

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4664346号
(P4664346)

(45) 発行日 平成23年4月6日(2011.4.6)

(24) 登録日 平成23年1月14日(2011.1.14)

(51) Int.Cl.

F I

G 1 1 B 27/00 (2006.01)

G 1 1 B 27/00 D

G 1 1 B 27/034 (2006.01)

G 1 1 B 27/034

H O 4 N 5/91 (2006.01)

H O 4 N 5/91 Z

H O 4 N 5/93 (2006.01)

H O 4 N 5/93 G

H O 4 N 5/85 (2006.01)

H O 4 N 5/93 Z

請求項の数 7 (全 59 頁) 最終頁に続く

(21) 出願番号 特願2007-295251 (P2007-295251)
 (22) 出願日 平成19年11月14日(2007.11.14)
 (62) 分割の表示 特願2007-180746 (P2007-180746)
 の分割
 原出願日 平成17年11月30日(2005.11.30)
 (65) 公開番号 特開2008-108419 (P2008-108419A)
 (43) 公開日 平成20年5月8日(2008.5.8)
 審査請求日 平成20年11月21日(2008.11.21)
 (31) 優先権主張番号 特願2004-349144 (P2004-349144)
 (32) 優先日 平成16年12月1日(2004.12.1)
 (33) 優先権主張国 日本国(JP)

(73) 特許権者 000005821
 パナソニック株式会社
 大阪府門真市大字門真1006番地
 (74) 代理人 100090446
 弁理士 中島 司朗
 (74) 代理人 100072442
 弁理士 松村 修治
 (74) 代理人 100125597
 弁理士 小林 国人
 (72) 発明者 橋本 敏史
 大阪府門真市大字門真1006番地 松下
 電器産業株式会社内
 (72) 発明者 大蘆 雅弘
 大阪府門真市大字門真1006番地 松下
 電器産業株式会社内

最終頁に続く

(54) 【発明の名称】 記録媒体、再生装置、プログラム、再生方法

(57) 【特許請求の範囲】

【請求項1】

多重化されたビデオストリーム及びオーディオストリームを含むデジタルストリームと、アプリケーションと、クリック音として出力するためのサウンドデータとが記録された記録媒体であって、

前記記録媒体には前記デジタルストリーム上の再生開始時刻及び再生終了時刻の組みを示すことにより、再生経路を規定する再生経路情報と、実行されるアプリケーションが選択可能な再生経路情報を示す情報を含む管理情報と、前記再生経路情報に対応するフラグとが更に記録されており、

前記フラグは、

前記実行されるアプリケーションに対するユーザ操作に応じたサウンドデータを用いたクリック音の出力と、前記実行されるアプリケーションにより選択された前記再生経路情報に規定された再生経路に従って再生されるデジタルストリームの音声出力とをミキシングするか否かを示す

ことを特徴とする記録媒体。

【請求項2】

アプリケーションを実行すると共に、デジタルストリームを再生する再生装置であって、

前記デジタルストリーム上の再生開始時刻及び再生終了時刻の組みを示すことにより、再生経路を規定する再生経路情報と、実行されるアプリケーションが選択可能な再生経路

情報を示す情報を含む管理情報と、前記再生経路情報に対応するフラグとを記録媒体から読み出す読出手段と、

前記記録媒体に記録されたアプリケーションを実行するプラットフォーム部と、

前記アプリケーションの実行中に、前記アプリケーションが選択した再生経路情報に規定された再生経路に従って再生されるデジタルストリームの再生を実行することにより、映像出力及び音声出力を行う再生制御エンジン部と、

前記アプリケーションが選択した再生経路情報に対応したフラグがオンである場合、前記実行中のアプリケーションに対するユーザ操作に応じたサウンドデータを用いたクリック音の出力と、前記デジタルストリームの音声出力とのミキシングを実行し、前記アプリケーションが選択した再生経路情報に対応したフラグがオフである場合、当該ミキシング

10

を実行しないミキシング部と、

を備える

ことを特徴とする再生装置。

【請求項 3】

前記プラットフォーム部は、当該アプリケーションの実行中に選択可能な再生経路情報を当該アプリケーションに通知し、

前記再生制御エンジン部による再生は、前記実行中のアプリケーションにより選択された再生経路情報に規定された再生経路に従ってデジタルストリームの再生を行なうことである

ことを特徴とする請求項 2 記載の再生装置。

20

【請求項 4】

アプリケーションを実行すると共に、デジタルストリームを再生する再生方法であって、

前記デジタルストリーム上の再生開始時刻及び再生終了時刻の組みを示すことにより、再生経路を規定する再生経路情報と、実行されるアプリケーションが選択可能な再生経路情報を示す情報を含む管理情報と、前記再生経路情報に対応するフラグとを記録媒体から読み出す読出ステップと、

前記記録媒体に記録されたアプリケーションを実行する実行ステップと、

前記アプリケーションの実行中に、前記アプリケーションが選択した再生経路情報に規定された再生経路に従ってデジタルストリームの再生を実行することにより、映像出力及び音声出力を行う再生制御ステップと、

30

前記アプリケーションが選択した再生経路情報に対応したフラグがオンである場合、前記実行中のアプリケーションに対するユーザ操作に応じたサウンドデータを用いたクリック音の出力と、前記デジタルストリームの音声出力とのミキシングを実行し、前記アプリケーションが選択した再生経路情報に対応したフラグがオフである場合、当該ミキシングを実行しないミキシングステップとを含む

再生方法。

【請求項 5】

前記実行ステップは、当該アプリケーションの実行中に選択可能な再生経路情報を当該アプリケーションに通知し、

40

前記再生制御ステップによる再生は、前記実行中のアプリケーションにより選択された再生経路情報に規定された再生経路に示されるデジタルストリームの再生を行なうことである

ことを特徴とする請求項 4 記載の再生方法。

【請求項 6】

アプリケーションの実行を管理すると共に、デジタルストリームの再生をコンピュータに実行させる管理プログラムであって、

前記デジタルストリーム上の再生開始時刻及び再生終了時刻の組みを示すことにより、再生経路を規定する再生経路情報と、実行されるアプリケーションが選択可能な再生経路情報を示す情報を含む管理情報と、前記再生経路情報に対応するフラグとを記録媒体から

50

読み出す読出ステップと、

前記記録媒体に記録されたアプリケーションを実行する実行ステップと、

前記アプリケーションの実行中に、前記アプリケーションが選択した再生経路情報に規定された再生経路に従ってデジタルストリームの再生を実行することにより、映像出力及び音声出力を行う再生制御ステップと、

前記アプリケーションが選択した再生経路情報に対応したフラグがオンである場合、前記実行中のアプリケーションに対するユーザ操作に応じたサウンドデータを用いたクリック音の出力と、前記デジタルストリームの音声出力とのミキシングを実行し、前記アプリケーションが選択した再生経路情報に対応したフラグがオフである場合、当該ミキシングを実行しないミキシングステップと

10

をコンピュータに実行させる管理プログラム。

【請求項 7】

前記実行ステップは、当該アプリケーションの実行中に選択可能な再生経路情報を当該アプリケーションに通知し、

前記再生制御ステップによる再生は、前記実行中のアプリケーションにより選択された再生経路情報に規定された再生経路に示されるデジタルストリームの再生を行なうことである

ことを特徴とする請求項 6 記載の管理プログラム。

【発明の詳細な説明】

【技術分野】

20

【0001】

クリック音再生の技術分野に属する発明である。

【背景技術】

【0002】

クリック音再生とは、映画作品の再生中において表示されたGUIに対し、ユーザが操作を行った場合、その操作に呼応して、映画作品中の再生音声に、クリック音をサウンドミキシングして出力する技術である。

例えば映画作品の再生中に表示されるGUIが、「ゲームの開始」といった機能の実行指示を受け付けるものであり、この実行指示に応じて、「これからゲームを始めるよ」との音声アナウンスをクリック音として、映画本編の音声にサウンドミキシングして出力する場合を考える。この音声アナウンスが、映画作品に登場するキャラクターの声によるものなら、幼年者たるユーザは、この音声アナウンスにより、映画作品に登場するキャラクターから語りかけられているような印象を受ける。以上のクリック音再生の導入により、幼年者が楽しく遊べるようなGUIを映画作品に導入することができる。

30

【0003】

ここで1つの音声出力に、別の音声出力をミキシングして出力する技術には、以下の特許文献 1 に記載されているものがある。

【特許文献 1】特開平 7-320411 号公報

【発明の開示】

【発明が解決しようとする課題】

40

【0004】

ところでクリック音のサウンドミキシングを行うには、デジタル化されたオーディオストリームを一旦、LPCM状態まで展開せねばならない。ここでサウンドミキシングの対象となる本編側のオーディオストリームが5.1チャンネルといったマルチチャンネル属性をもっている場合、展開されたLPCM状態でのデータ量は、多大なものとなる。従って、マルチチャンネル属性をもっているオーディオストリームに、クリック音をサウンドミキシングして、デジタル音声出力を行う場合、帯域上の制限から、ある種類のデジタルインターフェイスでは、デジタル音声出力が不可能になってしまうことがある。近年の映画は、音響面を重視していることが多く、またユーザも、アンプやサラウンドスピーカ等、巨額な私費を投じて、プライベートな音響システムを構築しているケースが多いので、クリック音をサウ

50

ンドミキシングしようとした途端、映画本編のデジタル音声出力が不可能になるというのでは、映画に対するマイナスイメージが図りしれない。

【 0 0 0 5 】

クリック音の導入により“マルチチャネルの音声出力が途切れるのではないか”という不安の呪縛がある限り、クリック音という機能自体が、映画制作に使用されないとのケースが多発する。かかる呪縛から、映画制作スタジオを解放するには、再生装置に再エンコードを行わせるという考えを、あらゆる再生装置に義務付ければよい。つまり再生装置内にエンコーダを具備しておき、5.1チャネルといったマルチチャネル属性をもっているオーディオストリームをLPCM状態に展開して、クリック音をミキシングした後、これを再エンコードした上で、デジタル出力を行えばよい。具体的にいうと、ミキシングした音声のデジタル音声出力をS/PDIF(Sony/PhilipsDigital InterConnect Format, IS060958-3規格)のようなデジタルインターフェースを介して、実行する場合、上述した再エンコードを行い、ドルビーデジタルやDTSへと変換することが必要になる。しかしかかる再エンコード

10

には、エンコーダの具備が必須になる。

【 0 0 0 6 】

本発明の目的は、音声出力に対する弊害をもたらすことなく、クリック音のサウンドミキシング出力を実現することができる記録媒体を提供することである。

上記課題を達成するため、本発明にかかる記録媒体は、多重化されたビデオストリーム及びオーディオストリームを含むデジタルストリームと、アプリケーションと、クリック音として出力するためのサウンドデータとが記録された記録媒体であって、

