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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 November 2014 (13.11.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/179888 Al

(51) International Patent Classification:
H04N 21/431 (201 1.01) *H04N 21/235* (201 1.01)
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/CA20 14/050440

(22) International Filing Date:
9 May 2014 (09.05.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/890,775 9' May 2013 (09.05.2013) US

(71) Applicant: ESPIAL GROUP INC. [CA/CA]; 200 Elgin Street, Suite 900, Ottawa, Ontario K2P 1L5 (CA).

(72) Inventors: DOAN, DAT; 17 Sutcliff Lane, Georgetown, Ontario L7G 0C2 (CA). YOGARATNAM, Kumanan; 104B Hopewell Ave., Ottawa, Ontario K1S 2Z1 (CA).

(74) Agents: SPRIGINGS, Mark et al; GOWLING LAFLEUR HENDERSON LLP, 160 Elgin Street, Suite 2600, Ottawa, Ontario KIP 1C3 (CA).

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

(84) Designated States (*unless otherwise indicated, for every kind cf regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.1 7(H))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: OPTIMIZED HTML5 ELECTRONIC PROGRAMMING GUIDE APPLICATION

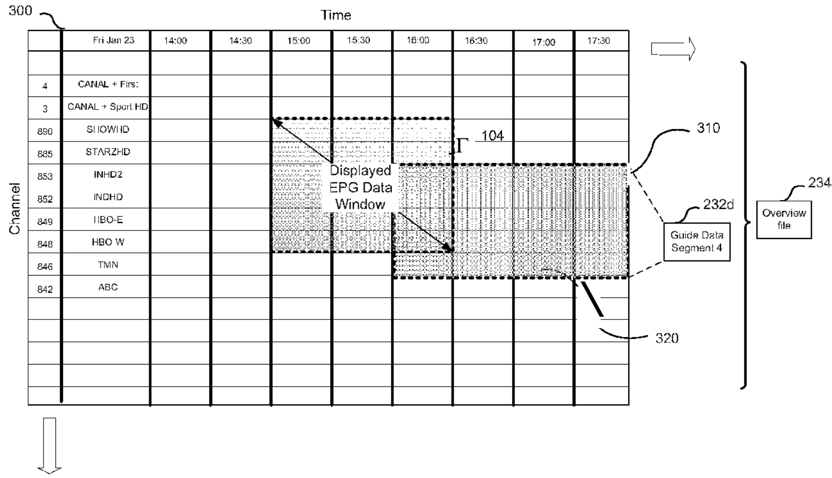


Figure 3

(57) Abstract: An electronic programming guide (EPG) data manager module is provided. The EPG data module provides an event driven infrastructure to provide EPG data to a hypertext markup language 5 (HTML5) applications. The HTML EPG application presents an EPG data window in a user interface on a display device. The EPG data is provided to the EPG application in scriptable data objects generated to be rendered by the HTML5 EPG application based upon the EPG data window to be presented.

OPTIMIZED HTML5 ELECTRONIC PROGRAMMING GUIDE APPLICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to United States non-provisional Patent Application No. 13/890,775 filed May 9, 2013, which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

The present disclosure relates to electronic programming guides (EPG) and in particular to management of EPG data and navigation in a hypertext markup language (HTML) based EPG application.

BACKGROUND

Set-top boxes, Smart TVs, tablets or smartphones can provide an electronic programming guide (EPG) to enable a user to navigate programming schedules and other broadcast media programming with continuously updated time related EPG information for current and upcoming programs. The EPGs are presented in an interactive user interface allowing the user to navigate programs based upon channel (or source) and time to select specific programs for viewing or recording. The EPGs requires data which is typically provided by a service provider and can span a defined time period, typically but not limited to 7 to 14 days, and can be provided in a compressed data format.

In order to present the EPG, the EPG data is retrieved via a network or pushed to the device, the files are then decompressed and the user interface is generated to present the EPG. Hypertext Markup Language 5 (HTML5), as developed by the World Wide Web Consortium (W3C) and JavaScript (JS), as defined in the ECMA-262 specification and ISO/IEC 16262, EPG implementations provide flexibility in user interface design; however they can be memory and processor intensive when working with large datasets such as provided in EPG data. The EPG data handling to generate the EPG results in a number of performance problems and can utilize a large amount of memory resources. There are currently a number of approaches to creating an EPG application in HTML. A first approach is to use a native visual plug-in that manages the EPG data and the rendering of the data in the user interface, which provides optimal performance but provides limited ability to customize the display of the EPG data and does not leverage the flexibility of HTML5. A second approach manages the EPG data and display entirely in HTML/JS implementation, which makes navigation of the EPG data slow and memory intensive since the EPG data needs to be cached in JS and all the logic of the EPG navigation needs to be done via JS and HTML5. For example a byte of compressed EPG data may result in a 100 fold increase in data size when loaded into JS and formatted in HTML. Creating an HTML5 EPG application is very complex since the HTML5 EPG application needs to handle all the aspects of the EPG application including, navigation, rendering, data management which can be taxing on the limited processing and memory resources in device such as set-top boxes, Smart TVs, tablets and smartphones.

Accordingly, systems and methods that enable improved EPG data management in the presentation of an EPG in an HTML5 application remain highly desirable.

BRIEF DESCRIPTION OF THE DRAWINGS

Further features and advantages of the present disclosure will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

FIG. 1 shows a representation of an electronic programming guide;

FIG. 2 shows a representation of EPG system;

FIG. 3 shows a representation of an EPG data set and a data window;

FIG. 4 shows a representation of a system for presenting an EPG;

FIG. 5 shows a method of EPG data manager module;

FIG. 6 shows a method of presenting an EPG data window in an EPG application;

FIG. 7 shows another method of presenting an EPG data window in an EPG application; and

FIG. 8 shows a method of presenting an EPG data window using an overview file.

It will be noted that throughout the appended drawings, like features are identified by like reference numerals.

DETAILED DESCRIPTION

In accordance with an aspect of the present disclosure there is provided a system for presenting an electronic program guide (EPG) user interface (UI) on a display device, the system comprising: an EPG application defined in an Hypertext Markup Language 5 (HTML5) based platform, the HTML5 EPG application presenting programming information defined within an EPG data window in the UI on the display device; and an EPG data manager module for providing EPG data received from one or more EPG data files to the HTML5 EPG application, the EPG data provided by scriptable data objects for the EPG data window, the scriptable data objects provided to the EPG application to render the EPG data window in the user interface on the display device.

In accordance with another aspect of the present disclosure there is provided a method of generating an electronic program guide (EPG) by a processor in a user interface presented on a display device, the method comprising: determining an EPG data window to be presented in the HTML5 EPG application in the user interface presented on the display device; generating at least one scriptable data object from EPG data based upon the determined EPG data window; and providing the generated scriptable data object to the HTML5 EPG application to instruct rendering of the EPG data in the user interface presented on the display device.

In accordance with yet another aspect of the disclosure there is provided A non-transitory computer readable memory containing instructions for generating an electronic program guide (EPG) by a processor in a user interface presented on a display device, the instructions comprising: determining an EPG data window to be presented in the HTML5 EPG application in the user interface presented on the display device; generating at least one scriptable data object from EPG data based upon the determined EPG data window; and providing the generated scriptable data object to the HTML5 EPG application to instruct rendering of the EPG data in the user interface presented on the display device.

Embodiments are described below, by way of example only, with reference to figures 1-8. In order to provide a more efficient electronic programming guide (EPG) in an HTML5 implementation, an EPG data manager is provided to interface with an EPG application executed in an HTML5 browser based platform which may also use an interpreted language such as JS to provide the HTML5 EPG application or functions of the EPG application. An intermediary event handler may be provided to efficiently manage the EPG data for instructing rendering functions of an HTML5 EPG application. The EPG data manager enables an HTML5 EPG application that is memory efficient, provides good performance and is easy to customize by offloading EPG data management, decompression and navigation from the presentation of the EPG. The event handler interfaces between the EPG data manager and the EPG application to service EPG events from the EPG data manager and instruct the EPG application to render the EPG. The event handler can be provided using a scripting language such as JS providing functions that allow the HTML5 EPG application to configure the display window of the EPG data. The event handler notifies the HTML5 EPG application when the EPG data should be rendered based upon the scriptable data objects received from the EPG data manager. The format of the data passed via the event handler will notify the HTML5 EPG application which channels and programs in the EPG display window has been ADDED, REMOVED, or UPDATED to allow the HTML5 EPG application to update its display with minimal effort. Each scriptable data object also contains information to aid the EPG application in rendering the scriptable data object. The additional information can come in the form of CSS selectors for each scriptable data object, or may be provided as coordinates if other aspects of HTML5-integrated presentation technologies are utilized to display EPG such as for example but not limited to SVG, Canvas, WebGL and other alternatives. The EPG application can apply the CSS selector values to the HTML display element representing the scriptable data object to quickly render the EPG data. The CSS selector values are determined based on attributes of the EPG service and program. This approach allows the management of the EPG data and navigation in native code for optimal performance and memory utilization while allowing the rendering of the EPG application to be done completely in HTML5 for complete UI flexibility/customization.

