

(19)



(11)

EP 1 803 299 B9

(12)

CORRECTED EUROPEAN PATENT SPECIFICATION

(15) Correction information:

Corrected version no 1 (W1 B1)
Corrections, see
Description Paragraph(s) 60
Claims EN 16

(51) Int Cl.:

H04N 7/24 (2011.01)

(86) International application number:

PCT/GB2005/002582

(48) Corrigendum issued on:

18.12.2013 Bulletin 2013/51

(87) International publication number:

WO 2006/043017 (27.04.2006 Gazette 2006/17)

(45) Date of publication and mention of the grant of the patent:

19.06.2013 Bulletin 2013/25

(21) Application number: **05756958.4**

(22) Date of filing: **28.06.2005**

(54) **SYSTEM FOR HANDLING PROGRAM META-DATA**

SYSTEM ZUR HANDHABUNG VON PROGRAMM-METADATEN

SYSTEME DE GESTION DE METADONNEES DE PROGRAMME

(84) Designated Contracting States:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU MC NL PL PT RO SE SI SK TR

- **PARSONS, Ian**
Bournemouth
Dorset BH6 3NP (GB)

(30) Priority: **20.10.2004 GB 0423323**
21.10.2004 US 620821 P

(74) Representative: **White, Duncan Rohan**
Marks & Clerk LLP
90 Long Acre
London
WC2E 9RA (GB)

(43) Date of publication of application:
04.07.2007 Bulletin 2007/27

(60) Divisional application:
12150380.9 / 2 439 950

(56) References cited:
US-A- 5 844 620 US-A1- 2001 027 562
US-A1- 2003 046 699 US-A1- 2003 090 524
US-A1- 2003 145 085 US-A1- 2003 208 761
US-A1- 2004 040 036

(73) Proprietor: **NDS Limited**
Staines, Middlesex TW18 4EX (GB)

(72) Inventors:

- **BASTABLE, Ian**
Fareham
Hampshire PO15 7LT (GB)

EP 1 803 299 B9

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

[0001] The present invention generally relates to television systems.

5 BACKGROUND OF THE INVENTION

[0002] There are known in the art schemes, such as schemes of Digital Video Broadcasting - System Information (DVB-SI), schemes of TV-Anytime, and schemes of Advanced Television Systems Committee (ATSC) A65/B Program and System Information Protocol (PSIP), that convey program meta-data organized by time.

10 **[0003]** DVB-SI schemes are described in the European Telecommunications Standards Institute (ETSI) document EN 300 468, which is available at ETSI website www.etsi.org.

[0004] TV-Anytime schemes are described in the ETSI document TS 102 822-3-1, which is available at the TV-Anytime Forum website www.tv- anytime.org.

[0005] ATSC A/65B PSIP schemes are described at the ATSC website www.atsc.org/standards.html.

15 **[0006]** DVB-SI and TV-Anytime schemes, for example, also group program meta-data for programs that appear more than once into a single virtual container. However, in such schemes there is no access to only portions of a single virtual container and thus the single virtual container cannot be addressed by time window alone.

20 **[0007]** US-A-2003/0208761 discloses a method implemented at a client in a broadcast architecture in which a carousel file server broadcasts carousel data to one or more clients, comprising receiving an index structure from the carousel file server, the index data structure having index values derived from content in corresponding segments of the broadcast carousel data, searching the index data structure to identify one of more particular segments of the broadcast carousel data for further searching, and retrieving the one or more particular segments from the carousel file server.

[0008] US-A-5844620 discloses an apparatus for providing an interactive program guide wherein schedule information for a current time period is transmitted in pages which are associated with time slots to enable receiver side filtering.

25

SUMMARY OF THE INVENTION

[0009] The present invention, in preferred embodiments thereof, seeks to provide a method of handling program meta-data and of enabling a set-top box (STB) to handle normalized program meta-data in a way that allows the STB to acquire only portions of the normalized program meta-data for rendering an Electronic Program Guide (EPG) which filters the program meta-data by time.

30 **[0010]** The term "meta-data" is used throughout the present specification and claims to include information descriptive of or otherwise referring to a content stream or a portion thereof. Meta-data may include, for example, pointers, tags, codes, flags, editorial information, and indexing information. At least part of the meta-data may be used to enable or disable interactive operations, such as EPG related operations, depending, for example, on values assigned to the tags or the flags. In a case where the content stream is associated with at least one television program, the pointers, tags, codes, flags, editorial information and indexing information may point to, describe, and/or index segments of the at least one television program as well as portions of the content stream.

35 **[0011]** It is appreciated that the contents of the meta-data may preferably pertain to either a plurality of users or an individual user. The meta-data may be used to enable operations on the at least one television program, if associated with the content stream, and/or on content in the content stream. The meta-data may be available from one or many broadcast sources at one or more times when the at least one television program is broadcasted or at times before the at least one television program is broadcasted.

40 **[0012]** The term "normalize", in all of its grammatical forms, is used throughout the present specification and claims to refer to grouping of instances of programs that share at least one editorial description item of the content.

45 **[0013]** The term "instance" is used throughout the present specification and claims to refer to an occurrence of a program or an event, broadcast or transmitted at a specific date and time. If the same program or event is broadcasted or transmitted more than once, each time the program or the event is broadcasted or transmitted constitutes an instance of the program or the event.

50 **[0014]** The invention provides an apparatus in accordance with claim 1 and a method in accordance with claim 16.

[0015] Advantageous embodiments are set out in the dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

55 **[0016]** The present invention will be understood and appreciated more fully from the following detailed description, taken in conjunction with the drawings in which:

Fig. 1 is a simplified partly pictorial, partly block diagram illustration of a preferred implementation of an EPG system

constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 2 is a simplified block diagram illustration of an implementation of apparatus for grouping program meta-data in the EPG system of Fig. 1, the apparatus being constructed and operative in accordance with the present invention;

5 Fig. 3 is a simplified block diagram illustration of an example of a data organization diagram produced by the apparatus of Fig. 2;

Fig. 4 is a bar graph illustration of an example of a distribution of bandwidth among a plurality of PID (Packet Identification) streams;

Fig. 5 is a simplified block diagram illustration of a preferred implementation of apparatus for accessing program meta-data.

10 Fig. 6 is a simplified flowchart illustration of a preferred method of operation of the apparatus of Fig. 2; and

Fig. 7 is a simplified flowchart illustration of a preferred method of operation of the apparatus of Fig. 5.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

15 **[0017]** In conventional EPG technology, data for drawing a single screen of an EPG grid is made available to the EPG at the time of drawing the EPG grid. Since typically an entire EPG dataset is not stored in an STB memory due, for example, to limitations of memory capacity at the STB, such EPG dataset is typically split up and only part of the EPG dataset is held in the STB memory at a time. The EPG dataset is typically split up by time, so that only part of the EPG dataset, which is required for a time period being displayed in the EPG grid, is actually loaded in the STB memory at
20 the time of display.

[0018] There are however systems, such as systems conforming to TV-Anytime specifications as described at the TV-Anytime Forum website www.tv-anytime.org, in which EPG grid data is normalized. In systems that conform to TV Anytime specifications for example, normalization is obtained by grouping by "program" and hence meta-data of all instances that share the same program details are grouped together.

25 **[0019]** If grouping is made by program, such grouping may conflict with EPG dataset splitting by time, since all instances of a program may be in different time groups. The present invention enables grouping of programs by time while maintaining a normalized EPG dataset.

[0020] According to the present invention normalization of the EPG dataset, or of parts of an EPG, includes grouping of instances of programs that share more than one editorial description item of the content, and preferably grouping of
30 instances of programs that share most of the editorial description items of the content or all of the editorial description items of the content.

[0021] Reference is now made to Fig. 1, which is a simplified partly pictorial, partly block diagram illustration of a preferred implementation of an EPG system 10 constructed and operative in accordance with a preferred embodiment of the present invention.

35 **[0022]** In the EPG system 10, a headend 20 preferably transmits to a user unit 30 or to a plurality of user units 30 EPG information as well as television programming for which the EPG information is intended. By way of example, which is not meant to be limiting, the headend 20 transmits the EPG information and the television programming via a satellite 40. Alternatively, the headend 20 may transmit the EPG information and the television programming to the user unit 30 or to the plurality of user units 30 via at least one of the following: a cable based communication network; a conventional
40 terrestrial broadcast television network; a telephony based communication network; a telephony based television broadcast network; a mobile-telephony based television broadcast network; an Internet Protocol (IP) television broadcast network; and a computer based communication network. An example of an appropriate telephony or IP based television broadcast network includes, for example, a Synamedia™ system, commercially available from NDS Limited, One London Road, Staines, Middlesex, TW 18 4EX, United Kingdom.

45 **[0023]** The television programming preferably includes at least one of the following: pay and/or non-pay television programming; multimedia information; audio programs; data; games; and information from computer based networks such as the Internet.

