a web browser program which uses the extracted URL from the decoder 52 to establish a connection with server 42 over network 40 in a manner similar to that described in conjunction with FIG. 3. The server 42 delivers a web page specified by the extracted URL to the computer 54, and the web page is displayed on a display 56 of the computer 54. The decoder 52 supplies the picture portion of the television signal along with the extracted URL to the television 36. The television display 38 displays the picture portion of the television signal as well as the extracted URL. A viewer using system 50 can thus view programming on the television 36 while simultaneously viewing related web content on the computer 54. As in the previous embodiments, the displayed web content changes in synchronization with the URLs transmitted in the television signal.

[0017] Alternative embodiments of the invention may utilize different mechanisms for delivering URLs or other network information identifiers in conjunction with related television programming. For example, although the illustrative embodiment of FIGS. 1 and 2 use URLs embedded in a closed caption text stream, other embodiments of the invention could incorporate the URLs into another portion of a transmitted television signal, such as onto a subcarrier or other signal associated with the television signal. It is also possible to transmit the URLs on another signal which is only roughly synchronized with the television signal. The URLs therefore need not be embedded in the transmitted television signal itself. Moreover, the URLs could be encoded or otherwise processed in any suitable manner prior to transmission rather than transmitted in a manner similar to that used to transmit closed-caption text. Furthermore, a variety of configurations other than those described in FIGS. 1 through 4 may be used for extracting URLs and displaying corresponding web content in accordance with the invention. For example, the invention may be implemented using a computer which includes a television tuner card, such that the television picture generated by the tuner card is displayed in one window on the computer monitor, while a web page specified by a URL embedded in the television signal is displayed in another window of the monitor. These and numerous other alternative embodiments within the scope of the following claims will be apparent to those skilled in the art.

Claims

1. A method of accessing information over a computer network, comprising the steps of:

   receiving at least one network information identifier transmitted with a television signal; and

   supplying the network information identifier to a device which utilizes the network information identifier to automatically retrieve information over the network, such that the retrieved information is displayed to a viewer in conjunction with related programming in the television signal.

2. The method of claim 1 wherein the network information identifier is a Uniform Resource Locator (URL) identifying a web site.

3. The method of claim 1 or claim 2 wherein the receiving step includes receiving the network information identifier in a closed caption portion of the television signal.

4. The method of claim 3 further including the step of decoding the closed caption portion of the television signal to recover the network information identifier.

5. The method of claim 3 wherein the network information identifier is embedded in a text stream of the closed caption portion of the television signal and delimited by one or more predetermined characters.

6. The method of any of the preceding claims wherein the step of supplying the network information identifier to a device which utilizes it to automatically retrieve information over the network further includes the step of supplying the network information identifier to a set-top box associated with a television set.

7. The method of claim 6 wherein the set-top box establishes a connection over the network with a server which delivers a web page for display on the television set along with a picture portion of the television signal.

8. The method of claim 6 or claim 7 wherein the set-top box includes a decoder for extracting the network information identifier from a closed caption portion of the television signal.

9. The method of any of claims 6 to 8 further including the step of displaying the retrieved information simultaneously with a picture portion of the television signal on the television set using a picture-in-picture arrangement.

10. The method of any of claims 6 to 8 further including the step of permitting a viewer to select between display of the retrieved information and display of the picture portion of the television signal.

11. The method of any of claims 1 to 5 wherein the step of supplying the network information identifier to a device which utilizes it to automatically retrieve information over the network further includes the step of supplying the network information identifier to a computer.

12. The method of claim 1 further including the step of displaying the retrieved information on a display screen of the computer.

13. The method of any of the preceding claims further including the step of preventing the retrieval and display of information corresponding to a particular network information identifier if the particular network information identifier has a predetermined form.

14. The method of any of the preceding claims further including the steps of storing a plurality of received network information identifiers, and allowing a viewer to select one or more of the plurality of identifiers for which corresponding retrieved information will be displayed in conjunction with the television signal.

15. The method of any of the preceding claims further including the step of allowing the viewer to select continued display of the retrieved information corresponding to a particular network information identifier.

16. The method of any of the preceding claims wherein said receiving step comprises receiving a sequence of said network information identifiers and said supplying step comprises supplying said identifiers to said device so that the retrieved information is displayed to said viewer in synchronization with said related programming.

17. An apparatus for accessing information over a computer network, comprising means arranged to carry out each step of a method as claimed in any of the preceding claims.