20

前記記録媒体には前記デジタルストリーム上の再生開始時刻及び再生終了時刻の組みを示すことにより、再生経路を規定する再生経路情報と、実行されるアプリケーションが選択可能な再生経路情報を示す情報を含む管理情報と、前記再生経路情報に対応するフラグとが更に記録されており、

前記フラグは、前記実行されるアプリケーションに対するユーザ操作に応じたサウンドデータを用いたクリック音の出力と、前記実行されるアプリケーションにより選択された再生経路情報に規定された再生経路に従って再生されるデジタルストリームの音声出力とをミキシングするか否かを示すことを特徴としている。

【 0 0 0 7 】

30

本発明は上述した構成を有しているので、実行中のアプリケーションに対するユーザ操作に応じたサウンドデータを用いたクリック音の出力と、実行中のアプリケーションにより選択された再生経路情報に規定された再生経路に示されたデジタルストリームの音声出力とをミキシングするか否かの調整を制作者側の意図に沿ったものとするのが可能となる。これは例えばマルチチャネルでの音声出力を意図するような再生制御の実行時にはサウンドミキシングを無効化しておき、マルチチャネルでの音声出力を意図しないような再生制御の実行時にはサウンドミキシングを有効化しておくことができるようになり、例えば映画の制作スタジオは、マルチチャネルでの再生を意図している場合は、クリック音を禁止し、代わりにクリック音での再生を意図している場合は、代わりにマルチチャネルによる音声出力を禁止するという調整が可能になるばかりか、アプリケーションが選択できる再生経路情報を管理情報に示された再生経路情報に制限することができる。

40

【 0 0 0 8 】

かかる調整の上で、クリック音を映画制作に導入すれば“マルチチャネルの音声出力が途切れるのではないか”という不安の呪縛から、制作スタジオを解放することができる。これにより映画作品作成にあたってのクリック音の導入に弾みをつけることができる。

【発明を実施するための最良の形態】

【 0 0 0 9 】

(第1実施形態)

以降、本発明に係る記録媒体の実施形態について説明する。先ず始めに、本発明に係る記録媒体の実施行為のうち、使用行為についての形態を説明する。図1は、本発明に

50

係る記録媒体の、使用行為についての形態を示す図である。図1において、本発明に係る記録媒体はBD-ROM100であり、BD-ROM100は、再生装置200、リモコン300、テレビ400、サラウンドスピーカに接続されたアンプ500により形成されるホームシアターシステムに、著作物を供給するという用途に供される。

【0010】

本システムにおいて再生装置200は、BD-ROM100に記録されている各種データを再生して、映像出力及び音声出力を行う。この再生装置200による映像出力は、HDMI(High Definition Multimedia Interface)、その他アナログインターフェイスを介して、テレビ400に出力される。また、再生装置200による音声出力は、S/PDIF又はHDMI、その他アナログインターフェイスを介して、テレビ400又はアンプ500に出力される。

10

【0011】

以上が本発明に係る記録媒体の使用形態についての説明である。

続いて本発明に係る記録媒体の生産行為について説明する。本発明に係る記録媒体は、BD-ROMのファイルシステム上における改良で実現することができる。図2は、BD-ROMにおけるファイル・ディレクトリ構成を示す図である。本図においてBD-ROMには、Rootディレクトリの下に、BDMVディレクトリがある。

【0012】

BDMVディレクトリには、拡張子bdmvが付与されたファイル(index.bdmv, MovieObject.bdmv)がある。そしてこのBDMVディレクトリの配下には、更にPLAYLISTディレクトリ、CLIPINFディレクトリ、STREAMディレクトリ、BDBJディレクトリ、BDJAディレクトリ、AUXDATA

20

ディレクトリと呼ばれる6つのサブディレクトリが存在する。

PLAYLISTディレクトリには、拡張子mplsが付与されたファイル(00001.mpls, 00002.mpls, 00003.mpls)がある。

【0013】

CLIPINFディレクトリには、拡張子clpiが付与されたファイル(00001.clpi, 00002.clpi, 00003.clpi)がある。

STREAMディレクトリには、拡張子m2tsが付与されたファイル(00001.m2ts, 00002.m2ts, 00003.m2ts)がある。

BDBJディレクトリには、拡張子BOBJが付与されたファイル(00001.bobj, 00002.bobj, 00003.bobj)が存在する。

30

【0014】

BDJAディレクトリには、拡張子jarが付与されたファイル(00001.jar, 00002.jar, 00003.jar)がある。

AUXDATAディレクトリには、ファイルsound.bdmvが格納される。

以上のディレクトリ構造により、互いに異なる種別の複数ファイルが、BD-ROM上に配置されていることがわかる。

【0015】

<BD-ROMの構成その1.AVClip>

先ず初めに、拡張子.m2tsが付与されたファイルについて説明する。図3は、拡張子.m2tsが付与されたファイルがどのように構成されているかを模式的に示す図である。拡張子.m2tsが付与されたファイル(00001.m2ts, 00002.m2ts, 00003.m2ts・・・)は、AVClipを格納している。AVClipはMPEG2-TransportStream形式のデジタルストリームである。このデジタルストリームは、フィルム映像、NTSC映像、PAL映像をデジタル化することにより得られたデジタルビデオ、LPCM音源、AC-3音源、DTS音源をデジタル化することにより得られたデジタルオーディオを(上1段目)、PESパケットからなるエレメンタリストリームに変換し(上2段目)、更にTSパケットに変換して(上3段目)、同じく字幕系のプレゼンテーショングラフィックスストリーム(PresentationGraphics(PG)ストリーム)及び対話系のインタラクティブグラフィックスストリーム(Interactive Graphics(IG)ストリーム)を(下1段目)を、PESパケット列に変換し(下2段目)、更にTSパケットに変換して(下3段目)、これらを多重化することで構成される。

40

50

【 0 0 1 6 】

PGストリームとは、動画の再生進行に伴った字幕表示を実現するエレメンタリストリームであり、IGストリームは、動画の再生進行に伴ったGUIを実現するエレメンタリストリームである。

PESパケット化されたデジタル映像はビデオストリームと呼ばれ、又はPESパケット化された音声は、オーディオストリームと呼ばれる。これらビデオストリーム、オーディオストリームはエレメンタリストリームであり、独立復号可能な単位として、“AccessUnit”が存在する。ビデオストリームでは、いわゆるGOP(Group Of Picture)が、この“AccessUnit”に該当する。

【 0 0 1 7 】

ビデオストリームのうち、1つのPTSで再生される再生単位(ピクチャ等)を、“Video Presentation Unit”という。オーディオストリームのうち、1つのPTSで再生される再生単位を、“AudioPresentation Unit”という。

ここでAVClipを構成するPESパケットは、1つ以上の“STC_Sequence”を構成する。“STC_Sequence”とは、PESパケットの配列であって、そのPTS、DTSが参照しているSystemTime Clock(STC)の値に、STC不連続点(system time-base discontinuity)が存在しないものをいう。STC不連続点がないことがSTC_Sequenceの要件であるので、1つのSTC_Sequenceを構成するPESパケット列のうち、STC不連続点の直後に位置するPESパケットであって、PCR(ProgramClock Reference)を包含したものから、次のSTC不連続点の直前までが1つのSTC_Sequenceになる。

【 0 0 1 8 】

続いて、以上のように構成されたAVClipが、BD-ROMにどのように書き込まれるかを説明する。図4は、AVClipを構成するTSパケットがどのような過程を経てBD-ROMに書き込まれるかを示す。本図の第1段目にAVClipを構成するTSパケットを示す。

AVClipを構成する188バイトのTSパケットは、第2段目に示すように4バイトのTS_extra_header(図中のハッチング部)、が付されて、192バイト長のSourceパケットになる。このTS_extra_headerは、Arrival_Time_Stampを含む。

【 0 0 1 9 】

AVClipを構成するSourceパケットは、第3段目におけるAVClipにおいて、1つ以上の“ATC_Sequence”を構成する。“ATC_Sequence”とは、Sourceパケットの配列であって、そのArrival_Time_Stampが参照しているArrival_Time_Clockに、不連続点(noarrival time-base discontinuity)が存在しないものをいう。いいかえれば、そのArrival_Time_Stampが参照しているArrival_Time_Clockに、連続性が存在するSourceパケット列を“ATC_Sequence”という。

【 0 0 2 0 】

かかるATC_SequenceがAVClipになり、xxxxx.m2tsというファイル名でBD-ROMに記録される。

かかるAVClipは、通常のコンピュータファイル同様、複数のファイルエクステンツに分割され、BD-ROM上の領域に記録される。第4段目はAVClipがどのようにBD-ROMに記録されるかを模式的に示す。この第4段目においてファイルを構成する各ファイルエクステンツは、予め定められたSextent以上のデータ長を有する。

【 0 0 2 1 】

AVClipを複数のエクステンツに分割して記録する場合の、エクステンツ一個当たりの最小データ長Sextentに検討する。

ここでBD-ROMにおいて光ピックアップのジャンプに要する時間は、

$$T_{\text{jump}} = T_{\text{access}} + T_{\text{overhead}}$$

で与えられる。

【 0 0 2 2 】

T_{access}は、ジャンプ距離に応じて与えられる時間(m秒)であり、ジャンプ距離(論理ブロック数)が0~5000であるなら179m秒、

10

20

30

40

50

ジャンプ距離(論理ブロック数)が5001~10,000であるなら210m秒、

ジャンプ距離(論理ブロック数)が10,001~20,000であるなら270m秒、ジャンプ距離がハーフストロークであるなら990m秒、ジャンプ距離がフルストロークであるなら1220m秒になる。

【0023】

BD-ROMから読み出されたTSパケットは、リードバッファと呼ばれるバッファに格納された上、デコーダに出力されるが、リードバッファへの入力、Rudというビットレートで行われ、ECCブロックにおけるセクタ数をSeccとした場合、

T overheadは、

$$T_{\text{overhead}} = (2 \times \text{Secc} \times 8) / \text{Rud} = 20\text{m秒}$$

という計算で与えられる。

10

【0024】

BD-ROMから読み出されたTSパケットは、Sourceパケットの状態、リードバッファに格納された上、TS_Recording_rateという転送レートで、デコーダに供給される。

TS_Recording_rateという転送レートでの、デコーダへのTSパケット供給が途絶えさせないためには、T jumpの間、リードバッファからデコーダへのTSパケット出力が継続している必要がある。ここでリードバッファからの出力は、TSパケットではなく、Sourceパケットの状態、TSパケットのSourceパケットとのサイズ比を192/188とした場合、T jumpの間、 $(192/188 \times \text{TS_Recording_rate})$ という転送レートにより、リードバッファからのSourceパケット出力が継続している必要がある。