[0024] For simplicity of depiction and description, and without limiting the generality of the foregoing, only one user unit 30 is illustrated in Fig. 1 and referred to below. The user unit 30 preferably includes a set-top box (STB) 50, and a display 60 which is operatively associated with the STB 50 and is operative to display EPG information and television programming received and processed at the STB 50. The display 60 may comprise any appropriate display such as a television or a computer monitor. By way of example, which is not meant to be limiting, the STB 50 receives the EPG information and the television programming via an antenna 70.

55 **[0025]** In operation, the headend 20 normalizes program meta-data of instances of programs which share at least one editorial description item of content and preferably many editorial description items of the content, and groups the program meta-data to produce grouped-normalized program meta-data which constitutes EPG information for an EPG. The headend 20 then preferably broadcasts television programming and the grouped-normalized program meta-data to the user unit 30.

[0026] At the user unit 30, the television programming is received and processed at the STB 50 and displayed on the display 60 in accordance with selections of a user (not shown) of the user unit 30. The grouped-normalized program meta-data is also received and processed at the STB 50 and the user may preferably enter an EPG selection, which EPG selection may be entered by at least one of the following: a time window; a service; a program; and at least one editorial attribute in the program meta-data. The at least one editorial attribute preferably includes any appropriate editorial attribute such as, for example, a program category or sub-category.

[0027] Reference is now made to Fig. 2, which is a simplified block diagram illustration of a preferred implementation of apparatus 100 for grouping program meta-data in the EPG system 10 of Fig. 1, the apparatus 100 being constructed and operative in accordance with the present invention.

[0028] The apparatus 100 is preferably comprised in the headend 20 of Fig. 1. The apparatus 100 preferably includes a meta-data processor 120 and an encapsulation unit 130. The meta-data processor 120 may, for example, include a conventional microprocessor or a conventional micro-controller. The meta-data processor 120 preferably processes meta-data of programs offered in an EPG. The meta-data processor 120 preferably normalizes program meta-data of instances of programs which share at least one editorial description item of content to produce a set of program identifiers, and groups the instances of the programs by time windows to produce a set of time-window identifiers.

[0029] The encapsulation unit 130 preferably places the program meta-data in virtual containers, and associates the virtual containers with the set of program identifiers and the set of time-window identifiers. The term "virtual container" is used throughout the present specification and claims to refer to a block of information to be transported, which block of information includes communication payload, addressing information, and overhead information. The encapsulation unit 130 thus prepares the program meta-data for addressing by a user, as described below, through an addressing mechanism which allows the user to filter the program meta-data by at least one of the following: time; service; program; and at least one editorial attribute in the program meta-data.

[0030] Preferably, in order to normalize the program meta-data, the meta-data processor 120 scans instances of programs in the EPG and compares content description of the instances of the programs to determine equality of meta-data, such as equality of title, equality of synopsis, etc. The meta-data processor 120 then preferably groups, for each program, all instances of the program which share at least one editorial description item of the content to produce a program identifier which may be used to refer to the program. For example, a first instance of a single episode of a soap opera that is scheduled on a Monday morning and a second instance of the same episode of the soap opera that is scheduled on a Tuesday evening are grouped together as a program with a unique program identifier. Preferably, such grouping of all the instances of the programs in the EPG during an EPG period of, for example, a week results in a set of normalized programs and a set of program identifiers.

[0031] The meta-data processor 120 also preferably groups the instances of the programs by time windows to produce a set of time-window identifiers. For example, the meta-data processor 120 may collect all programs for each unique set of days the programs have instances in. Thus, a group of programs is obtained which, for example, has the following instances: all in day 0 only, all in day 1 only, all in day 0 and day 1 only, etc. An example of such a data organization diagram, which is produced by the apparatus 100, is depicted in Fig. 3.

[0032] Preferably, the meta-data processor 120 selects the time windows according to one of the following: a linear time base; and a non-linear time base.

[0033] The term "linear time base" is used throughout the present specification and claims to refer to a time base in a linear form in which time windows are calculated as a polynomial of a first degree of a time unit quantity. For example, for a time unit of 1day, the time window may be an integer number of days, such as 1day, 2 days, 3days, etc.

[0034] Preferably, the linear time base includes a time unit equivalent to at least one of the following: a part of a day; and a day.

[0035] The term "non-linear time base" is used throughout the present specification and claims to refer to a time base other than a linear time base. For example, in a non-linear time base a first time window may be 6 hours, a second time window may be 6 hours, a third time window may be 12 hours, a fourth time window may be 24 hours, a fifth time window may be 24 hours, etc.

[0036] Preferably, regardless of whether a linear time base or a non-linear time base is used for selecting time windows, each time window preferably includes at least one of the following: a part of a day; a day; more than one day; a week; and a specific weekday within a period of at least one month, such as a first Saturday of a month.

[0037] Preferably, the encapsulation unit 130 assigns container identifiers (container_ids) to the virtual containers in which the program meta-data is placed for identifying programs contained within a group. The container identifiers are preferably assigned using a bitfield assignment.

[0038] The encapsulation unit 130 is preferably operatively associated with a transmitter 140 in the headend 20. The transmitter 140 preferably transmits to the user unit 30 of Fig. 1, for example via an antenna 150, the virtual containers, the associated set of program identifiers, and the associated set of time-window identifiers. It is appreciated that the virtual containers together with the associated set of program identifiers and the associated set of time-window identifiers may be conveyed, for example, as MPEG-2 private sections in a broadcast network, or as Internet Protocol (IP) datagrams

in an IP network. In a case where the virtual containers are conveyed as MPEG-2 private sections, the MPEG-2 table_id_extension and other fields are set as necessary to convey a container identifier. The MPEG-2 fields that are set are preferably determined according to a number of bits used for signalling the container identifier, and the used bits preferably include the first 96 bits in the private data of the MPEG-2 private section syntax. It is appreciated that any other suitable bits may alternatively or additionally be used.

[0039] The virtual containers together with the associated set of program identifiers and the associated set of time-window identifiers may preferably be carried in one or many different PID (PID - Packet Identification) streams. It is appreciated that assignment of container identifiers to PID streams is preferably performed in a way that reduces bandwidth per PID stream by distributing the virtual containers among a plurality of PID streams and also preferably allows for bit assignment filtering throughout the plurality of PID streams where the bit may, for example, correspond to a time window such as a specific day. Such filtering may, for example in a case where the amount of virtual containers is small, be enabled by conveying in a distinct MPEG-2 PID stream all virtual containers with identifiers that have only a single bit set, or by conveying in a distinct MPEG-2 PID stream all virtual containers that have a specific bit set. All other virtual containers may then be conveyed in a single PID stream that can be filtered using a conventional hardware bit mask section filtering. In another example, each group of virtual containers per a single unit of time may be carried in a distinct PID stream and all virtual containers that cover multiple periods of time may be carried in a separate PID stream.

[0040] An example of a distribution of bandwidth among a plurality of PID streams with 147 bytes per program, a cycle time of 3.5 seconds, and a trickle time of 60 seconds is shown in Table 1 below and depicted in the form of a bar graph in Fig. 4.

TABLE 1

Container Identifier	Programs	Percent	Bytes	kbps*
1 (Trickle)	1468	10.53%	215796	29
Odd (Trickle)	2181	15.64%	320607	43
2	1367	9.80%	200949	459
4	1208	8.66%	177576	406
8	1155	8.28%	169785	388
16	1117	8.01%	164199	375
32	1157	8.30%	170079	389
64	1302	9.34%	191394	437
128	316	2.27%	46452	106
Even	2675	19.18%	393225	899
Total	13946	100.00%	2050062	3,531

* kbps - kilo-bits per second

[0041] It is appreciated that conventionally a bit being set refers to the bit having a value "1" and a bit being clear refers to the bit having a value "0". It is however to be understood that referring to a bit set as having the value "0" and to a bit clear as having the value "1" is alternatively applicable in the present invention.

[0042] It is further appreciated that other transport mechanisms may alternatively be used. For example, broadcast or multi-cast IP may be used where filtering may be performed at one of the following layers of the International Standards Organization (ISO) seven-layer model: a Data Link Layer; a Network Layer; and a Transport Layer.

[0043] It is still further appreciated that PID assignment allows for a bandwidth per PID stream to be reduced while still allowing for low access times to data carried in the PID stream. Since some types of conventional STBs can only cope with a limited bandwidth in one PID stream, PID assignment in accordance with the present invention distributes the virtual containers over a few PID streams thus reducing bandwidth per PID stream to levels that the STBs can cope with. It is appreciated that access times to the data carried in the PID streams preferably remains acceptable.

