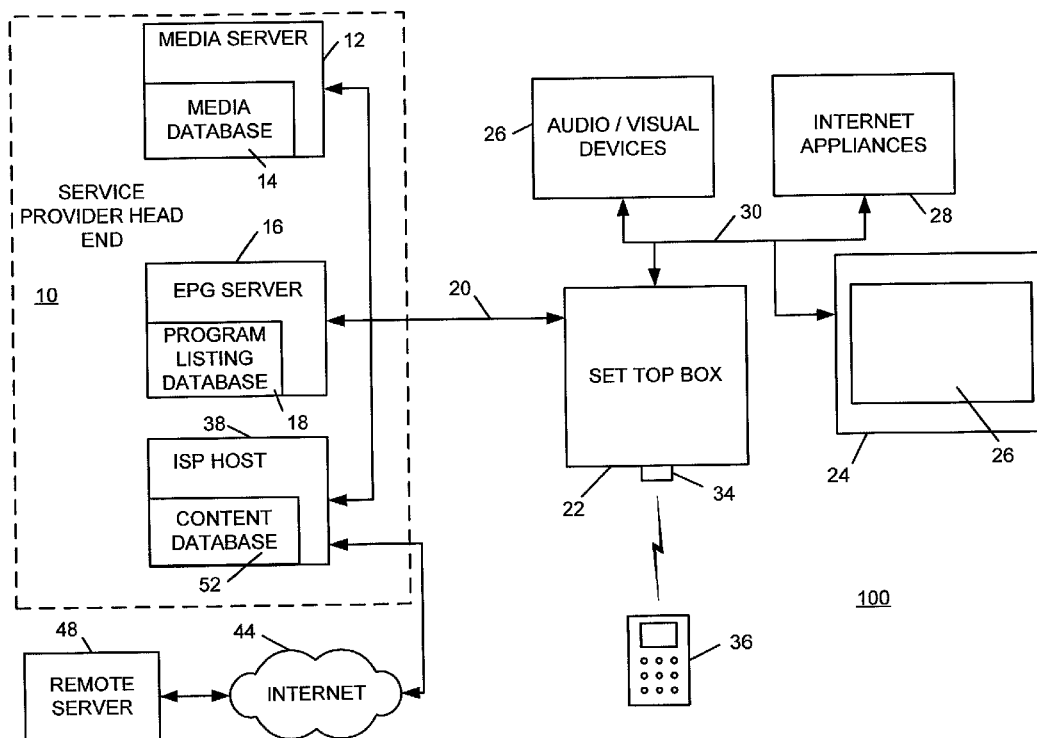


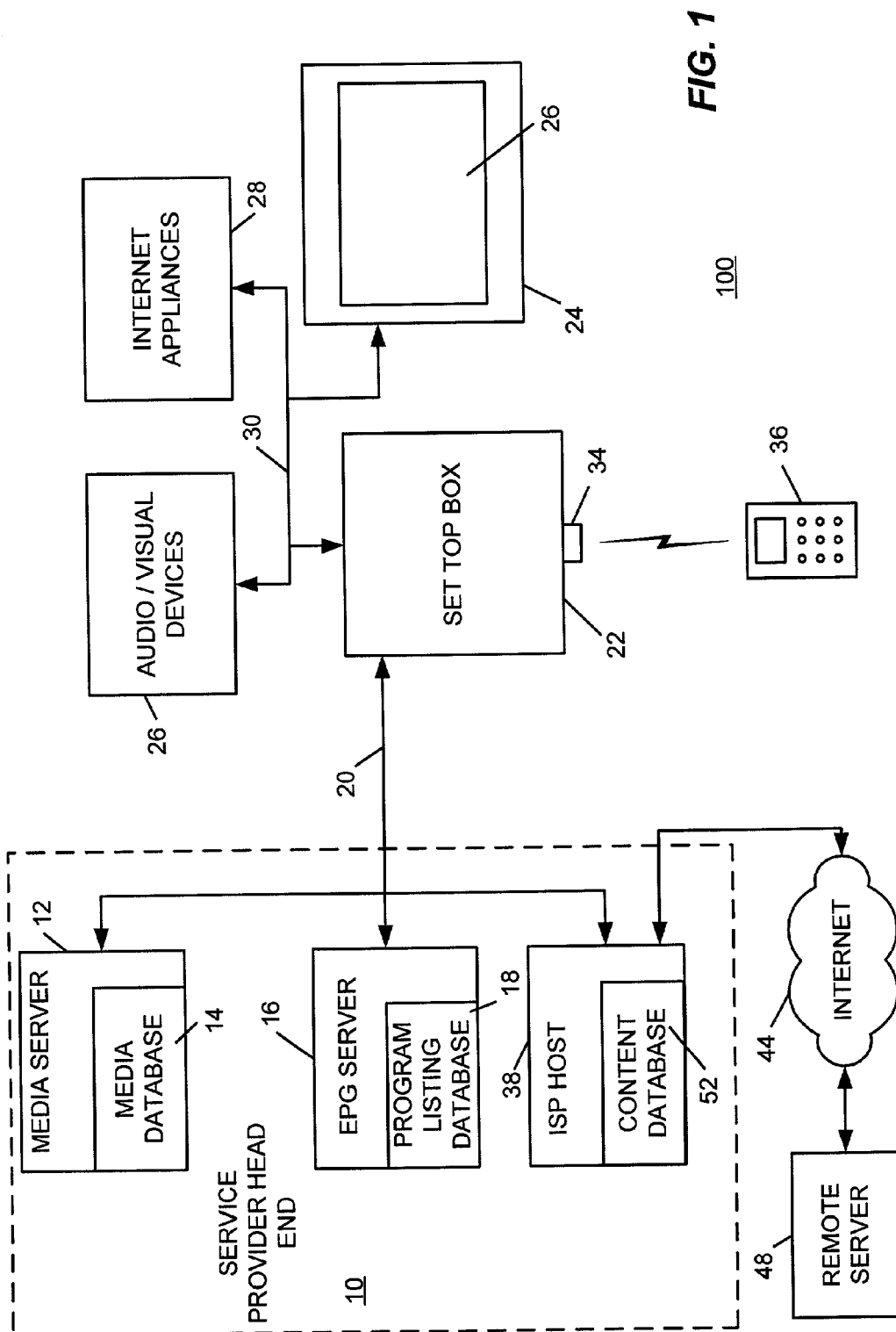
(19) **United States**(12) **Patent Application Publication**
Nguyen(10) **Pub. No.: US 2002/0145621 A1**(43) **Pub. Date: Oct. 10, 2002**(54) **WEB BROWSER AND SET TOP BOX
INTERFACE SYSTEM AND METHOD****Related U.S. Application Data**(63) Continuation-in-part of application No. 09/775,692,
filed on Feb. 2, 2001.(60) Provisional application No. 60/265,418, filed on Jan.
30, 2001.(76) Inventor: **Nga Marie Nguyen**, Carlsbad, CA (US)**Publication Classification**(51) **Int. Cl.⁷** **G09G 5/00**(52) **U.S. Cl.** **345/718; 345/738**

Correspondence Address:

Merle W. Richman, III**PO Box 3333****La Jolla, CA 92038-3333 (US)****ABSTRACT**

An interface for directing a web browser to load a web page having a desired video content description and generating a video window within the web page based on the desired video content description.

