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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 January 2007 (04.01.2007)

PCT

(10) International Publication Number
WO 2007/001607 A2

(51) International Patent Classification:
H04N 7/025 (2006.01) *G11B 27/00* (2006.01)
H04N 7/10 (2006.01) *G06F 13/00* (2006.01)
H04N 5/445 (2006.01) *G06F 3/00* (2006.01)

(21) International Application Number:
PCT/US2006/015671

(22) International Filing Date: 26 April 2006 (26.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/167,963 28 June 2005 (28.06.2005) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **LA CHAPELLE, Kevin Leigh**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **WALKER, Brian James**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **MERCER, Ian Cameron**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **KASE, Hiroshi**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **MATSUO, Keisuke**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **ASAYAMA, Sako**; One Microsoft Way, Redmond, Washington 98052-6399 (US).

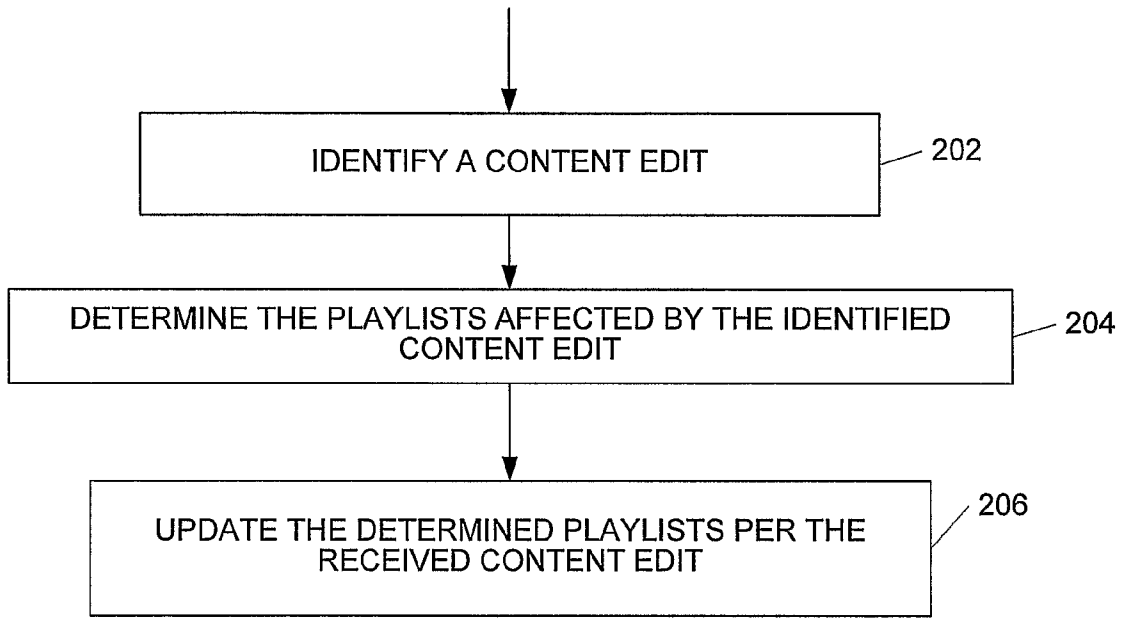
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC,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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** CREATING STANDARDIZED PLAYLISTS AND MAINTAINING COHERENCY



(57) **Abstract:** Updating automatically generated playlists in response to media content being edited. Aspects of the invention associate a playlist type or other code with each playlist to define the structure of the playlist. When media content is added to or deleted from a collection of media files, one or more playlist types corresponding to the media content are identified. The playlists with the identified playlist types are updated to reflect the addition or deletion of the media content.

WO 2007/001607 A2

CREATING STANDARDIZED PLAYLISTS AND MAINTAINING COHERENCY

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is a continuation-in-part of co-pending U.S. Patent Application Serial No. 10/273,415, filed October 17, 2002, entitled "Navigating Media Content via Groups within a Playlist," hereby incorporated by reference, which claims the benefit of U.S. Provisional Patent Application Serial No. 60/418,973, filed October 16, 2002, entitled "COMPRESSED MEDIA FORMAT SPECIFICATION," now abandoned.

BACKGROUND

[0002] Due to recent advances in technology, computer users are now able to enjoy many features that provide an improved user experience, such as playing various media and multimedia content on their personal or laptop computers. For example, most computers today are able to play compact discs (CDs) so users can listen to their favorite musical artists while working on their computers. Many computers are also equipped with digital versatile disc (DVD) drives enabling users to watch movies.

[0003] In some multimedia environments, a computer has access to a computer-readable medium storing compressed media files such as Moving Picture Experts Group audio layer-3 (MP3) files and WINDOWS MEDIA technologies audio (WMA) files. When the media files are rendered on a computer, the computer typically has access to a database storing metadata describing albums, artists, genres, years, or the like for the media files. The computer typically organizes the media files into playlists based on the metadata when the compressed media files are played on the computer. For example, in the case of audio media files, the files may be organized by album, artist, genre, year, or some user specified selection and ordering. This allows users to easily have access to all of their content regardless of whether or not the users manually created a playlist.

[0004] However, when compressed media files are transferred from the computer to an optical format such as CD or DVD for playback on a portable CD player, DVD player or other consumer electronic devices having a low-power processor, limited memory and often limited display and user input capabilities, the media files traditionally lose much of the functionality offered by metadata databases hosted on devices with greater computing power. For example, while a first device may be able to render a playlist authored by a second device, the first device may not be able to identify the type of the playlist. In one example, the first device may generically identify as "Playlist 1" an "All Songs by Genre" playlist created by the second device, possibly due to an inability by the first device to

interpret the headers of the playlist generated by the second device. Some existing systems lack a mechanism for interpreting the intent of a playlist and identifying playlists to all playback devices.

[0005] Some existing systems also lack a mechanism for marking or identifying outdated playlists. For example, a user may add or delete content, but a playback device with limited processing and/or memory capability may not be able to update any of the playlists or identify an outdated playlist to another device.

[0006] Accordingly, a system for creating standardized playlists and maintaining coherency on multiple authoring devices is desired to address one or more of these and other disadvantages.