[0044] Preferably, access to the virtual containers, the associated set of program identifiers, and the associated set of time-window identifiers is enabled to allow a user to filter the program meta-data for obtaining a subset of the program meta-data. The subset of the program meta-data preferably includes a program meta-data subset which corresponds to at least one of the following: a time window; a service; a program; and at least one editorial attribute in the program meta-data. Thus, in a case where the user filters the program meta-data for obtaining a subset of the program meta-data which corresponds to a specific time window, the user is preferably enabled to view a list of programs to be broadcast in the specific time window. In a case where the user filters the program meta-data for obtaining a subset of the program meta-data which corresponds to a specific service, the user is preferably enabled to view a list of programs offered in the specific service. In a case where the user filters the program meta-data for obtaining a subset of the program meta-

data which corresponds to a specific program, the user is preferably enabled to view various time windows in which the specific program is to be broadcast. In a case where the user filters the program meta-data for obtaining a subset of the program meta-data which corresponds to at least one editorial attribute in the program meta-data, the user is preferably enabled to view a list of programs that share the at least one editorial attribute.

5 [0045] Reference is now made to Fig. 5, which is a simplified block diagram illustration of a preferred implementation of apparatus 200 for accessing program meta-data.

[0046] The apparatus 200 is preferably comprised in or associated with the STB 50 of Fig. 1. Preferably, the apparatus 200 includes the following elements: a receiver 210; a filtering and demultiplexing unit 220; an input unit 230; a processor 240; and an on-screen display (OSD) unit 250. The receiver 210, the input unit 230, the processor 240, and the OSD unit 250 may preferably include conventional elements of a conventional STB. The filtering and demultiplexing unit 220 preferably includes conventional filtering and demultiplexing hardware, and in a case where MPEG-2 is used, the filtering and demultiplexing unit 220 preferably includes conventional MPEG-2 filtering and demultiplexing hardware.

10 [0047] The receiver 210 preferably receives a plurality of program meta-data virtual containers, at least one of the virtual containers comprising normalized program meta-data of programs having instances in a plurality of time windows. The receiver 210 also preferably receives a set of program identifiers and a set of time-window identifiers that are associated with the virtual containers. It is appreciated that the receiver 210 may be comprised in or associated with a conventional integrated receiver and decoder (IRD) (not shown) of the STB 50.

15 [0048] The input unit 230 preferably receives an input indicating a selected time window, the input preferably at least one of comprises and refers to a time-window identifier identifying the selected time window. The input unit 230 may, for example, include an interface unit (not shown) that receives the input indicating the selected time window, as well as other possible instructions, from a user (not shown) via a remote control 260 communicating via any appropriate communications channel such as, for example, a wireless infrared link.

20 [0049] The processor 240 preferably uses the time-window identifier for filtering the plurality of virtual containers so as to obtain a subset of the program meta-data which corresponds to the selected time window. The time-window identifier includes a bit identifying the selected time window, and preferably exactly one bit identifying the selected time window.

25 [0050] The selected time window preferably includes at least one of the following: a part of a day; a day; more than one day; a week; and a specific weekday within a period of at least one month, such as a first Saturday of a month.

30 [0051] It is appreciated that the input unit 230 may alternatively or additionally receive from the user an input indicating a selected program, the input preferably at least one of comprises and refers to a program identifier identifying the selected program. In such a case, the processor 240 preferably uses the program identifier to filter the plurality of virtual containers so as to obtain a subset of the program meta-data which corresponds to the selected program. The program identifier preferably includes a virtual container identifier (container_id).

35 [0052] Further alternatively or additionally, the input unit 230 may receive from the user an input indicating a selected service, the input preferably at least one of comprises and refers to a service identifier identifying the selected service. In such a case, the processor 240 preferably uses the service identifier to filter the plurality of virtual containers so as to obtain a subset of the program meta-data which corresponds to the selected service. The service identifier preferably includes a container_id.

40 [0053] Still further alternatively or additionally, the input unit 230 may receive from the user an input indicating at least one editorial attribute in the program meta-data. In such a case, the processor 240 preferably filters the plurality of virtual containers so as to obtain a subset of the program meta-data which corresponds to the at least one editorial attribute.

45 [0054] In operation, the receiver 210 preferably receives transmissions in an MPEG-2 format. The transmissions preferably include a plurality of program meta-data virtual containers as well as a set of program identifiers and a set of time-window identifiers that are associated with the virtual containers. The processor 240 preferably processes the virtual containers, the set of program identifiers and the set of time-window identifiers in accordance with inputs inputted by the user via the remote control 260. Processing results of the processor 240 are, for example, provided to the OSD unit 250 for displaying on the display 60 of Fig. 1.

50 [0055] Preferably, the conventional MPEG-2 filtering and demultiplexing hardware in the filtering and demultiplexing unit 220 allows for acquisition of sections using a "filter". Such a filter is preferably applied by the processor 240 and it allows for a number of bits of a start of an MPEG-2 section to be compared with a filter value in such a way that some bits can be ignored, and bits that are not ignored can be compared to a desired value. Thus, the MPEG-2 filtering and demultiplexing hardware at the STB 50 and the processor 240 can be used by the apparatus 200 to filter MPEG-2 sections on a PID table_id_extension and on other fields for a given bit or bits being set or clear within a container_id, and as a result to acquire only programs that are to be presented in an EPG for a pre-specified time window.

55 [0056] A person skilled in the art will appreciate that similar filtering may be performed on IP datagrams in an IP network.

[0057] It is appreciated that the apparatus 200 of Fig. 5 together with the apparatus 100 of Fig. 2 may be referred to as an addressing mechanism that allows the user to filter the program meta-data virtual containers by at least one of the following: time; service; program; and at least one editorial attribute in the program meta-data.

[0058] An example of the operation of the apparatus 100 of Fig. 2 and the apparatus 200 of Fig. 5 is now briefly described.

[0059] A program has, for example, instances occurring on days d_0, d_1, \dots, d_n , where n is the total number of discrete days in which the program has an instance, and d_m is a given day in which the program has an instance. The program is then preferably placed in a virtual container whose identifier is based on bit numbers d_0, d_1, \dots, d_n as follows:

$$\text{container_id} = 2^{d_0} + 2^{d_1} + \dots + 2^{d_n} \quad (1)$$

For example, if the program has instances on d_0 and d_3 , the container_id will be:

$$\text{container_id} = 2^{d_0} + 2^{d_3}$$

In order to acquire all virtual containers for all programs having instances on a given day d_m , all virtual containers that have container_id with bit d_m set are preferably acquired. Therefore, if the user inputs an instruction for displaying programs having instances on day d_m in the EPG, the apparatus 200 preferably filters by performing the following logical operation: container_id AND 2^{d_m} .

[0060] Referring, for example, to an EPG dataset of 8 days comprising days 0 - 7, there are 8 possible bit positions. The container_id may thus be any value from 1 to 255 for all permutations of the 8 days (container_id 0 does not occur because programs that occur in no days do not exist at all). All virtual containers for day 0 have bit 0 set, so 1, 3, 5, 7, ..., 253, and 255. Hence, day 0 would have 128 virtual containers. The same applies for any other day d_x which would have bit x set and 128 permutations of the remaining 7 bits. Thus, it is appreciated that 128 virtual containers are required for any day.

[0061] If all 128 virtual containers for day 0 are cached in a memory cache (not shown), then for any other day only 64 virtual containers are acquired. The reason for that is that 128 virtual containers have already been acquired for day 0 of which 50% have any given bit other than bit 0 set, and 50% have that bit clear, and thus 64 more virtual containers remain to be acquired (all the ones with bit 0 clear). Then, filtering by a logical operation comprising container_id AND $(2^0 | 2^m) = 2^m$ is performed in order to obtain the virtual containers with the bit set that are not in the memory cache, that is the ones with bit 0 clear.

[0062] As mentioned above, the time base does not have to be linear and does not have to refer to entire days. For example, units may be chosen in a non-linear mode to include, for example, the following:

- Unit 0 - the first 6 hours
- Unit 1 - the next 12 hours
- Unit 2 - the next 24 hours
- Unit 3 - the next 24 hours
- Etc.

In such a case, in expression (1) above Unit 0 preferably replaces d_0 , Unit 1 replaces d_1 , etc.

[0063] The cached virtual containers may be organized in a way which is more suitable for data caching. Referring, for example, to cached virtual containers at time window 0, a given number of virtual containers, for example 64 virtual containers, are preferably selected and filled with data equally. Thus, all the 64 virtual containers have approximately the same size. As a result, an integer number n of virtual containers may be cached by having only enough random access memory (RAM) storage capacity to store a total of $n+1$ virtual containers and still maintain access to data during an update.

[0064] Referring, for example to an EPG dataset of 7.25 days, the total number of programs is, for example 13946 and on average 135 bytes uncompressed are budgeted per program for title and synopsis strings. An instance of a program is, for example 10 bytes, and the strings can be compressed to 70%. Normalization is made by title and synopsis only. Taking, for example, the instances that have started or start in the first 6 hours to be 1357 and the amount of uncompressed strings required for these instances to be 183189 bytes, a required STB memory capacity is $((183189 * 70\%) + (1357 * 10)) * 65/64 = 141$ Kbytes (approximately).

