



(86) Date de dépôt PCT/PCT Filing Date: 1999/02/18
(87) Date publication PCT/PCT Publication Date: 1999/08/26
(45) Date de délivrance/Issue Date: 2007/12/18
(85) Entrée phase nationale/National Entry: 2000/08/17
(86) N° demande PCT/PCT Application No.: US 1999/003512
(87) N° publication PCT/PCT Publication No.: 1999/043159
(30) Priorité/Priority: 1998/02/20 (US60/075,412)

(51) Cl.Int./Int.Cl. *H04N 7/16* (2006.01),
H04N 5/445 (2006.01), *H04N 7/24* (2006.01)

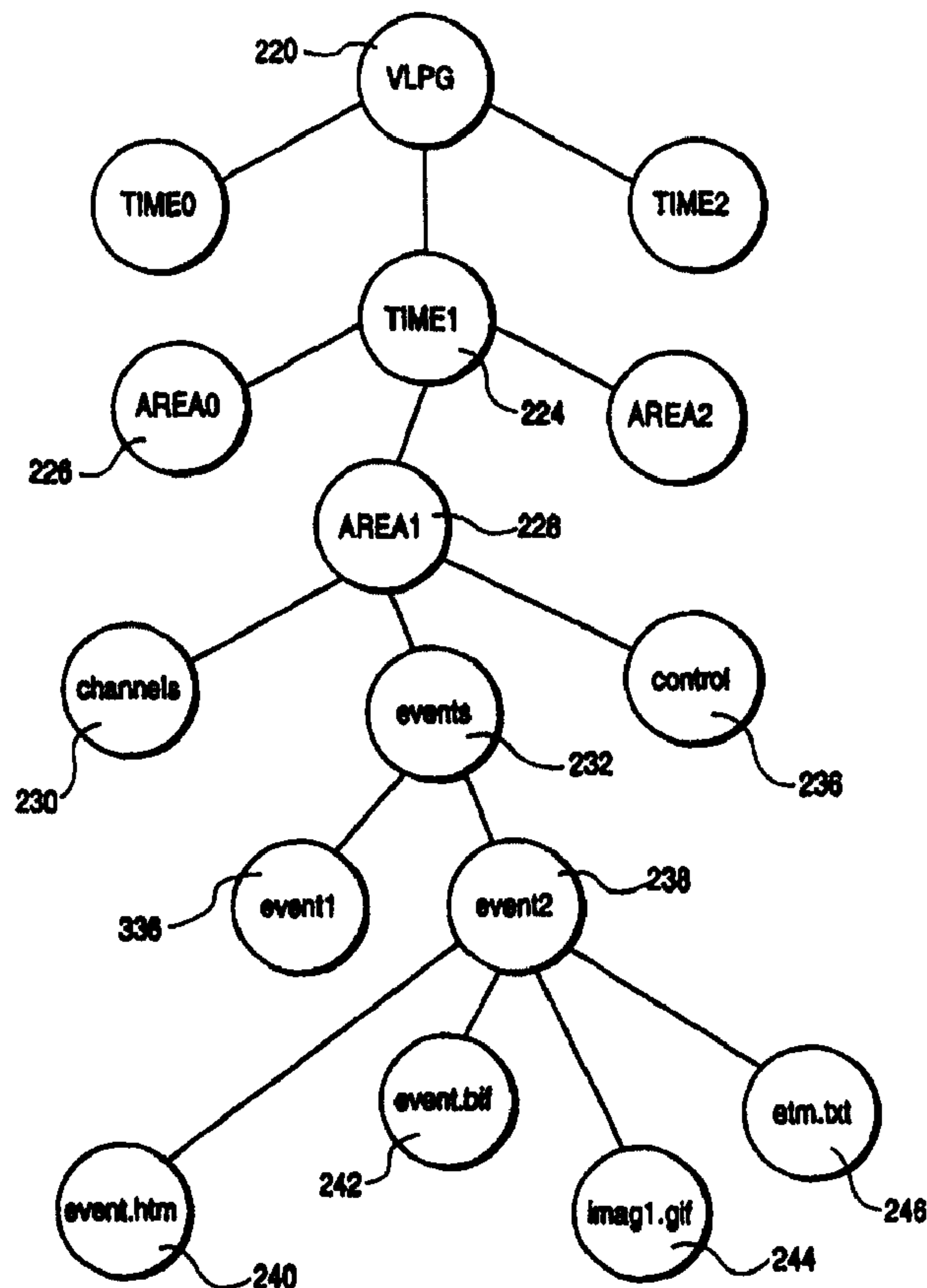
(72) Inventeurs/Inventors:
OZKAN, MEHMET KEMAL, TR;
TENG, CHIA-YUAN, US;
HEREDIA, EDWIN ARTURO, US

(73) Propriétaire/Owner:
THOMSON LICENSING S.A., FR

(74) Agent: CRAIG WILSON AND COMPANY

(54) Titre : SYSTEME DE FORMATION, DE PARTITIONNEMENT ET DE TRAITEMENT DE GUIDES ELECTRONIQUES
DE PROGRAMMES

(54) Title: SYSTEM FOR FORMING, PARTITIONING AND PROCESSING ELECTRONIC PROGRAM GUIDES



(57) Abrégé/Abstract:

A program guide information data structure and processing system facilitates decoding. A decoder acquires ancillary information conveyed in hierarchically ordered data tables comprising packetized program guide data. The ancillary information includes



(57) **Abrégé(suite)/Abstract(continued):**

hierarchically associated version identifiers enabling the decoder to identify change in data table content by examination of the version identifiers in hierarchical order. The decoder acquires cell numbers (e.g. identifying cell type, area, broadcast time and complexity level) assigned to individual partitions of the program guide information in order to adapt to dynamically re-partitioned program guide data.

**PCT**WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

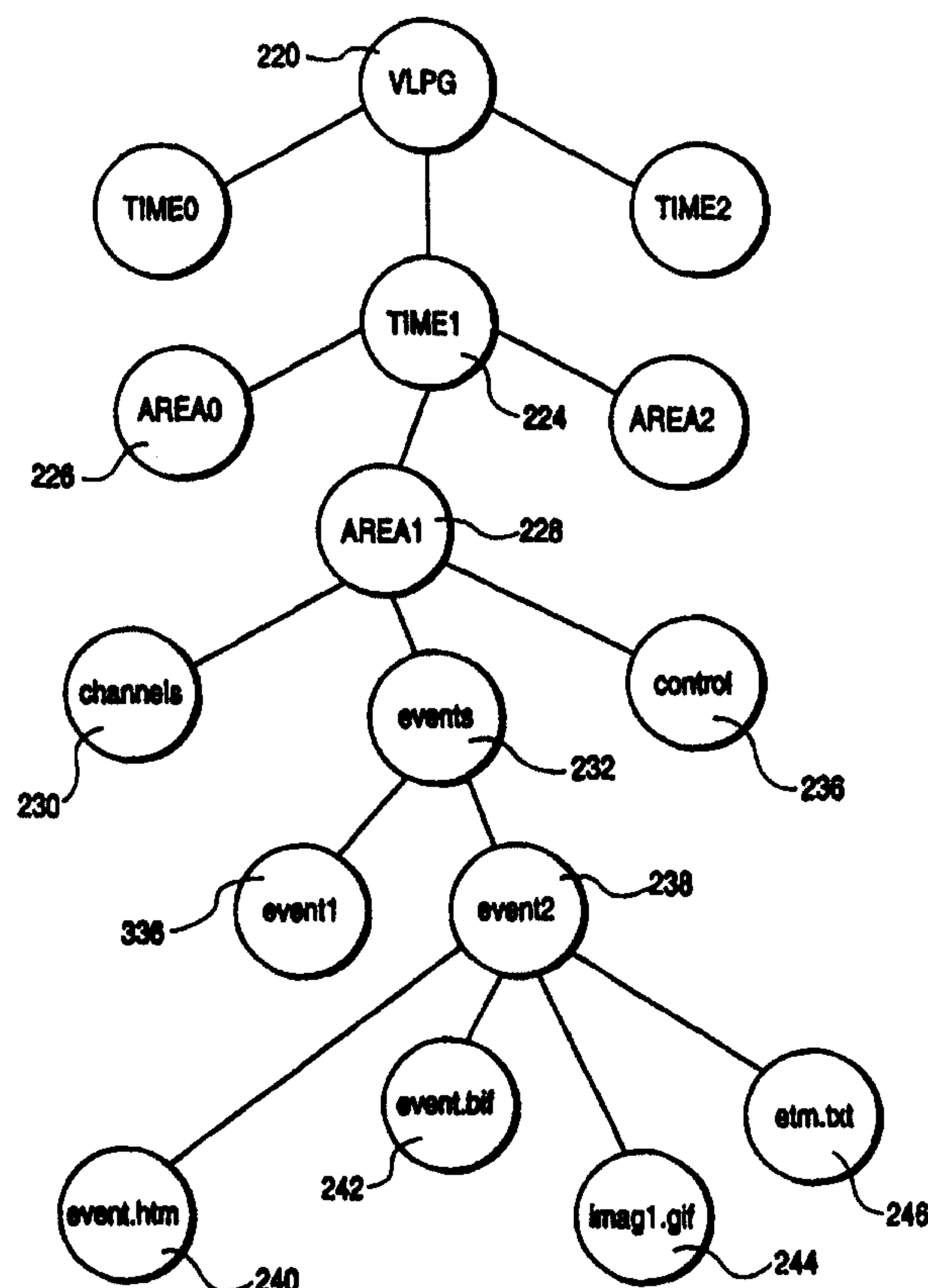
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| | | |
|--|-----------|--|
| (51) International Patent Classification ⁶ : H04N 7/16 | A1 | (11) International Publication Number: WO 99/43159 (43) International Publication Date: 26 August 1999 (26.08.99) |
| (21) International Application Number: PCT/US99/03512 (22) International Filing Date: 18 February 1999 (18.02.99) (30) Priority Data: 60/075,412 20 February 1998 (20.02.98) US (71) Applicant (for all designated States except US): THOMSON CONSUMER ELECTRONICS, INC. [US/US]; 10330 North Meridian Street, Indianapolis, IN 46290 (US). (72) Inventors; and (75) Inventors/Applicants (for US only): OZKAN, Mehmet, Kemal [TR/TR]; Savasci Sok. Bozokatt 19/1, Avcilar, 34840 Istanbul (TR). TENG, Chia-Yuan [-/US]; 7384 Celata Lane, San Diego, CA 92129 (US). HEREDIA, Edwin, Arturo [BO/US]; Apartment 4015, 8274 Lakeshore Circle, Indianapolis, IN 46250 (US). (74) Agents: TRIPOLI, Joseph, S. et al.; Thomson Multimedia Licensing Incorporated, P.O. Box 5312, Princeton, NJ 08540 (US). | | (81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i> |