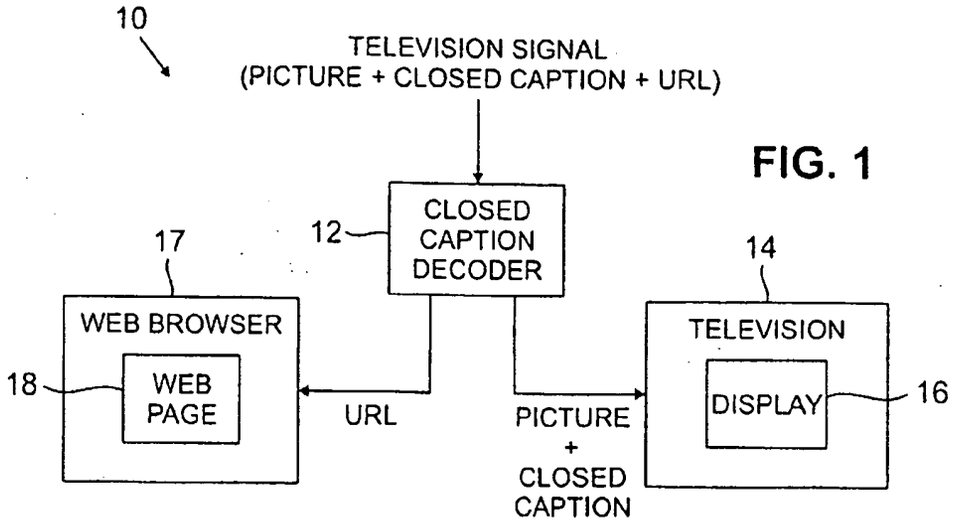

10
TELEVISION SIGNAL
(PICTURE + CLOSED CAPTION + URL)
12
CLOSED CAPTION DECODER
17
WEB BROWSER
18
WEB PAGE
URL
PICTURE + CLOSED CAPTION
14
TELEVISION
DISPLAY
16

FIG. 1

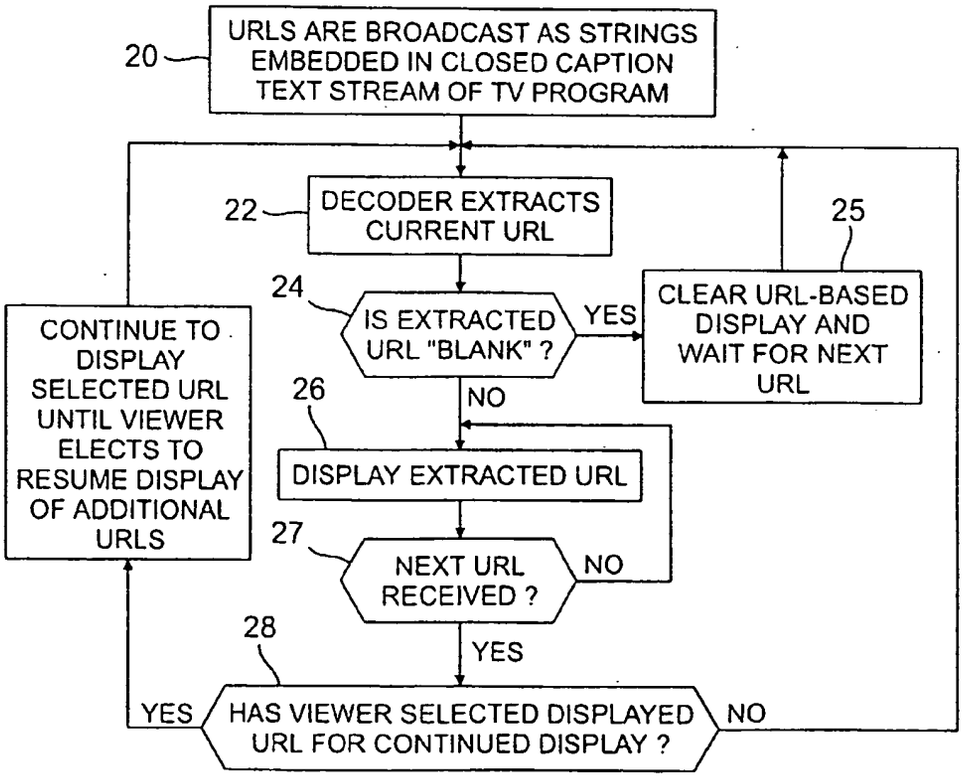

20
URLS ARE BROADCAST AS STRINGS EMBEDDED IN CLOSED CAPTION TEXT STREAM OF TV PROGRAM
22
DECODER EXTRACTS CURRENT URL
24
IS EXTRACTED URL "BLANK" ?
YES
CLEAR URL-BASED DISPLAY AND WAIT FOR NEXT URL
25
NO
26
DISPLAY EXTRACTED URL
27
NEXT URL RECEIVED ?
NO
YES
28
HAS VIEWER SELECTED DISPLAYED URL FOR CONTINUED DISPLAY ?
YES
NO
CONTINUE TO DISPLAY SELECTED URL UNTIL VIEWER ELECTS TO RESUME DISPLAY OF ADDITIONAL URLS

