



US 20060127051A1

(19) **United States**(12) **Patent Application Publication****Tsumagari et al.**(10) **Pub. No.: US 2006/0127051 A1**(43) **Pub. Date: Jun. 15, 2006**(54) **INFORMATION RECORDING MEDIUM,
INFORMATION PLAYBACK METHOD, AND
INFORMATION PLAYBACK APPARATUS**(30) **Foreign Application Priority Data**

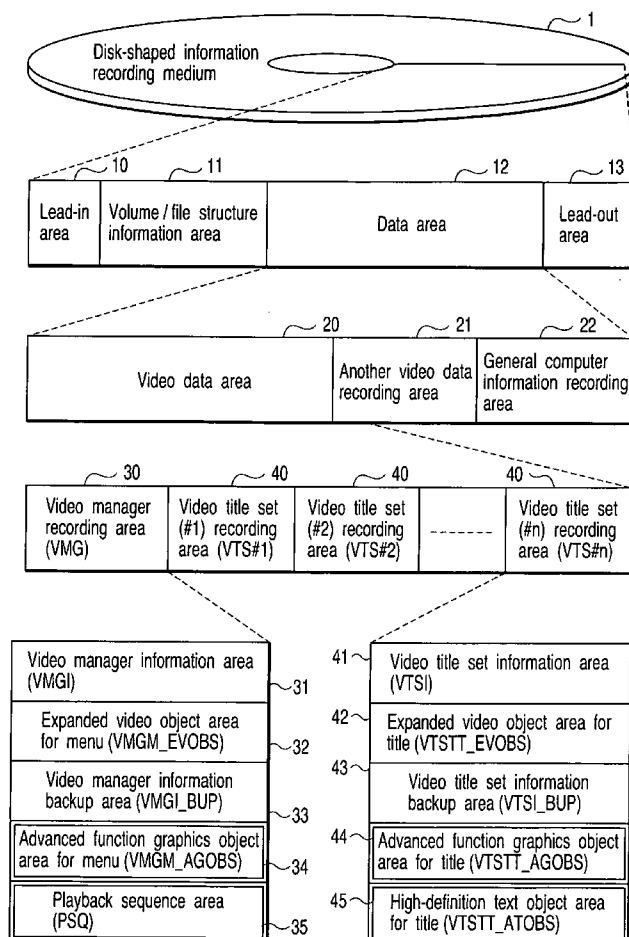
Jun. 16, 2004 (JP) 2004-177730

(76) Inventors: **Yasufumi Tsumagari**, Yokohama-shi
(JP); **Hideki Mimura**, Yokohama-shi
(JP); **Takero Kobayashi**, Akishima-shi
(JP); **Kazuhiko Taira**, Yokohama-shi
(JP); **Yoichiro Yamagata**,
Yokohama-shi (JP)**Publication Classification**(51) **Int. Cl.**
H04N 5/91 (2006.01)(52) **U.S. Cl.** **386/95**

Correspondence Address:

**OBLON, SPIVAK, MCCLELLAND, MAIER &
NEUSTADT, P.C.**
1940 DUKE STREET
ALEXANDRIA, VA 22314 (US)(21) Appl. No.: **11/350,373**(22) Filed: **Feb. 9, 2006****Related U.S. Application Data**(63) Continuation of application No. PCT/JP05/11097,
filed on Jun. 10, 2005.(57) **ABSTRACT**

This invention has as its object to provide more colorful menus and to improve interactiveness in contents playback from a recording medium. In an information recording medium having a data area including a management area and object area, the object area stores expanded video objects which undergo playback management for respective program chains, and advanced objects which are recorded independently of the expanded video objects. The management area stores a playback sequence which gives the playback conditions of the advanced objects. This playback sequence can include information used to update the playback sequence.