(54) Title: SYSTEM FOR FORMING, PARTITIONING AND PROCESSING ELECTRONIC PROGRAM GUIDES

(57) Abstract

A program guide information data structure and processing system facilitates decoding. A decoder acquires ancillary information conveyed in hierarchically ordered data tables comprising packetized program guide data. The ancillary information includes hierarchically associated version identifiers enabling the decoder to identify change in data table content by examination of the version identifiers in hierarchical order. The decoder acquires cell numbers (e.g. identifying cell type, area, broadcast time and complexity level) assigned to individual partitions of the program guide information in order to adapt to dynamically re-partitioned program guide data.



SYSTEM FOR FORMING, PARTITIONING AND PROCESSING ELECTRONIC PROGRAM GUIDES

Field of the Invention

10 This invention is related to the field of digital signal processing, and more particularly to program guides for channels and programs.

Background of the Invention

15 The formation and processing of large program guides conveying information concerning potentially thousands of broadcast program channels covering a wide geographic area poses a number of problems. The geographic area covered may
20 encompass the whole of the USA or whole continents, for example, and large quantities of information may have to be acquired, collated, encoded and broadcast in a format that facilitates subsequent decoding of the broadcast material. The bandwidth required to process such large quantities of information expands
25 in proportion to the quantity of information being processed. Therefore, there is a need to structure program guide data in order to optimize the use of the available bandwidth.

 The degree to which the program guide data structure may be optimized is constrained by the cost of a decoder unit for
30 receiving the structured data. In fact there is a compromise to be made between transmission bandwidth and decoder complexity. At one extreme of the compromise, all duplicative and redundant data elements in the program guide information are eliminated in order to minimize the required transmission and processing
35 bandwidth. As a result, each decoder needs to receive, buffer, parse and collate information from an entire program guide datastream, thereby necessitating a complex and costly decoder.

 At the other extreme of the compromise, the program guide information is partitioned into individual sub-sets tailored
40 to the requirements of a particular User or group of Users. This

2

5 means that each decoder needs to receive, buffer, parse and
collate targeted information containing minimal redundancy which
facilitates employing a simpler, more economical decoder
requiring less processing power. However, such partitioning
requires a larger transmission bandwidth to accommodate the
10 increased information redundancy resulting from the need to
incorporate duplicate program guide information items in multiple
different program guide sub-sets corresponding to different
partitions. The problems involved in processing large quantities of
program guide information and in achieving a desirable
15 compromise between transmission bandwidth and decoder
complexity are addressed by a system according to the present
invention. Derivative problems involved in structuring and
partitioning program guide data to facilitate both decoding and
selectable program guide generation by a decoder are also
20 addressed by a system according to the invention.

Summary of the Invention

A decoder acquires ancillary information conveyed in
25 hierarchically ordered data tables comprising packetized program
guide data. The ancillary information includes hierarchically
associated version identifiers enabling the decoder to identify
change in data table content by examination of the version
identifiers in hierarchical order. The decoder acquires cell
30 numbers (e.g. identifying cell type, area, broadcast time and
complexity level) assigned to individual partitions of the program
guide information in order to adapt to dynamically re-partitioned
program guide data.

35 *Brief Description of the Drawings*

In the drawing:

Figure 1 shows a very large program guide (VLPG)
hierarchical file/table format for use in conveying program
40 specific information, according to the invention.

5

Figure 2 shows a Master Guide Table (MGT) format for use in conveying program specific information, according to the invention.

10

Figure 3 shows a Channel Information Table (CIT) format for use in conveying program specific information incorporating area based partitioning, according to the invention.

Figure 4 shows a multimedia object data structure format incorporating area and time based partitioning, according to the invention.

Figures 5, 6 and 7 show examples of data structures for channel, event and control object basic information files, according to the invention.

Figure 8 shows a data structure for an MPEG compatible carouselId (as used in the tables of Figures 5, 6 and 7) including identifier fields allowing area and time based partitioning.

Figure 9 shows a Master Database Table data structure format incorporating hierarchical based version identifiers and cell partition identifiers supporting dynamic program guide re-partitioning, according to the invention.

Figure 10 shows an exemplary data structure for a cell type indicator (as used in the table of Figure 9) incorporating area, time and complexity based identifier fields, according to the invention.

Figure 11 shows a hierarchical directory format for an object database including channel, event and control sub-directories, according to the invention.

40

5 Figure 12 shows a method for generating program specific information according to the invention.

 Figure 13 is a block diagram of digital video receiving apparatus for demodulating and decoding broadcast signals
10 containing VLPG information, according to the principles of the invention.

Detailed Description of the Drawings

15 Broadcast programs transmitted in digital format are encoded and broadcast along with ancillary information including program specific information (PSI) used in decoding programs and associated data. Program specific information includes program guide data and information for use in identifying and assembling
20 individual data packets to recover the content of selected program channels. Program specific information and associated program content is advantageously structured to convey large program guides conveying information concerning potentially thousands of broadcast program channels and associated multimedia objects
25 covering a wide geographic area such as whole continents, countries or states, for example. The multimedia objects include audio clips, video clips, animation, still images, Internet data, Email messages, text and other types of data. Multimedia objects are data entities that may be viewed as independent units and are
30 associated with images within individual programs or with program guide components. The multimedia objects are incorporated into composite video images representing a program guide or a video program, for example. The ancillary information data structure supports uni-directional communication
35 applications e.g. passive viewing and bi-directional communication applications e.g. interactive type functions and also supports storage applications.

 The program specific information and associated program content may be delivered by different service providers
40 via the Internet in broadcast/multicast mode, or via terrestrial,

5 satellite or cable broadcast on a subscription or other pay per view basis. The data structure facilitates acquisition and decoding of multimedia objects encoded in different data formats and which are communicated in different communication protocols from both local and remote sources.

10 Hereinafter, data referred to as being MPEG compatible conforms to the MPEG2 (Moving Pictures Expert Group) image encoding standard, termed the "MPEG standard". This standard is comprised of a system encoding section (ISO/IEC 13818-1, 10th June 1994) and a video encoding section (ISO/IEC 13818-2, 20th 15 January 1995).

Data structure elements according to the invention principles may be conveyed in MPEG compatible format (per section 2.4.4 of the MPEG systems standard) or may be conveyed in a format compatible with the *Program and System Information* 20 *Protocol for Terrestrial Broadcast and Cable*, published by the Advanced Television Systems Committee (ATSC), 10 November 1997, hereinafter referred to as the PSIP standard, or other ATSC standards. Further, the data structure elements may be formed in accordance with other MPEG standards such as the MPEG-4 or 25 MPEG-7 standards or with the proprietary or custom requirements of a particular system.

The principles of the invention may be applied to terrestrial, cable, satellite, Internet or computer network broadcast systems in which the coding type or modulation format 30 may be varied. Such systems may include, for example, non-MPEG compatible systems, involving other types of encoded datastreams and other methods of conveying program specific information. Further, although the disclosed system is described as processing broadcast programs, this is exemplary only. The term 'program' is 35 used to represent any form of packetized data such as audio data, telephone messages, computer programs, Internet data or other communications, for example.

Figure 1 shows an overview of a very large program guide (VLPG) hierarchical file/table format for use in a transport 40 stream level data structure conveying program specific

5 information. The structure comprises multiple hierarchically arranged and inter-linked tables. The tables consist of arrays of data and parameters which are used to enumerate and describe collections or sequences of TV channels, TV programs, channel parameters, program parameters, associated multimedia objects
10 and object parameters, etc. The exemplary hierarchical table arrangement of Figure 1 includes a Master Guide Table (MGT) 120, Master Database Table (MDBT) 122, Content and Classification Table (CCT) 114, System Time Table (STT) 116 and Rating Region Table (RRT) 118. The Figure 1 hierarchy also shows Terrestrial,
15 Cable and Satellite Channel Information Tables (TCIT item 112, CCIT item 110, and SCIT item 108 respectively) in which channel information is collated by network provider e.g. CBS, NBC, HBO, Comcast etc. Additional tables include Schedule Information Tables (SIT 106, SIT 104 and SIT 102) in which programs or
20 services are collated by source.