[0065] Reference is now made to Fig. 6, which is a simplified flowchart illustration of a preferred method of operation of the apparatus 100 of Fig. 2.

[0066] Preferably, program meta-data of instances of programs which share at least one editorial description item of the content is normalized (step 300) to produce a set of program identifiers. Normalization is preferably performed in order to reduce program meta-data transmission bandwidth. The instances of the programs are preferably grouped (step

310) by time windows to produce a set of time-window identifiers. The program meta-data is then preferably placed (step 320) in virtual containers, and the virtual containers are preferably associated (step 330) with the set of program identifiers and the set of time-window identifiers.

[0067] Reference is now made to Fig. 7, which is a simplified flowchart illustration of a preferred method of operation of the apparatus 200 of Fig. 5.

[0068] Preferably, a plurality of program meta-data virtual containers is received (step 400). At least one of the virtual containers preferably includes normalized program meta-data of programs having instances in a plurality of time windows. An input indicating a selected time window is preferably received (step 410), where the input at least one of comprises and refers to a time-window identifier identifying the selected time window. The time-window identifier is preferably used (step 420) to filter the plurality of virtual containers so as to obtain a subset of the program meta-data which corresponds to the selected time window.

[0069] It is appreciated that various features of the invention which are, for clarity, described in the contexts of separate embodiments may also be provided in combination in a single embodiment. Conversely, various features of the invention which are, for brevity, described in the context of a single embodiment may also be provided separately or in any suitable subcombination.

[0070] It will be appreciated by persons skilled in the art that the present invention is not limited by what has been particularly shown and described hereinabove. Rather the scope of the invention is defined by the appended claims.

Claims

1. Apparatus for grouping program meta-data which relates to all programs offered for viewing in an Electronic Program Guide, the apparatus comprising:

a meta-data processor (120) which is operative to perform the following:

scan the program meta-data to determine program instances which share at least one item of editorial meta-data in the program meta-data;
 group the program meta-data into groups so that each group refers to a set of program instances that share at least one item of editorial meta-data of the program meta-data to produce normalized program meta-data;
 assign each group of grouped program meta-data a unique program identifier, thereby producing a set of program identifiers; and
 group sets of program instances by time window to produce a set of time window identifiers in which each time window identifier is associated with at least one of the time windows; and

an encapsulation unit (130) which is operative to place the program meta-data in virtual containers, and to associate each virtual container with at least one program identifier from the set of program identifiers and one time-window identifier from the set of time-window identifiers, wherein at least one of the virtual containers comprises normalised program metadata of programs having instances in a plurality of time windows.

2. Apparatus according to claim 1 and also comprising:

a transmitter (140) operatively associated with the encapsulation unit (130) and operative to transmit the virtual containers, the set of program identifiers, and the set of time-window identifiers.

3. Apparatus according to claim 2 and wherein the transmitter (140) is also operative to convey the virtual containers, the set of program identifiers, and the set of time-window identifiers as MPEG-2 private sections.

4. Apparatus according to claim 2 and wherein the transmitter (140) is also operative to convey the virtual containers, the set of program identifiers, and the set of time-window identifiers in at least one PID (Packet Identification) stream.

5. Apparatus according to claim 4 and wherein each of the virtual containers is assigned a container identifier, and the transmitter (140) is also operative to convey in a distinct PID stream all virtual containers with container identifiers that have a specific bit set.

6. Apparatus according to claim 4 and wherein the transmitter (140) is also operative to convey all virtual containers that cover multiple periods of time in a distinct PID stream.

7. Apparatus according to claim 4 and wherein the transmitter (140) is also operative to convey in each distinct PID stream a group of virtual containers that covers a single unit of time.
8. Apparatus according to claim 2 and wherein the transmitter (140) is also operative to reduce bandwidth per PID stream by distributing the virtual containers among a plurality of PID streams.
9. Apparatus according to claim 8 and wherein the transmitter (140) is also operative to convey one of the following in a separate one of the plurality of PID streams in order to enable bit assignment filtering of the plurality of PID streams:
- all virtual containers comprising container identifiers that have one of the following: only a single bit set; and a specific bit set; and
 - a group of virtual containers that covers a single unit of time.
10. Apparatus according to any of claims 1 - 9 and wherein the meta-data processor (120) is also operative to select the time windows according to a linear time base.
11. Apparatus according to claim 10 and wherein the linear time base comprises a time unit equivalent to at least one of the following: a part of a day; and a day.
12. Apparatus according to any of claims 1 - 9 and wherein the meta-data processor (120) is also operative to select the time windows according to a non-linear time base.
13. Apparatus according to any of claims 1 - 12 and wherein at least one of the time windows comprises at least one of the following: a part of a day; a day; more than one day; a week; and a specific weekday within a period of at least one month.
14. Apparatus according to any of claims 1 - 13 and wherein the encapsulation unit (130) is also operative to enable access to the virtual containers, the set of program identifiers, and the set of time-window identifiers to allow a user to filter the program meta-data for obtaining a subset of the program meta-data.
15. Apparatus according to claim 14 and wherein the subset of the program meta-data comprises a program meta-data subset which corresponds to at least one of the following: a time window; a service; a program; and at least one item of editorial meta-data in the program meta-data.
16. A method of grouping program meta-data which relates to all programs offered for viewing in an Electronic Program Guide, the method comprising:
- scanning the program meta-data to determine program instances which share at least one item of editorial meta-data in the program meta-data;
 - grouping the program meta-data into groups so that each group refers to a set of program instances that share at least one item of editorial meta-data in the program meta-data to produce normalised program meta-data;
 - assigning each group of grouped program meta-data a unique program identifier, thereby producing a set of program identifiers in which each time window identifier is associated with at least one of the time windows;
 - grouping sets of program instances by time window to produce a set of time window identifiers;
 - placing the program meta-data in virtual containers; and
 - associating each virtual container with at least one program identifier from the set of program identifiers and one time-window identifier from the set of time-window identifiers, wherein at least one of the virtual containers comprises normalised program metadata of programs having instances in a plurality of time windows.

Patentansprüche

1. Vorrichtung zum Gruppieren von Programmmetadaten, die sich auf alle in einem elektronischen Programmführer zum Ansehen angebotenen Programme beziehen, wobei die Vorrichtung umfasst:
- einen Metadatenprozessor (120), der betriebsfähig ist, Folgendes durchzuführen:

- Durchsuchen der Programmmetadaten, um Programminstanzen zu bestimmen, die mindestens einen Ein-

trag von redaktionellen Metadaten in den Programmmetadaten gemeinsam haben;
Gruppieren der Programmmetadaten in Gruppen, so dass sich jede Gruppe auf eine Menge von Programm-
instanzen, die mindestens einen Eintrag von redaktionellen Metadaten in den Programmmetadaten ge-
meinsam haben, bezieht, um normierte Programmmetadaten zu erzeugen;
5 Zuweisen einer eindeutigen Programmkennung zu jeder Gruppe von gruppierten Programmmetadaten,
wodurch eine Menge von Programmkennungen erzeugt wird; und
Gruppieren von Mengen von Programminstanzen nach Zeitfenstern, um eine Menge von Zeitfensterken-
nungen zu erzeugen, in der jede Zeitfensterkennung mit mindestens einem der Zeitfenster verknüpft ist; und

10 eine Verkapselungseinheit (130), die betriebsfähig ist, die Programmmetadaten in virtuellen Behältern unter-
zubringen und jeden virtuellen Behälter mit mindestens einer Programmkennung aus der Menge von Programm-
kennungen und einer Zeitfensterkennung aus der Menge von Zeitfensterkennungen zu verknüpfen, worin min-
destens einer der virtuellen Behälter normierte Programmmetadaten von Programmen mit Instanzen in einer
Vielzahl von Zeitfenstern umfasst.

15 **2.** Vorrichtung nach Anspruch 1, außerdem umfassend:

einen Sender (140), der mit der Verkapselungseinheit (130) betriebsfähig verknüpft ist und betriebsfähig ist,
20 die virtuellen Behälter, die Menge von Programmkennungen und die Menge von Zeitfensterkennungen zu
übertragen.

3. Vorrichtung nach Anspruch 2, worin der Sender (140) außerdem betriebsfähig ist, die virtuellen Behälter, die Menge
von Programmkennungen und die Menge von Zeitfensterkennungen als MPEG-2-Privatabschnitte zu befördern.

25 **4.** Vorrichtung nach Anspruch 2, worin der Sender (140) außerdem betriebsfähig ist, die virtuellen Behälter, die Menge
von Programmkennungen und die Menge von Zeitfensterkennungen in mindestens einem PID-(Paketkennungs-)
Strom zu befördern.