20

【0025】

従って、リードバッファが、アンダーフローしないためのバッファ蓄積量は、

$$B_{\text{occupied}} = (T_{\text{jump}}/1000 \times 8) \times ((192/188) \times \text{TS_Recording_rate})$$

となる。

リードバッファへの入力レートはRud、リードバッファからの出力レートはTS_Recording_rate $\times (192/188)$ であるので、リードバッファへの蓄積レートは、入力レート - 出力レートの計算で与えられ、 $(\text{Rud} - \text{TS_Recording_rate} \times (192/188))$ になる。

【0026】

このB occupiedを、リードバッファに蓄積するのに要する時間Txは、

$$T_x = B_{\text{occupied}} / (\text{Rud} - \text{TS_Recording_rate} \times (192/188))$$

になる。

30

BD-ROMからの読み出しには、この時間TxにおいてRudでのTSパケット入力を継続する必要がある。AVClipを複数のエクステンツに分割して記録する場合、エクステンツ一個当たりの最小データ長S extentは、 $S_{\text{extent}} = \text{Rud} \times T_x = \text{Rud} \times B_{\text{occupied}} / (\text{Rud} - \text{TS_Recording_rate} \times (192/188))$

$$\text{Rud} \times (T_{\text{jump}}/1000 \times 8) \times ((192/188) \times \text{TS_Recording_rate}) / (\text{Rud} - \text{TS_Recording_rate} \times (192/188))$$

$$(\text{Rud} \times T_{\text{jump}}/1000 \times 8) \times \text{TS_Recording_rate} \times 192 / (\text{Rud} \times 188 - \text{TS_Recording_rate} \times 192)$$

になる。よってS extent

40

$$(T_{\text{jump}} \times \text{Rud} / 1000 \times 8) \times (\text{TS_Recording_rate} \times 192 / (\text{Rud} \times 188 - \text{TS_Recording_rate} \times 192))$$

になる。

【0027】

AVClipを構成する各ファイルエクステンツは、こうして算出されたS extent以上のデータ長をもつことにより、AVClipを構成する各ファイルエクステンツが、BD-ROM上において離散的に位置されたとしても、再生時においてデコーダへのTSパケット供給が途絶えさせることなく、連続的に読み出されることになる。

図5は、BD-ROMの物理単位と、1つのファイルエクステンツを構成するSourceパケットとの対応関係を示す図である。第2段目に示すように、BD-ROM上には複数セクタが形成さ

50

れている。ファイルエクステン트를構成するSourceパケットは、第1段目に示すように、32個毎にグループ化されて、連続する3つのセクタに書き込まれる。32個のSourceパケットからなるグループは、6144バイト($=32 \times 192$)であり、これは3個のセクタサイズ6144バイト($=2048 \times 3$)と一致する。3個のセクタに収められた32個のSourceパケットを“Aligned Unit”といい、BD-ROMへの書き込みにあたっては、Aligned Unit単位で暗号化がなされる。

【0028】

第3段目においてセクタは、32個単位で誤り訂正符号が付され、ECCブロックを構成する。再生装置はAligned Unitの単位でBD-ROMをアクセスする限り、32個の完結したSourceパケットを得ることができる。以上がBD-ROMに対するAVClipの書き込みのプロセスである。

10

< BD-ROMの構成その2.Clip情報 >

続いて拡張子.clpiが付与されたファイルについて説明する。拡張子.clpiが付与されたファイル(00001.clpi, 00002.clpi, 00003.clpi・・・)は、Clip情報を格納している。Clip情報は、個々のAVClipについての管理情報である。図6は、Clip情報の内部構成を示す図である。本図の左側に示すようにClip情報は、

i) AVClipについての情報を格納した『ClipInfo()』、ii) ATC Sequence, STC Sequenceに関する情報を格納した『Sequence Info()』 iii) Program Sequenceに関する情報を格納した『Program Info()』 iv) 『Characteristic Point Info(CPI())』からなる。

【0029】

Sequence Infoは、AVClipに含まれる、1つ以上のSTC-Sequence、ATC-Sequenceについての情報である。これらの情報を設けておくことの意義は、STC、ATCの不連続点を、予め再生装置に通知するためである。つまりかかる不連続点が存在すると、AVClip内において同じ値のPTS, ATSが出現する可能性があり、再生時に不都合が生じる。STC, ATCが連続しているのは、トランスポートストリームのうち、どこからどこまでであるかを示すため、SequenceInfoは設けられている。

20

【0030】

Program Infoとは、Program内容が一定である区間(Program Sequence)を示す情報である。Programとは、同期再生のための時間軸を共有し合うエレメンタリーストリーム同士の集まりである。ProgramSequence情報を設けておくことの意義は、Program内容の変化点を、予め再生装置に通知するためである。ここでのProgram内容の変化点とは、ビデオストリームのPIDが変化したり、ビデオストリームの種類がSDTVからHDTVに変化している点等をいう。

30

【0031】

続いてCharacteristic Point Infoについて説明する。図中の引き出し線cu2は、CPIの構成をクローズアップしている。引き出し線cu2に示すように、CPIは、Ne個のEP_map_for_one_stream_PID(EP_map_for_one_stream_PID(0) ~ EP_map_for_one_stream_PID(Ne-1))からなる。これらEP_map_for_one_stream_PIDは、AVClipに属する個々のエレメンタリーストリームについてのEP_mapである。EP_mapは、1つのエレメンタリーストリーム上において、AccessUnitの先頭部分である、Access Unit Delimiterが存在するエントリー位置のパケット番号(SPN_EP_start)を、エントリー時刻(PTS_EP_start)と対応づけて示す情報である。図中の引き出し線cu3は、EP_map_for_one_stream_PIDの内部構成をクローズアップしている。

40

【0032】

これによると、EP_map_for_one_stream_PIDは、Nc個のEP_High(EP_High(0) ~ EP_High(Nc-1))と、Nf個のEP_Low(EP_Low(0) ~ EP_Low(Nf-1))とからなることがわかる。ここでEP_Highは、AccessUnitのSPN_EP_start及びPTS_EP_startの上位ビットを表す役割をもち、EP_Lowは、Access UnitのSPN_EP_start及びPTS_EP_startの下位ビットを示す役割をもち、

図中の引き出し線cu4は、EP_Highの内部構成をクローズアップしている。この引き出し線に示すように、EP_High(i)は、EP_Lowに対する参照値である『ref_to_EP_Low_id[i]』と、AccessUnitの先頭にあたるNon-IDRピクチャ、IDRピクチャのPTSの上位ビットを示

50

す『PTS_EP_High[i]』と、Access Unitの先頭にあたるNon-IDRピクチャ、IDRピクチャのSPNの上位ビットを示す『SPN_EP_High[i]』とからなる。ここで*i*とは、任意のEP_Highを識別するための識別子である。

【0033】

図中の引き出し線cu5は、EP_Lowの構成をクローズアップしている。引き出し線cu5に示すように、EP_Lowは、対応するAccess UnitがIDRピクチャか否かを示す『is_angle_change_point(EP_Low_id)』と、対応するAccessUnitのサイズを示す『l_end_position_offset(EP_Low_id)』と、対応するAccess Unit(Non-IDRピクチャ、IDRピクチャ)のPTSの下位ビットを示す『PTS_EP_Low(EP_Low_id)』と、対応するAccessUnit(Non-IDRピクチャ、IDRピクチャ)のSPNの下位ビットを示す『SPN_EP_Low(EP_Low_id)』とからなる。ここでEP_Low_idとは、任意のEP_Lowを識別するための識別子である。

10

【0034】

<Clip情報の説明その2 .EP_map>

以下、具体例を通じて、EP_mapについて説明する。図7は、映画のビデオストリームに対するEP_map設定を示す図である。第1段目は、表示順序に配置された複数のピクチャ(MPEG4-AVCに規定されたIDRピクチャ、Iピクチャ、Bピクチャ、Pピクチャ)を示し、第2段目は、そのピクチャにおける時間軸を示す。第4段目は、BD-ROM上のTSパケット列を示し、第3段目は、EP_mapの設定を示す。

【0035】

第2段目の時間軸において、時点t1～t7に、Access UnitとなるIDRピクチャ及びIピクチャが存在するものとする。そしてこれらのt1～t7の時間間隔が、1秒程度であるとする。映画に用いられるビデオストリームにおけるEP_mapは、t1～t7をエントリー時刻(PTS_EP_start)として示し、これに対応づけてエントリー位置(SPN_EP_start)を示すよう、設定される。

20

【0036】

<PlayList情報>

続いて、PlayList情報について説明する。拡張子“mpls”が付与されたファイル(00001.mpls)は、PlayList(PL)情報を格納したファイルである。

図8は、PlayList情報のデータ構造を示す図であり、本図において、引き出し線mp1に示すようにPlayList情報は、MainPathを定義するMainPath情報(MainPath())と、チャプターを定義するPlayListMark情報(PlayListMark())とからなる。

30

【0037】

<PlayList情報の説明その1 .MainPath情報>

先ずMainPathについて説明する。MainPathは、主映像たるビデオストリームやオーディオストリームに対して定義される再生経路である。

MainPathは、矢印mp1で示すように複数のPlayItem情報#1・・・#mから定義される。PlayItem情報は、MainPathを構成する1つ以上の論理的な再生区間を定義する。PlayItem情報の構成は、引き出し線hs1によりクローズアップされている。この引き出し線に示すようにPlayItem情報は、再生区間のIN点及びOut点が属するAVClipの再生区間情報のファイル名を示す『Clip_Information_file_name』と、AVClipの符号化方式を示す『Clip_codec_identifier』と、PlayItemがマルチアングルを構成するか否かを示す『is_multi_angle』と、このPlayItemが対象としているSTC_Sequenceを一意に示す『ref_to_STC_id[0]』と、このPlayItemと、その1つ前のPlayItemとの接続を、シームレスに行うか否かを示す『connection_condition』と、再生区間の始点を示す時間情報『In_time』と、再生区間の終点を示す時間情報『Out_time』と、このPlayItemにおいてマスクすべきユーザオペレーションがどれであることを示す『UO_mask_table』と、このPlayItemの途中へのランダムアクセスを寄許可するか否かを示す『PlayItem_random_access_flag』と、このPlayItemの再生終了後、最後のピクチャの静止表示を継続するか否かを示す『Still_mode』と、『STN_table』とから構成される。このうち、再生経路を構成するのは、再生区間の始点を示す時間情報『In_time』、再生区間の終点を示す時間情報『Out_time』の組みであり、再生

40

50

経路情報とは、この『In_time』及び『Out_time』の組みから構成される。

【 0 0 3 8 】

図 9 は、AVClipと、PlayList情報との関係を示す図である。第1段目は、PlayList情報がもつ時間軸を示す。第2段目から第5段目は、EP_mapにて参照されているビデオストリーム(図 7 に示したものと同一)を示す。