SUMMARY

[0007] Embodiments of the invention provide a way for automatically generated playlists to be kept consistent when a device changes media content stored thereon through edits (e.g., additions or deletions). In an embodiment, the invention extends rich playlist functionality by tagging each playlist with a code which defines the structure of the playlist and enables a device or software to identify the playlists needing to be updated when new content is added or deleted. If a device is processor constrained to such an extent that it cannot update a playlist, the device sets an inconsistency flag or the like which indicates to other devices and software that the media content has changed. The next time the media content is operated on by a device or software with authoring capability the playlists are reauthored, rebuilt, or otherwise updated. Aspects of the invention provide a consistent user experience throughout a media environment.

[0008] Alternatively, aspects of the invention may comprise various other methods and apparatuses.

[0009] Other features will be in part apparent and in part pointed out hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a block diagram illustrating an exemplary media environment in which the invention may be implemented.

[0011] FIG. 2 is an exemplary flow chart illustrating operation of an embodiment of the invention.

[0012] FIG. 3 is an exemplary flow chart illustrating adding a song to a computer-readable medium according to an embodiment of the invention.

[0013] FIG. 4 is a block diagram illustrating one example of a suitable computing system environment in which aspects of the invention may be implemented.

[0014] Corresponding reference characters indicate corresponding parts throughout the drawings.

DETAILED DESCRIPTION

[0015] Referring first to FIG. 1, a block diagram illustrates an exemplary media environment in which the invention may be implemented. A system 100 has one or more computers 102 coupled to one or more consumer electronic devices 112 providing media content including audio data, video data, and/or still image data. For example, the devices 112 may include a compact disc (CD) player 104, a camcorder 106, or a camera 108. Additionally, the devices 112 may include other personal computers, removable hard drives, network shares, a Moving Picture Experts Group audio layer-3 (MP3) player, an audio system in an automobile, a personal digital assistant, a cellular telephone, or the like. The consumer electronic devices 112 may include any suitable rendering filter or media player or device (e.g., a portable media device) that is configured to render digital media so that the user can experience the content that is embodied on the consumer electronic device 112. For example, suitable media player applications include a compact disc (CD) media player and a digital versatile disc or digital video disc (DVD) media player. The computer 102 also has rendering capability including a processor and rendering software (e.g., a media player).

[0016] A media environment typically has playlists generated from a plurality of media files stored on a computer-readable medium. When dealing with large quantities of media files and/or devices that can add and delete media files from the computer-readable medium, there is a need to keep generated playlists up to date to provide the user with a consistent experience. In an embodiment, the invention associates a code with each playlist to identify the structure of the playlist. In particular, the invention tags each playlist with a 'Playlist Type' field, an inconsistency flag, or other code which enables the authoring software or device to know which playlists to update when media content is added or deleted.

[0017] The consumer electronic devices 112 or computer 102 may have access to one or more computer-readable media (e.g., memory area 122). While the memory area 122 is illustrated to be part of any of the consumer electronic devices 112 in FIG. 1, the memory area 122 may be separate from the consumer electronic devices 112 yet accessible to the consumer electronic devices 112, for example, via a network. In one embodiment, memory area 122 includes one or more computer-readable media for storing a plurality of media files 124 and a plurality of data structures 126. Each of the plurality of data

structures 126 represents a playlist. Each of the data structures 126 stores a list of media files representing the playlist. The list of media files is selected from the plurality of media files 124. Each of the data structures 126 has a playlist type field storing a value identifying the playlist as a user-created playlist or an automatically generated playlist. Each of the data structures 126 further has an inconsistency field storing a value indicating whether the playlist is out-of-date with respect to the plurality of media files 124.

[0018] One aspect of the present invention enables the user or, particularly, enables a media player program executing on computing device 112, to access, retrieve, and display for the user, so-called metadata. Those skilled in the art are familiar with metadata, which is simply information about data. In the context of the illustrated embodiment, metadata includes information related to specific content of a digital media file being played on the media player. Basic metadata includes, but is not limited to, title, performer, genre, track number, and the like. Extended metadata includes, but is not limited to, cover art, composer, description of content, performer biographies, reviews, ratings, related performers, where to buy similar items, upcoming concerts, ticket sales, URLs to other related experiences including purchase opportunities, studio, director, and the like. In one embodiment, extended metadata may be organized into two main categories: metadata retrieved or downloaded, and metadata computed from the media file (e.g., digital signal processing of the file stream). The metadata may be stored within the media file or stored in another file accessible and known to the media file.

[0019] In one embodiment, the consumer electronic devices 112 (e.g., a portable media device) are configured to execute computer-executable instructions for identifying a content edit. The content edit indicates media content to be added to or deleted from a plurality of media files 124 stored on an accessible computer-readable medium such as memory area 122. The media content has metadata associated therewith. The consumer electronic devices 112 are further configured to determine, as a function of the metadata, one or more of the data structures 126 affected by the identified content edit. The consumer electronic devices 112 are further configured to set the value in the inconsistency field for each of the determined data structures 126.

[0020] The computer 102, or other device or software with playlist authoring capability, also has one or more exemplary modules or components for implementing aspects of the invention. For example, the computer 102 may have an interface component 114, a table component 116, and an author component 118. The interface component 114 identifies a content edit. The content edit indicates the media content to be added to or

deleted from the plurality of media files 124. The media content has metadata associated therewith. The table component 116 determines, as a function of the metadata, one or more of the playlists affected by the content edit identified by the interface component 114. In one embodiment, the table component 116 determines the one or more playlists by identifying playlists having a predefined playlist type. For example, if the metadata associated with the media content indicates that the media content is a song, then the playlists affected by the media content include those automatically generated playlists containing songs (e.g., "all songs by artist," "all songs by album," etc.). In another example, if the metadata associated with the media content indicates that the media content is an image file, then the playlists affected by the media content include those automatically generated playlists containing image files.

[0021] The author component 118 updates the playlists determined by the table component 116 per the received content edit. The table component 116 and the author component 118 execute automatically in response to execution of the interface component 114. In an embodiment in which the content edit identified by the interface component 114 indicates that the media content is to be added to the computer-readable medium, the author component 118 updates the playlists determined by the table component 116 by reauthoring the playlists identified by the table component 116 with the media content. In an embodiment in which each of the playlists has an inconsistency flag associated therewith, the author component 118 updates each of the playlists having an inconsistency flag that is set.