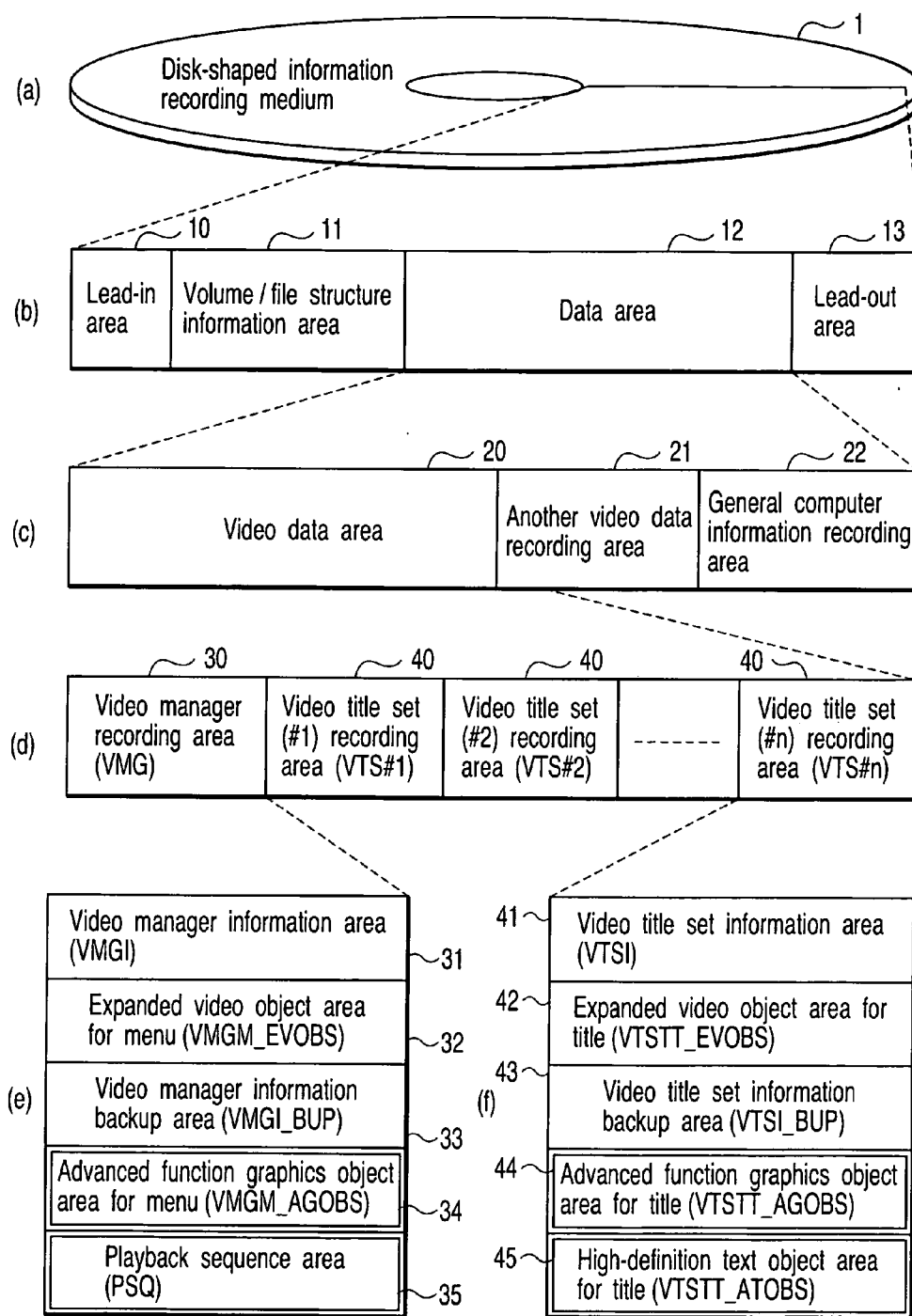


FIG. 1

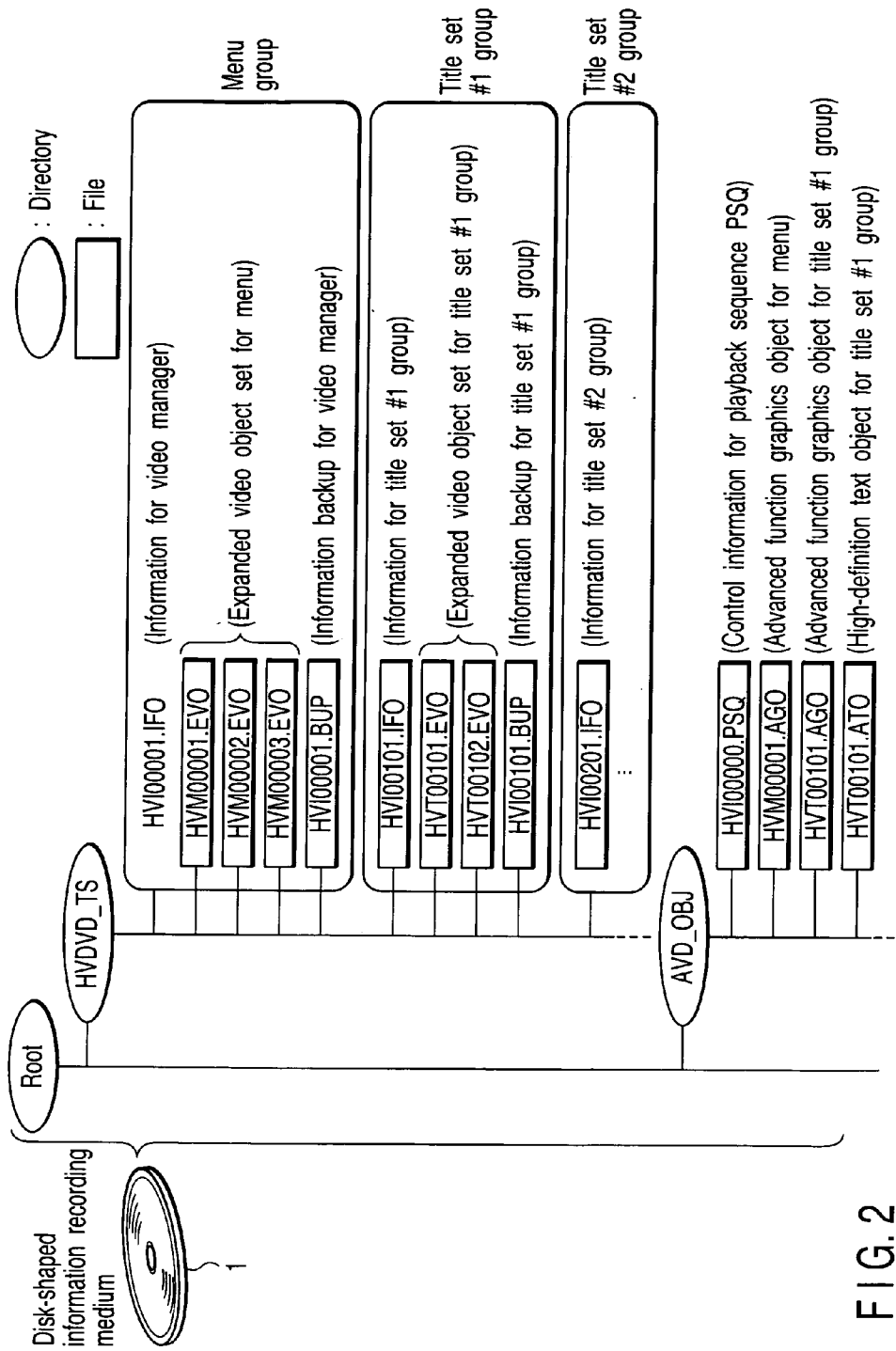


FIG. 2

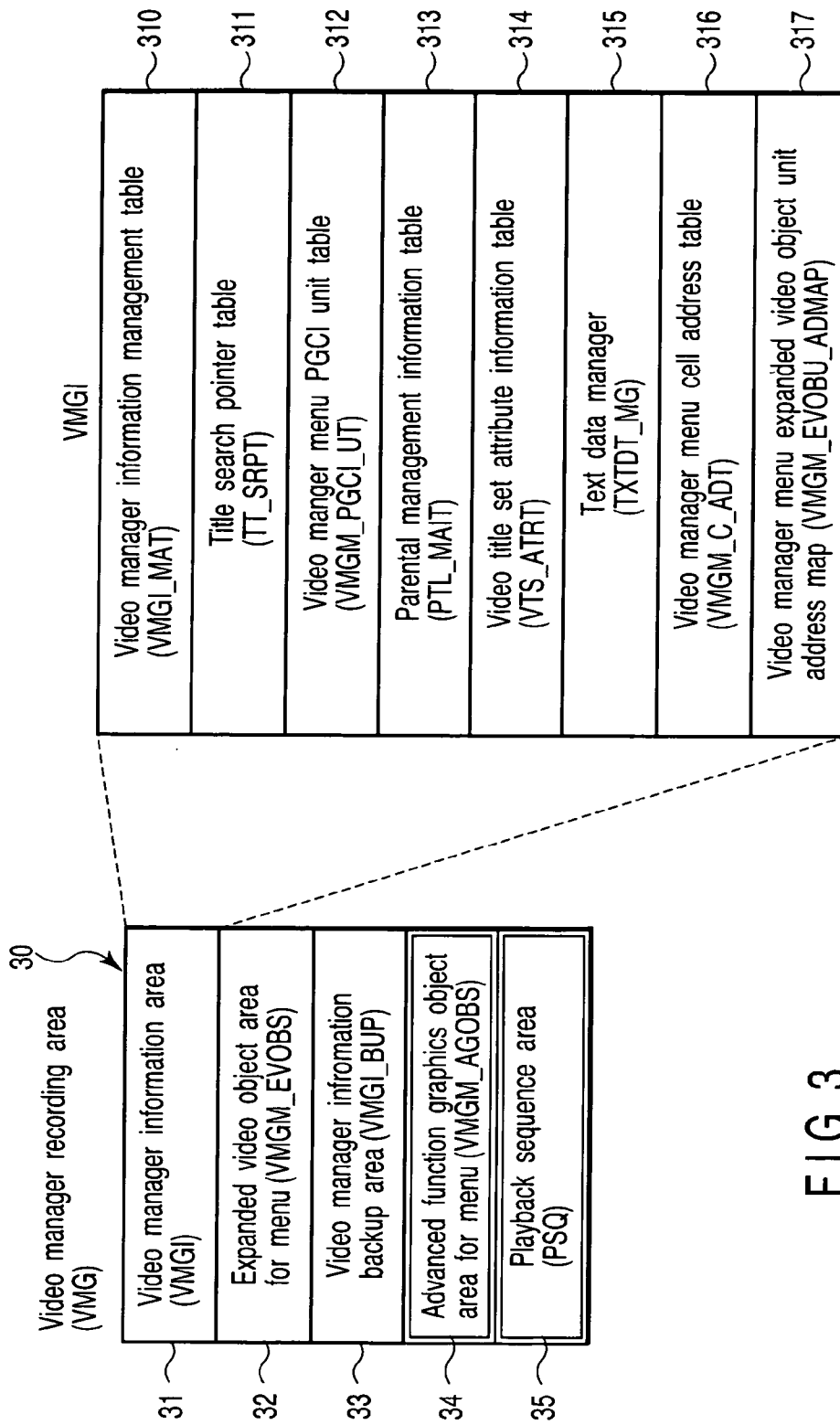


FIG. 3

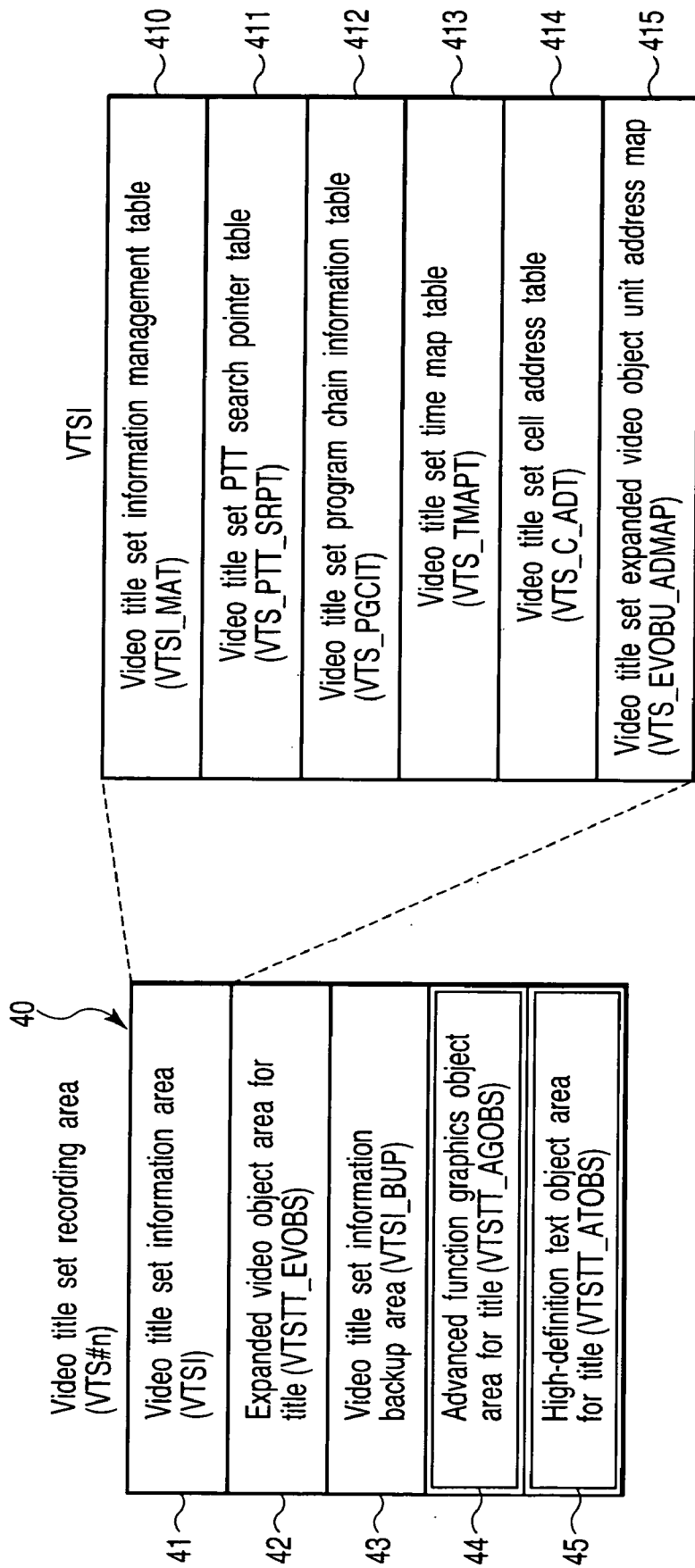


FIG. 4

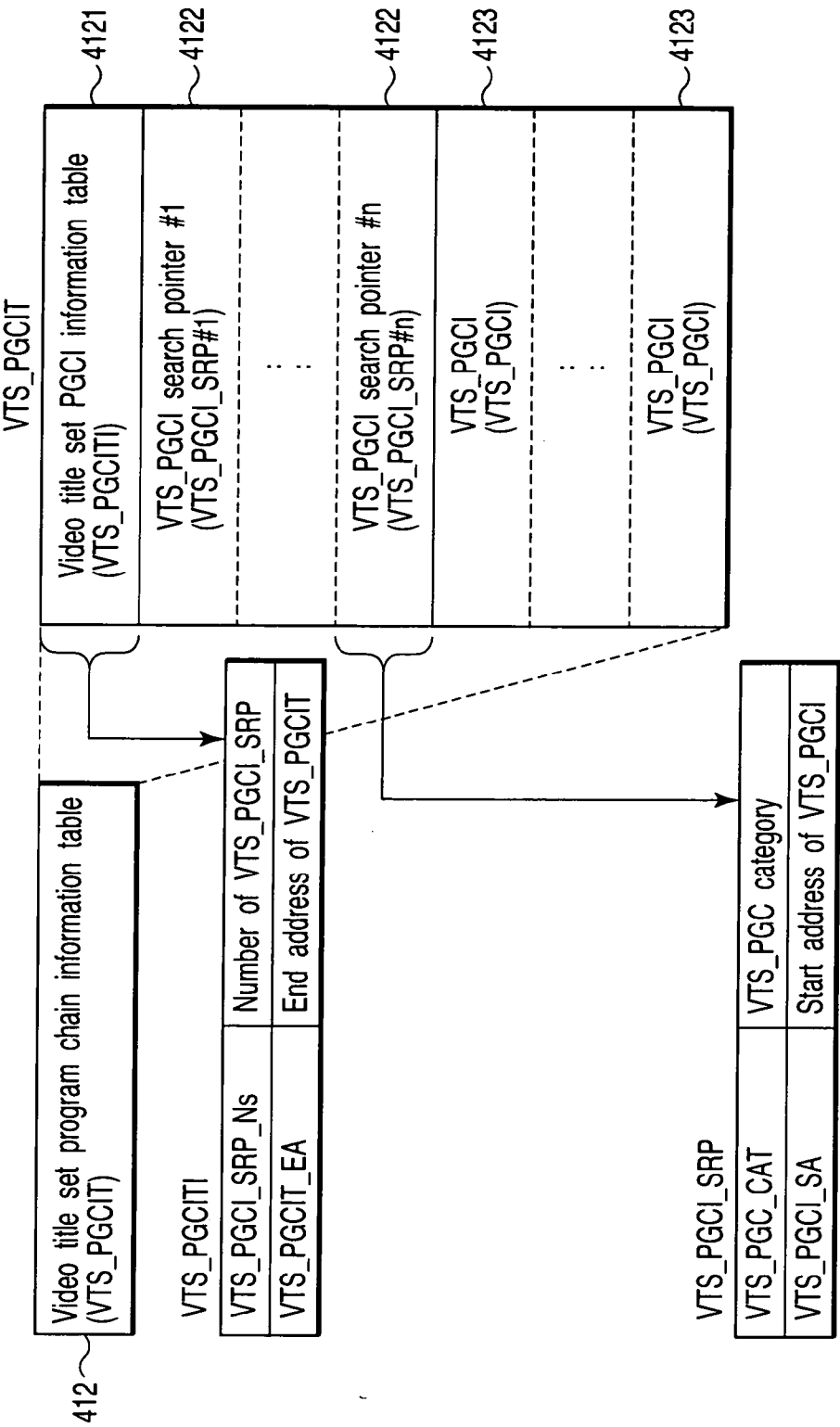


FIG. 5

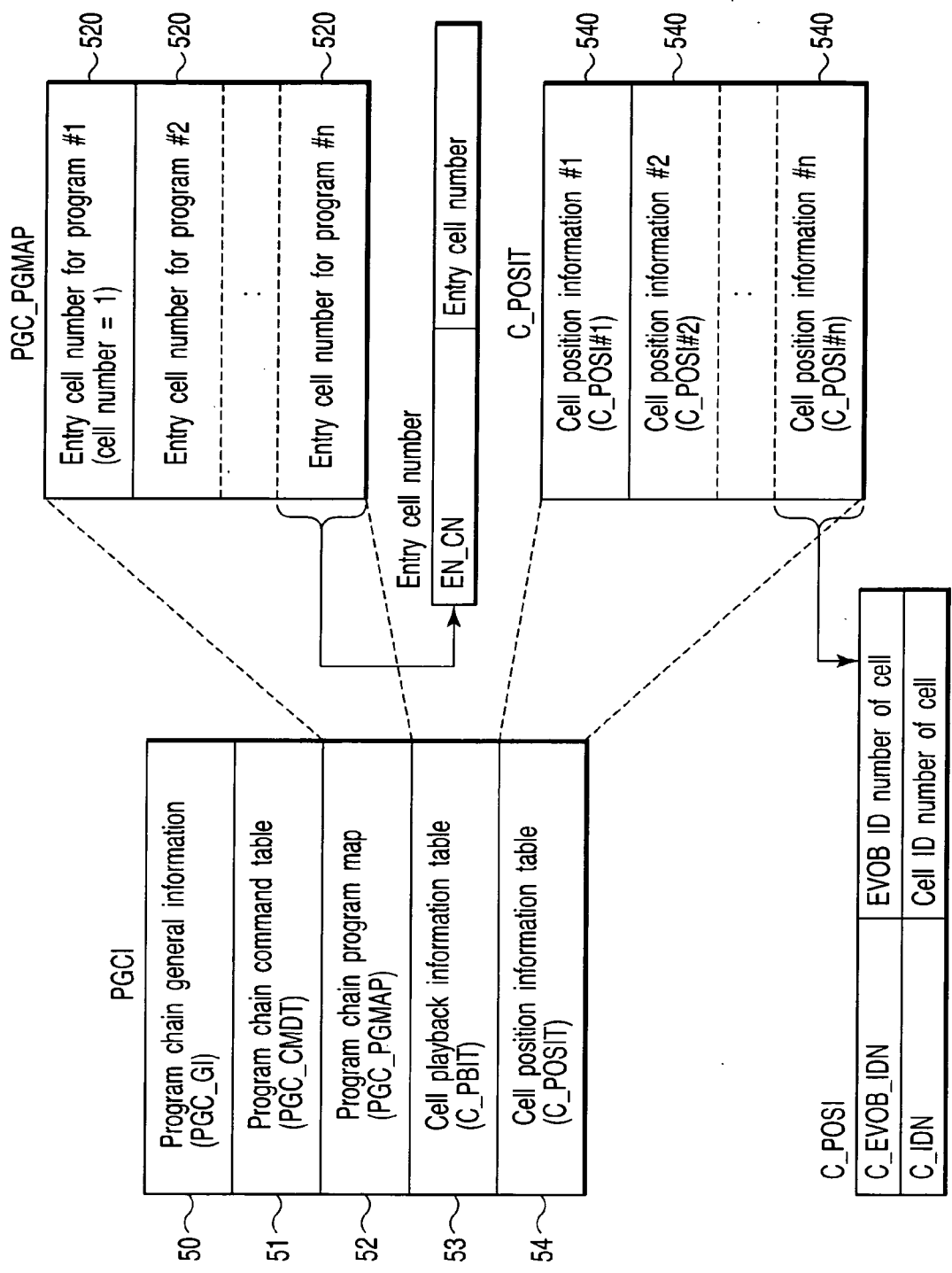


FIG. 6

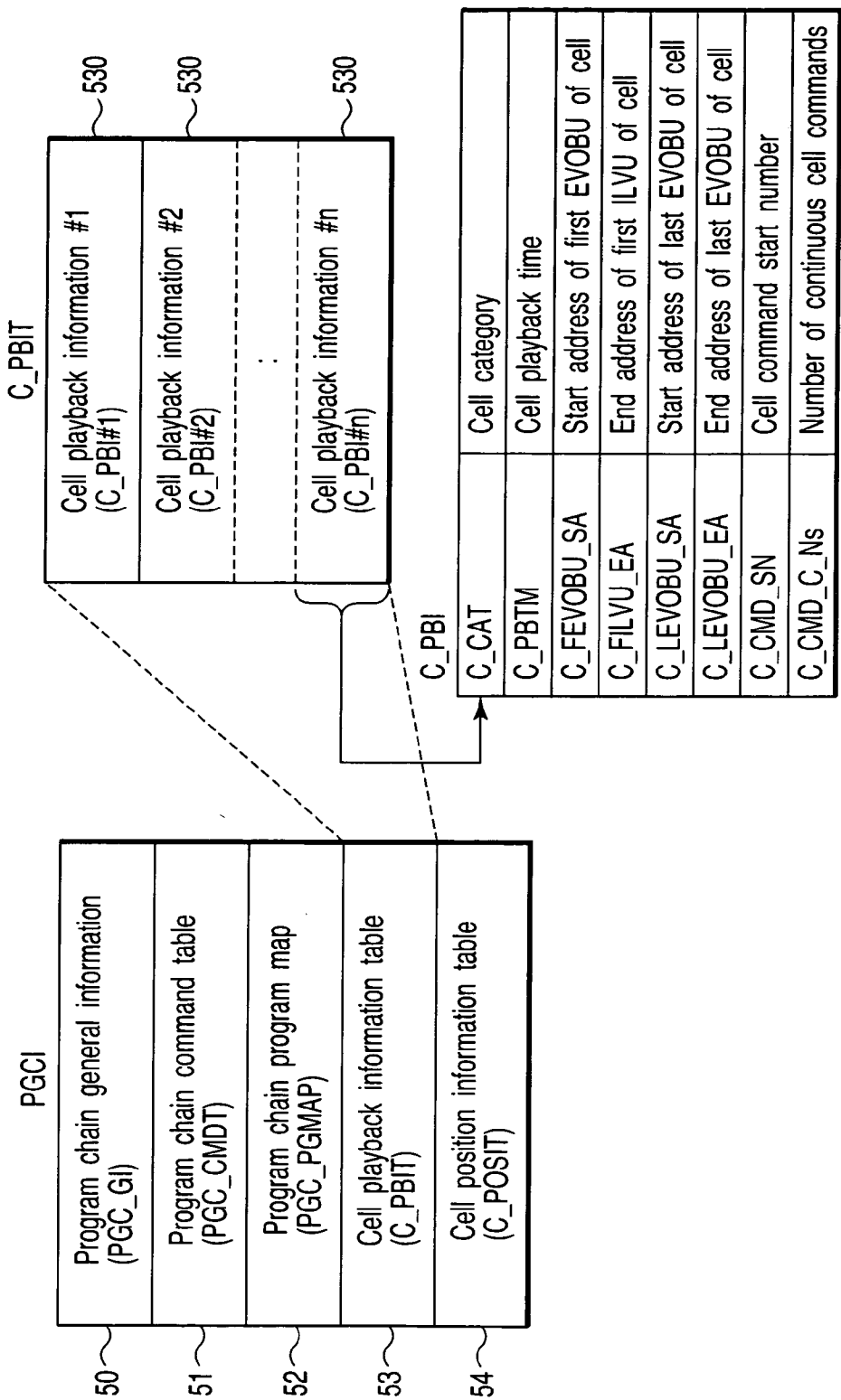


FIG. 7

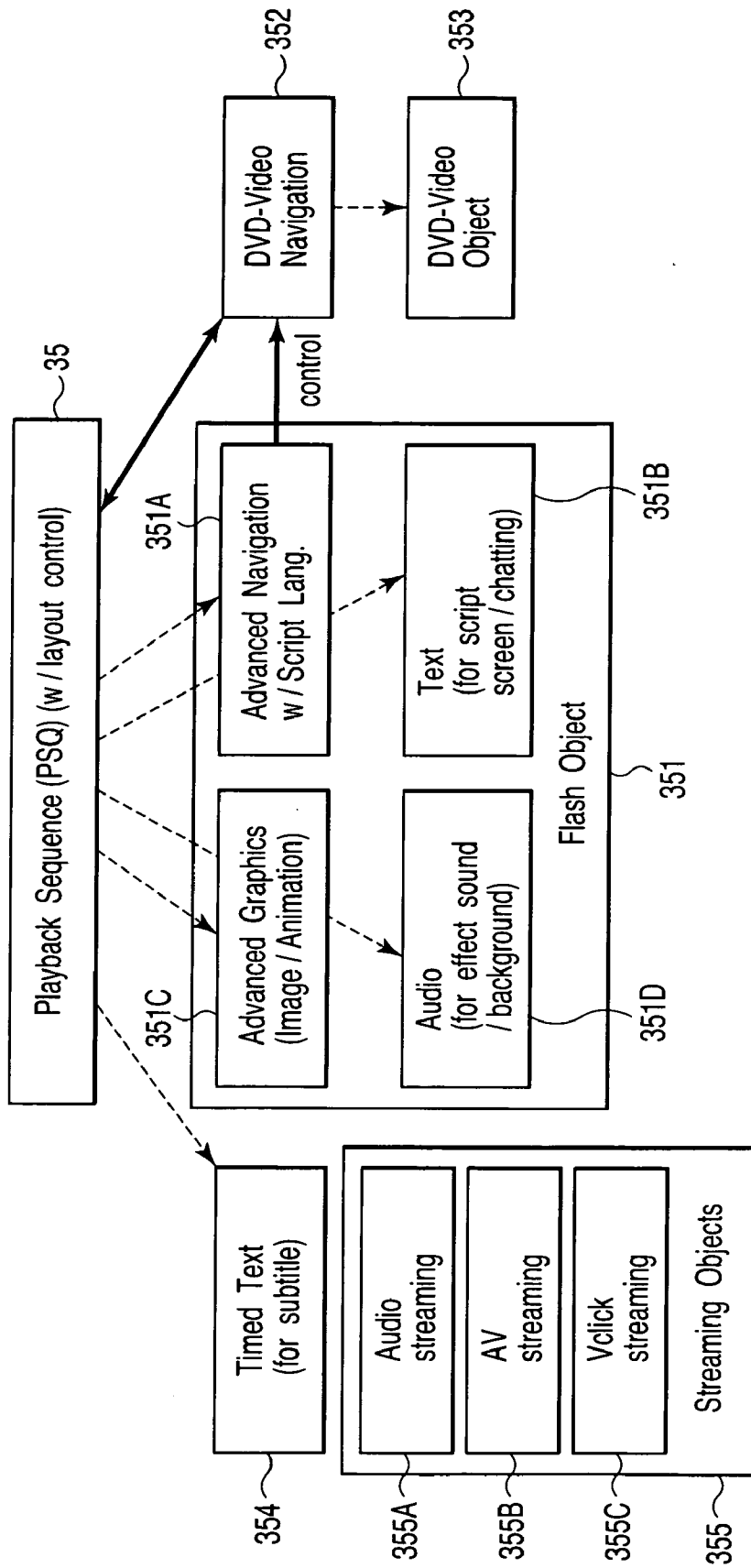


FIG. 8

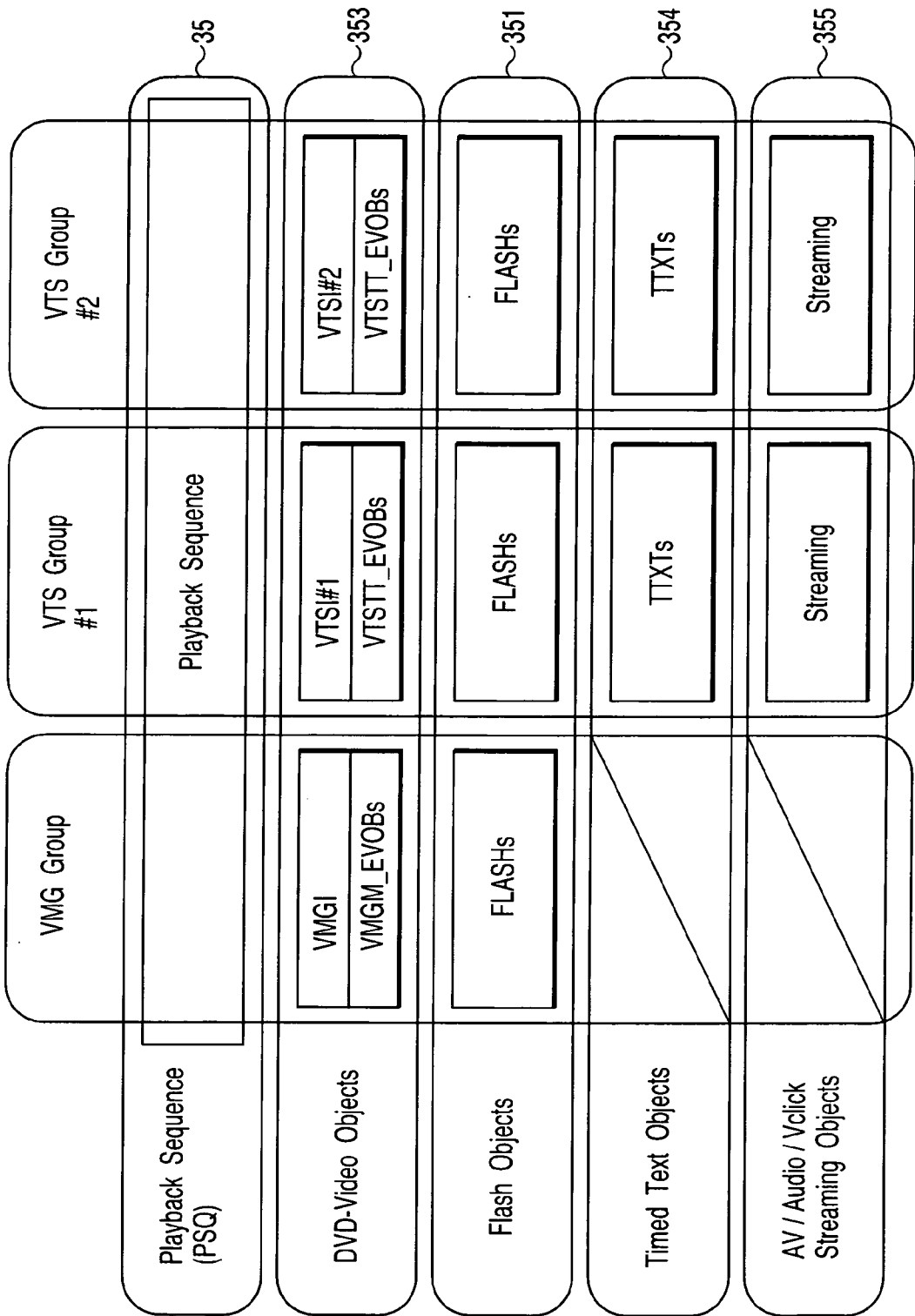


FIG. 9

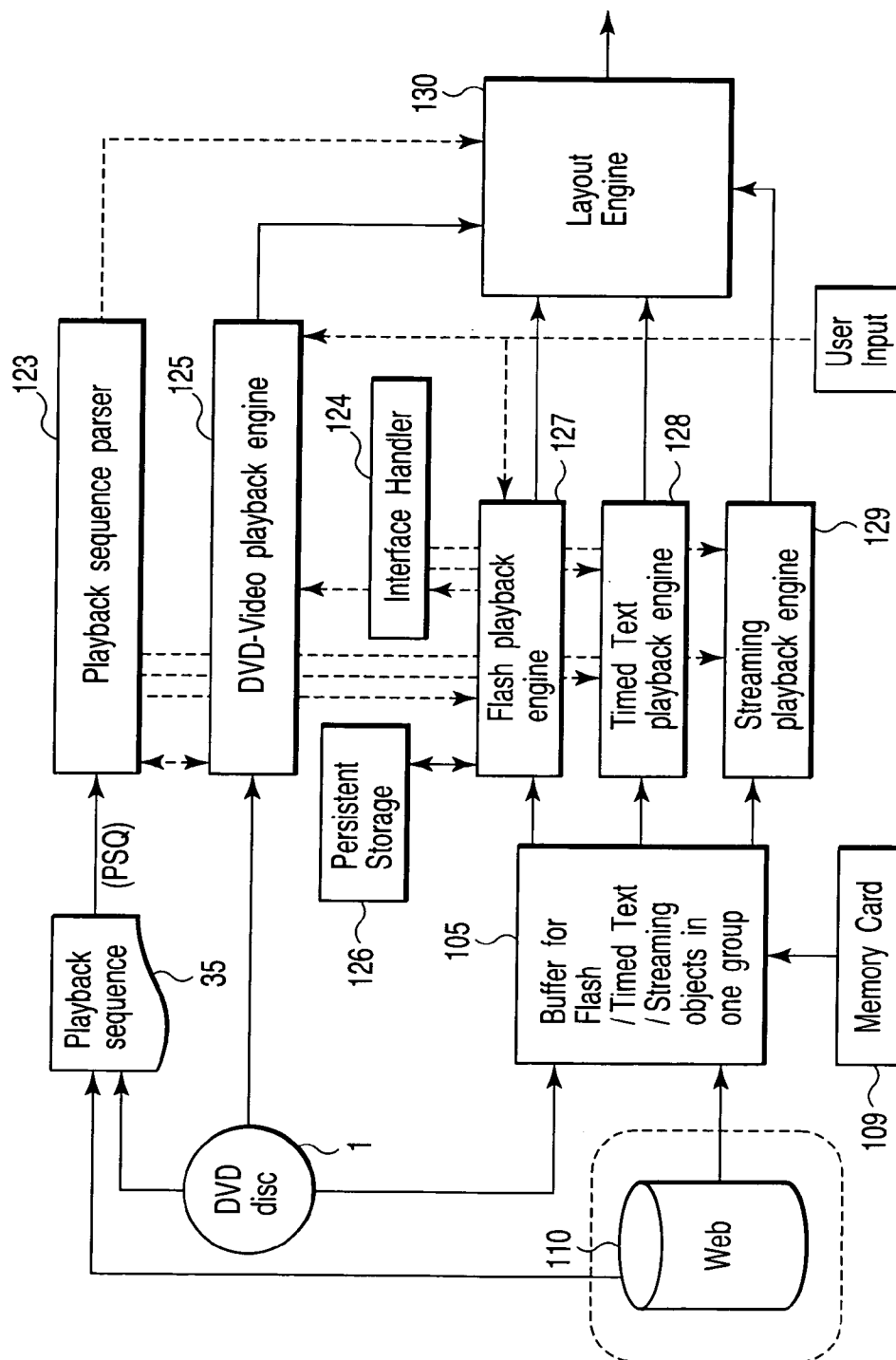


FIG. 10

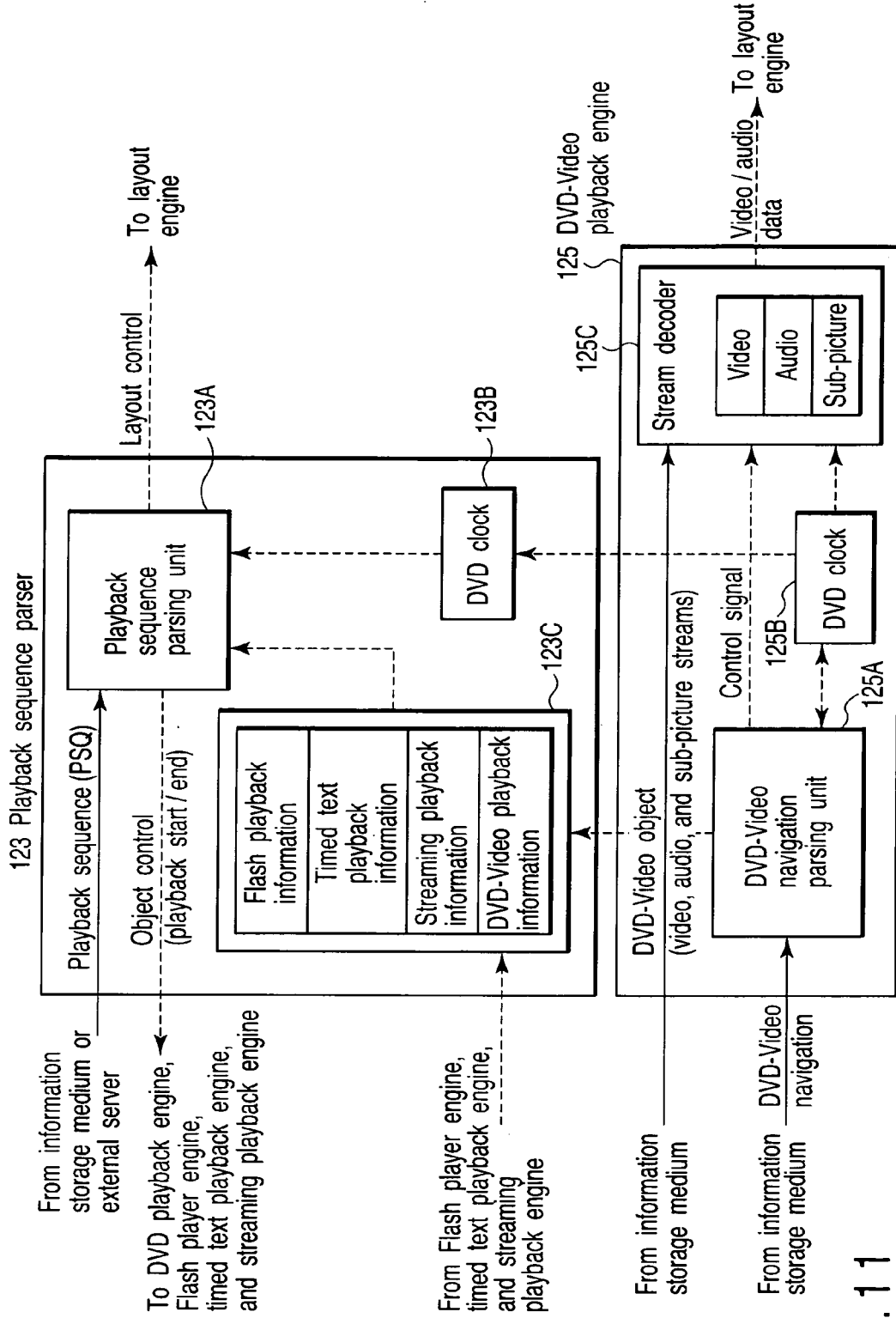


FIG. 11

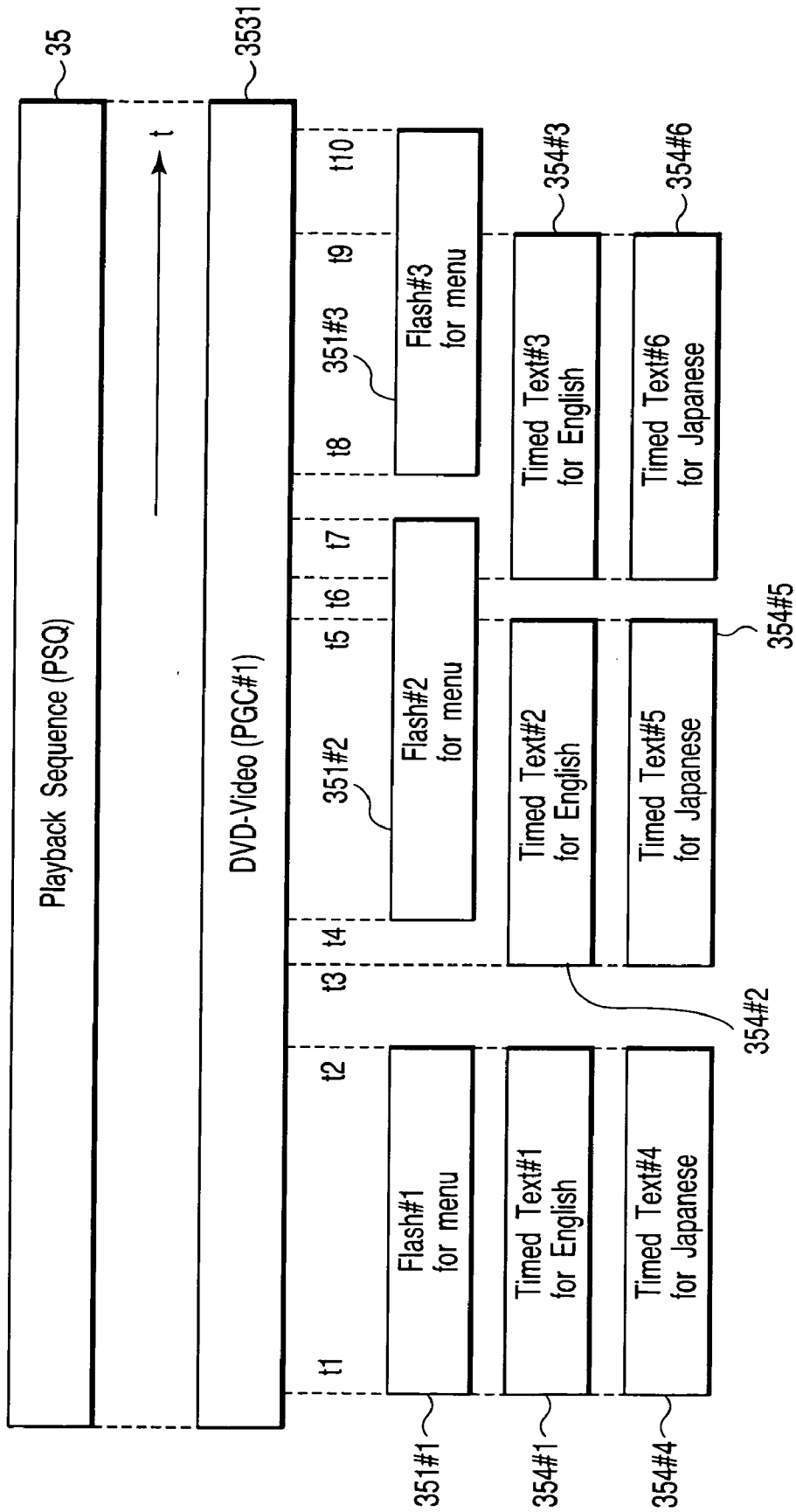


FIG. 12

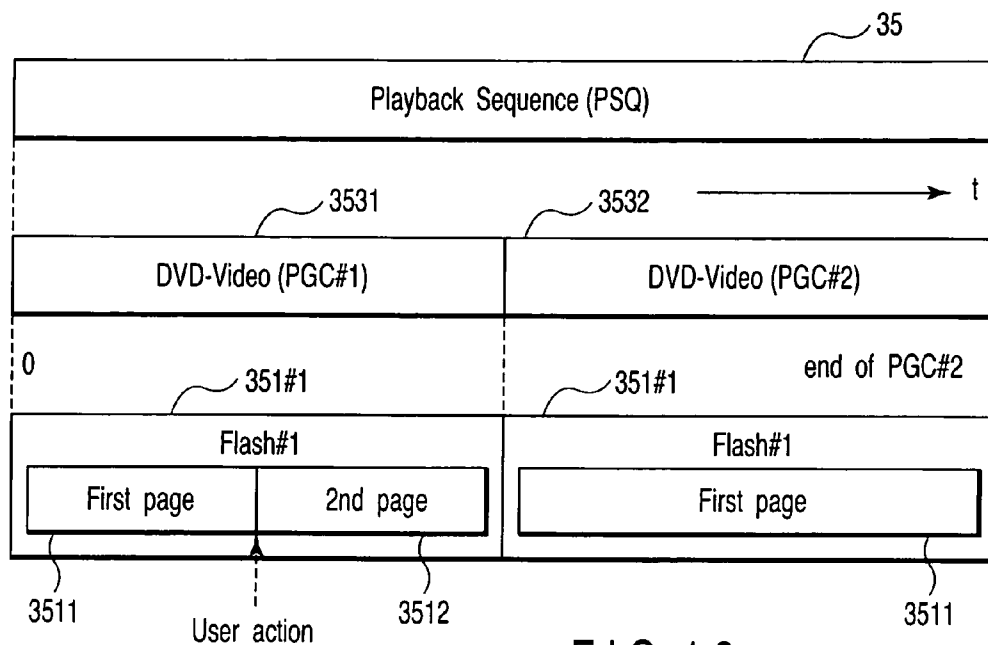


FIG. 13

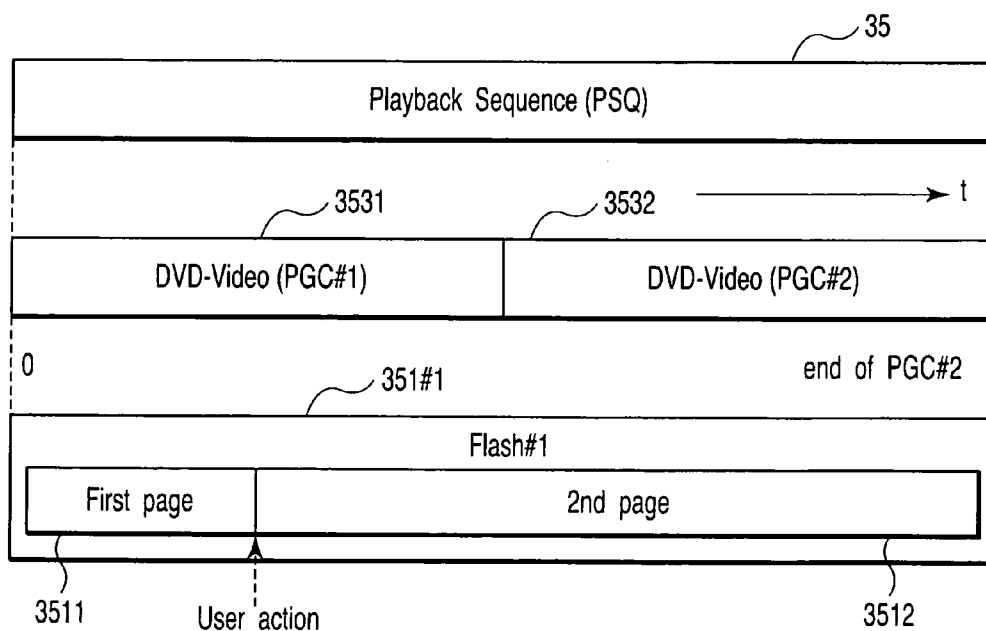


FIG. 14

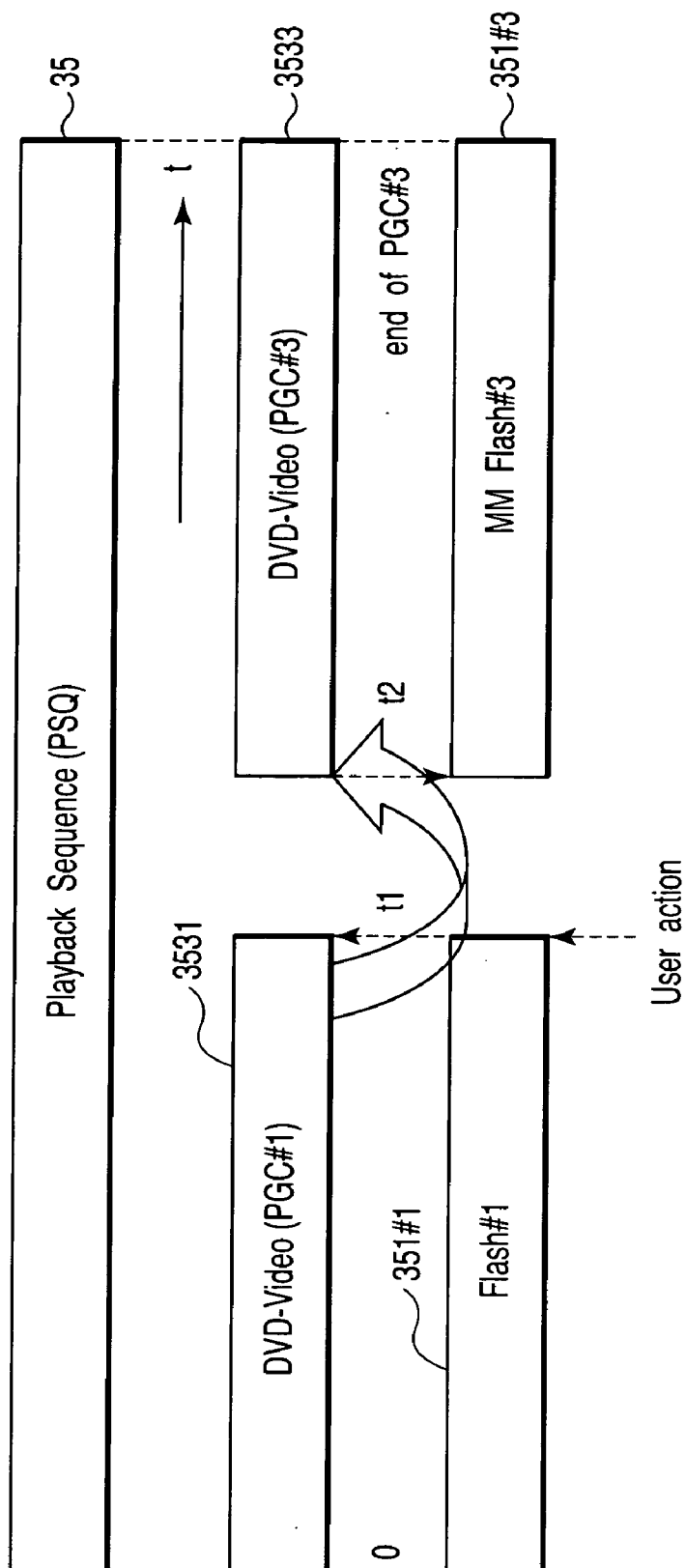


FIG. 15

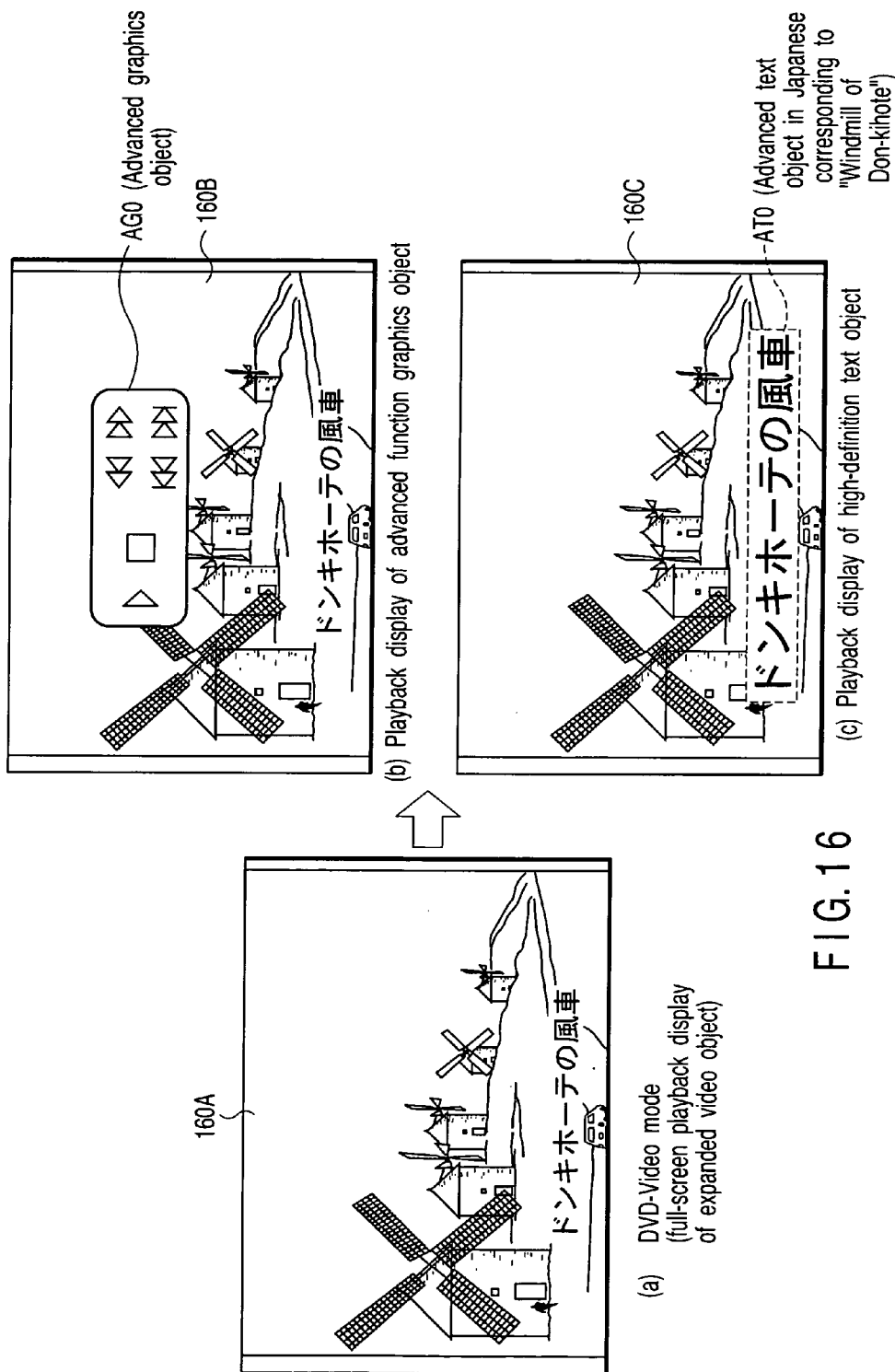
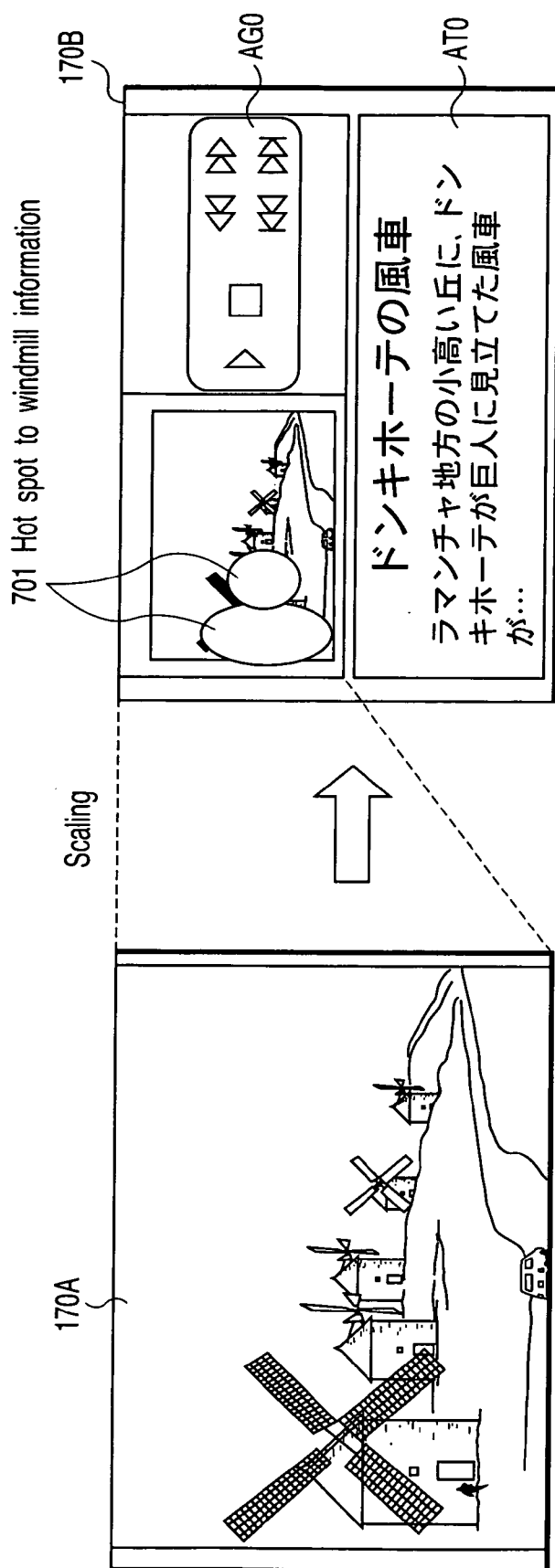


FIG. 16



(a) DVD-Video mode

(b) Playback display of advanced function graphics object, high-definition text object, and the like

FIG. 17

```

<?xml version="1.0"?>
<!DOCTYPE video_pbseq PUBLIC "-//DVD/DTD DVD-VIDEO 2.0//EN" dtd / dvd-video-
2-0-content.dtd">
<video_pbseq>
<vmg>
<vmgm num="1">
<pgc num="1">
<object data="file://dvdrom:/adv_obj/flash1.swf" />~801
// Append Flash object to PGC of VMG menu
</pgc>
</vmgm>
<vmgm num="n">
<object data="http://www.hddvd.com/adv_obj/commentary.ac3" />~802
// Append audio streaming object to VMG menu
</vmgm>
</vmg>
</video_pbseq>
<vts num="1">
<vts_ft num="1">
<pgc num="1">
<object data="http://www.hddvd.com/adv_obj/spot.vck" />~803
// Append Vclick streaming object to PGC of VTS title
</pgc>
</vts_ft>
<vts_ft num="n">
<object data="file://dvdrom:/adv_obj/index.xhtm" />~804
// Append Markup language object to VTS title
<object data="file://dvdrom:/adv_obj/text.xml" />~805
// Append timed text object to VTS title
</vts_ft>
</vts>
</video_pbseq>

```

FIG. 18

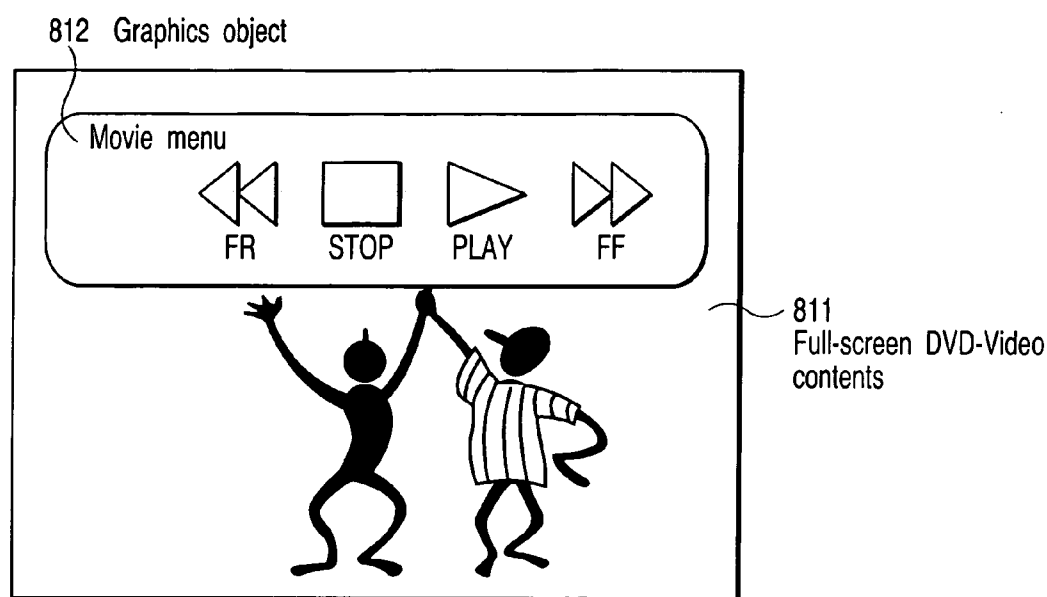


FIG. 19

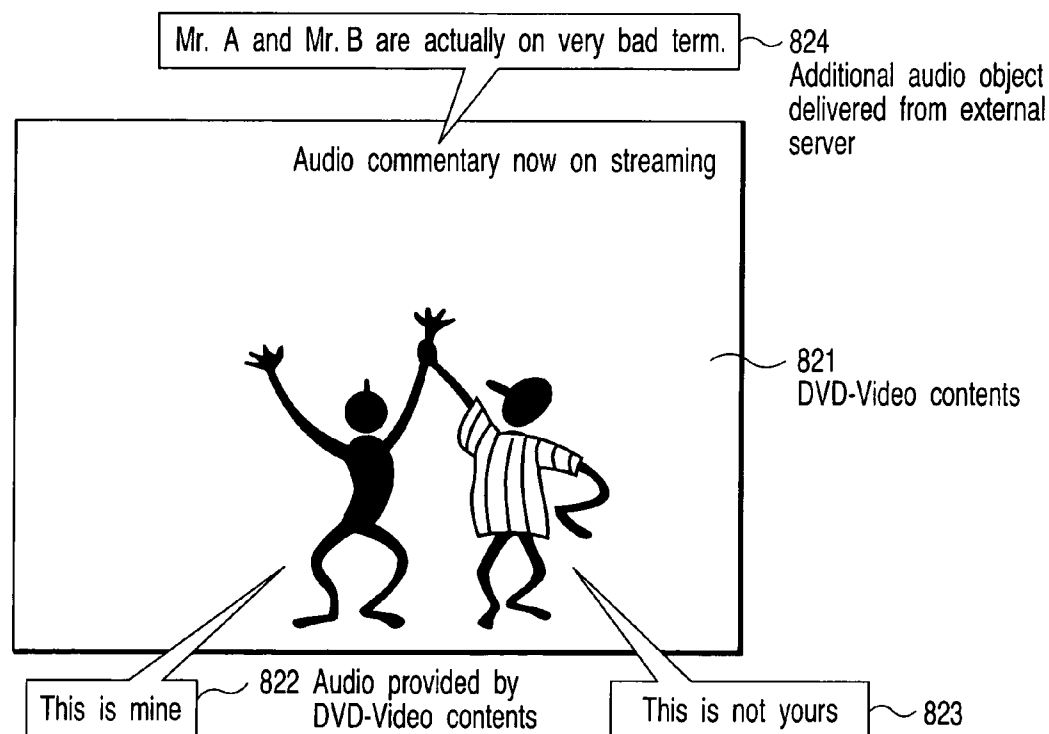


FIG. 20

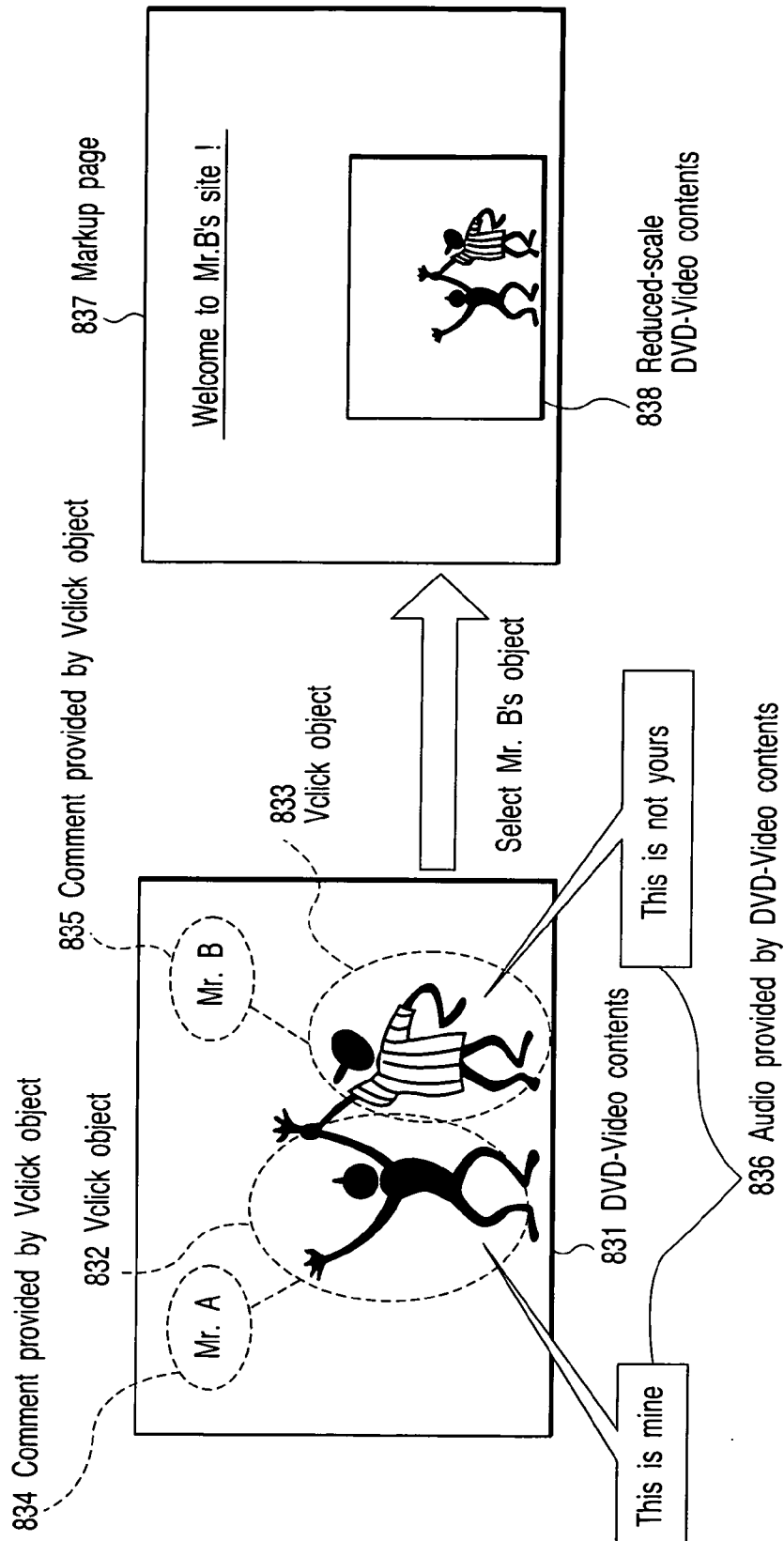


FIG. 21

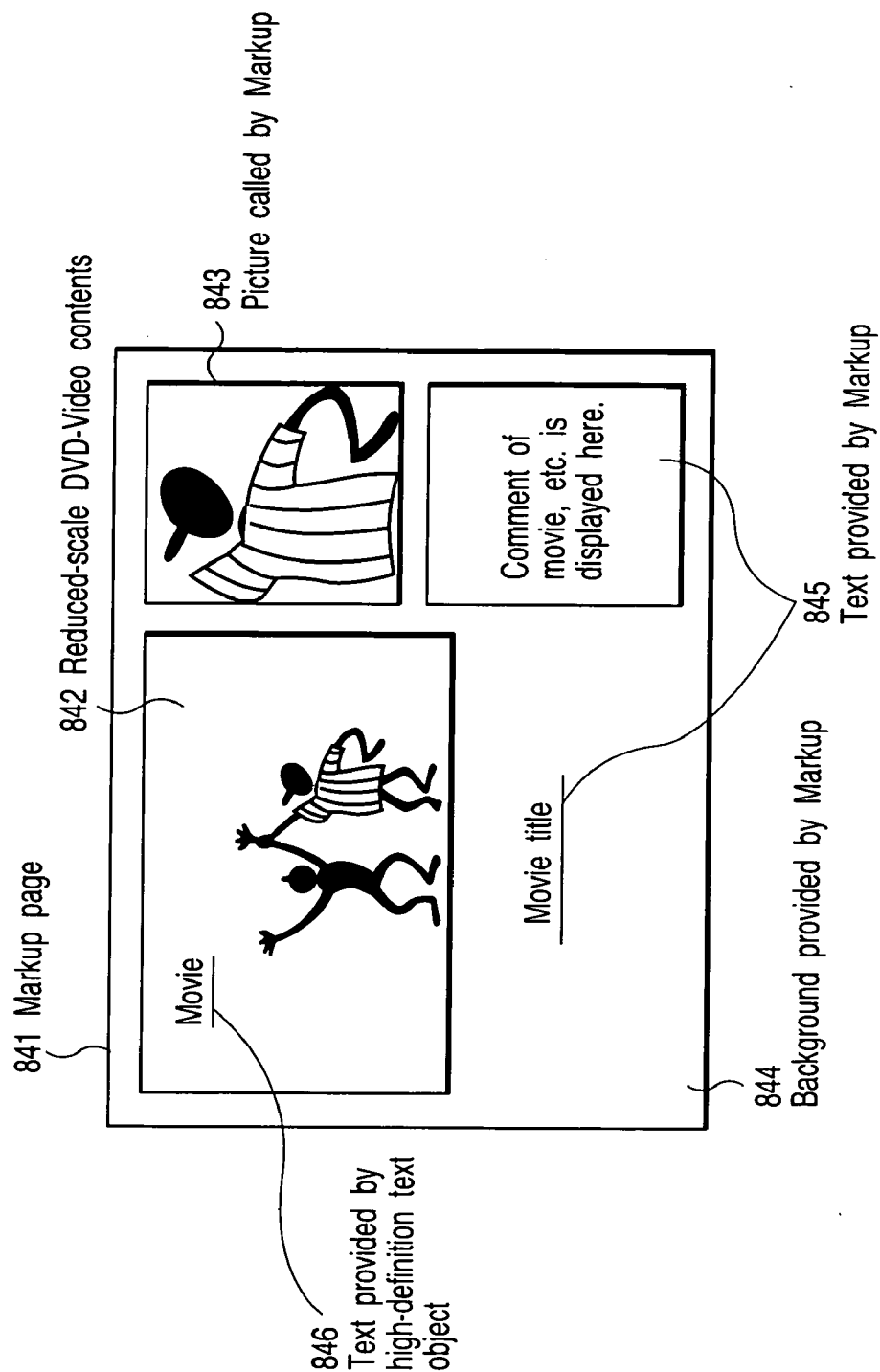


FIG. 22

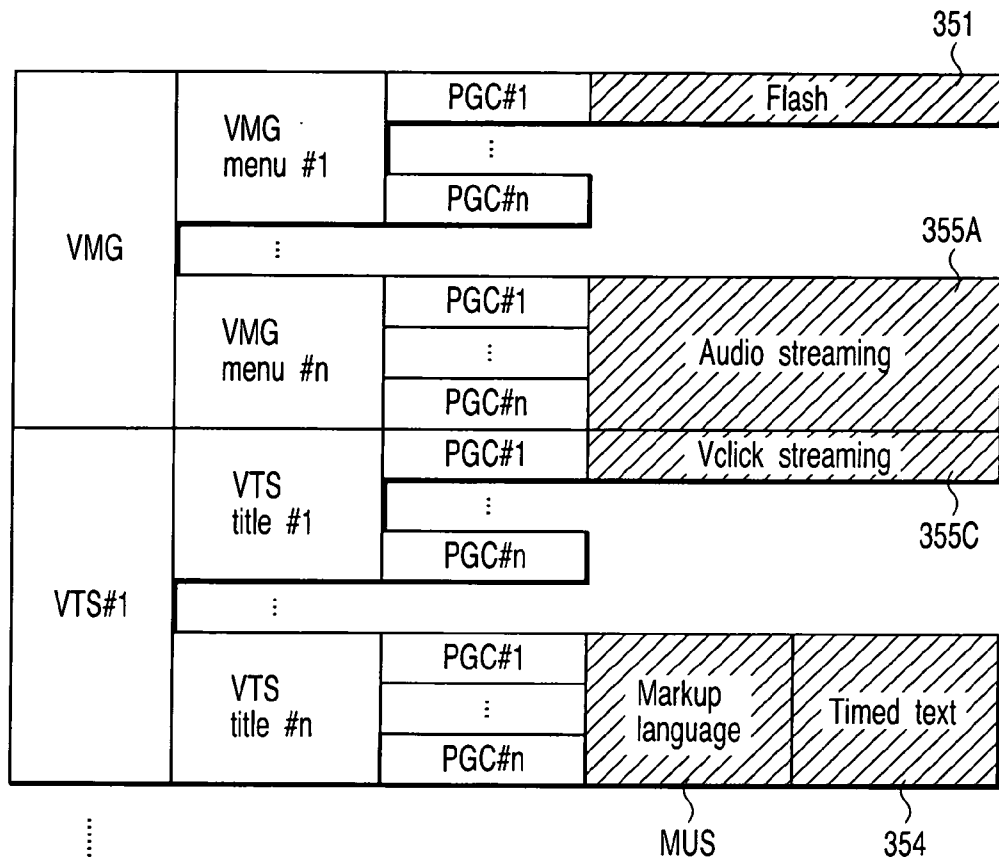


FIG. 23

~851	<pgc num="1"> // Display graphics object <object data="file:///dvdrom:/adv_obj/flash1.swf width="800" height="200" position="20,20" start="00:00:00.00" end="00:05:00.00" priority="1" /> </pgc>
~852	<pgc num="2"> // Display graphics object (PTM description) <object data="file:///dvdrom:/adv_obj/flash1.swf width="800" height="200" position="20,20" start_ptm="0" end_ptm="27000000" priority="1" /> </pgc>
~853	<pgc num="3"> // Display reduced-scale DVD-V/ideo and graphics object <object data="dvd:/hvdvd_ts/hv00001.ifo" width="600" height="400" position="200,400" priority="0" alpha="0" /> <object data="file:///dvdrom:/adv_obj/flash2.swf priority="1" alpha="125" /> </pgc>
~854	<pgc num="4"> // Display graphics object for each audio stream <object data="file:///dvdrom:/adv_obj/flash3.swf audio="1" /> <object data="file:///dvdrom:/adv_obj/flash4.swf audio="2" /> </pgc>
~855	<pgc num="5"> // Display graphics object for each subpicture stream <object data="file:///dvdrom:/adv_obj/flash5.swf subpic="1" /> <object data="file:///dvdrom:/adv_obj/flash6.swf subpic="2" /> <object data="file:///dvdrom:/adv_obj/flash7.swf subpic="3" /> </pgc>
~856	<pgc num="6"> // Display graphics object for each angle <object data="file:///dvdrom:/adv_obj/flash8.swf angle="1" /> <object data="file:///dvdrom:/adv_obj/flash9.swf angle="3" /> </pgc>
~857	<pgc num="7"> // Display graphics object for each display mode <object data="file:///dvdrom:/adv_obj/flash10.swf aspect="16:9" display="wide" /> <object data="file:///dvdrom:/adv_obj/flash11.swf aspect="4:3" display="lb" /> <object data="file:///dvdrom:/adv_obj/flash12.swf aspect="4:3" display="ps" /> </pgc>
~858	<pgc num="8"> // Display graphics object for each display mode <object data="file:///dvdrom:/adv_obj/flash13.swf aspect="4:3" display="normal" /> </pgc>
~859	<pgc num="9"> // Definition of graphics object based on audio / subpicture stream and angle <object data="file:///dvdrom:/adv_obj/flash14.swf audio="1" subpic="1" angle="1" /> <object data="file:///dvdrom:/adv_obj/flash15.swf audio="1" subpic="2" angle="1" /> <object data="file:///dvdrom:/adv_obj/flash16.swf angle="2" /> <object data="file:///dvdrom:/adv_obj/flash17.swf audio="2" subpic="2" /> </pgc>