An additional method is illustrated in Figure 7. In this method, the HTML application manages the navigation of the EPG display by calling the EPG Manager to retrieve a window of EPG data based on navigation request by the user. This embodiment allows the HTML application to dynamically change the size of the EPG data window based on the applications navigation rules. The HTML application calls the EPG Manager with configuration information about the size of the EPG data window it wants to retrieve as well as an event handler to be notified when the data of the request EPG data window is available for display. The EPG Manager will then retrieve the requested EPG data (if not already in cache), returns the data to the HTML application by the provided event handler.

Figure 1 shows a representation of an EPG as presented on a display such as a television 100. The EPG 102 may be presented on the display by processing functionality integrated with the display, such as in a television, tablet, smartphone or by an external device such as a digital media receiver, for example a set top box device, which may interface with cable, satellite, wired or wireless networks to provide programming to the television display. Content may be delivered to the device by Internet Protocol (IP) services such as IPTV. The EPG may be presented in various formats, however it typically is presented in a grid 104 providing an EPG data window having a time dimension or axis and a channel dimension or axis displaying program titles positioned relative to their start and stop times. However, the EPG may alternatively be presented in different formats such as a Zap grid which shows a couple of channels and two fixed width columns with "Now" and "Next" programming, or a single channel grid which shows, for a channel, all programs as rows. The user can navigate the grid using directional input and select programs for viewing and possibly recording. Metadata on a particular program can also be displayed such as description, rating, recording status, duration, actors, air date, episode, etc. Although the EPG application may be capable of providing data that can be navigated through for numerous days, typically but not limited to 7 to 14 days, due to the presentation area of the display device 100 only a limited portion of the EPG data can be displayed by the EPG at any one time. As the user moves through the EPG, the EPG data must be retrieved and formatted for display which would be intensive if all the functions such as data retrieval, data extraction, navigation, formatting and rendering were performed within the HTML5 browser.

The EPG 102 is generated by an HTML5 platform based EPG application executed by a processor of the display device 100 or digital media receiver 200. The HTML5 EPG application renders the user interface and inserts the EPG data for formatting and display using HTML5 to provide flexibility in appearance and presentation. HTML5 applications typically utilize cascading style sheets (CSS) for describing the presentation semantics of the user interface to receive and process EPG data and determine user feedback and transitions. In addition, JS may be utilized for implementing portions of the EPG user interface and data interaction through control of CSS properties as well as other HTML5-integrated presentation technologies, such as SVG, Canvas, WebGL and other alternatives). HTML5 provides flexibility in allowing user interface to be readily implemented however the EPG user interface can limit the flexibility when tied to underlying data processing functions required for the EPG presentation due to the increased memory requirement when processing EPG data within the HTML 5 EPG application.

Figure 2 shows a representation of an EPG system. The EPG 102 may be presented in display device 100, but may be generated by a processor directly integrated with the display device 100 or by an external digital media receiver 200, such as a set-top box. The display device 100/ digital media receiver 200 are capable of accessing a network 210 to receive or retrieve EPG data as well as the programming content for viewing. The network 210 may be a packet-switched network such as the Internet to provide IPTV service. A server 220 maintains EPG data 232 and overview data 234 for access by multiple consumption devices such as display device 100 and digital media receiver 200. The EPG data 232 and overview data 234 is stored on a storage device 230 in a single data file or in multiple segments in compressed data files 232a...232n. The data files can be compressed using known compression formats such as zip, gzip, tar, rar, or a proprietary method. Depending on the particular implementation the compressed data files may be segmented by a time period and by groups of channels, content sources, program types or other metadata fields. If the EPG data 232 is provided in segments, the time period may for example be a range of three hours of data while the channel group may be defined by numeric ranges, alphabetical channel associations or programming packages. The EPG data 232 may be retrieved and decompressed on an as needed basis or the display device 100 / digital media receiver 200 may maintain a sliding window of EPG data and download new data segments as they are available. For example the display device 100 / digital media receiver 200 may maintain 14 days of EPG data and each night download new data files or segments, alternatively the display device 100/ digital media receiver 200 may query the server 220 when the user moves to a position in the EPG that data is not present or defined at the display device 100/ digital media receiver 200. Alternatively, the EPG data 232 can be provided by in-band data in the live TV streams to the display device 100/ digital media receiver 200, either incrementally or as a periodic download of data for a defined period. The EPG data may also have one or more associated overview files 234. Each file 234a or 234b can contain high-level EPG data for a period of time such as 7 to 14 days. The overview file can be considerably smaller in size than the data segments provide faster processing. For example the overview file data such as but not limited to channel, program title, air date, production date, genre, rating, start time and duration where as the full EPG file segments 232 contain full metadata such as but not limited to channel, program title, air date, production date, genre, rating, start time, air date, production date, description, actors, icons, etc. The EPG overview data 234 can be used to enable presentation of the EPG while the full EPG data which provides further or additional data is retrieved and decompressed. The overview file provides a minimum amount of information to enable the user to navigate the EPG more quickly while the guide is constructed with the additional level of detail.

Figure 3 shows a representation of an EPG data set 300. The representation is intended to illustrate a data window of the EPG displayed data 104 within an EPG data set 300. The HTML5 EPG is typically presented in a two-dimensional format to present the program data where navigation within the EPG is limited to be within the EPG grid. The dimensions may for example be relative to channel and time axes, although the EPG information may be presented against different data such as genre data, cast data, language data, air date, description, time, etc. The EPG data may also be presented in alternative visual representations such a list or carousel structure that allows navigation of EPG data and may also be ordered relative to different metadata parameters. The grid is typically divided into half-hour blocks 310 however programs may not necessarily be associated with the defined blocks. The number of blocks 310 required for a program, or the number of programs associated with a block is dependent on the start time and duration of the associated programs. The EPG data, comprising the full EPG data, can consist of 7 to 14 days and may be divided into smaller segments for storage or delivered incrementally. For example a portion 320 of the EPG data may be provided by data segment 3 232d, where multiple segments are required to gather the data for the EPG data window. The smaller compressed segments enable the EPG to be easily updated, as only portions of the EPG need to be retrieved. The storage of the EPG data defines program information and does not need to be structured in a table format but in a flat file format with each program being associated with a channel and provide metadata information such as title, description, duration, start time, rating, etc.. The displayed EPG data 104 may span multiple EPG data file segments and therefore require decompression of multiple files based upon the data window, or may be provided in a large compressed data file. Overview data may be represented by the overview file 234a. The overview files provide a high-level EPG data for each time entry which can span multiple guide data segments. The overview file provides quick processing and generation of guide data and can enable more detailed EPG data to be retrieved and decompressed in the background. As the user navigates the EPG grid additional data items may be requested for presentation requiring additional file segments to be retrieved and decompressed. Executing decompression in HTML5 or in an intermediary plug-in can be intensive and sub-optimal in data representation efficiency and resource utilization. The EPG data manager provides a dedicated application implementation which can efficiently process EPG data, receive navigation inputs, determine scriptable data objects that need to be displayed and react to EPG application requirements.

Figure 4 shows a representation of a system for presenting an EPG. The EPG 102 is presented by the display device 100/ digital media receiver 200. The EPG 102 is generated by a processor 400 coupled to a memory 406. The memory 406 contains instructions 410 for generation and presentation of the EPG 102. A display interface 404 is coupled to a display device 100; the connection to the display interface 404 is dependent on where the EPG is generated by the digital media receiver 200, such as a settop box or directly in a tablet, smartphone or smart television. A network interface 402, which may be a wired or wireless interface, connects to the network 210 for retrieval of the EPG overview files 234 and EPG data 232, from storage device 230 coupled to server 220. The instructions 410 comprise an HTML5 browser 420 for providing a user interface to the user. The EPG application 422 can be presented by HTML5-integrated presentation technologies such as CSS defining the visual aspects of the EPG and may also provide function using JS. The event handler 430 can be JS interpreted by a JS engine and provide an application interface for EPG data manager module 440 to provide scriptable data objects to be used by the EPG application 422 to update the EPG data window 104.

The EPG data manager module 440 may be presented as a plug-in to the HTML5 Browser 420 and the HTML5 EPG application 422. The event handler 430 provides a defined interface to facilitate transfer of EPG data between the EPG data manager module 440 and the EPG application 422 and initiate visual updates by the EPG application 422 in response to EPG events such as navigation inputs or changes in EPG data or timeframes. The EPG data manager module 440 handles all complex functionality of the EPG application 422 such as navigation, caching and view window management. The EPG data manager 400 can be provided as a module within the HTML5/JS browser or as a native code application that is a compiled intermediate level language product such as C and C++ to compile directly to the hardware instruction set. In response to a navigation input or changes in the EPG data set, such as a time transition or programming changes, the EPG data manager module 440 determines the data for the view window that is displayed in the EPG application 422. The EPG data manager module 440 determines the EPG data required to update the data window to reflect the navigation or EPG data changes to modify the subset of EPG data displayed in the UI. The EPG data manager module 440 retrieves the required EPG data segment(s), if not already cached, and decompresses the data and provides the data objects via the event handler 430 to the EPG application 422 to update the EPG data window. The event handler 430 instructs the EPG application 422 to render the EPG components when the view window has been updated. The event driven infrastructure notifies the HTML5 EPG application 422 when portions of the EPG user interface display needs to be updated and limits the processing requirements on the EPG application 422. Calls from the EPG data manager module 440 providing a scriptable data object to the event handler 432 enables updates of EPG data to be provided to the EPG application 422 and pushed to the user interface. Functions may be performed such as providing an EPG data update to notify the user interface that the EPG display needs to be updated with the given EPG data, a time slot update to notify the user interface that the EPG visible time slots needs to be updated, or a focus/selection update to notify a change in the user-selected program to be displayed within the user interface.