(21) Appl. No.: **10/060,644**(22) Filed: **Jan. 30, 2002**



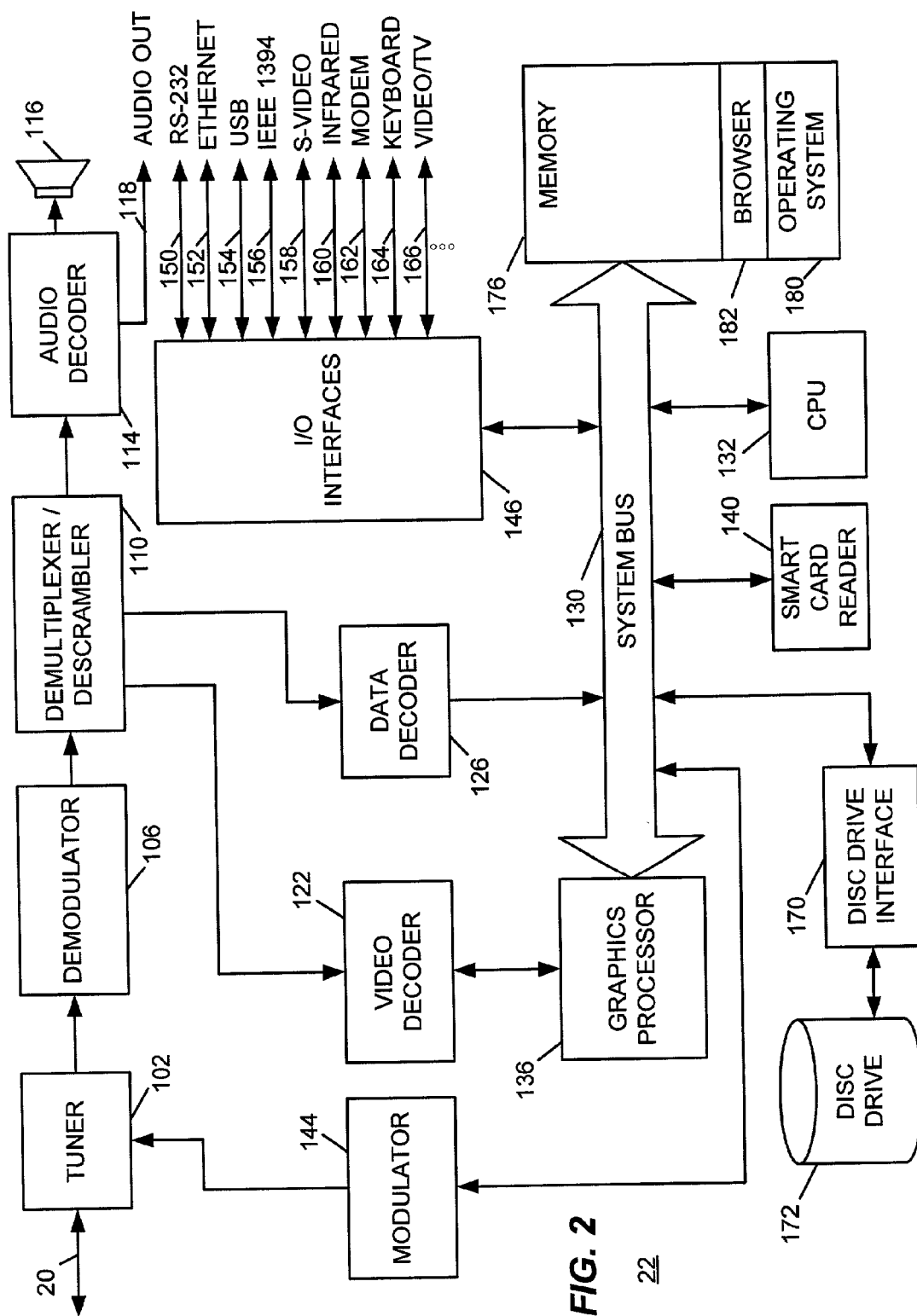


FIG. 2

22

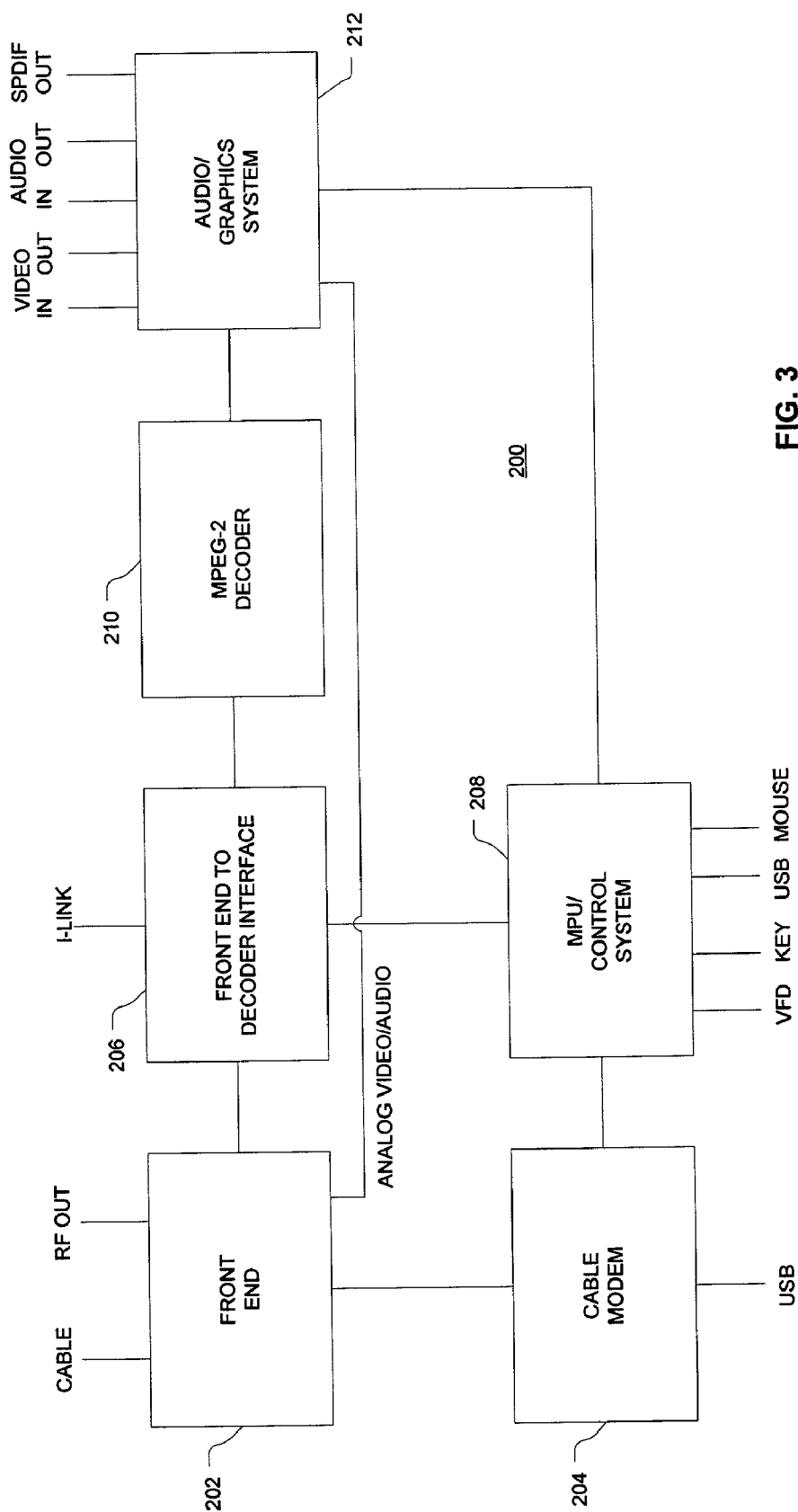


FIG. 3

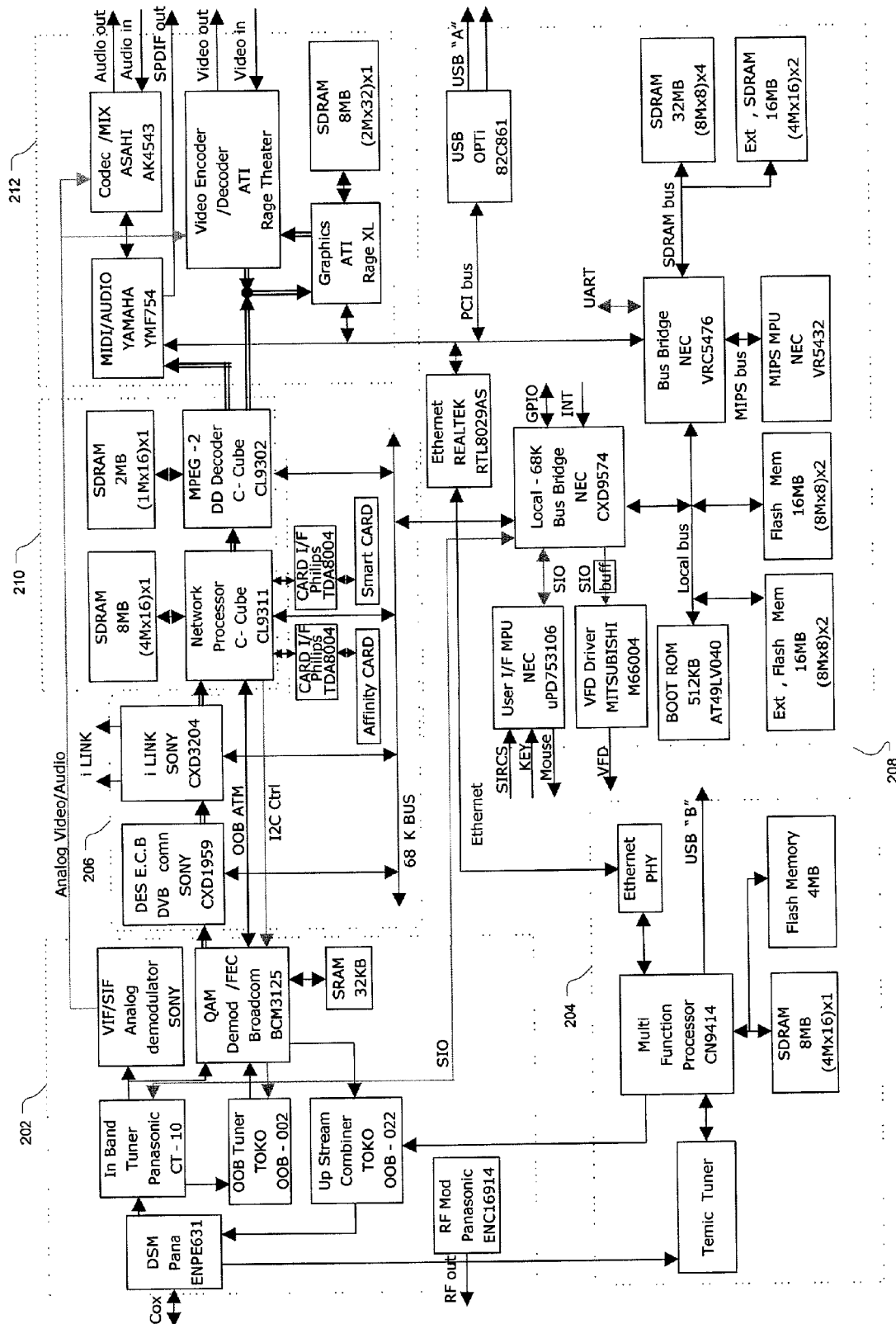


FIG. 4

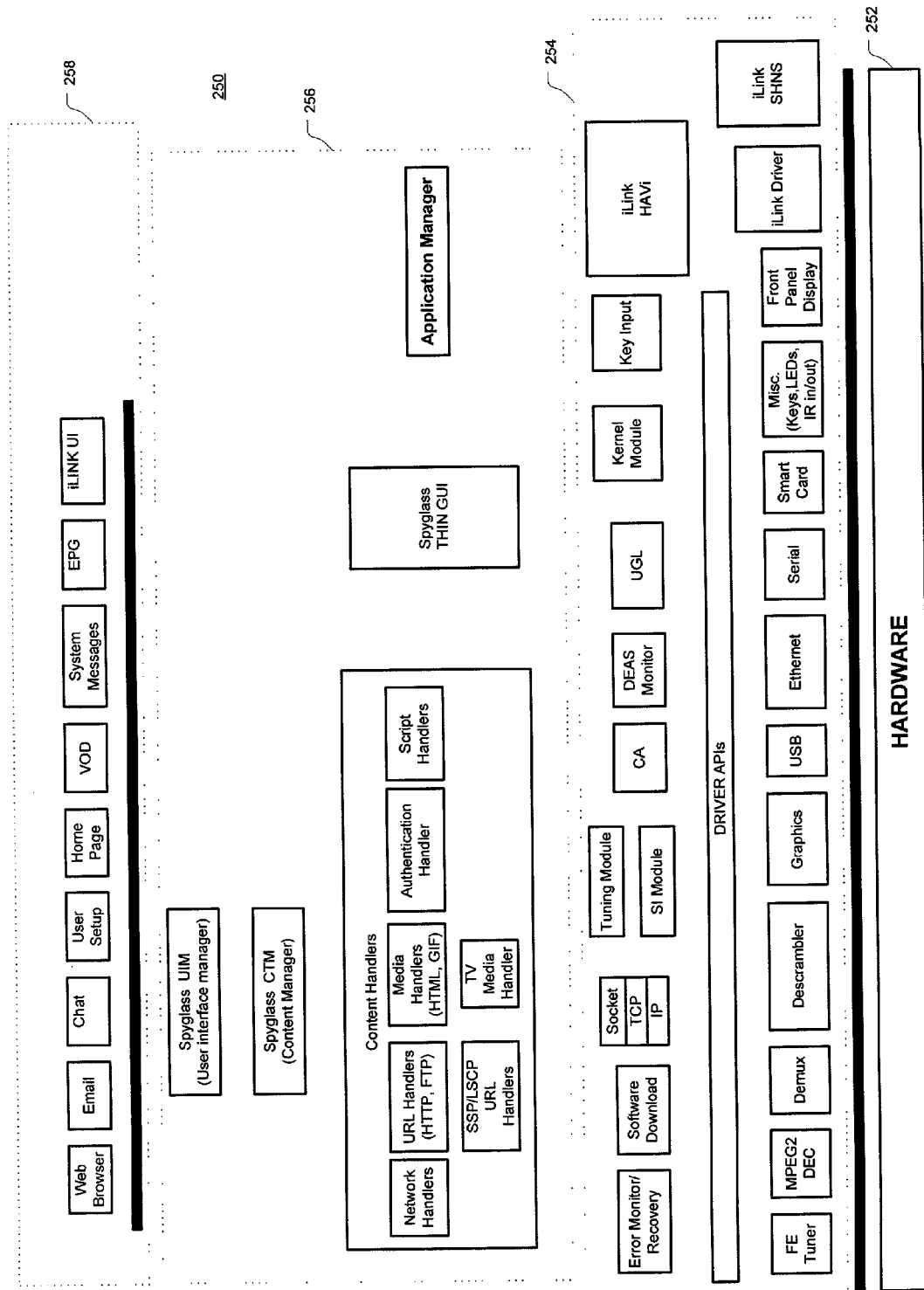


FIG. 5

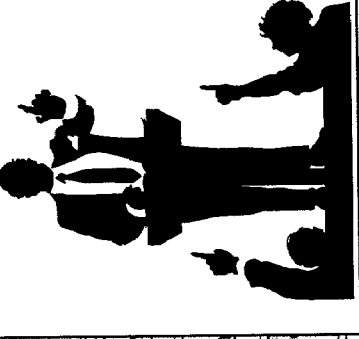
260		270	
	5:00PM		
ABC 10	10 News Live at 5		
BS 11	Arthur TVY, CC		
OOR012 12	< Local Origination		
JPN 13	Jenny Jones TVP CC		
NB 14	7th Heaven TVG, CC	Sabrina, the Teenag TVG, CC	
OOR015 15	< Local Origination		
OOR016 16	< Local Origination		