30 **5.** Vorrichtung nach Anspruch 4, worin jedem der virtuellen Behälter eine Behälterkennung zugewiesen ist und der
Sender (140) außerdem betriebsfähig ist, alle virtuellen Behälter mit Behälterkennungen, bei denen eine spezifische
Bitmenge gesetzt ist, in einem gesonderten PID-Strom zu befördern.

6. Vorrichtung nach Anspruch 4, worin der Sender (140) außerdem betriebsfähig ist, alle virtuellen Behälter, die mehrere
35 Zeiträume überdecken, in einem gesonderten PID-Strom zu befördern.

7. Vorrichtung nach Anspruch 4, worin der Sender (140) außerdem betriebsfähig ist, in jedem gesonderten PID-Strom
eine Gruppe von virtuellen Behältern zu befördern, die eine einzige Zeiteinheit überdeckt.

40 **8.** Vorrichtung nach Anspruch 2, worin der Sender (140) außerdem betriebsfähig ist, die Bandbreite pro PID-Strom
durch Verteilen der virtuellen Behälter über eine Vielzahl von PID-Strömen zu verringern.

9. Vorrichtung nach Anspruch 8, worin der Sender (140) außerdem betriebsfähig ist, in einem separaten der Vielzahl
von PID-Strömen eines von Folgendem zu befördern, um Bitzuweisungsfilterung der Vielzahl von PID-Strömen zu
ermöglichen:

alle virtuellen Behälter, die Behälterkennungen umfassen, bei denen eines von Folgendem gesetzt ist: nur eine
einzigste Bitmenge; und eine spezifische Bitmenge; und
eine Gruppe von virtuellen Behältern, die eine einzige Zeiteinheit überdeckt.

50 **10.** Vorrichtung nach einem der Ansprüche 1 bis 9, worin der Metadatenprozessor (120) außerdem betriebsfähig ist,
die Zeitfenster gemäß einer linearen Zeitbasis auszuwählen.

11. Vorrichtung nach Anspruch 10, worin die lineare Zeitbasis eine Zeiteinheit umfasst, die mindestens einem der
Folgenden entspricht: ein Teil eines Tages; und ein Tag.

55 **12.** Vorrichtung nach einem der Ansprüche 1 bis 9, worin der Metadatenprozessor (120) außerdem betriebsfähig ist,
die Zeitfenster gemäß einer nichtlinearen Zeitbasis auszuwählen.

13. Vorrichtung nach einem der Ansprüche 1 bis 12, worin mindestens eines der Zeitfenster mindestens eines von Folgendem umfasst: ein Teil eines Tages; ein Tag; mehr als ein Tag; eine Woche; und ein spezifischer Wochentag innerhalb einer Periode von mindestens einem Monat.

14. Vorrichtung nach einem der Ansprüche 1 bis 13, worin die Verkapselungseinheit (130) außerdem betriebsfähig ist, den Zugriff auf die virtuellen Behälter, die Menge von Programmkennungen und die Menge von Zeitfensterkennungen zu ermöglichen, um einem Benutzer zu gestatten, die Programmmetadaten zu filtern, um eine Teilmenge der Programmmetadaten zu erhalten.

15. Vorrichtung nach Anspruch 14, worin die Teilmenge der Programmmetadaten eine Programmmetadaten-Teilmenge umfasst, die mindestens einem von Folgendem entspricht: ein Zeitfenster; ein Dienst; ein Programm; und mindestens ein Eintrag von redaktionellen Metadaten in den Programmmetadaten.

16. Verfahren zum Gruppieren von Programmmetadaten, die sich auf alle in einem elektronischen Programmführer zum Ansehen angebotenen Programme beziehen, wobei das Verfahren umfasst:

Durchsuchen der Programmmetadaten, um Programminstanzen zu bestimmen, die mindestens einen Eintrag von redaktionellen Metadaten in den Programmmetadaten gemeinsam haben;

Gruppieren der Programmmetadaten in Gruppen, so dass sich jede Gruppe auf eine Menge von Programminstanzen, die mindestens einen Eintrag von redaktionellen Metadaten in den Programmmetadaten gemeinsam haben, bezieht, um normierte Programmmetadaten zu erzeugen;

Zuweisen einer eindeutigen Programmkennung zu jeder Gruppe von gruppierten Programmmetadaten, wodurch eine Menge von Programmkennungen erzeugt wird, in der jede Zeitfensterkennung mit mindestens einem der Zeitfenster verknüpft ist;

Gruppieren von Mengen von Programminstanzen nach Zeitfenstern, um eine Menge von Zeitfensterkennungen zu erzeugen;

Unterbringen der Programmmetadaten in virtuellen Behältern; und

Verknüpfen jedes virtuellen Behälters mit mindestens einer Programmkennung aus der Menge von Programmkennungen und einer Zeitfensterkennung aus der Menge von Zeitfensterkennungen, worin mindestens einer der virtuellen Behälter normierte Programmmetadaten von Programmen mit Instanzen in einer Vielzahl von Zeitfenstern umfasst.

Revendications

1. Dispositif destiné à grouper des métadonnées de programmes lesquelles concernent la totalité des programmes proposés à l'affichage dans un guide électronique des programmes, le dispositif comprenant :

un processeur de métadonnées (120) lequel est exploitable de manière à mettre en oeuvre les étapes ci-dessous consistant à :

balayer les métadonnées de programmes en vue de déterminer des instances de programmes qui partagent au moins un élément de métadonnées éditoriales dans les métadonnées de programmes ;

regrouper les métadonnées de programmes en des groupes, de sorte que chaque groupe renvoie à un ensemble d'instances de programmes qui partagent au moins un élément de métadonnées éditoriales dans les métadonnées de programmes, en vue de produire des métadonnées de programmes normalisées ;

affecter, à chaque groupe de métadonnées de programmes regroupées, un identifiant de programme unique, ce qui permet de produire par conséquent un ensemble d'identifiants de programmes ; et

regrouper des ensembles d'instances de programmes par fenêtre temporelle en vue de produire un ensemble d'identifiants de fenêtres temporelles, dans lequel chaque identifiant de fenêtre temporelle est associé à au moins l'une des fenêtres temporelles ; et

une unité d'encapsulation (130) qui est exploitable de manière à placer les métadonnées de programmes dans des conteneurs virtuels, et à associer chaque conteneur virtuel à au moins un identifiant de programme de l'ensemble d'identifiants de programmes et un identifiant de fenêtre temporelle de l'ensemble d'identifiants de fenêtres temporelles, dans lequel au moins l'un des conteneurs virtuels comprend des métadonnées de programmes normalisées de programmes présentant des instances dans une pluralité de fenêtres temporelles.

2. Dispositif selon la revendication 1, comprenant en outre :

un émetteur (140) fonctionnellement associé à l'unité d'encapsulation (130) et exploitable de manière à transmettre les conteneurs virtuels, l'ensemble d'identifiants de programmes et l'ensemble d'identifiants de fenêtres temporelles.

3. Dispositif selon la revendication 2, dans lequel l'émetteur (140) est en outre exploitable de manière à acheminer les conteneurs virtuels, l'ensemble d'identifiants de programmes et l'ensemble d'identifiants de fenêtres temporelles sous la forme de sections privées au format MPEG-2.

4. Dispositif selon la revendication 2, dans lequel l'émetteur (140) est en outre exploitable de manière à acheminer les conteneurs virtuels, l'ensemble d'identifiants de programmes et l'ensemble d'identifiants de fenêtres temporelles dans au moins un flux d'identification de paquets (PID).

5. Dispositif selon la revendication 4, dans lequel à chacun des conteneurs virtuels est affecté un identifiant de conteneur, et l'émetteur (140) est en outre exploitable de manière à acheminer dans un flux d'identification PID distinct tous les conteneurs virtuels dont les identifiants de conteneurs présentent un ensemble de bits spécifique.

6. Dispositif selon la revendication 4, dans lequel l'émetteur (140) est en outre exploitable de manière à acheminer tous les conteneurs virtuels qui couvrent plusieurs périodes de temps dans un flux d'identification PID distinct.

7. Dispositif selon la revendication 4, dans lequel l'émetteur (140) est en outre exploitable de manière à acheminer, dans chaque flux d'identification PID distinct, un groupe de conteneurs virtuels qui couvre une unique unité de temps.

8. Dispositif selon la revendication 2, dans lequel l'émetteur (140) est en outre exploitable de manière à réduire la bande passante par flux d'identification PID, en distribuant les conteneurs virtuels sur une pluralité de flux d'identification PID.

9. Dispositif selon la revendication 8, dans lequel l'émetteur (140) est en outre exploitable de manière à acheminer l'un des éléments ci-dessous dans un flux individuel de la pluralité de flux d'identification PID, afin de permettre un filtrage d'affectation de bits de la pluralité de flux d'identification PID :

tous les conteneurs virtuels comprenant des identifiants de conteneurs qui présentent l'un des éléments ci-après : un ensemble de bits unique ; et un ensemble de bits spécifique ; et un groupe de conteneurs virtuels qui couvre une unique unité de temps.