A MGT contains information for use in acquiring program specific information conveyed in other tables. A channel information table - CIT (e.g. TCIT, CCIT, OR SCIT) contains information for tuning and navigation to receive a User selected
25 program channel. A SIT contains descriptive lists of programs (events) receivable on the channels listed in the CIT. Either a CIT, SIT or other table may be used to convey information enabling a user to select and tune to a particular program. A CIT is typically used to convey parameters for acquiring audiovisual program
30 content data that remains constant over several events (TV programs). An SIT is typically used to convey parameters of audiovisual program content data that remain constant for an event (individual TV program). Additional program specific information describing and supplementing items within the
35 hierarchical tables is conveyed within descriptor information elements.

In order to accommodate data sufficient for a wide area program guide, individual tables in the hierarchy of Figure 1 are advantageously partitionable by both area (e.g. a geographic,
40 broadcast, or network market area), and scheduled broadcast

5 time. In addition, the table data may be additionally partitioned in accordance with a third parameter such as the complexity level of data or a multimedia object in order to permit scalability in decoding, for example. This type of program guide partitioning is represented by the 3-dimensional cells depicted in diagram 100.

10 In fact, the VLPG structure of Figure 1 supports virtually any type of partitioning that a guide provider may require. The area and time based partitioning is achieved by including both time and area identifiers in one or more of the tables of Figure 1 and in associated object data. The advantage of including the area and

15 time partition identifiers in the transport level of the communication protocol in this manner is that it reduces the processing burden (e.g. for parsing the data) on a decoder that otherwise occurs if the partitioning is done at a higher level, say the Application layer level. As a result program guide filtering is

20 done at the transport layer and is performed directly by a transport chip contained in a decoder unit. However, it is possible to include the time and area identifiers for both table data and objects at such a higher level if desired.

In transmitting and processing a program guide, the

25 bandwidth demand grows with the quantity of channel and program information and number of associated objects that need to be transmitted. For a large program guide, even a simple case may require transmission of thousands of items of information and objects. A simple program guide may have neither images,

30 nor audio, nor video clips, but would still need at least text descriptions for the thousands of programs (events) that it carries. It is possible to send the program guide information without redundancy and without partitions in a single bin or data file. In such a case, and in the absence of transport-level partitions, the

35 text descriptions (for example) will end up in a single data file. This means, for example, that a decoder in San Diego will receive all the material (event text descriptions, images, or any other objects) from all the other cities in the US and will be unable to discard useless material at the transport level. Consequently, in

40 the absence of transport-level partitions, it is necessary for a

5 decoder to filter received program guide information at the application level. This is a processor intensive, time-consuming and burdensome task that requires sophisticated software and significant processing power and raises the cost of a decoder unit.

The VLPG data structure of Figure 1 advantageously
10 provides the option of employing area-based partitioning at the transport layer level. Program guide information may be partitioned in an Eastern area, a Central area, a Mountain area, and a Pacific area, for example. Then a decoder in San Diego would no longer need to receive program guide information from the
15 other 3 regions. Therefore, such partitioning significantly reduces the parsing and filtering burden on a decoder and smaller partitions (e.g. on a state by state basis) further reduce the parsing and filtering burden. Another advantage of such partitioning is that the time involved in downloading applicable
20 partitioned program guide information is reduced.

However, partitioning program guide information involves introducing redundant program guide data because such partitioning requires duplication of data items. As an example, if a basketball game is aired in the Pacific and Mountain regions, then
25 two copies of associated text descriptive information need to be transmitted, one copy for each of the partitions targeted at the pacific and mountain regions. It can be seen that, as the number of partitions increases, the quantity of redundant information also increases necessitating larger transmission bandwidth.
30 Consequently, there is a trade-off between bandwidth and the burden of information filtering. A large number of partitions imply fast information filtering but at the cost of increased bandwidth. When there is only one partition, there are no redundancies and therefore bandwidth is minimal but the filtering
35 burden is larger since all the program guide information items need to be parsed.

Time and area cells may be mapped into an MPEG-2 compatible data structure by using the MPEG2 PSI and DSM-CC fields. Not all of the tables may need to include area-based
40 identifiers. Program content ratings, for example, are typically

5 applicable everywhere in the US. In the VLPG architecture of Figure 1, significant advantage is obtained by applying area-based partitioning to the Channel Information Table (CIT) and the Master Guide Table. A CIT defines the channel line-up (list of available channels) for a service provider and depends on the
10 geographical area covered by the service provider. The channel line-up for terrestrial broadcast in Indianapolis is different than the channel line-up for a cable provider in Philadelphia, for example. In the system of Figure 1 the MGT also depends on geographical area but this is not necessarily the case.

15 The capability of performing focused targeting of program guide information to particular audiences is one advantage derived from being able to make fine area-based divisions of channel line-ups at the broadcast market level. In order to accomplish this, different table "instances" are created. A
20 table instance is a version of a table that is targeted to a particular market area and incorporates an area identifier for identifying the applicable market area. Multiple instances of a single table can be concurrently transmitted each carrying different information. Different table instances are recognized using the
25 "table_id_extension" field of the MPEG-2 protocol.

Figures 2 and 3 show a Master Guide Table (MGT) format and Channel Information Table (CIT) format respectively for use in conveying program specific information and incorporating a table_id_extension field for market area
30 identification. In the MGT of Figure 2 and in the CIT of Figure 3, this market area identification field is termed a "network_provider" and is shown in the MGT data structure as item 130, and in the CIT data structure as item 140. The network_provider area identifier field is a 16-bit field used to
35 uniquely identify a network provider. The meaning of the network provider depends on the transmission medium. Specifically, for terrestrial broadcast, a network provider is a collection of stations within a geographical region, for cable broadcast, a network provider is a local cable service provider and

10

5 for satellite broadcast, a network provider is a satellite service provider.

The data structure of Figure 1 advantageously allows different types of program specific and program guide information to be targeted to different areas. This feature permits flexibility in
10 selecting an acceptable compromise between decoder complexity and processing bandwidth involved in broadcasting and receiving the program guide data. As an example, it may be acceptable to partition multimedia objects into coarser areas than channel line-up information. The data structure of Figure 1 gives the guide
15 providers the capability of partitioning different types of data in different gradations of area ranging from coarse areas to fine areas (e.g. areas as large as countries, states, or counties ranging to areas as fine as cities, towns, city blocks or even individual customers).

20 Further, program guide information may be collated in a decoder to provide a User a choice between program guides for different areas (e.g. between two neighboring areas or a choice of guides from any of the available areas) or for different periods of broadcast time. As such, a program guide may be selected in a
25 decoder from one or more available program guides associated with different areas, in response to a User selection input via a remote unit or other data entry device. In performing such a selection, a decoder compares a region identification designation (associated with received program guide information) with a pre-
30 stored region identification designation representing the decoder location. Such a region identification designation may comprise, a zip code, a telephone area code, and any other region identification code.

Figure 4 shows a multimedia object data structure
35 format for conveying objects within a VLPG. The multimedia object data structure supports area and time based partitioning through the use of area and time identifier fields within an MPEG DSM-CC compatible carouselId identifier (item 150 in Figure 4).

Figures 5, 6 and 7 show examples of the data structure
40 of objects comprising channel, event and control objects

11

5 respectively. Specifically, Figure 5 shows a Channel Basic Information File (channel BIF) binary file, Figure 6 shows an Event Basic Information File (Event BIF) binary file and Figure 7 shows a Control Basic Information File (Control BIF) binary file. In similar fashion to Figure 4, the channel, event and control object data of
10 Figures 5-7 include area and time identifier fields within MPEG DSM-CC compatible carouselId identifiers (items 153, 157 and 159 of Figure 5-7 respectively).

Figure 8 item 152, shows an exemplary carouselId data structure for an MPEG compatible carouselId (as used in the
15 tables of Figures 4, 5, 6 and 7). The carouselId includes a 16 bit directory item identifier for use as a data base reference, an 8 bit time identifier and an 8 bit area identifier. These fields enable a decoder to selectively filter program guide data based on area and time based partitions.

20 In the VLPG data structure of Figure 1, a Master Database Table (MDBT item 122) is used to define program guide partitions (cells) and to inform a decoder of the cells available in its location for decoding. Figure 9 shows a Master Database Table data structure incorporating hierarchical based version identifiers
25 and cell partition identifiers advantageously enabling dynamic program guide re-partitioning. The code between items 170 and 178 comprises a loop that defines partitioned cells. Within this loop, the field called "cell_type" (item 172) determines an index of a cell as illustrated in Figure 10.

30 Figure 10 item 179 shows an exemplary data structure for a cell type indicator. The cell type indicator includes an 8 bit complexity level identifier used to define a complexity level of an object. The cell type indicator also includes an 8 bit area identifier and an 8 bit time identifier for defining area and time based
35 partitions.

An individual object or program guide information item includes a carouselId (as defined in Figure 8 and as shown item 150 in Figure 4) for linking the object to a time and area index of its mother cell. Dynamic program guide re-partitioning is
40 achieved by re-arranging the cell list in the MDBT and by dynamic

12

5 alteration in the mother cell time and area index identifiers within
 a carouselId. Consequently, a guide provider is able to
 dynamically re-partition a program guide data structure to adapt
 to changes in available transmission bandwidth or decoder
 sophistication. A guide provider may use finer partitions to
 10 provide faster object filtering times if increased bandwidth
 becomes available or may use coarser partitions to preserve
 bandwidth if decoder processing capabilities improve. As such, a
 decoder in San Diego may today have access to program guide
 information associated with areas 0 and 7 while in the future the
 15 decoder may be given access to program guide information
 associated with areas 0 and 9, reflecting finer partitioning, for
 example. In the future, it is entirely feasible to be able to provide
 real-time access to multimedia objects present in guides covering
 areas as large as the USA by selecting an appropriate compromise
 20 between bandwidth and partitioning.