FIG. 5 shows a method of EPG data manager 400 module. The HTML5 EPG application 422 configures the EPG data manager module 440 by calling a set of application programming interfaces (APIs) and providing parameters that identify the EPG data window (e.g. number of channels, time range), additional data such as CSS selector values based on channel/program attributes and an event handler callback. The event handler 430 provides the configuration parameters to the EPG data manager (502). Based upon the configuration the EPG data manager determines an EPG data window (504), and may retrieve compressed data files for the determined relevant portions of EPG data for the EPG data window. EPG events such as navigation events are triggered by the EPG application 422 calling APIs (e.g. next channel, previous channel, next program, previous program, go to channel and go to specific time) and providing the navigation event to the EPG data manager 440 (506). Alternatively the EPG event may be an EPG data event triggered by the EPG data manager 400 based upon changes in the EPG data or time that would trigger changes in the EPG data window (506) which are pushed to the EPG application 422. Based on the configured EPG data window the EPG data manager 440 will update the EPG data window by adding, removing or updating objects (channels, programs), for each added or updated object by determining the data associated with the EPG event by generating a scriptable data object (508). The EPG data is formatted by the EPG data manger module 440 into the scriptable data object and provided to the event handler 430 (510) to notify it that the display of the EPG data window needs to be updated. Data in the scriptable data object provided the event handler 430 enables the EPG application 422 to know which aspects of the EPG data window is to be updated, removed, added or unchanged, making it efficient for the EPG application 422 to update the display. Information in the event handler 430 also provides the CSS selectors for the visual aspects for each object if required, allowing the EPG application 422 to use CSS to simply apply these selectors to the display element of the scriptable data object. The EPG application 422 can display the scriptable data object with the desired effect without having to do any computation to retrieve, decompress and format EPG data reducing processing requirements on the EPG application.

Figure 6 shows a flowchart of a method of presenting an EPG data window. The EPG application provides parameters defining the EPG data window to the EPG data manager (600). The data window defines the range of data that is required to be presented in the EPG based upon the user interface view shown on the display device. For example, when activating the EPG the default window may be based upon the current time and a listing of channels defined relative to the currently viewed channel. The EPG data manager retrieves or receives EPG data from a remote server (602). The EPG data is provided in a single compressed file, incrementally or may be provided in defined for segments of the EPG, for example an EPG may span three hours and cover a range of channels. The EPG data is centrally updated and stored and distributed to service providers. In order to provide a full EPG, for example 7 days, the data may be represented by hundreds of EPG data segments in the form of compressed files may be required to complete the EPG. Dividing the EPG into multiple file segments allows the EPG to be progressively updated with individual segments being added as required. Alternatively, the EPG may be provided incrementally or as a full periodic update providing a full EPG data window in the update. An EPG event is received by the EPG data manager (604) from user interaction within the EPG application, for example navigation input and is sent to the EPG data manager, or a change in the EPG data is determined by the EPG data manager. When the EPG view is displayed within the EPG application, the EPG data manager determines the size and range of EPG data window associated with EPG event (606). For example an incremental single line movement within the EPG where only one line is updated or a page increment where all data in the display is updated. In addition, the EPG data window is dependent on the current view selected for displaying the EPG and how much information can be viewed in that display. For example a typical grid formation may show a number of channels, vertically for a specific time period such as five channels for a three hour time period. The EPG may be displayed in alternative formats, for example where the channels are displayed horizontally or displayed for a single channel. Based upon the EPG events received, for example navigation events moving within the EPG grid, displaying a new channel group, changing the time period displayed, etc, the EPG data manager determines if the required data is available and has been decompressed to generate associated scriptable data objects. If the EPG data has already been previously cached (YES at 608) the EPG data is formatted into a scriptable data object (610) required to display the new information. The scriptable data object may define status actions such as ADDED, UPDATE, REMOVED, NOCHANGED, along with display CSS selectors based on the EPG data attributes that are provided to the event handler (612). The event handler then instructs the EPG application to update and render the user interface (614) using the updated scriptable data object. The EPG application can then render the EPG (61 6) to present the selected EPG data window. If the EPG data is not already decompressed (NO at 608), the appropriate file segment(s) for the data window is determined and retrieved (618) or a request is provided to the server for an updated version of the EPG data. The retrieved file is decompressed (620) and the updated EPG data is formatted into a scriptable data object (610). Alternatively scriptable data objects may be generated in response to EPG data events such as EPG data changes based upon time or programming changes when the EPG is being viewed. When the scriptable data objects are generated by EPG events the EPG data manager will determine the required object and pass objects to the event handler.

Figure 7 shows a flowchart of an alternative method of presenting an EPG data window. The EPG application makes an API call into the EPG manager providing parameters for an EPG data window and an event handler (700). If the requested EPG data has already been previously cached (YES at 702), the EPG data is formatted into a scriptable object. The EPG manager then provides the scriptable EPG data object back to the EPG application via the provided event handler (706). The EPG application can then render the requested EPG data (708). If the EPG data is not already decompressed (NO at 702), the appropriate file segments for the requested data window are retrieved (710) and decompressed (712) and the scriptable data object is generated (704).

Figure 8 shows a method of presenting an EPG data window using an overview file. The EPG application makes an API call into the EPG manager providing parameters for an EPG data window and an event handler (800) for a defined period of time. An overview file that covers the requested time period is retrieved (802). If the full EPG data for the time period has already been previously cached (YES at 804), the EPG data is formatted into a scriptable object (814). The EPG manager then provides the scriptable EPG data object back to the EPG application via the provided event handler. The EPG application can then render the requested EPG data (816). If the EPG data is not already decompressed (NO at 804), a scriptable data object is generated for the EPG from data provided in the overview file (806) and the EPG guide data is rendered using the overview data (808). The one or more data file segments having the full data for the requested data window is determined and retrieved (812) and decompressed in the background (812). Scriptable data objects are then generated from the full EPG data (814). The EPG manager then provides the scriptable EPG data objects back to the EPG application via the provided event handler and the EPG application can then render the requested EPG data (816) using the full data set. The method enables the EPG to be presented to the user to enable viewing and navigation without a lag in presentation by providing overview information first and as additional data is retrieved provide additional or full level of detailed program information in the EPG.

Referring to Figure 3, the EPG data manager may have data for most of the EPG data window, for example file segments 1-3 have been decompressed, whereas a portion of the window is not available and would require file segment 4 to be retrieved and/or decompressed. In addition, the EPG data manager may maintain a buffer window around the EPG data window to speed navigation or predicatively decompress EPG overview and full data based upon previous actions.

Although the implementation is described relative a grid format EPG, the implementation may be utilized in other implementations such as carousel navigation. In addition, although HTML5 has been described, the described system, method and computer readable memory may be applicable to future HTML standards based upon HTML5 framework.

In some embodiments, any suitable computer readable media can be used for storing instructions for performing the methods described herein. For example, in some embodiments, computer readable media can be transitory or non-transitory. For example, non-transitory computer readable media can include media such as magnetic media (such as hard disks, etc.), optical media (such as compact discs, digital video discs, blu-ray™ discs, etc.), semiconductor media (such as flash memory, electrically programmable read only memory (EPROM), electrically erasable programmable read only memory (EEPROM), etc.), any suitable media that is not fleeting or devoid of any semblance of permanence during transmission, and/or any suitable tangible media.

Although the description discloses example methods, system and apparatus including, among other components, software executed on hardware, it should be noted that such methods and apparatus are merely illustrative and should not be considered as limiting. For example, it is contemplated that any or all of these hardware and software components could be embodied exclusively in hardware, exclusively in software, exclusively in firmware, or in any combination of hardware, software, and/or firmware. Accordingly, while the following describes example methods and apparatus, persons having ordinary skill in the art will readily appreciate that the examples provided are not the only way to implement such system, methods and apparatus.

CLAIMS:

1. A system for presenting an electronic program guide (EPG) user interface (UI) on a display device, the system comprising:

   an EPG application defined in an Hypertext Markup Language 5 (HTML5) based platform, the HTML5 EPG application presenting programming information defined within an EPG data window in the UI on the display device; and

   an EPG data manager module for providing EPG data received from one or more EPG data files to the HTML5 EPG application, the EPG data provided by scriptable data objects for the EPG data window, the scriptable data objects provided to the EPG application to render the EPG data window in the user interface on the display device.

2. The system of claim 1 wherein the EPG data window is a subset of available EPG data.

3. The system of claim 1 or 2 wherein the scriptable data object is generated in response to a navigation event generated by the EPG application changing the EPG data rendered in the EPG data window.

4. The system of claim 1 or 2 wherein the scriptable data object is generated in response to an API call from the EPG application.

5. The system of claim 1 or 2 wherein the scriptable data object is generated in response to an EPG event associated with a change in the EPG data which results in a change in the EPG data window.

6. The system of claim 1 or 2 wherein the scriptable data object is generated in response to an EPG event associated with a change in time resulting in a change in the EPG data window.