FIG. 24

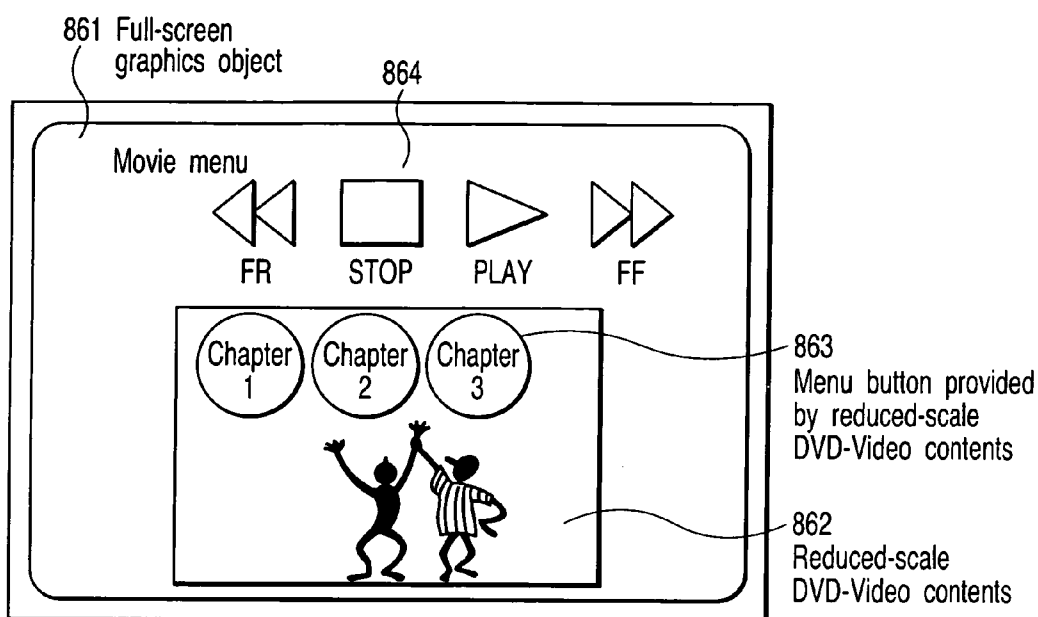


FIG. 25

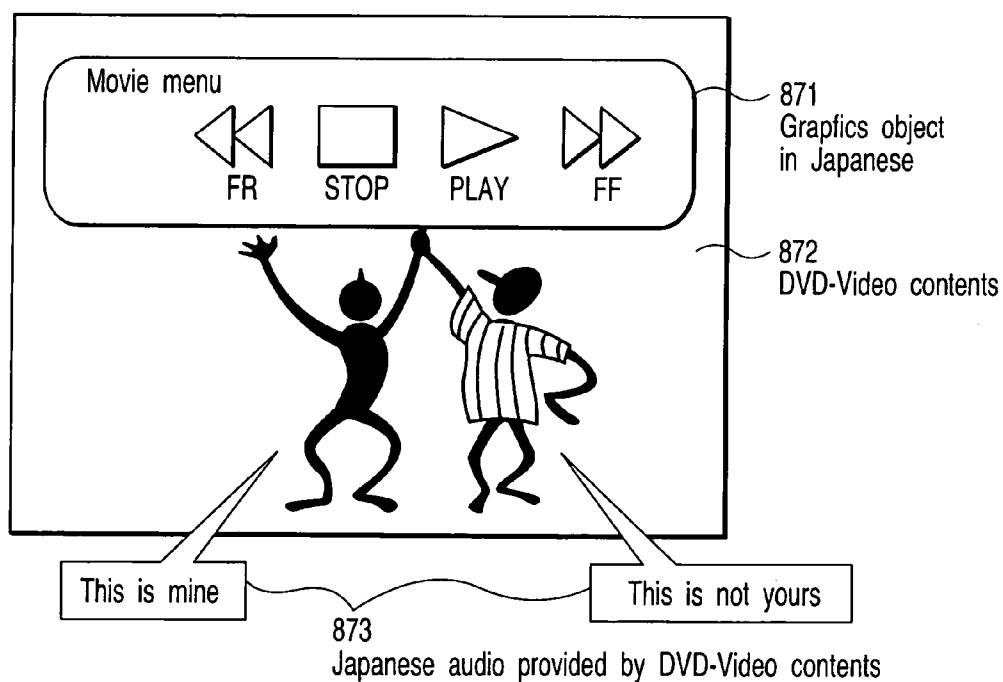


FIG. 26

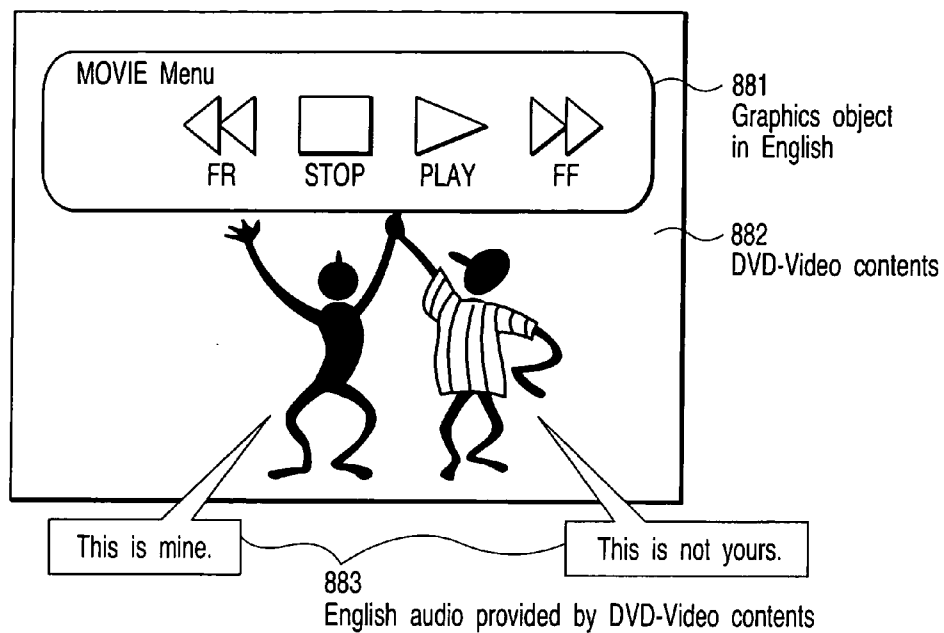


FIG. 27

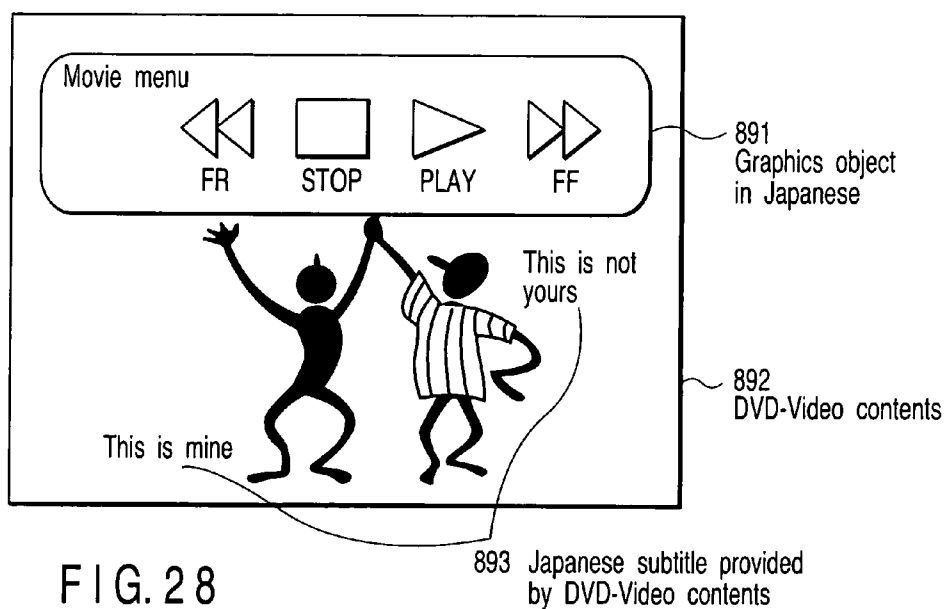


FIG. 28

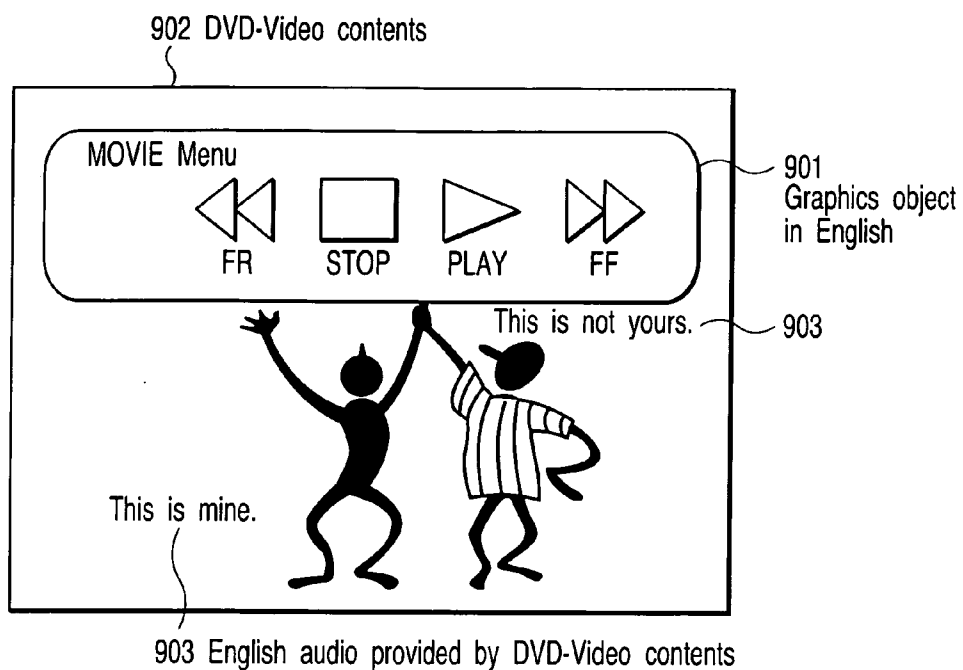


FIG. 29

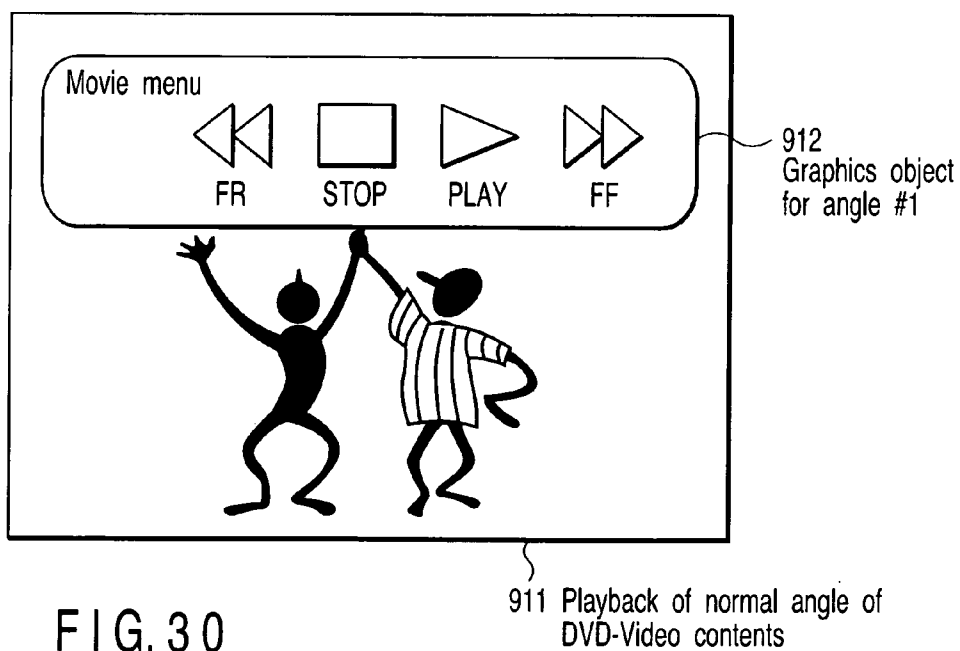


FIG. 30

921 Playback of another angle of DVD-Video contents

922 Graphics object for angle #3



FIG. 31

931 DVD-Video contents at aspect ratio:16:9 and in display mode:wide

932 Graphics object for aspect ratio:16:9 and in display mode:wide



FIG. 32

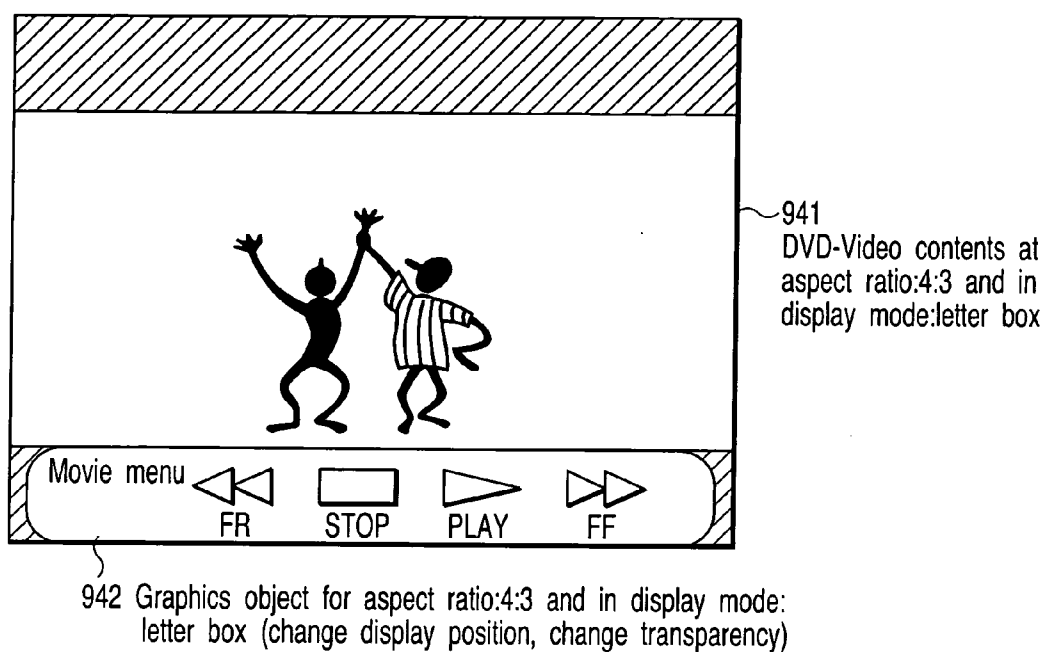


FIG. 33

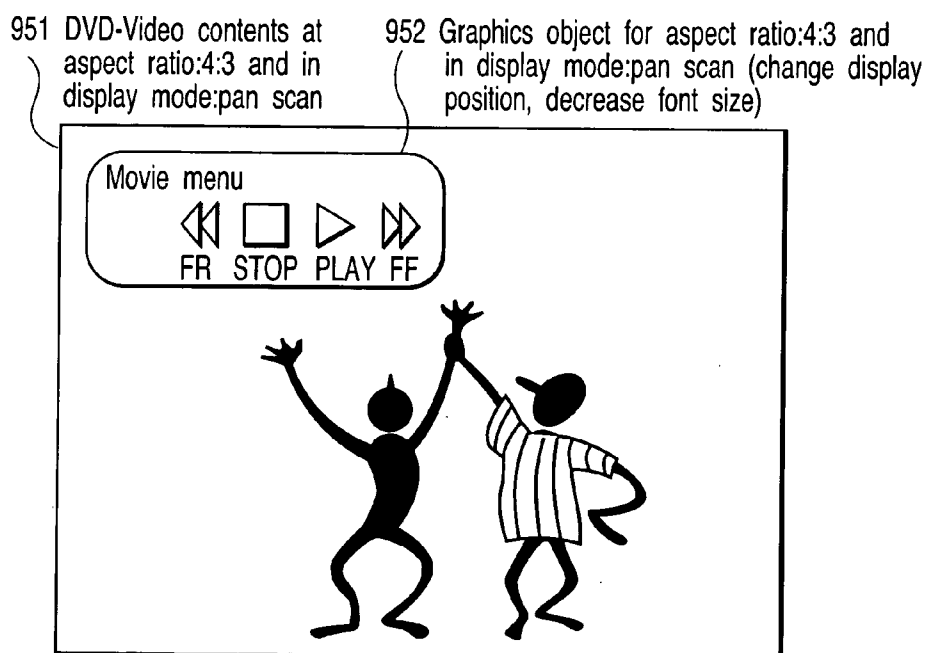


FIG. 34

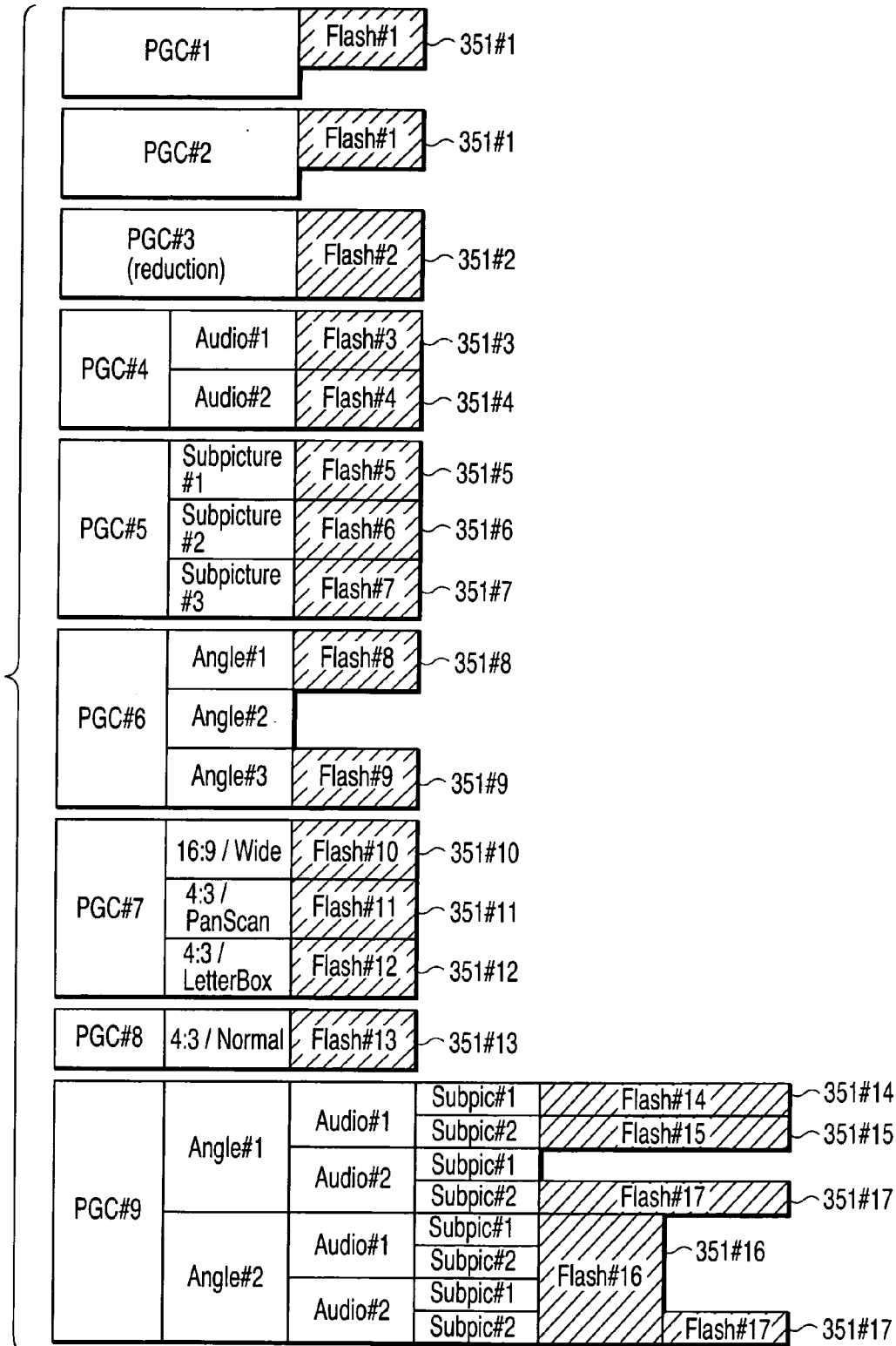


FIG. 35

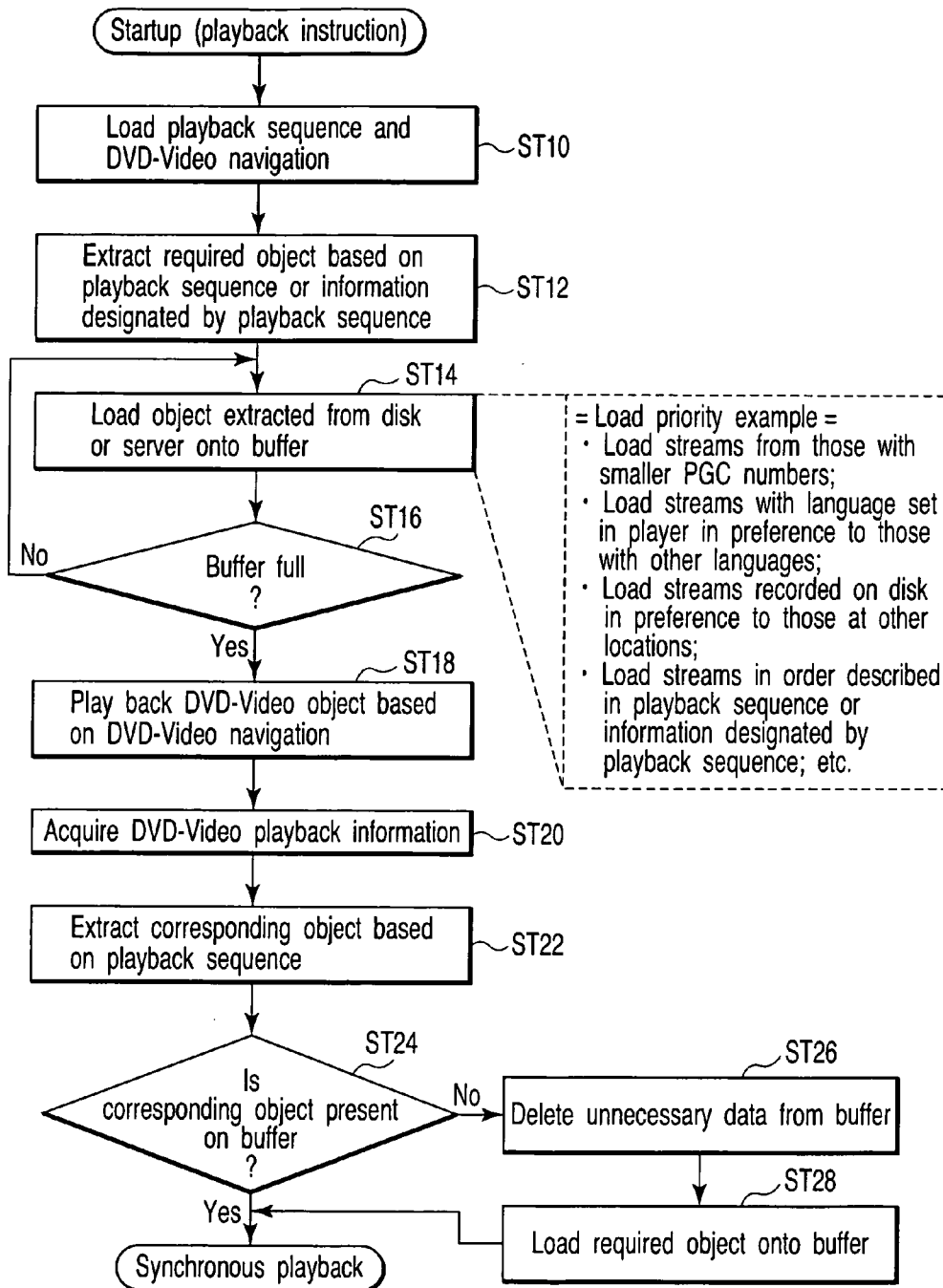


FIG. 36

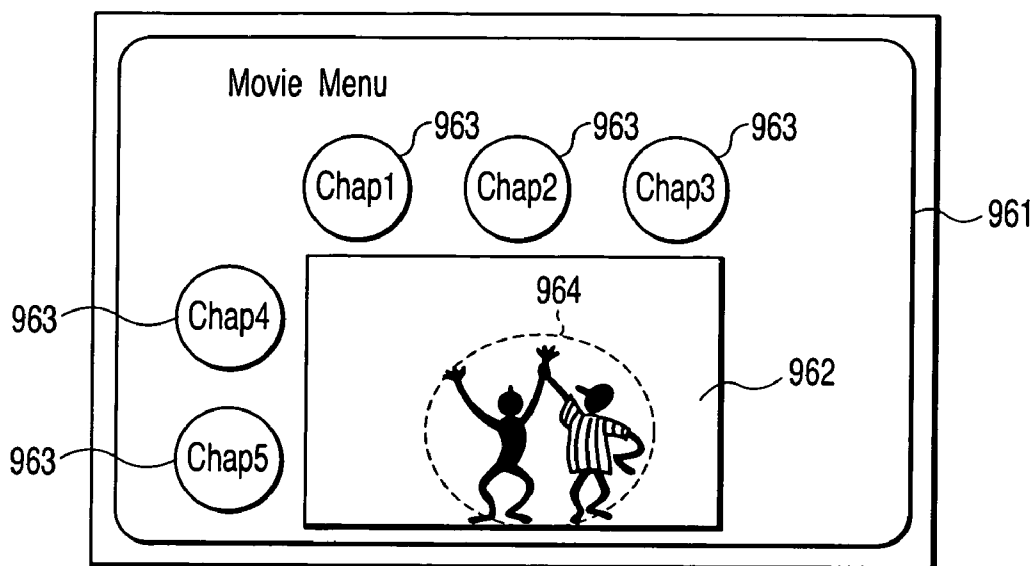


FIG. 37

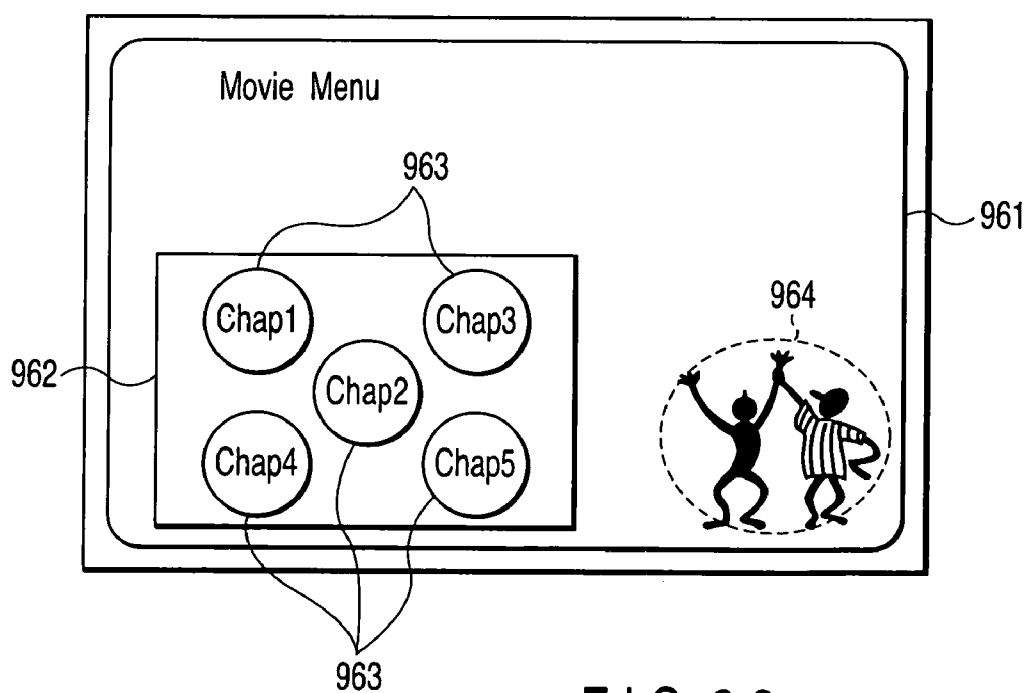


FIG. 38

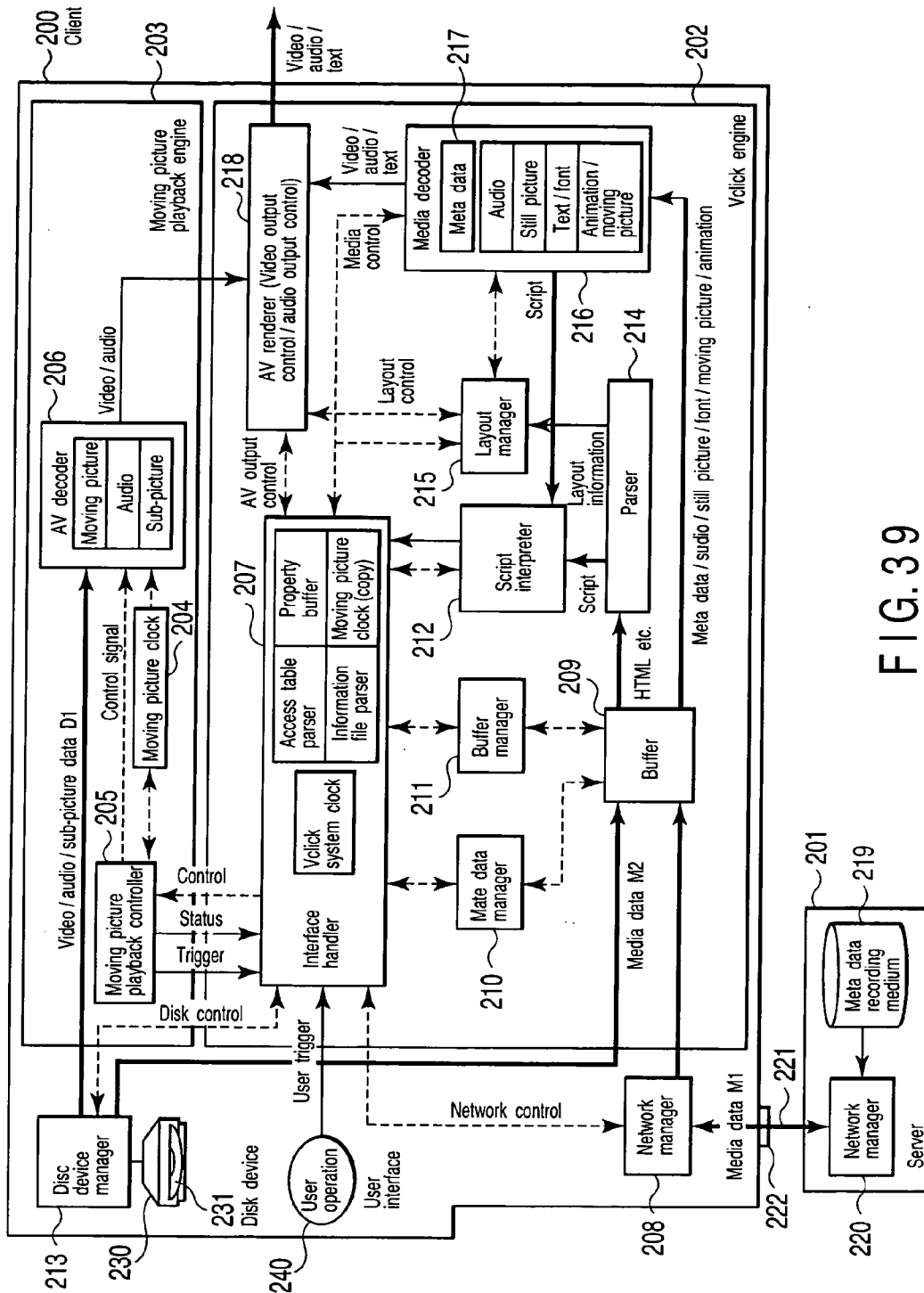


FIG. 39

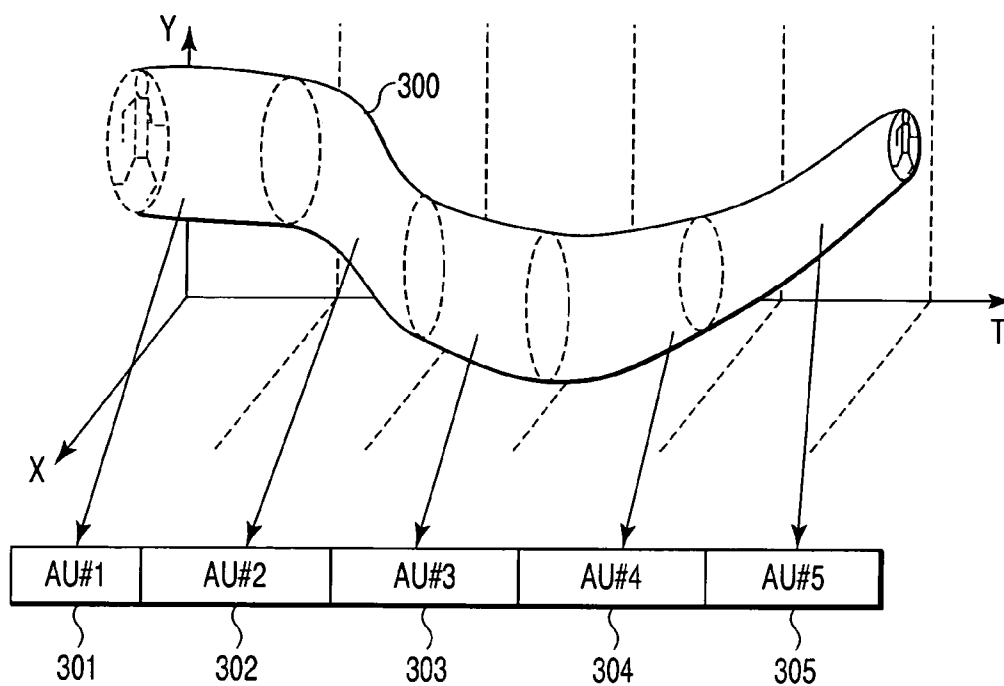


FIG. 40

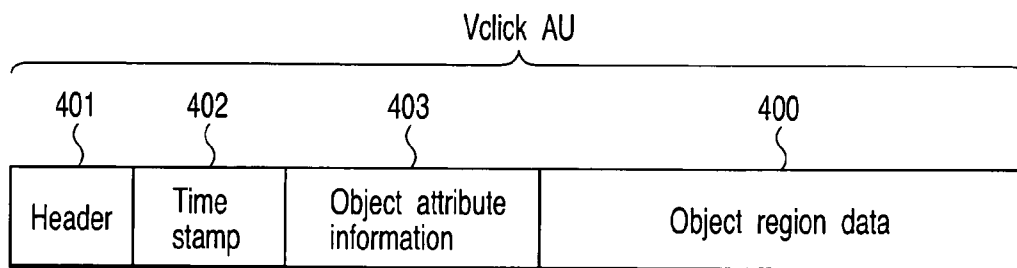


FIG. 41

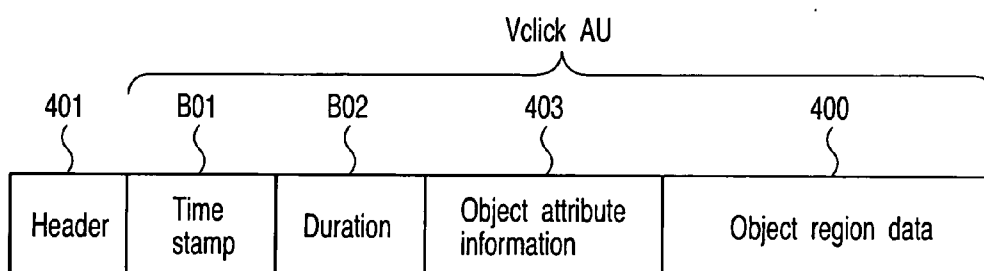


FIG. 42

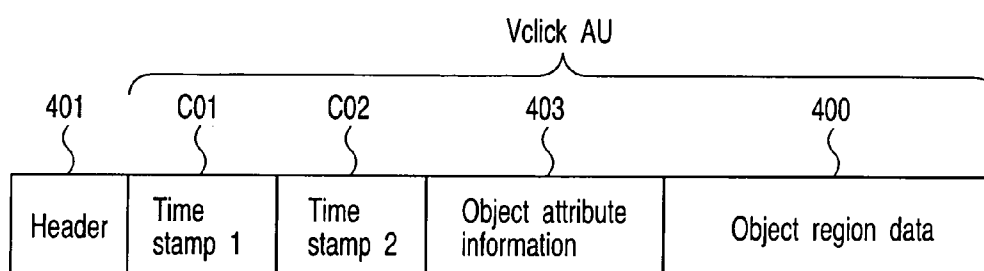


FIG. 43

Time stamp	Access point
time#1	offset#1
time#2	offset#2
time#3	offset#3
time*	NULL
time#4	offset#4
⋮	⋮
time#n	offset#n