FIG. 2

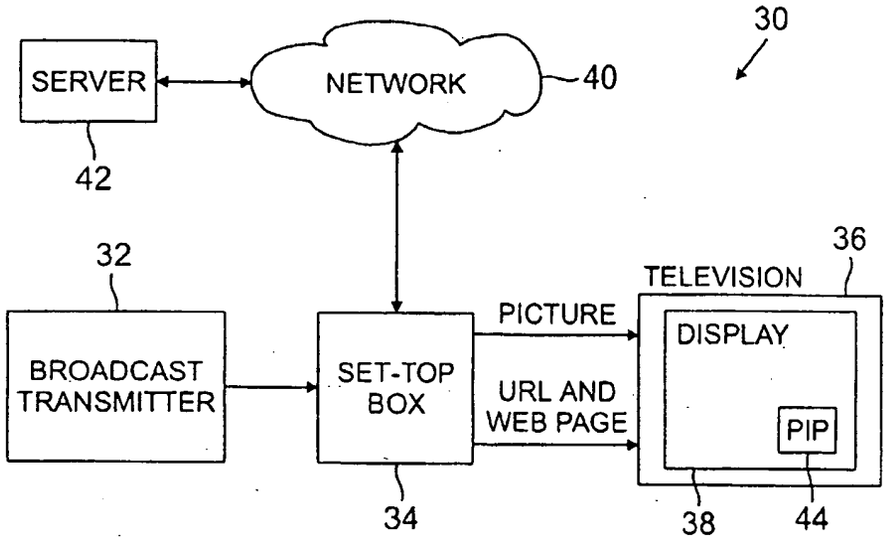

30
SERVER
NETWORK
40
42
32
36
TELEVISION
PICTURE
DISPLAY
BROADCAST TRANSMITTER
SET-TOP BOX
URL AND WEB PAGE
PIP
34
38
44

FIG. 3

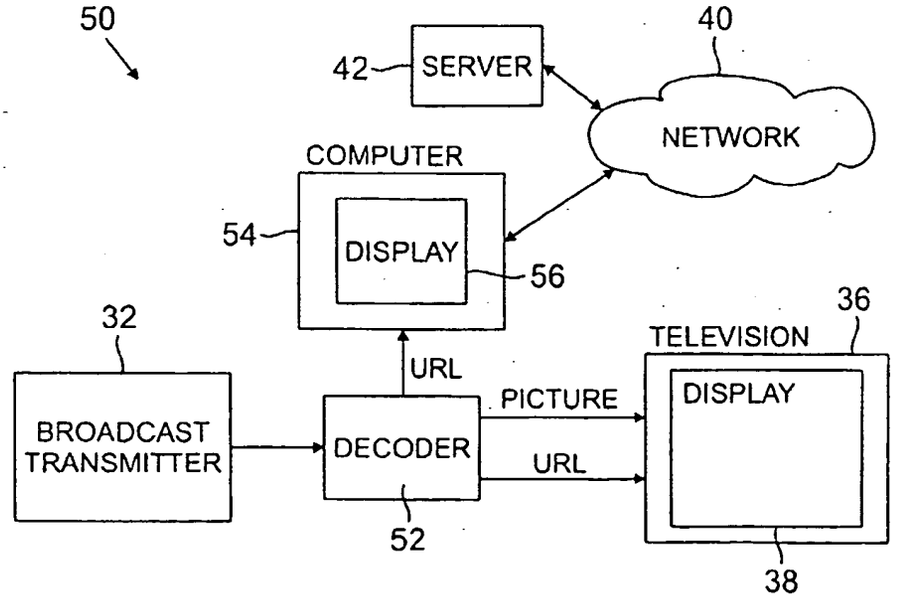

50
40
42
SERVER
NETWORK
COMPUTER
54
DISPLAY
56
32
36
TELEVISION
URL
PICTURE
DISPLAY
BROADCAST TRANSMITTER
DECODER
URL
52
38

FIG. 4

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 August 2001 (09.08.2001)

PCT

(10) International Publication Number
WO 01/58159 A1

(51) International Patent Classification[7]: H04N 7/16, 7/173

(21) International Application Number: PCT/US01/03591

(22) International Filing Date: 2 February 2001 (02.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/179,825 2 February 2000 (02.02.2000) US
09/754,650 3 January 2001 (03.01.2001) US

(71) Applicant: WINK COMMUNICATIONS, INC. [US/US]; 1001 Marina Village Parkway, Alameda, CA 94501 (US).