7. The system of any one of claims 1 to 6 further comprising an event handler for providing EPG events to the EPG data manager module and receiving scriptable data objects from the EPG data manager module and instructing

the EPG application to render the EPG using data from the scriptable data objects.

8. The system of claim 7 wherein the event handler generates navigation events in response to user navigation within the Ul, the navigation event provided to the EPG data manager module.

9. The system of any one of claims 1 to 8 wherein the EPG data manager module determines an EPG data file from a plurality of EPG data files based upon the EPG data window, each EPG data file containing programming data for one or more content sources for a defined time period.

10. The system of claim 9 wherein the EPG data manager module decompresses one or more EPG data files from the plurality of EPG data files.

11. The system of claim 10 wherein the EPG data manager module extracts EPG data larger than the EPG data window from the one or more EPG data files and caches data based upon predictive cache actions based upon event requests received from an event handler.

12. The system of any one of claims 1 to 11 wherein the EPG data manager module is a compiled intermediate level language product compiled directly to a hardware instruction set.

13. The system of claim 12 wherein the EPG data manager module is generated in native code compiled from C or C++.

14. The system of any one of claims 1 to 13 wherein the an EPG data manager module retrieves an overview file for rendering the EPG data window prior to retrieving EPG data from the one or more EPG data files the overview file providing high-level EPG data while detailed EPG data is retrieved from the one or more EPG data files.

15. The system of claims 14 wherein the overview file provides one or more of channel, program title, genre, rating, start time or duration.

16. The system of claims 15 wherein the one or data files provides one or more of channel, program title, genre, rating, air date, production date, start time, duration, description, actor, icons.

17. A method of generating an electronic program guide (EPG) by a processor in a user interface presented on a display device, the method comprising:

determining an EPG data window to be presented in an Hypertext Markup Language 5 (HTML5) EPG application in the user interface presented on the display device;

generating at least one scriptable data object from EPG data based upon the determined EPG data window; and

providing the generated scriptable data object to the HTML5 EPG application to instruct rendering of the EPG data in the user interface presented on the display device.

18. The method of claim 17 wherein the EPG data window is determined by receiving EPG configuration parameters from the HTML5 EPG application, the configuration parameters defining the EPG data window to be presented from the EPG data.

19. The method of claims 17 or 18 wherein the EPG data window is a subset of available EPG data.

20. The method of any one of claims 17 to 19 wherein determining the EPG data window further comprises receiving a navigation event generated by the EPG application changing EPG data to be presented in the EPG data window wherein the scriptable data object is generated in response to the navigation event.

21. The method of any one of claims 17 to 20 wherein determining the EPG data window further comprises receiving an EPG event associated with a change in the EPG data therefore changing the EPG data window wherein the scriptable data object is generated in response to the EPG event.

22. The method of any one of claims 17 to 21 wherein determining the EPG data window further comprises determining an EPG event based upon a change in time therefore changing the EPG data to be presented in the EPG data window wherein the scriptable data object is generated in response to the change in time.

23. The method in any one of claims 17 to 22 wherein the EPG data is provided in one or more data files, wherein the EPG data is retrieved by decompressing data from the one or more data files to retrieve the EPG data to generate the scriptable data objects.

24. The method in claim 23 wherein generating the at least one scriptable data object from EPG data further comprises:

determining one or more data files that are associated with the data window;

retrieving the determined one or more data files; and

decompressing the retrieved one or more data files.

25. The method of any one of claims 17 to 24 wherein each scriptable data object contains status information NOCHANGED, ADDED, UPDATED or REMOVED to aid to the EPG application in efficiently updating rendering of the EPG data window.

26. The method of claim 25 where each scriptable data object contains one or more CSS selectors that reflect a display state of the scriptable data object.

27. The method of claim 26 where the CSS selectors are calculated by based on the scriptable data object attributes.

28. The method of claim 23 further comprising the retrieving an overview file of EPG data for generating the at least one scriptable data object.

29. The method of claim 28 further comprising retrieving an overview file of EPG data for generating the at least one scriptable data object before retrieving the one or more data files.

30. The method of claim 23 further comprising:

retrieving an overview file of EPG data for generating the at least one scriptable data object before retrieving the one or more data files, the one or more data file provided additional EPG data;

generating at least one scriptable data object from EPG data based upon the determined EPG data window from the overview file; and

providing the generated scriptable data object to the HTML5 EPG application to instruct rendering of the EPG data in the user interface presented on the display device before retrieving the one or more data files.

31. The method of claims 30 wherein the overview file provides one or more of channel, program title, genre, rating, start time or duration.

32. The method of claims 31 wherein the one or data files provides one or more of channel, program title, genre, rating, air date, production date, start time, duration, description, actor, icons.

33. A non-transitory computer readable memory containing instructions for generating an electronic program guide (EPG) by a processor in a user interface presented on a display device, the instructions comprising for implementing the method of claims 17 to 32.

34. A method of presenting an electronic programming guide (EPG) the method comprising:

determining an EPG data window for display in the EPG;

retrieving an overview file that covers a requested EPG data window;

generating a scriptable data object from overview EPG data in the overview file;

rendering the EPG guide using scriptable data object having the overview data;

retrieving one or more data file segments having full data for the requested EPG data window;

generating an updated scriptable data object from EPG data in the one or more data file segments; and

rendering the EPG guide using the updated scriptable data object from the EPG data in the EPG data window.

35. The method of claim 34 further comprising determining if full EPG data for the EPG data window has already been previously cached, if the full EPG data has been cached the EPG guide is rendered using the full EPG data.

36. The method of claims 34 or 35 further comprising decompressing the one or more data file segments prior to generating the updated scriptable data object.

37. The method of claim 36 wherein the one or data files segments is decompressed in the background.

38. The method of claim 37 wherein an EPG application is defined in a Hypertext Markup Language 5 (HTML5) based platform, the HTML5 EPG application presenting programming information defined within an EPG data window in the user interface on a display device.

39. The method of claim 38 wherein the EPG data manager provides EPG data received from one or more EPG data files to the HTML5 EPG application, the EPG data provided by scriptable data objects for the EPG data window, the scriptable data objects provided to the EPG application to render the EPG data window in the user interface on the display device.

40. The method of claim 39 wherein the EPG data window is determined by receiving EPG configuration parameters from the HTML5 EPG application, the configuration parameters defining an EPG time period for the EPG data window to be presented from the EPG data.

41. The method of claim 40 wherein the EPG data window is a subset of available EPG data.

42. The method of any one of claims 34 or 41 wherein determining the EPG data window further comprises receiving a navigation event generated by the EPG application changing EPG data to be presented in the EPG data window wherein the scriptable data object is generated in response to the navigation event.

43. The method of any one of claims 34 to 41 wherein determining the EPG data window further comprises receiving an EPG event associated with a change in the EPG data therefore changing the EPG data window wherein the scriptable data object is generated in response to the EPG event.

44. The method of any one of claims 34 to 41 wherein determining the EPG data window further comprises determining an EPG event based upon a change in time therefore changing the EPG data to be presented in the EPG data window wherein the scriptable data object is generated in response to the change in time.

45. The method of any one of claims 34 to 44 wherein each scriptable data object contain status information NOCHANGED, ADDED, UPDATED or REMOVED to aid to the EPG application in efficiently updating rendering of the EPG.

46. The method of claim 45 where each scriptable data object contains one or more CSS selectors that reflect a display state of the scriptable data object.