10. Dispositif selon l'une quelconque des revendications 1 à 9, dans lequel le processeur de métadonnées (120) est en outre exploitable de manière à sélectionner les fenêtres temporelles selon une base temporelle linéaire.

11. Dispositif selon la revendication 10, dans lequel la base temporelle linéaire comprend une unité de temps équivalente à au moins l'un des éléments ci-après : une partie d'un jour ; et un jour.

12. Dispositif selon l'une quelconque des revendications 1 à 9, dans lequel le processeur de métadonnées (120) est en outre exploitable de manière à sélectionner les fenêtres temporelles selon une base temporelle non linéaire.

13. Dispositif selon l'une quelconque des revendications 1 à 12, dans lequel au moins l'une des fenêtres temporelles comprend au moins l'un des éléments ci-après : une partie d'un jour ; un jour ; plus qu'un jour ; une semaine ; un jour spécifique de la semaine sur une période d'au moins un mois.

14. Dispositif selon l'une quelconque des revendications 1 à 13, dans lequel l'unité d'encapsulation (130) est en outre exploitable de manière à permettre l'accès aux conteneurs virtuels, à l'ensemble d'identifiants de programmes et à l'ensemble d'identifiants de fenêtres temporelles, afin d'autoriser un utilisateur à filtrer les métadonnées de programmes en vue d'obtenir un sous-ensemble des métadonnées de programmes.

15. Dispositif selon la revendication 14, dans lequel le sous-ensemble des métadonnées de programmes comprend un sous-ensemble de métadonnées de programmes qui correspond à au moins l'un des éléments ci-après : une fenêtre temporelle ; un service ; un programme ; et au moins un élément de métadonnées éditoriales dans les métadonnées de programme.

16. Procédé de groupement de métadonnées de programmes lesquelles concernent la totalité des programmes proposés à l'affichage dans un guide électronique des programmes, le procédé consistant à :

5 balayer les métadonnées de programmes en vue de déterminer des instances de programmes qui partagent au moins un élément de métadonnées éditoriales dans les métadonnées de programmes ;
regrouper les métadonnées de programmes en des groupes, de sorte que chaque groupe renvoie à un ensemble d'instances de programmes qui partagent au moins un élément de métadonnées éditoriales dans les métadonnées de programmes, en vue de produire des métadonnées de programmes normalisées ;
10 affecter, à chaque groupe de métadonnées de programmes regroupées, un identifiant de programme unique, ce qui permet de produire par conséquent un ensemble d'identifiants de programmes, dans lequel chaque identifiant de fenêtre temporelle est associé à au moins l'une des fenêtres temporelles ;
regrouper des ensembles d'instances de programmes par fenêtre temporelle en vue de produire un ensemble d'identifiants de fenêtres temporelles ;
15 placer les métadonnées de programmes dans des conteneurs virtuels ; et
associer chaque conteneur virtuel à au moins un identifiant de programme de l'ensemble d'identifiants de programmes et un identifiant de fenêtre temporelle de l'ensemble d'identifiants de fenêtres temporelles, dans lequel au moins l'un des conteneurs virtuels comprend des métadonnées de programmes normalisées de programmes présentant des instances dans une pluralité de fenêtres temporelles.