FIG. 44

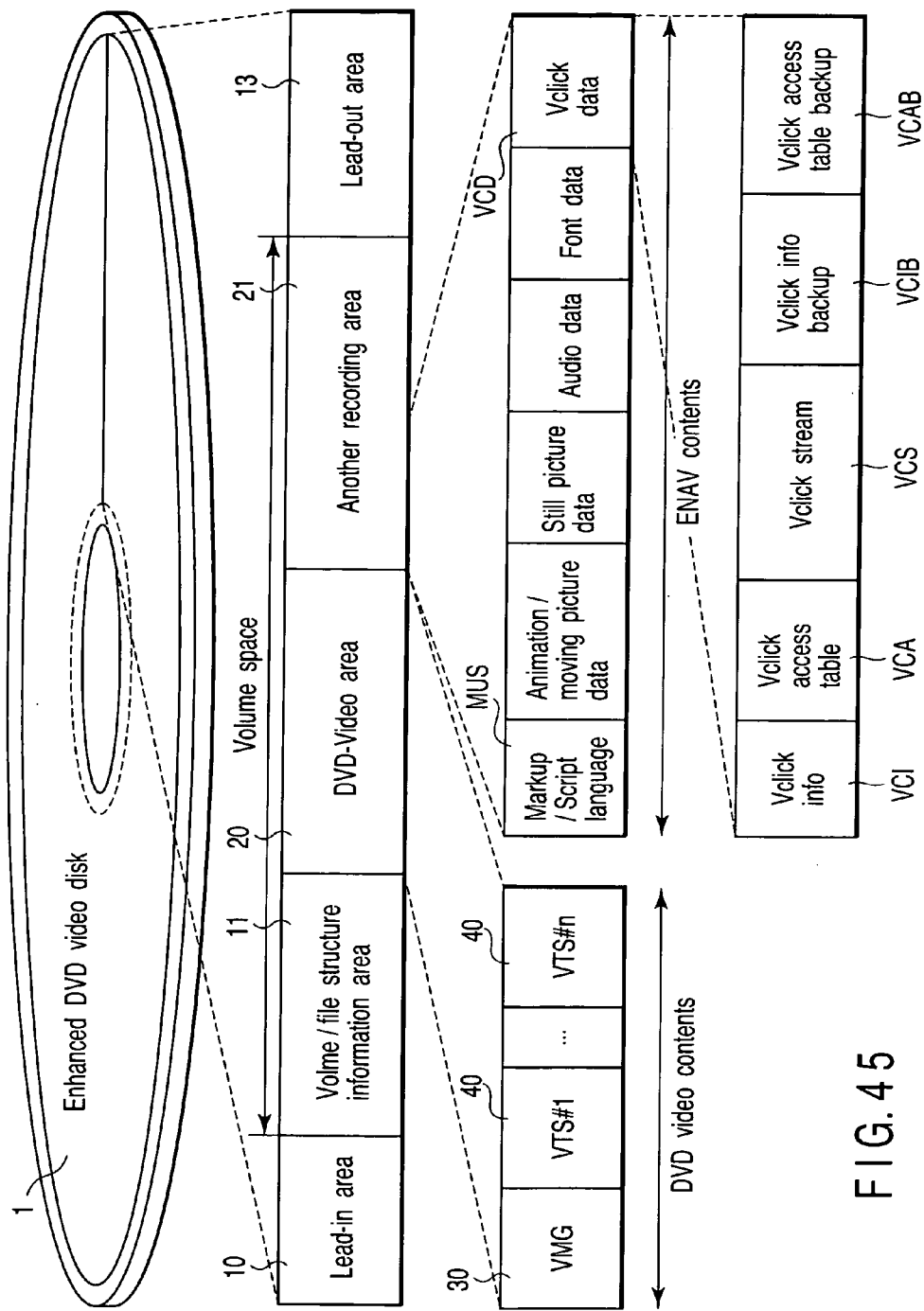


FIG. 45

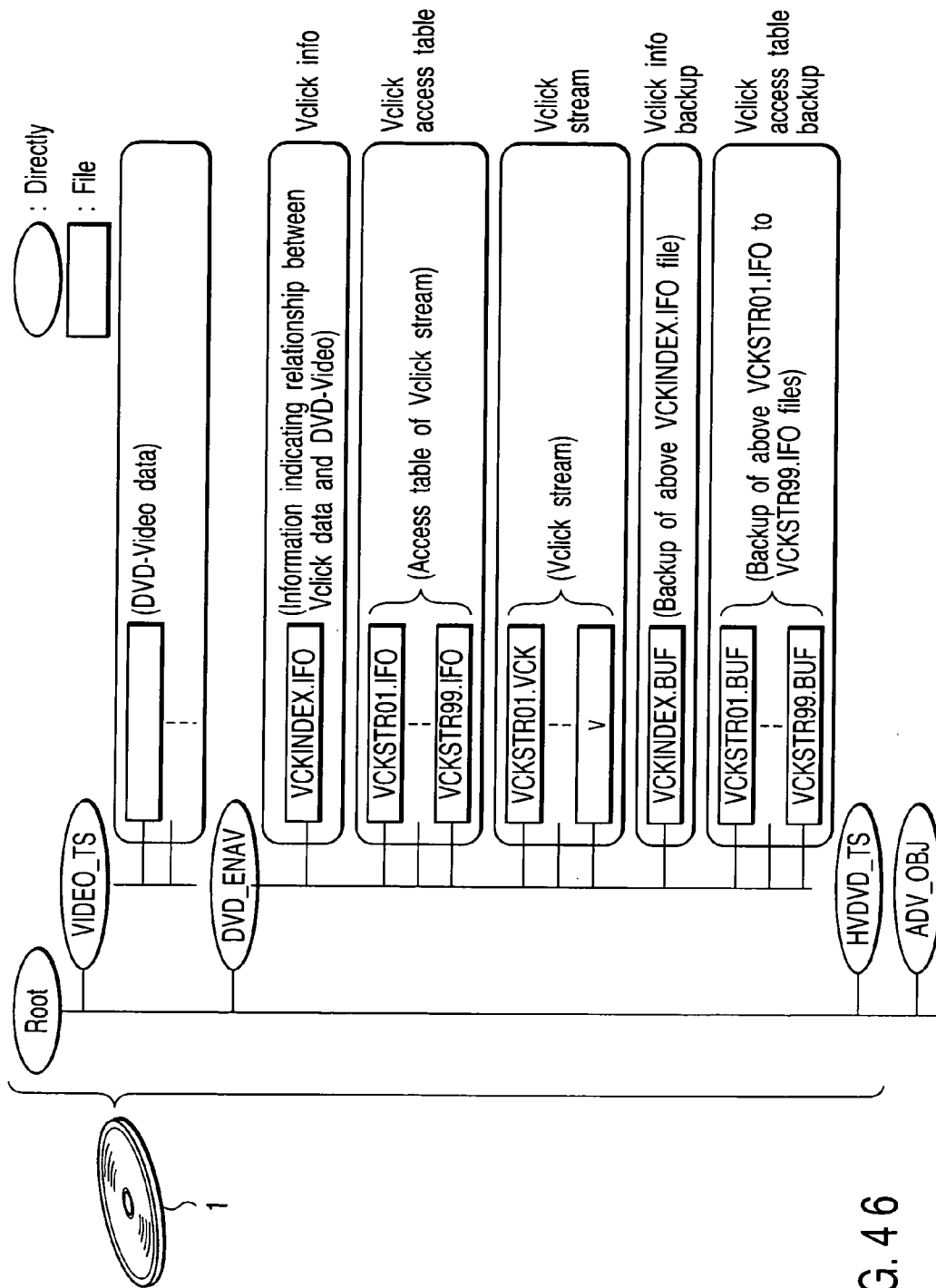


FIG. 46

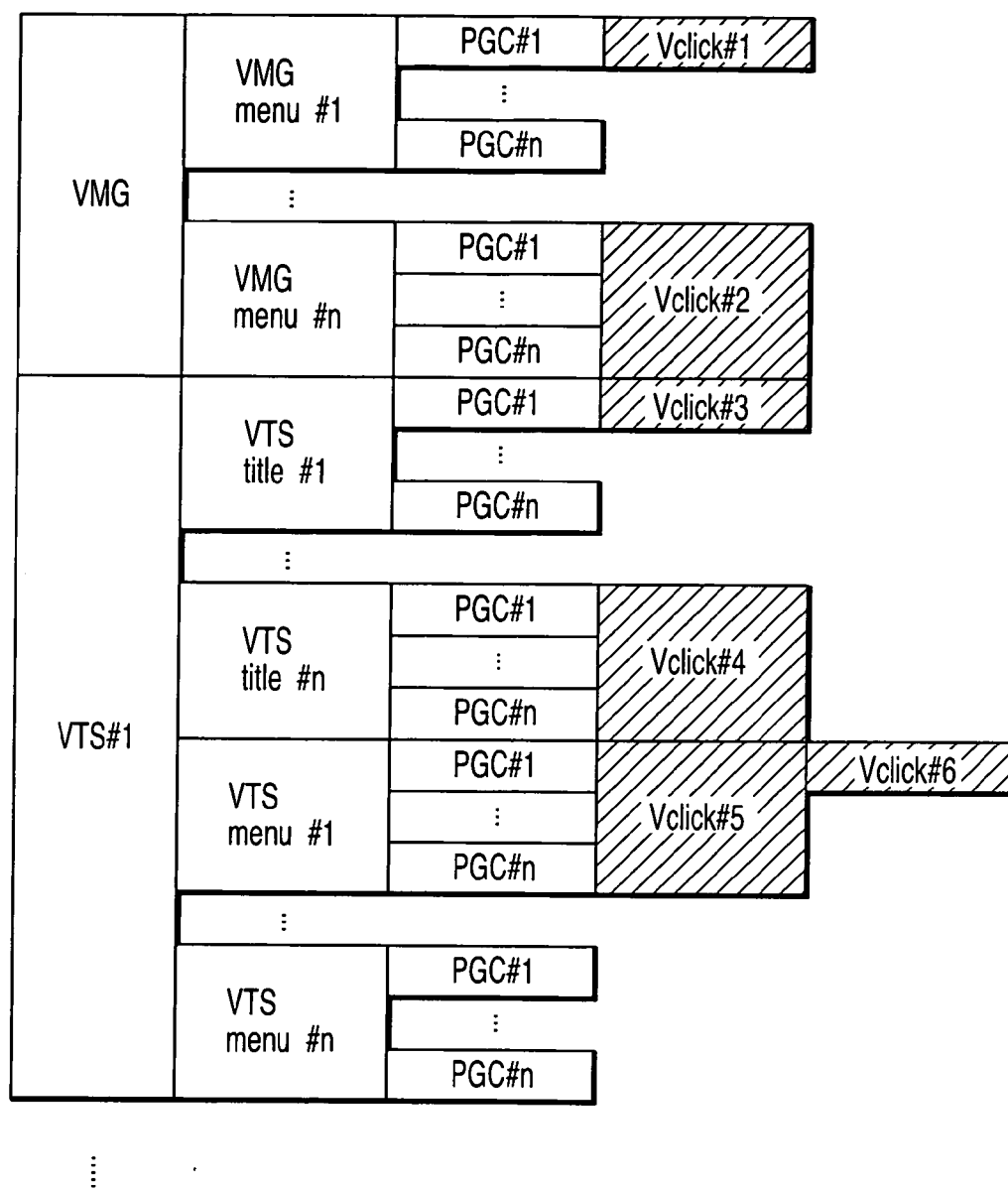


FIG. 47

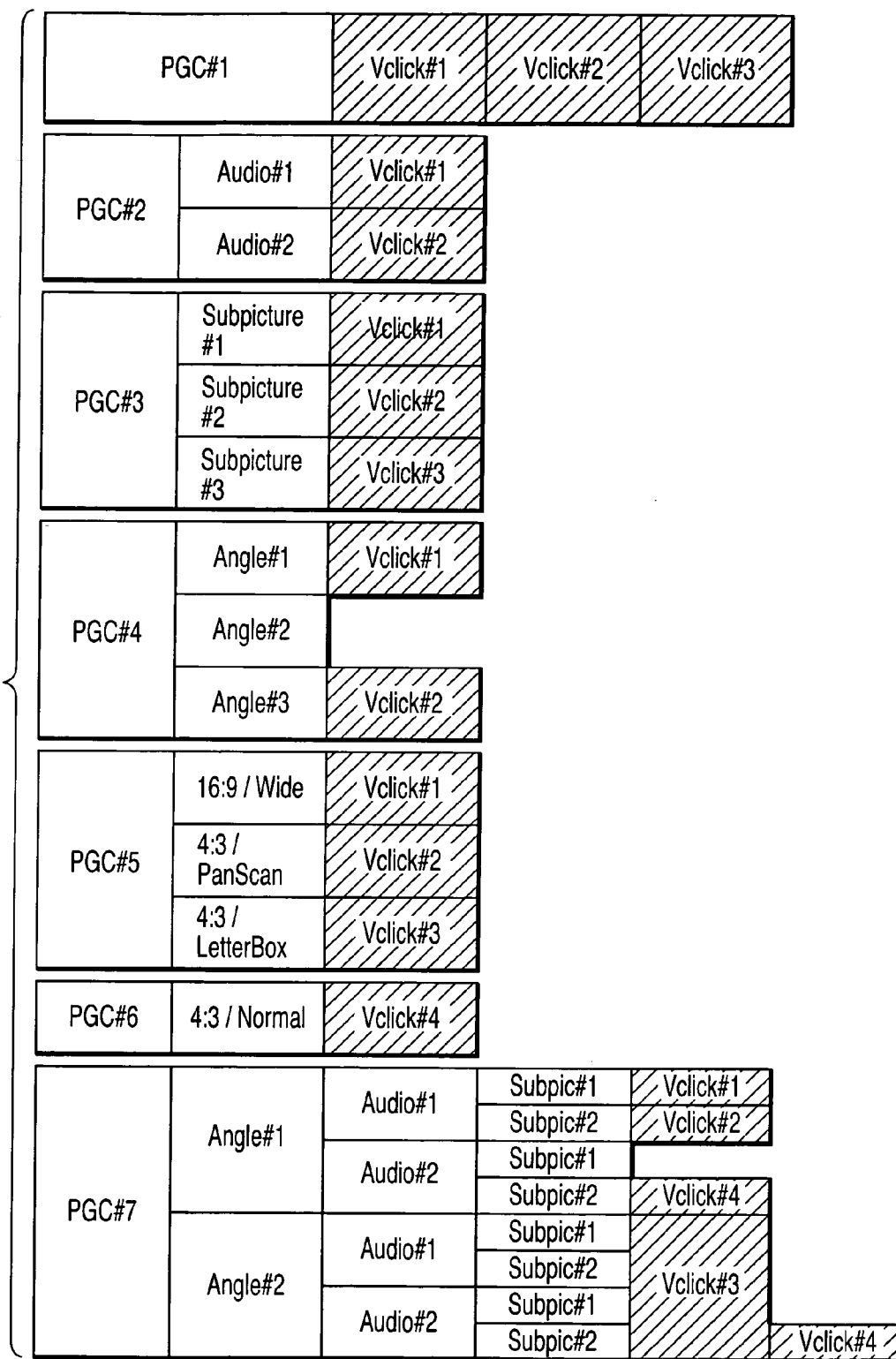


FIG. 48

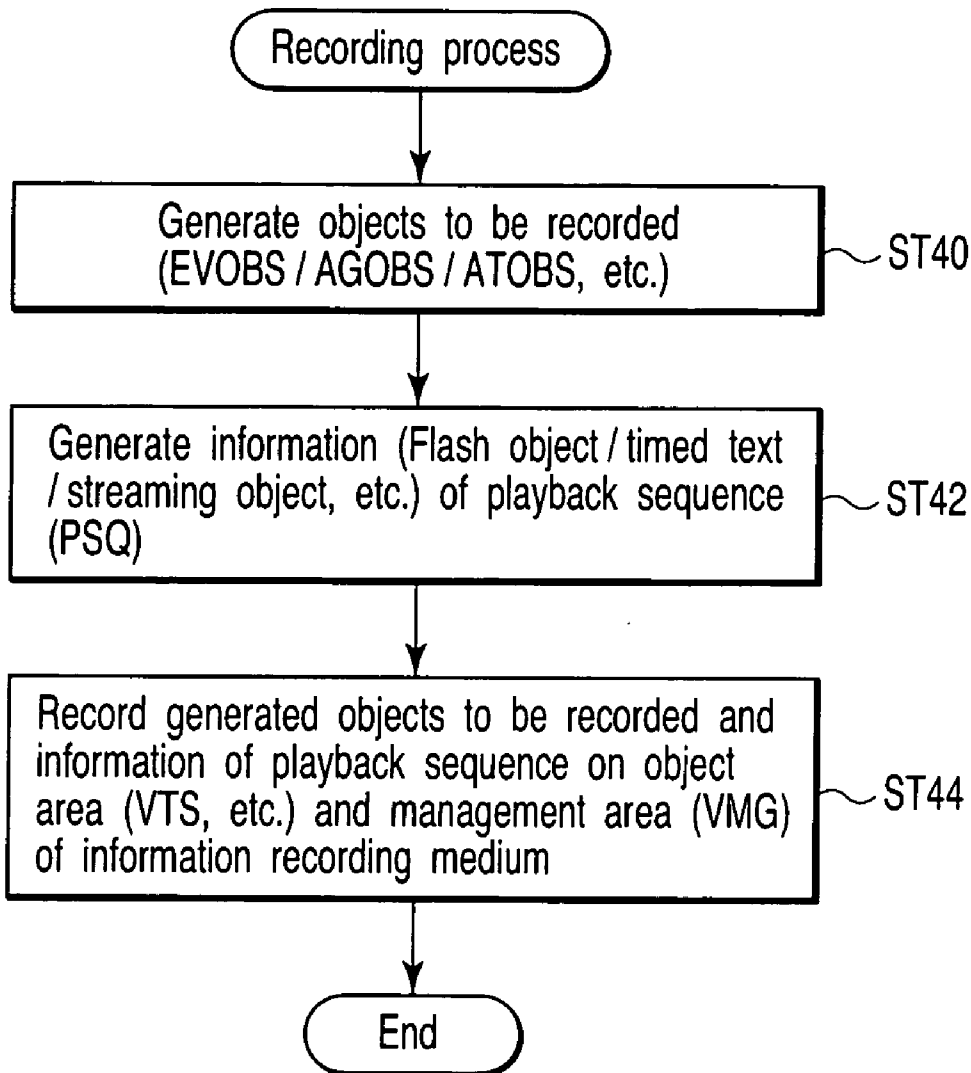


FIG. 49

Tag name	Meaning of tag
video_pbseq	Start of description of playback sequence data
vmg	Corresponding to VMG space of DVD-Video
vmgm	Corresponding to VMG menu domain of DVD-Video
fp	Corresponding to first play domain of DVD-Video
vtg	Corresponding to VTS space of DVD-Video
vtgm	Corresponding to VTS menu domain of DVD-Video
vtg_tt	Corresponding to VTS title domain of DVD-Video
idle	Used when other objects are displayed without displaying any DVD-Video objects
pgc	Corresponding to PGC of DVD-Video (except for idle)
object	Used when object data (DVD-Video object etc.) is called (played back)
version	Version number of playback sequence data
update	URI information of substitute playback sequence data (normally enabled when substitute data is recorded on external server and data has version newer than version number of playback sequence data recorded on disc)
region	Region code of the disc (normally, playback sequence recorded on external server has this tag, and can change region code set on disc itself by overwriting)

FIG. 50

Tag name	Tag that uses attribute	Application	Meaning of attribute
num	vmgm, vts, vtsm, vis_ti, pgc	Condition	Number
lang	vmgm	Condition	Language code (ISO639)
data	pgc	Execution	URI of object data to be played back
start_ptm	data	Condition	Playback start time (relative time using PTM description) of object
end_ptm	data	Condition	Playback end time (relative time in PTM description) of object
start	data	Condition	Playback start time (relative time using HH:MM:SS:FF description) of object
end	data	Condition	Playback end time (relative time using HH:MM:SS:FF description) of object
audio	data	Condition	Audio stream number of DVD-Video
subpic	data	Condition	Sub-picture stream number of DVD-Video
angle	data	Condition	Angle number of DVD-Video
gprm	data	Condition	General parameter value of DVD-Video
sprm	data	Condition	System parameter value of DVD-Video
priority	data	Execution	Priority upon superimpose display or priority of user input process
alpha	data	Execution	Alpha value upon alpha blending
position	data	Execution	x- and y-coordinate positions for displaying object
height	data	Execution	Height after scaling upon scaling object
width (or length)	data	Execution	Length after scaling upon scaling object
style	data	Execution	Display information of object
tmap_data	data	Execution	URI of time map data (time-to-address conversion table) of object
cont	data	Execution	Continuation (without reset) of clocks from previous PGC for object other than DVD-Video object when it is "yes"
update_data	data	Execution	URI information of substitute object data (normally enabled when substitute data is recorded on external server, and new data is determined by comparing version etc. of data recorded on disc)
update_tmap	data	Execution	Execute substitution as substitute time map data when substitute object data described in "update_data" is played back
set_audio	data	Execution	Set audio stream to designated number (mute, i.e., silent when it is zero) when substitute object data described in "update_data" is played back
set_subpic	data	Execution	Set sub-picture stream to designated number (non-display when it is zero) when substitute object data described in "update_data" is played back
set_angle	data	Execution	Set angle to designated number when substitute object data described in "update_data" is played back
set_gprm	data	Execution	Set general parameter to designated value when substitute object data described in "update_data" is played back
set_sprm	data	Execution	Set system parameter to designated value when substitute object data described in "update_data" is played back
program	data	Execution	Normally play back program of designated value for audio streaming object which includes a plurality of programs
dvd_mixlev	data	Execution	Normally designate DVD mixing level for audio streaming object
audio_mixlev	data	Execution	Normally designate mixing level of audio streaming object for audio streaming object
meta_priority	data	Execution	Normally designate one of 1. use mixing level (dvd/audio_mixlev) designated by playback sequence, 2. use mixing level included in audio streaming object, and 3. use these two mixing levels together, for audio streaming object
chromakey	data	Execution	Normally designate one chromakey value for Flash object (color of designated value is not displayed as original color, and color of another superposed object is displayed, i.e., designated color becomes transparent)
scaling_position	data	Execution	Change object to designated position when display mode switch instruction is issued
scaling_height	data	Execution	Change object to designated height when display mode switch instruction is issued
scaling_width	data	Execution	Change object to designated width when display mode switch instruction is issued

FIG. 51

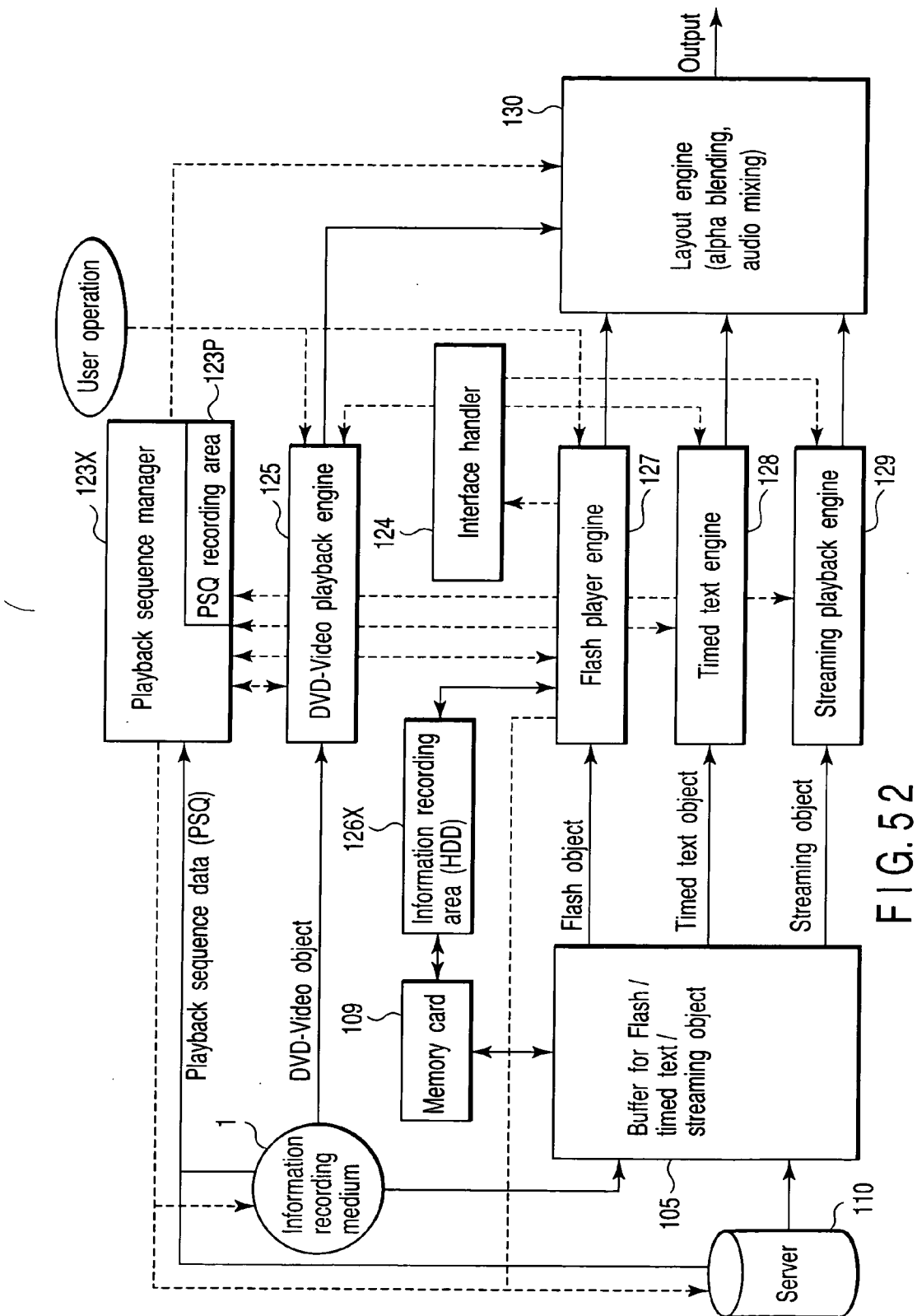


FIG. 52

```

<?xml version="1.0"?>
<!DOCTYPE video_pbseq PUBLIC "-//DVD/DTD DVD-VIDEO 2.0//EN" "dtd/dvd-video-2-0-
  content.dtd">
<version> 1.0 </version>
<update data="http://xx.xx/pbseq-1-2.xml" /> ~ 964
<video_pbseq>
  <vts num="1">
    <vts_tt num="1">
      <object data="file://dvdrom:/menu.swf"tmap="file://dvdrom:/tmap01.ifo"
        update_data="http://xx.xx/new_menu.swf" update_tmap="http://xx.xx/tmap02.ifo" /> ~ 965
      // Append latest menu object and time map
    </vts_tt>
    <vts_tt num="2">
      <object data="file://dvdrom:/eng_menu.swf"param="0" tmap="file://dvdrom:/ ~ 966
        tmap03.ifo" />
      // Append English menu object and time map on disc
    <object data="http://xx.xx/jp_menu.swf"param="1" tmap="http://xx.xx/tmap04.ifo" /> ~ 967
      // Append Japanese menu object and time map on external server
    </vts_tt>
  </vts>
</video_pbseq>

```

FIG. 53

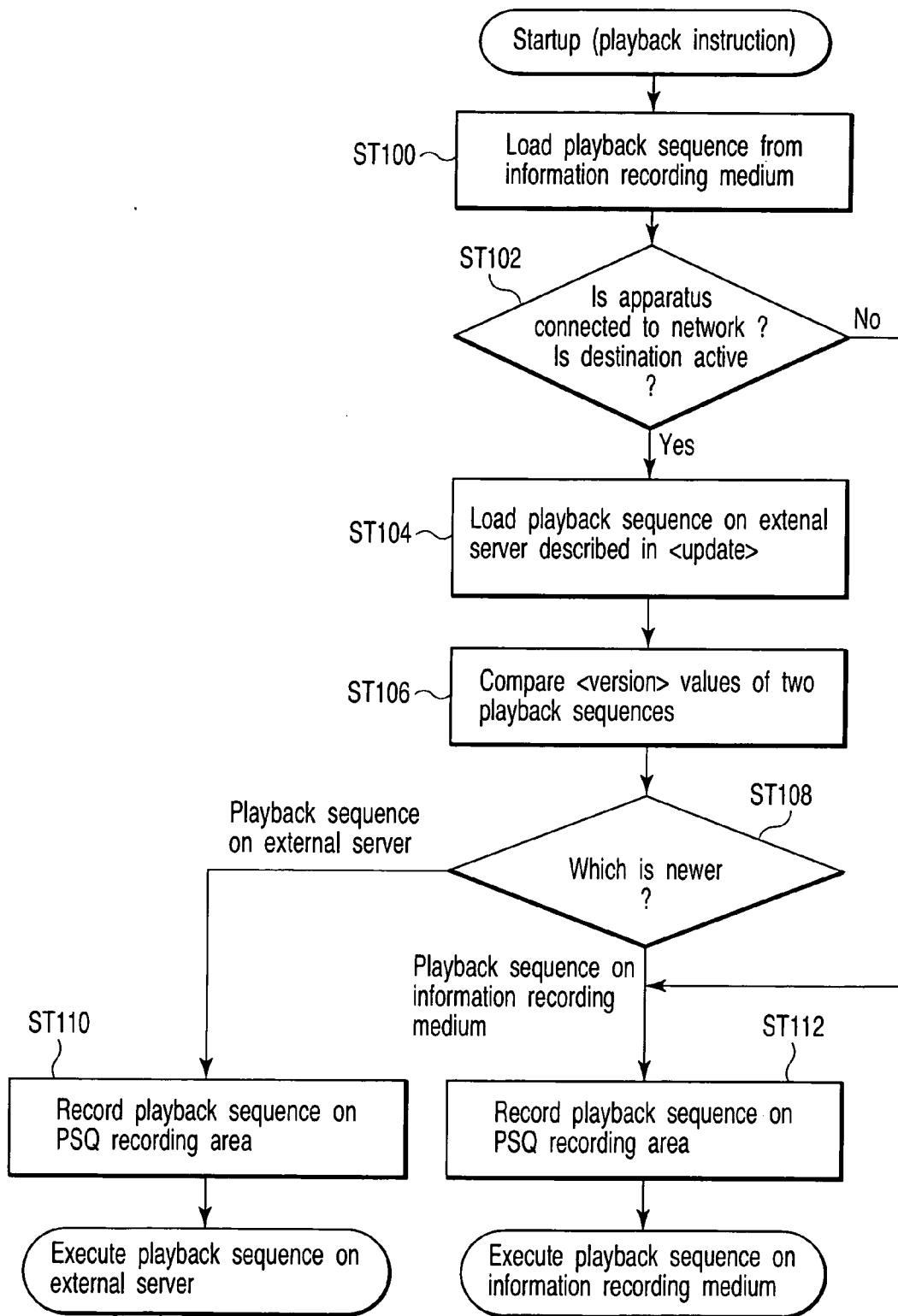
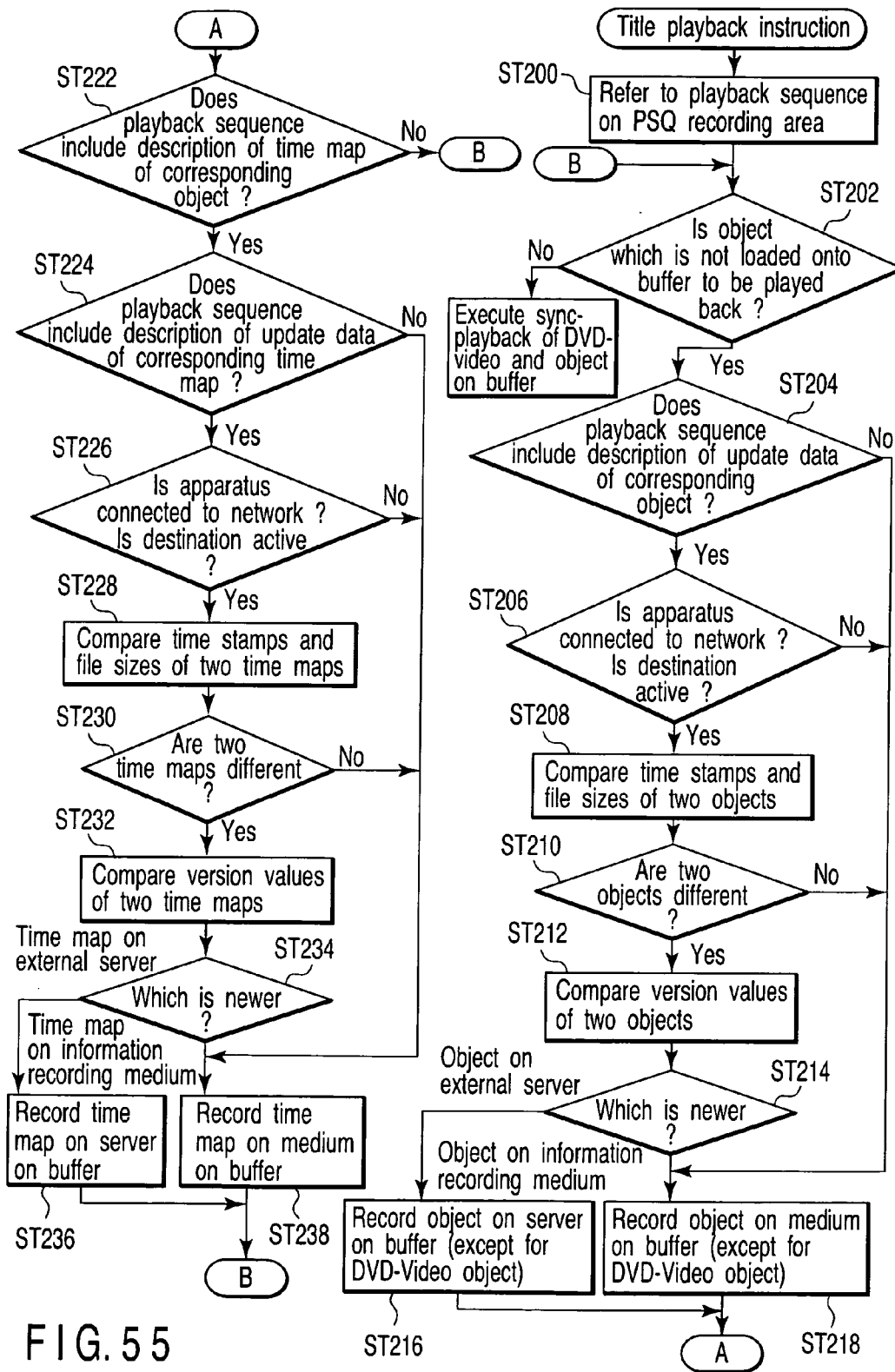