PlayList情報は、PlayItem情報#1,#2という2つのPlayItem情報を含んでおり、これらPlayItem情報#1,#2のIn_time,Out_timeにより、2つの再生区間が定義されることになる。これらの再生区間を配列させると、AVClip時間軸とは異なる時間軸が定義されることになる。これが第1段目に示すPlayItem時間軸である。このように、PlayItem情報の定義により、AVClipとは異なる時間軸の定義が可能になる。

10

【 0 0 3 9 】

< STN_table >

STN_tableは、Play ItemのClip_Information_file_nameで指定されているAVClipに多重化された複数エレメンタリストリームのうち、再生可能なものを示すテーブルである。具体的にいうとSTN_tableは、複数エレメンタリストリームのそれぞれについてのentryを、attributeと対応付けることで構成される。

【 0 0 4 0 】

図 1 0 (a) は、STN_tableの内部構成を示す図である。本図に示すようにSTN_tableは、STN_tableにおけるentryと、attributeとの組み(entry-attribute)を複数含み、これらentry - attributeの組みの個数(number_of_video_stream_entries,number_of_audio_stream_entries,number_of_PG_stream_entries,number_of_IG_stream_entries)を示すデータ構造になっている。

20

【 0 0 4 1 】

entry-attributeの組みは、図中の括弧記号 " { " に示すように、Play Itemにおいて再生可能なビデオストリーム、オーディオストリーム、PG_stream、IGストリームのそれぞれに対応している。

entry - attributeの詳細について説明する。図 1 0 (b) ~ (e) は、entry - attributeの詳細を示す図である。

【 0 0 4 2 】

図 1 0 (b) は、ビデオストリームに対応したentry - attributeの組みを示す図である。

30

ビデオストリームにおけるentryは、AVClipを多重分離するにあたって、当該ビデオストリームの抽出に用いられるPIDを示す『ref_to_stream_PID_of_mainClip』を含む。

ビデオストリームにおけるattributeは、0x02に設定された『stream_coding_type』と、ビデオストリームの表示レートを示す『frame_rate』等を含む。

【 0 0 4 3 】

図 1 0 (c) は、オーディオストリームに対応したentry - attributeの組みを示す図である。

オーディオストリームにおけるentryは、AVClipを多重分離するにあたって、当該オーディオストリームの抽出に用いられるPIDを示す『ref_to_stream_PID_of_mainClip』を含む。

40

【 0 0 4 4 】

オーディオストリームにおけるattributeは、0x80(LinearPCM),0x81(AC-3),0x82(DTS)の何れかに設定されることによりオーディオストリームのコーディングタイプを示す『stream_coding_type』と、対応するオーディオストリームのチャンネル構成を示し、マルチチャンネル出力の可否を示す『audio_presentation_type』と、対応するオーディオストリームの言語属性を示す『audio_languagecode』等からなる。

【 0 0 4 5 】

マルチチャンネルには、5.1CHのサラウンド音声の他、ステレオ音声も含まれるが、以降の説明では、マルチチャンネルは、5.1CHのサラウンド音声のみを意味するとして説明を進

50

める。

図10(d)は、PGストリームに対応したentry - attributeの組みを示す図である。

PGストリームにおけるentryは、AVClipを多重分離するにあたって、当該PGストリームの抽出に用いられるPIDを示す『ref_to_stream_PID_of_mainClip』を含む。

【0046】

PGストリームにおけるattributeは、0x90に設定されることによりPGストリームのコーディックを示す『stream_coding_type』と、対応するPGストリームの言語属性を示す『PG_languagecode』とからなる。

図10(e)は、IGストリームに対応したentry - attributeの組みを示す図である。

IGストリームにおけるentryは、AVClipを多重分離するにあたって、当該IGストリームの抽出に用いられるPIDを示す『ref_to_stream_PID_of_mainClip』を含む。

【0047】

IGストリームにおけるattributeは、0x91に設定されることによりIGストリームのコーディックを示す『stream_coding_type』と、対応するIGストリームの言語属性を示す『languagecode』とからなる。

<PlayList情報の説明その2.PlayListMark>

以上が、本実施形態に係るPlayItem情報についての説明である。続いてPlayListMark情報について説明する。

【0048】

図11は、PlayList情報の、PlayListMark情報の内部構成を示す図である。本図の図中の引き出し線pm0に示すように、PlayListMark情報は、複数のPLMark情報(#1~#n)からなる。PLmark情報(PLmark())は、PL時間軸のうち、任意の区間を、チャプター点として指定する情報である。引き出し線pm1に示すようにPLmark情報は、チャプター指定の対象たるPlayItemを示す『ref_to_PlayItem_Id』と、そのPlayItemにおける、チャプター位置を時間表記により示す『mark_time_stamp』とを含む。

【0049】

図12は、PlayList情報の、PLMark情報によるチャプター位置の指定を示す図である。本図の第2段目から第5段目は、図10に示した、EP_mapと、AVClipとを示す。

本図の第1段目は、PLMark情報と、PL時間軸とを示す。この第1段目には、2つのPLMark情報#1~#2が存在する。矢印kt1,2は、PLMark情報のref_to_PlayItem_Idによる指定を示す。この矢印からもわかるように、PLMark情報のref_to_PlayItem_Idは、PlayItem情報のそれぞれを指定していることがわかる。また、Mark_time_Stampは、PlayItem時間軸のうち、Chapter#1,#2になるべき時点を示す。このように、PLMark情報は、PlayItem時間軸上に、チャプター点を定義することができる。

【0050】

AVClip - SubClipを同期させ得る同期区間の定義を可能とするのが、BD-ROMにおけるプレイリスト情報の特徴である。以上のClip情報及びプレイリスト情報は、“静的シナリオ”に分類される。何故なら、以上のClip情報及びプレイリスト情報により、静的な再生単位であるプレイリストが定義されるからである。以上で静的シナリオについての説明を終わる。

【0051】

続いて“動的シナリオ”について説明する。動的シナリオとは、AVClipの再生制御を動的に規定するシナリオデータである。“動的に”というのは、再生装置における状態変化やユーザからのキーイベントにより再生制御の中身がかわることをいう。BD-ROMでは、この再生制御の動作環境として2つのモードを想定している。1つ目は、DVD再生装置の動作環境と良く似た動作環境であり、コマンドベースの実行環境である。2つ目は、Java(登録商標)仮想マシンの動作環境である。これら2つの動作環境のうち1つ目は、HDMVモードと呼ばれる。2つ目は、BD-Jモードと呼ばれる。これら2つの動作環境があるため、動的シナリオはこのどちらかの動作環境を想定して記述される。HDMVモードを想定した動的シナリオはMovieObjectと呼ばれる。一方BD-Jモードを想定した動的シナリオはBD-J Object

10

20

30

40

50

と呼ばれる。

【 0 0 5 2 】

先ず初めにMovie Objectについて説明する。

<Movie Object>

Movie Objectは、図 2 に示したMovieObject.bdmvというファイルに格納され、ナビゲーションコマンド列を含む。

ナビゲーションコマンド列は、条件分岐、再生装置における状態レジスタの設定、状態レジスタの設定値取得等を実現するコマンド列からなる。Movie Objectにおいて記述可能なコマンドを以下に示す。

【 0 0 5 3 】

PlayPLコマンド

書式:PlayPL(第 1 引数, 第 2 引数)

第 1 引数は、プレイリストの番号で、再生すべきプレイリストを指定することができる。第 2 引数は、そのプレイリストに含まれるPlayItemや、そのプレイリストにおける任意の時刻、Chapter、Markを用いて再生開始位置を指定することができる。

【 0 0 5 4 】

PlayItemによりPL時間軸上の再生開始位置を指定したPlayPL関数をPlayPLatPlayItem()

ChapterによりPL時間軸上の再生開始位置を指定したPlayPL関数をPlayPLatChapter()、時刻情報によりPL時間軸上の再生開始位置を指定したPlayPL関数をPlayPLatSpecified Time()という。

【 0 0 5 5 】

JMPコマンド

書式: JMP 引数

JMPコマンドは、現在の動的シナリオを途中で廃棄し(discard)、引数たる分岐先動的シナリオを実行するという分岐である。JMP命令の形式には、分岐先動的シナリオを直接指定している直接参照のものと、分岐先動的シナリオを間接参照している間接参照のものがある。

【 0 0 5 6 】

Movie Objectにおけるナビゲーションコマンドの記述は、DVDにおけるナビゲーションコマンドの記述方式と良く似ているので、DVD上のディスクコンテンツを、BD-ROMに移植するという作業を効率的に行うことができる。MovieObjectについては、以下の国際公開公報に記載された先行技術が存在する。詳細については、本国際公開公報を参照されたい。

【 0 0 5 7 】

国際公開公報W0 2004/074976

以上でMovie Objectについての説明を終える。続いてBD-J Objectについて説明する。

<BD-J Object>

BD-J Objectは、Java (登録商標) プログラミング環境で記述された、BD-Jモードの動的シナリオであり、00001~00003.bobjというファイルに格納される。

【 0 0 5 8 】

図 1 3 は、BD-J Object.bdmvの内部構成を示す図である。アプリケーション管理テーブル(AMT)、プレイリスト管理テーブル(PLMT)、サウンド管理テーブル(SMT)からなる。MovieObjectとの違いは、BD-J Objectにコマンドが直接記述されていない点である。つまりMovie Objectにおいて制御手順は、ナビゲーションコマンドにより直接記述されていた。これに対しBD-JObjectでは、Java (登録商標) アプリケーションに対する指定をアプリケーション管理テーブルに記載することにより、間接的に制御手順を規定している。このような間接的な規定により、複数動的シナリオにおいて制御手順を共通化するという、制御手順の共通化を効率的に行うことができる。

【 0 0 5 9 】

またMovieObjectにおけるプレイリスト再生は、プレイリスト再生を命じるナビゲーションコマンド(PlayPIコマンド)の記述によりなされるが、BD-JObjectにおけるプレイリスト再生は、プレイリスト再生手順を示すプレイリスト管理テーブルをBD-J Objectに組み込むことで記述が可能になる。

このBD-JモードにおけるJava(登録商標)アプリケーションについて説明する。ここでBD-Jモードが想定しているJava(登録商標)プラットフォームは、Java(登録商標)2Micro_Edition(J2ME)Personal Basis Profile(PBP 1.0)と、Globally Executable MHPspecification(GEM1.0.2)for package media targetsとをフル実装したものである。

【0060】

このBD-JモードにおけるJava(登録商標)アプリケーションは、xletインターフェイスを通じて、Application Managerにより、制御される。xletインターフェイスは、“loaded”、“paused”、“active”、“destroyed”といった4つの状態をもつ。