20

25

30

35

40

45

50

55

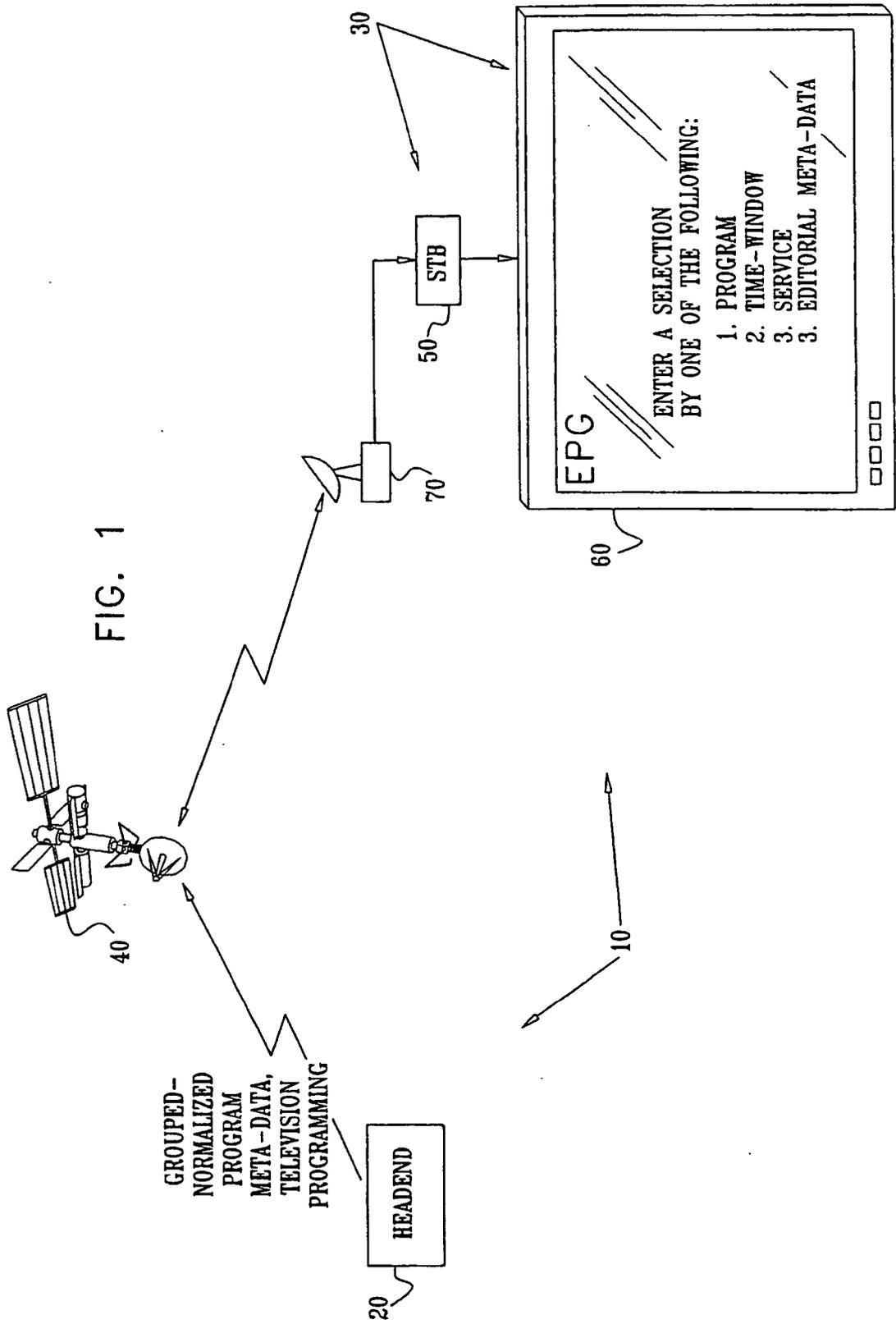


FIG. 2

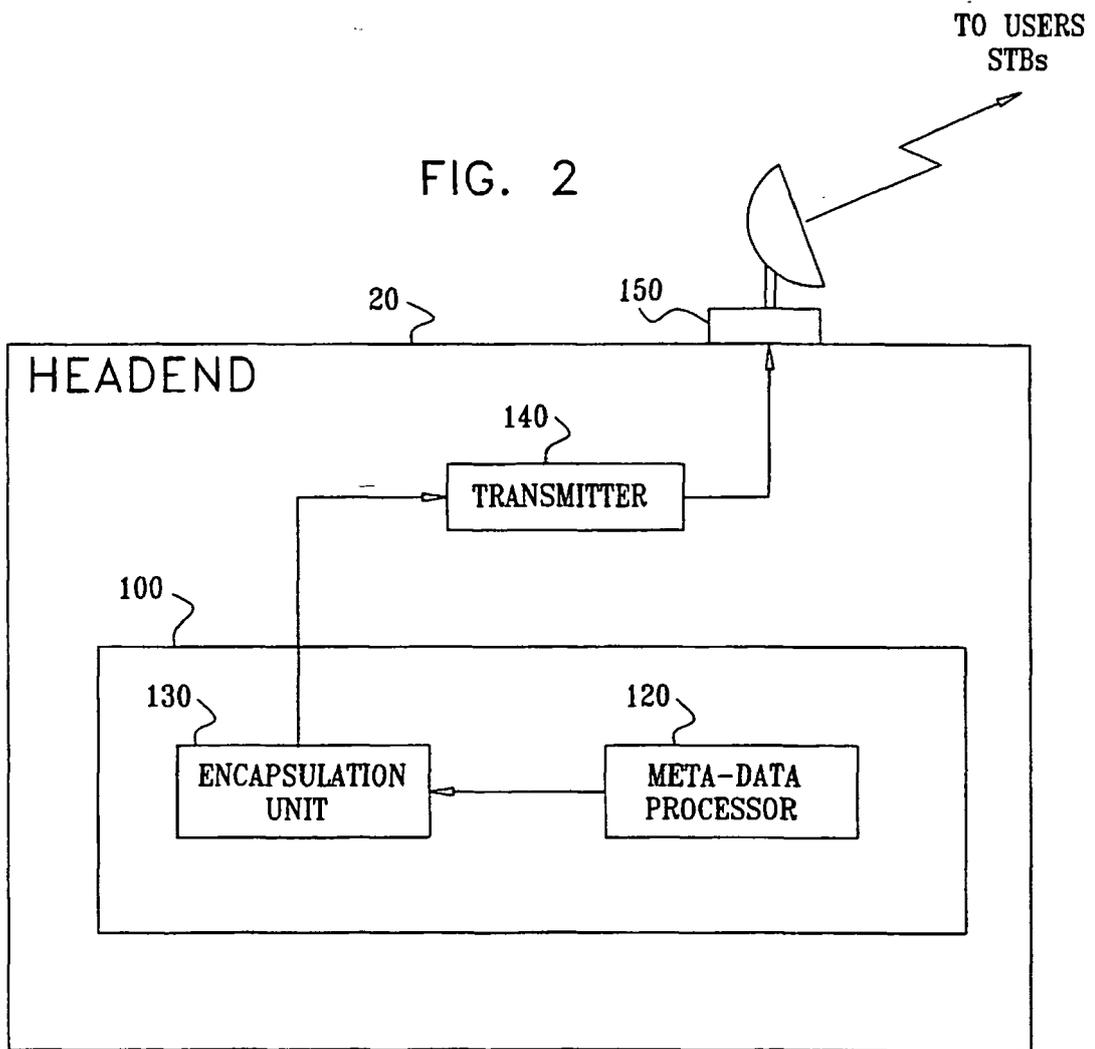


FIG. 3

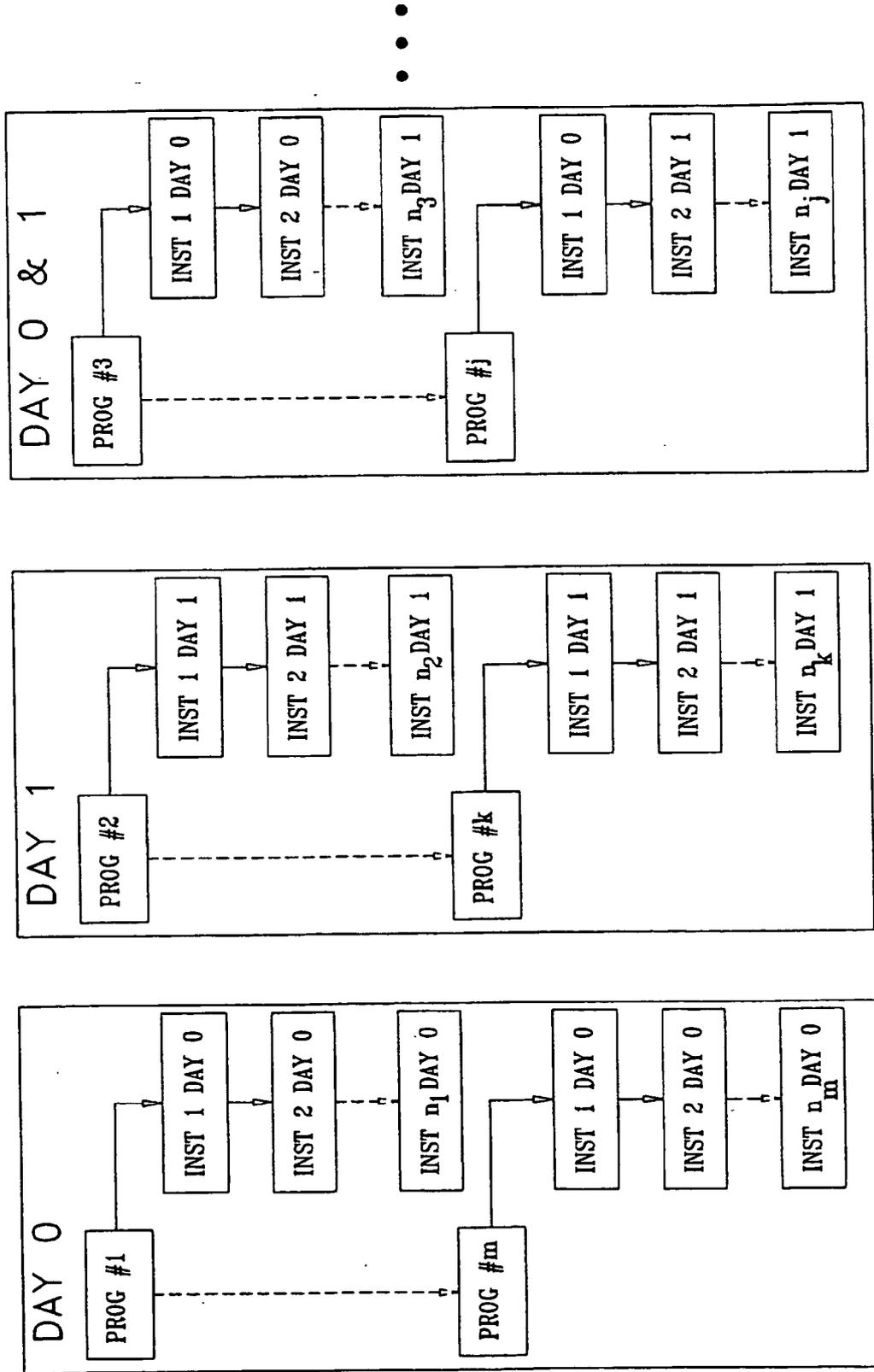


FIG. 4

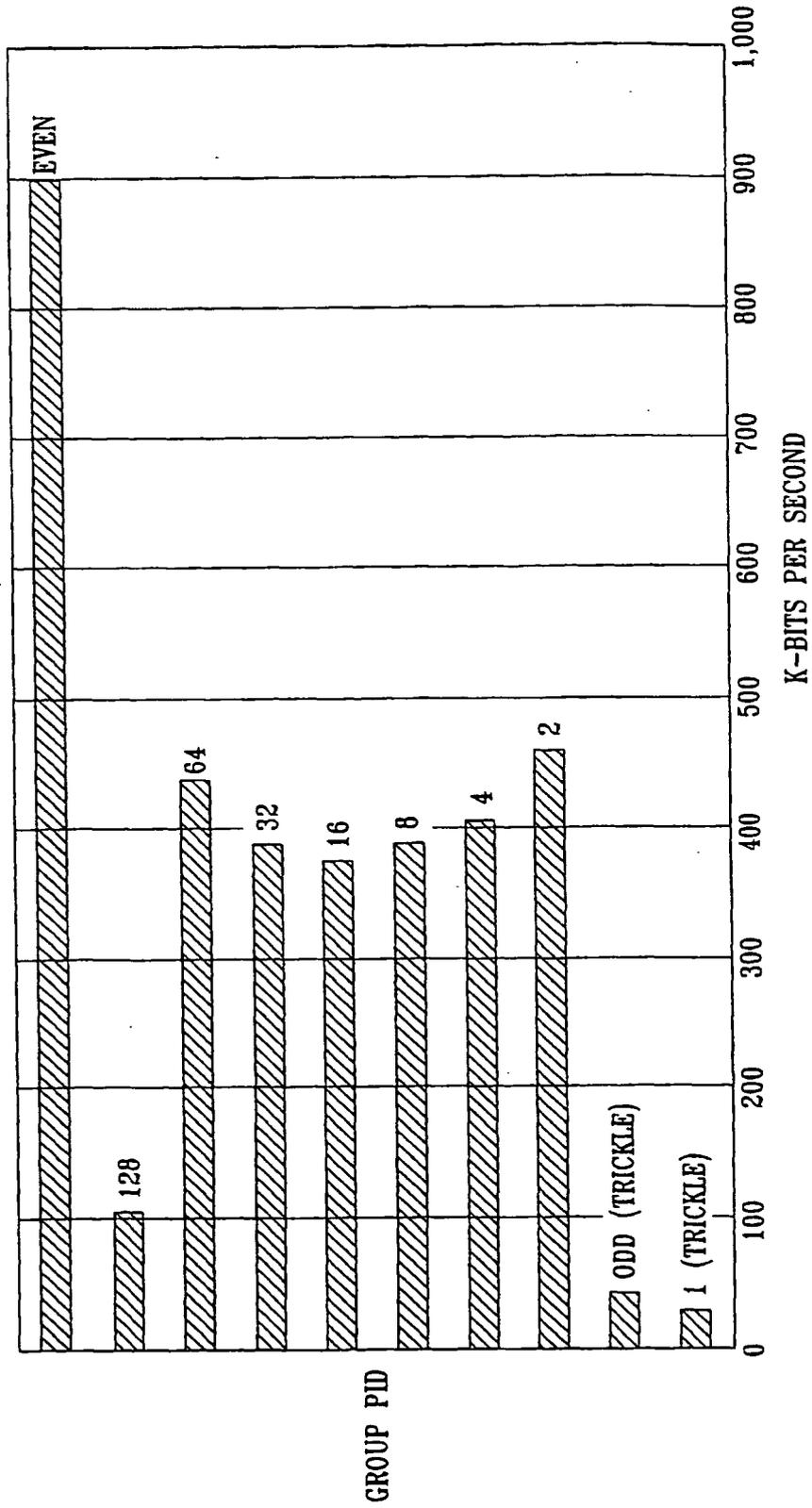


FIG. 5