Figure 11 shows a hierarchical directory format for an
 object database including object files under channel, event and
 control sub-directories. Figure 11 shows that a particular event
 such as "event2" can have its own directory containing its
 25 required files, e.g., "event2" has four associated objects (items 240,
 242, 244 and 246 in Figure 11). Using the object database a
 directory-based address is mapped into transport-level fields for
 processing. Given a path such as
 /VLPG/TIME1/AREA1/events/event2/event.bif, (path through
 30 directory items 220, 224, 228, 232, 238, 242 of Figure 11) for
 example, there is one and only one object with a certain
 carouselId and moduleId. For example, in this case, the following
 mapping applies:

35 /VLPG -----> Files extracted from VLPG object data base
 /TIME1 -----> 0x01 (time variable)
 /AREA1 -----> 0x01 (area variable)
 /events/event2 ---> 0x3005 (directory number variable).

5 Therefore, this database address is mapped to a carouselId of 0x01013005. Further, the event.bif (item 242) has a moduleId which may be determined from the program guide information (such as value Cx0002 in this example). The inverse mapping from transport-level fields to a directory-based address
10 is also unique and may be similarly derived.

 The directory structure exemplified in Figure 11 supports the operation of software for processing and interacting with transmitted objects. The processing software may be transmitted in file form, together with objects, and then
15 interpreted or compiled and run by a decoder. Such processing software may be used for numerous applications including for advantageously creating content-based program guides in a decoder in a language such as HTML (Hyper Text Mark-up Language), SGML (Standardized Generalized Mark-up Language),
20 Java, ActiveX and any other decoder supported language. As an example, each Sunday a guide provider may want to prepare a web site describing all available Sunday movies for transmission to decoder units containing a web browser and supporting HTML software. The Sunday guide information is encoded in HTML and
25 transmitted to decoders as HTML software files that describe the special web site. The HTML files are located anywhere in the directory structure of the Object Database and together generate images, text, video, and audio files that form the Sunday movie guide listing comprising the special web site. As such, the Sunday
30 guide web site is broadcast as part of an object database and is not conventionally accessed from an Internet server. Other special program guide web sites that may be conveyed in this way may list, for example, (a) a DVD/VCR programs available for play, (b) other Internet web sites, (c) pre-stored fax/phone numbers for
35 access, (d) videophone functions and (e) home appliance control functions.

 A User may initiate display of the special Sunday guide broadcast-web site contained in the Object Database by using a remote unit or other data entry device to select an
40 associated displayed menu item or display icon and may similarly

14

5 navigate the broadcast-web site and review the advertised movies. In addition, a User may be able to initiate commands via the web site such as (a) programming a VCR or DVD unit, (b) tuning to a desired channel, or (c) accessing other Internet sites that are either similarly broadcast as part of the object database
10 or that are conventionally accessed via telephone (or cable) line. Further, in initiating such telephone (or cable) line Internet access a decoder may acquire access information from a broadcast or other source. Such access information includes (a) an Internet URL, (b) an Internet IP address, (c) an Email address, and (d) a
15 telephone/fax/videophone number, for example.

The hierarchical directory structure of Figure 11 illustrates another advantage offered by the VLPG structure of Figure 1. In a large program guide, managing the update of constituent tables and objects involves examining a large quantity
20 of version numbers (thousands of version numbers may be involved). A decoder downloads a particular program guide table or object upon determination of a version number change and disregards those tables or objects where no version number change is indicated. This task may be aided by listing all the
25 version numbers of tables and objects that are susceptible to change in one table such as a MGT. The MGT is broadcast at a sufficiently fast rate to enable a decoder to examine the MGT to determine which tables or objects have changed and to acquire a changed table or object in a timely fashion. However, parsing
30 every entry in the MGT may become very time consuming in a large guide structure.

This problem is addressed by employing a system of hierarchical version control in which there are several tables that perform version control. These tables are arranged in a tree
35 structure as exemplified by Figure 11. In Figure 11, table 238 (event2) controls the version of those tables/files under table 238. Table 232 (events) controls all the versions of tables 336 (event1), and 238. Table 228 (AREA1) controls the versions of tables 230 (channels), 232 and 236 (control) only. In this way, the version
40 number information stored in each table is small and by

15

5 traversing the tree from top to down, it is possible to quickly find those files, tables or objects that need to be updated.

Although a multiple-level structure is described above, a two layer version control structure may also be used for version control of objects in the database in the VLPG of Figure 1. In the
10 two layer example, the upper layer of the tree hierarchy is the Master Database Table (MDBT) as exemplified in Figure 9. The second level, below the MDBT level, consist of cells comprising Channel, Event and Control Basic Information Files, as exemplified by the previously described data structures of Figures 5, 6 and 7
15 respectively. A change in any of the Channel, Event or Control files is signaled by a change in their respective version numbers, item 160 (Figure 5), item 163 (Figure 6) and item 167 (Figure 7). Further, any change in the Channel, Event or Control file version numbers is signaled by a change in the next hierarchical (cell)
20 level version number i.e. such a change is signaled by a change in version number 176 in the MDBT of Figure 9. As a specific example, if a channel logo (which is an image in the database) changes from one version to another, then the channel BIF will reflect this change in item 160 (Figure 5). The MDBT will also
25 signal the change in item 176 at the cell level (Figure 9). A decoder examines first the MDBT and determines the cell version number has changed and then examines the BIF files to identify objects that have changed within the cell.

The Master Data Base Table (MDBT) structure of Figure
30 9 provides another advantage in processing large program guides. Decoder sophistication and processing power and ability to process complex multimedia objects evolves with time. For example, the first generations of set top box decoders were largely restricted to processing images in bit map form. However, newer generations of
35 decoder may use decompression software to download JPEG, GIF or other image formats and future generations will be able to process not only images but also movie clips in multiple formats. Consequently, it is desirable to structure program guide data to support decoder scalability i.e. to allow a range of decoders of
40 varying complexity to process program guide information using

16

5 the level of processing power that they are endowed with. Thereby, low-complexity decoders are able to identify objects that they are able to process and to discard objects exceeding their processing capabilities. Otherwise high-complexity objects may impair the operation of low-complexity decoders by causing
10 buffer overflow or other problems.

The MDBT data structure of Figure 9 (and Figure 10) advantageously supports efficient multimedia object complexity discrimination in a decoder unit. For this purpose, the MDBT assigns PID (packet identifier) values to cells in the database. In
15 Figure 9, item 172 identifies a particular cell and is associated with a PID value by item 174. Further, item 172 defines the 24-bit field cell_type that defines the time, area, and complexity coordinates of a cell (see Figure 10). As such, an object complexity level indicator is included in an MPEG-2 compatible transport-
20 layer field. Thereby objects in the data base belonging to different complexity levels are conveyed in streams identified by different PIDs. A decoder is pre-assigned a complexity level and the decoder applies the MDBT (specifically item 172) in selecting and caching the PID values for those cells with complexity levels that
25 match or are below the pre-assigned decoder complexity level. Objects of complexity level exceeding the decoders capabilities are advantageously discarded at the transport level.

Figure 12 shows a flowchart of a method for forming program specific information, according to the invention. The
30 method of Figure 12 generates program specific information including MGT, MDBT, CCT, STT, RRT, TCIT, CCIT, SCIT and SIT data and descriptors containing the advantageous features previously described. The method may be employed at an encoder for broadcasting program guide data or, may be employed for
35 encoding program guide data within a decoder unit for transmission to another device.

Following the start at step 250 of Figure 12, in step 253, a method based on the previously described data structuring principles is selected for partitioning program specific information.
40 The program specific information is partitioned in accordance with

17

5 time segments and areas, network types, complexity levels, cells, and programs (events). In step 255, PID values are assigned to accommodate the partitioned program specific information. If a non-MPEG transport protocol is used, PID values may be replaced by proper parameters that identify logical channels. In step 257 a
10 MGT and MDBT (or other type of control tables) are generated to include those formed during partition operations. The MGT conveys information for use in acquiring program specific information conveyed in other tables. The MDBT conveys information for use in acquiring multimedia objects from a
15 transport stream.

In step 260 individual CCT, STT, RRT, TCIT, CCIT, SCIT and SIT etc. tables are formed complying with the partitioned structure. The individual tables incorporate multimedia object links, version numbers and identifiers derived according to the
20 previously described invention principles. A CIT (e.g. TCIT, CCIT and SCIT) is formed containing channel and program identification information enabling acquisition of available broadcast programs and channels containing packet identifiers for identifying individual packetized datastreams that constitute individual
25 programs to be transmitted on particular channels. Further, in step 260, an SIT is generated containing program guide schedule information including descriptive lists of programs (events) receivable on the channels listed in the CIT.

In step 263, the tables formed in step 260, together
30 with associated multimedia objects, are formatted to be compatible with a desired data format and protocol. Such data formats and protocols include, for example, MPEG2 compatible Program Specific Information, MPEG2 DSM-CC, DSS, and an Internet compatible file transfer format. In step 265, the resulting
35 formatted tables and multimedia objects are incorporated into a datastream in their designated locations for terrestrial transmission. The MGT and MDBT are incorporated into the datastream in step 267.

In step 270, the program specific information
40 produced in step 267, together with video and audio program

18

5 representative components (and other data) for multiple channels, is multiplexed and formatted into a transport stream for output. In step 270, the output transport stream is further processed to be suitable for terrestrial transmission to another device such as a receiver, video server, or storage device for recording on a storage
10 medium, for example. The processes performed in step 270 include known encoding functions such as data compression Reed-Solomon encoding, interleaving, scrambling, trellis encoding, and carrier modulation. The process is complete and terminates at step 275. In the process of Figure 12, multiple CIT, SIT and associated
15 extension tables may be formed and incorporated in the program specific information in order to accommodate expanded numbers of channels. Further, in other embodiments the tables may be similarly processed for satellite, cable or Internet transmission, for example.