[0022] Those skilled in the art will note that the invention software may be implemented with any number and organization of components or modules. That is, the invention is not limited to the specific configuration of the interface component 114, the table component 116, and the author component 118, but may include more or less components having more or less individual functionality than described herein. Further, the invention may be embodied in hardware, software, or a combination thereof in a media player, operating system, DVD recorder, CD recorder, video camera, hard drive, flash drive, personal digital assistant, wireless device (e.g., cellular telephone), or the like.

[0023] In one example, additional metadata is available from the metadata provider 111 via a data communication network 113. The computer 102 and metadata provider 111 are coupled to the data communication network 113. While the network 113 includes the Internet in one example, the teachings of the invention may be applied to any data

communication network. Data communication network 113 may support, for example, client/server communications or peer-to-peer connections.

[0024] Referring next to FIG. 2, an exemplary flow chart illustrates operation of an embodiment of the invention in a media environment having playlists. The playlists are automatically generated from a plurality of media files stored on a computer-readable medium. The flow chart in FIG. 2 illustrates a method for updating the playlists. The method includes identifying a content edit at 202. The content edit indicates media content to be added to or deleted from the plurality of media files. The media content has metadata associated therewith. The method also includes determining, as a function of the metadata, one or more of the playlists affected by the identified content edit at 204. In one embodiment, determining the one or more playlists includes identifying playlists having a predefined playlist type corresponding to the metadata. The method further includes updating the determined playlists per the received content edit at 206. The determining and updating occur automatically in response to identifying the content edit.

[0025] If the identified content edit indicates that the media content is to be added to the computer-readable medium, the method updates the determined playlists by reauthoring the determined playlists with the media content. If the identified content edit indicates that the media content is to be deleted from the computer-readable medium, the method updates the determined playlists by reauthoring the determined playlists without the media content.

[0026] In one example, the playlist type may include a 7- or 8-bit value representing an integer. An exemplary list of playlist types is shown below in Table 1. The playlist type definitions shown in Table 1 are only exemplary. Users, manufacturers, and the like may define other playlist types.

Playlist Type	Value	Description
0	User Playlist	A user created playlist or one like a .M3U, .WPL, etc.
1	All Songs by Artist	All of the songs grouped by Artist Name in ascending order.
2	All Songs by Genre	All of the songs grouped by Genre Name in ascending order.
3	All Songs by Album	All of the songs grouped by Album Name in ascending order.
4	Individual Artist	One playlist containing all of the songs for a single Artist.
5	Individual Album	One playlist containing all of the songs for a single Album sorted in ascending track number order.
6	Individual Genre	One playlist containing all of the songs for a music Genre.
7	All Images	All of the images in a single playlist grouped by the directory that they are in.
8	Video	A playlist that contains a single Video File.
9-127	Reserved	Reserved for future use.

Table 1. Exemplary Playlist Types and Corresponding Values and Descriptions.

[0027] When a device according to the invention adds a new media file, the device adds the new media file to all of the relevant playlists. For example, if the media file is a song, then all playlists of types 1-6 are updated. If the media file is an image, then the playlists of type 7 are updated. If the device does not have sufficient resources (e.g., computing power or memory) to update the relevant playlists, then the device sets an inconsistency flag for each of the relevant playlists. The inconsistency flag is used by authoring devices or software to know whether a playlist is out-of-date. For example, an inconsistency flag that is set (e.g., set to one or true) indicates that a data set (e.g., the media files) has changed but that changes to the playlists have not been performed. A set inconsistency flag triggers an authoring device or software to rebuild, re-author, or recreate the particular playlist with the set inconsistency flag. Similarly, an inconsistency flag that is not set (e.g., set to zero or false) indicates that the playlist is up-to-date with any changes that may have occurred to the data set.

[0028] The inconsistency flag may be any code or indicator conveying a state of a playlist. In other embodiments, for example, the code may be a consistency flag which is set to indicate that the playlist is up-to-date and not set when the playlist has been

determined to be out-of-date. The code may be represented by a single-bit value, an eight-bit value, or any number of bits.

[0029] In one example, a rendering device or software determines a status of a playlist being rendered and conveys that information to a user. For example, the rendering device determines whether or not the inconsistency flag has been set. If the rendering device lacks sufficient resources to update a playlist that has a set inconsistency flag, the device informs the user of the out-of-date status of the playlist. For example, the device may shade the playlist name gray to identify to the user that the playlist is outdated.

[0030] The playlist types may also be used when creating new playlists. For example, a method according to the invention for creating automatically generated playlists includes automatically selecting one or more of a plurality of media files as a function of an item of metadata. For example, the metadata may indicate whether the media content includes audio content, video content, or image content. The method further includes creating a playlist with the selected media content. The created playlist includes a playlist header. The method further determines a playlist type to be associated with the created playlist wherein the determined playlist type corresponds to the item of the metadata. The method adds the determined playlist type to the playlist header.

[0031] For user-created playlists, a method of the invention includes receiving a list of user selected content. The method further includes creating a playlist with the received list of user selected content and adding a particular playlist type to the created playlist. The particular playlist type identifies the playlist as being user created.

[0032] One aspect of the invention (not shown in FIG. 2) includes rendering the media content from one of the playlists, determining a string name associated with said one of the updated playlists as a function of the predefined playlist type, and displaying the determined string name to a user.

[0033] One or more computer-readable media have computer-executable instructions for performing the method illustrated in FIG. 2.

[0034] Referring next to FIG. 3, an exemplary flow chart illustrates adding a song to a computer-readable medium (e.g., a disc) in one embodiment. At 302, a song is to be added to the disc by a device. If the device is not capable of updating playlists at 304, the inconsistency flag is set on affected playlists (e.g., all songs by album, artist, and genre) at 306 and the process finishes at 312. If the device is capable of updating playlists at 304, the song is added to the playlists at 308. If the device is capable of fixing (e.g., identifying and updating) inconsistent playlists at 310, the inconsistent playlists are fixed (e.g., identified

and updated) at 314 and the process in this embodiment finishes at 312. If the device is not capable of fixing inconsistent playlists at 310, the process in this embodiment finishes at 312.