FIG. 54



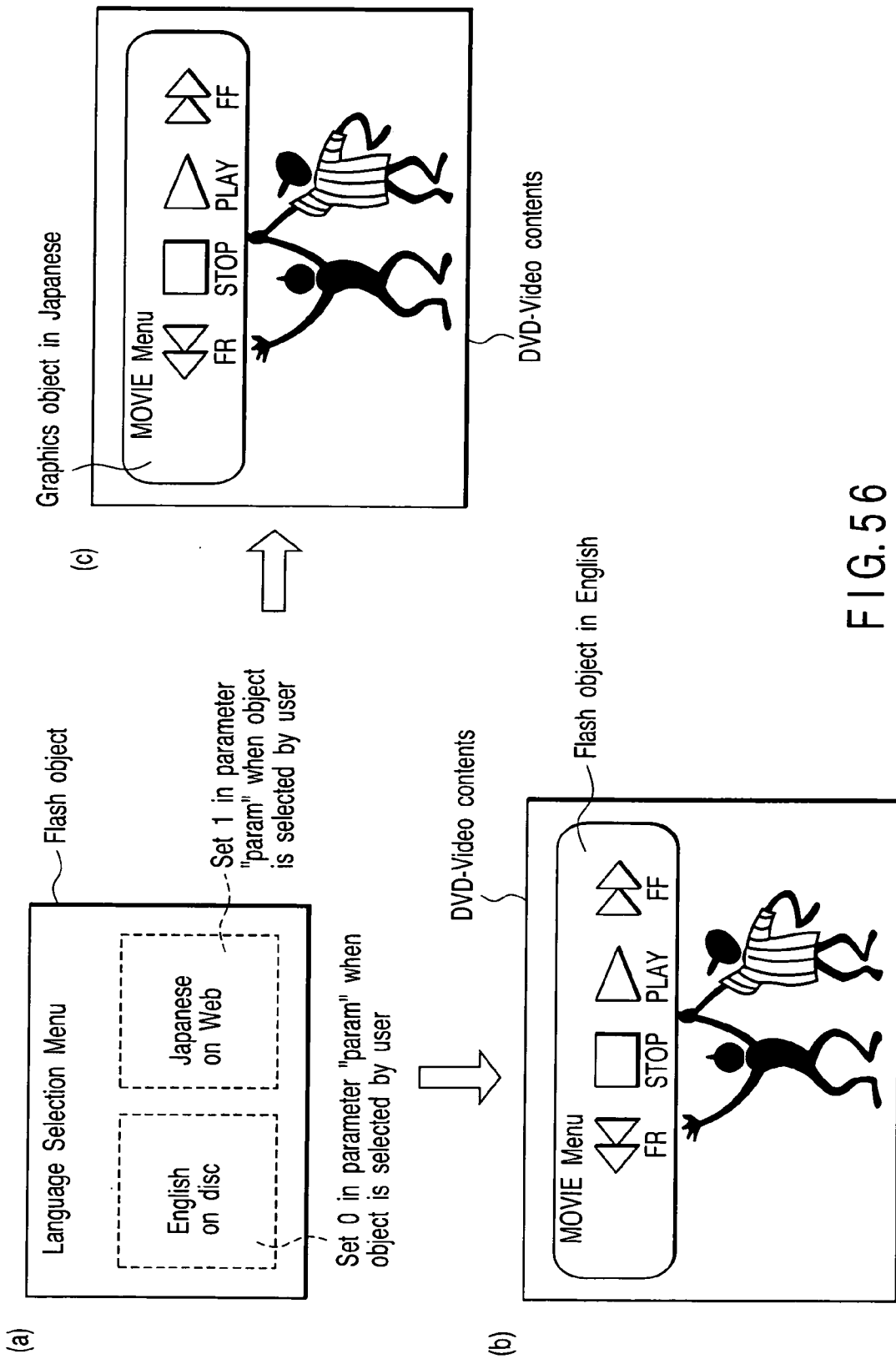


FIG. 56

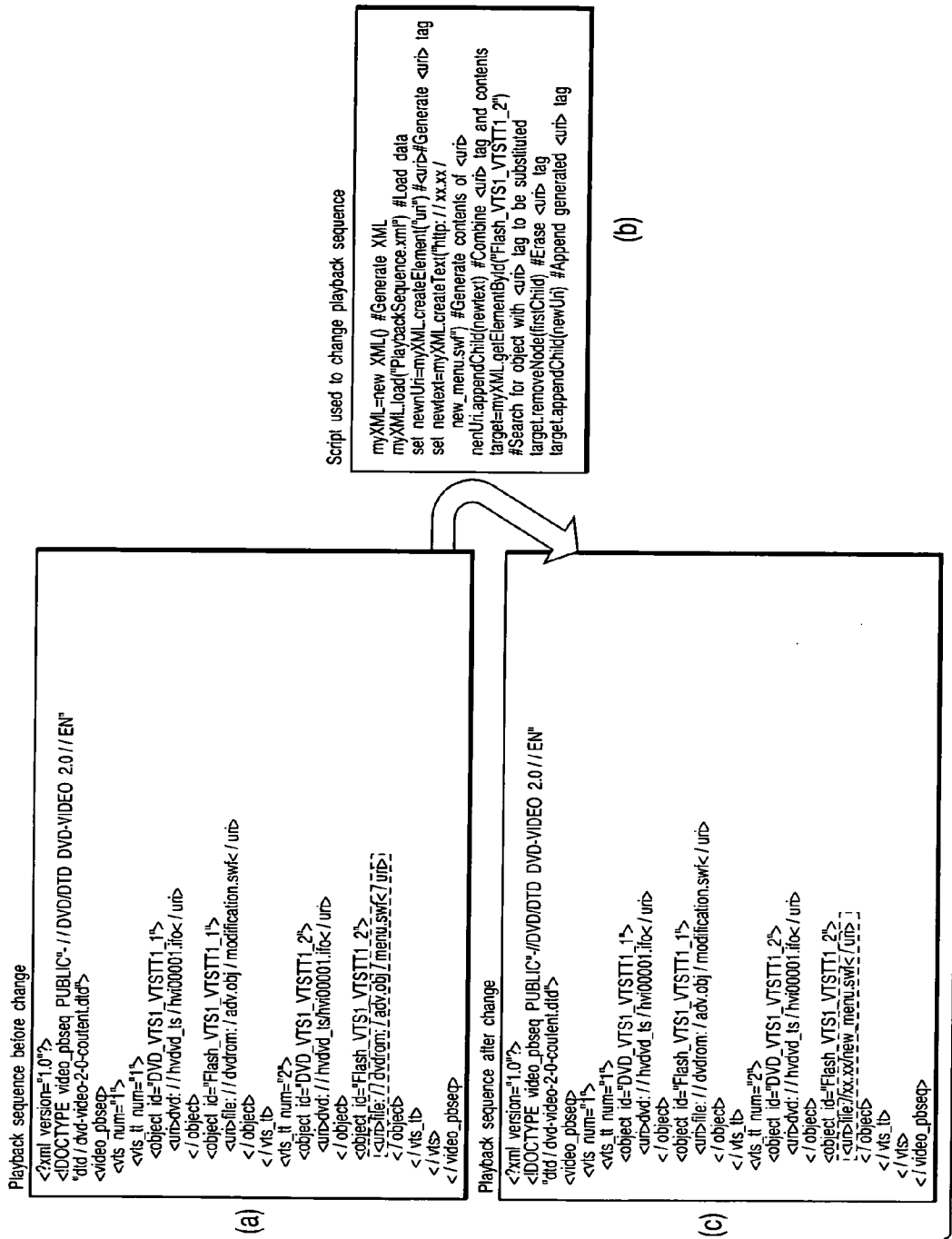


FIG. 57

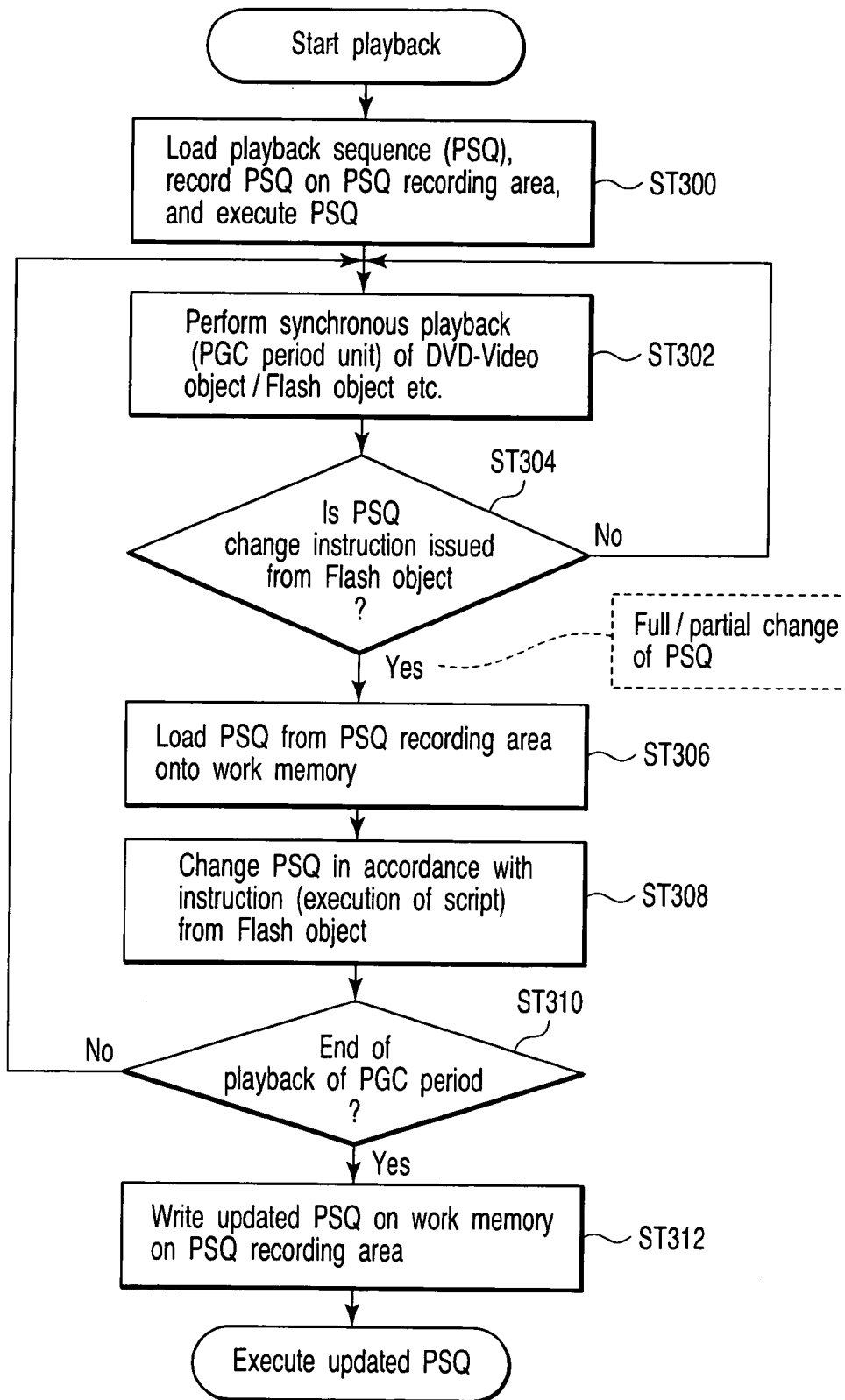


FIG. 58

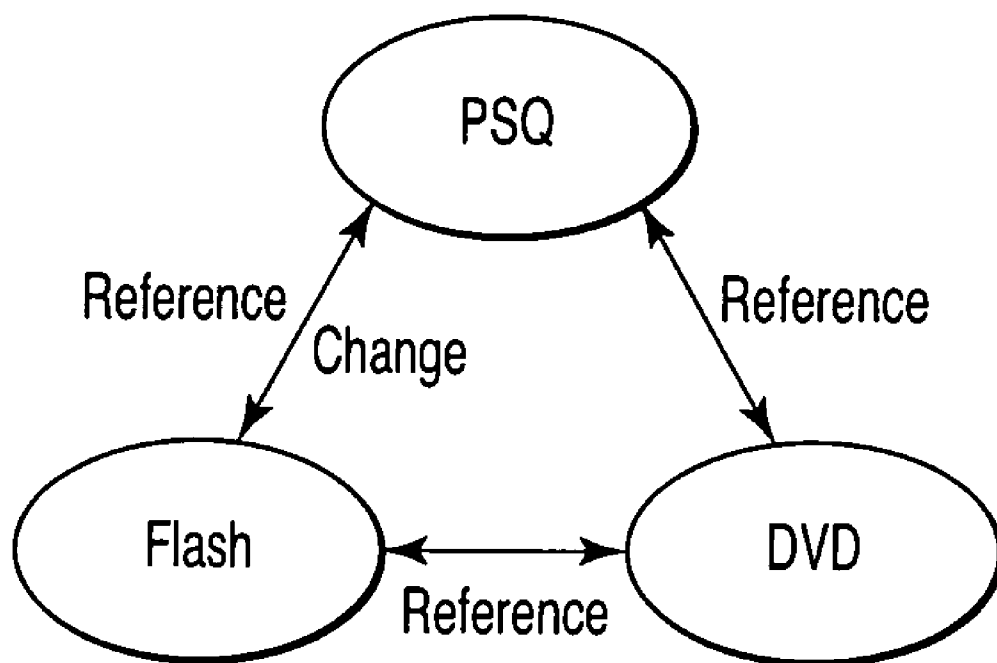


FIG. 59

INFORMATION RECORDING MEDIUM, INFORMATION PLAYBACK METHOD, AND INFORMATION PLAYBACK APPARATUS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This is a Continuation Application of PCT Application No. PCT/JP2005/011097, filed Jun. 10, 2005, which was published under PCT Article 21(2) in English.

[0002] This application is based upon and claims the benefit of priority from prior Japanese Patent Application No. 2004-177730, filed Jun. 16, 2004, the entire contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0003] 1. Field of the Invention

[0004] The present invention relates to an information recording medium such as an optical disc or the like and an information playback apparatus for playing back this information recording medium.

[0005] 2. Description of the Related Art

[0006] In recent years, DVD-Video discs having high image quality and advanced functions, and video players that play back such discs have prevailed, and the range of choice for peripheral devices and the like that can be used to play back such multi-channel audio has broadened. An environment that can personally implement a home theater and allows users to freely enjoy movies, animations, and the like with high image quality and high sound quality at home has become available. As described in Jpn. Pat. Appln. KOKAI Publication No. 10-50036 (p. 18 to 20, FIGS. 50 to 57), a playback apparatus which can superimpose various menus by changing, e.g., text colors and the like for playback video pictures from a disc has been proposed.

[0007] However, in recent years, along with the improvement of the image compression technique, a demand has arisen for realization of higher image quality from both the users and contents providers. In addition to realization of higher image quality, the contents providers require an environment that can provide more attractive contents to users by upgrading and expanding the contents (e.g., more colorful menus, improvement of interactiveness, and the like) in contents such as menu windows, bonus video pictures, and the like as well as a title itself. Furthermore, some users require to freely enjoy contents by playing back still picture data sensed by the user, subtitle text data acquired via Internet connection, and the like by freely designating their playback positions, playback regions, or playback times.

BRIEF SUMMARY OF THE INVENTION

Problems to be Solved by Invention

[0008] As described above, an environment that can provide more attractive contents to users by upgrading and expanding the contents (e.g., more colorful menus, improvement of interactiveness, and the like) in contents such as menu windows, bonus video pictures, and the like is required.

[0009] The present invention has been made in consideration of such situation, and has as its one object to provide an information recording medium and its playback apparatus, which can implement colorful expressions which display buttons with still pictures or small animations at arbitrary positions and arbitrary sizes on the screen together with background audio playback, and highlight such buttons, and can form attractive contents.

Means for Solving Problems

[0010] An information recording medium (1) according to an embodiment of the present invention has a data area (12) including a management area (30) for recording management information and an object area (40) for recording objects to be managed using this management information. In this information recording medium, the object area (40) is configured to store expanded video objects (EVOBS) which undergo playback management using logical units called program chains, and advanced objects (AGOBS/ATOBS) recorded independently of the expanded video objects. The management area (30) is configured to store a playback sequence (PSQ) that gives playback conditions (playback timings, picture output positions, display sizes, and the like) of the advanced objects. Note that the playback conditions can be described by providers and the like using a predetermined language (markup language or the like).

[0011] The playback sequence is configured to further include information (update_data in FIG. 53) used to change or update this playback sequence. The playback sequence is configured to be able to include information (param="0" or param="1" in FIG. 53) used to change the advanced objects to be played back.

[0012] By practicing the present invention, an information recording medium and its playback apparatus, which can implement colorful expressions and can form attractive contents can be provided.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

[0013] FIG. 1 shows an example of the data structure of recording information on an information recording medium according to an embodiment of the present invention;

[0014] FIG. 2 shows an example of the file/directory structure according to the embodiment of the present invention;

[0015] FIG. 3 shows an example of the detailed data structure of a video manager information area shown in FIG. 1(e);

[0016] FIG. 4 shows an example of the detailed data structure of a video title set information area shown in FIG. 1(f);

[0017] FIG. 5 shows an example of the data structure in a video title set program chain information table shown in FIG. 4;

[0018] FIG. 6 shows an example of the data structure in program chain information of VMGM_PGCI stored in video manager menu PGCI unit table information shown in FIG. 3 or of VTS_PGCI stored in the video title set program chain information table shown in FIG. 4;

[0019] FIG. 7 shows a sequel to the data structure in the program chain information shown in FIG. 6;

[0020] FIG. 8 is a block diagram showing an example of a system of a reference profile;

[0021] FIG. 9 shows an example of a contents image of an expanded profile as a matrix;

[0022] FIG. 10 is a block diagram showing an example of a system of the expanded profile;

[0023] FIG. 11 is a block diagram showing an example of details of some system blocks shown in FIG. 10;

[0024] FIG. 12 shows an example of a playback image to be played back by a playback apparatus according to the embodiment of the present invention;

[0025] FIG. 13 shows another example of a playback image to be played back by the playback apparatus according to the embodiment of the present invention;

[0026] FIG. 14 shows still another example of a playback image to be played back by the playback apparatus according to the embodiment of the present invention;

[0027] FIG. 15 shows yet another example of a playback image to be played back by the playback apparatus according to the embodiment of the present invention;

[0028] FIG. 16 shows an example of the screen configuration in the reference profile;

[0029] FIG. 17 shows an example of the screen configuration in the expanded profile;

[0030] FIG. 18 shows a description example of a playback sequence file;

[0031] FIG. 19 shows a configuration example of a screen on which a graphics object is appended to DVD-Video contents;

[0032] FIG. 20 shows a configuration example of a screen on which an audio object is appended to the designated DVD-Video contents;

[0033] FIG. 21 shows a configuration example of a screen on which a Vclick object is appended to the designated DVD-Video contents;

[0034] FIG. 22 shows a configuration example of a screen on which a Vclick object is appended to the designated DVD-Video contents;

[0035] FIG. 23 shows the relationship between additional objects and the DVD-Video contents described in the description example of the playback sequence shown in FIG. 18;

[0036] FIG. 24 shows other description examples different from that of the playback sequence file shown in FIG. 18;

[0037] FIG. 25 shows a configuration example of a screen by a description associated with PGC#3 shown in FIG. 24;

[0038] FIG. 26 shows a configuration example of a screen by a description associated with PGC#4 shown in FIG. 24;

[0039] FIG. 27 shows a configuration example of a screen by a description associated with PGC#4 shown in FIG. 24;

[0040] FIG. 28 shows a configuration example of a screen by a description associated with PGC#5 shown in FIG. 24;

[0041] FIG. 29 shows a configuration example of a screen by a description associated with PGC#5 shown in FIG. 24;

[0042] FIG. 30 shows a configuration example of a screen by a description associated with PGC#6 shown in FIG. 24;

[0043] FIG. 31 shows a configuration example of a screen by a description associated with PGC#6 shown in FIG. 24;

[0044] FIG. 32 shows a configuration example of a screen by a description associated with PGC#7 shown in FIG. 24;

[0045] FIG. 33 shows a configuration example of a screen by a description associated with PGC#7 shown in FIG. 24;

[0046] FIG. 34 shows a configuration example of a screen by a description associated with PGC#7 shown in FIG. 24;

[0047] FIG. 35 shows PGCs of the DVD-Video contents and flash objects appended in correspondence with their attributes;

[0048] FIG. 36 is a flowchart for explaining a startup processing sequence according to the embodiment of the present invention;

[0049] FIG. 37 shows an example of a layout image in the reference profile;

[0050] FIG. 38 shows another example of a layout image in the reference profile;

[0051] FIG. 39 is a schematic block diagram showing the arrangement of a streaming apparatus (network compatible disc player) according to the embodiment of the present invention;

[0052] FIG. 40 is a view for explaining the relationship between an object region and object region data according to the embodiment of the present invention;

[0053] FIG. 41 is a view for explaining an example of the data structure of an access unit of object meta data according to the embodiment of the present invention;

[0054] FIG. 42 is a view for explaining another example of the data structure of an access unit of object meta data according to the embodiment of the present invention;

[0055] FIG. 43 is a view for explaining still another example of the data structure of an access unit of object meta data according to the embodiment of the present invention;

[0056] FIG. 44 is a view for explaining an example of the configuration of a Vclick access table according to the embodiment of the present invention;

[0057] FIG. 45 is a view for explaining an example of the structure of an enhanced DVD-Video disc according to the embodiment of the present invention;

[0058] FIG. 46 is a view for explaining an example of the directory structure in the enhanced DVD-Video disc according to the embodiment of the present invention;

[0059] FIG. 47 is a view for explaining an example of the structure of Vclick information according to the embodiment of the present invention;

[0060] FIG. 48 is a view for explaining another example of the structure of Vclick information according to the embodiment of the present invention;

[0061] FIG. 49 is a flowchart for explaining an example of an information recording method using the information recording medium shown in FIG. 1;

[0062] FIG. 50 is a view for explaining tags that can be used in the embodiment of the present invention;

[0063] FIG. 51 is a view for explaining attributes that can be used in the embodiment of the present invention;

[0064] FIG. 52 is a block diagram for explaining the system block arrangement according to another embodiment of the present invention;

[0065] FIG. 53 is a view for explaining an example of a playback sequence (PSQ) which includes information used to update the playback sequence, and information used to update an object;

[0066] FIG. 54 is a flowchart for explaining an example of the processing sequence upon acquiring a new playback sequence on the basis of version information and update information;

[0067] FIG. 55 is a flowchart for explaining an example of the processing sequence upon acquiring an object and time map on the basis of their update information;

[0068] FIG. 56 is a view for explaining an example of a case wherein an object is selected or updated using a flash object;

[0069] FIG. 57 is a view for explaining an example of a playback sequence before change, a playback sequence changed by a script, and the script used to apply such change;

[0070] FIG. 58 is a flowchart for explaining an example of the processing sequence updating the contents of a playback sequence as needed using a script of a flash object; and

[0071] FIG. 59 is a view for exemplifying a reference relationship and the like among the playback sequence, flash object, and DVD-Video object.

DETAILED DESCRIPTION OF THE INVENTION

[0072] An information recording medium and its playback apparatus according to an embodiment of the present invention will be described hereinafter with reference to the accompanying drawings. FIG. 1 shows an example of the data structure of recording information on an information recording medium according to the present invention. FIG. 1(a) shows disc-shaped information recording medium (optical disc complying with the existing or future DVD standard) 1. Details of information recorded on this disc 1 are shown in FIGS. 1(b) to 1(f).

[0073] Information recorded on disc 1 includes lead-in area 10, volume/file structure information area 11, data area 12, and lead-out area 13 from the inner periphery side, as shown in FIG. 1(b). The information recording medium of this embodiment adopts the ISO9660 and UDF bridge structures as a file system, and has ISO9660 and UDF volume/file structure information area 11 in a part of data area 12. Data area 12 allows mixed allocations of video data

recording area 20, another video data recording area 21, and general computer information recording area 22, as shown in FIG. 1(c). (As will be described later with reference to FIG. 45, another video data recording area 21 can record Vclick data according to the present invention.)

[0074] The video data recording area includes video manager recording area (VMG: Video Manager) 30 that records management information associated with the entire DVD-Video contents recorded in video data recording area 20, and video title set recording areas (VTS: Video Title Set) 40 which are arranged for respective titles, and record management information and video information (video objects) for respective titles together, as shown in FIG. 1(d).

[0075] Video manager recording area (VMG) 30 includes video manager information area (VMGI: Video Manager Information) 31 that indicates management information associated with overall video data recording area 20, expanded video object area (VMGM_EVOBS) 32 for a menu, which records background frames for a menu used in entire video data recording area 20, and video manager information backup area (VMGI_BUP) 33 that records the same information as in video manager information area (VMGI) 31 as a backup of video manager information area (VMGI) 31, as shown in FIG. 1(e).

[0076] In addition to the above areas, video manager recording area (VMG) 30 includes advanced function graphics object area (VMGM_AGOBS) 34 for a menu, which allows playback of button layout, button highlight indication, background audio, effect sound, moving picture, and animation, and playback sequence (PSQ) area 35 which specifies playback control of objects other than expanded video objects (EVOB).

[0077] One video title set recording area (VTS) 40 that records management information and video information (video object) together for each title includes video title set information area (VTSI) 41 which records management information for all contents in video title set recording area (VTS) 40, expanded video object area (VTSTT_EVOBS) 42 for a title, which records video object data (video information of a title) in this video title set, and video title set information backup area (VTSI_BUP) 43 which records the same information as in video title set information area (VTSI) 41 as backup data of video title set information area (VTSI) 41, as shown in FIG. 1(f).

[0078] Furthermore, each video title set recording area 40 includes advanced function graphics object area (VTSTT_AGOBS) 44 for a title that allows playback of button layout, button highlight indication, background audio, effect sound, moving picture, and animation, and high-definition text object (VTSTT_ATOBS) 45 which can be played back as a subtitle.

[0079] FIG. 2 shows an example of the file/directory structure according to this embodiment. As shown in FIG. 2, files stored in the disc as information recording medium 1 are managed by the file system such as ISO9660, UDF, or the like. An HVDVD_TS directory for storing information files that handle High-Definition video data, and an ADV_OBJ directory for storing information files that handle advanced object data are allocated under a Root directory.

[0080] The HVDVD_TS directory broadly includes a group of files which belong to a menu group used for a menu, and groups of files which belong to title set groups used for titles. As the group of files that belong to the menu group, an information file (HVI00001.IFO) for a video manager, its backup file (HVI00001.BUP), and playback data files (HVM00001.EVO to HVM00003.EVO) of expanded video object sets for a menu used as background frames of a menu are stored.

[0081] Furthermore, as the group of files that belong to a title set #n group, an information file (HVIxxx01.IFO: xxx=001 to 999) for a video title set having information used to manage title set #n, its backup file (HVIxxx01.BUP: xxx=001 to 999), playback data files (HVTxxxxy.EVO: xxx=001 to 999, yy=01 to 99) of expanded video object sets for title set #n used as a title are stored.

[0082] The ADV_OBJ directory stores control information files (HVI . . . PSQ) for playback sequence PSQ, advanced function graphics object files (HVM . . . AGO) for a menu (this object can be formed using a technique such as Macromedia Flash® or the like, which forms contents by combining audio data and vector graphics animation data), advanced function graphics object files (HVT . . . AGO) for respective title set (#1 to #n) groups (this object can also be formed using Macromedia Flash® or the like), and advanced function text object files (HVT . . . ATO) for respective title set (#1 to #n) groups (this object can be formed using Open type font, True type font, and the like).

[0083] Note that the playback data file (e.g., HVM00001.AGO) of the advanced function graphics object for a menu can be mixed to the playback data files (HVM00001.EVO to HVM00003.EVO), which belong to the menu group in the HVDVD_TS directory, by a blend. This mixing allows button layout and button highlight indication on the screen, and also allows playback of a small animation with background audio.

[0084] Note that the α blend mixes RGB data with a transparency α , and can superimpose an image on another image so that a lower image can be seen. “ α ” in this case indicates the contrast of an upper image to be superimposed on a lower image to the lower image. For example, when $\alpha=100\%$, the upper image is displayed to completely cover the lower image; when $\alpha=0\%$, the upper image to be superimposed disappears. For example, when α is around 50%, the upper image to be superimposed is displayed as a translucent image on the lower image.

[0085] Each playback data file (HVTxxxxy.AGO: xxx=001 to 999, yy=01 to 99) of the advanced function graphics object can be mixed to the playback data files (HVTxxxxy.EVO: xxx=001 to 999, yy=01 to 99), which belong to the title set #n group in the HVDVD_TS directory, by α blend. This mixing allows button layout and button highlight indication on the screen and also allows playback of small animation with background audio together with title contents.

[0086] Each playback data file (HVTxxxxy.ATO: xxx=001 to 999, yy=01 to 99) of the high-definition text object can be played back in place of sub-picture data which is recorded in the expanded video object set for title set #n and is used as a subtitle (or together with sub-picture data as needed). By playing back this high-definition text object, a high-definition subtitle can be superimposed on the lower image (main picture) (since the high-definition subtitle can

be displayed, not only fine subtitle characters can be displayed, but also many characters can be displayed within a limited display space).

[0087] Each control information file (HVI00000.PSQ) for playback sequence PSQ, which defines the playback sequence in advance, describes the playback conditions (timings, positions, sizes, and the like) and/or user action conditions (operation regulations, valid period, and the like) of advanced function graphics objects (.AGO extension) and high-definition text objects (.ATO extension), which can be played back together with the expanded video object sets (.EVO extension). This description can use a language such as XML (Extended Markup Language), JAVA®, and the like.

[0088] FIG. 3 shows the detailed data structure in video manager information area (VMGI) 31 shown in FIG. 1(e). As shown in FIG. 3, video manager information area (VMGI) 31 has video manager information management table (VMGI_MAT) information 310 that records management information common to data recorded in video data recording area 20 and to the entire DVD-Video contents together, title search pointer table (TT_SRPT) information 311 that records information helpful to search for (to detect the start positions of) titles present in the DVD-Video contents, video manager menu PGCI unit table (VMGM_PGCI_UT) information 312 that records management information of a menu screen, which is separately allocated for each menu description language code used to display a menu, parental management information table (PTL_MAIT) information 313 that records information for managing pictures fit or unfit for children to see as parental information, video title set attribute information table (VTS_ATTRT) information 314 that records attributes of title sets together, text data manager (TXTDT_MG) information 315 that records text information to be displayed for the user together, video manager menu cell address table (VMGM_C_ADT) information 316 that records information helpful to search for the start address of a cell that forms the menu screen, and video manager menu expanded video object unit address map (VMGM_EVOBU_ADMAP) information 317 that records address information of VOBUs which indicates a minimum unit of video objects that form the menu screen.

[0089] FIG. 4 shows the detailed data structure in video title set information area (VTSI) 41 shown in FIG. 1(f). As shown in FIG. 4, video title set information area (VTSI) 41 is divided into respective areas (management information groups): video title set information management table (VTSI_MAT) 410, video title set PTT search pointer table (VTS_PTT_SRPT) 411, video title set program chain information table (VTS_PGCIT) 412, video title set time map table (VTS_TMAPT) 413, video title set cell address table (VTS_C_ADT) 414, and video title set expanded video object unit address map (VTS_VOBU_ADMAP) 415.

[0090] Video title set information management table (VTSI_MAT) 410 records management information common to a video title set of interest. Since this common management information is allocated in the first area (management information group) in video title set information area (VTSI) 41, the common management information in the video title set can be quickly loaded, the playback control process of the information playback apparatus can be simplified, and its control processing time can be shortened.

[0091] FIG. 5 shows the data structure in video title set program chain information table (VTS_PGCIT) 412 shown in FIG. 4. As shown in FIG. 5, video title set program chain information table (VTS_PGCIT) 412 records information of video title set PGCI information table (VTS_PGCI) 4121 that includes the number of VTS_PGCI_SRP (VTS_PGCI_SRP_Ns), and information of the end address (VTS_PGCIT_EA) of VTS_PGCIT. Also, VTS_PGCI search pointer (VTS_PGCI_SRP) 4122 records the start address (VTS_PGCI_SA) of video title set program chain (VTS_PGCI) 4123 (a program chain will be described later) together with a VTS_PGC category (VTS_PGC_CAT).

[0092] FIG. 6 shows the data structure in program chain information (PGCI: Program Chain Information) of VMGM_PGCI (not shown) stored in video manager menu PGCI unit table (VMGM_PGCI_UT) information 312 shown in FIG. 3 or VTS_PGCI 4123 (FIG. 5) stored in video title set program chain information table (VTS_PGCIT) 412 shown in FIG. 4. The program chain information (PGCI) includes program chain general information (PGC_GI) 50, program chain command table (PGC_CMDT) 51, program chain program map (PGC_PGMAT) 52, cell playback information table (C_PBIT) 53, and cell position information table (C_POSIT) 54.

[0093] In program chain program map (PGC_PGMAT) 52, a plurality of pieces of program entry cell number 520 information that record entry cell numbers (EN_CN) indicating the cell numbers corresponding to entries are allocated in correspondence with the number of entries. Cell position information table (C_POSIT) 54 has a structure in which a plurality of pieces of cell position information (C_POSI) 540 each formed of a pair of a cell EVOB ID number (C_EVOB_IDN) and cell ID number (C_IDN) are allocated in turn.

[0094] FIG. 7 shows a sequel to the data structure in the program chain information (PGCI) shown in FIG. 6. Cell playback information table (C_PBIT) 53 in PGCI as management information of a corresponding PGC, which records management information associated with each individual cell that forms the PGC, includes one or more pieces of cell playback information (C_PBI) 530. This cell playback information (C_PBI) 530 records a cell category (C_CAT), a cell playback time (C_PBTM) indicating a playback time required to fully play back the corresponding cell, start address position information (C_FEVOBU_SA) of the first EVOBU of a cell, end address position information (C_FILVU_EA) of first interleaved unit ILVU of a cell, start address position information (C_LEVOBU_SA) of the last EVOBU of a cell, and end address position information (C_LEVOBU_EA) of the last EVOBU of the cell.

[0095] Note that the cell category (C_CAT) indicates the start or last cell of an interleaved block when the cell of interest forms an interleaved block corresponding to multi-angle playback, or a part of a general continuous block, or a part of an interleaved block corresponding to multi-angle playback.

[0096] Cell playback information (C_PBI) 530 further records information such as cell command start number information (C_CMD_SN) as information associated with the first cell command number from which a sequential process of a plurality of cell commands that can be designated for each cell is started, cell command continuous

number information (C_CMD_CN) indicating the number of commands, the command processes of which are to be continuously executed as well as the cell command designated by the cell command start number information (C_CMD_SN), and the like.

[0097] FIG. 8 shows the relationship (basic configuration of PSQ information) among data recorded on information recording medium (DVD disc) 1 according to the embodiment of the present invention. DVD-Video navigation information 352 is required to manage playback of DVD-Video object 353 (information having functions corresponding to 31 and 41 in FIG. 1). DVD-Video object 353 includes video information, audio information, subtitle information, and the like (corresponding to 32 and 42 in FIG. 1). Playback sequence (PSQ) 35 is described by, e.g., XML (Extended Markup Language), and is required to play back other objects on the basis of DVD-Video navigation information 352 and playback time information of DVD-Video object 353. Playback sequence (PSQ) 35 describes information such as playback start time information and playback end time information required for synchronization with playback of DVD, display position information on the screen, and the like.

[0098] In the embodiment shown in FIG. 8, as other objects which form playback sequence (PSQ) 35, advanced navigation object 351A that controls the DVD-Video navigation, text object 351B for script screen (script, plot) and chatting, advanced graphics object 351C for still picture, moving picture, and animation data, and audio object 351D for background audio and effect sound are prepared.

[0099] Note that advanced navigation object 351A can be formed using a Script language such as ECMA (European Computer Manufacturers Association) Script, JavaScript, Action Script, or the like. Text object 351B can be formed using a Markup language such as HTML (Hyper Text Markup Language), XHTML (extensible Hyper Text Markup Language), SMIL (Synchronized Multimedia Integration Language), or the like. Advanced graphics object 351C can include still picture data such as JPEG, GIF, PNG, bitmap, or the like, moving picture data such as MPEG-4, MPEG-2, or the like, or animation data such as animation GIF, MNG, SVG (Scalable Vector Graphics), or the like. Audio object 351D can include audio data such as MPEG, AC-3, DTS, MP3, or the like.

[0100] Furthermore, all these objects (advanced navigation object 351A, text object 351B, advanced graphics object 351C, and audio object 351D) can be formed using the aforementioned Macromedia Flash® (corresponding to advanced function graphics objects 34 and 44 in FIG. 1). Such object is defined as a flash object. Note that advanced navigation object 351A can control the playback timings and the like of DVD-Video navigation information 352, and can change attributes (see a description of FIG. 18) of other objects.

[0101] That is, using the objects shown in FIG. 8, a menu formed by this object can be displayed during playback of DVD-Video and a menu with advanced functions compared to that formed by DVD-Video can be displayed. These objects can record required information on a user information storage area (e.g., which is assured on a nonvolatile memory or hard disc drive, and corresponds to storage 126

in FIG. 10 or to information recording area 126X that exploits an HDD or the like in FIG. 52) in the playback apparatus.

[0102] Timed text object 354 includes text data and font data (corresponding to high-definition text object 45 in FIG. 1). The conventional DVD-Video uses sub-picture data to display a subtitle. Timed text object 354 can provide a high-definition subtitle with advanced functions with a smaller data size compared to the sub-picture data formed of compressed bitmap data. For example, text data of timed text object 354 describes, using XML, information such as “display start time, display end time” indicating the display period of that data, “display position” required to lay out data on the screen, “font name, font size, font color” required to display data, “pre-display effect, display effect, post-display effect” indicating effects upon displaying data, and the like.

[0103] As font data, vector font data, such as Open type font data, True type font data, or the like is used. The text data is rendered using this vector font data in accordance with its additional information. The playback apparatus can store the aforementioned font data in advance (e.g., in media decoder 216 in FIG. 39 to be described later).

[0104] Assume that the aforementioned data are recorded in advance on information recording medium 1. However, for example, when these data are to be changed according to the intention of the contents provider (contents producer), changed data may be stored in an external server. In this way, the latest data can be provided to the playback apparatus.

[0105] In addition, in the embodiment of the present invention, objects such as an audio streaming object, AV streaming object, Vclick streaming object, and the like can be played back. Since these data have a large data size, they are recorded on an external server (in, e.g., server 201 in FIG. 39 to be described later), and the playback apparatus (client 200 in FIG. 39) downloads such data for a necessary size at a necessary timing, and deletes unnecessary data. However, these objects can also be recorded on information recording medium 1.

[0106] Audio streaming object 355A in FIG. 8 can be used to play back an audio commentary that outputs commentary audio of a movie given by a director, actor, or the like together with DVD-Video audio, or to play back an audio language other than those of the DVD-Video recorded on information recording medium 1. Audio stream object 355A can include audio data such as MPEG, AC-3®, DTS®, MP3, and the like. Audio stream object 355A can use, e.g., HTTP streaming. HTTP streaming is a method of partially retrieving data on a server using a partial GET request of the HTTP protocol with respect to data on the server.

[0107] The playback apparatus sends position information (e.g., relative address information from the head of a file to be retrieved) or time information of data to be retrieved to the server, and the server sends a part of corresponding data to the playback apparatus. When the time information is sent, the server must convert the time information into position information. (For this purpose, the server preferably has the conversion table.) In order to synchronize audio streaming object 355A with audio of the DVD-Video, meta data included in audio streaming object 355A or that (e.g., mixing coefficients, priority information, and the like) defined by playback sequence PSQ can be used.

[0108] AV streaming object 355B can be used in the same manner as aforementioned audio streaming object 355A. That is, AV streaming object 355B can be used to output commentary audio given by a director or actor of a movie together with his (her) video picture in synchronism with the video and audio data of the DVD-Video, and to deliver video data different from that of the DVD-Video. AV streaming object 355B can include AV data such as MPEG-4, MPEG-2, WindowsMedia®, and the like.

[0109] Vclick streaming object 355C can include information used to display fields that can be clicked by the user, comment information for such clickable fields, information of actions to be taken after the user clicks that field, and the like. This Vclick streaming object will be described later with reference to FIG. 39 and subsequent figures.

[0110] The aforementioned embodiment can be summarized as follows.

<Playback Control>

[0111] The playback order is based on program chains (VTS_PGCI) 4123 shown in FIG. 5 as a basic unit of the playback sequence in the DVD-Video, and controls playback of expanded video objects (EVOB).

[0112] The playback sequence (PSQ) can control the playback timings of respective objects using some triggers (e.g., a time or event defined as an application interface API). The playback sequence (PSQ) has a scaling function (which can designate an arbitrary position, size, and the like).

[0113] In addition, the playback conditions (timings, positions, sizes, and the like) of the advanced function graphics object (AGOB) and high-definition text object (ATOB), which can be played back together with the expanded video object (EVOB), can be defined in advance as the playback sequence (PSQ) that can perform playback control using XML. Also, these objects can be mixed by a blend.

<Playback Object>

[0114] Three different playback objects, i.e., the expanded video object (EVOB), advanced function graphics object (AGOB), and high-definition text object (ATOB), are defined.

[0115] The expanded video object (EVOB) is formed by multiplexing a plurality of streams to an MPEG program stream as in the conventional DVD-Video playback object, and is obtained by expanding streams to support high-definition data and the like.

[0116] The advanced function graphics object (AGOB) can be mixed to the expanded video object (EVOB) by α blend that mixes RGB data with transparency α . This mixing can lay out buttons on the mixing screen, and can highlight buttons. Furthermore, this mixing can play back a small animation with background audio (this advanced function graphics object can be formed using the aforementioned technique such as Macromedia Flash or the like).

[0117] The high-definition text object (ATOB) is multiplexed into the expanded video object (EVOB) and can be defined in addition to a sub-picture stream (low resolution: bitmap format) which is used in subtitle application. The high-definition text object can play back a high-definition subtitle (vector format) in place of the sub-picture stream upon playback.

[0118] The audio streaming object (ASOB), still picture image object (SIOB), expanded text object (ETOB), and other objects (AV streaming object and the like) are defined.