20 In the video receiver system of Figure 13, a broadcast carrier modulated with signals carrying audio, video and associated data representing broadcast program content is received by antenna 10 and processed by unit 13. The resultant digital output signal is demodulated by demodulator 15. The
25 demodulated output from unit 15 is trellis decoded, mapped into byte length data segments, deinterleaved and Reed-Solomon error corrected by decoder 17. The corrected output data from unit 17 is in the form of an MPEG compatible transport datastream containing program representative multiplexed audio, video and
30 data components. The transport stream from unit 17 is demultiplexed into audio, video and data components by unit 22 which are further processed by the other elements of decoder system 100. In one mode, decoder 100 provides MPEG decoded data for display and audio reproduction on units 50 and 55
35 respectively. In another mode, the transport stream from unit 17 is processed by decoder 100 to provide an MPEG compatible datastream for storage on storage medium 105 via storage device 90.

A user selects for viewing either a TV channel (user
40 selected channel-SC) or an on-screen menu, such as a program

5 guide, by using a remote control unit 70. Controller 60 uses the selection information provided from remote control unit 70 via interface 65 to appropriately configure the elements of Figure 13 to receive a desired program channel for viewing. Controller 60 comprises processor 62 and processor 64. Unit 62 processes (i.e.
10 parses, collates and assembles) system timing information and program specific information including program guide information. Processor 64 performs the remaining control functions required in operating decoder 100. Although the functions of unit 60 may be implemented as separate elements 62
15 and 64 as depicted in Figure 13, they may alternatively be implemented within a single processor. For example, the functions of units 62 and 64 may be incorporated within the programmed instructions of a microprocessor.

Controller 60 configures processor 13, demodulator 15,
20 decoder 17 and decoder system 100 to demodulate and decode the input signal format and coding type. Further, controller 60 configures units 13, 15, and 17 for other communication modes, such as for receiving cable television (CATV) signals and for bi-directional communication via coaxial line 14 or for bi-directional
25 (e.g. Internet) communication, for example, via telephone line 11. In an analog video mode, an NTSC compatible signal is received by units 13, 15 and 17 and processed by decoder 100 for video display and audio reproduction on units 50 and 55 respectively. Units 13, 15, 17 and sub-units within decoder 100 are
30 individually configured for the input signal type by controller 60 setting control register values within these elements using a bi-directional data and control signal bus C.

The transport stream provided to decoder 100 comprises data packets containing program channel data and
35 ancillary system timing information and program specific information including program guide information. Unit 22 directs the ancillary information packets to controller 60 which parses, collates and assembles this information into the previously described hierarchically arranged tables (as exemplified in Figure
40 1). Individual data packets comprising the User selected program

20

5 channel SC are identified and assembled using the assembled program specific information. Further, the program specific information contains conditional access, network information and identification and linking data enabling the system of Figure 13 to
10 complete programs. The program specific information also contains data supporting the identification and assembly of the ancillary information.

The program specific and system timing information is assembled by controller 60 into multiple hierarchically arranged
15 and inter-linked tables per the structure of Figure 1. The STT contains a time reference indicator and associated correction data sufficient for a decoder to establish a time of transmission of a program by a broadcast source. The MGT contains information for acquiring program specific information conveyed in other tables
20 such as identifiers for identifying data packets associated with the other tables. The CIT (e.g. TCIT) contains information for tuning and navigation to receive a User selected program channel. The SIT contains descriptive lists of programs (events) receivable on the channels listed in the CIT. The RRT contains program content
25 rating information such as the MPAA (Motion Picture Association of America) or V-chip compatible rating information that is collated by region (e.g. by country or by state within the USA). Additional program specific information describing and supplementing items within the hierarchical tables is conveyed
30 within descriptor information elements. The program specific and system timing information acquired by controller 60 via unit 22 is stored within internal memory of unit 60. Controller 60 uses the acquired program guide information in conditioning access to programs and in scheduling program processing functions
35 including program viewing, recording and playback.

Controller 60 and processor 22 determine from the CIT the PIDs of video, audio and sub-picture streams in the packetized decoded transport stream input to decoder 100 from unit 17. The video, audio and sub-picture streams constitute the desired
40 program being transmitted on selected channel SC. Processor 22

21

5 provides MPEG compatible video, audio and sub-picture streams to video decoder 25, audio decoder 35 and sub-picture processor 30 respectively. The video and audio streams contain compressed video and audio data representing the selected channel SC program content. The sub-picture data contains the SIT, CCT and
10 RRT information associated with the channel SC program content.

Decoder 25 decodes and decompresses the MPEG compatible packetized video data from unit 22 and provides decompressed program representative pixel data to NTSC encoder 45 via multiplexer 40. Similarly, audio processor 35 decodes the
15 packetized audio data from unit 22 and provides decoded and amplified audio data, synchronized with the associated decompressed video data, to device 55 for audio reproduction. Processor 30 decodes and decompresses sub-picture data received from unit 22.

20 Processor 30 assembles, collates and interprets RRT, CCT, CIT and data objects from unit 22 to produce formatted program guide data for output to OSD 37. OSD 37 processes the SIT, RRT, and CCT and other information to generate pixel mapped data representing subtitling, control and information menu
25 displays including selectable menu options and other items for presentation on the display device 50. The control and information menus that are displayed enable a user to select a program to view and to schedule future program processing functions including a) tuning to receive a selected program for viewing, b)
30 recording of a program onto storage medium 105, and c) playback of a program from medium 105.

The control and information displays, including text and graphics produced by OSD generator 37, are generated in the form of overlay pixel map data under direction of controller 60.
35 The overlay pixel map data from unit 37 is combined and synchronized with the decompressed pixel representative data from MPEG decoder 25 in encoder 45 via multiplexer 40 under direction of controller 60. Combined pixel map data representing a video program on channel SC together with associated sub-picture

22

5 data is encoded by NTSC encoder 45 and output to device 50 for display.

In a storage mode of the system of Figure 13, the corrected output data from unit 17 is processed by decoder 100 to provide an MPEG compatible datastream for storage. In this mode, 10 a program is selected for storage by a user via remote unit 70 and interface 65. Processor 22, in conjunction with processor 60 forms condensed program specific information including MGT, MDBT, CCT, STT, RRT, TCIT and SIT data and descriptors containing the advantageous features previously described. The condensed 15 program specific information supports decoding of the program selected for storage but excludes unrelated information. Processor 60, in conjunction with processor 22 forms a composite MPEG compatible datastream containing packetized content data of the selected program and associated condensed program specific 20 information. The composite datastream is output to storage interface 95.

Storage interface 95 buffers the composite datastream to reduce gaps and bit rate variation in the data. The resultant buffered data is processed by storage device 90 to be suitable for 25 storage on medium 105. Storage device 90 encodes the buffered datastream from interface 95 using known error encoding techniques such as channel coding, interleaving and Reed Solomon encoding to produce an encoded datastream suitable for storage. Unit 90 stores the resultant encoded datastream incorporating the 30 condensed program specific information on medium 105.

The architecture of Figure 13 is not exclusive. Other architectures may be derived in accordance with the principles of the invention to accomplish the same objectives. Further, the functions of the elements of decoder 100 of Figure 13 and the 35 process steps of Figure 12 may be implemented in whole or in part within the programmed instructions of a microprocessor. In addition, the principles of the invention apply to any form of MPEG or non-MPEG compatible electronic program guide. A datastream formed according to the invention principles may be 40 used in a variety of applications including video server or PC type

WO 99/43159

PCT/US99/03512

23

5 communication via telephone lines, for example. A program
datastream with one or more components of video, audio and data
formed to incorporate program specific information according to
invention principles may be recorded on a storage medium and
transmitted or re-broadcast to other servers, PCs or receivers. The
10 key elements of the data structure described herein may be
advantageously used for conveying program ancillary information
in a wide variety of data transport structures that may be used to
deliver program content or program guide information. Such
transport structures, for example, may include MPEG-PSI, Internet
15 TCP/IP (Transport Control Protocol/Internet Protocol), DSS (Digital
Satellite System), ATM (Asynchronous Transfer Mode) etc.

What is claimed is:

1. Apparatus for acquiring packetized program data from at least a first source, comprising:

a decoder for providing the packetized program data; and a controller

(1) for acquiring program guide information and for acquiring ancillary information conveyed in hierarchically ordered data tables in said packetized program data, said ancillary information including,

(a) a first version identifier conveyed in a primary data table and updated in response to a version change in at least one of a plurality of secondary tables hierarchically linked to said primary data table, and

(b) a second version identifier conveyed in a secondary data table and updated in response to at least one of,

a version change in said secondary table, and

a version change in a tertiary table hierarchically linked to said secondary table;

(2) for determining change in said secondary data table content by examining said second version identifier for a change following determination of a change in said first version identifier, and

(3) for acquiring said secondary data table in response to said determination of change.

2. Apparatus according to claim 1, wherein said primary data table comprises a root database table for indicating version change in hierarchically ordered program guide data tables.

3. Apparatus according to claim 1, wherein said secondary data table is used to indicate change in multimedia objects comprising objects associated with at least one of (a) broadcast channels, (b) broadcast programs, and (c) User interface controls.

4. Apparatus according to claim 1, wherein said primary data table is used to indicate change in at least one of (a) electronic program guide information tables and (b) MPEG compatible program specific information.

5. Apparatus according to claim 1, wherein said ancillary information is a two level hierarchical arrangement containing only a primary table and secondary tables.