Examples

[0035] In an embodiment of the invention, a playlist file table stores information about the playlists (e.g., playlist files) on a device or computer-readable medium. The playlist file table includes a list of playlist file entries each corresponding to a playlist file. An exemplary format for each playlist file entry is shown below in Table 2.

Offset	Byte Length	Field Name
0	4	Playlist Identifier
4	1	Playlist Summary Type (e.g., Audio, Image, Video)
5	1	Playlist Type

Table 2. Playlist File Entry.

[0036] The playlist identifier is a 4-byte value that is a unique identifier that corresponds to this playlist. The playlist summary type is a 1-byte value summarizing the types of all the individual media files or playlist groups contained within the playlist. In one embodiment, the summary type is created by a logical OR of the values shown below in Table 3. The playlist summary type allows a particular device or software to filter the list of playlists to show only the playlists that can be rendered by the particular device or software.

5 Bits	1 Bit	1 Bit	1 Bit
RESERVED	Image Flag	Video Flag	Audio Flag

Table 3. Values Used to Create the Playlist Summary Type.

[0037] Exemplary values for the image flag, video flag, and audio flag in Table 3 are shown below in Table 4, Table 5, and Table 6, respectively.

Image Flag	Value
0_b	No Timed Image or Parallel Image Audio Playlist Group
1_b	At least one Timed Image or Parallel Image Audio Playlist Group

Table 4. Image Flag.

Video Flag	Value
0_b	No Video Playlist Group
1_b	At least one Video Playlist Group

Table 5. Video Flag.

Audio Flag	Value
0_b	No Audio Playlist Group
1_b	At least one Audio Playlist Group

Table 6. Audio Flag.

[0038] The playlist type is a 1-byte entry used to identify a playlist that should be updated or replaced by authoring software or devices when new media files are added to a computer-readable medium or old media files are removed from the computer-readable medium. When authoring software or recording devices do not update the playlist or all or part of any grouped playlists, the inconsistency flag is set to one. The inconsistency flag is set to zero if the playlist type is a user playlist.

1 Bit	7 Bits
Inconsistency Flag	Playlist Type

Table 7. Playlist Type Field.

Exemplary Operating Environment

[0039] FIG. 4 shows one example of a general purpose computing device in the form of a computer 130. In one embodiment of the invention, a computer such as the computer 130 is suitable for use in the other figures illustrated and described herein. Computer 130 has one or more processors or processing units 132 and a system memory 134. In the illustrated embodiment, a system bus 136 couples various system components including the system memory 134 to the processors 132. The bus 136 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0040] The computer 130 typically has at least some form of computer readable media. Computer readable media, which include both volatile and nonvolatile media, removable and non-removable media, may be any available medium that may be accessed

by computer 130. By way of example and not limitation, computer readable media comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. For example, computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that may be used to store the desired information and that may be accessed by computer 130. Communication media typically embody computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and include any information delivery media. Those skilled in the art are familiar with the modulated data signal, which has one or more of its characteristics set or changed in such a manner as to encode information in the signal. Wired media, such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared, and other wireless media, are examples of communication media. Combinations of any of the above are also included within the scope of computer readable media.

[0041] The system memory 134 includes computer storage media in the form of removable and/or non-removable, volatile and/or nonvolatile memory. In the illustrated embodiment, system memory 134 includes read only memory (ROM) 138 and random access memory (RAM) 140. A basic input/output system 142 (BIOS), containing the basic routines that help to transfer information between elements within computer 130, such as during start-up, is typically stored in ROM 138. RAM 140 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 132. By way of example, and not limitation, FIG. 4 illustrates operating system 144, application programs 146, other program modules 148, and program data 150.

[0042] The computer 130 may also include other removable/non-removable, volatile/nonvolatile computer storage media. For example, FIG. 4 illustrates a hard disk drive 154 that reads from or writes to non-removable, nonvolatile magnetic media. FIG. 4 also shows a magnetic disk drive 156 that reads from or writes to a removable, nonvolatile magnetic disk 158, and an optical disk drive 160 that reads from or writes to a removable, nonvolatile optical disk 162 such as a CD-ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that may be used in the exemplary operating environment include, but are not limited to, magnetic tape

cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 154, and magnetic disk drive 156 and optical disk drive 160 are typically connected to the system bus 136 by a non-volatile memory interface, such as interface 166.

[0043] The drives or other mass storage devices and their associated computer storage media discussed above and illustrated in FIG. 4, provide storage of computer readable instructions, data structures, program modules and other data for the computer 130. In FIG. 4, for example, hard disk drive 154 is illustrated as storing operating system 170, application programs 172, other program modules 174, and program data 176. Note that these components may either be the same as or different from operating system 144, application programs 146, other program modules 148, and program data 150. Operating system 170, application programs 172, other program modules 174, and program data 176 are given different numbers here to illustrate that, at a minimum, they are different copies.

[0044] A user may enter commands and information into computer 130 through input devices or user interface selection devices such as a keyboard 180 and a pointing device 182 (e.g., a mouse, trackball, pen, or touch pad). Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to processing unit 132 through a user input interface 184 that is coupled to system bus 136, but may be connected by other interface and bus structures, such as a parallel port, game port, or a Universal Serial Bus (USB). A monitor 188 or other type of display device is also connected to system bus 136 via an interface, such as a video interface 190. In addition to the monitor 188, computers often include other peripheral output devices (not shown) such as a printer and speakers, which may be connected through an output peripheral interface (not shown).

[0045] The computer 130 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 194. The remote computer 194 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 130. The logical connections depicted in FIG. 4 include a local area network (LAN) 196 and a wide area network (WAN) 198, but may also include other networks. LAN 136 and/or WAN 138 may be a wired network, a wireless network, a combination thereof, and so on. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and global computer networks (e.g., the Internet).