(72) Inventors: DEL SESTO, Eric, E.; 1133 Estates Drive, Lafayette, CA 94549 (US). MCCULLOUGH, Charles; 61 Agnes Street, Berkeley, CA 94618 (US). MICHEL, Christopher; 5399 Bryant Avenue, Oakland, CA 94618-1429 (US). TRAVAILLE, Timothy; 5810-134th Plse., Bellevue, CA 98006 (US).

(74) Agents: RAO, Dana, S. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

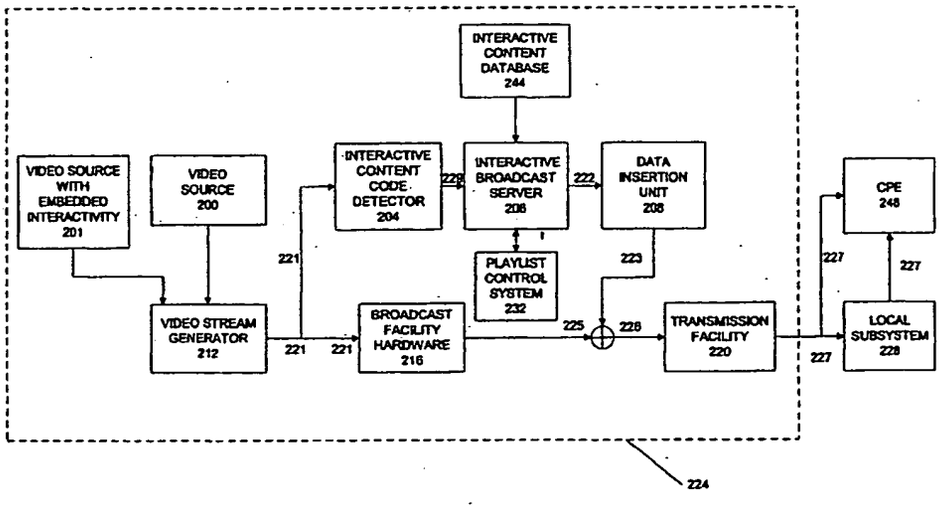
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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: ENSURING RELIABLE DELIVERY OF INTERACTIVE CONTENT



(57) Abstract: Interactive content preservation and customization technology is placed at the broadcast facility (224) to ensure reliable transmission of the interactive content to a local subsystem (228). An interactive content code detector (204) detects interactive content codes in the video stream at the broadcast facility (224). The interactive content code detector (204) is placed in the transmission path before the video stream is transmitted to broadcast facility hardware (216) that may strip out, destroy, corrupt or otherwise modify the interactive content and interactive content codes. Once an interactive content code detector (204) detects a code, an interactive broadcast server (206) determines what action to take, and instructs a data insertion unit (208) accordingly. The interactive content codes or interactive content may be placed in a portion of the video that is guaranteed by the broadcast facility (224) to be transmitted, for example, the closed caption region of the vertical blanking interval.

WO 01/58159 A1

ENSURING RELIABLE DELIVERY OF INTERACTIVE CONTENT

Cross-Reference to Related Application:

The present application claims priority from U.S. Provisional Patent Application S/N 60/179,825, entitled "Synchronizing the Delivery of Digital Information -- Interactive Applications, Data, and/or Content -- With Broadcast Television Programs and Advertisements", filed on February 2, 2000.

Field of the Invention

The present invention relates to the field of interactive television, and, more specifically, to synchronizing and customizing interactive content.

Background of the Invention

Delivery of interactive content with broadcast programming presents several special problems. First, for the interactive content to be effective, it must be synchronized with the broadcast. This entails delivering an interactive complement to an advertisement at the same time that the broadcast advertisement airs, or coordinating interactivity with events in the ad.

Second, the delivery of interactive content must be reliable. Since the interactive content is broadcast in a data stream parallel to the video stream, there is the possibility that the interactive content might be stripped out or corrupted at the broadcast site, or at the local subsystem, such as a local network affiliate or a cable headend. Finally, it is often desirable to target the interactive content to the specific viewer or delivery system. Digital cable systems, for example, have much higher bandwidth for delivering interactive content than the limited bandwidth provided by the vertical blanking interval (VBI) data transmission methods available to analog cable systems. It is advantageous to build a system that selectively utilizes this higher bandwidth when present in the broadcast environment. Additionally, demographic factors might motivate targeting different interactive content to different delivery systems.