[0119] The audio streaming object (ASOB) is played back in the full title of the DVD-Video while being mixed to audio data of the main title for the purpose of, e.g., audio commentary that a director or the like gives about his or her work. Also, audio data to be played back can be switched like only audio data of the main title, only audio commentary, or both audio data.

[0120] The still picture image object (SIOB) can use photos taken by the user, still picture data delivered on the Internet, and the like. Such still picture image objects can be simply played back like a slide show. As the image format of this still picture image object, JPEG, PNG, and the like can be used.

[0121] The expanded text object (ETOB) can be used to display text, messages, and the like on the script screen.

<Network Connection Function>

[0122] Internet connection can be made.

[0123] The expanded video object (EVOB) and other objects acquired via the Internet connection can be synchronously played back.

[0124] The Internet connection timing is determined by the user or by the timing (disc playback time, position, and the like) the contents provider of the disc intended.

[0125] A chatting function that allows to exchange views on a work such as a movie or the like recorded on the disc with a director is provided.

[0126] A function that allows to download contents and information associated with a work recorded on the disc via the Internet connection is provided.

[0127] A function that permits to play back contents hidden on the disc via the Internet connection is provided.

[0128] A function that allows to purchase goods associated with a work recorded on the disc via the Internet connection is provided.

<Other Functions>

[0129] The expanded video object (EVOB) and other objects supplied from an external device such as a memory card or the like can be synchronously played back using a playback sequence (PSQ) supplied at the same time.

[0130] FIG. 9 shows a display example of the contents image of an expanded profile in the form of a matrix. As shown in FIG. 9, the ordinate plots playback sequence (PSQ) 35 and the types of objects to be supported, and the abscissa plots the groups that form the contents, thus forming the overall matrix.

[0131] The objects to be supported include DVD-Video expanded video objects (EVOB) 353; Flash objects (FLASH) 351 (corresponding to the aforementioned advanced function graphics object); timed text objects (TTXT) 354 (corresponding to the aforementioned high-definition text object); and streaming objects 355 such as AV streaming objects, audio stream objects, and Vclick streaming objects. As a unit that plays back these objects, zero or one VMG group and one or more VTS groups are defined on information recording medium 1.

[0132] The VMG group is used to form various menus, and is an area that records data used to form menus such as a root menu, title menu, audio language menu, subtitle language menu, angle menu, and the like. Each VTS group is an area that records data used to form a title. One playback sequence (PSQ) 35 is included in information recording medium 1, and defines the times, positions, and the like of the objects to be played back using the VMG group and VTS group.

[0133] In the VMG group, zero or more VMGM_EVOBs are set as DVD-Video objects (EVOB) for a menu, and the video manager information (VMGI) which stores program chain (PGC) information that controls playback of objects is set as its DVD navigation (management information). Furthermore, zero or more Flash objects are set in the VMG group.

[0134] In each VTS group, one or more VTSTT_EVOBs are set as DVD-Video objects (EVOB), and video title set information (VTSI) which stores program chain (PGC) information that controls playback of this object is set as DVD navigation (management information). Furthermore, zero or more Flash objects, zero or more timed text objects, and zero or more streaming objects are set in each VTS group.

[0135] In the example of FIG. 9, the groups have been explained while being divided into two groups, i.e., VMG and VTS groups. However, only one group may be set, and need not be distinguished for VMG and VTS. Also, in the example of FIG. 9, only DVD-Video objects 353 and Flash objects 351 are played back in the VMG group. However, timed streaming objects and streaming objects may be played back as needed even in the VMG group.

[0136] FIG. 10 shows the system block arrangement according to the embodiment of the present invention. Playback sequence (PSQ) 35 read out from information recording medium (DVD disc) 1 or designated external server 110 is input to playback sequence parser 123. Parser 123 parses "playback conditions (playback timings, display positions, display sizes, and the like) of objects other than the expanded video objects of the DVD-Video" described in playback sequence (PSQ) 35, thus making the playback control according to the playback conditions.

[0137] On the other hand, DVD-Video navigation information read out from information recording medium 1 is parsed by DVD-Video playback engine 125. In DVD-Video playback engine 125, an MPEG stream formed by multiplexing DVD-Video objects is demultiplexed, and demultiplexed video (main picture), audio (audio), and sub-picture streams are processed by corresponding decoders (not shown). The processed data are sent to layout engine 130. Since playback time information of the DVD-Video objects is also required upon playing back objects other than the DVD-Video objects, it is sequentially sent to playback sequence parser 123 and is used in playback control of respective objects.

[0138] Furthermore, Flash objects, timed text objects, and the like read out from information recording medium 1 or designated external server 110 are temporarily stored in buffer 105 for each group. The Flash objects buffered by buffer 105 are sent to Flash playback engine 127, and the timed text objects are sent to timed text playback engine

128. Data of these objects can be accessed without disturbing playback of the DVD-Video by storing them in buffer **105**, i.e., by avoiding access to information recording medium **1**. In this manner, synchronous playback of the DVD-Video objects and these objects (Flash objects, timed text objects, and the like) can be implemented. Note that the object data are stored in buffer **105** for each group. Alternatively, the object data may be stored for a plurality of groups or for an information recording medium depending on the size of buffer **105**.

[**0139**] The Flash objects are parsed and decoded by Flash playback engine **127**. Note that Flash playback engine **127** parses a user input. Then, engine **127** sends a command to interface handler **124** as an action corresponding to “that user input” set in each Flash object in advance. As this command, a command for controlling playback of the DVD-Video, a command for controlling playback of the timed text objects and streaming objects, and a command for changing the attributes of the timed text objects and streaming objects are available.

[**0140**] Interface handler **124** transfers commands sent from Flash playback engine **127** to respective engines (**125**, **128**, **129**). Flash playback engine **127** can write and read user information to or from user information storage area (persistent storage) (which is assured on, e.g., a nonvolatile memory, hard disc drive, or the like) **126** in the playback apparatus. The user information includes user's personal information, access history, game score, playback sequence data changed by the user input, and the like.

[**0141**] The timed text objects are parsed and decoded by timed text playback engine **128**. Each timed text object describes information such as text information to be displayed, a font name (font data name) used in display, a font size to be displayed, a font color to be displayed, display effects, and the like, and is rendered using corresponding font data according to these pieces of information. Also, streaming objects are parsed and decoded by streaming playback engine **129**.

[**0142**] Layout engine **130** scales (enlarges/reduces in scale) decoded object data sent from the respective engines (**125**, **127** to **129**) in accordance with the designation of playback sequence parser **123**. Furthermore, layout engine **130** forms a screen layout based on a layout designated from playback sequence parser **123**, and applies RGB mixing with transparency α to respective objects in accordance with an α value (a value indicating the transparency or contrast in %) designated by playback sequence parser **123** to composite pictures, thus generating an output picture. Layout engine **130** changes the output levels of respective audio object data, and mixes respective audio object data.

[**0143**] Note that the Flash objects and timed text objects may be downloaded from an external server (e.g., **201** in **FIG. 39** to be described later) via another medium other than information recording medium **1**, e.g., via memory card **109**, or Internet (Web) **110** connection onto buffer **105** when they are used.

[**0144**] **FIG. 11** shows an example of the internal arrangement of playback sequence parser **123** (or playback sequence manager **123X** in **FIG. 52** to be described later) and DVD-Video playback engine **125** in the system block diagram shown in **FIG. 10**. DVD-Video playback engine

125 is a block that plays back DVD-Video data including DVD-Video navigation information and DVD-Video objects, and comprises DVD-Video navigation parsing unit **125A**, DVD clock **125B**, and stream decoder **125C**. DVD-Video navigation parsing unit **125A** parses DVD-Video navigation data loaded from information recording medium **1**, and makes DVD-Video playback control. Stream decoder **125C** includes a video stream decoder, audio stream decoder, and sub-picture stream decoder. Stream decoder **125C** demultiplexes DVD-Video object data in the MPEG-2 format which is loaded from information recording medium **1** into video (main picture), audio, and sub-picture streams, decodes the streams using the corresponding decoders (those in **125C**), and outputs them as video/audio data to layout engine **130**. The DVD-Video object data is decoded under the control of DVD-Video navigation parsing unit **125A**. DVD clock **125B** generates clocks used to synchronously output independently decoded video, audio, and sub-picture streams.

[**0145**] Playback sequence parser **123** (or playback sequence manager **123X** to be described later) is a block for parsing playback sequence data and controlling respective object playback blocks, and includes playback sequence parsing unit **123A**, DVD clock **123B**, and playback information processor **123C**. Playback information processor **123C** includes processing units of Flash playback information, timed text playback information, stream playback information, and DVD-Video playback information.

[**0146**] Playback sequence parsing unit **123A** parses the playback sequence acquired from information recording medium **1** or an external server (**110** in **FIG. 10** or **201** in **FIG. 39**). Playback sequence parsing unit **123A** performs control of playback start, playback end, and the like of designated objects for respective playback engines (DVD-Video playback engine **125**, Flash playback engine **127**, timed text playback engine **128**, streaming playback engine **129**) at timings designated by playback sequence PSQ with reference to the DVD clocks, DVD-Video playback information, and the like. Playback sequence parsing unit **123A** transfers information to layout engine **130** on the basis of layout information designated by playback sequence PSQ.

[**0147**] DVD clock **123B** sequentially receives the same values as those of the DVD clocks in DVD-Video playback engine **125**. The DVD-Video playback information processing unit in playback information processor **123C** sequentially receives playback information (e.g., VMG space or VTS space, title number, PGC number, cell number, audio stream number, sub-picture stream number, angle number, and the like) of DVD-Video playback engine **125**. The Flash playback information processing unit in playback information processor **123C** sequentially receives playback information (e.g., object file name, playback time information, and the like) of Flash playback engine **127**. The timed text playback information processing unit in playback information processor **123C** sequentially receives playback information (e.g., object file name, font name, font size, font color, effects, and the like) of timed text playback engine **128**. Likewise, the streaming playback information processing unit in playback information processor **123C** sequentially receives playback information of streaming playback engine **129**.

[0148] FIGS. 12 to 15 show examples of playback images to be played back by the playback apparatus according to the embodiment of the present invention. FIG. 12 shows a case wherein the playback start/playback end timings of a plurality of Flash objects and a plurality of timed text objects are defined as relative times (relative PTS: Presentation Time Stamp) from the head of one program chain (PGC). Also, these times can be freely set, as shown in FIG. 12.

[0149] As shown in FIG. 12, start time t1 and end time t2 are respectively set for Flash #1351#1 for a playback menu, timed text #1354#1 for an English subtitle, and timed text #4354#4 for a Japanese subtitle with respect to PGC#13531 of the DVD-Video. By setting the same start time/end time, these pieces of information (351#1, 354#1, 354#4) can be synchronously played back. Start time t4 and end time t7 are set for Flash #2351#2 for a playback menu, and start time t3 and end time t5 are set for timed text #2354#2 for an English subtitle and timed text #5354#5 for a Japanese subtitle. In this way, by setting different start times/end times for corresponding information (351#2, and 354#2 and 354#5), synchronous playback (synchronous playback with different playback times; asynchronous playback depending on the perspective) at different timings can be designated.

[0150] Start time t8 and end time t10 are set for Flash #3351#3 for a playback menu, and start time t6 and end time t9 are set for timed text #3354#3 for an English subtitle and timed text #6354#6 for a Japanese subtitle. In this way, information (354#3) of timed text #3 and that (354#6) of timed text #6 can be designated to partially overlap the playback period of Flash #2351#2 (t6 to t7).

[0151] Note that “Flash” of “MM Flash #3351#3” indicates an advanced function graphics object (Flash object) formed using, e.g., Macromedia Flash®. “Flash” can be used as an elaborate object for a menu, which uses a graphical user interface (GUI) and menu expression of the existing DVD-Video during moving picture playback, and has contents which can include animation data and/or still picture data.

[0152] Although not shown, for example, timed text #1354#1 can be set to have a playback period from t1 to t5, and timed text #4354#4 can be set to have a playback period from t1 to t10. In this case, the start times of timed text #1 and timed text #4 matches that of Flash #1 (start synchronous) but the end times of timed text #1 and timed text #4 do not match that of Flash #1 (end asynchronous). In this case, an irregular setting is available: the end time of timed text #4 is matched with that of another Flash #3. Such synchronous/asynchronous settings of “start time” and/or “end time” can be freely set by description contents of “start_ptm=“” and/or “end_ptm=“” in the following PSQ description example.

[0153] A description example of playback sequence PSQ in the above example (FIG. 12) is as follows:

```
<pgc num="1">
<object data="dvd://hvdvd_ts/hvi00001.ifo"/>
<object data="file://dvdrom:/adv_obj/flash1.swf start_ptm=
it1 end_ptm="t2"/> . . . (Description example of Flash#1
from t1 to t2)
<object data="file://dvdrom:/adv_obj/ttext1.xml start_ptm=
"t1" end_ptm="t2"/> . . . (Description example of timed
text#1 from t1 to t2)
```

```
<object data="file://dvdrom:/adv_obj/ttext4.xml start_ptm=
"t1" end_ptm="t2"/> . . . (Description example of timed
text#4 from t1 to t2)
```

```
<object data="file://dvdrom:/adv_obj/flash2.swf start_ptm=
"t4" end_ptm="t7"/> . . . (Description example of Flash#2
from t4 to t7)
```

```
<object data="file://dvdrom:/adv_obj/ttext2.xml start_ptm=
"t3" end_ptm="t5"/> . . . (Description example of timed
text#2 from t3 to t5)
```

```
<object data="file://dvdrom:/adv_obj/ttext5.xml start_ptm=
"t3" end_ptm="t5"/> . . . (Description example of timed
text#5 from t3 to t5)
```

```
<object data="file://dvdrom:/adv_obj/flash3.swf start_ptm=
"t8" end_ptm="t10"/> . . . (Description example of Flash#3
from t8 to t10)
```

```
<object data="file://dvdrom:/adv_obj/ttext3.xml start_ptm=
"t6" end_ptm="t9"/> . . . (Description example of timed
text#3 from t6 to t9)
```

```
<object data="file://dvdrom:/adv_obj/ttext6.xml start_ptm=
"t6" end_ptm="t9"/> . . . (Description example of timed
text#6 from t6 to t9)
```

```
</pgc>
```

[0154] FIG. 13 exemplifies a case wherein Flash objects to be displayed are switched in response to a switch request from the user. This example is defined so that Flash #1351#1 begins to be played back from the head of the program chain (PGC#1) and automatically ends at the end of that program chain for PGC#13531 of the DVD-Video, and the same Flash #1351#1 begins to be played back from the head of the program chain (PGC#2) and automatically ends at the end of that program chain for PGC#23532 of the DVD-Video.

[0155] In the example of FIG. 13, no user's switch request is generated in PGC#2, and first page 3511 of Flash #1 is kept played back from the beginning to the end of playback of PGC#2. In the example of FIG. 13, a user's switch request (User action) is generated during playback of PGC#1. When the user's switch request is generated, the page of Flash #1 which is played back so far is switched to a new page (e.g., second page 3512 of Flash #1), and the new page is kept played back until the end of playback of PGC#2.

[0156] In the example of FIG. 13, PGC#1 and PGC#2 refer to the same Flash object. In this example, it is defined so that a clock (timer) of the Flash object is temporarily reset upon switching PGC#1 to PGC#2. That is, the first page of Flash #1 is switched to the second page in PGC#1, but display starts from the first page as a default state of Flash #1 (since the clock is reset) in PGC#2.

[0157] A description example of playback sequence PSQ in the above example (FIG. 13) is as follows:

```
<pgc num="1">
<object data="dvd://hvdvd_ts/hvi00001.ifo"/>
<object data="file://dvdrom:/adv_obj/flash1.swf/>
(Although not described in this example, the page of Flash
#1 is switched in response to the User action in FIG. 13.)
</pgc>
<pgc num="2">
```

```
<object data="dvd://hvdvd_ts/hvi00001.ifo">
<object data="file://dvdrom:/adv_obj/flash1.swf">
</pgc>
```

[0158] FIG. 14 exemplifies a case wherein an identical Flash object can be played back across a plurality of program chains (PGC#1 and PGC#2). This example is set to synchronously play back identical Flash #1351#1 with respect to PGC#13531 and PGC#23532 of the DVD-Video. In this case, it is defined so that Flash #1 begins to be played back from the head of the first program chain (PGC#1), and automatically ends at the end of the second program chain (PGC#2). When the example of FIG. 14 is compared with FIG. 13, a user's switch request is generated during playback of PGC#1, display is switched from the first page to the second page of Flash #1, and the second page is kept played back in accordance with a continue flag (cont="yes") without resetting the clock (timer) of Flash #1 even after completion of PGC#1 in the example of FIG. 14. For this reason, the second page of Flash #1 is displayed from PGC#1 to PGC#2.

[0159] A description example of playback sequence PSQ in the above example (FIG. 14) is as follows:

```
<pgc num="1">
<object data="dvd://hvdvd_ts/hvi00001.ifo">
<object data="file://dvdrom:/adv_obj/flash1.swf">

(Although not described in this example, the page of Flash
#1 is switched in response to the User action in FIG. 14
during playback of PGC#1.)

</pgc>
<pgc num="2">
<object data="dvd://hvdvd_ts/hvi00001.ifo">
<object data="file://dvdrom:/adv_obj/flash1.swf cont=
"yes">
</pgc>
```

[0160] FIG. 15 exemplifies a case wherein playback of Flash objects is automatically changed in response to a change in playback of PGCs. This example is defined so that Flash #1351#1 begins to be played back from the beginning of PGC#1 and automatically ends at the end of PGC#1 with respect to PGC#13531 of the DVD-Video. Also, it is defined so that Flash #3351#3 begins to be played back from the beginning of PGC#3 and automatically ends at the end of PGC#3 with respect to PGC#33533 of the DVD-Video.

[0161] Assume that the user issues a jump instruction to given time t2 in PGC#3 at time t1 during playback of PGC#1 of the DVD-Video. At this time, the Flash object interrupts playback of Flash #1, which is being played back, and starts playback from time t2 of Flash #3 corresponding to PGC#3. In this manner, even when a discontinuous action such as jump or the like occurs during playback, the DVD-Video and Flash objects can be synchronously played back.

[0162] A description example of playback sequence PSQ in the above example (FIG. 15) is as follows:

```
<pgc num="1">
<object data="dvd://hvdvd_ts/hvi00001.ifo">
```

```
<object data="file://dvdrom:/adv_obj/flash1.swf">
</pgc>
```

(Although not described in this example, Flash #1 is switched to Flash #3 corresponding to PGC#3 as a jump destination in response to the User action in FIG. 15 during playback of PGC#1.)

```
<pgc num="3">
<object data="dvd://hvdvd_ts/hvi00001.ifo">
<object data="file://dvdrom:/adv_obj/flash3.swf">
</pgc>
```

[0163] FIG. 16 shows an example of the screen configuration according to the embodiment of the present invention. FIG. 16(a) shows a case wherein an expanded video object (EVOB) is played back and displayed in a DVD-Video mode (full-screen display) as the conventional display screen. FIG. 16(b) shows an example wherein a Flash object having playback control buttons is superimposed as a playback control menu by α blend on the expanded video object (EVOB) in the DVD-Video mode shown in FIG. 16(a). FIG. 16(c) shows an example wherein a timed text object such as outline font, vector font, or the like is superimposed by α blend in place of a sub-picture subtitle in the DVD-Video mode shown in FIG. 16(a).

[0164] FIG. 17 shows an example of the screen configuration according to the embodiment of the present invention. As in FIG. 16, FIG. 17(a) shows a case wherein an expanded video object (EVOB) is played back and displayed in a DVD-Video mode (full-screen display) as the conventional display screen. FIG. 17(b) shows an example wherein a display area is divided into some areas, and object sizes are changed to display objects on the respective areas.

[0165] In this example, the screen size of the expanded video object (EVOB) in the DVD-Video mode shown in FIG. 17(a) is reduced by the scaling function and is laid out on the upper left area. A Flash object (advanced function object AG0) is embedded on the upper right area. A timed text object is displayed on the lower half area of the screen together with a comment of the screen. Furthermore, hot spots (fields on the screen where some process is executed upon clicking a mouse button; to be also referred to as "Vclick" hereinafter as needed) 701 to windmill information are superimposed on the expanded video object (EVOB) on the upper left area. When the user clicks the hot spot, display jumps to (windmill) related information (not shown), thus playing back that related information.

[0166] The file configuration and screen display examples based on that configuration of the playback sequence (PSQ) that defines the playback conditions of Flash objects, timed text objects, and the like will be described in detail below. FIG. 18 shows an example of the configuration of a playback sequence (PSQ) file. The playback sequence (PSQ) file is described using XML, use of "XML" is declared first, and a playback sequence (PSQ) file described in XML is then declared. Furthermore, the contents of the playback sequence (PSQ) file are described using a <video_pbseq> tag.

[0167] The <video_pbseq> field includes zero or one <vmg> tag, zero or one or more <vts> tags, and zero or one <idle> tag. The <vmg> field represents the VMG space in

the DVD-Video. That is, the <vmg> field indicates that additional objects such as a Flash object (to be referred to as a graphics object hereinafter), timed text object (to be referred to as a text object hereinafter), audio streaming object, AV streaming object, and Vclick streaming object described in the <vmg> field are appended to DVD-Video data on the VMG space.

[0168] The <vts> field represents the VTS space in the DVD-Video. That is, the <vts> field designates the VTS space number by appending a num attribute in the <vts> tag, and indicates that additional objects such as a Flash object (to be referred to as a graphics object hereinafter), timed text object (to be referred to as a text object hereinafter), audio streaming object, AV streaming object, and Vclick streaming object described in the <vts> field are appended to DVD-Video data on the VTS space. For example, <vts num="n"> represents the n-th VTS space. That is, <vts num="n"> indicates that the aforementioned additional objects described in the <vts num="n"> field are appended to DVD-Video data that forms the n-th VTS space.

[0169] The <vmgm> field includes zero or one or more <vmgm> tags, and zero or one or more <fp> tags. The <vmgm> field represents a VMG menu domain on the VMG space, and designates the VMG menu domain number by appending a num attribute in the <vmgm> tag. For example, <vmgm num="n"> represents the n-th VMG menu domain. <vmgm num="n"> indicates that the aforementioned additional objects described in the <vmgm num="n"> field are appended to DVD-Video data that forms the n-th VMG menu domain. Note that the VMG space includes language blocks. That is, one VMG menu domain corresponds to one language unit. Therefore, the VMG menu domains can be managed using language codes in place of numbers. In this case, each VMG menu domain can be expressed by <vmgm lang="xx"> (xx is a language code specified by ISO639) using a lang attribute in place of the num attribute. For example, <vmgm lang="jp"> indicates a VMG menu domain in Japanese, and <vmgm lang="en"> indicates a VMG menu domain in English.

[0170] Furthermore, the <vmgm> field includes zero or one or more <pgc> tags. The <pgc> field represents a PGC (Program Chain) in the VMG menu domain, and designates the PGC number by appending a num attribute in the <pgc> tag. For example, <pgc num="n"> represents the n-th PGC. <pgc num="n"> indicates that the aforementioned objects described in the <pgc num="n"> field are appended to DVD-Video data that forms the n-th PGC.

[0171] Although not shown, the <fp> field represents a first play domain on the VMG space, and includes zero or one or more <pgc> tags. This <pgc> field indicates a PGC (Program Chain) to be executed by the playback apparatus first.

[0172] The <vts> field includes one or more <vts_tt> tags, and zero or one or more <vtsm> tags. The <vts_tt> field represents a title domain on the VTS space, and designates the title domain number by appending a num attribute in the <vts_tt> tag. For example, <vts_tt num="n"> indicates the n-th title domain. <vts_tt num="n"> indicates that the aforementioned additional objects described in the <vts_tt num="n"> field are appended to DVD-Video data that forms the n-th title domain.

[0173] The <vtsm> field represents a VTS menu domain on the VTS space, and designates the VTS menu domain number by appending a num attribute in the <vtsm> tag. For example, <vtsm num="n"> indicates the n-th VTS menu domain. <vtsm num="n"> indicates that the aforementioned additional objects described in the <vtsm num="n"> field are appended to DVD-Video data that forms the n-th VTS menu domain.

[0174] Since the VTS space includes language blocks, i.e., since one VTS menu domain corresponds to one language unit, the VTS menu domains can be managed using language codes in place of numbers. In this case, each VTS menu domain can be expressed by <vtsm lang="xx"> (xx is a language code specified by ISO639) using a lang attribute in place of the num attribute. For example, <vtsm lang="jp"> indicates a VTS menu domain in Japanese, and <vtsm lang="en"> indicates a VTS menu domain in English.

[0175] Furthermore, the <vts_tt> or <vtsm> field includes zero or one or more <pgc> tags. The <pgc> field represents a PGC (Program Chain) in the title domain or VTS menu domain, and designates the PGC number by appending a num attribute in the <pgc> tag. For example, <pgc num="n"> represents the n-th PGC. <pgc num="n"> indicates that the aforementioned objects described in the <pgc num="n"> field are appended to DVD-Video data that forms the n-th PGC.

[0176] Finally, although not shown, an <idle> tag represents a state which is not synchronized with playback of the DVD-Video. That is, in the state defined by the <idle> tag, no DVD-Video objects are played back, and this state includes only additional objects such as a Flash object (to be referred to as a graphics object hereinafter), timed text object (to be referred to as a text object hereinafter), audio streaming object, AV streaming object, and Vclick streaming object. The <idle> tag includes zero or one or more <pgc> tags. This <pgc> tag represents a PGC (Program Chain). However, the PGC represented by this tag has no DVD-Video time information, and the playback start time and playback end time cannot be designated, since no DVD-Video display is made.

[0177] In the example shown in FIG. 18, four additional objects are appended to the DVD-Video contents. For example, the first Flash object (to be referred to as a graphics object hereinafter) is designated using an <object> tag in <pgc num="1"> in <vmgm num="1"> in <vmg>. This indicates that the additional object designated by the <object> tag is appended to the first PGC in the first VMG menu domain on the VMG space.

[0178] The <object> tag indicates the location of the additional object using a "data" attribute, i.e., a URI (Uniform Resource Identifier). In this example, the location of the graphics object is designated by "file://dvdrom:/adv_obj/flash1.swf" (see 801 in FIG. 18). Note that "file://dvdrom:/" indicates that the graphics object is present in the information recording medium (disc). Furthermore, "adv_obj/" indicates that the object is present under the "ADV_OBJ" directory, and "flash1.swf" indicates the file name of the graphics object.

[0179] With this description, graphics object 812 can be appended to designated DVD-Video contents 811, as shown in FIG. 19. Especially, graphics object 812 can process

operations from the user. By assigning commands for controlling DVD-Video contents **811** in response to operations from the user in graphics object **812**, graphics object **812** can be used as a menu for controlling DVD-Video contents **811**.

[0180] The next additional object is designated using an <object> tag in <vmgm num="n"> in <vmg>. This indicates that the additional object designated by the <object> tag is appended to the entire first VMG menu domain on the VMG space. The <object> tag indicates the location of the additional object using a "data" attribute. In this example, the location of an audio streaming object is designated by "http://www.hddvd.com/adv_obj/commentary.ac3" (see **802** in **FIG. 18**). Note that "http://www.hddvd.com/adv_obj/" indicates that the audio streaming object is present in an external server, and "commentary.ac3" indicates the file name of the audio streaming object.

[0181] With this description, as shown in **FIG. 20**, the audio streaming object can be appended to the designated DVD-Video contents. For example, audio commentary **824** of a director or actor, which is delivered from the external server (or is recorded in advance on the disc) can be appended to audio data **822** and **823** of DVD-Video contents **821**.

[0182] The third additional object is designated using an <object> tag in <pgc num="1"> in <vts_tt num="1"> in <vts num="1">. This indicates that the additional object designated by the <object> tag is appended to the first PGC in the first title domain on the first VTS space. The <object> tag indicates the location of the additional object using a "data" attribute. In this example, the location of a Vclick streaming object is designated by "http://www.hddvd.com/adv_obj/hotspot.vck" (see **803** in **FIG. 18**). Note that "http://www.hddvd.com/adv_obj/" indicates that the Vclick streaming object is present in the external server, and "hotspot.vck" indicates the file name of the Vclick streaming object.

[0183] With this description, as shown in **FIG. 21**, Vclick objects **832** and **833** can be appended to designated DVD-Video contents **831**. Note that the Vclick object is an object which indicates a clickable field on the DVD-Video contents, and can describe an action after clicking. More specifically, Vclick objects **832** and **833** are appended to two persons by DVD-Video contents **831** in **FIG. 21**. Furthermore, text messages **834** and **835** that give explanations of these objects **832** and **833** can be displayed for objects **832** and **833**. Note that reference numeral **836** denotes audio data provided by DVD-Video contents **831**.

[0184] For example, when the user selects (clicks) Vclick object **833** on the right side of the screen, an action corresponding to this clicking is described in Vclick object **833**, and the playback apparatus operates according to this description. In this example, object **833** describes playback of Markup page **837**, and also a reduction instruction of the DVD-Video contents in response to user's clicking (reference numeral **838** denotes DVD-Video contents reduced in scale).

[0185] The fourth additional stream is designated using an <object> tag in <vts_tt num="n"> in <vts num="1">. This indicates that the additional object designated by the <object> tag is appended to the n-th title domain on the first VTS space.

[0186] The <object> tag indicates the location of the additional object using a "data" attribute. In this example, the location of a Markup language object is designated by "file://dvdrom:/adv_obj/index.xhtm" (see **804** in **FIG. 18**). Note that "file://dvdrom:/adv_obj/" indicates that the Markup language object is present under the "ADV_OBJ" directory in the disc, and "index.xhtm" indicates the file name of the Markup language object. With this description, as will be described using **FIG. 22**, a Markup language object (see **841** in **FIG. 22**) can be appended to designated DVD-Video contents (see **842** in **FIG. 22**). Using the Markup language object, additional objects such as a new image (see **843** in **FIG. 22**), background (see **844** in **FIG. 22**), and text data (see **845** in **FIG. 22**) can be loaded and displayed.

[0187] Furthermore, the fifth object is also designated using an <object> tag in <vts_tt num="n"> in <vts num="1">. This indicates that the additional object designated by the <object> tag is appended to the n-th title domain on the first VTS space. That is, the two additional objects are appended to the n-th title domain on the first VTS space.

[0188] The <object> tag indicates the location of the additional object using a "data" attribute. In this example, the location of a text object is designated by "file://dvdrom:/adv_obj/ttext.xml" (see **805** in **FIG. 18**). Note that "file://dvdrom:/adv_obj/" indicates that the text object is present under the "ADV_OBJ" directory in the disc, and "ttext.xml" indicates the file name of the text object. With this description, as will be described using **FIG. 22**, a timed text object (see **846** in **FIG. 22**) can be appended to the designated DVD-Video contents (see **842** in **FIG. 22**).

[0189] **FIG. 22** shows an example of a screen formed of a Markup object that forms a background, text, and picture, reduced-scaled DVD-Video contents, and a timed text object laid out on the DVD-Video contents. In **FIG. 22**, reference numeral **841** denotes a Markup page; **842**, reduced-scaled DVD-Video contents; **843**, a picture called by the Markup object; **844**, a background provided by the Markup object; **845**, text provided by the Markup object; and **846**, text provided by the timed text object superimposed on reduced-scale DVD-Video contents **842**.

[0190] **FIG. 23** exemplifies the relationship between the DVD-Video contents and the additional objects described in the aforementioned description example of the playback sequence (PSQ). In this example, Flash object **351** is set in PGC#1 of VMG menu #1 in video manager VMG, and audio streaming object **355A** is set in PGC#1 to PGC#n of VMG menu #n in that video manager. Also, a Vclick stream is set in PGC#1 of VTS title #1 in video title set VTS#1, and Markup language object MUS and timed text object **354** are set in PGC#1 to PGC#n of VTS title #n in that video title set.

[0191] **FIG. 24** shows other description examples (a total of nine examples) of the playback sequence (PSQ) file. In the first example (see **851** in **FIG. 24**), one graphics object (graphics object #1) recorded on the disc is appended to one PGC (PGC#1). Note that "width", "height", "position", "start", "end", and "priority" attributes are described in an <object> tag.

[0192] The "width" attribute indicates the length (unit: pixels) in the horizontal direction upon displaying the additional object. The "height" attribute indicates the length

(unit: pixels) in the vertical direction upon displaying the additional object. When the length designated by “width/height” is smaller than the original length of the additional object, the additional object is reduced; when it is larger than the original length of the additional object, the additional object is enlarged. The “position” attribute indicates the coordinate position (unit: pixels) upon displaying the additional object. Note that the coordinate position can be expressed by “(x, y)” of a system which has the upper left point as an origin “(0, 0)”, the abscissa as the x-axis, and the ordinate as the y-axis.

[0193] Note that the “width”, “height”, and “position” attributes can be expressed using a “style” attribute as follows:

```
<OBJECT style=
“position:fixed;top:X1px;left:Y1px;width:X2px;height:Y2
px”
```

```
data=“dvd://hvdvd_ts/hvi00001.ifo”/>
```

[0194] Note that “top:X1px” indicates the X-coordinate (X1 is a pixel value) of the layout position of an object (DVD-Video contents in the following example) from the upper left corner of the screen. “left:Y1px” indicates the Y-coordinate (Y1 is a pixel value) of the layout position of an object from the upper left corner of the screen. These pieces of information are equivalent to those designated by the aforementioned “position” attribute. Also, “width:X2px” indicates the length of an object to be laid out in the horizontal direction (X2 is a pixel value). This information is equivalent to that designated by the aforementioned “width” attribute. Furthermore, “height:Y2px” indicates the length of an object to be laid out in the vertical direction (Y2 is a pixel value). This information is equivalent to that designated by the aforementioned “height” attribute. Also, “position:fixed;” indicates that the layout method designates an absolute position and fixed position.

[0195] The “start” attribute expresses a relative value of the display start time of the additional object with a precision of ‘HH:MM:SS:FF’ (time:minute:second:frame) format. The “end” attribute expresses a relative value of the display end time of the additional object with a precision of ‘HH:MM:SS:FF’ (time:minute:second:frame) format. Note that the “start” and “end” attributes represent relative times from the PGC start position when the additional object is appended to the PGC as in this example. If the additional object is appended to a title domain (“<vts_tt>”), these attributes represent the relative times from the start position of the title domain.

[0196] The “priority” attribute indicates the relationship between the additional object and DVD-Video contents. The “priority” attribute of the DVD-Video contents normally indicates “0” as a default value. If the “priority” attribute assumes a positive value, the additional object is laid out on the front side of the DVD-Video contents; if the “priority” attribute assumes a negative value, the additional object is laid out behind the DVD-Video contents. That is, the object laid on the front side is visually prioritized and is also prioritized to a user input. For example, when two objects are superimposed, which object is to be displayed on the front side can be determined based on the “priority” attributes. As another example, when two objects can respectively process a user input, which object preferentially

processes the user input can be determined based on the “priority” attributes. If there are plurality of objects, they are laid out in turn on the front side as they have larger values. (Even when the value of the “priority” attribute is defined to be located on the front side (to have higher priority) with decreasing value, the same effect can be obtained.)

[0197] FIG. 19 above shows the configuration example of the screen based on PGC#1. On this screen, DVD-Video contents 811 are full-screen displayed, and graphics object 812 is displayed at the designated position to have the designated size. Since the “priority” attribute of graphics object 812 is larger than that of DVD-Video contents 811, graphics object 812 is displayed on the front side, and processes user’s operations first. Furthermore, the display period of graphics object 812 is five minutes from immediately after the beginning of PGC#1.

[0198] Note that priority upon receiving processes of user’s operations can be designated using the “priority” attribute (or another new attribute). For example, assume that a menu screen formed by the DVD-Video contents and that formed by the graphics object are simultaneously displayed. In this case, the “priority” attribute of the DVD-Video contents normally indicates “0” as a default value. If the “priority” attribute of the graphics object assumes a positive value, the graphics object processes user’s operations prior to the DVD-Video contents. If the “priority” attribute of the graphics object assumes a negative value, the DVD-Video contents processes user’s operations prior to the graphics object. If there are a plurality of objects, they have higher priority as they have larger values.

[0199] In the second example (see 852 in FIG. 24), one graphics object (graphics object #1) recorded on the disc is appended to one PGC (PGC#2). Note that “width”, “height”, “position”, “start_ptm”, “end_ptm”, and “priority” attributes are described in an <object> tag. The “start_ptm” attribute represents a relative value of the display start time of the additional object with a precision of PTM (Presentation Time: a counter using 90-kHz clocks). The “end_ptm” attribute represents a relative value of the display end time of the additional object with a precision of PTM. In this case, the “start_ptm” and “end_ptm” attributes represent the relative times from the PGC start position when the additional object is appended to the PGC like in this example. If the additional object is appended to a title domain (“<vts_tt>”), these attributes represent the relative times from the title domain start position.

[0200] In the third example (see 853 in FIG. 24), one graphics object (graphics object #1) recorded on the disc is appended to one PGC (PGC#3). Unlike in the above two examples, this example reduces the DVD-Video contents in scale. When the value of a “data” attribute in an <object> tag is “dvd://hvdvd_ts/hvi00001.ifo”, this means to display the DVD-Video contents. In the above examples, a description using the <object> tag is omitted since the DVD-Video contents are full-screen displayed. In this example, since values are described in the <object> tag required to display the DVD-Video contents using “width”, “height”, and “position” attributes, the DVD-Video contents can be displayed in a reduced-scale.

[0201] Note that an “alpha” attribute represents an alpha value, i.e., transparency α . The alpha value can designate from “0” to “255”: 0 indicates transparency, and “255”,

opacity. Also, the alpha value can be expressed in percentage figures, i.e., from “0%” to “100%”. At this time, “0%” indicates transparency, and “100%”, opacity. The next <object> tag is used to display the graphics object, and “width”, “height”, and “position” attributes are omitted since the graphics object is to be full-screen displayed.

[0202] FIG. 25 shows a configuration example of the screen based on PGC#3. In this example, graphics object 861 is full-screen displayed, and DVD-Video contents 862 are displayed at a designated position to have a designated size. Since the “priority” attribute of graphics object 861 is smaller than that of DVD-Video contents 862, reduced-scale DVD-Video contents 862 is displayed on the front side, and processes user’s operations first. This example includes menu buttons 863 which are provided by DVD-Video contents 862 and are used for chapter playback, and playback menu buttons 864 provided by the graphics object. Since DVD-Video contents 862 have higher priority, it is checked first if DVD-Video contents 862 have settings for user’s operations. If such settings are found, DVD-Video contents 862 process user’s operations; otherwise, graphics objects 861 processes user’s operations.

[0203] In the fourth example (see 854 in FIG. 24), two graphics objects (Flash #3, Flash #4) recorded on the disc are appended to one PGC (PGC#4). Note that an “audio” attribute in an <object> tag corresponds to the audio stream number. In this example, when audio stream #1 of the DVD-Video contents is played back, Flash #3 (flash3.swf) is synchronously played back; when audio stream #2 is played back, Flash #4 (flash4.swf) is synchronously played back.