6. Apparatus for adaptively decoding re-partitionable packetized program guide data, comprising:

a decoder for providing a transport stream include program data guide data; and

a controller (a) for acquiring the program guide data, the program guide data comprising hierarchically ordered data table partitions and including partitioning information, said partitioning information including,

partition identifiers assigned to individual partitions of said program guide data, wherein said program guide data partitions are dynamically re-partitionable by re-assignment of said partition identifiers in said partitioning information; and

(b) for identifying said re-assigned partition identifiers and for acquiring additional program guide data in response to said identified re-assigned partition identifiers.

7. Apparatus according to claim 6, wherein said partition identifiers identify program guide data partitions based on at least one of, (a) an area, (b) a broadcast time, (c) a complexity level, and (d) a partition type.

8. A method for forming packetized program data to be suitable for processing in a decoder, comprising the steps of:

forming program guide information and ancillary information into hierarchically ordered data tables and including in said ancillary information,

(a) a first version identifier conveyed in a primary data table and updated in response to a version change in at least one of a plurality of secondary tables hierarchically linked to said primary data table, and

(b) a second version identifier conveyed in a secondary data table and updated in response to at least one of,

a version change in said secondary table, and

a version change in a tertiary table hierarchically linked to said secondary table; and

incorporating said ancillary information and said program guide information into packetized data for output to a transmission channel.

9. A method according to claim 8, including the step of forming said primary data table to comprise a root database table for indicating version change in hierarchically ordered program guide data tables.

10. A method according to claim 8, wherein forming said secondary data table to indicate change in multimedia objects comprising objects associated with at least one of (a) broadcast channels, (b) broadcast programs, and (c) User interface controls.

11. A method according to claim 8, wherein forming said primary data table to indicate change in at least one of (a) electronic program guide information tables and (b) MPEG compatible program specific information.

12. A method according to claim 8, wherein said ancillary information is a two level hierarchical arrangement containing only a primary table and secondary tables.

13. A method for forming packetized program data to be suitable for processing in a decoder, comprising the steps of:

partitioning program guide information and ancillary information into hierarchically ordered data table partitions and including a database in said ancillary information, said database including,

(a) updateable version numbers for indicating content change of a partition, and

(b) cell numbers assigned to individual partitions of said program guide information, wherein said program guide information cell partitions are dynamically re-partitionable by re-assignment of said cell number in said database; and

incorporating said ancillary information and said program guide information into packetized data for output to a transmission channel.

14. A method according to claim 13, wherein said ancillary information contains a multimedia object comprising objects associated with at least one of (a) broadcast channels, (b) broadcast programs, and (c) User interface controls.

15. A method according to claim 14, wherein an object comprises at least one of (a) a video segment, (b) an audio segment, (c) text, (d) an icon representing a user selectable item for display, (e) an HTML or SGML document (f) a menu of selectable items, (g) an image window for presentation within an encompassing image, and (h) an image window for initiating a multimedia function.

16. A method according to claim 13, wherein a cell number incorporates at least one of, (a) an area identifier, (b) a broadcast time identifier, and (c) a complexity level identifier.

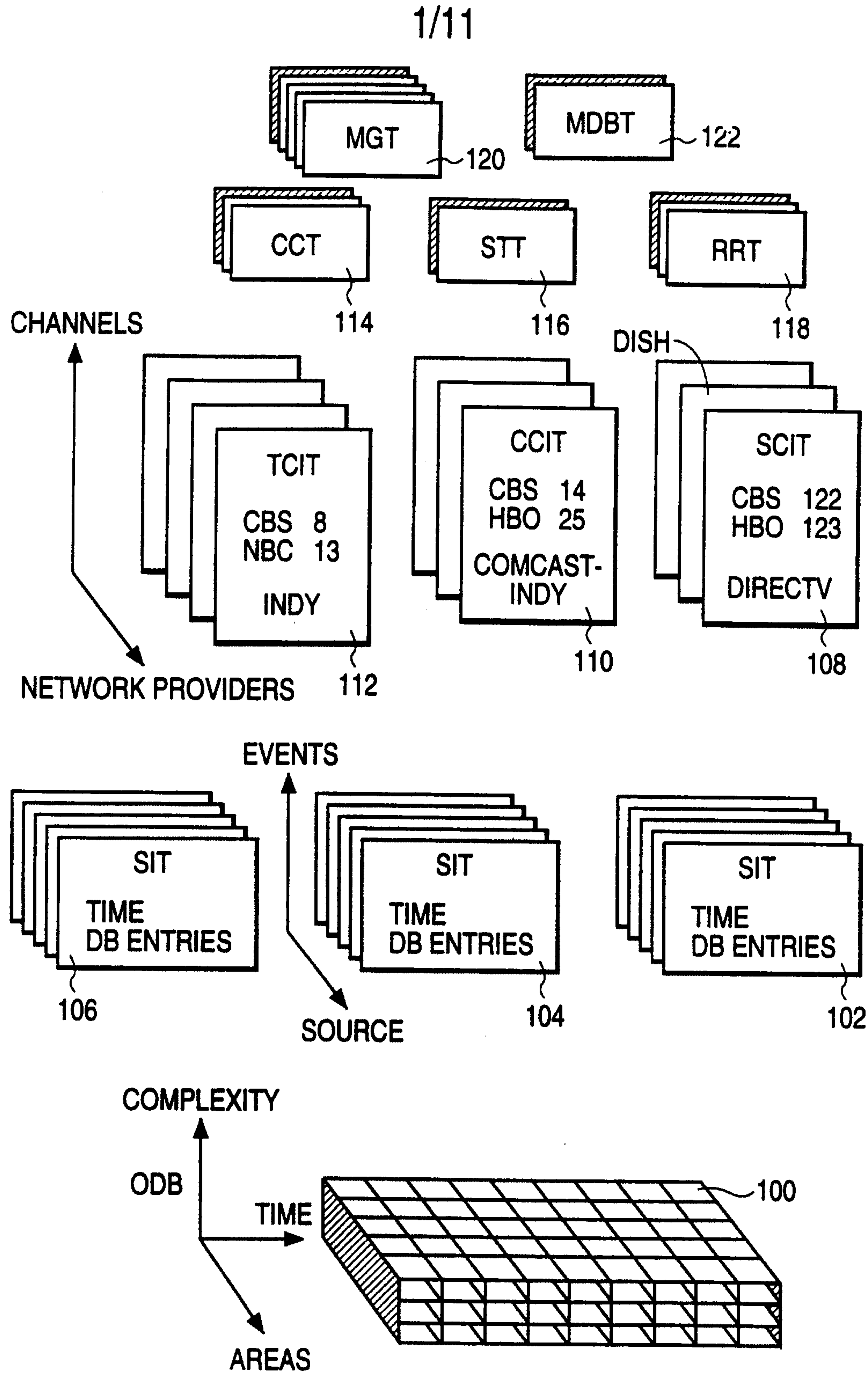


FIG. 1

2/11

| SYNTAX | BITS | FORMAT |
|-------------------------------------|------|--------|
| master_guide_table_section () { | | |
| table_id | 8 | 0x91 |
| section_syntax_indicator | 1 | '1' |
| private_indicator | 1 | '1' |
| reserved | 2 | '11' |
| section_length | 12 | uimsbf |
| network_provider | 16 | uimsbf |
| reserved | 2 | '11' |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | '1' |
| section_number | 8 | 0x00 |
| last_section_number | 8 | 0x00 |
| protocol_version | 8 | uimsbf |
| number_networks_defined | 16 | uimsbf |
| table_types_defined | 16 | uimsbf |
| for (i=0;<table_types_defined;i++){ | | |
| table_type | 16 | uimsbf |
| reserved | 3 | '111' |
| table_type_PID | 13 | uimsbf |
| reserved | 3 | '111' |
| table_type_version_number | 5 | uimsbf |
| number_bytes | 32 | uimsbf |
| reserved | 4 | '1111' |
| table_type_descriptors_length | 12 | uimsbf |
| for (k=0;k<N;k++){ | | |
| descriptor() | var | |
| } | | |
| reserved | 4 | '1111' |
| descriptors_length | 12 | uimsbf |
| for (l = 0;l<N;l++){ | | |
| descriptor() | var | |
| CRC_32 | 32 | rpchof |
| } | | |

FIG. 2

3/11

| SYNTAX | BITS | FORMAT |
|--|------|-------------|
| terrestrial_channel_information_table_section () { | | |
| table_id | 8 | 0x95 |
| section_syntax_indicator | 1 | '1' |
| private_indicator | 1 | '1' |
| reserved | 2 | '11' |
| section_length | 12 | uimsbf |
| network_provider | 16 | uimsbf |
| reserved | 2 | '11' |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |
| protocol_version | 8 | uimsbf |
| num_channels_in_section | 8 | uimsbf |
| for (i=0;i<num_channels_in_section;i++){ | | |
| short name | 7*16 | unicode BMP |
| reserved | 4 | '1111' |
| major_channel_number | 10 | uimsbf |
| minor_channel_number | 10 | uimsbf |
| modulation_mode | 8 | uimsbf |
| carrier_frequency | 32 | uimsbf |
| channel_TSID | 16 | uimsbf |
| program_number | 16 | uimsbf |
| access_controlled | 1 | bslbf |
| hidden | 1 | bslbf |
| reserved | 8 | 0xFF |
| service_type | 6 | uimsbf |
| source_id | 16 | uimsbf |
| carousel_id | 32 | uimsbf |
| reserved | 6 | '111111' |
| descriptors_length | 10 | uimsbf |
| for (i=0;i<N;i++) { | | |
| descriptors() | | |
| } | | |
| } | | |
| reserved | 6 | '111111' |
| additional_descriptors_length | 10 | uimsbf |
| for(j=0;j<N;j++) { | | |
| additional_descriptors() | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