FIG. 6B

260		5:00PM	5:30PM	6:00PM
		10 News Live at 5 CC	Arthur TVY, CC	ABC News CC
ABC 10		Arthur TVY, CC	Nightly Business Re	
BS 11				
OOR012 12		< Local Origination		
JPN 13		Jenny Jones TVPG, Repeat, CC	Cosby TVG, CC	
NB 14		7th Heaven TVG, CC	Sabrina, the Teenag TVG, CC	
OOR015 15		< Local Origination		
OOR016 16		< Local Origination		

FIG. 6A

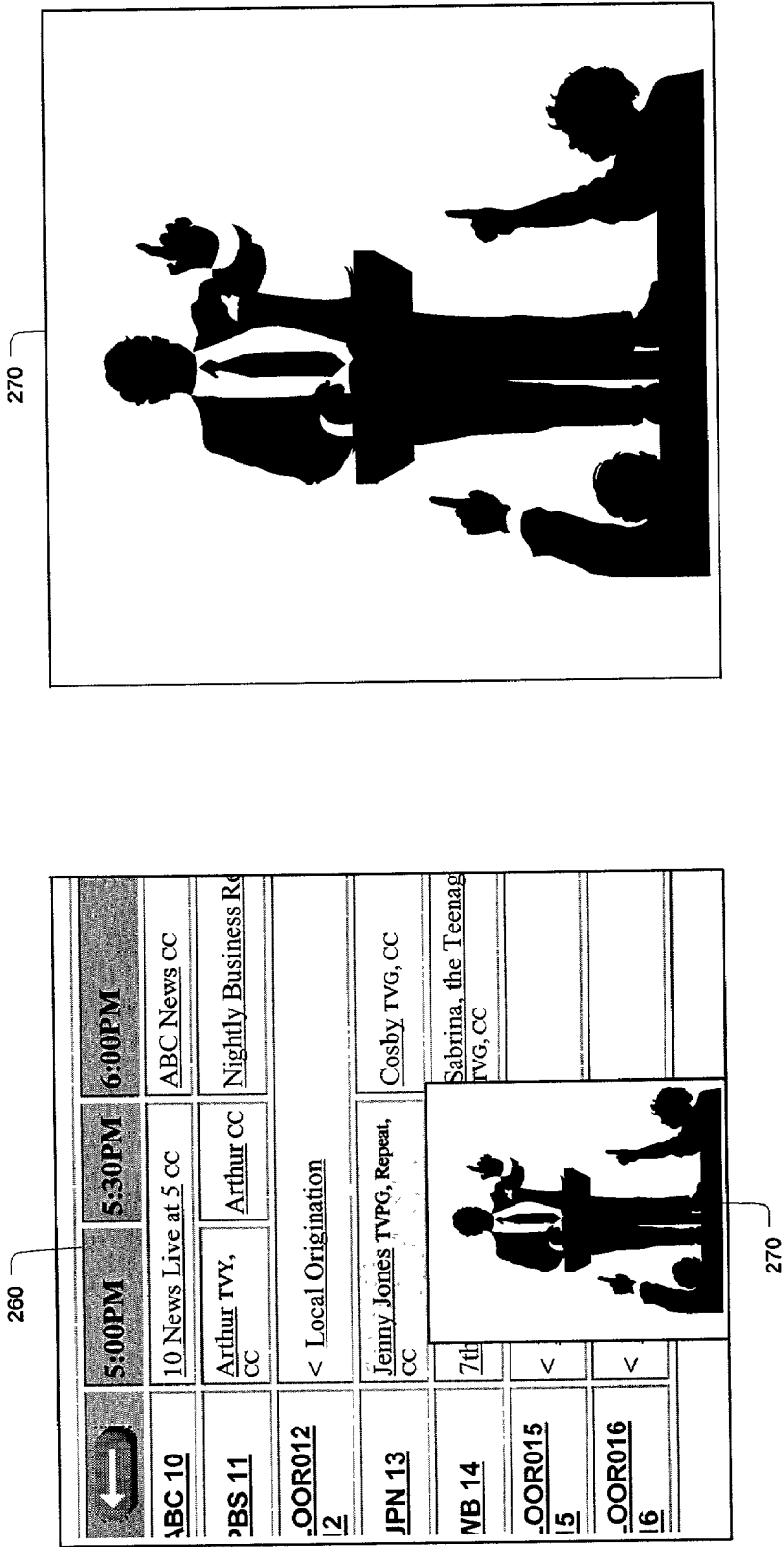


FIG. 6D

FIG. 6C

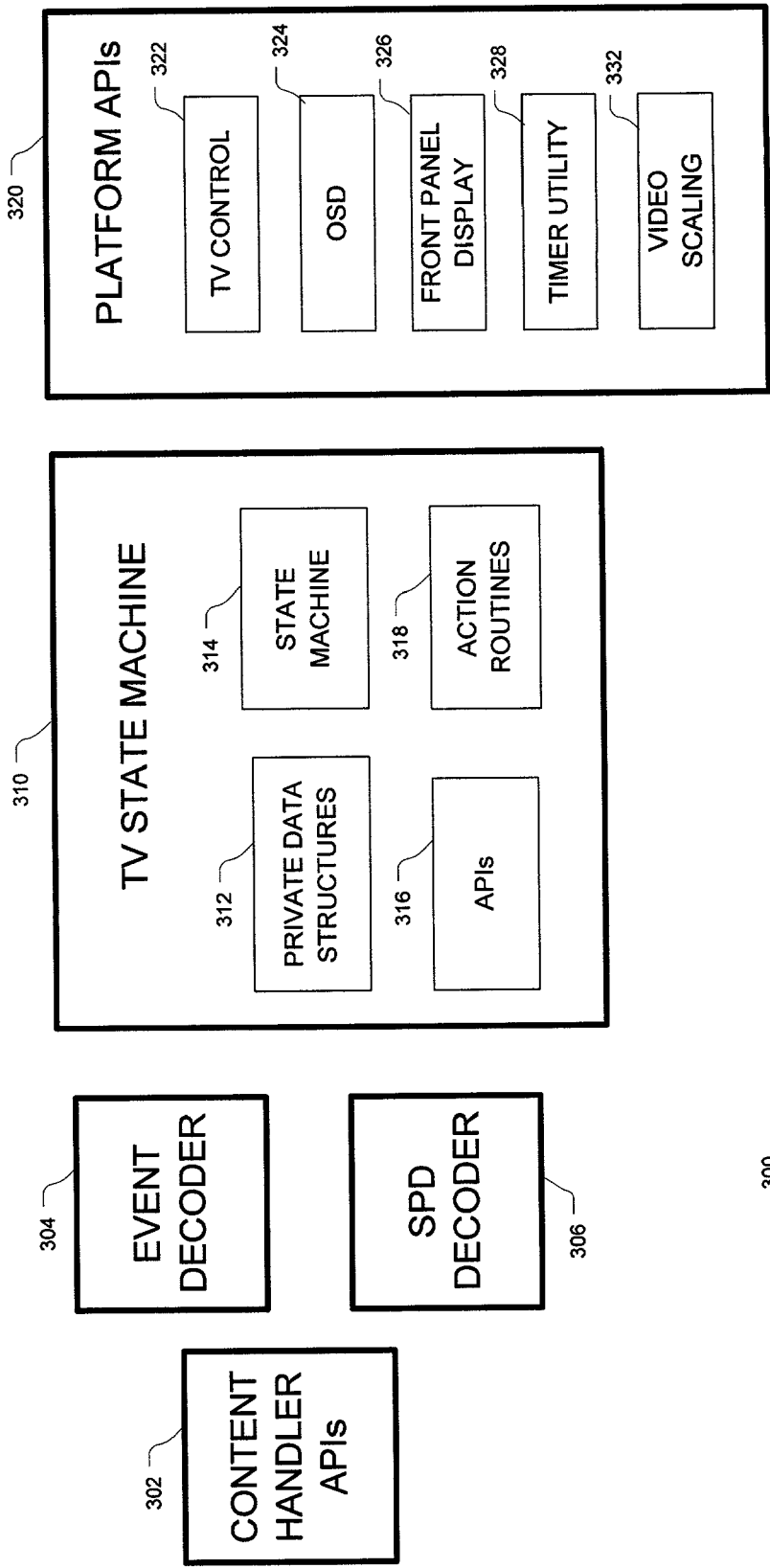


FIG. 7

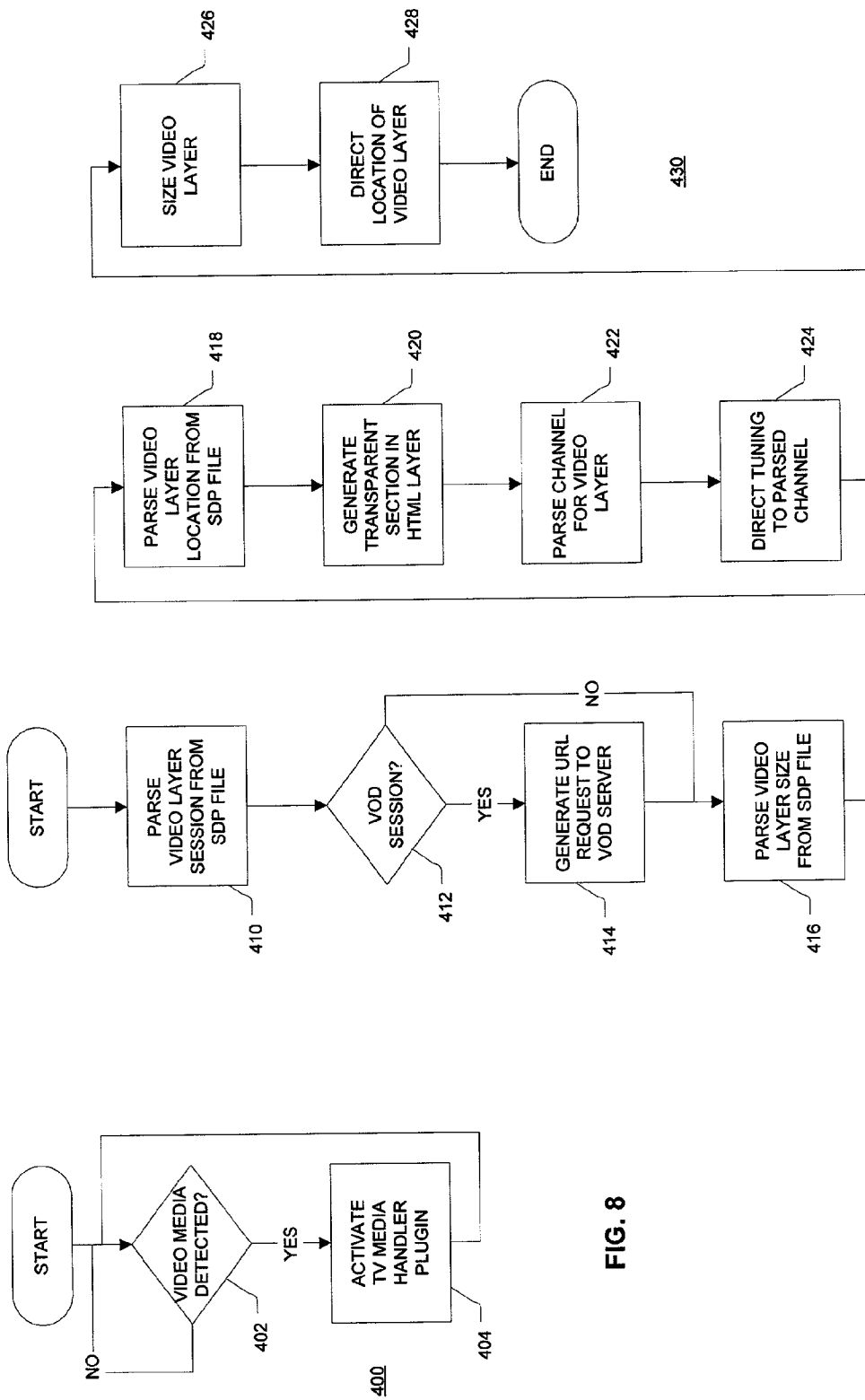


FIG. 9

FIG. 8

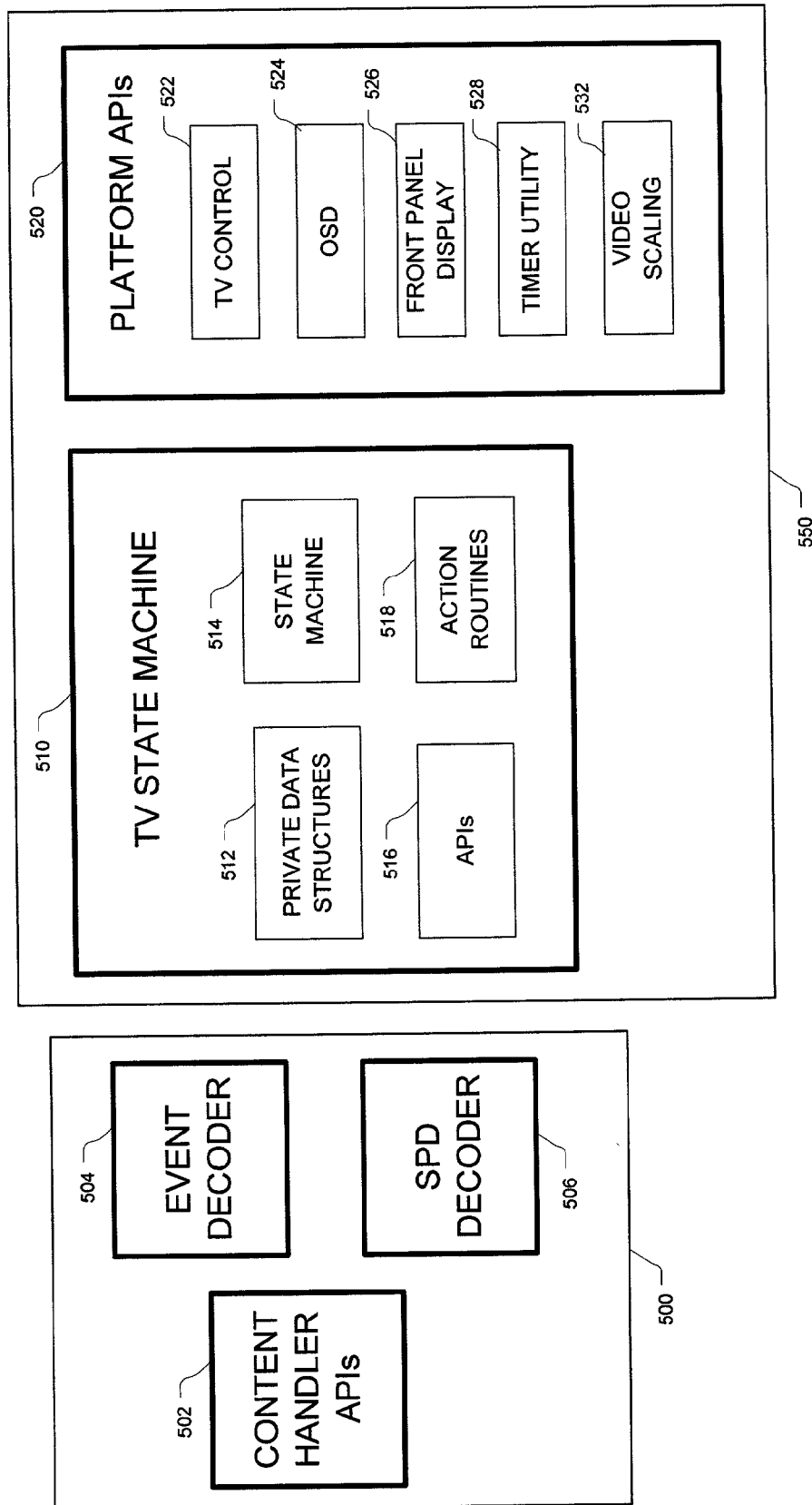


FIG. 10

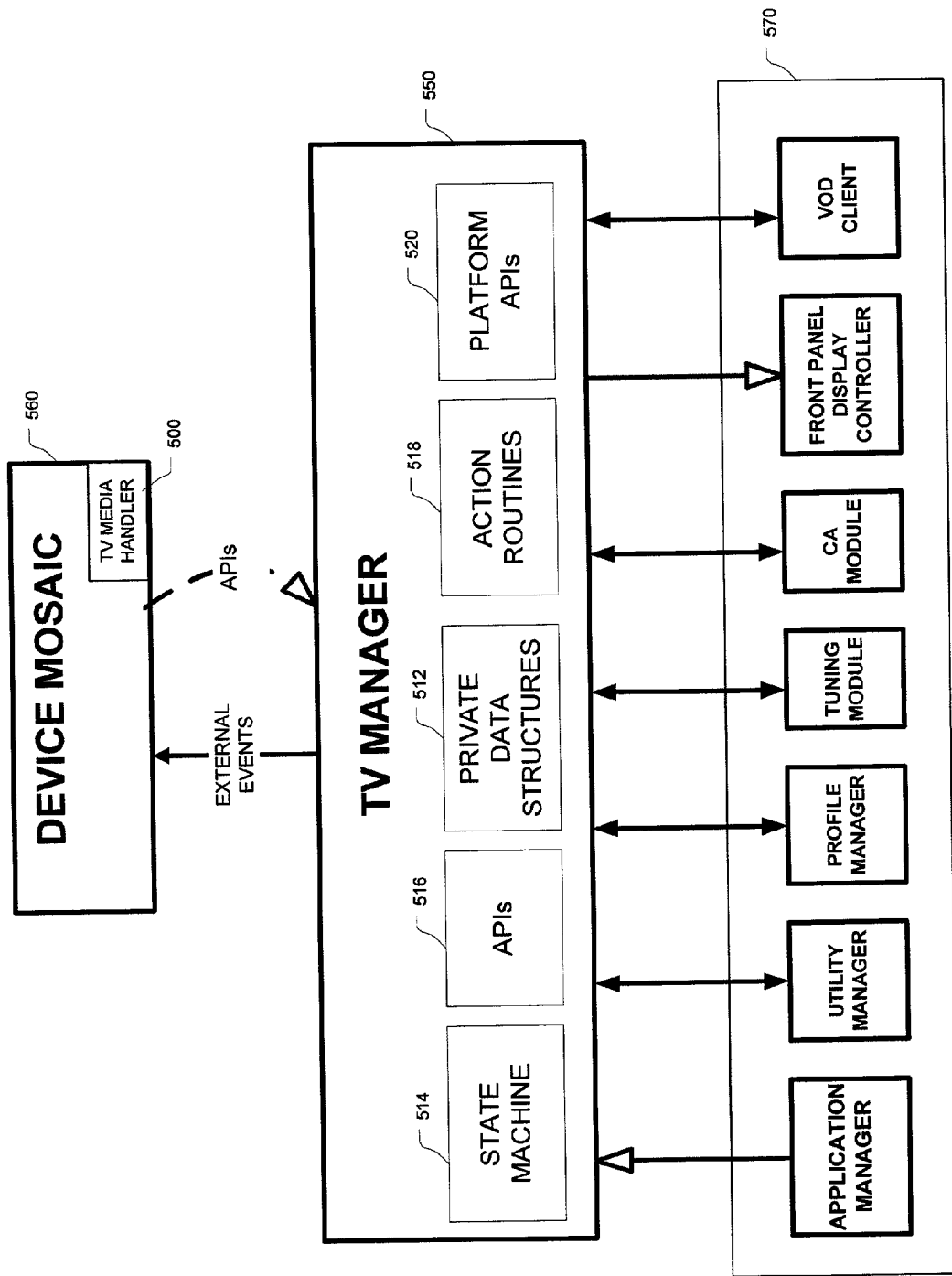


FIG. 11

580

600

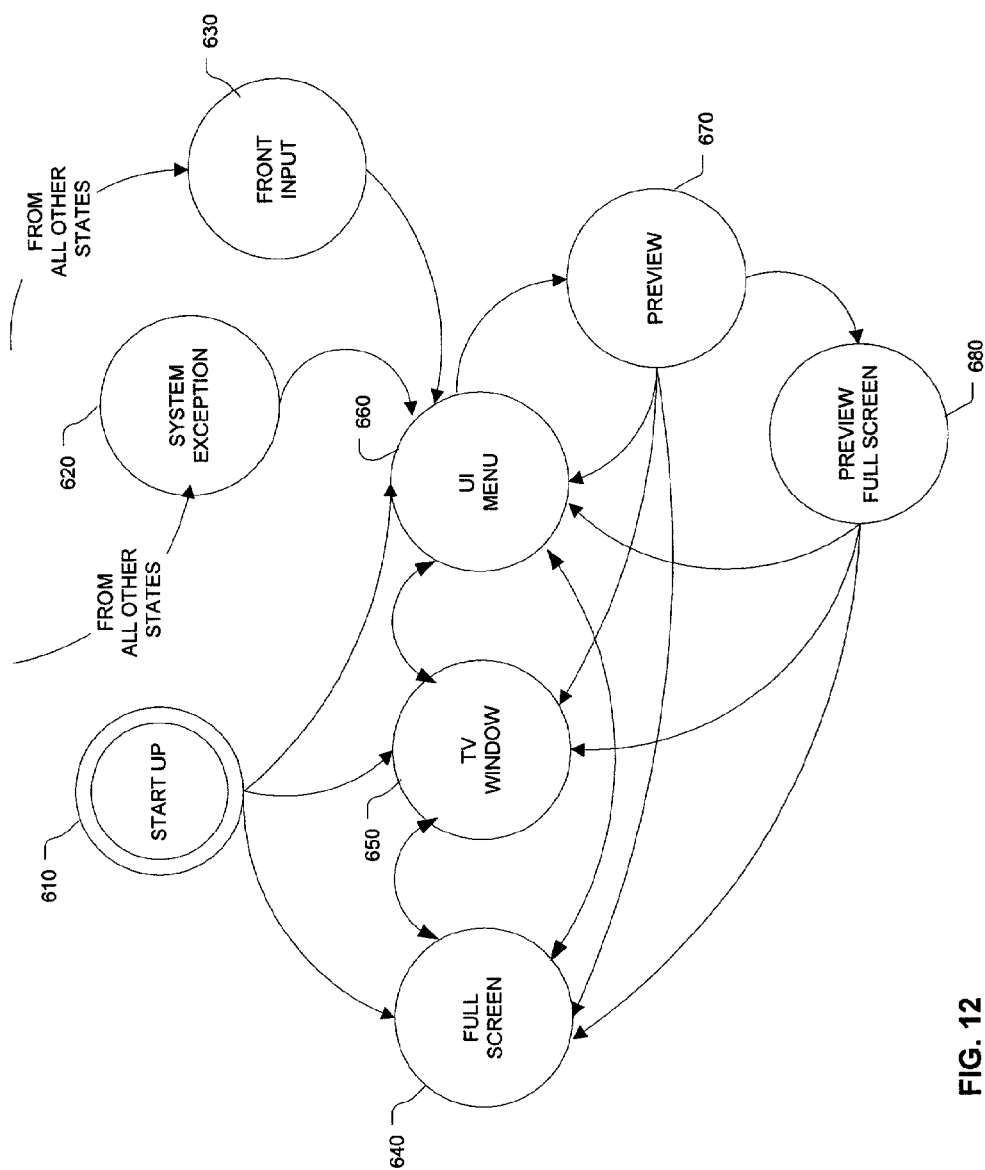


FIG. 12

WEB BROWSER AND SET TOP BOX INTERFACE SYSTEM AND METHOD

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation-in-part (“CIP”) of Utility patent application Ser. No. 09/775,692 filed Feb. 2, 2001, Attorney Docket Number 50N3463.02.01, and entitled “WEB Browser Plug-in for TV” and is related to Provisional Patent Application No. 60/265,418, filed Jan. 30, 2001, Attorney Docket Number SNY001V, and entitled “Web Browser and Set Top Box Interface System and Method”, each of which is hereby incorporated by reference for their teachings.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] This invention relates to interfacing a web browser within a TV set top box.

[0004] 2. Description of Related Art

[0005] As noted in the above referenced and incorporated CIP application in the United States a substantial majority of homes have at least one television (there are about 2.24 televisions (“TV”s) per household according to some sources). Video Cassette Recorders (“VCRs”) and Digital Video Disc (“DVD”) players enable users to selectively view and review video segments on a TV. The advent of digital video media content technologies, the personal computer (“PC”) market, and growing usage of a highly accessed computer network of networks (termed the Internet) has raised user’s expectation of video media content quality, availability, and features of viewing the same.

[0006] For these reasons systems have been developed to combine TV, VCR, DVD, computer, and Internet technologies. For example, TV manufacturers have developed TVs that include a VCR or DVD player. Cable and Satellite companies have introduced pay per view systems that allow a user to select and view limited selections of video content at fixed times. Web developers have created web sites on the Internet that provide interactive electronic program guides (“EPG”). In addition, DVD players and TV tuners have been incorporated into PCs to enable users to view video media content on a PC via specialized software. It is desirable to enable the integration of WEB information and video media with limited dependency on the web browser device mosaic.

SUMMARY OF THE INVENTION

[0007] The present invention includes an apparatus for and a method of displaying a desired video content frame within a WEB browser based content frame in a windowless environment. One embodiment includes directing the WEB browser to load a page having a desired video content frame definition and generating a transparent section in the browser based content frame based on the frame definition. Then the embodiment overlaps the video content frame in the transparent section of the browser based content frame.

[0008] In some embodiments the displayed size of the video content frame is smaller than the displayed size of the browser based content frame. Also, the video content may be related to the browser based content. In another embodi-

ment, the method may decode the desired video frame size from the frame definition and generate the transparent section in the browser based content frame based on the decoded frame definition. The method may also decode the video frame location within the browser frame from the frame definition.

[0009] The present invention also includes an apparatus for and a method of displaying a desired video content frame within a WEB browser based content frame. The invention includes directing the WEB browser to load a page having a desired video content frame definition and generating the desired video content frame definition in the browser based content frame based on the frame definition.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a block diagram digital cable television system in accordance with the present invention.

[0011] FIG. 2 is a block diagram of the set top box shown in FIG. 1.

[0012] FIG. 3 is a block diagram of a set top box according to an embodiment of the present invention.

[0013] FIG. 4 is a detailed block diagram of the set top box of FIG. 3.

[0014] FIG. 5 is a block diagram of the software architecture of the set top box of FIG. 4.

[0015] FIGS. 6A to 6D are diagrams of browser and video content screen configurations in accordance with the present invention.