上述したJava(登録商標)プラットフォームは、JFIF(JPEG)やPNG,その他のイメージデータを表示するためのスタンダードJava(登録商標)ライブラリを含む。このため、Java(登録商標)アプリケーションは、HDMVモードにおいてIGストリームにより実現されるGUIとは異なるGUIフレームワークを実現することができる。Java(登録商標)アプリケーションにおけるGUIフレームワークは、GEM1.0.2にて規定されたHAViフレームワークを含み、GEM1.0.2におけるリモートコントロールナビゲーション機構を含む。

【0061】

これにより、Java(登録商標)アプリケーションは、HAViフレームワークに基づくボタン表示、テキスト表示、オンライン表示(BBSの内容)といった表示を、動画像の表示と組み合わせた画面表示を実現することができ、リモートコントロールを用いて、この画面表示に対する操作を行うことができる。

このJava(登録商標)アプリケーションの実体にあたるのが、図2におけるBDMVディレクトリ配下のBDJAディレクトリに格納されたJava(登録商標)アーカイブファイル(00001.jar,00002.jar)である。以降、Java(登録商標)アーカイブファイルについて説明する。

【0062】

<Java(登録商標)アーカイブファイル>

Java(登録商標)アーカイブファイル(図2の00001.jar,00002.jar)は、1つ以上のクラスファイル、1つ以上のデータファイル等を1つにまとめることで得られるファイルであり、BD-Jモードにおいて動作すべきJava(登録商標)アプリケーションを構成する。

図14は、アーカイブファイルにより収められているプログラム、データを示す図である。本図におけるプログラム、データは、枠内に示すディレクトリ構造が配置された複数ファイルを、java(登録商標)アーカイバでまとめたものである。枠内に示すディレクトリ構造は、Rootディレクトリ、Java(登録商標)1,2,3ディレクトリ、Image1,2,3ディレクトリとからなり、Rootディレクトリにcommon.pkgが、Java(登録商標)1,Java(登録商標)2,Java(登録商標)3ディレクトリにクラスファイル(00001.class~00007.class)が、Image1,Image2,Image3ディレクトリに、00001.JPEG~00003.JPEG,00004.PNG~00006.PNGが配置されている。java(登録商標)アーカイブファイルは、これらをjava(登録商標)アーカイバでまとめることで得られる。かかるクラスファイル及びデータは、BD-ROMからキャッシュに読み出されるにあたって展開され、キャッシュ上で、ディレクトリに配置された複数ファイルとして取り扱われる。Java(登録商標)アーカイブファイルのファイル名における“zzzzz”という5桁の数値は、アプリケーションのID(applicationID)を示す。本Java(登録商標)アーカイブファイルがキャッシュに読み出された際、このファイル名における数値を参照することにより、任意のJava(登録商標)アプリケーションを構成するプログラム、データを取り出すことができる。

【0063】

尚、本実施形態においてアプリケーションを構成するプログラム、データは、Java(登録商標)アーカイブファイルにまとめられたが、LZHファイル、zipファイルであってもよ

10

20

30

40

50

い。

以上が、BD-Jモードにおける動的シナリオについての説明である。

< sound.bdmv > 続いてsound.bdmvについて説明する。sound.bdmvは、Java（登録商標）アプリケーションのGUIフレームワークに対して操作がなされた場合、クリック音として出力すべきオーディオデータ（かかるオーディオデータを、サウンドデータという）が格納されるファイルである。

【 0 0 6 4 】

図 1 5 は、ファイルsound.bdmvの構成を示す図である。ファイルsound.bdmvは、Sound Data()と、Sound Index()とからなる。SoundData()は、複数のサウンドデータ(sound_data(0), sound_data(1))からなる。これらのサウンドデータのうち、sound_data(0)は、Java（登録商標）アプリケーションのGUIフレームワークに対する操作時に、第 1 のクリック音として出力される音源である。sound_data(1)は、Java（登録商標）アプリケーションのGUIフレームワークに対する操作時に、第 2 のクリック音として出力される音源である。これらのサウンドデータは、sound_IDと呼ばれる識別子にて指示される。

【 0 0 6 5 】

Sound Index()は、サウンド数(number_of_sound_entries)、sound_data(0)に対するインデックス、sound_data(1)に対するインデックスからなる。

インデックスは、モノラル/ステレオの別などの各サウンドの属性(sound_attributes)、対応するサウンドデータのアドレス(sound_data_start_address)、対応するサウンドデータの連続長(sound_data_length)からなる。

【 0 0 6 6 】

図 2 ~ 図 6 に示したように、映画の劇中に使用される音声(台詞、BGM、効果音)の音源は、オーディオストリームとして、AVClip内に多重化される。これは、映画劇中の音声を表すオーディオストリームを、ビデオストリームの読み出しと同時に再生装置に供給するためである。これに対し、ファイルsound.bdmvは、AVClipとは別個のファイルとしてBD-ROMに記録される。ファイルsound.bdmvは、AVClipとは別個のファイルとして記録されるので、AVClipの読み出し中に、サウンドデータを出力させようとする、ファイルsound.bdmvを読み出すための光ピックアップのジャンプが生じ、AVClipの読み出しが中断せざるを得なくなる。かかる中断が生じれば、AVClip再生に途切れが出る。

【 0 0 6 7 】

かかるAVClipの再生途切れを避けるには、AVClipの再生がなされていない時点において、ファイルsound.bdmvを予めバッファにプリロードしておく必要がある。つまりAVClipの再生に先立ち、ファイルsound.bdmv内のサウンドデータを、プリロードしておく必要がある。以上がファイルsound.bdmvについての説明である。

< Index.bdmv >

Index.bdmvは、タイトルを構成する、Movie Object又はBD-J Objectを示すテーブルである。

【 0 0 6 8 】

Titleにおいて、あるTitleの構成要素となるMovieObjectはどれであるか、又は、あるTitleの構成要素となるBD-J Objectはどれであるのかを定義する。

Index.bdmvについては、以下の国際公開公報に詳細が記載されている。詳細については、本公報を参照されたい。

国際公開公報WO 2004/025651 A1公報

以降、図 1 5 に示したアプリケーション管理テーブル、プレイリスト管理テーブル、サウンド管理テーブルのそれぞれについてより詳しく説明する。

【 0 0 6 9 】

< アプリケーション管理テーブル >

アプリケーション管理テーブル(AMT)について説明する。アプリケーション管理テーブル(AMT)とは、上述したGEM1.0.2for packagemedia targetsにおける“アプリケーション シグナリング”を実装するテーブルである。“アプリケーション シグナリング”とは

、GEM1.0.2が規定するMHP(MultimediaHome Platform)において、“サービス”を生存区間としてアプリケーションの起動、実行を行う制御をいう。本実施形態におけるアプリケーション管理テーブルは、この“サービス”の代わりに、BD-ROMにおける“タイトル”を生存区間にして、アプリケーションの起動、実行の制御を実現する。

【0070】

図16(a)は、アプリケーション管理テーブルの内部構成を示す図である。本図に示すようにアプリケーション管理テーブルは、『life_cycle』と、『apli_id_ref』と、『run_attribute』と、『run_priority』からなる。

図16(b)は、アプリケーション管理テーブルを構成する情報要素の意味内容を示す。

【0071】

『life_cycle』は、アプリケーションの“生存区間”を示す。

『apli_id_ref』は、“アプリケーション識別子”に対する参照値が記述されることにより、左記の生存区間をもつアプリケーションがどれであるかを示す。アプリケーション識別子は、Java(登録商標)アーカイブファイルにおいて、ファイル名として付与された5桁の数値zzzzzで表現される。『apli_id_ref』には、この5桁の数値が記述される。

【0072】

『run_attribute』は、当該生存区間におけるアプリケーションの“起動属性”が記述される。起動属性には、AutoRun、Present、Suspendといった種別がある。

『run_priority』は、当該生存区間におけるアプリケーションの“起動優先度”が記述される。BD-J Objectでは、これらの情報を用いてアプリケーションの挙動を制御する。

<生存区間>

アプリケーション管理テーブルに規定される情報のうち、生存区間について説明する。

【0073】

生存区間とは、BD-ROMに記録されたコンテンツ全体の時間軸において、仮想マシンのワークメモリ上でアプリケーションが生存し得る区間を示す。ワークメモリにおける“生存”とは、そのアプリケーションを構成するxletプログラムが、Java(登録商標)仮想マシン内のワークメモリに読み出され、Java(登録商標)仮想マシンによる実行が可能になっている状態をいう。

【0074】

Java(登録商標)仮想マシンにおいてアプリケーションを動作させる場合、時間軸の何処からアプリケーションによるサービスを開始し、時間軸の何処でアプリケーションによるサービスを終わるかという“サービスの開始点・終了点”を明確に規定することが重要になる。このサービスの開始点・終了点を規定するのが、アプリケーション管理テーブルにおける生存区間である。

【0075】

一方、DVD-Videoのような読出専用ディスクで供給されるディスクコンテンツは、トップメニュータイトルを中核とした構造になっている。そのトップメニュータイトルから、個々の著作物へと分岐して再生を行い、その後再び、トップメニュータイトルに戻るという独特の状態遷移をなす。図17は、ディスクコンテンツにおける状態遷移を示す図である。本図における四角枠は、Titleである。Titleとは、ディスクコンテンツ特有の状態遷移において、1つの“状態”にあたる再生単位であり、このタイトルが、Java(登録商標)アプリケーションの生存区間として取り扱われる。

【0076】

Titleには、BD-ROMのローディング時に最初に再生される『FirstPlayTitle』、Top-Menuを構成する『Top_menuTitle』、これら以外の一般的な『Title』がある。また、図中の矢印jh1,2,3,4,5,6,7,8は、Title間の分岐を象徴的に示す。本図に示される状態遷移とは、BD-ROMローディング時に、『FirstPlayTitle』が再生され、『Top_menuTitle』への分岐が発生して、トップメニューに対する選択待ちになるというものである。

【0077】

10

20

30

40

50

トップメニューに対する選択操作がユーザによりなされれば、選択に従って該当Titleの再生を行い、再びTopMenu Titleに戻るとの処理を、BD-ROMのイジェクトがなされるまで延々と繰り返すというのが、ディスクコンテンツ特有の状態遷移である。

それでは、図17のような状態遷移をなすディスクコンテンツにおいて、Titleは、どのように生存区間として規定されるのであろうか。BD-ROMのローディングがなされた後、図17において矢印jh1,2,3,4・・・に示された参照符号の数値順に分岐がなされ、BD-ROMがイジェクトされたものとする。そうすると、BD-ROMがローディングされてから、イジェクトされるまでの連続時間帯を一本の時間軸と同視することができる。この時間軸を、ディスク全体の時間軸とする。図18(a)は、ディスク全体の時間軸を示す図であり、図18(b)は、この時間軸における構成を示す。図18(b)に示すように、ディスク全体の時間軸は、FirstPlayTitleが再生されている区間、TopMenu Titleが再生されている区間、title#1が再生されている区間等からなる。これらTitleの再生区間はどのように規定されているかという点、Titleは、唯一のBD-JObjectから構成されるから、どれかのMovieObject又はBD-J Objectが、有効になっている期間をTitleの再生区間と考えることができる。