FIG. 3

4/11

| SYNTAX | BITS | FORMAT |
|---|------|--------|
| channel_basic_information_file_section () { | | |
| table_id | 8 | 0x3C |
| section_syntax_indicator | 1 | '1' |
| private_indicator | 1 | '1' |
| reserved | 2 | '11' |
| section_length | 12 | uimsbf |
| moduleId | 16 | uimsbf |
| reserved | 2 | '11' |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | '1' |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |
| protocolDiscriminator | 8 | 0x11 |
| dsmccType | 8 | 0x03 |
| messageId | 16 | 0x1003 |
| carouselId | 32 | uimsbf |
| reserved | 8 | 0xFF |
| adaptionLength | 8 | 0x00 |
| messageLength | 16 | uimsbf |
| moduleId | 16 | uimsbf |
| moduleVersion | 8 | uimsbf |
| reserved | 8 | 0xFF |
| blockNumber | 16 | uimsbf |
| Object_data () | | |
| CRC_32 | 32 | rpchof |
| } | | |

FIG. 4

5/11

| SYNTAX | BITS | FORMAT |
|---|------|--------|
| channel_basic_information_file_section () { | | |
| table_id | 8 | 0x3C |
| section_syntax_indicator | 1 | '1' |
| private_indicator | 1 | '1' |
| reserved | 2 | '11' |
| section_length | 12 | uimsbf |
| moduleId | 16 | 0x0001 |
| reserved | 2 | '11' |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | '1' |
| section_number | 8 | 0x00 |
| last_section_number | 8 | 0x00 |
| protocolDiscriminator | 8 | 0x11 |
| dsmccType | 8 | 0x03 |
| messageId | 16 | 0x1003 |
| carouselId | 32 | uimsbf |
| reserved | 8 | 0xFF |
| adaptionLength | 8 | 0x00 |
| messageLength | 16 | uimsbf |
| moduleId | 16 | 0x0001 |
| moduleVersion | 8 | uimsbf |
| reserved | 8 | 0xFF |
| blockNumber | 16 | 0x0000 |
| number_modules | 16 | uimsbf |
| for (i=0;i<number_modules;i++){ | | |
| moduleId | 16 | uimsbf |
| number_blocks | 16 | uimsbf |
| reserved | 4 | '1111' |
| moduleSize | 28 | uimsbf |
| moduleVersion | 8 | uimsbf |
| } | | |
| reserved | 4 | '1111' |
| descriptors_length | 12 | |
| for (i=0;i<N;i++) { | | |
| descriptor() | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

FIG. 5

6/11

| SYNTAX | BITS | FORMAT |
|---|------|--------|
| event_basic_information_file_section () { | | |
| table_id | 8 | 0x3C |
| section_syntax_indicator | 1 | '1' |
| private_indicator | 1 | '1' |
| reserved | 2 | '11' |
| section_length | 12 | uimsbf |
| moduleId | 16 | 0x0002 |
| reserved | 2 | '11' |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | '1' |
| section_number | 8 | 0x00 |
| last_section_number | 8 | 0x00 |
| protocolDiscriminator | 8 | 0x11 |
| dsmccType | 8 | 0x03 |
| messageId | 16 | 0x1003 |
| carouselId | 32 | uimsbf |
| reserved | 8 | 0xFF |
| adaptionLength | 8 | 0x00 |
| messageLength | 16 | uimsbf |
| moduleId | 16 | 0x0002 |
| moduleVersion | 8 | uimsbf |
| reserved | 8 | 0xFF |
| blockNumber | 16 | 0x0000 |
| title_length | 8 | uimsbf |
| title_text() | var | |
| number_modules | 16 | uimsbf |
| for (i=0;i<number_modules;i++) { | | |
| moduleId | 16 | uimsbf |
| number_blocks | 16 | uimsbf |
| reserved | 4 | '1111' |
| moduleSize | 28 | uimsbf |
| moduleVersion | 8 | uimsbf |
| } | | |
| reserved | 4 | '1111' |
| descriptors_length | 12 | |
| for (i=0;i<N;i++) { | | |
| descriptor() | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

FIG. 6

| | SYNTAX | BITS | FORMAT |
|-----|---|------|--------|
| | control_basic_infcrmation_file_section () { | | |
| | table_id | 8 | 0x3C |
| | section_syntax_indicator | 1 | '1' |
| | private_indicator | 1 | '1' |
| | reserved | 2 | '11' |
| | section_length | 12 | uimsbf |
| | moduleId | 16 | 0x0003 |
| | reserved | 2 | '11' |
| | version_number | 5 | uimsbf |
| | current_next_indicator | 1 | '1' |
| | section_number | 8 | 0x0000 |
| | last_section_number | 8 | 0x0000 |
| | protocolDiscriminator | 8 | 0x11 |
| | dsmccType | 8 | 0x03 |
| | messageId | 16 | 0x1003 |
| | carouselId | 32 | uimsbf |
| 159 | reserved | 8 | 0xFF |
| | adaptionLength | 8 | 0x00 |
| | messageLength | 16 | uimsbf |
| | moduleId | 16 | 0x0003 |
| | moduleVersion | 8 | uimsbf |
| | reserved | 8 | 0xFF |
| | blockNumber | 16 | 0x0000 |
| | number_modules | 16 | uimsbf |
| | for (i=0;i<number_modules;i++) { | | |
| | moduleId | 16 | uimsbf |
| | number_blocks | 16 | uimsbf |
| | reserved | 4 | '1111' |
| | moduleSize | 28 | uimsbf |
| | moduleVersion | 8 | uimsbf |
| 167 | } | | |
| | reserved | 4 | '1111' |
| | descriptors_length | 12 | |
| | for (i=0;i<N;i++) { | | |
| | descriptor() | | |
| | } | | |
| | CRC_32 | 32 | rpchof |
| | } | | |

FIG. 7

| | MSB | | | | LSB | |
|----------------|------|----|------|----|-----------|---|
| Bit | 31 | 24 | 23 | 16 | 15 | 0 |
| 152 carouselId | time | | area | | dirNumber | |

FIG. 8

| SYNTAX | | BITS | FORMAT |
|---|-----|------|--------|
| master_database_table_section () { | | | |
| table_id | | 8 | 0x92 |
| section_syntax_indicator | | 1 | '1' |
| private_indicator | | 1 | '1' |
| reserved | | 2 | '11' |
| section_length | | 12 | uimsbf |
| table_id_extension | | 16 | 0x0000 |
| reserved | | 2 | '11' |
| version_number | | 5 | uimsbf |
| current_next_indicator | | 1 | '1' |
| section_number | 170 | 8 | uimsbf |
| last_section_number | | 8 | uimsbf |
| protocol_version | | 8 | uimsbf |
| number_cells_defined | | 24 | uimsbf |
| number_cells_in_section | 172 | 24 | uimsbf |
| for (i=0;i<number_cells_in_section;i++) { | | | |
| cell_type | | 24 | uimsbf |
| reserved | | 3 | '111' |
| cell_type_PID | | 13 | uimsbf |
| reserved | | 3 | '111' |
| cell_type_version_number | 174 | 5 | uimsbf |
| number_bytes | | 32 | uimsbf |
| reserved | | 4 | '1111' |
| cell_type_descriptors_length | 176 | 12 | uimsbf |
| for (k=0;k<N;k++) | | | |
| descriptor() | | var | |
| } | | | |
| CRC_32 | 178 | 32 | rpchof |
| } | | | |

FIG. 9

| | | | | | | |
|-----------|------|----|------|---|------------|---|
| | MSB | | | | LSB | |
| Bit | 23 | 16 | 15 | 8 | 7 | 0 |
| cell_type | time | | area | | complexity | |