[0204] For example, when audio stream #1 of the DVD-Video contents corresponds to Japanese language, and audio stream #2 corresponds to English language, Flash #3 (see 871) is configured using Japanese (that is, display of the graphics object is described in Japanese or the access destination of the graphics object is contents described in Japanese), as shown in FIG. 26. Also, Flash #4 (881) is configured using English (that is, display of the graphics object is described in English or the access destination of the graphics object is contents described in English), as shown in FIG. 27. In this way, the audio language (883 in FIG. 27 or 893 in FIG. 28) of the DVD-Video contents (882 in FIG. 27 or 892 in FIG. 28) can be matched with that of the graphics object (881 in FIG. 27 or 891 in FIG. 28).

[0205] In practice, the playback apparatus searches this playback sequence (PSQ) file for the corresponding graphics object with reference to a system parameter indicating the audio stream number in the playback apparatus, and plays it back. For example, when audio stream #1 is played back, and corresponding Flash #3 is displayed, if the user changes audio to be played back to audio stream #2 using a remote controller or the like, Flash #4 corresponding to the changed audio stream is displayed as the graphics object to be displayed according to the playback sequence file.

[0206] In this example, audio streams are managed using the stream numbers, but they may be managed using language codes in place of the stream numbers. In this case, each audio stream can be expressed by audio_lang=“xx” (xx is a language code specified by ISO639) using an audio_lang attribute in place of the audio attribute. For example, audio_lang=“jp” indicates an audio stream in Japanese, and audio_lang=“en” indicates an audio stream in English.

[0207] In the fifth example (see 855 in FIG. 24), three graphics objects (Flash #5, Flash #6, Flash #7) recorded on the disc are appended to one PGC (PGC#5). Note that a “subpic” attribute in an <object> tag corresponds to the sub-picture stream number (sub-picture number). In this example, when sub-picture stream #1 of the DVD-Video contents is played back, Flash #5 (flash5.swf) is synchronously played back. When sub-picture stream #2 is played back, Flash #6 (flash6.swf) is synchronously played back. Also, when sub-picture stream #3 is played back, Flash #7 (flash7.swf) is synchronously played back.

[0208] For example, when sub-picture stream #1 of the DVD-Video contents corresponds to a Japanese subtitle, and sub-picture stream #3 corresponds to an English subtitle, Flash #5 (see 891 in FIG. 28) is configured using Japanese (that is, display of the graphics object is described in Japanese or the access destination of the graphics object is contents described in Japanese), as shown in FIG. 28. Also, Flash #7 (see 901 in FIG. 29) is configured using English (that is, display of the graphics object is described in English or the access destination of the graphics object is contents described in English), as shown in FIG. 29. Hence, the subtitle language (893 in FIG. 28 or 903 in FIG. 29) of the DVD-Video contents (892 in FIG. 28 or 902 in FIG. 29) can be matched with that of the graphics object (891 in FIG. 28 or 901 in FIG. 29).

[0209] In practice, the playback apparatus searches this playback sequence (PSQ) file for the corresponding graphics object with reference to a system parameter indicating the sub-picture stream number in the playback apparatus, and plays it back. For example, when sub-picture stream #1 is played back, and corresponding Flash #5 is displayed, if the user changes a subtitle (sub-picture) to be played back to sub-picture stream #3 using a remote controller or the like, Flash #7 corresponding to the changed sub-picture stream is displayed as the graphics object to be displayed according to the playback sequence file.

[0210] In this example, sub-picture streams are managed using the stream numbers, but they may be managed using language codes in place of the stream numbers. In this case, each sub-picture stream can be expressed by subpic_lang=“xx” (xx is a language code specified by ISO639) using a subpic_lang attribute in place of the subpic attribute. For example, subpic_lang=“jp” indicates a sub-picture stream in Japanese, and subpic_lang=“en” indicates a sub-picture stream in English.

[0211] In the sixth example (see 856 in FIG. 24), two graphics objects (Flash #8, Flash #9) recorded on the disc are appended to one PGC (PGC#6). Note that an “angle” attribute in an <object> tag corresponds to the angle number. In this example, when angle #1 of the DVD-Video contents is played back (see 911 in FIG. 30), Flash #8 (flash8.swf) is synchronously played back (see 912 in FIG. 30). When angle #3 is played back (see 921 in FIG. 31), Flash #9 (flash9.swf) is synchronously played back (see 922 in FIG. 31). Also, when angle #2 is played back, no graphics object is played back.

[0212] Normally, in case of different angles, since the layouts of persons, buildings, and the like differ, the sizes and positions of graphics objects are preferably independently set for respective angles. (Respective graphics object data may be multiplexed to one graphics object.) In practice,

the playback apparatus searches this playback sequence (PSQ) file for the corresponding graphics object with reference to a system parameter indicating the angle number in the playback apparatus, and plays it back.

[0213] In the seventh example (see 857 in FIG. 24), three graphics objects (Flash #10, Flash #11, Flash #12) recorded on the disc are appended to one PGC (PGC#7). Note that an “aspect” attribute in an <object> tag corresponds to a (default) display aspect ratio, and a “display” attribute in the <object> tag corresponds to a (current) display mode. In this example, the DVD-Video contents themselves have an aspect ratio “16:9”. That is, in this example, “wide” output (see 931 in FIG. 32) is permitted for a TV monitor having an aspect ratio “16:9”, and a “letter box (lb)” output (see 941 in FIG. 33) or “pan scan (ps)” output (see 951 in FIG. 34) is permitted for a TV monitor having an aspect ratio “4:3”.

[0214] Based on such settings, when the (default) display aspect ratio is “16:9” and the (current) display mode is “wide”, Flash #10 is synchronously played back as a graphics object (see 932 in FIG. 32). When the (default) display aspect ratio is “4:3” and the (current) display mode is “lb”, Flash #11 is synchronously played back (see 942 in FIG. 33). When the (default) display aspect ratio is “4:3” and the (current) display mode is “ps”, Flash #12 is synchronously played back (see 952 in FIG. 34).

[0215] For example, a graphics object which is displayed right beside a person at the aspect ratio “16:9” is displayed on the upper or lower (black) portion of the screen when the aspect ratio is “4:3” in the “letter box” display mode. At the aspect ratio “4:3” in the “pan scan” display mode, the right and left portions of the screen are cut, but a graphics object is changed to a displayable position. According to the screen configuration, the size of a graphics object can be reduced or enlarged, or the text size in the graphics object can be reduced or enlarged. As a result, a graphics object can be displayed in correspondence with the display state of the DVD-Video contents. In practice, the playback apparatus searches this playback sequence (PSQ) file for the corresponding graphics object with reference to system parameter indicating “default display aspect ratio” and “current display mode” in the playback apparatus, and plays it back.

[0216] In the eighth example (see 858 in FIG. 24), one graphics object (Flash #13) recorded on the disc is appended to one PGC (PGC#8). As in the above example, an “aspect” attribute in an <object> tag corresponds to a (default) display aspect ratio, and a “display” attribute in the <object> tag corresponds to a (current) display mode. In this example, the DVD-Video contents itself have an aspect ratio “4:3”, and this sequence is applied to a TV monitor having an aspect ratio “4:3” upon outputting in a “normal” mode.

[0217] Finally, a case will be exemplified below wherein the aforementioned functions can be used in combination. That is, in the ninth example (see 859 in FIG. 24), four graphics objects (Flash #13, Flash #14, Flash #15, Flash #16) recorded on the disc are appended to one PGC (PGC#9). In this example, when audio stream #1 of the DVD-Video contents is played back, sub-picture stream #1 is played back, and angle #1 is played back, Flash #13 (“flash13.swf”) is synchronously played back. When audio stream #1 is played back, sub-picture stream #2 is played back, and angle #1 is played back, Flash #14 (“flash14.swf”) is synchronously played back. When angle #2 is played

back, Flash #15 (“flash15.swf”) is synchronously played back. When audio stream #2 is played back and sub-picture stream #2 is played back, Flash #16 (“flash16.swf”) is synchronously played back.

[0218] FIG. 35 shows an example of the correspondence between PGCs of the DVD-Video contents, and Flash objects appended to their attributes in association with the above nine examples.

[0219] The playback apparatus of this embodiment loads the playback sequence (PSQ) file in advance or refers to it as needed prior to playback of the DVD-Video contents, thus changing, in real time, additional objects such as a Flash object, timed text object, audio streaming object, Markup language object, and Vclick streaming object, which are to be appended sequentially, in accordance with the playback state of the DVD-Video contents. In this way, even when the user has changed the playback state, an additional object suited to the changed playback state can be played back.

[0220] The number of files (the number of objects) of one additional object is increased to reduce the file size, thereby decreasing an area (buffer) used to store additional objects required for the playback apparatus. Although the file size becomes large, if the number of files is decreased (i.e., one object includes a plurality of additional objects), when the playback state of the DVD-Video contents has changed, additional objects can be smoothly switched. Hence, a high degree of freedom in authoring can be assured upon forming additional objects.

[0221] FIG. 36 is a flowchart for explaining the startup processing sequence according to the embodiment of the present invention. The playback apparatus loads the playback sequence (PSQ) and DVD-Video navigation information from the information recording medium, and stores them in its memory (work memory or buffer; e.g., 105 in FIG. 10 or 209 in FIG. 39) (step ST10). Note that these pieces of information can be loaded from an external server on the basis of version information included in the information itself. From this playback sequence, information such as the locations of objects to be played back, their file names, and the like can be acquired. Based on the acquired information, required objects are extracted (step ST12). Information of each object to be extracted may be recorded on the information recording medium or external server as, e.g., loading information, and may be referred to from the playback sequence.

[0222] The extracted object data are stored in the buffer (step ST14). Upon storing the object data in the buffer, some methods of determining the order of buffering of objects (methods of setting priority of loading onto the buffer) are available.

[0223] In the first example of that method, objects are loaded in turn from those with smaller PGC numbers of the DVD-Video contents corresponding to respective streams. For example, when object #1 is appended to PGC#1, object #2 is appended to PGC#2, and streams to be loaded are objects #1 and #2, object #1 is loaded first onto the buffer, and object #2 is then loaded onto the buffer.

[0224] In the second example, the priority of loading is determined in accordance with a language pre-set in a player (client 200 in FIG. 39 or the like). For example, assume that Japanese is set in the player as the first language (or default

language), and English is set as the second language. In this case, when an English attribute (e.g., English is designated by a “language” attribute) is assigned to object #1, and a Japanese attribute (e.g., Japanese is designated by a “language” attribute), object #2 is loaded first onto the buffer, and object #1 is then loaded onto the buffer.

[0225] In the third example, streams to be loaded, which are recorded on the disc, are given priority over those recorded on the external server. For example, when object #2 is recorded on the external server, and objects #1 and #3 are recorded on the disc, objects #1 and #3 are loaded onto the buffer in the order of their numbers, and object #2 is then loaded onto the buffer upon loading objects #1 to #3.

[0226] In the fourth example, objects are loaded onto the buffer in the order of objects described in the playback sequence (PSQ) or in the order of objects described in information of objects to be extracted (corresponding to the aforementioned loading information) designated by the playback sequence. In this case, the contents provider can determine the priority of objects, and the playback sequence or loading information must be created accordingly.

[0227] After objects are stored up to a size assigned to the buffer (for example, this size is designated by the playback sequence) (YES in step ST16), playback of the DVD-Video contents starts. Alternatively, if the size of objects stored in the buffer has reached a predetermined playback size (for example, this size is designated by the playback sequence), playback of the DVD-Video contents may start.

[0228] After playback of the DVD-Video contents has started (step ST18), its playback information (title, PGC number, audio stream number, sub-picture stream number, angle number, aspect ratio information, playback time information, and the like) is acquired (step ST20), and the playback sequence is searched for the corresponding object on the basis of at least some pieces of information (e.g., PGC number) of the acquired information (step ST22). If data of the object to be played back is currently stored in the buffer (YES in step ST24), playback of that object immediately starts in synchronism with the current DVD playback.

[0229] If data of the object is not currently stored in the buffer (NO in step ST24), data of the retrieved object is loaded onto the buffer (step ST28) by deleting unnecessary data from the buffer (step ST26), or by overwriting the buffer area of unnecessary data. If the data size to be loaded onto the buffer has reached a minimum playback size, playback of the buffered object starts in synchronism with the current DVD playback.

[0230] As a result of this synchronous playback, a layout shown in, e.g., FIG. 37 or 38 can be played back. In the example of FIG. 37, picture 964 of the DVD-Video contents is displayed within DVD-Video display area 962, and buttons 963 indicating chapters 1 to 5 of the DVD-Video are displayed with $\alpha=100\%$ within display area (movie menu) 961 of an advanced graphics object (corresponding to 351C in FIG. 8). In the example of FIG. 38, picture 964 of the DVD-Video contents is displayed within display area 961, and buttons 963 indicating chapters 1 to 5 are displayed within display area 962. The display methods shown in FIGS. 37 and 38 can be arbitrarily switched by a user's instruction (or a description of the Markup language or the like).

[0231] FIG. 39 is a schematic block diagram showing the arrangement of a streaming apparatus (network compatible disc player) according to the embodiment of the present invention.

[0232] Reference numeral 200 denotes a client; 201, a server; and 221, a network that connects the server 201 and client 200. Client 200 comprises moving picture playback engine 203, Vclick engine 202, disc device 230, user interface 240, network manager 208, and disc device manager 213. Reference numerals 204 to 206 denote devices included in the moving picture playback engine; 207, 209 to 212, and 214 to 218, devices included in the Vclick engine; and 219 and 220, devices included in server 201. Client 200 can play back moving picture data, and can display a document described in a markup language (e.g., HTML or the like), which are stored in disc device 230. Also, client 200 can display a document (e.g., HTML) on the network.

[0233] When meta data associated with moving picture data stored in client 200 is stored in server 201, client 200 can execute the following playback process using this meta data and the moving picture data in disc device 230. Server 201 sends media data M1 to client 200 via network 221 in response to a request from client 200. Client 200 processes the received media data in synchronism with playback of a moving picture to implement additional functions of hypermedia and the like (note that “synchronization” is not limited to a physically perfect match of timings but allows some timing error).

[0234] Moving picture playback engine 203 is used to play back moving picture data stored in disc device 230, and has devices 204, 205, and 206. Reference numeral 231 denotes a moving picture data recording medium (more specifically, a DVD, video CD, video tape, hard disc, semiconductor memory, or the like). Moving picture data recording medium 231 records digital and/or analog moving picture data. Meta data associated with moving picture data may be recorded on moving picture data recording medium 231 together with the moving picture data. Reference numeral 205 denotes a moving picture playback controller, which can control playback of video/audio/sub-picture data D1 from moving picture data recording medium 231 in accordance with a “control” signal output from interface handler 207 of Vclick engine 202.

[0235] More specifically, moving picture playback controller 205 can output a “trigger” signal indicating the playback status of video/audio/sub-picture data D1 to interface handler 207 in accordance with a “control” signal which is transmitted upon generation of an arbitrary event (e.g., a menu call or title jump based on a user instruction) from interface handler 207 in a moving picture playback mode. In this case (at a timing simultaneously with output of the trigger signal or an appropriate timing before or after that timing), moving picture playback controller 205 can output a “status” signal indicating property information (e.g., an audio language, sub-picture subtitle language, playback operation, playback position, various kinds of time information, disc contents, and the like set in the player) to interface handler 207. By exchanging these signals, a moving picture data read process can be started or stopped, and access to a desired location in moving picture data can be made.

[0236] AV decoder 206 has a function of decoding video data, audio data, and sub-picture data recorded on moving picture data recording medium 231, and outputting decoded video data (mixed data of the aforementioned video and sub-picture data) and audio data. Moving picture playback engine 203 can have the same functions as those of a playback engine of a normal DVD-Video player which is manufactured on the basis of the existing DVD-Video standard. That is, client 200 in FIG. 39 can play back video data, audio data, and the like with the MPEG2 program stream structure in the same manner as a normal DVD-Video player, thus allowing playback of existing DVD-Video discs (discs complying with the conventional DVD-Video standard) (to assure playback compatibility with existing DVD software).

[0237] Interface handler 207 makes interface control among modules such as moving picture playback engine 203, disc device manager 213, network manager 208, meta data manager 210, buffer manager 211, script interpreter 212, media decoder 216 (including meta data decoder 217), layout manager 215, AV renderer 218, and the like. Also, interface handler 207 receives an input event by a user operation (operation to an input device such as a mouse, touch panel, keyboard, or the like) from user interface 240, and transmits the event to an appropriate module.

[0238] Interface handler 207 has an access table parser that parses a Vclick access table (corresponding to VCA which will be described later with reference to FIG. 45), an information file parser that parses a Vclick information file (corresponding to VCI which will be described later with reference to FIG. 45), a property buffer that records property information managed by the Vclick engine, a system clock of the Vclick engine, a moving picture clock as a copy of moving picture clock 204 in the moving picture playback engine, and the like.

[0239] Network manager 208 has a function of acquiring a document (e.g., HTML), still picture data, audio data, and the like onto buffer 209 via the network, and controls the operation of Internet connection unit 222. When network manager 208 receives a connection/disconnection instruction to/from the network from interface handler 207 that has received a user operation or a request from meta data manager 210, it switches connection/disconnection of Internet connection unit 222. Upon establishing connection between server 201 and Internet connection unit 222 via the network, network manager 208 exchanges control data and media data (object meta data). Note that buffer 209 corresponds to buffer 105 in FIG. 10, and can be formed using a ring buffer to which a predetermined size is assigned.

[0240] Data to be transmitted from client 200 to server 201 include a session open request, session close request, media data (object meta data) transmission request, status information (OK, error, etc.), and the like. Also, status information of the client may be exchanged. On the other hand, data to be transmitted from server 201 to client 200 include media data (object meta data) and status information (OK, error, etc.).

[0241] Disc device manager 213 has a function of acquiring a document (e.g., HTML), still picture data, audio data, and the like onto buffer 209, and a function of transmitting video/audio/sub-picture data D1 to moving picture playback

engine 203. Disc device manager 213 executes a data transmission process in accordance with an instruction from meta data manager 210.

[0242] Buffer 209 temporarily stores media data M1 which is sent from server 201 via the network (via the network manager). Moving picture data recording medium 231 records media data M2 in some cases. In such case, media data M2 is stored in buffer 209 via the disc device manager. Note that media data includes Vclick data (object meta data), a document (e.g., HTML), and still picture data, moving picture data, and the like attached to the document.

[0243] When media data M2 is recorded on moving picture data recording medium 231, it may be read out from moving picture data recording medium 231 and stored in buffer 209 in advance prior to the start of playback of video/audio/sub-picture data D1. This is for the following reason: since media data M2 and video/audio/sub-picture data D1 have different data recording locations on moving picture data recording medium 231, if normal playback is made, a disc seek or the like occurs and seamless playback cannot be guaranteed. The above process can avoid such problem.

[0244] As described above, when media data M1 downloaded from server 201 is stored in buffer 209 as in media data M2 recorded on moving picture data recording medium 231, video/audio/sub-picture data D1 and media data can be simultaneously read out and played back.

[0245] Note that the storage capacity of buffer 209 is limited. That is, the data size of media data M1 or M2 that can be stored in buffer 209 is limited. For this reason, unnecessary data may be erased under the control (buffer control) of meta data manager 210 and/or buffer manager 211.

[0246] Meta data manager 210 manages meta data stored in buffer 209, and transfers meta data having a corresponding time stamp to media decoder 216 upon reception of an appropriate timing ("moving picture clock" signal) synchronized with playback of a moving picture from interface handler 207.

[0247] When meta data having a corresponding time stamp is not present in buffer 209, it need not be transferred to media decoder 216. Meta data manager 210 controls to load data for a size of the meta data output from buffer 209 or for an arbitrary size from server 201 or disc device 230 onto buffer 209. As a practical process, meta data manager 210 issues a meta data acquisition request for a designated size to network manager 208 or disc device manager 213 via interface handler 207. Network manager 208 or disc device manager 213 loads meta data for the designated size onto buffer 209, and sends a meta data acquisition completion response to meta data manager 210 via interface handler 207.

[0248] Buffer manager 211 manages data (a document (e.g., HTML), still picture data and moving picture data appended to the document, and the like) other than meta data stored in buffer 209, and sends data other than meta data stored in buffer 209 to parser 214 and media decoder 216 upon reception of an appropriate timing ("moving picture clock" signal) synchronized with playback of a moving picture from interface handler 207. Buffer manager 211 may delete data that becomes unnecessary from buffer 209.

[0249] Parser 214 parses a document written in a markup language (e.g., HTML), and sends a script to script interpreter 212 and information associated with a layout to layout manager 215.

[0250] Script interpreter 212 interprets and executes a script input from parser 214. Upon executing the script, information of an event and property input from interface handler 207 can also be used. When an object in a moving picture is designated by the user, a script is input from meta data decoder 217 to script interpreter 212.

[0251] AV renderer 218 has a function of controlling video/audio/text outputs. More specifically, AV renderer 218 controls, e.g., the video/text display positions and display sizes (often also including the display timing and display time together with them) and the level of audio (often also including the output timing and output time together with it) in accordance with a "layout control" signal output from layout manager 215, and executes pixel conversion of a video in accordance with the type of a designated monitor and/or the type of a video to be displayed. The video/audio/text outputs to be controlled are those from moving picture playback engine 203 and media decoder 216. Furthermore, AV renderer 218 has a function of controlling mixing or switching of video/audio data input from moving picture playback engine 203 and video/audio/text data input from the media decoder in accordance with an "AV output control" signal output from interface handler 207.

[0252] Layout manager 215 outputs a "layout control" signal to AV renderer 218. The "layout control" signal includes information associated with the sizes and positions of moving picture/still picture/text data to be output (often also including information associated with the display times such as display start/end timings and duration), and is used to designate AV renderer 218 about a layout used to display data. Layout manager 215 checks input information such as user's clicking or the like input from interface handler 207 to determine a designated object, and instructs meta data decoder 217 to extract an action command such as display of associated information which is defined for the designated object. The extracted action command is sent to and executed by script interpreter 212.

[0253] Media decoder 216 (including the meta data decoder) decodes moving picture/still picture/text data. These decoded video data and text image data are transmitted from media decoder 216 to AV renderer 218. These data to be decoded are decoded in accordance with an instruction of a "media control" signal from interface handler 207 and in synchronism with a "timing" signal from interface handler 207.

[0254] Reference numeral 219 denotes a meta data recording medium of server 201 such as a hard disc, optical disc, semiconductor memory, magnetic tape, or the like, which records meta data to be transmitted to client 200. This meta data is associated with moving picture data recorded on moving picture data recording medium 231. This meta data includes object meta data to be described later. Reference numeral 220 denotes a network manager of server 201, which exchanges data with client 200 via network 221.

(Overview of Data Structure and Access Table)

[0255] A Vclick stream includes data associated with regions of objects (e.g., persons, articles, and the like) that appear in the moving picture recorded on moving picture data recording medium 231, display methods of the objects

in client 200, and data of actions to be taken by the client when the user designates these objects. An overview of the structure of Vclick data and its elements will be explained below. Object region data as data associated with a region of an object (e.g., a person, article, or the like) that appears in the moving picture will be explained first.

[0256] FIG. 40 is a view for explaining the relationship between an object region and object region data according to the embodiment of the present invention. Reference numeral 300 denotes a locus of a region of one object on a three-dimensional (3D) coordinate system of X (the horizontal coordinate value of a video picture), Y (the vertical coordinate value of the video picture), and Z (the time of the video picture). An object region is converted into object region data for each predetermined time range (e.g., between 0.5 sec to 1.0 sec, between 2 sec to 5 sec, or the like). In FIG. 40, one object region 300 is converted into five object region data 301 to 305, which are stored in independent Vclick access units (AU: to be described later). As a conversion method at this time, for example, MPEG-4 shape encoding, an MPEG-7 spatio-temporal locator, or the like can be used. Since the MPEG-4 shape encoding and MPEG-7 spatio-temporal locator are schemes for reducing the data size by exploiting temporal correlation among object regions, they suffer problems: data cannot be decoded from halfway through, and if data at a given time is omitted, data at neighboring times cannot be decoded. Since the region of the object that continuously appears in the moving picture for a long period of time, as shown in FIG. 40, is converted into data by dividing it in the time direction, easy random access is allowed, and the influence of omission of partial data can be reduced. Each Vclick_AU is effective in only a specific time interval in a moving picture. A time interval in which a Vclick_AU is effective is called a lifetime of the Vclick_AU.

[0257] FIG. 41 shows an example of the data structure of the access unit of the object meta data according to the embodiment of the present invention. FIG. 41 expresses the structure of one unit (Vclick_AU), which can be accessed independently, in Vclick stream VCS used in the embodiment of the present invention. Reference numeral 400 denotes object region data. As has been explained using FIG. 40, the locus of one object region in a given continuous time interval is converted into data. The time interval in which the object region is described is called an active time of that Vclick_AU. Normally, the active time of a Vclick_AU is equal to the lifetime of that Vclick_AU. However, the active time of a Vclick_AU can be set as a part of the lifetime of that Vclick_AU.

[0258] Reference numeral 401 denotes a header of the Vclick_AU. Header 401 includes an ID used to identify the Vclick_AU, and data used to specify the data size of that AU. Reference numeral 402 denotes a time stamp which indicates that of the start of the lifetime of this Vclick_AU. Since the active time and lifetime of Vclick_AU are normally equal to each other, the time stamp also indicates a time of the moving picture corresponding to the object region described in object region data 400. As shown in FIG. 40, since the object region covers a certain time range, time stamp 402 normally describes the time of the head of the object region. Of course, the time stamp may describe the time interval or the time of the end of the object region described in the object region data. Reference numeral 403

denotes object attribute information, which includes, e.g., the name of an object, an action description upon designation of the object, a display attribute of the object, and the like. These data in the Vclick_AU will be described in detail later. The server (201 in FIG. 39 or the like) preferably records Vclick AUs in the order of time stamps so as to facilitate transmission.

[0259] FIG. 42 is a view for explaining an example of the data structure of the access unit of object meta data according to another embodiment of the present invention. FIG. 42 shows an example of the data structure of a Vclick_AU, which is different from FIG. 41. The difference from FIG. 41 is that data used to specify the lifetime of a Vclick_AU is a combination of time stamp B01 and endurance or duration B02 in place of the time stamp alone. Time stamp B01 is the start time of the lifetime of the Vclick_AU, and duration B02 is a duration from the start time to the end time of the lifetime of the Vclick_AU. The duration may have, for example, the following practical configuration. That is, "time_type" and "duration" are prepared as information of the duration. Note that "time_type" is an ID used to specify that data means a duration, and "duration" is a duration. "duration" indicates a duration using a predetermined unit (e.g., 1 msec, 0.1 sec, or the like).

[0260] An advantage offered when the duration is also described as data used to specify the Vclick_AU lies in that the duration of the Vclick_AU can be detected by checking only Vclick_AU to be processed. Therefore, for example, when valid Vclick_AUs with a given time stamp are to be found, it is checked without checking other Vclick_AU data if the Vclick_AU of interest is the one to be found. However, the data size increases by duration B02 compared to FIG. 41.

[0261] FIG. 43 is a view for explaining an example of the data structure of the access unit of object meta data according to still another embodiment of the present invention. FIG. 43 shows an example of the data structure of a Vclick_AU, which is different from FIG. 42. In this example, as data for specifying the lifetime of a Vclick_AU, time stamp C01 that specifies the start time of the lifetime of the Vclick_AU and time stamp C02 that specifies the end time are used. The advantage offered upon using this data structure is substantially the same as that upon using the data structure of FIG. 42.

[0262] FIG. 44 shows an example of Vclick access table VCA. This table is prepared in advance, and is recorded in server 201. This table can also be stored in the same file as Vclick information file VCI. Reference numeral 850 denotes a time stamp sequence, which lists time stamps of the moving picture. Reference numeral 851 denotes an access point sequence, which lists offset values from the head of Vclick stream VCS in correspondence with the time stamps of the moving picture. If a value corresponding to the time stamp of the random access destination of the moving picture is not stored in Vclick access table VCA, an access point of a time stamp with a value close to that time stamp is referred to, and a transmission start location is sought while referring to time stamps in Vclick stream VCS near that access point. Alternatively, Vclick access table VCA is searched for a time stamp of a time before that of the random access destination of the moving picture, and Vclick stream VCS is transmitted from an access point corresponding to the time stamp.

[0263] Server 201 stores Vclick access table VCA and uses it to search for Vclick data to be transmitted in response to random access from the client. However, Vclick access table VCA stored in server 201 may be downloaded to client 200, which may search the table for Vclick stream VCS. Especially, when Vclick streams VCS are simultaneously downloaded from server 201 to client 200, Vclick access tables VCA are also simultaneously downloaded from server 201 to client 200.

[0264] On the other hand, a moving picture recording medium such as a DVD or the like which records Vclick streams VCS may be provided. In this case as well, it is effective for client 200 to use Vclick access table VCA so as to search for data to be used in response to random access of playback contents. In such case, the Vclick access tables VCA are recorded on the moving picture recording medium as in Vclick streams VCS, and client 200 reads out and uses Vclick access table VCA of interest from the moving picture recording medium onto its internal main memory or the like. Random playback of Vclick streams VCS, which occurs upon random playback of a moving picture or the like, is processed by meta data decoder 217.

[0265] In Vclick access table VCA shown in FIG. 44, a time stamp "time" is time information which has a time stamp format of a moving picture recorded on the moving picture recording medium. For example, when the moving picture is compressed by MPEG-2 upon recording, "time" has an MPEG-2 PTS (Presentation Time Stamp) format. Furthermore, when the moving picture has a navigation structure of titles, program chains, and the like as in DVD, parameters (title numbers TTN, video title set numbers VTS_TTN, title program chain numbers TT_PGCN, part-of-title numbers PTTN, and the like) that express them are included in the format of "time".

[0266] Assume that some natural total order is defined for a set of time stamp values. For example, as for PTS, a natural order in terms of a time can be introduced. As for time stamps including DVD parameters, an order can be introduced according to a natural playback order of the DVD.

[0267] Each Vclick stream VCS satisfies the following conditions:

[0268] i) Vclick_AUs in Vclick stream VCS are arranged in ascending order of time stamp. At this time, the lifetime of each Vclick_AU is determined as follows. Let t be the time stamp value of a given AU. Time stamp values u of AUs after the given AU satisfy $u \geq t$ under the above condition. Let t' be a minimum one of such " u "s, which satisfies $u \neq t$. A period which has time t as the start time and t' as the end time is defined as the lifetime of the given AU. If there is no AU which has time stamp value u that satisfies $u > t$ after the given AU, the end time of the lifetime of the given AU matches the end time of the moving picture.

[0269] ii) The active time of each Vclick_AU corresponds to the time range of the object region described in the object region data included in that Vclick_AU.

[0270] Note that the following constraint associated with the active time is set for a Vclick stream VCS. That is, the active time of a Vclick_AU is included in the lifetime of that AU.

[0271] A Vclick stream VCS which satisfies the above constraints i) and ii) has the following good properties:

[0272] First, high-speed random access of Vclick stream VCS can be made, as will be described later. Second, a buffer process upon playing back Vclick stream VCS can be simplified. The buffer (209 in FIG. 39 or the like) stores Vclick stream VCS for respective Vclick_AUs, and erases AUs from those which have larger time stamps. If the two assumptions above are absent, a large buffer and complicated buffer management are required so as to hold effective AUs on the buffer. The following description will be given under the assumption that Vclick stream VCS satisfies the above two conditions i) and ii).

[0273] In Vclick access table VCA shown in FIG. 44, access point "offset" indicates a position on a Vclick stream VCS. For example, Vclick stream VCS is a file, and "offset" indicates a file pointer value of that file. The relationship of access point "offset", which forms a pair with time stamp "time", is as follows:

[0274] i) A position indicated by "offset" is the head position of a given Vclick_AU;

[0275] ii) A time stamp value of that AU is equal to or smaller than the value of "time"; and

[0276] iii) A time stamp value of an AU immediately before that AU is truly smaller than "time".

[0277] In Vclick access table VCA, "time"s may be arranged at arbitrary intervals but need not be arranged at equal intervals. However, they may be arranged at equal intervals in consideration of convenience for a search process and the like.

[0278] In the example of FIG. 44, a case wherein when time stamp 850 is time*, corresponding access point 851 is "NULL" is also exemplified. That is, FIG. 44 exemplifies Vclick access table VCA when a Null Pointer (one of file pointers fp) indicating "NULL" is used.

[0279] "NULL" in access point 851 in FIG. 44 is a flag which means that "the active time of an AU in Vclick stream VCS of interest has no intersection (or no relation) to a time range equal to or larger than time* and less than time#4". Assume that moving image clock T supplied from interface handler 207 to meta data manager 210 in FIG. 39 satisfies:

$$\text{time}^* \leq T < \text{time}^{\#4}$$

[0280] At this time, meta data manager 210 searches Vclick access table VCA in FIG. 44 for the "NULL" flag. When the "NULL" flag is obtained, meta data manager 210 ends its operation or starts the next operation without loading any Vclick stream VCS.

[0281] FIG. 45 is a view for explaining an example of the structure of an enhanced DVD-Video disc according to the embodiment of the present invention. FIG. 45 shows an example of the data structure when an enhanced DVD-Video disc is used as moving picture data recording medium 231. A DVD-Video area of the enhanced DVD-Video disc stores DVD-Video contents (having the MPEG-2 program stream structure) having the same data structure as the DVD-Video standard. Furthermore, another recording area of the enhanced DVD-Video disc stores enhanced navigation (to be abbreviated as ENAV hereinafter) contents which allow

various playback processes of video contents. Note that the "other recording area" is also recognized by the DVD-Video standard.

[0282] A basic data structure of the DVD-Video disc will be described below. The recording area of the DVD-Video disc includes a lead-in area, volume space, and lead-out area in turn from its inner periphery. The volume space includes a volume/file structure information area and DVD-Video area (DVD-Video zone), and can also have another recording area (DVD other zone) as an option.

[0283] The volume/file structure information area is assigned for the UDF (Universal Disk Format) bridge structure. The volume of the UDF bridge format is recognized according to ISO/IEC13346 Part 2. A space that recognizes this volume includes successive sectors, and starts from the first logical sector of the volume space in FIG. 45. First 16 logical sectors are reserved for system use specified by ISO9660. In order to assure compatibility to the conventional DVD-Video standard, the volume/file structure information area with such contents is required.

[0284] The DVD-Video area records management information called video manager VMG and one or more video contents called video title sets VTS (VTS#1 to VTS#n). The VMG is management information for all VTSs present in the DVD-Video area, and includes control data VMGI, VMG menu data VMGM_VOBS (option), and VMG backup data. Each VTS includes control data VTSI of that VTS, VTS menu data VTSM_VOBS (option), data VTSTT_VOBS of the contents (movie or the like) of that VTS (title), and VTSTI backup data. To assure compatibility to the conventional DVD-Video standard, the DVD-Video area with such contents is also required.

[0285] A playback select menu or the like of respective titles (VTS#1 to VTS#n) is given in advance by a provider (the producer of a DVD-Video disc) using the VMG, and a playback chapter select menu, the playback order of recorded contents (cells), and the like in a specific title (e.g., VTS#1) are given in advance by the provider using the VTSTI. Therefore, the viewer of the disc (the user of the DVD-Video player) can enjoy the recorded contents of that disc in accordance with menus of the VMG/VTSTI prepared in advance by the provider and playback control information (program chain information PGCI) in the VTSTI. However, with the DVD-Video standard, the viewer (user) cannot play back the contents (movie or music) of each VTS by a method different from the VMG/VTSTI prepared by the provider.

[0286] The enhanced DVD-Video disc shown in FIG. 45 is prepared for a scheme that allows the user to play back the contents (movie or music) of each VTS by a method different from the VMG/VTSTI prepared by the provider, and to play back while adding contents different from the VMG/VTSTI prepared by the provider. ENAV contents included in this disc cannot be accessed by a DVD-Video player which is manufactured on the basis of the conventional DVD-Video standard (even if the ENAV contents can be accessed, their contents cannot be used). However, a DVD-Video player according to the embodiment of the present invention (for example, client 200 which equips Vclick engine 202 in FIG. 39) can access the ENAV contents, and can use their playback contents.

[0287] The ENAV contents include data such as audio data, still picture data, font/text data, moving picture data, animation data, Vclick data, and the like, and also an ENAV document (described in a Markup/Script language) as information for controlling playback of these data. This playback control information describes, using a Markup language or Script language, playback methods (display method, playback order, playback switch sequence, selection of data to be played back, and the like) of the ENAV contents (including audio, still picture, font/text, moving picture, animation, Vclick, and the like) and/or the DVD-Video contents. For example, Markup languages such as HTML (Hyper Text Markup Language)/XHTML (extensible Hyper Text Markup Language), SMIL (Synchronized Multimedia Integration Language), Script languages such as an ECMA (European Computer Manufacturers Association) script, JavaScript®, and so forth, may be used in combination.

[0288] Since the contents of the enhanced DVD-Video disc in FIG. 45 except for the other recording area comply with the DVD-Video standard, video contents recorded on the DVD-Video area can be played back using an already prevalent DVD-Video player (i.e., this disc is compatible to the conventional DVD-Video disc). The ENAV contents recorded on the other recording area cannot be played back (or cannot be used) by the conventional DVD-Video player but can be played back and used by a DVD-Video player according to the embodiment of the present invention. Therefore, when the ENAV contents are played back using the DVD-Video player according to the embodiment of the present invention, the user can enjoy not only the contents of the VMG/VTSI prepared in advance by the provider but also a variety of video playback features.

[0289] Especially, as shown in FIG. 45, the ENAV contents include Vclick data VCD, which includes Vclick information file (Vclick Info) VCI, Vclick access table VCA, Vclick stream VCS, Vclick information file backup (Vclick Info backup) VCIB, and Vclick access table backup VCAB.

[0290] Vclick information file VCI is data indicating a portion of DVD-Video contents where Vclick stream VCS (to be described below) is appended (e.g., to the entire title, the entire chapter, a program chain, program, or cell as a part thereof, or the like of the DVD-Video contents). Vclick access table VCA is assured for each Vclick stream VCS (to be described below), and is used to access Vclick stream VCS. Vclick stream VCS includes data such as location information of an object in a moving picture, an action description to be made upon clicking the object, and the like. Vclick information file backup VCIB is a backup of the aforementioned Vclick information file VCI, and always has the same contents as Vclick information file VCI. Vclick access table backup VCAB is a backup of the above-mentioned Vclick access table VCA, and always has the same contents as Vclick access table VCA.

[0291] In the example of FIG. 45, Vclick data VCD is recorded on the enhanced DVD-Video disc. However, as described above, Vclick data VCD is stored in server 201 on the network in some cases. That is, Vclick data VCD can be prepared inside/outside the disc. When Vclick data VCD is prepared outside the disc, playback using Vclick data VCD can be made even in contents playback of an old type disc (a disc sold in the past or the like) that does not record any

Vclick data VCD or in playback of contents that record TV broadcasting (when Vclick data VCD are created in correspondence with these contents).