[0046] When used in a local area networking environment, computer 130 is connected to the LAN 196 through a network interface or adapter 186. When used in a wide area networking environment, computer 130 typically includes a modem 178 or other means for establishing communications over the WAN 198, such as the Internet. The modem 178, which may be internal or external, is connected to system bus 136 via the user input interface 184, or other appropriate mechanism. In a networked environment, program modules depicted relative to computer 130, or portions thereof, may be stored in a remote memory storage device (not shown). By way of example, and not limitation, FIG. 4 illustrates remote application programs 192 as residing on the memory device. The network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0047] Generally, the data processors of computer 130 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described herein.

[0048] For purposes of illustration, programs and other executable program components, such as the operating system, are illustrated herein as discrete blocks. It is recognized, however, that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

[0049] Although described in connection with an exemplary computing system environment, including computer 130, the invention is operational with numerous other general purpose or special purpose computing system environments or configurations. The computing system environment is not intended to suggest any limitation as to the scope of use or functionality of the invention. Moreover, the computing system environment should not be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment. Examples of well known computing systems, environments, and/or configurations that may be

suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, mobile telephones, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0050] The invention may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include, but are not limited to, routines, programs, objects, components, and data structures that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0051] An interface in the context of a software architecture includes a software module, component, code portion, or other sequence of computer-executable instructions. The interface includes, for example, a first module accessing a second module to perform computing tasks on behalf of the first module. The first and second modules include, in one example, application programming interfaces (APIs) such as provided by operating systems, component object model (COM) interfaces (e.g., for peer-to-peer application communication), and extensible markup language metadata interchange format (XMI) interfaces (e.g., for communication between web services).

[0052] The interface may be a tightly coupled, synchronous implementation such as in Java 2 Platform Enterprise Edition (J2EE), COM, or distributed COM (DCOM) examples. Alternatively or in addition, the interface may be a loosely coupled, asynchronous implementation such as in a web service (e.g., using the simple object access protocol). In general, the interface includes any combination of the following characteristics: tightly coupled, loosely coupled, synchronous, and asynchronous. Further, the interface may conform to a standard protocol, a proprietary protocol, or any combination of standard and proprietary protocols.

[0053] The interfaces described herein may all be part of a single interface or may be implemented as separate interfaces or any combination therein. The interfaces may execute locally or remotely to provide functionality. Further, the interfaces may include additional or less functionality than illustrated or described herein.

[0054] In operation, computer 130 executes computer-executable instructions such as those illustrated in the figures to update an automatically generated playlist in response to a content edit indicating media content to be added to or deleted from a collection of media files. Computer-executable instructions (e.g., as software, hardware, or both), a file system driver, input/output driver, and an electronic device constitute exemplary means for identifying the content edit and exemplary means for determining the data structures (e.g., playlists) affected by the identified content edit.

[0055] In the examples described herein, the media content of the digital media file is described in the context of content embodied on a CD or a DVD. It is to be appreciated and understood that the media content may be embodied on any suitable media and that the specific examples described herein are given to further understanding of the inventive principles. For convenience, a digital media file refers to one or more files representing, for example, a single song track or a collection of tracks such as would be found on an audio CD. The media content may include, without limitation, specially encoded media content (e.g., audio, video, or still images) in the form of an encoded media file.

[0056] The exemplary media file operations illustrated in the drawings and described herein are merely exemplary. Other variations of these file operations are within the scope of the invention. Alternatively or in addition, other media file operations not described herein yet embodying the invention are also within the scope of the invention.

[0057] The order of execution or performance of the methods illustrated and described herein is not essential, unless otherwise specified. That is, elements of the methods may be performed in any order, unless otherwise specified, and that the methods may include more or less elements than those disclosed herein. For example, it is contemplated that executing or performing a particular element before, contemporaneously with, or after another element is within the scope of the invention.

[0058] When introducing elements of the present invention or the embodiment(s) thereof, the articles "a," "an," "the," and "said" are intended to mean that there are one or more of the elements. The terms "comprising," "including," and "having" are intended to be inclusive and mean that there may be additional elements other than the listed elements.

[0059] In view of the above, it will be seen that the several objects of the invention are achieved and other advantageous results attained.

[0060] As various changes could be made in the above constructions, products, and methods without departing from the scope of the invention, it is intended that all matter

contained in the above description and shown in the accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

CLAIMS

What is claimed is:

1. In a media environment having playlists, said playlists being automatically generated from a plurality of media files stored on a computer-readable medium, a method for updating the playlists comprising:

identifying a content edit, said content edit indicating media content to be added to or deleted from the plurality of media files, said media content having metadata associated therewith;

determining, as a function of the metadata, one or more of the playlists affected by the identified content edit; and

updating the determined playlists per the received content edit, wherein said determining and said updating occur automatically in response to said identifying the content edit.

2. The method of claim 1, wherein determining the one or more playlists comprises identifying playlists having a predefined playlist type, said predefined playlist type corresponding to the metadata.

3. The method of claim 2, wherein the playlist type comprises an 8-bit value.

4. The method of claim 3, wherein the 8-bit value represents an integer.

5. The method of claim 2, further comprising:

rendering the media content from one of the updated playlists;

determining a string name associated with said one of the updated playlists as a function of the predefined playlist type; and

displaying the determined string name to a user.

6. The method of claim 1, wherein the identified content edit indicates that the media content is to be added to the computer-readable medium, and wherein updating the determined playlists includes reauthoring the determined playlists with the media content.

7. The method of claim 1, wherein the identified content edit indicates that the media content is to be deleted from the computer-readable medium, and wherein updating the determined playlists includes reauthoring the determined playlists without the media content.

8. The method of claim 1, further comprising:

automatically selecting one or more of the plurality of media files as a function of an item of the metadata;

creating a playlist with the selected media content, said created playlist including a playlist header;

determining a playlist type to be associated with the created playlist, said determined playlist type corresponding to the item of the metadata; and

adding the determined playlist type to the playlist header.

9. The method of claim 1, further comprising:

receiving a list of user selected content;

creating a playlist with the received list of user selected content; and

adding a playlist type to the created playlist, wherein the playlist type identifies the playlist as being user created.

10. The method of claim 1, wherein one or more computer-readable media have computer-executable instructions for performing the method recited in claim 1.