10

【0078】

つまりFirstPlay Title、TopMenu Title、その他のTitleは、何れも動的シナリオから構成されるから、Titleを構成するBD-JObjectのうち、どれかカレントBD-J ObjectとしてActivatedされ、再生装置内において解読・実行に供されている期間を、Titleの再生区間と定義することができる。図19(a)は、BD-ROM全体の時間軸において、識別子bobj_idにより特定されるBD-JObjectから特定されるタイトル再生区間を示す図である。ここで識別子bobj_idにより特定されるBD-J Objectが、1つのTitleを構成しているなら、その識別子bobj_idにより特定されるBD-JObjectが有効になっているBD-ROM時間軸上の一区間を、Titleの再生区間と考えることができる。。

20

【0079】

ここでBD-J ObjectがActivateされている期間の終期は、Title分岐がなされるまでである。つまり、Title分岐がなされるまで、実行の対象になっている動的シナリオは、カレントBD-JObjectとして扱われるから、そのBD-J ObjectにおいてJumpTitleが発生するまでの1つの区間を、Title区間として扱う。

続いてTitle区間と、PL時間軸との関係について説明する。上述したようにMovieObject、BD-J Objectでは、1つの処理手順としてプレイリスト再生手順を記述することができる。プレイリスト再生手順の記述があれば、上述したPL時間軸の全部又は一部がTitle区間に帰属することになる。図19(a)の一例においてBD-JObjectに、プレイリスト管理テーブルが記述されているとする。この場合、BD-J Objectに対応するTitle区間には、図19(b)に示すように、PL時間軸が帰属する。このPL時間軸には更に、複数チャプター(Chapter#1,#2,#3)が定義され得るため、BD-ROM上の時間軸には、BD-ROM全体 - Title - プレイリスト - チャプターというドメインが存在することになる。これらのドメインを用いて、アプリケーションの生存区間を記述することができる。尚、プレイリスト再生は、アプリケーション実行と同時になされたため、プレイリスト再生の途中で、Title分岐が発生することがある。この場合、1つのTitle再生区間内にはプレイリスト時間軸全体ではなく、プレイリスト時間軸の一部分のみが帰属することになる。つまり1つのTitleの再生区間において、プレイリスト時間軸の全体が帰属するか、その一部分が帰属するかは、Title分岐が何時発生するかによって変わる。

30

40

【0080】

図20は、図19(b)の時間軸上に規定される、生存区間の典型を示す図である。本図に示すようにアプリケーションには、Titleを生存区間にした”タイトルバウンダリアプリケーション”、Title内におけるチャプターを生存区間にした”チャプターバウンダリアプリケーション”、BD-ROM全体の時間軸を生存区間にした”タイトルアンバウンダリアプリケーション”という3つの典型がある。

【0081】

50

このうちタイトルバウンダリアプリケーションの生存区間は、そのタイトルの識別子を用いて定義することができる。またチャプターバウンダリアプリケーションの生存区間は、チャプターが属するタイトルの識別子と、そのチャプターの識別子との組みを用いて定義することができる。

プラットフォームが動作していたとしても、Titleやチャプターという生存区間が終われば、リソースをアプリケーションから回収することができる。リソース回収の機会を保証するので、プラットフォームの動作を安定化させることができる。

【 0 0 8 2 】

近い将来、実施されるであろうディスクコンテンツを題材に選んで、アプリケーション管理テーブルにおける生存区間記述について、具体例を交えて説明する。ここで題材にするディスクコンテンツは、映像本編を構成する本編タイトル(title#1)、オンラインショッピングを構成するオンラインショッピングタイトル(title#2)、ゲームアプリケーションを構成するゲームタイトル(title#3)という、性格が異なる3つのタイトルを含むものである。図 2 1 は、本編タイトル、オンラインショッピングタイトル、ゲームタイトルという3つのタイトルを含むディスクコンテンツを示す図である。本図における右側にはIndex.bdmvを記述しており、左側には3つのタイトルを記述している。

【 0 0 8 3 】

右側における破線枠は、各アプリケーションがどのタイトルに属しているかという帰属関係を示す。3つのタイトルのうちtitle#1は、application#1、application#2、application#3という3つのアプリケーションからなる。title#2は、application#3、application#4という2つのアプリケーション、title#3は、application#5を含む。図 2 1 の一例においてapplication#3は、title#1、title#2の双方で起動される。

【 0 0 8 4 】

図 2 1 の破線に示される帰属関係から各アプリケーションの生存区間をグラフ化すると、図 2 2 (a) のようになる。本図において横軸は、タイトル再生区間であり、縦軸方向に各アプリケーションの生存区間を配置している。ここでapplication#1、application#2は、title#1のみに帰属しているので、これらの生存区間は、title#1内に留まっている。application#4は、title#2のみに帰属しているので、これらの生存区間は、title#2内に留まっている。application#5は、title#3のみに帰属しているので、これらの生存区間は、title#3内に留まっている。application#3は、title#1及びtitle#2に帰属しているので、これらの生存区間は、title#1 - title#2にわたる。この生存区間に基づき、アプリケーション管理テーブルを記述すると、title#1、#2、#3のアプリケーション管理テーブルは図 2 2 (b) のようになる。このようにアプリケーション管理テーブルが記述されれば、title#1の再生開始時においてapplication#1、application#2、application#3をワークメモリにロードしておく。そしてtitle#2の開始時にapplication#1、application#2をワークメモリから削除してapplication#3のみにするという制御を行う。これと同様にtitle#2の再生開始時においてapplication#4をワークメモリにロードしておき、title#3の開始時にapplication#3、#4をワークメモリから削除するという制御を行いうる。

【 0 0 8 5 】

更に、title#3の再生中においてapplication#5をワークメモリにロードしておき、title#3の再生終了時にapplication#5をワークメモリから削除するという制御を行いうる。

タイトル間分岐があった場合でも、分岐元 - 分岐先において生存しているアプリケーションはワークメモリ上に格納しておき、分岐元にはなく、分岐先にのみ存在するアプリケーションをワークメモリに読み込めば良いから、アプリケーションをワークメモリに読み込む回数は必要最低数になる。このように、読込回数を少なくすることにより、タイトルの境界を意識させないアプリケーション、つまりアンバウンダリなアプリケーションを実現することができる。

【 0 0 8 6 】

続いてアプリケーションの起動属性についてより詳しく説明する。起動属性には、自動的な起動を示す「AutoRun」、自動起動の対象ではないが、仮想マシンのワークメモリに

10

20

30

40

50

置いて良いことを示す「Present」、仮想マシンのワークメモリにはおかれるが、CPUパワーの割り当ては不可となる「Suspend」がある。

「AutoRun」は、対応するタイトルの分岐と同時に、そのアプリケーションをワークメモリに読み込み、且つ実行する旨を示す属性である。あるタイトルから、別のタイトルへの分岐があると、アプリケーション管理を行う管理主体(アプリケーションマネージャ)は、その分岐先タイトルにおいて生存しており、かつ起動属性がAutoRunに設定されたアプリケーションを仮想マシンのワークメモリに読み込み実行する。これによりそのアプリケーションは、タイトル分岐と共に自動的に起動されることになる。

【 0 0 8 7 】

起動属性「Present」は、継続属性であり、分岐元titleにおけるアプリケーションの状態を継続することを示す。また対応するアプリケーションを実行してよいことを示す属性である。起動属性が「Present」である場合、この起動属性が付与されたアプリケーションは、他のアプリケーションからの呼び出しが許可されることになる。アプリケーション管理を行う管理主体(アプリケーションマネージャ)は、起動中のアプリケーションから呼出があると、そのアプリケーションのapplicationIDが、アプリケーション管理テーブルに記述されていて、起動属性が「Present」であるか否かを判定する。「Present」であれば、そのアプリケーションをワークメモリにロードする。一方、その呼出先アプリケーションのapplicationIDがアプリケーション管理テーブルに記述されていない場合、そのアプリケーションはワークメモリにロードされない。アプリケーションによる呼出は、この「Present」が付与されたアプリケーションに限られることになる。「Present」は、起動属性を明示的に指定しない場合に付与されるデフォルトの起動属性であるから、あるアプリケーションの起動属性が無指定「 - - 」である場合、そのアプリケーションの起動属性の起動属性はこのPresentであることを意味する。

【 0 0 8 8 】

「Suspend」とは、リソースは割り付けられているが、CPUパワーは割り当てられない状態にアプリケーションが置かれることをいう。かかるSuspendは、例えばゲームタイトルの実行中に、サイドパスを経由するという処理の実現に有意義である。

図 2 3 は、起動属性がとり得る三態様(Present、AutoRun、Suspend)と、直前タイトルにおけるアプリケーション状態の三態様(非起動、起動中、Suspend)とがとりうる組合せを示す図である。直前状態が「非起動」である場合、起動属性が「AutoRun」であるなら、分岐先タイトルにおいてそのアプリケーションは、起動されることになる。

【 0 0 8 9 】

直前状態が「非起動」であり、起動属性が「Present」、「Suspend」であるなら、分岐先タイトルにおいてそのアプリケーションは、何もせず、状態を継続することになる。

直前状態が「起動中」である場合、起動属性が「Present」、「AutoRun」であるなら、分岐先タイトルにおいてそのアプリケーションは、何もせず、状態を継続することになる。

【 0 0 9 0 】

起動属性が「Suspend」であるなら、アプリケーションの状態はSuspendされることになる。直前状態が「Suspend」である場合、分岐先タイトルの起動属性が「Suspend」ならSuspendを維持することになる。「Present」又は「AutoRun」であるなら、分岐先タイトルにおいてそのアプリケーションは、レジュームすることになる。アプリケーション管理テーブルにおいて生存区間及び起動属性を定義することにより、タイトル再生区間の進行に沿って、Java(登録商標)アプリケーションを動作させるという同期制御が可能になり、映像再生と、プログラム実行とを伴った、様々なアプリケーションを世に送り出すことができる。

【 0 0 9 1 】

尚、直前状態が「Suspend」であり、分岐先タイトルの起動属性が「Present」の場合は、直前状態、すなわちサスペンド状態を維持しても良い。

最後に、各アプリケーションに対する「起動優先度」について説明する。

この起動優先度は、0～255の値をとり、メモリリソース枯渇時や、CPU負荷が高まった時に、どのアプリケーションを強制的に終了させるか、また、どちらのアプリケーションからリソースを奪うかという処理をアプリケーションマネージャが行うにあたっての判断材料になる。この場合、アプリケーションマネージャは、起動優先度が低いアプリケーションの動作を終了し、起動優先度が高いアプリケーションの動作を継続させるとの処理を行う。