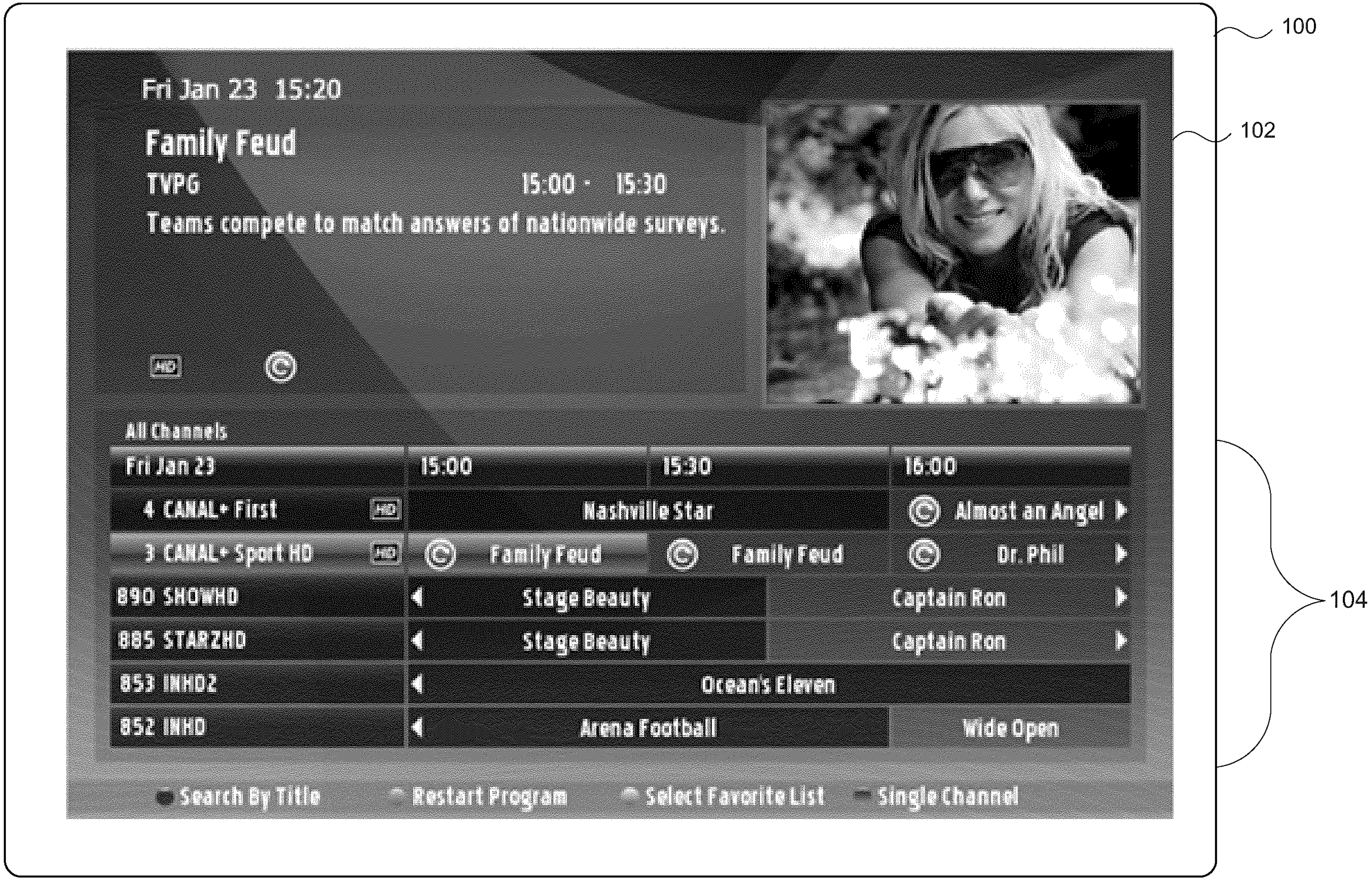
47. The method of claim 46 where the CSS selectors are calculated based on the scriptable data object attributes.

48. The method of any one of claims 34 to 47 wherein the overview file provides one or more of channel, program title, genre, rating, start time or duration.

49. The method of claim 48 wherein the one or data files provides one or more of channel, program title, genre, rating, air date, production date, start time, duration, description, actor, and icons.

50. A non-transitory computer readable memory containing instructions for generating an electronic program guide (EPG) by a processor in a user

interface presented on a display device, the instructions comprising for implementing the method of claims 34 to 49.



Fri Jan 23 15:20

Family Feud

TVPG 15:00 - 15:30

Teams compete to match answers of nationwide surveys.

HD ©

All Channels

Fri Jan 23	15:00	15:30	16:00
4 CANAL+ First HD	Nashville Star		© Almost an Angel ▶
3 CANAL+ Sport HD HD	© Family Feud	© Family Feud	© Dr. Phil ▶
890 SHOWHD	◀ Stage Beauty		Captain Ron ▶
885 STARZHD	◀ Stage Beauty		Captain Ron ▶
853 INHD2	◀ Ocean's Eleven		
852 INHD	◀ Arena Football		Wide Open

● Search By Title ● Restart Program ● Select Favorite List ▬ Single Channel

Figure 1

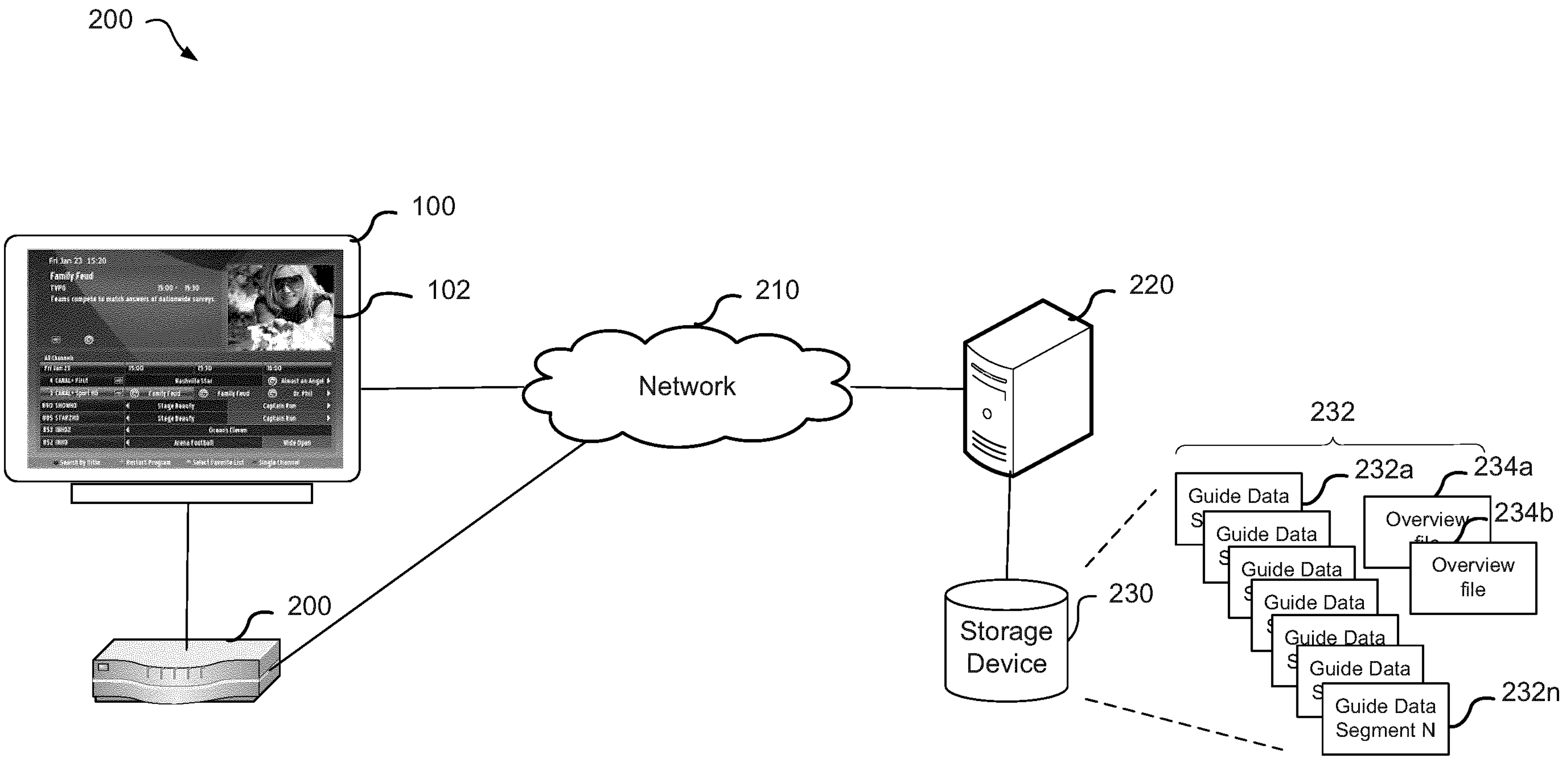

200
100
102
Fri Jan 23 15:20
Family Feud
TVPG
15:00 - 15:30
Teams compete to match answers of nationwide surveys.
All Channels
Fri Jan 23
15:00
15:30
16:00
4 CANAL+ First
Nashville Star
Almost an Angel
3 CANAL+ Sport HD
Family Feud
Family Feud
Dr. Phil
890 SHOWHD
Stage Beauty
Captain Ron
895 STARZHD
Stage Beauty
Captain Ron
853 HBOHD
Ocean's Eleven
852 HBO
Arena Football
Wide Open
Search By Title
Restart Program
Select Favorite List
Single Channel
210
Network
220
230
Storage Device
232
232a
Guide Data S
Guide Data S
Guide Data S
Guide Data S
Guide Data S
Guide Data S
Guide Data Segment N
232n
234a
Overview file
234b
Overview file
200