[0016] FIG. 7 is a block diagram of the software architecture of the TV media handler of the set top box of FIG. 4.

[0017] FIG. 8 is a flowchart of a process of invoking the TV media handler in accordance with the present invention.

[0018] FIG. 9 is a flowchart of a TV media handler process in accordance with the present invention.

[0019] FIG. 10 is a block diagram of another embodiment of software architecture of a TV media handler and TV Manager/Interface of the set top box of FIG. 4.

[0020] FIG. 11 is a block diagram of another embodiment of software architecture of the set top box of FIG. 4.

[0021] FIG. 12 is a functional flow diagram for the TV Manager/Interface’s state machine of FIG. 10.

[0022] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0023] Throughout this description, the preferred embodiment and examples shown should be considered as exemplars, rather than as limitations on the present invention.

[0024] FIG. 1 is a block diagram for an exemplary interactive cable or satellite television (TV) system 100 in which the present invention may be employed. The system 100 includes a service provider head end 10, remote server 48, Internet 44, audio/visual devices 26, Internet appliances 28, television 24, set-top box (“STB”) 22, and remote control

36. The head end of the service provider **10** includes a media server **12**, EPG server **16**, and ISP Host **38**. The media server **12** of the head end **10** provides on demand movies and other programming such as interviews with actors, games, advertisements, available merchandise, associated Web pages, and other related content obtained from a media database **14**. The electronic programming guide (EPG) server **16** includes a program listing database **18** for generating an EPG. The ISP host **38** includes a content database **52** and is coupled to remote servers **48** via the Internet **44**. The remote servers may include another content such as video on demand ("VOD") content or EPG content. The EPG content received from the remote server **48** may be used to populate or update the program listing database **18** of the EPG server **16**. The ISP host **38** includes protocols that enable communication between remote servers **48** via the Internet **44**.

[0025] The media server **12** and EPG server **16** are coupled by a transmission medium **20** to the set top box (STB) **22**. The transmission medium **20** may include, for example, a conventional coaxial cable television network, a fiber optic cable network, telephone system, twisted pair, a satellite communication system, a radio frequency (RF) system, a microwave system, other wireless systems, a combination of wired and wireless systems or any of a variety of known electronic transmission mediums. In the case of a coaxial cable television network, transmission medium **20** is commonly realized at the subscriber's premises as a coaxial cable that is connected to a suitable cable connector at the rear panel of the STB **22**.

[0026] As noted, system **100** further includes a TV **24**, such as a digital television. The TV **24** includes a display **26** for displaying programming, an EPG, web browser and other content. The STB **22** may be coupled to the TV **24** and various other audio/visual devices **26** and Internet Appliances **28** by an appropriate interface **30**, which can be any suitable analog or digital interface including an Institute of Electrical and Electronics Engineers (IEEE) 1394 standard interface, S-Video, Component Video, NTSC, PAL, or other analog television interface.

[0027] Set-top box **22** can generally provide for bi-directional communication over a transmission medium **20** in the case of a cable STB **22**. In other embodiments, bi-directional communication can be effected using asymmetrical communication techniques possibly using dual communication media, one for the uplink and one for the downlink. In any event, the STB **22** can have its own Universal Resource Locator (URL) assigned thereto to provide for direct addressing by the head end and users of the Internet. In the case of a Direct Satellite System (DSS), the STB **22** is often referred to as an Integrated Receiver Decoder (IRD). The transmission medium is a satellite transmission at an appropriate microwave band. A satellite dish antenna with an integral Low Noise Block (LNB) is used to receive such transmissions. A down-converter converts the received signal to a lower frequency (baseband frequency) for processing by the STB **22**.

[0028] As shown in **FIG. 2**, the STB **22** may include a central processing unit (CPU) **132** and memory such as Random Access Memory (RAM) **176**, Read Only Memory (ROM), flash memory, mass storage such as a hard disc drive **172**, floppy disc drive, optical disc drive or may accommodate other electronic storage media. Such memory