179

FIG. 10

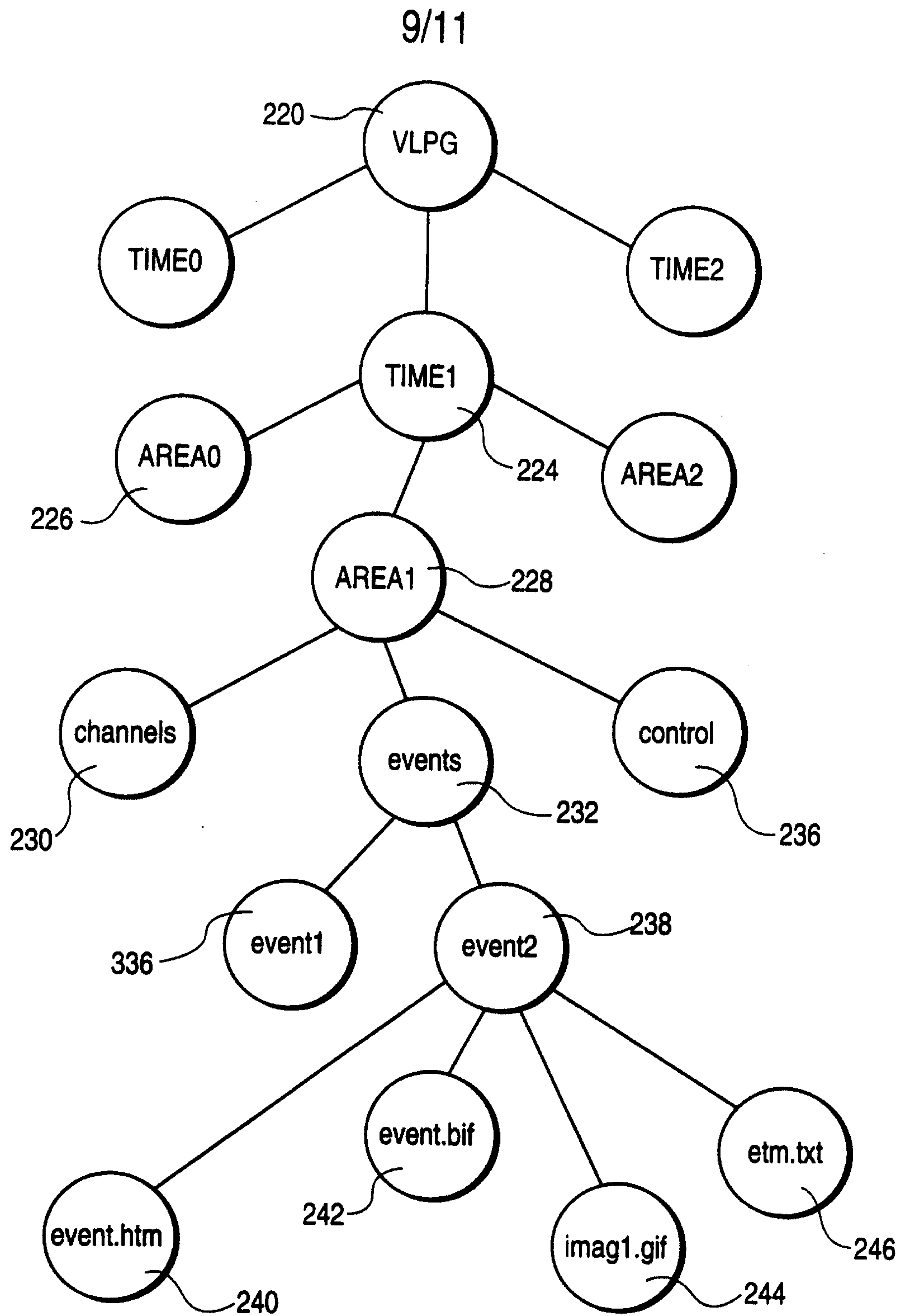


FIG. 11

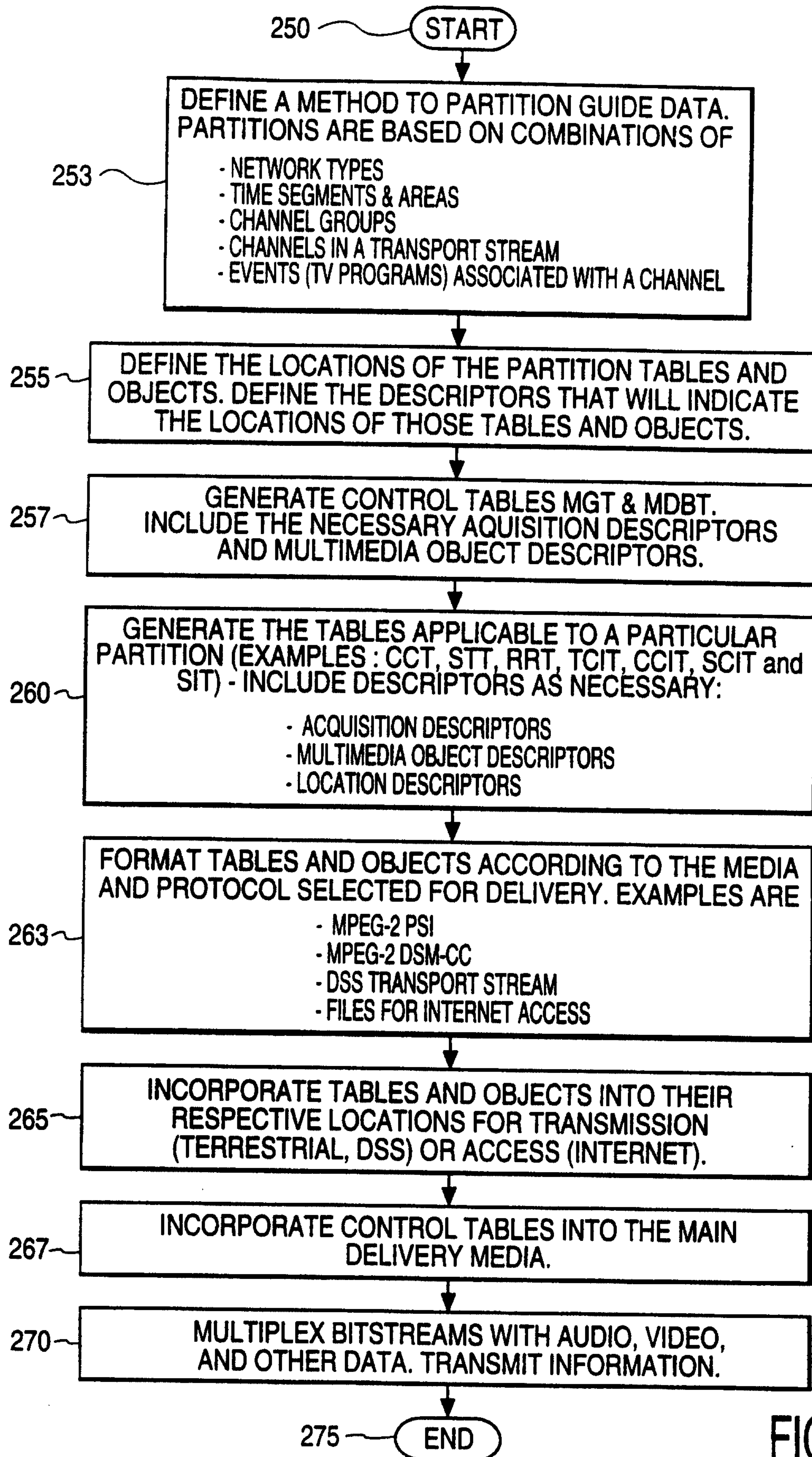


FIG. 12

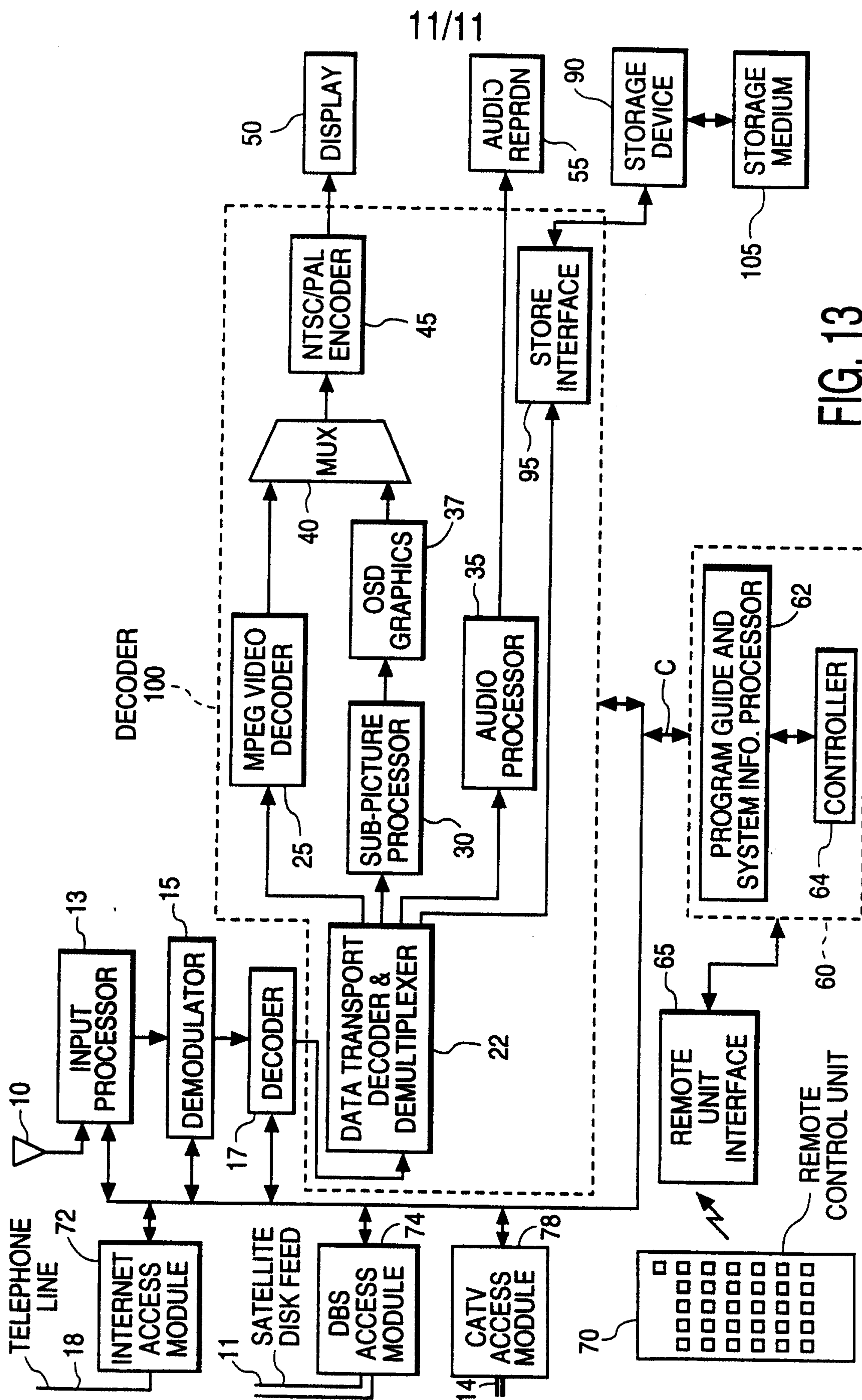


FIG. 13