Figure 2

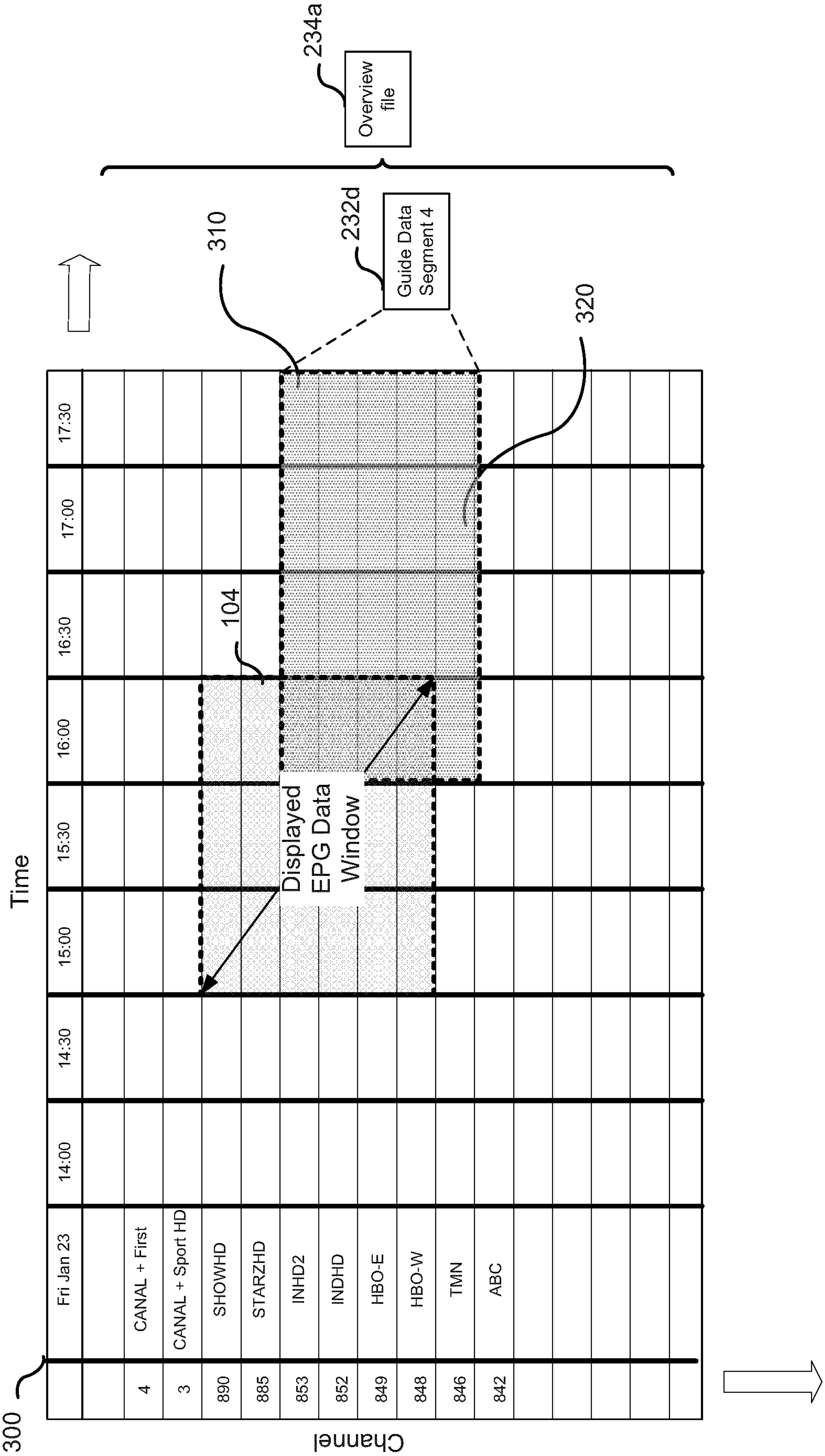

234a
Overview file
310
232d
Guide Data Segment 4
320
104
Displayed EPG Data Window
Time
Channel
300
Fri Jan 23
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
4 CANAL + First
3 CANAL + Sport HD
890 SHOWHD
885 STARZHD
853 INHD2
852 INDHD
849 HBO-E
848 HBO-W
846 TMN
842 ABC

Figure 3

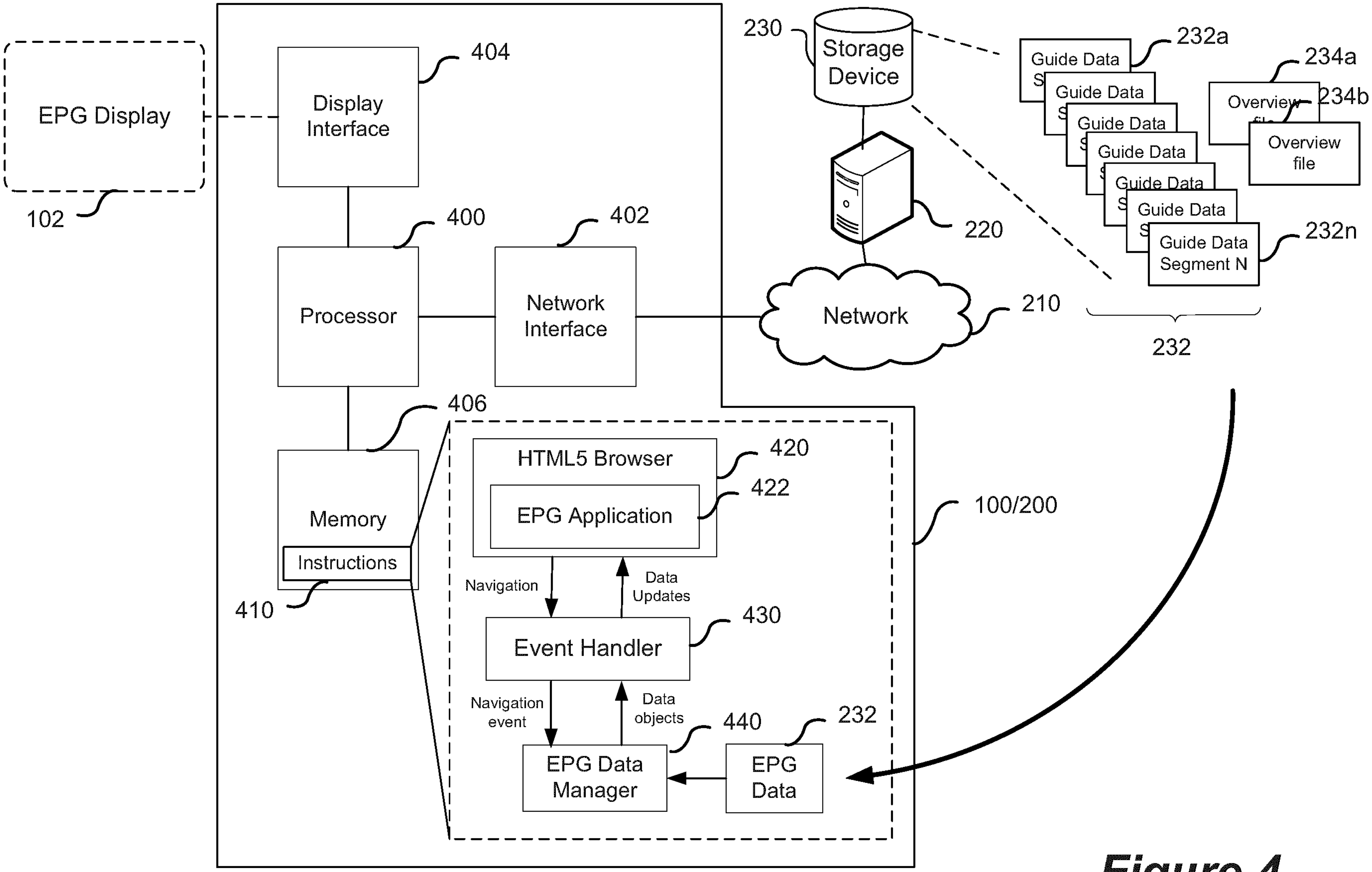

EPG Display
102
Display Interface
404
Processor
400
Network Interface
402
Memory
406
Instructions
410
HTML5 Browser
420
EPG Application
422
Navigation
Data Updates
Event Handler
430
Navigation event
Data objects
EPG Data Manager
440
EPG Data
232
100/200
Storage Device
230
220
Network
210
Guide Data S
Guide Data S
Guide Data S
Guide Data S
Guide Data S
Guide Data S
Guide Data Segment N
232a
232n
232
Overview file
Overview file
234a
234b

Figure 4

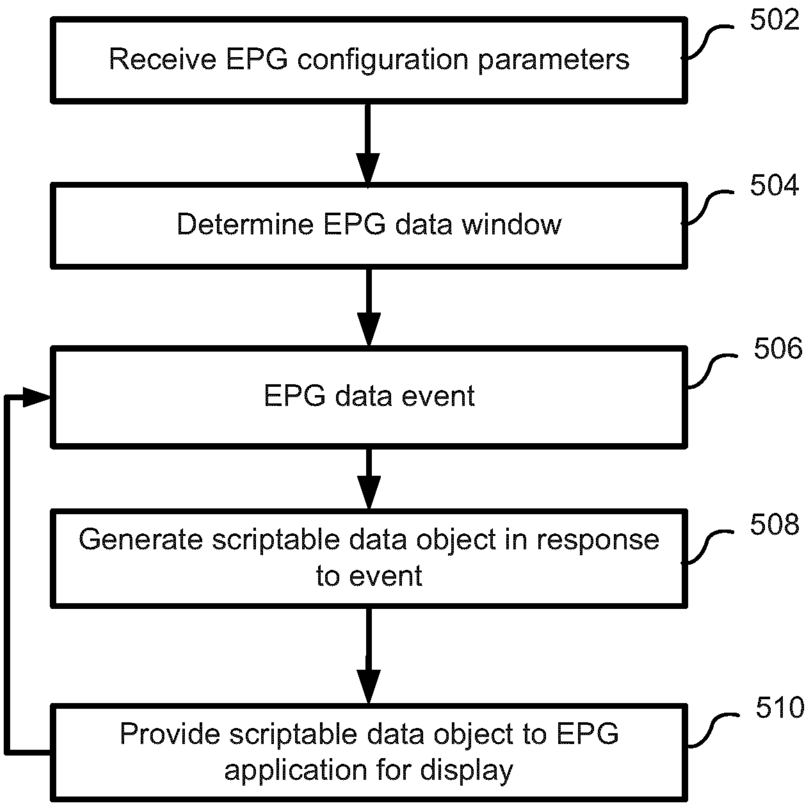

Receive EPG configuration parameters
502
Determine EPG data window
504
EPG data event
506
Generate scriptable data object in response to event
508
Provide scriptable data object to EPG application for display
510