and storage media is suitable for storing data as well as program instructions for processes to be executed by the CPU. Information and programs stored on the electronic storage media or memory may also be transported over any suitable transmission medium such as that illustrated as **20**. STB **22** may include circuitry suitable for audio decoding and processing **114**, the decoding of video data **122** compressed in accordance with a compression standard such as the Motion Pictures Experts Group (MPEG) standard and other processing. It is noted that these components may be incorporated into the TV **24**, eliminating the STB **22**. In addition, a computer may substitute the TV **24** and STB **22**. The computer may include a variety of devices capable of generating video media including a tuner card coupled to a digital network, cable television network, or DSS network.

[0029] It is noted that the STB **22** may be coupled to additional devices such as a personal computer, video cassette recorder, camcorder, digital camera, personal digital assistant and other audio/visual or Internet related devices (not shown). In addition, data transport architecture, such as that set forth by an industry group which includes Sony Corporation and known as the Home Audio-Video Interoperability ("HAVi") architecture may be utilized to enable interoperability among devices on a network regardless of the manufacturer of the device. This architecture may be used to create a home network system between electronic devices and Internet appliances. The STB **22** may run an operating system suitable for a home network system such as Sony Corporation's Aperiostm real time operating system. Other operating systems could also be used.

[0030] As shown in **FIG. 1**, the STB **22** includes an infrared (IR) receiver **34** for receiving IR signals from an input device such as the remote control **36**. Alternatively, it is noted that many other control communication methods may be utilized besides IR, such as wired or wireless radio frequency, etc. In addition, it can be readily appreciated that the input device **36** may be any device suitable for controlling the STB **22** such as a remote control, personal digital assistant, laptop computer, keyboard, or computer mouse. In addition, an input device in the form of a control panel located on the TV **24** or the STB **22** can be provided.

[0031] The STB **22** may also be coupled to an independent service provider (ISP) host **38** by a suitable connection including dial-up connections, DSL (Digital Subscriber Line) or the same transmission medium **20** described above (e.g. using a cable modem) to, thus, provide access to services and content from the ISP and the Internet. STB **22** may also be used as an Internet access device to obtain information and content from remote servers such as remote server **48** via the Internet **44** using host **38** operating as an Internet portal, for example. In certain satellite STB environments, the data can be downloaded at very high speed from a satellite link, with asymmetrical upload speed from the set-top box provided via a dial-up or DSL connection.

[0032] One configuration of a digital STB **22** is shown in detail in **FIG. 2**. The STB **22** includes a tuner **102**, demodulator **106**, demultiplexer/descrambler **110**, audio decoder **114**, modulator **144**, video decoder **122**, data decoder **126**, I/O interfaces **146**, system bus **130**, graphics processor **136**, memory **176**, central processing unit ("CPU") **132**, smart card reader **140**, disc drive interface **170**, and disc drive **172**. A transmission medium **20**, such as a coaxial cable, is

coupled by a suitable interface to the tuner **102**. Tuner **102** may include a broadcast in-band tuner for receiving content, an out-of-band (“OOB”) tuner for receiving data transmissions and a return path tuner for providing an OOB return path for outbound data (destined for example for the head end). A separate tuner (not shown) may be provided to receive conventional RF broadcast television channels. Demodulator **106** may demodulate any modulated information from the tuner **102** such as MPEG-2 formatted data. The demultiplexer/descrambler circuit **110** separates the demodulated information into discrete channels of programming. The programming is divided into packets, each packet bearing an identifier called a Packet ID (PID) that identifies the packet as containing a particular type of data (e.g. audio, video, and data). The demultiplexer/descrambler circuit **110** also decrypts encrypted information in accordance with a decryption algorithm to prevent unauthorized access to programming content, for example.