[0292] Furthermore, the user creates an original disc using a video recordable medium (e.g., a DVD-R disc, DVD-RW disc, DVD-RAM disc, hard disc, or the like) and a video recorder (e.g., a DVD-VR recorder, DVD-SR recorder, HD-DVD recorder, HDD recorder, or the like). In such case, if the user records ENAV contents including Vclick data VCD or prepares Vclick data VCD on a data storage of a personal computer other than this disc and connects this personal computer and recorder, he or she can enjoy meta data playback in the same manner as in the DVD-ROM video+the ENAV player in FIG. 39.

[0293] FIG. 46 shows an example of the directory configuration in the enhanced DVD-Video disc according to the embodiment of the present invention. Under the root directory, subdirectories "HVDVD_TS" and "ADV_OBJ" shown in FIG. 2 are allocated. FIG. 46 shows an example of files which form Vclick information file VCI, Vclick access table VCA, Vclick stream VCS, Vclick information file backup VCIB, and Vclick access table backup VCAB mentioned above. A file (VCKINDEX.IFO) that forms Vclick information file VCI is described in XML (extensible Markup Language), and describes Vclick streams VCS and the location information (VTS numbers, title numbers, PGC numbers, and the like) of the DVD-Video contents where the Vclick streams are appended. Vclick access table VCA is made up of one or more files (VCKSTR01.IFO to VCKSTR99.IFO or arbitrary file names), and one access table VCA file corresponds to one Vclick stream VCS.

[0294] A Vclick stream file describes the relationship between location information (a relative byte size from the head of the file) of each Vclick stream VCS and time information (a time stamp of a corresponding moving picture or relative time information from the head of the file), and allows to search for a playback start position corresponding to a given time.

[0295] Vclick stream VCS includes one or more files (VCKSTR01.VCK to VCKSTR99.VCK or arbitrary file names), and can be played back together with the appended DVD-Video contents with reference to the description of the aforementioned Vclick information file VCI. If there are a plurality of attributes (e.g., Japanese Vclick data VCD, English Vclick data VCD, and the like), different Vclick streams VCS (i.e., different files) may be formed in correspondence with different attributes. Alternatively, respective attributes may be multiplexed to form one Vclick stream VCS (i.e., one file).

[0296] In case of the former configuration (a plurality of Vclick streams VCS are formed in correspondence with different attributes), the occupied size of the buffer (e.g., 209 in the example of FIG. 39) upon temporarily storing Vclick data in the playback apparatus (player) can be reduced. In case of the latter configuration (one Vclick stream VCS is formed to include different attributes), one file can be kept played back without switching files upon switching attributes, thus assuring high switching speed.

[0297] Note that each Vclick stream VCS and Vclick access table VCA can be associated using, e.g., their file names. In the aforementioned example, one Vclick access

table VCA (VCKSTRXX.IFO; XX=01 to 99) is assigned to one Vclick stream VCS (VCKSTRXX.VCK; XX=01 to 99). Hence, by adopting the same file name except for extensions, association between each Vclick stream VCS and Vclick access table VCA can be identified.

[0298] In addition, Vclick information file VCI describes association between each Vclick stream VCS and Vclick access table VCA (more specifically, the VCI parallelly describes descriptions of VCS and those of VCA), thereby identifying association between each Vclick stream VCS and Vclick access table VCA.

[0299] Vclick information file backup VCIB is formed of a VCKINDEX.BUP file, and has the same contents as the aforementioned Vclick information file VCI (VCKINDEX.IFO). If VCKINDEX.IFO cannot be loaded for some reason (due to scratches, stains, and the like on the disc), desired procedures can be made by loading this VCKINDEX.BUP instead. Vclick access table backup VCAB is formed of VCKSTR01.BUP to VCKSTR99.BUP files, which have the same contents as the aforementioned Vclick access tables VCA (VCKSTR01.IFO to VCKSTR99.IFO). One Vclick access table backup VCAB (VCKSTRXX.BUP; XX=01 to 99) is assigned to one Vclick access table VCA (VCKSTRXX.IFO; XX=01 to 99), and the same file name is adopted except for extensions, thus identifying association between each Vclick access table VCA and Vclick access table backup VCAB. If VCKSTRXX.IFO cannot be loaded for some reason (due to scratches, stains, and the like on the disc), desired procedures can be made by loading this VCKSTRXX.BUP instead.

[0300] FIG. 47 shows the relationship between Vclick streams VCS described in the above description example of Vclick Info VCI, and the DVD-Video contents. In this example, the aforementioned fifth Vclick stream VCS (Vclick#5) and sixth Vclick stream VCS (Vclick#6) are appended to the first PGC (PGC#1) in the first VTS menu domain (VTS menu #1) in the first VTS space (VTS#1). This represents that two Vclick streams VCS (Vclick#5 and Vclick#6) are appended to the DVD-Video contents. These streams (Vclick#5 and Vclick#6) can be switched by, e.g., the user or contents provider (contents author).

[0301] When the user switches these streams, a "Vclick switch button" that can be used to switch Vclick streams VCS is provided to a remote controller attached to the apparatus shown in FIG. 39. With this button, the user can freely change two or more Vclick streams. Although not shown, this remote controller has the "Vclick switch button" in addition to buttons of a remote controller of a general DVD-Video player, and upon depression of this button, the player enters a Vclick stream switch mode. In this mode, when the user clicks the "Vclick switch button" or presses the up and down or right and left cursor keys on the remote controller (not shown), he or she can sequentially switch designation of the stream number of a Vclick stream. Alternatively, a method of directly designating the stream number of Vclick stream VCS using a ten-key pad of the remote controller (not shown) in this mode may be used.

[0302] On the other hand, when the contents provider changes Vclick streams VCS, a Vclick switching command (a description format is, e.g., "changeVclick()") is described in a Markup language, and this switch command is issued at

a timing designated by the contents provider in the Markup language, thus freely changing two or more Vclick streams VCS.

[0303] FIG. 48 is a view for explaining another configuration example of Vclick information according to the embodiment of the present invention. FIG. 48 shows the relationship between the PGC data of the DVD-Video contents and Vclick streams to be appended to their attributes. In the example of FIG. 48, Vclick streams VCS are roughly assigned to respective PGC data, and the assignment method is segmented in accordance with the attributes and the like of respective PGC data.

[0304] More specifically, streams Vclick#1 to Vclick#3 are assigned to whole PGC#1. This example can be configured as follows. That is, stream Vclick#1 is, e.g., an English page, stream Vclick#2 is, e.g., a Japanese page, and stream Vclick#3 is, e.g., a Chinese page, so that these streams can be appropriately switched and selected (the configuration that selects a stream of meta data in accordance with the PGC playback period of the video contents).

[0305] In PGC#2 of FIG. 48, stream Vclick#1 is assigned to its audio #1, and stream Vclick#2 is assigned to its audio #2. In PGC#3, stream Vclick#1 is assigned to its sub-picture #1 (sub-picture such as a subtitle or the like), stream Vclick#2 is assigned to its sub-picture #2, and stream Vclick#3 is assigned to its sub-picture #3. In PGC#4, stream Vclick#1 is assigned to its angle #1, no Vclick stream is assigned to its angle #2, and stream Vclick#2 is assigned to its angle #3. In PGC#5, stream Vclick#1 is assigned in case of "wide" at the display aspect ratio 16:9, stream Vclick#2 is assigned in case of "pan scan" at the display aspect ratio 4:3, and stream Vclick#3 is assigned in case of "letter box" at the display aspect ratio 4:3. In PGC#6, stream Vclick#4 is assigned when the display aspect ratio is 4:3 as normal one.

[0306] In PGC#7 in FIG. 48, stream Vclick#1 (e.g., English page) is assigned to sub-picture #1 (e.g., English subtitle) which links to audio #1 (e.g., English audio) of angle #1, and stream Vclick#2 (e.g., Japanese page) is assigned to sub-picture #2 (e.g., Japanese subtitle) which links to audio #1 (e.g., English audio) of angle #1. Also, no Vclick stream is assigned to sub-picture #1 (English subtitle) that links to audio #2 (e.g., Japanese audio) of angle #1, and stream Vclick#4 (e.g., another Japanese page) is assigned to sub-picture #2 (Japanese subtitle) which links to audio #2 (Japanese audio) of angle #1. Furthermore, stream Vclick#3 (e.g., Chinese page) is assigned to sub-picture #1 (English subtitle) and sub-picture #2 (Japanese subtitle) which link to audio #1 (English audio) of angle #2, and stream Vclick#3 (e.g., Chinese page) is assigned to sub-picture #1 (English subtitle) and sub-picture #2 (Japanese subtitle) which link to audio #2 (Japanese audio) of angle #2. Of these streams, stream Vclick#4 (another Japanese page) is further assigned to sub-picture #2 (Japanese subtitle) which links to audio #2 (Japanese audio) of angle #2, in addition to stream Vclick#3 (Chinese page). For sub-picture #2 (Japanese subtitle) which links to audio #2 (Japanese audio) of angle #2, stream Vclick#3 (Chinese page) or Vclick#4 (another Japanese page) can be selected.

[0307] When an object to be synchronously played back with Vclick streams is DVD-Video contents, Vclick streams can be switched for respective titles (VTS) of DVD-Video as

a largest unit, and can be switched for respective parts-of-title (chapters) as a smaller unit. Also, Vclick streams can be switched for respective program chains (PGC) as a still smaller unit, for respective programs (PG) as a yet smaller unit, or for respective cells as a smallest unit.

[0308] When Vclick streams according to the embodiment of the present invention are applied to a recording/playback system such as a DVD-VR recorder, DVD-SR recorder, HD-DVD recorder, or the like, Vclick streams may be switched for respective user-defined PGC data (playlists) or respective entry points marked locally in programs that form PGC data.

[0309] The playback apparatus (enhanced DVD player) according to the embodiment of the present invention can sequentially change Vclick streams to be appended in correspondence with the playback state of the DVD-Video contents by loading Vclick information file VCI in advance or referring to that file as needed, prior to playback of the DVD-Video contents. In this manner, a high degree of freedom can be assured upon forming Vclick streams, and the load on authoring can be reduced.

[0310] By increasing the number of files (the number of streams) of unitary Vclick contents, and decreasing each file size, an area (buffer 209 in the apparatus of FIG. 39) required for the playback apparatus to store Vclick streams VCS can be reduced.

[0311] By decreasing the number of files (i.e., forming one stream to include a plurality of Vclick data) although the file size increases, Vclick data can be switched smoothly when the playback state of the DVD-Video contents has changed (since the information size of buffered Vclick data is large).

[0312] Note that the configuration of FIG. 48 is not exclusive to that of FIG. 35, and both the configurations can be used together as needed.

[0313] FIG. 49 is a flowchart for explaining an example of the information recording method using information recording medium 1 in FIG. 1. The contents provider or the like prepares expanded video objects (VTSTT_EVOBS) and advanced objects (VTSTT_AGOBS/VTSTT_ATOBS) as objects to be recorded (step ST40). Information (Flash object 351/timed text 354/streaming object 355, etc.) shown in, e.g., FIG. 8 is prepared as information of playback sequence PSQ (step ST42). After that, the prepared objects to be recorded are recorded on the object area (video title set recording area 40), and the prepared information of the playback sequence is recorded on the management area (video manager recording area 30) (step ST44).

[0314] FIG. 50 shows a list of tags which can be used in the playback sequence (PSQ) according to the embodiment of the present invention. In the description of this list, tag names are listed on the left side, and their meanings are listed on the right side as follows.

<Tag name: meaning of tag>

[0315] video_pbseq: start of a description of playback sequence data

[0316] vmg: corresponding to the VMG space of DVD-Video

[0317] vmgm: corresponding to the VMG menu domain of DVD-Video

[0318] fp: corresponding to the first play domain of DVD-Video

[0319] vts: corresponding to the VTS space of DVD-Video

[0320] vtsm: corresponding to the VTS menu domain of DVD-Video

[0321] vts_tt: corresponding to the VTS title domain of DVD-Video

[0322] idle: used when other objects are displayed without displaying any DVD-Video objects

[0323] pgc: corresponding to a PGC of DVD-Video (except for idle)

[0324] object: used when object data such as a DVD-Video object or the like is called (played back)

[0325] version: version number of playback sequence data

[0326] update: URI information of substitute playback sequence data (normally enabled when substitute data is recorded on the external server and data has a version newer than the version number of playback sequence data recorded on the disc)

[0327] region: a region code of the disc. Normally, the playback sequence recorded on the external server has this tag, and can change the region code set on the disc itself by overwriting.

[0328] FIG. 51 shows a list of attributes which can be used in the playback sequence (PSQ) according to the embodiment of the present invention. In the description of this list, attribute names, tag names that use attributes, applications of attributes, and meanings of attributes are listed in turn from the left side. Especially, applications are categorized into two, i.e., "condition" and "execution". When an attribute of "condition" is satisfied, an attribute of "execution" is played back. For example, for a description:

```
<object data="file://dvdrom/adv_obj/flash.swf" audio="1"
angle="2" set_subpic="0"/>
```

attributes serving as conditions are "audio" and "angle", and attributes to be executed are "data" and "set_subpic".

[0329] This example indicates that when an audio stream number=1 and an angle number=2, an object described in "data" is played back, and the number of a sub-picture stream is set to be zero (i.e., non-display). A practical example of the aforementioned list of attributes is as follows.

<Attribute Name: Tag that Uses Attribute: Application: Meaning of Attribute>

[0330] num: vmgm, vts, vtsm, vts_tt, pgc: condition: number

[0331] lang: vmgm: condition: language code (ISO639)

[0332] data: pgc: execution: URI of object data to be played back

[0333] start_ptm: data: condition: a playback start time (relative time using a PTM description) of an object

[0334] end_ptm: data: condition: a playback end time (relative time in a PTM description) of an object

[0335] start: data: condition: a playback start time (relative time using an HH:MM:SS:FF description) of an object

[0336] end: data: condition: a playback end time (relative time using an HH:MM:SS:FF description) of an object

[0337] audio: data: condition: an audio stream number of DVD-Video

[0338] subpic: data: condition: a sub-picture stream number of DVD-Video

[0339] angle: data: condition: an angle number of DVD-Video

[0340] gprm: data: condition: a general parameter value of DVD-Video

[0341] sprm: data: condition: a system parameter value of DVD-Video

[0342] priority: data: execution: priority upon superimpose display or priority of a user input process

[0343] alpha: data: execution: an alpha value upon alpha blending

[0344] position: data: execution: x- and y-coordinate positions for displaying an object

[0345] height: data: execution: a height after scaling upon scaling an object

[0346] width (or length): data: execution: a length after scaling upon scaling an object

[0347] style: data: execution: display information of an object

[0348] tmap_data: data: execution: URI of time map data (a time-to-address conversion table) of an object

[0349] cont: data: execution: continuation (without reset) of clocks from the previous PGC for an object other than a DVD-Video object when it is "yes"

[0350] update_data: data: execution: URI information of substitute object data (normally enabled when substitute data is recorded on the external server, and new data is determined by comparing the version and the like of data recorded on the disc)

[0351] update_tmap: data: execution: execute substitution as substitute time map data when substitute object data described in "update_data" is played back

[0352] set_audio: data: execution: set an audio stream to the designated number (mute, i.e., silent when it is zero) when substitute object data described in "update_data" is played back

[0353] set_subpic: data: execution: set a sub-picture stream to the designated number (non-display when it is zero) when substitute object data described in "update_data" is played back

[0354] set_angle: data: execution: set an angle to the designated number when substitute object data described in "update_data" is played back

[0355] set_gprm: data: execution: set a general parameter to the designated value when substitute object data described in "update_data" is played back

[0356] set_sprm: data: execution: set a system parameter to the designated value when substitute object data described in "update_data" is played back

[0357] program: data: execution: normally play back a program of the designated value for an audio streaming object which includes a plurality of programs

[0358] dvd_mixlev: data: execution: normally designate a DVD mixing level for an audio streaming object

[0359] audio_mixlev: data: execution: normally designate a mixing level of an audio streaming object for an audio streaming object

[0360] meta_priority: data: execution: normally designate one of 1. use a mixing level (dvd/audio_mixlev) designated by the playback sequence, 2. use a mixing level included in an audio streaming object, and 3. use these two mixing levels together, for an audio streaming object

[0361] chromakey: data: execution: normally designate one chromakey value for a Flash object (the color of the designated value is not displayed as an original color, and the color of another superposed object is displayed, i.e., the designated color becomes transparent)

[0362] scaling_position: data: execution: change an object to the designated position when a display mode switch instruction is issued

[0363] scaling_height: data: execution: change an object to the designated height when a display mode switch instruction is issued

[0364] scaling_width: data: execution: change an object to the designated width when a display mode switch instruction is issued

[0365] FIG. 52 shows the system block arrangement according to another embodiment of the present invention. Note that circuit blocks denoted by the same reference numerals in the block arrangements shown in FIGS. 10 and 52 are configured to have equivalent functions. In the arrangement of FIG. 52, a playback sequence (PSQ) read out from information recording medium (DVD disc) 1 or designated external server 110 is input to playback sequence manager 123X, and is saved in PSQ recording area 123P in the playback sequence manager. Manager 123X parses "playback conditions (playback timings, display positions, display sizes, and the like) of objects other than the expanded video objects of the DVD-Video" and "playback conditions (display positions, display sizes, and the like) of expanded video objects of the DVD-Video" described in the playback sequence (PSQ), thus making the playback control according to the playback conditions.

[0366] On the other hand, DVD-Video navigation information read out from information recording medium 1 is parsed and processed by DVD-Video playback engine 125 as in FIG. 10. Since playback time information of the DVD-Video objects is also required upon playing back objects other than the DVD-Video objects, it is sequentially sent to playback sequence manager 123X and is used in playback control of respective objects.

[0367] Flash objects, timed text objects, and the like read out from information recording medium 1 or designated external server 110 are processed in the same manner as in FIG. 10. In this manner, synchronous playback of the

DVD-Video objects and these objects (Flash objects, timed text objects, and the like) can be implemented. Note that the object data are stored in buffer 105 for each group. Alternatively, the object data may be stored for a plurality of groups or for an information recording medium depending on the size of buffer 105.

[0368] The Flash objects are parsed and decoded by Flash playback engine 127. At this time, Flash playback engine 127 acquires time information from DVD-Video playback engine 125 as needed for synchronous playback of the Flash objects and DVD-Video objects, so as to achieve synchronization with DVD-Video playback engine 125. Flash playback engine 127 parses a user input as in FIG. 10. Then, engine 127 sends a command to interface handler 124 as an action corresponding to "that user input" set in each Flash object in advance. As this command, a command for controlling playback of the DVD-Video, a command for controlling playback of the timed text objects and streaming objects, a command for changing the attributes of the timed text objects and streaming objects, and the like are available as in FIG. 10.

[0369] The Flash object can control the contents of playback sequence PSQ which is being currently executed. Flash playback engine 127 is configured to substitute whole playback sequence PSQ recorded on PSQ recording area 123P of playback sequence manager 123X, to add or erase a description to or from a part of the playback sequence, and so forth. In this way, the data of playback sequence PSQ can be changed as needed, i.e., the playback order and the like can be dynamically changed according to the situation.

[0370] Interface handler 124 transfers commands sent from Flash playback engine 127 to respective engines (125, 128, 129) as in FIG. 10. Flash playback engine 127 can write and read user information to or from user information storage area (which is assured on, e.g., a nonvolatile memory, hard disc drive, or the like) 126X in the playback apparatus. The user information includes user's personal information, access history, game score, playback sequence data changed by the user input, and the like.

[0371] Such data can be saved in another medium (e.g., memory card 109 or the like) other than information recording medium 1, and can be used in another playback apparatus. As another example, authentication data (personal information, time information, playback availability information, and the like) are recorded on memory card 109, and a Flash object is set to refer to the authentication data recorded on this memory card 109 when Flash player engine 127 plays back the information recording medium, thus putting viewing control on a specific individual, specific time, and the like.

[0372] The timed text objects are parsed and decoded by timed text playback engine 128 as in FIG. 10. Each timed text object describes information such as text information to be displayed, a font name (font data name) used in display, a font size to be displayed, a font color to be displayed, display effects, and the like, and is rendered using corresponding font data according to these pieces of information. Also, streaming objects are parsed and decoded by streaming playback engine 129.

[0373] Layout engine 130 scales (enlarges/reduces in scale) decoded video object data sent from the respective engines (125, 127 to 129) in accordance with the designation of playback sequence manager 123X as in FIG. 10. Furthermore, layout engine 130 forms a screen layout based on a layout designated from playback sequence manager 123X, and applies RGB mixing with transparency α to respective objects in accordance with an α value (a value indicating the transparency or contrast in %) designated by playback sequence manager 123X to composite pictures, thus generating an output picture. Layout engine 130 changes the output levels of respective audio object data, and mixes respective audio object data.

[0374] Note that the Flash objects and timed text objects may be downloaded from an external server (e.g., 201 in FIG. 39) via another medium other than information recording medium 1, e.g., via memory card 109, or Internet (Web) 110 connection onto buffer 105 when they are used, in the same manner as in FIG. 10.

[0375] An example of updating an additional object on the basis of update information described in the playback sequence (PSQ) will be described below. FIG. 53 shows an example of a playback sequence (PSQ) which includes information used to update the playback sequence, and information used to update an object. In this example, the playback sequence includes version information specified by a <version> tag, and update information specified by an <update> tag (964 in FIG. 53). The version information specified by the <version> tag indicates the version of this playback sequence itself.

[0376] In this example, this playback sequence has version 1.0. The update information specified by the <update> tag includes a data attribute, and can be configured to download new playback sequence information used to entirely substitute this playback sequence from a URI described in the data attribute. With this configuration, the playback apparatus shown in FIG. 52 automatically establishes connection to external server 110, compares the version information of the playback sequence recorded on information recording medium 1 with that of a playback sequence on external server 110, and can download newer data (data with newer version). In this manner, when the information of the playback sequence becomes old, new playback sequence information can be acquired. Also, when the living playback sequence suffers errors (bugs), playback can be made based on the latest playback sequence free from such errors.

[0377] An <object> tag used to describe an object (965 in FIG. 53) includes a data attribute (URI of the object) used to describe object information, an update_data attribute (URI of update information of the object), a tmap attribute (URI of a time map) used to describe time map information, and an update_tmap attribute (URI of update information of the time map) used to describe update information of the time map.

[0378] Note that the time map is a table (or equivalent one) required to convert time information into location information in a file. Using this time map table, if an arbitrary time is given, a corresponding file pointer can be acquired. With this time map, random access (fastforwarding, rewinding, jump to an arbitrary location) of an object can be easily implemented.

[0379] The update information of an object is used to substitute the object itself. The update information of an object can download, from external server 110, a new object which is prepared to substitute an original object recorded on information medium 1. Also, an original object which records an English subtitle, audio, and the like can be substituted by an object which is recorded on the external server and records a Japanese subtitle, audio, and the like.

[0380] FIG. 54 is a flowchart for explaining an example of the processing sequence upon acquiring a new playback sequence on the basis of the version information (information specified by the <version> tag in FIG. 53) and update information (information specified by the <update> tag in FIG. 53). Playback sequence PSQ is loaded from, e.g., information recording medium (corresponding to the disc in FIG. 1 and/or FIG. 45) 1 in FIG. 52 (step ST100). It is checked if the apparatus (FIG. 52) which executes the process in FIG. 54 is connected to the network (external server 110) or if the server as a destination is active (step ST102). If the apparatus is not connected to the network or if the destination is inactive even when it is connected (NO in step ST102), playback sequence PSQ loaded from medium 1 is recorded on PSQ recording area 123P in FIG. 52 (step ST112) and, for example, synchronous playback of DVD-Video objects and Flash objects is executed using this PSQ.

[0381] On the other hand, if the apparatus is connected to the network and the destination is active (YES in step ST102), a playback sequence on external server 110, which is described in the <update> tag in the playback sequence in FIG. 53, is loaded (step ST104). Next, the version of the playback sequence loaded in step ST100 is compared to that of the playback sequence loaded in step ST104 (step ST106).

[0382] If the two versions are the same or if the version of the playback sequence loaded from medium 1 is newer as a result of comparison, playback sequence PSQ loaded from medium 1 is recorded on PSQ recording area 123P (step ST112) and, for example, synchronous playback of DVD-Video objects and Flash objects is executed using this PSQ.

[0383] On the other hand, if the version of the playback sequence loaded from external server 110 is newer, playback sequence PSQ loaded from external server 110 is recorded on PSQ recording area 123P (step ST110) and, for example, synchronous playback of DVD-Video objects and Flash objects is executed using this PSQ.

[0384] According to the process in FIG. 54, even if disc 1 in use gets old, the user can use the playback sequence of the latest version upon making synchronous playback of DVD-Video objects and Flash objects.

[0385] The method of updating the playback sequence recorded on the information recording medium to that on the external server has been described. Conversely, a method of updating an object while the playback sequence remains the same is also available. That is, data is updated for each object by appending update information (in advance) to each individual object described in the playback sequence without updating the playback sequence itself.

[0386] FIG. 55 is a flowchart for explaining an example of the processing sequence upon acquiring an object and time map, in association with the method of updating an object while the playback sequence remains the same.

[0387] If a title playback instruction of medium 1 is issued, the playback sequence recorded on playback sequence recording area 123P (FIG. 52) is referred to (step ST200). If an object described in the playback sequence is to be played back (YES in step ST202), it is checked if playback sequence PSQ includes a description of update data (e.g. update_data in FIG. 53) of the corresponding object. If the description of update data is included (YES in step ST204), it is checked if the apparatus (FIG. 52) which executes the process in FIG. 55 is connected to the network (external server 110) or if the server as a destination is active (step ST206). If the apparatus is connected to the network and the destination is active (YES in step ST206), the time stamps and file sizes of two objects, i.e., an object from external server 110 and the object to be updated, are compared (step ST208). If at least either of these time stamps and file sizes are different, it is determined that these two objects are different.

[0388] If these two objects are different (YES in step ST210), the versions of these two objects are compared (step ST212). In this case, an object with a larger version value is newer (step ST214). If the two objects whose versions are compared have the same version or if the object on medium (disc) 1 has a larger version value (i.e., it is newer), the object (an advanced object or the like other than DVD-Video objects) on medium 1 is recorded on buffer 105 (step ST218). On the other hand, if the object on external server 110 has a larger version value (i.e., it is newer), the object (an advanced object or the like other than DVD-Video objects) on server 110 is recorded on buffer 105 (step ST216).

[0389] If the playback sequence referred to in step ST200 has no description of update data of the corresponding object (NO in step ST204) or if the apparatus is not connected to the network if this description is included or if the destination is inactive if the apparatus is connected (NO in step ST206), the control skips the processes in steps ST208 to ST214, and enters the process in step ST218. On the other hand, even if the apparatus is connected to the network and the destination is active (YES in step ST206), if the two objects are the same (NO in step ST210), it is determined that the object is not updated in practice, and the control enters the process in step ST218.

[0390] Next, it is checked if the playback sequence referred to in step ST200 includes a description of a time map ("tmap" in the PSQ description of FIG. 53) of the corresponding object (step ST222). If this description of the time map is not included (NO in step ST222), the flow returns to the process in step ST202. If this description of the time map is included (YES in step ST222), it is checked if the playback sequence referred to in step ST200 includes a description of a update data ("update_tmap" in the PSQ description of FIG. 53) of the time map (step ST224).

[0391] If the description of update data of the corresponding time map is included (YES in step ST224), it is checked if the apparatus (FIG. 52) which executes the process in FIG. 55 is connected to the network (external server 110) or if the server as a destination is active (step ST226). If the apparatus is connected to the network and the destination is active (YES in step ST226), the time stamps and file sizes of two objects, i.e., an object from external server 110 and the object to be updated, are compared (step ST228). If at

least either of these time stamps and file sizes are different, it is determined that the time maps of these two objects are different.

[0392] If the time maps of these two objects are different (YES in step ST230), the versions of these time maps are compared (step ST232). In this case, the time map with a larger version value is newer (step ST234). If the time maps whose versions are compared have the same version or if the time map of the object on medium (disc) 1 has a larger version number (i.e., it is newer), the time map of the object on medium 1 is recorded on buffer 105 (step ST238). On the other hand, if the time map of the object on external server 110 has a larger version number (i.e., it is newer), the time map of the object on external server 110 is recorded on buffer 105 (step ST236). After that, the flow returns to the process in step S202.

[0393] If an object described in the playback sequence is not played back (NO in step ST202), synchronous playback of objects on the DVD-video (medium 1 in FIG. 52) and buffer (105 in FIG. 52), which are ready to be played back at that time, is performed.

[0394] If the playback sequence referred to in step ST200 has no description of update data of the corresponding time map (NO in step ST224) or if the apparatus is not connected to the network if this description is included or if the destination is inactive if the apparatus is connected (NO in step ST226), the control skips the processes in steps ST228 to ST234, and enters the process in step ST238. On the other hand, even if the apparatus is connected to the network and the destination is active (YES in step ST226), if the two time maps are the same (NO in step ST230), it is determined that the time map is not updated in practice, and the control enters the process in step ST238.

[0395] FIG. 56 shows an example when an object is selected or updated using a Flash object. In 966 and 967 in FIG. 53, an <object> tag and param attribute are used in descriptions of objects. The param attribute means an arbitrary variable. This example means that when the value of the variable is "0", an object on information recording medium 1, which is described first, is played back; when the value of the variable is "1", an object on external server 110, which is described second, is played back.

[0396] For example, as shown in FIG. 56(a), a language selection menu is formed using a Flash object. In this example, the Flash object includes a button used to play back an English object on information recording medium 1 and a button used to play back a Japanese object on external server 110. In this case, the Flash object is set as follows: upon clicking the English button, the variable value is set to be "0" (see FIG. 56(b)); upon clicking the Japanese button, the variable value is set to be "1" (see FIG. 56(c)). In accordance with this variable set value, the playback apparatus (FIG. 52 or the like) plays back one (FIG. 56(b) or 56(c)) of objects described in the playback sequence.

[0397] In the above example, an object to be played back is selected using the param attribute. Alternatively, the same effect can be realized by referring to a GPRM (General Parameter) value using a gprm attribute. The GPRM value can be set not only by the Flash object but by a DVD-Video object. For this reason, an object to be played back can be arbitrarily set in accordance with the playback condition of the DVD-Video contents.

[0398] FIG. 57 is a view for explaining an example of a playback sequence before change, a playback sequence which is changed using a script, and the script used to change the playback sequence. In this example, in the arrangement shown in, e.g., FIG. 52, the contents of a playback sequence (PSQ) temporarily loaded from information recording medium 1 and/or external server 110 are partially changed using a script of a Flash object in accordance with the situation. The playback sequence is described in XML, as described above, and playback sequence manager 123X internally forms an XML tree of the playback sequence so as to parse that sequence. In this example, branches and leaves of the formed tree undergo addition, erasure, substitution, and the like as needed using a script language such as a Flash object or the like.

[0399] That is, in FIG. 57(a), a playback sequence before change plays back the first title of a DVD-Video object together with a Flash object, and then plays back the second title of the DVD-Video object together with a Flash object on information recording medium 1. However, assume that since the Flash object executes a script in FIG. 57(b) during playback of the first title, the contents of the playback sequence are changed upon completion of playback of the first title, so that the second title of the DVD-Video object is to be played back together with a new Flash object on external server 110. In the playback sequence of this example, each object is configured to use an <object> tag together with an id attribute, and to have a <uri> tag below the <object> tag. The id attribute is identification information (ID) of this object, and the <uri> tag indicates the URI of the location of that object. In this example, the script of the Flash object searches the original playback sequence for this <uri> tag, and substitutes it by a <uri> tag with new contents.

[0400] Note that the script in FIG. 57(b) assures a work playback sequence data area by `myXML=newXML()`, and loads a playback sequence file (PlaybackSequence.xml) onto this area by `myXML.load("PlaybackSequence.xml")`. Furthermore, the script creates a "uri" element having contents "`http://xx.xx/new_menu.swf`" by `set newUri=myXML.createElement("Uri")`, `set newtext=myXML.createTextNode("http://xx.xx/new_menu.swf")` and `newUri.appendChild(newtext)`. Next, the script searches the playback sequence for an object having an ID "Flash_VTS1_VTSTT1_2" by `target=myXML.getElementById("Flash_VTS1_VTSTT1_2")`, erases its child element (<uri> element) by `target.removeChild(firstChild)`, and appends the newly created <uri> element by `target.appendChild(newUri)`. This erase and append manipulations may be replaced using "`target.replaceChild(newUri, firstChild)`".

[0401] In the above example, a <uri> element as a child element of an <object> element is defined for that <object> element. Alternatively, a uri attribute as an attribute of an <object> element may be defined for that <object> element, thus obtaining the same effect.

[0402] FIG. 58 is a flowchart for explaining an example of the processing sequence upon updating the contents of a playback sequence as needed using a script of a Flash object. In the system arrangement in FIG. 52, a playback sequence (PSQ) is loaded from medium 1 and/or external server 110, and is recorded on PSQ recording area 123P (step ST300). In this case, required objects are loaded from medium 1

and/or external server **110** onto buffer **105**, and the playback sequence (PSQ) is executed by reading out the objects in the buffer as needed (step **ST300**). In this way, synchronous playback of DVD-Video objects, Flash objects, and the like is made in a program chain (PGC) period unit (step **ST302**). This synchronous playback in the PGC period unit is continued before a PSQ change instruction is issued by the Flash object (NO in step **ST304**).

[**0403**] If a PSQ change instruction is issued by the Flash object (YES in step **ST304**), the PSQ is to be fully or partially changed. More specifically, the playback sequence (PSQ) is loaded from PSQ recording area **123P** onto a work memory (not shown) (a work memory area can be assured in a RAM for a control MPU (not shown) in playback sequence manager **123X**) (step **ST306**). Next, the playback sequence is changed in accordance with an instruction (e.g., an instruction upon execution of the script in **FIG. 57(b)**) from the Flash object (step **ST308**). Before playback of the PGC period is completed (NO in step **ST310**), the processes in steps **ST302** to **ST308** are repeated. On the other hand, if playback of the PGC period is complete (YES in step **ST310**), the updated playback sequence on the work memory (not shown) is written on PSQ recording area **123P** (step **ST312**), and the updated sequence is executed.

[**0404**] **FIG. 59** generally exemplifies the reference relationship and the like among the playback sequence, Flash object, and DVD-Video object. When such reference relationship is built, if information of playback sequence PSQ becomes old, new playback sequence information can be acquired as needed (=change of PSQ). On the other hand, if the living playback sequence suffers errors (bugs) or the like, playback can be made based on the latest playback sequence free from any errors (bug-fixed PSQ or the like).

[**0405**] As described above, according to various embodiments of the present invention, since a Flash object, which can display buttons with a still picture image, background audio, or small animation and allows highlight display at arbitrary positions to have arbitrary sizes on the screen, and a timed text object can be set in addition to the conventional video contents, the degree of freedom of the contents provider can be improved, thus providing more colorful contents to the user.

[**0406**] Points of the embodiments of the present invention will be summarized below.

[**0407**] (1) Of objects stored in the disc, only minimum required objects (video, audio, and sub-picture in case of DVD ROM Video) are multiplexed using an MPEG program stream. Objects which can be superimposed later are recorded and stored on the disc as independent objects. Each independent object is superimposed as needed on an object obtained from the MPEG program stream by a blend (that provides transparency α (contrast corresponding to α %) to RGB mixing).

[**0408**] (2) With the concept of (1), a high-definition subtitle is implemented by a timed text object in addition to a (conventional, low-resolution) subtitle using a sub-picture object stored in the disc.

[**0409**] (3) With the concept of (1), colorful graphics such as a menu and the like can be implemented by a Flash object which can superimpose and highlight buttons on a video object stored in the disc by α blend.

[**0410**] (4) In addition to the concept of (1), a common application interface (API) that can control the playback states of additional objects such as a timed text object, Flash object, and the like is introduced to implement playback control of all objects stored in the disc and objects recorded on the external server.

[**0411**] (5) The playback sequence (PSQ) explained with reference to **FIG. 8** and the like is present independently of DVD-Video. This PSQ has contents that can manage the playback timings of all objects other than DVD-Video, and can specify display layouts of all these objects. Since one PSQ can manage one or more VTSSs, this invention can be practiced as long as one disc includes at least one PSQ (one disc may include a plurality of PSQs).

[**0412**] (6) The playback sequence (PSQ) can have information (update_data in **FIG. 53**) used to change this playback sequence, and also information (param="0" or param="1" in **FIG. 53**) used to change an object to be played back. The contents of this playback sequence can be changed using the script shown in **FIG. 57(b)**. For this reason, even when disc (ROM disc) **1** gets old, its playback method (playback result viewed from the user) can be variously changed for the future.

[**0413**] Note that the present invention is not limited to the aforementioned specific embodiments, but can be embodied by variously modifying constituent elements without departing from the scope of the invention when it is practiced. For example, the present invention can be applied not only to DVD-ROM Video that is currently spread worldwide but also to recordable/reproducible DVD-VR (video recorder) whose demand is increasing in recent years. Furthermore, the present invention can be applied to a playback system or a recording/playback system of next-generation HD-DVD which will be spread in the near future.

[**0414**] Furthermore, various inventions can be formed by appropriately combining a plurality of required constituent elements disclosed in the respective embodiments. For example, some required constituent elements may be omitted from all required constituent elements disclosed in the respective embodiments. Furthermore, required constituent elements across different embodiments may be appropriately combined.

What is claimed is:

1. An information recording medium having a data area including a management area which records management information, and an object area which records objects to be managed by the management information, wherein

the object area is configured to store an expanded video object which undergoes playback management using a logical unit called a program chain, and an advanced object recorded independently of the expanded video object,

the management area is configured to store a playback sequence that gives playback conditions of the advanced object, and

the playback sequence is configured to include information used to change this playback sequence.

2. An information recording medium according to claim 1, wherein the playback sequence is configured to include information used to change the advanced object to be played back.

3. An information recording medium according to claim 1, wherein the information recording medium further has a file information area which stores file information corresponding to recorded contents of the data area, the file information area stores a control information file for the playback sequence, and the playback sequence is configured to store data associated with a playback timing, playback position, and playback size of the advanced object as the playback conditions of the advanced object which is configured to be played back together with the expanded video object.

4. An information recording medium according to claim 1, wherein the playback sequence is formed of a programmable playback sequence which gives playback conditions of the expanded video object and the advanced object, respectively.

5. An information recording medium according to claim 1, wherein information which forms the playback sequence is configured to include one of a timed text object using an outline font or vector font, a stream object including video or audio related information, and a Flash object including navigation information and graphics information.

6. An information recording medium according to claim 1, wherein the data area is configured to store a stream which is formed of an access unit that has moving picture meta data

which is configured to be played back upon playback of video contents configured to include the expanded video object or the advanced object, and serves as a data unit which is configured to be independently processed.

7. An information playback method using an information recording medium of claim 1 and an external server, comprising:

comparing a playback sequence loaded from the information recording medium and a playback sequence loaded from the external server, and playing back at least one of the expanded video object and the advanced object from the object area on the basis of contents of the playback sequence of a relatively newer version.

8. An information playback apparatus using an information recording medium of claim 1, comprising:

a playback sequence manager which has a playback sequence recording area that records a playback sequence loaded from the information recording medium; and

a playback engine which plays back at least one of the expanded video object and the advanced object from the object area on the basis of contents of the playback sequence recorded on the playback sequence recording area.

* * * * *