Figure 5

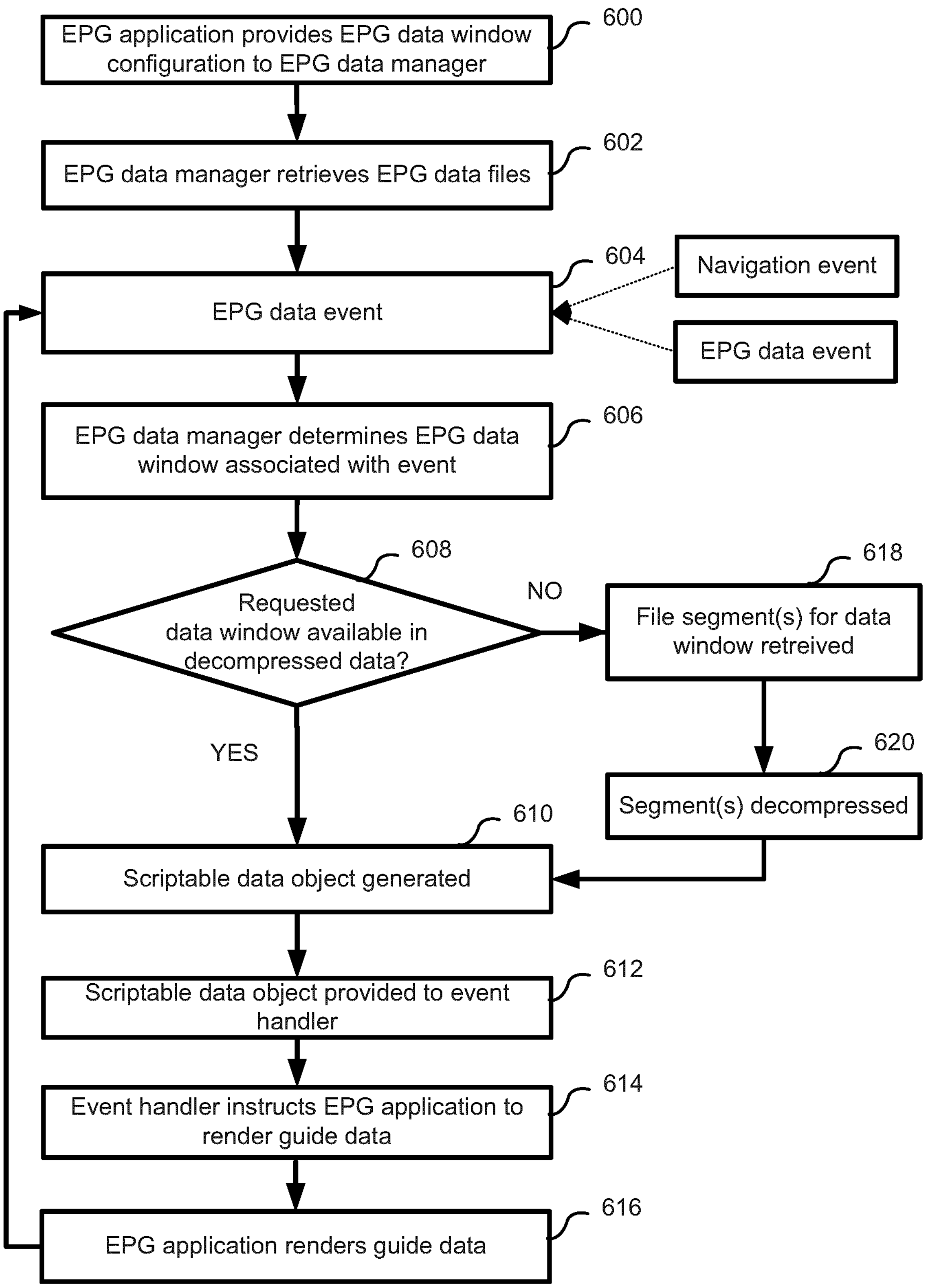

EPG application provides EPG data window configuration to EPG data manager
600
EPG data manager retrieves EPG data files
602
EPG data event
604
Navigation event
EPG data event
EPG data manager determines EPG data window associated with event
606
Requested data window available in decompressed data?
608
NO
File segment(s) for data window retreived
618
YES
Segment(s) decompressed
620
Scriptable data object generated
610
Scriptable data object provided to event handler
612
Event handler instructs EPG application to render guide data
614
EPG application renders guide data
616

Figure 6

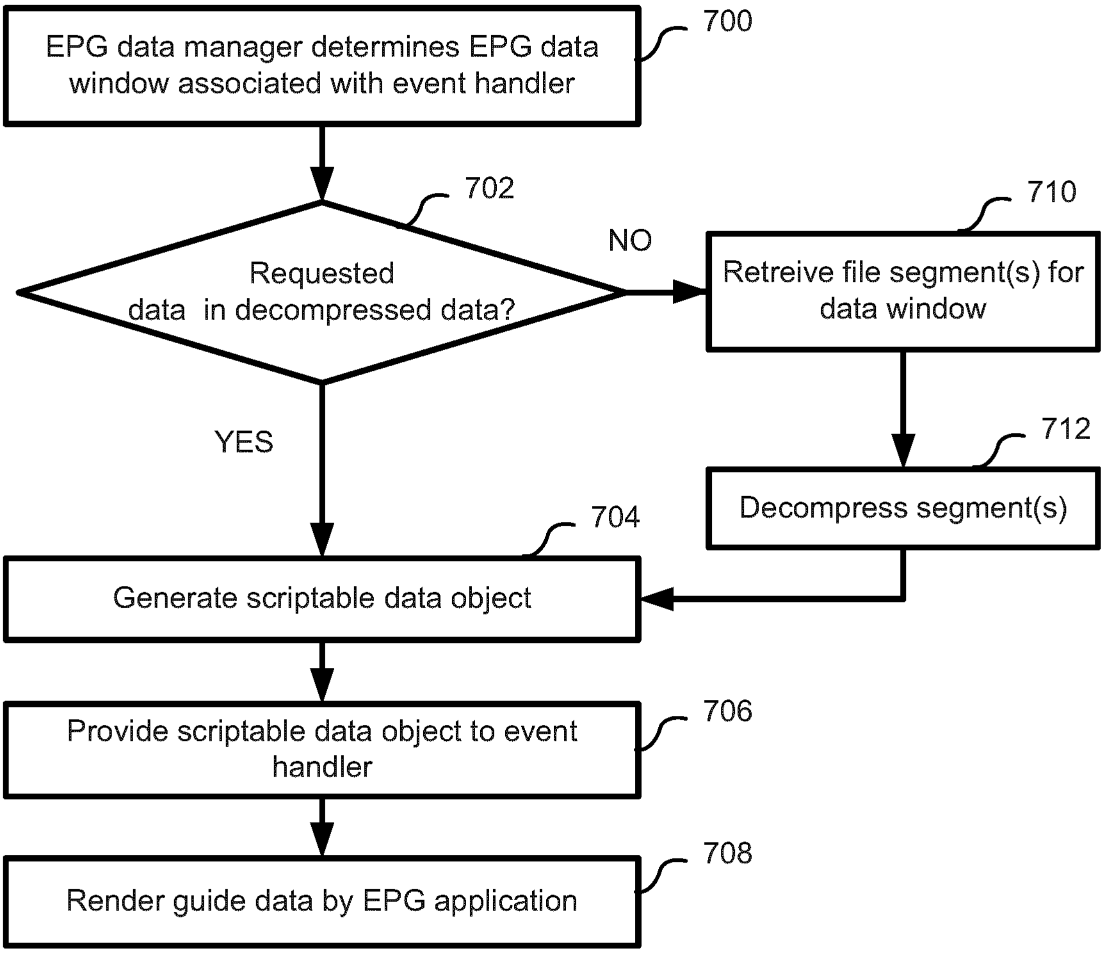

EPG data manager determines EPG data window associated with event handler
700
Requested data in decompressed data?
702
NO
Retreive file segment(s) for data window
710
Decompress segment(s)
712
YES
Generate scriptable data object
704
Provide scriptable data object to event handler
706
Render guide data by EPG application
708