[0033] Audio packets from the circuit **110** (those identified with an audio PID) are decrypted and forwarded to an audio decoder **114**. The audio decoder **114** may convert the audio packets to analog audio to drive a speaker system (e.g. stereo or home theater multiple channel audio systems) or other audio system **116** (e.g. stereo or home theater multiple channel amplifier and speaker systems) or may simply provide decoded audio out at **118**. Video packets from the circuit **110** (those identified with a video PID) are decrypted and forwarded to the video decoder **122**. Similarly, data packets from the circuit **110** (those identified with a data PID) are decrypted and forwarded to the data decoder **126**.

[0034] The data decoder **126** transmits decoded data packets to the CPU **132** via the system bus **130**. Video decoder **122** passes video data to the graphics processor **136**. The graphics processor is a computer optimized to process graphics information rapidly, in particular graphics intensive data associated with Internet browsing, gaming, and multimedia applications such as those associated with MHEG (Multimedia and Hypermedia information coding Experts Group) set-top box applications. Graphics processor **136** is also coupled to the system bus **130** and operates under the control of CPU **132**. It should be noted that the function of a graphics processor **136** may be unnecessary in set-top box designs having lower capabilities. Also the CPU **132** may function as a graphics processor in some applications.

[0035] The STB may include a smart card reader **140** for communicating with a so called “smart card”, where the smart card reader **140** acts as a Conditional Access Module (CAM). In CAM systems the smart card reader may include a central processor unit (CPU) with associated RAM and ROM memory. Such smart card based CAMs are conventionally utilized for authentication of the user, of transactions carried out by the user, and of services and storage of cryptography keys. For example, the CAM may be used to provide the key for decoding incoming cryptographic data. STB **22** may operate in a bi-directional communication mode. Accordingly, data and other information may be transmitted from the head end **10** to the STB **22** and from the STB **22** using an out-of-band channel. In one embodiment, the data passes through the system bus **130**, modulator **144**, and the tuner **102** (operating as a return path OOB tuner) to the transmission medium **20**. This enables the STB **22** user to send information to the head end **10**, e.g., service requests or changes and registration information.

[0036] Set-top box **22** may include any of a plurality of I/O (Input/Output) signals at I/O interface **146** for interconnection with other devices. By way of example, and not limitation, a serial RS-232 signal may be provided at port **150** to enable interconnection to any suitable serial device supported by the STB **22**'s internal software. Similarly, communication with appropriately compatible devices can be provided via an Ethernet port **152**, a USB (Universal Serial Bus) port **154**, an IEEE 1394 (Firewire or I-Link) port **156**, S-video port **158**, or infrared port **160**. These interfaces may be utilized to interconnect the STB **22** with any of a variety of devices such as storage devices, audio/visual devices **24**, gaming devices (not shown), and Internet Appliances **28**.

[0037] I/O interfaces **146** can include a modem port **162** to facilitate high speed or alternative access to the Internet or other data communication functions. In one preferred embodiment, modem port **162** includes a DOCSIS (Data Over Cable System Interface Specification) cable modem. This modem facilitates high speed network access over a cable system when port **162** is appropriately coupled to a transmission medium **20** embodied as a coaxial cable. A PS/2 or other keyboard/mouse/joystick coupled to port **164** may be used to enable data entry into the STB **22**. STB **22** also may include a basic video output port **166** for direct connection to a television set such as **24**. In one embodiment, Video output port **166** can provide composite video formatted as National Television System Committee (“NTSC”) video. In some embodiments, the video output port **166** may be coupled directly to the graphics processor **136** or the demultiplexer/descrambler **110** rather than passing through the system bus **130** as illustrated in the exemplary block diagram. S-Video signals at output port **158** can be similarly provided without passing through the system bus **130** if desired in other embodiments.

[0038] The infrared port **160** may be embodied as an infrared receiver **34** as illustrated in FIG. 1. The infrared port **160** may receive commands from an infrared remote control **36**, infrared keyboard or other infrared control device. Although not explicitly shown, front panel controls may be used in some embodiments to directly control the operation of the STB **22** through a front panel control interface coupled to the I/O interfaces **146**. Selected interfaces such as those described above and others can be provided in STB **22** in various combinations as required or desired.

[0039] STB **22** may also include a disc drive interface **170** and disc drive mass storage **172** for storage of content and data as well as providing storage of programs operating on CPU **132**. STB **22** may also include other storage mediums such as a floppy disc drive, CD ROM drive, CD R/W drive, DVD drive, and others. CPU **132** is coupled through the system bus **130** to the memory **176**. Memory **176** may include any suitable memory technology including Random Access Memory (RAM), Read Only Memory (ROM), Flash memory, Electrically Erasable Programmable Read Only Memory (EEPROM), and others.

[0040] FIG. 3 is a basic block diagram of an exemplary STB **200** capable of use with the present invention. A detailed block diagram of the STB **200** is shown in FIG. 4. STB **200** is described in detail in provisional Patent Application No. 60/197,233, filed Apr. 14, 2000, Attorney Docket

Number 50P3877, and entitled "Cable Modem Set Top Box" which is incorporated by reference herein for its teachings on the STB 200. Accordingly, the STB 200 is only briefly described with reference to FIGS. 3 and 4. The STB 200 includes a front end 202, cable modem 204, front end to decoder interface 206, MPU/control system 208, MPEG-2 Decoder 210, and Audio/Graphics System 212. The front end 202 with a digital cable television provider via a coaxial cable coupled thereto. The front end 202 could be modified to communicate with alternative digital or analog content providers. The front end to decoder interface 206 links the front end 202, MPU/control system 208, and MPEG-2 decoder 210. The interface 206 includes card readers and an iLink™ interface. The MPEG-2 decoder 210 receives MPEG-2 content from the front end 202 (via the interface 206), and decodes the MPEG-2 content into frames for processing by the Audio/graphics system 212. The micro-processor unit ("MPU")/control system 208 controls the primary operation of the STB 200. The system 208 includes a MPU that supports layers for drivers up to application program interfaces ("APIs") that control the interaction of the components of the STB 200.

[0041] The system 208 may receive control data from the front end 202 (via the interface 206) and send control data to the front end (and ultimately a content provider) via the cable modem 204 and front end 202. The cable modem 204 is coupled to the front end 202 and MPU/control system 208 and can retrieve and place digital data packets on the cable system (in this embodiment). The audio/graphics system 212 can receive video and audio content information from the front end (for analog video/audio), the MPEG-2 decoder (digital audio and video), and the MPU/control system 208.

[0042] A block diagram of software architecture 250 for the STB 200 is shown in FIG. 5. The architecture 250 depicts the hardware layer 252, hardware layer interface/driver layer 254, middleware layer 256, and local content/application layer 258. The middleware layer 256 and local content/application layer 258 comprise the device mosaic. During normal operation of the STB 200, the driver APIs are loaded in the memory of the control system 208. The driver APIs enable communication of events between the MPU and the hardware modules of the STB 200. As shown in FIG. 5, the hardware modules include the Front End Tuner, MPEG-2 Decoder, Demultiplexer, Descrambler, Graphics, Ethernet, Serial port, Smart Card, miscellaneous hardware including keyboard, light-emitting-diodes, infrared, and front panel display.

[0043] The middleware layer 256 includes a group of content handlers, spyglass content manager, spyglass user interface manager, spyglass thin graphical user interface ("GUI"), and application manager. The middleware layer 256 enables the handlers and managers to run on multiple platforms with little regard for the actual operating system in place. At the top layer is the application layer where user applications reside (e.g. web browser, email, Chat, user setup, home page of STB, Video On Demand (VOD), EPG, and iLink user interface). In the present invention, the browser enables Hyper Text Markup Language ("HTML") based pages or screens and browser related pages to be formatted for graphic generation by the audio/graphics system 212. The browser also serves as a jumping point for a mechanism for viewing video media content received from the front end 202 and MPEG-2 decoder 210.

[0044] A block diagram of another embodiment of software architecture 580 for the STB 200 is shown in FIG. 11. The architecture 580 depicts the hardware layer interface/driver layer 570, a TV manager/browser-set-top-hardware-interface 550, and device mosaic 560 where the device mosaic includes a middleware layer and local content/application layer. In this embodiment, the driver APIs are also loaded in the memory of the control system 208. The device mosaic 560 may include a graphical user interface ("GUI"), application manager, and user applications (e.g. web browser, email, Chat, user setup, home page of STB, Video On Demand (VOD), EPG, and iLink user interface). In this embodiment, the browser also enables Hyper Text Markup Language ("HTML") based pages or screens and browser related pages to be formatted for graphic generation by the audio/graphics system 212.

[0045] The software architecture 580 also includes a TV Manager 550. The TV Manager 550 serves as a dynamic interface between the device mosaic and set top box hardware drivers. The TV Manager 550 sends external events to the device mosaic 560 and handles API's generated by the device mosaic 560. The TV manager 550 includes a State machine 514, APIs 516, Private data structures 512, action routines 518, and platform APIs 520. The Web browser of the device mosaic 560 serves as a jumping point for a mechanism for viewing video media content received from the front end 202 and MPEG-2 decoder 210 via a TV media handler 500 browser plug-in. However, the web browser is state-less unlike the front end 202 and MPEG-2 decoder 210. The TV manager 550, due to its independence from the device mosaic 560 in this architecture 500 maintains the state of these devices and other elements of the set top box 200. The TV Manager 550 may be configured to maintain persistency of data and the selection of the web browser is not limited by the set top box architecture. Instead the TV media handler browser plug-in 500 is used to enable the web browser to provide a video section within the Web browser.

[0046] FIG. 6A is an image of a screen generated by the STB 200. The screen includes a HTML page 260. In this case, the page represents EPG from <http://tv.yahoo.com>. In the STB 200, when a Uniform Resource Locator ("URL") is selected within the page 260 that returns with a header that indicates video media content, the STB 200 invokes the TV media browser plug-in 500. A block diagram of one embodiment of the software architecture of the TV media handler 300 is shown in FIG. 7. This embodiment is employed in the software architecture 250 of FIG. 5. As shown in FIG. 7, the TV media handler 300 includes content handler APIs 302, an event decoder 304, a SPD decoder 306, a TV state machine 310, and platform APIs 320. A block diagram of another embodiment of the software architecture of the TV media handler 500 and TV Manager is shown in FIG. 10. This embodiment is employed in the software architecture 580 of FIG. 11. As shown in FIG. 10, the TV media handler 500 includes content handler APIs 502, an event decoder 504 and an SPD decoder 506. The TV Manager 550 includes a TV state machine 510 and platform APIs 520.

[0047] In one embodiment, a web browser invokes the TV media handler 300 of FIG. 7 or 500 of FIG. 10 via the process 400 as shown in the flowchart of FIG. 8. In particular, when the browser application (of the device mosaic) detects a URL with a video media type at step 402,

the browser application invokes the TV media handler **300** of **FIG. 7** or **500** of **FIG. 10** at step **404**.