【0092】

また起動優先度は、再生中プレイリストに対する要求が競合した場合のアプリケーション間の調停でも利用される。ここであるアプリケーションが、あるプレイリストの早送りしているものとする。ここで別のアプリケーションが同じプレイリストに対するポーズ要求を行ったとすると、これらのアプリケーションに付与された起動優先度を比較する。そして早送りを命じたアプリケーションの起動優先度が高いなら、かかるアプリケーションによる早送りを継続して行う。逆にポーズを命じたアプリケーションの起動優先度が高いなら、早送り中プレイリストのポーズを行う。

【0093】

以上の生存区間・起動属性・起動優先度により、仮想マシン上で動作し得るアプリケーションの数を所定数以下に制限するよう規定しておくことがオーサリング時に可能なる。そのため、アプリケーションの安定動作を保証することができる。

<プレイリスト管理テーブル>

以上がアプリケーション管理テーブルについての説明である。続いてプレイリスト管理テーブル(PLMT)について説明する。プレイリスト管理テーブルとは、アプリケーションの生存区間において、各アプリケーション実行と同時に行うべき再生制御を示すテーブルである。アプリケーションの動作というのは不安定であり、起動の失敗や異常終了がありうる。そこで起動失敗、異常終了があった場合のFailSafe機構として、本実施形態ではアプリケーションの生存区間毎に、プレイリスト管理テーブルを設けている。プレイリスト管理テーブルは、あるアプリケーションの生存区間が開始した際、これと同時に進行すべき再生制御を規定する情報である。この再生制御とは、プレイリスト情報に基づくAVClip再生であり、プレイリスト情報による再生制御を同時に行うことで、アプリケーション実行と、プレイリスト再生とが同時になされることになる。プレイリスト管理テーブルは、アプリケーションの生存区間毎に設けられるとしたが、プレイリスト管理テーブルが設けられるアプリケーションは、タイトルバウンダリのアプリケーションに限られる。何故ならタイトルアンバウンダリーアプリケーションは、全タイトルを生存区間にしているため、アプリケーション実行と同時にプレイリスト再生を行うという制御は、適合しないからである。

【0094】

チャプターバウンダリアプリケーションは、1つのプレイリスト内のチャプターからアプリケーション実行を開始するという前提の下で生存区間が規定されているため、プレイリスト再生を規定する必要はないからである。以上のことからプレイリスト管理テーブルは、1つ以上のTitleからなる生存区間に定義されることになる。

図24(a)は、プレイリスト管理テーブルの内部構成を示す図である。本図に示すようにプレイリスト管理テーブルは、『PL_id_ref』と、『Playback_Attribute』とからなる。

【0095】

図24(b)は、プレイリスト管理テーブルを構成する情報要素の意味内容を示す。

『PL_id_ref』は、プレイリスト識別子に対する“参照値”が記述されることにより、アプリケーションの生存区間において再生可能となるプレイリストがどれであることを示す。プレイリスト識別子は、ファイルYYYYY.MPLSにおいて、ファイル名として付与された5桁の数値YYYYYで表現される。このYYYYYが記述されることにより、『PL_id_ref』は、対応するTitleにおいて再生可能となるプレイリストがどれであることを示す。

【0096】

10

20

30

40

50

『Playback_Attribute』は、アプリケーション管理テーブルにおける起動属性に倣った属性であり、『PL_id_ref』に記述されたプレイリストを、タイトル開始時において、どのように再生するかを規定する再生属性である。プレイリストに対する再生属性には、『AutoPlay』、『Present』といった種別がある。

『AutoPlay』とは、対応するタイトルの分岐と同時に、そのプレイリストを再生させる旨を示す属性である。あるタイトルから、別のタイトルへの分岐があると、アプリケーション管理を行う管理主体(アプリケーションマネージャ)は、その分岐先タイトルにおいて再生可能であり、かつ再生属性がAutoPlayに設定されたプレイリストの再生を開始する。これにより起動属性がAutoPlayに設定されたプレイリストは、タイトル分岐と共に自動的に起動されることになる。

【 0 0 9 7 】

『Present』とは、起動属性におけるPresent同様、継続属性であり、分岐元titleにおけるプレイリストの状態を継続することを示す。また対応するプレイリストを再生してよいことを示す属性である。例えば連続して再生される2つのTitleがあり、前のタイトル側のプレイリスト管理テーブルでは、あるプレイリストの再生属性がAutoPlayに設定され、カレントタイトル側のプレイリスト管理テーブルでは、そのプレイリストの再生属性がPresentに設定されているものとする。ここでプレイリストの再生時間が2時間長であり、このうち1時間が経過した時点で分岐が発生したとする。この場合カレントタイトルでは、再生属性がPresentに設定されているので、カレントタイトルにおいて、そのプレイリストは、1時間という再生済み区間の直後から、再生されることになる。このように再生属性をPresentに設定しておけば、Title間の分岐があっても、プレイリスト再生をその残りの部分から開始することができる。これにより分岐し合う一連のTitleにおいて、共通のプレイリストを再生するという「タイトル間におけるプレイリスト再生の共通化」を容易に実現することができる。また分岐先タイトルが複数ある場合、これら複数タイトルの再生属性を何れもPresentにしておけば、複数のうちどれに分岐したとしても、1つの共通のプレイリスト再生を継続させることができる。

【 0 0 9 8 】

尚Titleの境界は、シームレス再生を保証しなくてもよいので、上述したように複数Title間で1つのプレイリストを再生しようとする場合、分岐前後でプレイリスト再生を中断させることは許容される。

また、再生属性が「Present」である場合、この再生属性が付与されたプレイリストは、他のアプリケーションからの再生要求により再生されることになる。アプリケーション管理を行う管理主体(アプリケーションマネージャ)は、起動中のアプリケーションから、プレイリストの再生要求があると、要求を受けたプレイリストのPL_id_refが、プレイリスト管理テーブルに記述されていて、再生属性が「AutoPlay」か「Present」のいずれかか否かを判定する。「AutoPlay」か「Present」のいずれかであれば、そのプレイリストを再生する。一方、要求を受けたプレイリストのPL_id_refがプレイリスト管理テーブルに記述されていない場合、そのプレイリストを再生しない。アプリケーションの要求によるプレイリスト再生は、この「AutoPlay」か「Present」のいずれかが付与されたプレイリストに限られることになる。「Present」は、再生属性を明示的に指定しない場合に付与されるデフォルトの再生属性であるから、あるプレイリストの再生属性が無指定「 - - 」であるとそのプレイリストの再生属性はこのPresentであることを意味する。

【 0 0 9 9 】

図25は、プレイリスト管理テーブル、アプリケーション管理テーブルにより規定されるタイトルの具体例を示す。図25の第1段目は、Titleの再生映像を示し、第2段目は、Titleの時間軸を示す。第3段目はPLMTにより再生が規定されるプレイリスト、第4段目は、アプリケーション実行を示す。第4段目においてapplication#1は、Titleの開始と共に起動されており、その後、時点t1において動作状態になる。一方Playlist#1は、Titleの開始と共に再生が開始されている。Playlist#1の再生は、Titleの開始と同じ時点に開始されているので、第1段目の左側に示すように、Titleの再生開始直後から、アプリケーション

ンが動作状態になるまでのスタートアップディレイにおいて、プレイリストの再生画像gj1がフルスクリーン表示される。プレイリスト管理テーブルの再生属性を、“AutoPlay”に設定しておくことにより、Java（登録商標）アプリケーションが動作状態になるまで5～10秒という時間がかかったとしても、その間、“とりあえず何かが写っている状態”になる。この“とりあえず何かが写っている状態”によりタイトル実行開始時のスタートアップディレイを補うことができる。

【0100】

一方、application#1は、時点t1で動作状態になるので、プレイリスト再生画像を子画面、アプリケーションの実行画像を親画面にした合成画像gj2が時点t1において表示されることになる。アプリケーションの実行画像は、Startボタン、continueボタン、POWERインディケータを配置したゲーム用のGUIフレームワークであり、かかるGUIフレームワークの描画処理をJava（登録商標）アプリケーションが実行することでなされる。

10

【0101】

こうした、プレイリストの再生映像と、Java（登録商標）アプリケーションのGUIフレームワークとを組み合わせた再生映像をなすタイトルを構成することができるのが、PLMTの特徴である。

図26は、カレントタイトルがとり得る三態様(プレイリスト管理テーブル無し(i)、プレイリスト管理テーブル有りかつAutoPlay(ii)、プレイリスト管理テーブル有りかつ無指定(iii))と、直前タイトルにおけるプレイリストの状態(非再生状態、再生中状態)とがとりうる6通りの組合せを示す図である。

20

【0102】

本図における6通りの組合せのうち、“直前状態 = 非再生状態”と、“カレントタイトル = プレイリスト管理テーブル有り、かつ、カレントタイトルの再生属性 = AutoPlay”との組合せにおいて、分岐先タイトルにおけるプレイリストの再生は、自動的に開始することになる。

また“直前状態 = 再生中状態”と、“カレントタイトル = プレイリスト管理テーブル無し”との組合せにおいて、分岐先タイトルでのプレイリストの再生は、自動的に停止することになる。

【0103】

そしてこれら2つの組合せ以外は全て、前のタイトルの状態を継続することになる。プレイリスト管理テーブルに基づくプレイリスト再生の開始は、分岐元タイトルにおいて非再生状態であり、分岐先タイトルにおいてAutoPlay属性が付与されている場合に限られるので、タイトルの分岐発生する毎に、プレイリスト再生を開始させる必要はない。タイトル間の分岐が多数発生したとしても、プレイリスト再生を開始させる回数を必要最低数にすることができる。

30

【0104】

プレイリスト管理テーブル及びアプリケーション管理テーブルの記述例について図27(a)を参照しながら説明する。ここで想定する具体例は、2つの連続するTitle(title#1、title#2)であり、そのうちtitle#1では、AutoRunアプリケーションとしてapplication#1、application#2が記述されている。title#2ではAutoRunアプリケーションとしてapplication#2、application#3が記述されている。一方、title#1のプレイリスト管理テーブルには、AutoPlayプレイリストとしてPlaylist#1が、title#2のプレイリスト管理テーブルには、AutoPlayプレイリストとしてPlaylist#2が記述されているものとする。図27(b)は、図27(a)のように記述されたアプリケーション管理テーブル、プレイリスト管理テーブルによりプレイリスト再生、アプリケーション実行がどのように進行するかを示す図である。