Figure 7

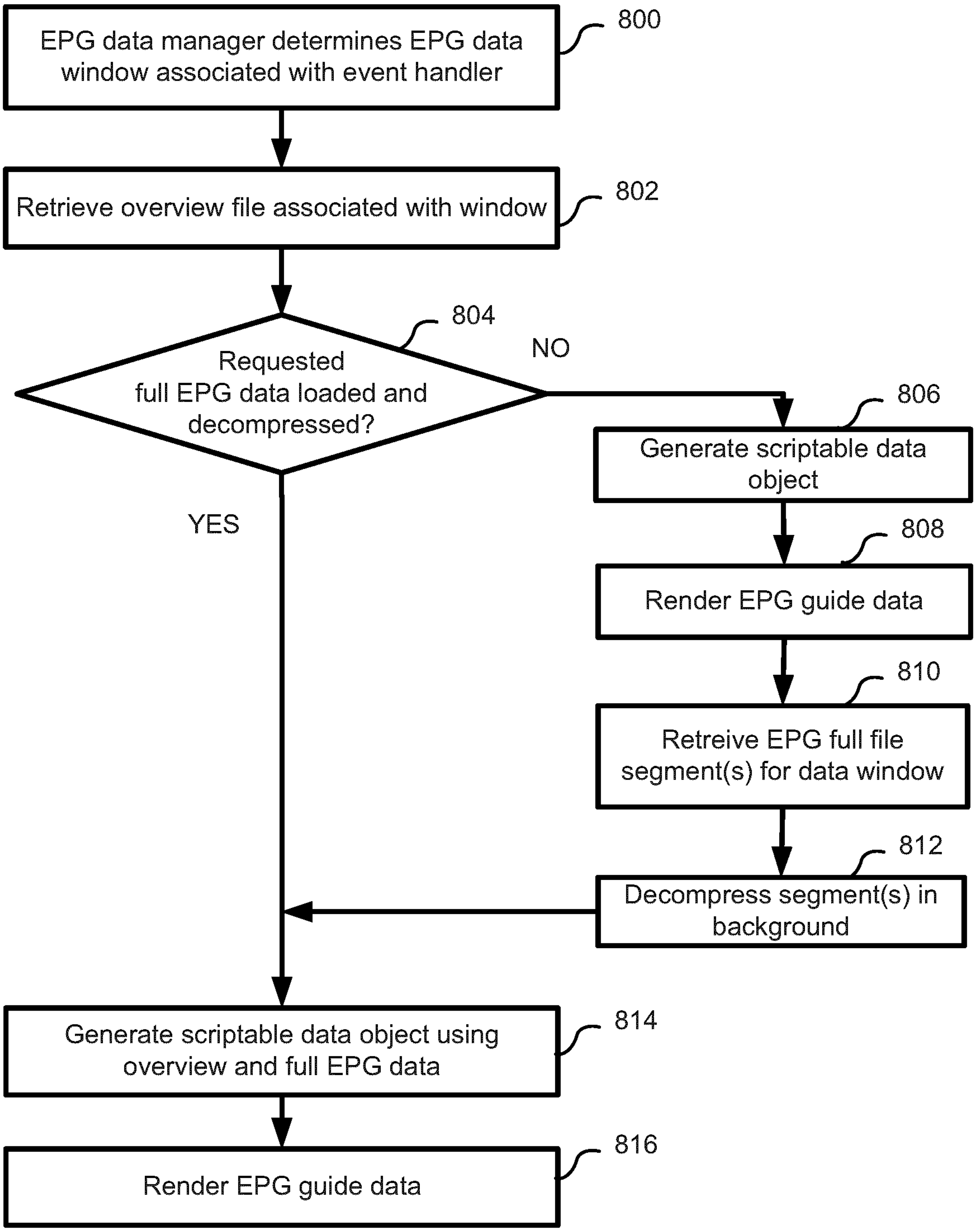

EPG data manager determines EPG data window associated with event handler
800
Retrieve overview file associated with window
802
Requested full EPG data loaded and decompressed?
804
NO
YES
Generate scriptable data object
806
Render EPG guide data
808
Retreive EPG full file segment(s) for data window
810
Decompress segment(s) in background
812
Generate scriptable data object using overview and full EPG data
814
Render EPG guide data
816

Figure 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2014/050440

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***H04N 21/431*** (201 1.01) , ***H04L 29/06*** (2006.01) , ***H04N21/235*** (201 1.01)

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)
IPC: H04N (201 1.01) , H04L (2006.01)

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

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

Canadian Patent Database, Total Patent, Keywords: Electronic Programming guide, program guide, EPG, HTML, HTML5, script, data, program, listings, set-top, application, render

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO2012051539A2 (Wetzer et al.), 19 April 2012 (19-04-2012) *See pars. [0050]-[0055], [0109], [0122], [0198]-[0201], [0268]-[0276], figures 1-1 1 and 27	1 to 50
A	US2012099494A1 (Salinger), 26 April 2012 (26-04-2012) *See entire document	1 to 50
A	US2013014138A1 (Bhatia et al.), 10 January 2013 (10-01-2013) *See entire document	1 to 50
P, A	US2013219429A1 (Hirsch et al.), 22 August 2013 (22-08-2013) *See entire document	1 to 50

□ 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 application or patent but 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 11 July 2014 (1 1-07-2014)	Date of mailing of the international search report 22 July 2014 (22-07-2014)
Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C I 14 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Authorized officer **Lambros Pezoulas (819) 934-2677**

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/CA2014/050440

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
WO2012051539A2	19 April 2012 (19-04-2012)	WO2012051539A2	19 April 2012 (19-04-2012)
		WO2012051539A3	05 July 2012 (05-07-2012)
		EP2304584A1	06 April 2011 (06-04-2011)
		EP2304584A4	09 November 2011 (09-11-2011)
		IL209288D0	31 January 2011 (31-01-2011)
		JP2011523737A	18 August 2011 (18-08-2011)
		JP5325286B2	23 October 2013 (23-10-2013)
		KR20110063617A	13 June 2011 (13-06-2011)
		US2010122196A1	13 May 2010 (13-05-2010)
		US8499250B2	30 July 2013 (30-07-2013)
		US2011252461A1	13 October 2011 (13-10-2011)
		US8578285B2	05 November 2013 (05-11-2013)
		US2011258573A1	20 October 2011 (20-10-2011)
		US8595641B2	26 November 2013 (26-11-2013)
		US2011252132A1	13 October 2011 (13-10-2011)
		US8751948B2	10 June 2014 (10-06-2014)
		US2011252460A1	13 October 2011 (13-10-2011)
		US2012144416A1	07 June 2012 (07-06-2012)
		WO2009140386A1	19 November 2009 (19-11-2009)
		WO2012103121A1	02 August 2012 (02-08-2012)
US2012099494A1	26 April 2012 (26-04-2012)	US2012099494A1	26 April 2012 (26-04-2012)
		US8588198B2	19 November 2013 (19-11-2013)
		CA2726095A1	25 April 2012 (25-04-2012)
		US2014177508A1	26 June 2014 (26-06-2014)
US2013014138A1	10 January 2013 (10-01-2013)	US2013014138A1	10 January 2013 (10-01-2013)
		US2013014158A1	10 January 2013 (10-01-2013)
		US8607295B2	10 December 2013 (10-12-2013)
		US2013014223A1	10 January 2013 (10-01-2013)
		US8631473B2	14 January 2014 (14-01-2014)
		US2013014222A1	10 January 2013 (10-01-2013)
		US8635674B2	21 January 2014 (21-01-2014)
		US2013014146A1	10 January 2013 (10-01-2013)
		US8650587B2	11 February 2014 (11-02-2014)
		US2013014145A1	10 January 2013 (10-01-2013)
		US8667520B2	04 March 2014 (04-03-2014)
		US2013014136A1	10 January 2013 (10-01-2013)
		US2013014137A1	10 January 2013 (10-01-2013)
		US2013014141A1	10 January 2013 (10-01-2013)
		US2013014143A1	10 January 2013 (10-01-2013)
		US2013014144A1	10 January 2013 (10-01-2013)
		US2013014153A1	10 January 2013 (10-01-2013)
		US2013019258A1	17 January 2013 (17-01-2013)
		US2013019262A1	17 January 2013 (17-01-2013)
		US2014173641A1	19 June 2014 (19-06-2014)
		US2014173643A1	19 June 2014 (19-06-2014)
US2013219429A1	22 August 2013 (22-08-2013)	US2013219429A1	22 August 2013 (22-08-2013)
		AU2012254064A1	17 October 2013 (17-10-2013)
		CA2832172A1	15 November 2012 (15-11-2012)
		EP2695058A1	12 February 2014 (12-02-2014)
		JP2014512611A	22 May 2014 (22-05-2014)
		KR20140018329A	12 February 2014 (12-02-2014)
		US8261231B1	04 September 2012 (04-09-2012)
		US2012260232A1	11 October 2012 (11-10-2012)

US2013205276A1	08 August 2013 (08-08-2013)
US2013205278A1	08 August 2013 (08-08-2013)
US2013247005A1	19 September 2013 (19-09-2013)
US2013283305A1	24 October 2013 (24-10-2013)
US2013305218A1	14 November 2013 (14-11-2013)
US2014109046A1	17 April 2014 (17-04-2014)
WO2012154347A1	15 November 2012 (15-11-2012)
WO2013121285A2	22 August 2013 (22-08-2013)
WO2013121285A3	17 April 2014 (17-04-2014)
WO2013121286A2	22 August 2013 (22-08-2013)
WO2013121286A3	08 May 2014 (08-05-2014)
WO2013121293A2	22 August 2013 (22-08-2013)
WO2013121293A3	03 January 2014 (03-01-2014)
WO2013121301A2	22 August 2013 (22-08-2013)
WO2013121301A3	27 December 2013 (27-12-2013)
WO2013121302A2	22 August 2013 (22-08-2013)
WO2013121302A3	27 December 2013 (27-12-2013)