11. In a media environment having playlists, said playlists being automatically generated from a plurality of media files stored on a computer-readable medium, one or more computer-readable media having computer-executable components for updating the playlists, said components comprising:

an interface component for identifying a content edit, said content edit indicating media content to be added to or deleted from the plurality of media files, said media content having metadata associated therewith;

a table component for determining, as a function of the metadata, one or more of the playlists affected by the content edit identified by the interface component; and

an author component for updating the playlists determined by the table component per the received content edit, wherein the table component and the author component execute automatically in response to execution of the interface component.

12. The computer-readable media of claim 11, wherein the table component determines the one or more playlists by identifying playlists having a predefined playlist type.

13. The computer-readable media of claim 11, wherein the content edit identified by the interface component indicates that the media content is to be added to the computer-readable medium, and wherein the author component updates the playlists determined by the table component by reauthoring the playlists identified by the table component with the media content.

14. The computer-readable media of claim 11, wherein each of the playlists has an inconsistency flag associated therewith, and wherein the author component updates each of the playlists having an inconsistency flag that is set.

15. A system comprising:

a computer readable medium for storing:

a plurality of media files; and

a plurality of data structures each representing a playlist, each of said data structures storing a list of media files representing the playlist, said list of media files being selected from the plurality of media files, each of said data structures having a playlist type field storing a value identifying the playlist as a user-created playlist or an automatically generated playlist, each of said data structures further having an inconsistency field storing a value indicating whether the playlist is out-of-date with respect to the plurality of media files; and

a media device configured to execute computer-executable instructions for:

identifying a content edit, said content edit indicating media content to be added to or deleted from the plurality of media files stored on the computer-readable medium, said media content having metadata associated therewith;

determining, as a function of the metadata, one or more of the data structures affected by the identified content edit; and

setting the value in the inconsistency field for each of the determined data structures.

16. The system of claim 15, further comprising means for identifying the content edit.

17. The system of claim 15, further comprising means for determining the data structures affected by the identified content edit.

18. The system of claim 15, wherein the media device comprises a media player.

19. The system of claim 15, wherein one or more of the following stored on the computer readable medium are compressed: the plurality of media files and the plurality of data structures.

20. The system of claim 15, wherein the media device is further configured to execute computer-executable instructions for updating each of the determined data structures with the identified content edit.

FIG. 1

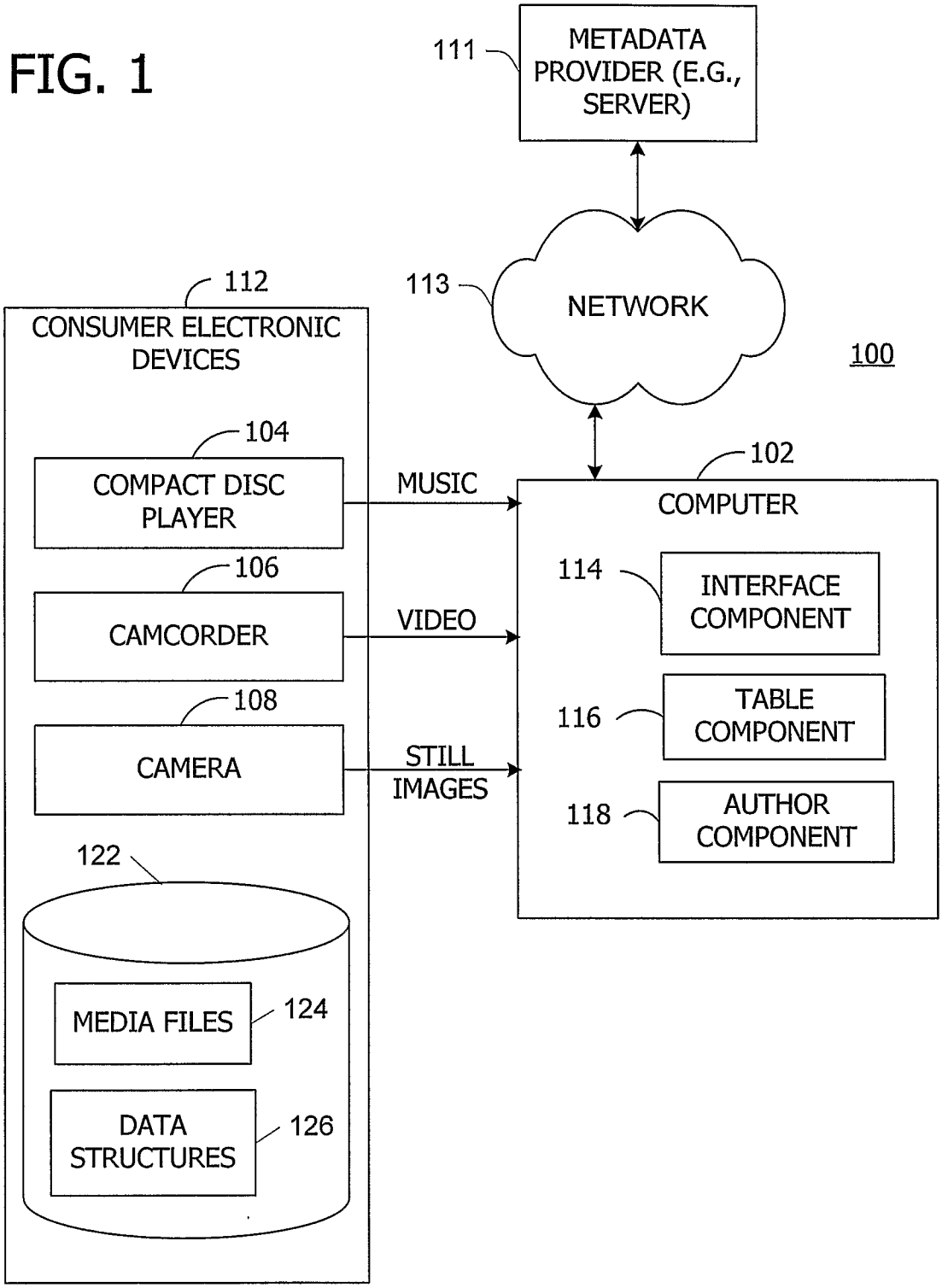
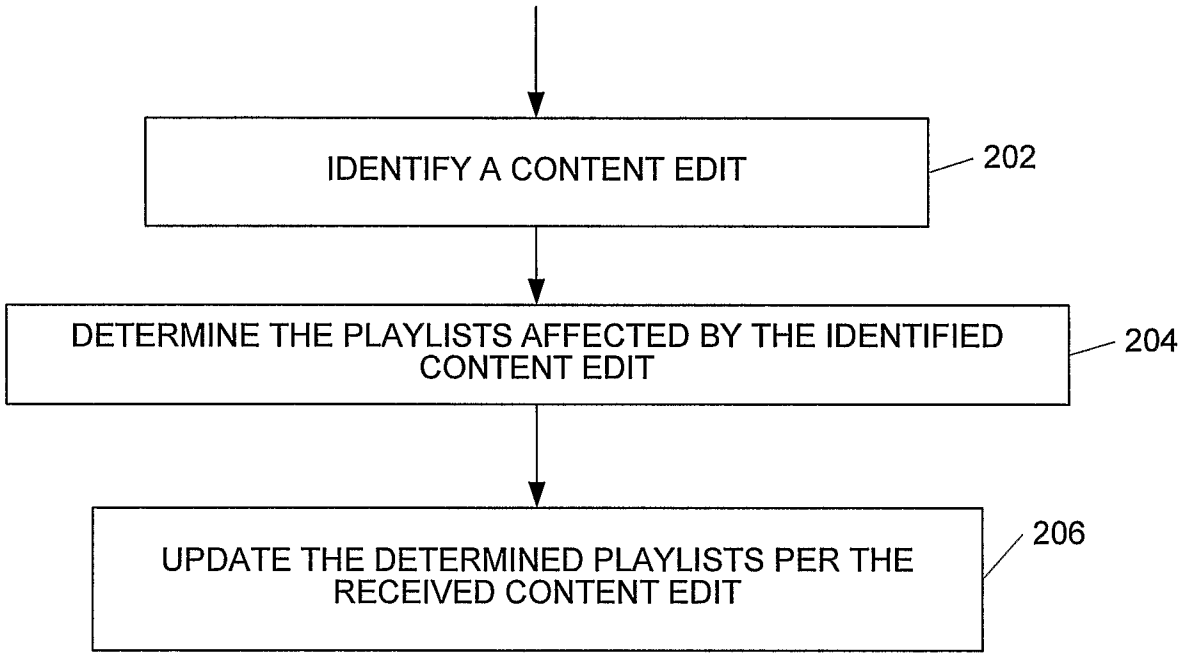