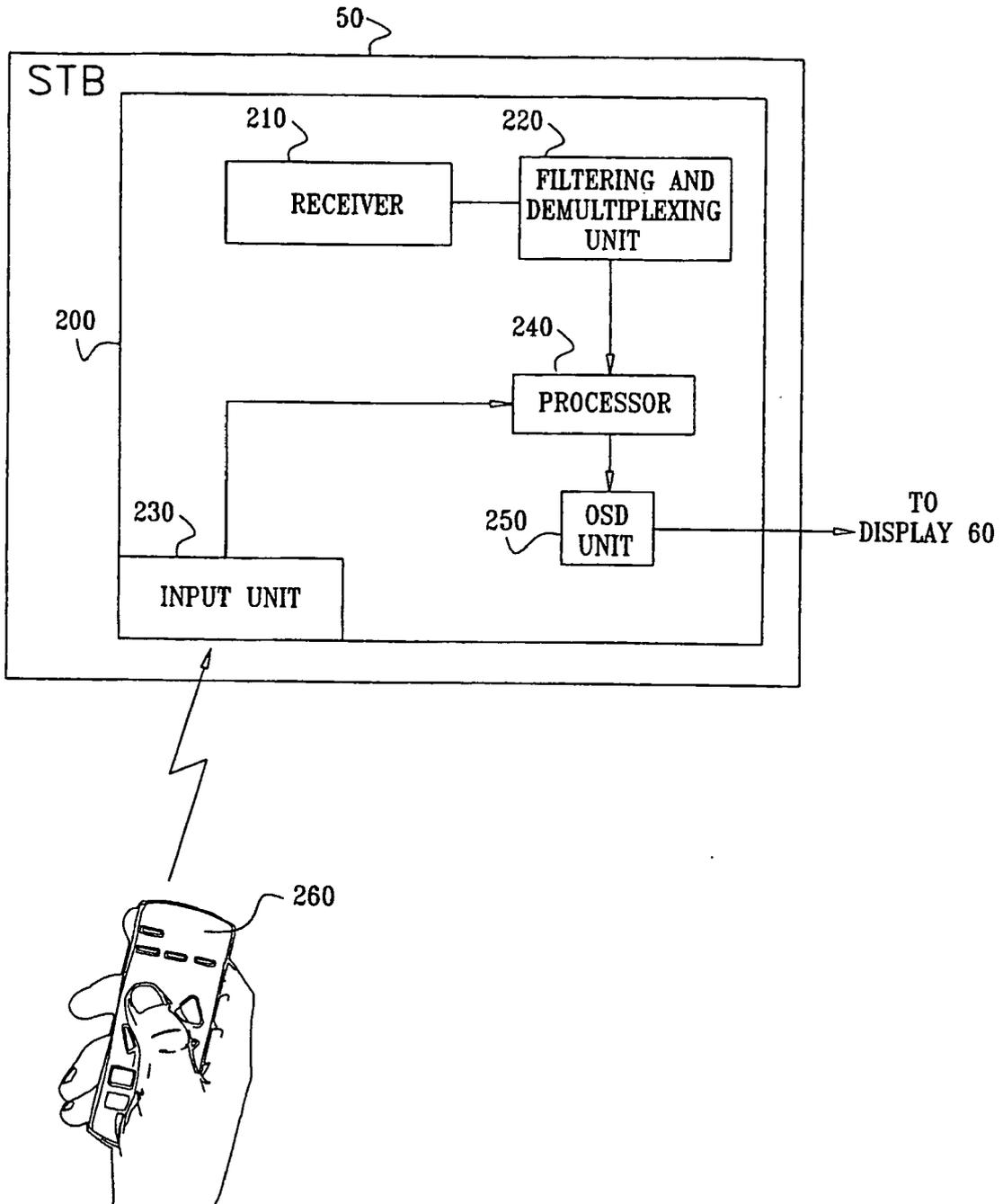


FIG. 6

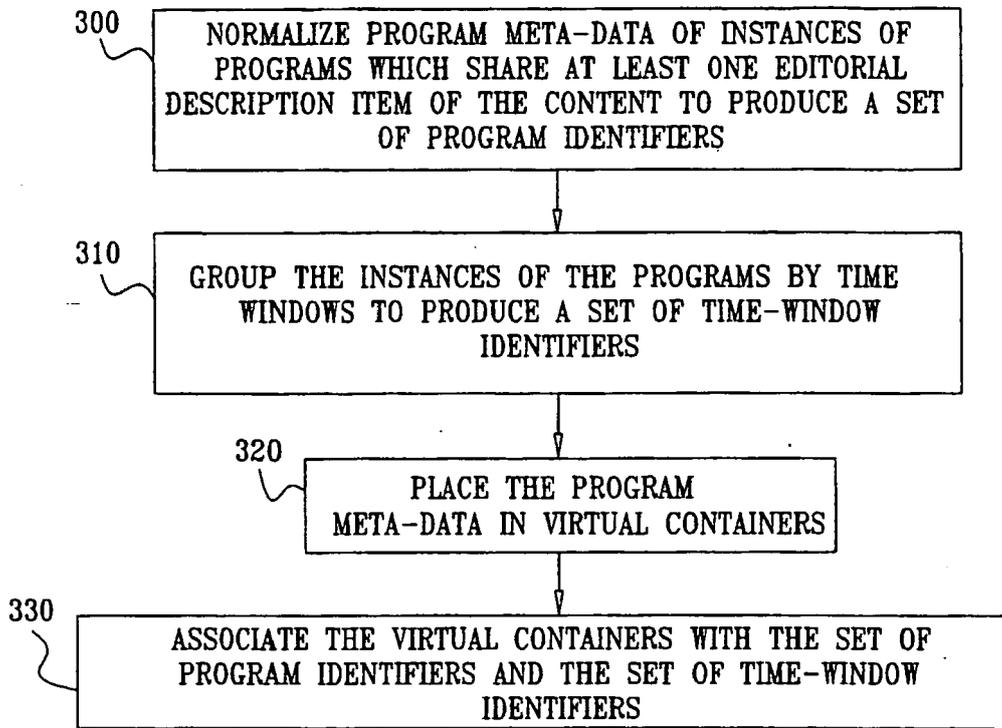
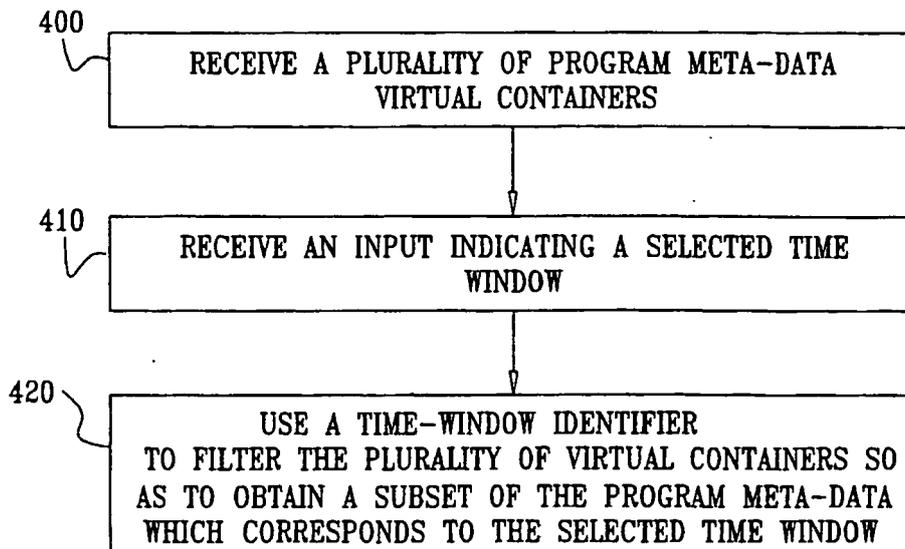


FIG. 7



REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20030208761 A [0007]
- US 5844620 A [0008]