A first order approach, as described in co-pending application 09/333,724, entitled "Automatic Control of Broadcast and Execution of Interactive Applications to Maintain

Synchronous Operation With Broadcast," filed June 15, 1999, assigned to the assignee of the present application and which is hereby incorporated by reference, involves the placement of an interactive broadcast server (IBS) at a broadcast facility to insert interactive content into a video stream. This server utilizes the network playlist system or other program identifying techniques in order to insert the appropriate interactive content for the current broadcast program or segment. A broadcast facility using this technology is illustrated in Figure 1.

The broadcast facility uses various equipment along a broadcast facility transmission path to generate, enhance, modify, and transmit a video stream. In one embodiment, a video tape player 146 generates a video stream 145 from video sources 112, and places the stream 145 on the transmission path. A server module (software application) running on the IBS (interactive broadcast server) 104 receives real-time information on the current video program being broadcast from a playlist control system 108. The playlist control system 108 controls the sequencing of the video sources 112, which are typically pre-recorded video content (such as programs and advertisements). The IBS 104 uses the playlist information to retrieve an interactive content and uses a data insertion unit 116 to insert a retrieved interactive content or data into the video stream 148. In some systems, the sequencing of pre-recorded video content is performed manually or program information is pre-recorded on the video and detected by the IBS 104 (for example, through VBI encoding of an interactive content identifier).

In an alternate embodiment of this interactive content delivery system, tapes are pre-prepared with interactive content and placed in the video source. A video tape player 146 plays the prepared tapes to generate a video stream with interactive content, and the interactive content is then delivered to the local subsystem 124. In this embodiment, the interactive content is pre-stored in a vertical blanking interval, for example, in VBI line 15, and an IBS 104 may not be used. In either embodiment, the combined video and interactive content stream 148 is then delivered to a communications system 120 (typically via satellite), to a local subsystem (such as a local network affiliate) 124 where the interactive content is then either delivered over the air through an antenna 149, to an interactive customer premise equipment (CPE) 128, i.e. a set top box or similar device, or over a cable system 149 that re-retransmits the local signal to a CPE 128.

While the above system can provide both synchronization and data reliability, some problems remain in ensuring that interactive content reaches its final destination due to broadcast facility or local subsystem equipment used along the transmission path. For example, many networks have multiple feeds to provide time shifted delivery of content for appropriate time zones. The equipment used by the broadcast facility to record and replay

video content for different time zones may inadvertently delete, corrupt, or otherwise effectively modify (corrupt) the inserted interactive content. Also, broadcast facilities may have the ability to visually shrink a broadcast signal to allow the broadcaster to insert their own video content, e.g., a stock ticker, into the broadcast signal. This insertion process may destroy the interactive content that is already encoded in the broadcast signal. Further, a broadcast facility may convert an analog broadcast signal to provide digital signals to subscribers. However, as the analog-to-digital conversion equipment is typically located downstream of the point where interactive content is added to the video stream, the conversion equipment may corrupt the interactive content encoded in the signal. In general, any device that modifies the video stream may corrupt or destroy interactive content.

Additionally, local subsystems 124 may also inadvertently corrupt inserted interactive content along a local subsystem transmission path, similar to the broadcast facility transmission path. Video manipulation in both paths can result in the loss of interactive content. For example, a local subsystem may choose to time shift a video stream by a half-hour to suit its own programming choices. In this situation, the equipment used to tape and delay the video stream may inadvertently corrupt the inserted interactive content or simply not re-record it. Also, a local subsystem 124 may insert their own content into a video stream, for example, identifying call letters. The equipment used to insert this content may inadvertently corrupt the inserted interactive content. Thus, a system is needed to ensure the delivery of the desired interactive content to end users.

SUMMARY OF INVENTION

This invention provides technology for ensuring reliable transmission of interactive content. This is achieved through the insertion of interactive content codes in the video signal at certain points in the path the video takes from the broadcast origination to the customer premise equipment. Servers collocated at the broadcasting facility bridge around the hardware that destroys the interactive content detect the interactive content codes and use the codes as triggers for the blocking, re-inserting, or substituting of interactive content in the transmission path.

Interactive Content Preservation at the Broadcast Facility

In a preferred embodiment, this technology is applied to solve the problem where interactive content is deleted, destroyed, corrupted, striped out or otherwise modified ("corrupted") during the transmission path of the broadcast facility due to processing of the video stream carrying the interactive content prior to transmission of the video stream, i.e. to a satellite or other transmission means. More specifically, interactive content codes preferably