[**0048**] In either embodiment, the TV media handler **300** or **500** is a content handler in device mosaic that is responsible for controlling a region of the screen and painting that region with a transparent color to allow an underlying video frame or layer to show through the browser or HTML layer or frame based on data returned with the URL indicating video media content. The TV media handler **300** of **FIG. 7** communicates with the STB **200** front end **202** and audio/graphics system **212**. The TV media handler **500** of **FIG. 10** communicates with the TV Manager **550** and the TV Manager **550** STB **200** communicates with the front end **202** and the audio/graphics system **212**.

[**0049**] In one embodiment, when a handler **300** or **500** is instantiated via a URL request that returns a header for video/mpeg media content, data in the format of a Session Description Protocol ("SDP") file is also received. The SDP file indicates parameters for the video session (layer). These parameters may include the channel number to be selected by the front end **202**, the size of the video layer within the HTML layer, and the location of the video layer within the HTML layer. For example, **FIG. 6B** represents a video session where the video layer is positioned in the upper right corner of the HTML layer. **FIG. 6C** represents a video session where the video layer is positioned in the lower middle of the HTML layer. **FIG. 6D** represents a session where the video layer is positioned over the entire the HTML layer.

[**0050**] In this embodiment, the STB **200** generates a HTML layer and video layer and the audio/graphics system **212** is directed to draw these layers. In **FIG. 6A**, the HTML layer consists of the entire screen. In this case, the graphics system **212** is not directed to clear a section of the HTML layer for the video layer. In **FIG. 6B**, the video layer consumes a segment of the upper right corner of the screen. In this case, the TV media handler **300** of **FIG. 7** directs the corresponding section of the HTML layer to be cleared and drawn with a transparent color by the graphics handler. The handler **300** also directs the front end to tune to the appropriate channel for the video signal for the video layer. The media handler **300** also directs the graphics system **212** to size the video layer to fit within the transparent window of the HTML layer. The TV media handler **500** of **FIG. 10** directs the corresponding section of the HTML layer to be cleared and drawn with a transparent color by the graphics handler via the TV Manager **550**. The handler **500** also requests the front end to tune to the appropriate channel for the video signal for the video layer via the TV Manager **550**. The media handler **500** also requests the graphics system **212** to size the video layer to fit within the transparent window of the HTML layer via the TV Manager **550**.

[**0051**] When the handler **300** or **500** receives the SDP file, the handler parses information in the file using the SPD decoder **306** or **506**. Based on the parsed data, the handler makes calls to the TV State Machine **310** or TV Manager **550** to perform the appropriate actions as mandated by the SDP file. When the SDP file indicates that this instance is a VOD session, then media handler **300** or **500** will communicate with the VOD server through the use of a special URL request. The URL request generates URL handlers that will parse the request and communicate with the VOD system at

the head end of the respective system to create a new session. The URL handlers will pass the information from the VOD (server) system back to the TV media handler **300** or **500**.

[**0052**] A flowchart of one TV media handler process **430** is shown in **FIG. 9**. As shown in **FIG. 9**, at step **410**, the handler **300** or **500** parses the video layer session from the SDP file. The session indicates the source of the video signal to be shown in the video layer. The source may be directly tunable at the front end or may require a URL request to generate a new video stream to be transmitted to the STB **200** and retrieved by the front end **202**. Step **412** determines whether the session is a Video on Demand (VOD) session. When the SDP file indicates a VOD session, the handler **300** or **500** generates a URL request to be directed to the respective VOD server as described.

[**0053**] The handler **300** or **500** also parses the video layer size (step **416**) and location (step **418**)

[**0054**] from the SDP file. There may be default locations and sizes in some applications. The handler **300** or **500** directs (or requests via the TV Manager **550**) a transparent section of the HTML or browser layer to be formed (step **420**). The present invention is a windowless system where the screen consists of one HTML or browser layer. To display a video layer in this system, a section (or all) of the HTML or browser layer is cleared and filled with a transparent layer for the video layer to show there through. This reduces the complexity of the STB **200** in terms of MPU cycles, memory overhead and graphics system **212** requirements.

[**0055**] At step **424**, the handler **300** or **500** directs (or requests via the TV Manager **550**) the front end to tune to the channel corresponding to the video signal to be shown in the video layer after parsing the channel from the SDP file at step **422**. The channel may correspond to a VOD channel directed to the STB **200**. The handler **300** or **500** sizes the video layer at step **426** and specifies the location of the video layer on the screen at step **428**.

[**0056**] It is noted that other events that cause the TV media handler **300** to activate. A channel may be selected via a remote control, keyboard, mouse, or front panel of the STB **200** for example. In this case, the event decoder of the handler **300** directs the TV state machine **310** to act accordingly. Other events may include Channel Up, Fast Forward, and on-screen display ("OSD") Program Info Display (such as the current settings of the STB **200**). In software architecture **580** of **FIG. 11**, these other events are handled by the TV Manager **550**. In this embodiment, the TV Manager **550** sends these events to the device mosaic **560** where the event decoder of the handler **500** directs the TV Manager **550** to act accordingly via an API. Other events may include Channel Up, Fast Forward, and on-screen display ("OSD") Program Info Display (such as the current settings of the STB **200**).

[**0057**] In the TV media handler **500**, the TV Manager's **550** platform APIs **520** provide an abstraction layer to enable the TV Media Handler **500** to interface with the STB **200**. The platform APIs include the TV control **522**, OSD **524**, Front Panel Display **526**, Timer Utility **528**, and Video Scaling **532**. The TV control section will provide an API to the rest of the TV Manager **550** that will abstract the

functions of interacting with the actual platform tuning module and others. The On Screen Display (OSD) section, like the TV Tuning section, will provide an interface to the rest of the TV Manager **550** that will abstract the functions and control of running the OSD. This subsection will receive commands through its interface and make the necessary drawing calls to the Device Mosaic **560** to draw the information onto the correct area of the TV media handler's window. The Front Panel Display APIs provides an abstract for the TV Manager **550** module to control the visual display of the front panel.

[0058] The timer utility section provides a set of APIs for the TV Manager **550** to set up timing specific events. The video scaling section provides the TV Manager **550** a method to perform video scaling and/or re-positioning the video window on screen. The content handler APIs **502** of the TV Manager **550** provide a communications channel between the TV Manager **550** and the device mosaic **560**. The API's also provide channels to the data sources (URL handlers) and subviews (gif/jpeg images). Provisional Patent Application No. 60/265,418, filed Jan. 30, 2001, Attorney Docket Number SNY001V, and entitled "Web Browser and

Set Top Box Interface System and Method", which has been incorporated by reference for its teachings provides exemplary TV Manager APIs and screen state. An exemplary state diagram/transition **600** for the set top box screen state is shown in **FIG. 12**.

[0059] Due to the independence of the TV Manager **550** from the device mosaic **560**, the screen state may be determined based on many variables including its current state, previous state, external events as described, and internal events of the state machine **514**. The state machine **514** has eight specific states in one embodiment: startup **610**, system exception **620**, front input **630**, full screen **640**, TV window **650**, User Interface ("UI") menu **660**, Preview **670**, and Preview Full screen **680**. Exemplary parsing grammar that dictates the transition between these states based the variables is detailed in the Provisional Patent Application No. 60/265,418, filed Jan. 30, 2001, Attorney Docket Number SNY001V, and entitled "Web Browser and Set Top Box Interface System and Method". The exemplary parsing grammar for the state "Start-up" is discussed, the language includes:

```

<start up>=
<start up>=

    <start up>
    <initialize pvt data> /* Register with profile manager for email notification */
    <go to init page>
    {
        read default page from preference file
        send external event to DM to go to default URL
    }
    <login as "household" account>
    <select default favorite channel list>
<end new page loaded>

    <start up>
    <new_page_type>
<end key power on>

    <start up>
    <verify that we are really turning the system ON>
    {
        getsystemStatus and verify }
    <set AV on>
    <set "GraphicSubsystemStandby (standby = FALSE)>
    <SmartCardStatus>
<end NO SMART CARD>

    <start up>
    <switch to system exception state>
<end FULL PAGE>

    <start up>
    <checkOsdFlag>
    {
        if (osdFlag)
            OSD_DISPLAY Event to DM (OsdText)
    }
    <CheckSignalStatus>
    {
        If(GetSignalStatus() == UNLOCKED)
            Enqueue OSD message ("Signal not available")
    }
    <display OSD>
    {send external event to DM to display highest priority OSD in queue, set
    OSD timer if necessary}
    <ShowTitleBar for full screen TV>
    <SetTitleBarTimer>
    <display/update Front Panel, and current viewer mail status at FP>
    <AddtoHistory current channel>

```

-continued

```

    <a_set AV on>
<IE WINDOW PAGE>

    <s_TVMGR_TvWindowWithMenu>
    <a_checkOsdFlag>
    <a_CheckSignalStatus>
    <a_display OSD>
    <a_ShowTitleBar for window page>
    <a_SetTitleBarTimer>
    <a_display/update Front Panel, and current viewer mail status at FP>
<IE UI PAGE>

<IE ONLINE PAGE>

    <s_TVMGR_UIMenu>
    <a_update title bar regarding channel info area should be blank>
    <a_ShowTitleBar>
    <a_SetTitleBarTimer for UI page (?? Usually has no timeout)>
    <a_display/update Front Panel, and current viewer mail status at FP>
    <a_set AV of>
<c new page type>

/* 1st time init is TRUE, still waiting for power on key at this point */
{FULL_SCREEN_TV}
    <a_set next_state = FULL_SCREEN>
{TV_WINDOW}
    <a_set next_state = TV_WINDOW>
{UI_MENU}
    <a_set next_state = UI_MENU>
{ONLINE_PAGE} /* online page is detected by not having the pagetype
extended body tag */
    <a_set next_state = UI_MENU>
<c SmartCardStatus >=

{NO_CARD}
    <a_Goto Event to DM (Max TV url)>
    <a_enqueue OSD message "Please insert your access card"*/
    <a_send internal event "IE_NO_SMARTCARD">
{OTHERWISE}
    <a_InitialTuning>
    {
        ch = GetLastChFromPrefFile();
        tuneStatus = TuneCh(ch);
        set last channel = ch;
        set current channel = ch;
        if (status == STATION_NOT_AVAILABLE)
        {
            osdFlag = TRUE
            osdText = "Station not available"
            osdPriority = priority;
        }
    }
    <a_set viewer timeout timer>
    {
        If current viewer is not "household"
            PMViewerTimeoutGet()
        if there is viewer timeout, set timer
    }
    <a_get mail status>
    <a_update Title bar and Front panel regarding current viewer mail status>
    <a_go to next state>
    {
        if "next state" = FULL_PAGE
            send internal event "IE_FULL_PAGE"
        if "next state" = TV_WINDOW_PAGE
            send internal event "IE_WINDOW_PAGE"
        if "next state" = UI_PAGE
            send internal event "IE_UI_PAGE"
        if "next state" = ONLINE_PAGE
            send internal event "IE_ONLINE_PAGE"
    }

```

[0060] The parsing grammar for the state TV Manager start up <s_TVMGR_start up> recognizes three external events, five internal events, and two conditional checks. Because the TV Manager is segmented from the device

mosaic ("DM"), this exemplary parsing grammar is possible. The external events are start up, new page loaded, and power on where the event startup <e_start up> is processed when control transitions to <s_TVMGR_start up>. For this

external event, several actions are performed including initializing the private data structure, reading a default page from a preference file, sending an external event to the DM (device mosaic) to load a default URL or web page. The other actions include setting the account to the household account and setting the default favorite channel list. Control then remains for this state until another external event occurs.