111
METADATA PROVIDER (E.G., SERVER)
113
NETWORK
100
112
CONSUMER ELECTRONIC DEVICES
104
COMPACT DISC PLAYER
106
CAMCORDER
108
CAMERA
MUSIC
VIDEO
STILL IMAGES
102
COMPUTER
114
INTERFACE COMPONENT
116
TABLE COMPONENT
118
AUTHOR COMPONENT
122
MEDIA FILES
124
DATA STRUCTURES
126

FIG. 2


IDENTIFY A CONTENT EDIT
202
DETERMINE THE PLAYLISTS AFFECTED BY THE IDENTIFIED CONTENT EDIT
204
UPDATE THE DETERMINED PLAYLISTS PER THE RECEIVED CONTENT EDIT
206

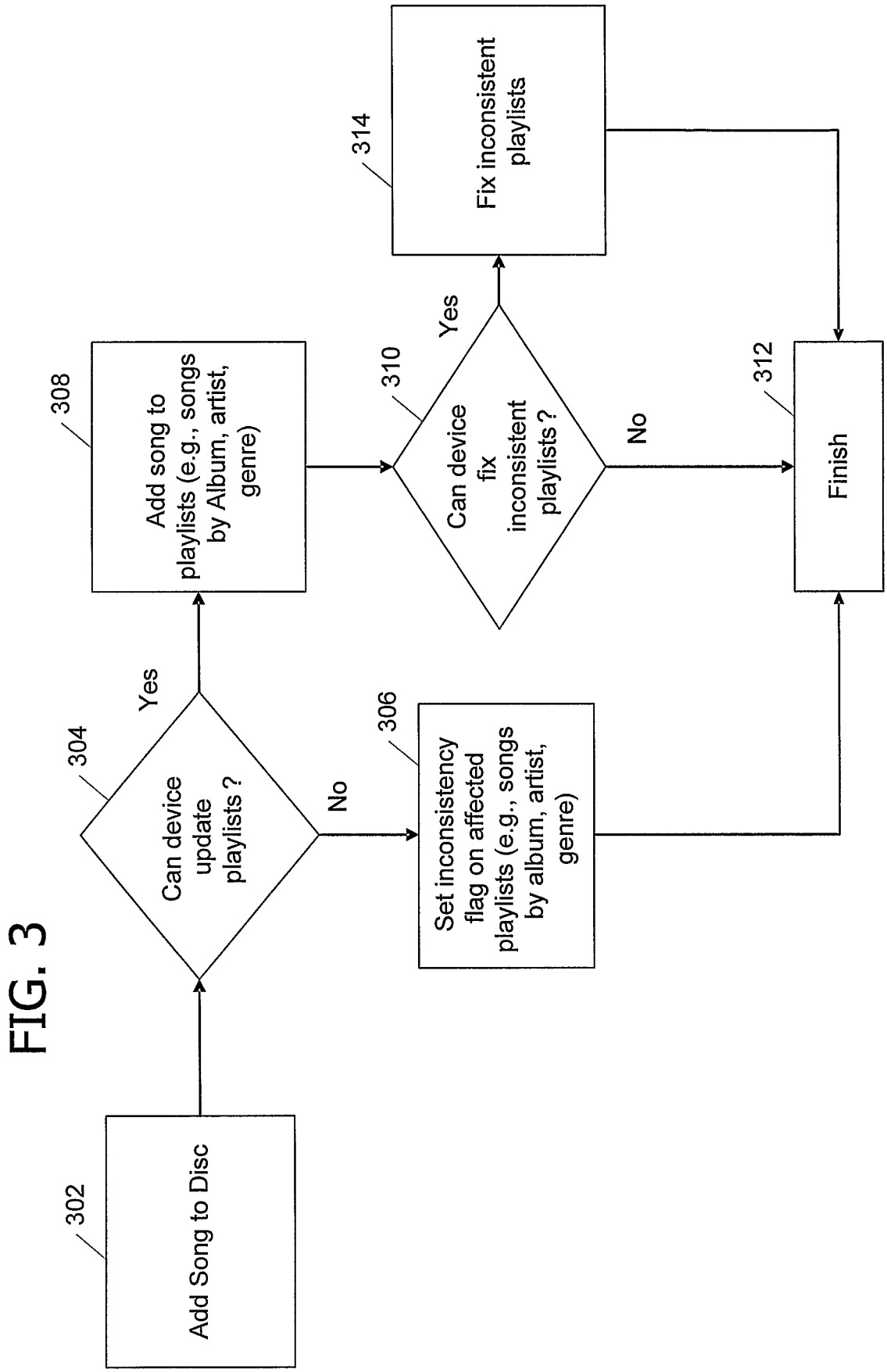

302
Add Song to Disc
304
Can device update playlists ?
Yes
No
306
Set inconsistency flag on affected playlists (e.g., songs by album, artist, genre)
308
Add song to playlists (e.g., songs by Album, artist, genre)
310
Can device fix inconsistent playlists ?
Yes
No
312
Finish
314
Fix inconsistent playlists

FIG. 3

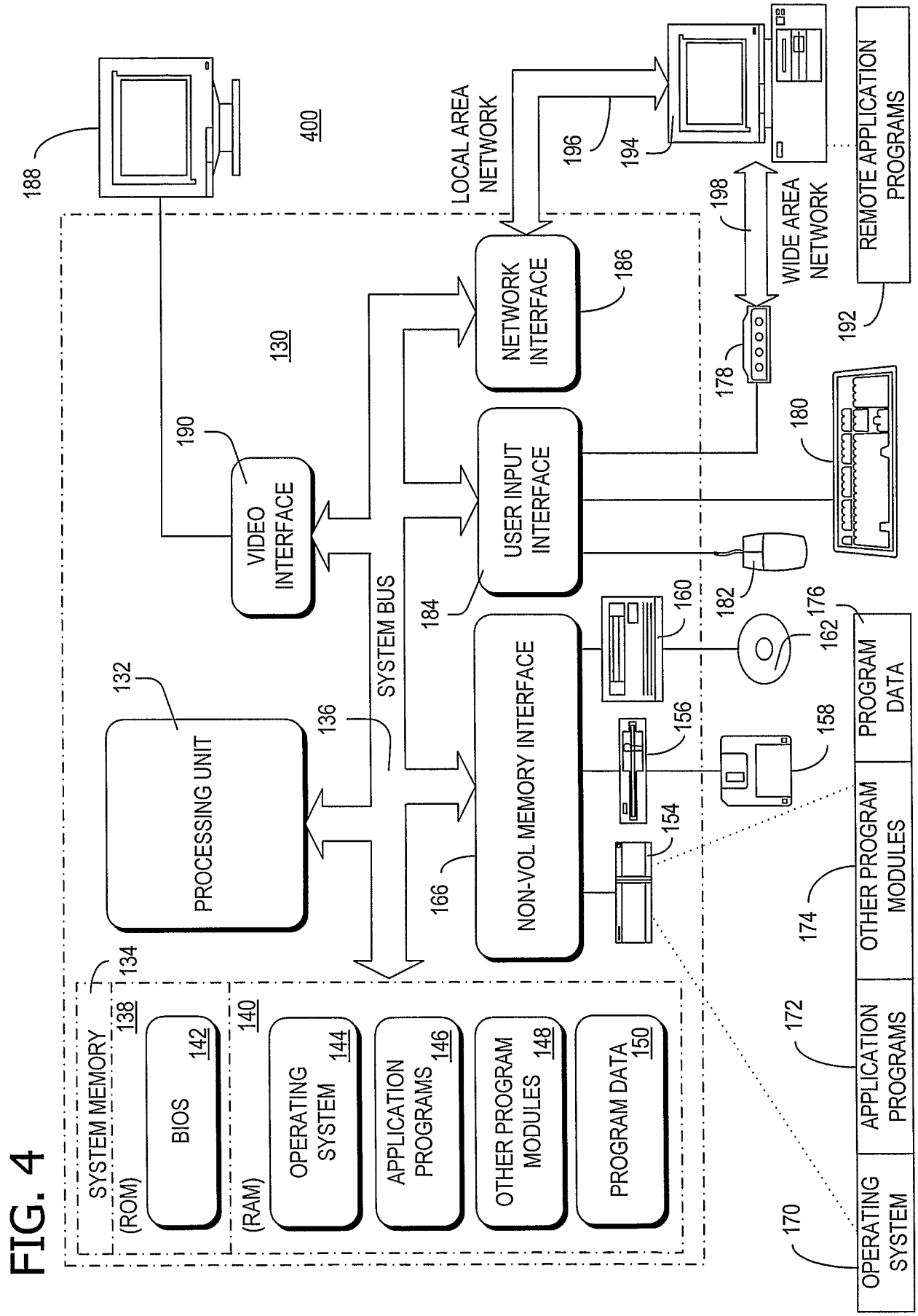

400
188
LOCAL AREA NETWORK
196
194
198
WIDE AREA NETWORK
REMOTE APPLICATION PROGRAMS
192
178
180
130
NETWORK INTERFACE
186
190
VIDEO INTERFACE
USER INPUT INTERFACE
184
182
160
176
162
132
PROCESSING UNIT
136
SYSTEM BUS
NON-VOL MEMORY INTERFACE
156
158
154
166
PROGRAM DATA
OTHER PROGRAM MODULES
174
APPLICATION PROGRAMS
172
OPERATING SYSTEM
170
134
SYSTEM MEMORY (ROM)
138
BIOS 142
140
(RAM)
OPERATING SYSTEM 144
APPLICATION PROGRAMS 146
OTHER PROGRAM MODULES 148
PROGRAM DATA 150

FIG. 4