[0061] A second external event is a new page loaded <e_new_page_loaded>. Control again remains with <s_TVMGR_start up> and the condition new_page_type is performed. The condition <c_new_page_type> sets the variable "next_state" to the selected page type. The third external event is triggered when the set top box is powered on <e_key_power_on>. Control again initially remains with <s_TVMGR_start up>. Then several actions are performed and the condition of the smart card is determined by the parsing grammar for <c_SmartCardStatus>. When a smart card is not detected "{NO_CARD}", the DM is directed to switch the browser to a "MAX TV url" and display a message "Please insert your assert card". In addition, an internal event "IE_NO_SMARTCARD" is set. By directing the DM to switch to a specific URL, URL type, or web page that is encoded (via associated SDP file) to have a maximum TV or video window, the TV Manager 550 can direct the TV Media Handler 500 to generate the TV window dictated by the current state and events.

[0062] Otherwise, when a smart card is detected "{OTHERWISE}", several actions are performed for initial tuning of the set top box and an internal event based on the variable or flag "next state" is set, e.g., when "next state" is equal to "FULL PAGE", an internal event "IE_FULL_PAGE" is set. In either case (no smart card or smart card), control remains with start up state until the set internal events are evaluated.

[0063] When the internal event "IE_NO_SMARTCARD" is set, the parsing grammar <ie_NO_SMART_CARD> is processed. In particular in this embodiment the next state is set to system exception "<s_TVMGR_SYSTEM_EXCEPTION>" and control is passed to this state. Control may be passed to Full screen, TV Window or UI Menu when the corresponding internal event is set, e.g., "IE_FULL_PAGE", "IE_WINDOW_PAGE", and "IE_UI_PAGE" or "IE_ONLINE_PAGE". Accordingly in the embodiment employing a TV Manager 550, a complex state control environment may be developed that evaluates external events, preset values, and internal events (within a state). The TV Manager 550 may also determine the TV window size and location by directing the DM to a URL that includes a SDP file with the desired TV or video window characteristics.

[0064] As noted, when the TV medial handler is initiated by a selection in a HTML page, the returned data will be in the form of a session description protocol (SDP) file. In one embodiment, this resource will be returned with the content-type of video/mpeg. The TV media handler will begin reading the incoming SDP file and decode the data to set up the TV session. In one embodiment the set of parameters that are available in a SDP file include:

[0065] Session description

[0066] v=(protocol version)

[0067] o=(owner/creator and session identifier).

[0068] s=(session name)

[0069] i=*(session information)

[0070] u=*(URI of description)

[0071] e=*(email address)

[0072] p=*(phone number)

[0073] c=*(connection information—not required if included in all media)

[0074] b=*(bandwidth information)

[0075] One or more time descriptions

[0076] z=*(time zone adjustments)

[0077] k=*(encryption key)

[0078] a=*(zero or more session attribute lines)

[0079] Zero or more media descriptions

[0080] Time description

[0081] t=(time the session is active)

[0082] r=*(zero or more repeat times)

[0083] Media description

[0084] m=(media name and transport address)

[0085] i=*(media title)

[0086] c=*(connection information—optional if included at session-level)

[0087] b=*(bandwidth information)

[0088] k=*(encryption key)

[0089] a=*(zero or more media attribute lines)

[0090] Window Size

[0091] WindowWidth:Width Value

[0092] WindowHeigh:Height Value

[0093] WindowXPos:Xposition

[0094] WindowYPos:YPosition

[0095] Socket ID

[0096] SocketID:SocketNumber

[0097] TV Commands

[0098] Service to tune

[0099] ServiceID:Channel#

[0100] Service name

[0101] ServiceName:ChannelDescriptorString

[0102] Location of channel graphic (logo) resource.

[0103] LogoURI:PathToLogoData

[0104] VOD Commands

[0105] Initial Data

[0106] VODAssetID:AssetID

[0107] SRMAddress:MachineIP (Connection Address to VOD SRM)

[0108] VODApplicationID:ApplicationID

[0109] SSP Possible Data**[0110]** SSPSocket:SocketID**[0111]** SSPMpegProgram:ProgNumber**[0112]** SSPTransportID:TransportID**[0113]** SSPPhysicalResource:PhysicalResource**[0114]** SSPStreamIPAddress:IPAddress**[0115]** SSPStreamIPPort:IPPort**[0116]** SSPStreamID:StreamID**[0117]** SSPKeepAlive:PeriodInMinutes**[0118] LSCP Possible Data****[0119]** NPTTime:Milliseconds**[0120]** NPTNumerator:Numerator**[0121]** NPIDenominator: Denominator

[0122] While this invention has been described in terms of a best mode for achieving this invention's objectives, it will be appreciated by those skilled in the art that variations may be accomplished in view of these teachings without deviating from the spirit or scope of the present invention. For example, the present invention may be implemented using any combination of computer programming software, firmware or hardware (e.g., a software language other than Java, such as C++ or others may be used to implement the invention). As a preparatory step to practicing the invention or constructing an apparatus according to the invention, the computer programming code (whether software or firmware) according to the invention will typically be stored in one or more machine readable storage mediums such as fixed (hard) drives, diskettes, optical disks, magnetic tape, semiconductor memories such as ROMs, PROMs, etc., thereby making an article of manufacture in accordance with the invention. The article of manufacture containing the computer programming code is used by either executing the code directly from the storage device, by copying the code from the storage device into another storage device such as a hard disk, RAM, etc. or by transmitting the code on a network for remote execution.

What is claimed is:

1. A method of displaying a desired video content frame within a WEB browser based content frame in a windowless environment, comprising the steps of:

- a) directing the WEB browser to load a page having a desired video content frame definition;
- b) generating a transparent section in the browser based content frame based on the decoded frame definition; and
- c) overlapping the video content frame in the transparent section of the browser based content frame.

2. The method of displaying a video content frame within a WEB browser based content frame in a windowless environment of claim 1, wherein the displayed size of the video content frame is smaller than the displayed size of the browser based content frame.

3. The method of displaying a video content frame within a WEB browser based content frame in a windowless environment of claim 2, wherein video content is related to the browser based content.

4. The method of displaying a video content frame within a WEB browser based content frame in a windowless environment of claim 2, wherein step b) includes the steps of:

- a) decoding the desired video frame size from the frame definition; and
- b) generating a transparent section in the browser based content frame based on the decoded frame definition.

5. The method of displaying a video content frame within a WEB browser based content frame in a windowless environment of claim 4, wherein step b) further includes decoding the video frame location within the browser frame from the frame definition.

6. A method of displaying a desired video content frame within a WEB browser based content frame, comprising the steps of:

- a) directing the WEB browser to load a page having a desired video content frame definition; and
- b) generating the desired video content frame definition in the browser based content frame based on the decoded frame definition.

7. The method of claim 6, wherein the WEB browser based content frame is in a windowless environment.

8. The method of claim 6, wherein step b) comprises the steps of:

- i) generating a transparent section in the browser based content frame based on the decoded frame definition; and
- ii) overlapping the video content frame in the transparent section of the browser based content frame.

9. The method of claim 6, wherein the displayed size of the video content frame is smaller than the displayed size of the browser based content frame.

10. The method of claim 6, wherein video content is related to the browser based content.

11. The method of claim 8, wherein step i) includes the steps of:

- a) decoding the desired video frame size from the frame definition; and
- b) generating a transparent section in the browser based content frame based on the decoded frame definition.

12. The method of claim 11, wherein step i) further includes decoding the video frame location within the browser frame from the frame definition.

13. An article of manufacture for use in displaying a desired video content frame within a WEB browser based content frame in a windowless environment, the article of manufacture comprising computer readable storage media including program logic embedded therein that causes control circuitry to perform the steps of:

- a) directing the WEB browser to load a page having a desired video content frame definition;
- b) generating a transparent section in the browser based content frame based on the decoded frame definition; and
- c) overlapping the video content frame in the transparent section of the browser based content frame.

14. The article of manufacture of claim 13, wherein the displayed size of the video content frame is smaller than the displayed size of the browser based content frame.

15. The article of manufacture of claim 14, wherein video content is related to the browser based content.

16. The article of manufacture of claim 14, wherein step b) includes performing the steps of:

i) decoding the desired video frame size from the frame definition; and

ii) generating a transparent section in the browser based content frame based on the decoded frame definition.

17. The article of manufacture of claim 16, wherein step b) further includes decoding the video frame location within the browser frame from the frame definition.

18. An article of manufacture for use in displaying a desired video content frame within a WEB browser based content frame, the article of manufacture comprising computer readable storage media including program logic embedded therein that causes control circuitry to perform the steps of:

a) directing the WEB browser to load a page having a desired video content frame definition; and

b) generating the desired video content frame definition in the browser based content frame based on the decoded frame definition.

19. The article of manufacture of claim 18, wherein the WEB browser based content frame is in a windowless environment.

20. The article of manufacture of claim 18, wherein step b) comprises the steps of:

i) generating a transparent section in the browser based content frame based on the decoded frame definition; and

ii) overlapping the video content frame in the transparent section of the browser based content frame.

21. The article of manufacture of claim 18, wherein the displayed size of the video content frame is smaller than the displayed size of the browser based content frame.

22. The article of manufacture of claim 19, wherein video content is related to the browser based content.

23. The article of manufacture of claim 22, wherein step i) includes performing the steps of:

a) decoding the desired video frame size from the frame definition; and

b) generating a transparent section in the browser based content frame based on the decoded frame definition.

24. The article of manufacture of claim 23, wherein step i) further includes decoding the video frame location within the browser frame from the frame definition.

25. An apparatus for displaying a desired video content frame within a WEB browser based content frame in a windowless environment, comprising:

a) means for directing the WEB browser to load a page having a desired video content frame definition;

b) means for generating a transparent section in the browser based content frame based on the decoded frame definition; and

c) means for overlapping the video content frame in the transparent section of the browser based content frame.

26. The apparatus of claim 25, wherein the apparatus is set top box.

27. The apparatus of claim 26, wherein video content is related to the browser based content.

28. The apparatus of claim 25, wherein the means for generating a transparent section includes:

a) means for decoding the desired video frame size from the frame definition; and

b) means for generating a transparent section in the browser based content frame based on the decoded frame definition.

29. The apparatus of claim 28, wherein the means for generating a transparent section further includes means for decoding the video frame location within the browser frame from the frame definition.

30. An apparatus for displaying a desired video content frame within a WEB browser based content frame, comprising:

a) means for directing the WEB browser to load a page having a desired video content frame definition; and

b) means for generating the desired video content frame definition in the browser based content frame based on the decoded frame definition.

31. The apparatus of claim 30, wherein the WEB browser based content frame is in a windowless environment.

32. The apparatus of claim 30, wherein means for generating the desired video content frame comprises:

i) means for generating a transparent section in the browser based content frame based on the decoded frame definition; and

ii) means for overlapping the video content frame in the transparent section of the browser based content frame.

33. The apparatus of claim 32, wherein the apparatus is a set top box.

34. The apparatus of claim 33, wherein video content is related to the browser based content.

35. The apparatus of claim 33, wherein means for generating a transparent section includes:

a) means for decoding the desired video frame size from the frame definition; and

b) means for generating a transparent section in the browser based content frame based on the decoded frame definition.

36. The apparatus of claim 35, wherein the means for generating a transparent section further includes means for decoding the video frame location within the browser frame from the frame definition.

* * * * *