40

【0105】

title#1においてアプリケーション管理テーブル、プレイリスト管理テーブルは上述したように設定されているので、title#1の開始時にはapplication#1、application#2が自動的に起動され、Playlist#1の再生が自動的に開始される。

50

title#2においてアプリケーション管理テーブル、プレイリスト管理テーブルは上述したように設定されているので、title#1側に記載はあるが、title#2側には記載がないapplication#1の実行は停止させられる。同じくtitle#1側に記載があるが、title#2側に記載がないPlayList#1の再生も停止させられる。

【 0 1 0 6 】

title#1側に記載はないが、title#2側に記載があるPlayList#2、application#3は再生及び実行が自動的に開始することになる。タイトル分岐があれば、その分岐を契機に、再生すべきプレイリストを他のプレイリストに切り換えることができる。このようにアプリケーション管理テーブル、プレイリスト管理テーブルを用いることで分岐を契機にして、プレイリスト再生を切り換えるという処理をオーサリング段階において規定しておくことができる。

10

【 0 1 0 7 】

また図 2 7 では、application#1、application#2、application#3にはそれぞれ200,128,200の起動優先度が与えられている。これらの起動優先度を付与することにより、PlayList#1、PlayList#2に対する制御要求が競合した場合の調停を行わせることができる。ここでapplication#1がPlayList#1に対し早送りを命じているものとする。一方、application#2がポーズ要求を行ったとする。この場合、アプリケーション管理テーブルに各アプリケーションに対する起動優先度が規定されているため、この起動優先度に従って、両アプリケーションに対する調停がなされることになる。その結果、application#2による要求を退け、application#1による制御を継続するという処理をオーサリング時に規定しておくことができる。起動優先度をプレイリスト管理テーブルと併せて利用することにより、プレイリストに対する制御が競合した場合の調停さえも再生装置に行わせることができる。

20

【 0 1 0 8 】

プレイリスト管理テーブル記述の他の具体例について説明する。図 2 8 (a) は、プレイリスト管理テーブルの他の記述例を示す図である。本図で想定しているのは、2つの連続するタイトル(title#1、title#2)において、title#1側のプレイリスト管理テーブルには、AutoPlayプレイリストとしてPlayList#1が、再生可能なプレイリストとしてPlayList#2が記述され、title#1側のアプリケーション管理テーブルには、AutoPlayアプリケーションであるapplication#1と、実行可能なアプリケーションとしてapplication#2が記述されている。一方title#2側のプレイリスト管理テーブルには再生可能なプレイリストとしてPlayList#2、PlayList#3が記述され、アプリケーション管理テーブルには、AutoRunアプリケーションとしてapplication#3が記述されている。図 2 8 (b) は、図 2 8 (a) のケースに基づくアプリケーション実行及びプレイリスト再生の進行を示す図である。title#1のアプリケーション管理テーブルには、AutoRunアプリケーションとしてapplication#1が記述されているので、title#1の開始時にはapplication#1が自動起動される。一方、title#1のアプリケーション管理テーブルには、実行可能アプリケーションとしてapplication#2が記述されているので、application#1からの呼出yd1によりapplication#2が起動される。

30

【 0 1 0 9 】

title#2側のアプリケーション管理テーブルにおいてapplication#1、application#2が非生存になっており、代わりにAutoRunアプリケーションとしてapplication#3が記述されている。そのためtitle#1 - title#2の境界部では、application#1、application#2を停止し、application#3を自動的に起動するとの処理がなされる。プレイリスト管理テーブルを参照すると、title#1側のプレイリスト管理テーブルは、PlayList#1、PlayList#2が再生可能と記述されており、そのうちPlayList#1はAutoPlay属性になっている。そのためPlayList#1は、title#1の開始時において自動的に再生される。

40

【 0 1 1 0 】

title#1側のプレイリスト管理テーブルには、PlayList#1の他、PlayList#2が再生可能であると記述されているので、application#1はPlayList#1の再生を停止させ、代わりにPlayList#2の再生を要求することにより、プレイリスト交代を実行することができる。

50

title#2側のプレイリスト管理テーブルには、再生可能なプレイリストとしてPlayList#2、PlayList#3が記述されている。そしてAutoPlay属性が付与されたプレイリストはない。そのため、仮にtitle#1開始時において自動再生されたPlayList#1の再生がtitle#2まで継続したとしても、PlayList#1の再生は自動的に終了することになる。

【0111】

しかしPlayList#2再生が継続したままtitle#2に至れば、PlayList#2再生はtitle#2開始以降も継続する。title#2のプレイリスト管理テーブルには、再生可能なプレイリストとしてPlayList#2、PlayList#3が記述されている。そのため、title#2で実行中となるapplication#3は、PlayList#2の再生を停止し、代わりにPlayList#3の再生を要求することにより、再生中のプレイリストを交代させることができる。

10

【0112】

以上のようにプレイリスト管理テーブルの再生属性を、“AutoPlay”に設定しておけば、Java（登録商標）アプリケーションの起動に、5～10秒という時間がかかったとしても、その起動がなされている間、“とりあえず何かが写っている状態”になる。タイトル実行開始時において、アプリケーション起動に時間がかかったとしても、画面は、“とりあえず何かが写っている状態”になる。これにより、アプリケーション起動に時間がかかることによるスタートアップディレイの長期化を補うことができる。

【0113】

アプリケーション管理テーブル及びプレイリスト管理テーブルを定義することにより、タイトル再生区間の進行に沿って、Java（登録商標）アプリケーションを動作させるという同期制御が可能になり、映像再生と、プログラム実行とを伴った、様々なアプリケーションを世に送り出すことができる。

20

<サウンド管理テーブル>

以降サウンド管理テーブル(SMT)について説明する。

【0114】

図29は、サウンド管理テーブルの内部構成を示す図である。本図に示すようにサウンド管理テーブルは、Mixing_Onフラグを含む。Mixing_Onフラグとは、0又は1に設定されることにより、サウンドミキシングを有効とするか、無効とするかを示すフラグである。“有効”とは、オーディオストリームの再生音声に対してミキシングを行った上で出力することをいい、“無効”とは、オーディオストリームの再生音声をスルー出力すること

30

【0115】

Mixing_Onフラグが1(ON)に設定された場合、ファイルsound.bdmvによるクリック音を、同じBD-J Objectに属するPLMTにて再生がなされるプレイリストの再生音声にミキシングする旨を示す。

Mixing_Onフラグが0(OFF)に設定された場合、ファイルsound.bdmvによるクリック音を、同じBD-J Objectに属するPLMTにて再生がなされるプレイリストの再生音声にミキシングしない旨を示す。

【0116】

ファイルsound.bdmvによるクリック音をミキシングするかどうかを、Java（登録商標）アプリケーションの生存区間の単位、つまりタイトルの単位で規定することができる。そして、このMixing_Onフラグをどのような値に設定するかは、そのプレイリストが有するSTN_Tableの設定内容に応じたものとなる。

40

図30(a)(b)は、STN_Tableにおけるオーディオストリームについてのentry-attributeと、Mixing_Onフラグの設定との因果関係を示す図である。図30(a)において、entry-attributeにおけるaudio_presentation_typeが、“マルチチャネル”を示しているなら、Mixing_Onフラグは、0(OFF)に設定される。図30(b)において、entry-attributeにおけるaudio_presentation_typeが、“非マルチチャネル”を示しているなら、Mixing_Onフラグは、1(ON)に設定される。以上のような設定を行うことで、“マルチチャネル”の再生を行うようなプレイリスト(PlayItem)と、アプリケーションとを同時実行する

50

場合、再生装置における音声再生とのミキシングを無効化することができ、“非マルチチャンネル”の再生を行うようなプレイリスト(PlayItem)と、アプリケーションとを同時実行する場合、再生装置における音声再生とのミキシングを有効化することができる。

【0117】

以降、図27(a)におけるサウンド管理テーブルの一例に対して、再生がどのように行われるかについて説明する。図27(a)におけるTitle#1,#2のうち、Title#1のMixing_Onフラグは、音声再生とのミキシングを有効化する旨を示しており、Title#2におけるサウンド管理テーブル#2のMixing_Onフラグは、音声再生とのミキシングを無効化する旨を示しているものとする。

【0118】

更に、Title#1の再生時には、図25のようなGUIフレームワークと、プレイリストの再生映像とが出力されたものとする。ユーザがリモコンを介して、図31に示すように、このGUIフレームワークにおけるボタンを操作したものとする。この場合、このGUIフレームワークを描画したJava(登録商標)アプリケーションのアプリケーション管理テーブルと、同じBD-JObjectに属するサウンド管理テーブルが、Mixing_Onフラグ=1に設定されているなら、ファイルsound.bdmvにおけるサウンドデータが、プレイリストの再生音声にミキシングされる。

【0119】

ここでプレイリストの再生音声は、映画劇中のBGMであり、ファイルsound.bdmvにおけるサウンドデータが、「さあゲームを始めるよ!」という音声案内であれば、この映画劇中のBGMと、音声案内とがミキシングされて出力されることになる。

更に、Title#2の再生時には、図25とは異なるGUIフレームワークと、プレイリストの再生映像とが出力されているものとする。そしてユーザがリモコンを介して、図32に示すように、このGUIフレームワークにおけるボタンを操作したものとする。この場合、このGUIフレームワークを描画したJava(登録商標)アプリケーションのアプリケーション管理テーブルと、同じBD-JObjectに属するサウンド管理テーブルが、Mixing_Onフラグ=0に設定されているなら、ファイルsound.bdmvにおけるサウンドデータは、プレイリストの再生音声にミキシングされない。ここでプレイリストの再生音声は、映画劇中のマルチチャンネルのBGMであれば、この映画劇中のBGMのみが出力されることになる。

【0120】

<Mixing_Onフラグの意義>

再生しようとしているAVClipに、5.1Chのサラウンド音声のオーディオストリームと、モノラル音声のオーディオストリームが多重化されていたとしても、PlayItemのSTN_Tableにおいて、そのモノラル音声のオーディオストリームの再生が許可され、5.1Chのサラウンド音声のオーディオストリームの再生が許可されていないような場合は、そのPlayItemに対応するMixing_OnフラグをONにして、クリックオンとのミキシングを実行することができる。こうすることにより、5.1Chのサラウンド音声のオーディオストリームと、モノラル音声のオーディオストリームとが同一のAVClipを構成したとしても、当該AVClipが5.1Chのサラウンド音声の再生を許していないPlayItemにて、再生されるような場合は、そのPlayItemの再生時におけるクリックオンのミキシングを実行することができる。

【0121】

以上のことを考えれば、クリックオンのミキシングを意図するようなタイトルと、ミキシングを意図しないようなタイトルとを、1つのBD-ROMに記録しようとする場合、5.1Chのサラウンド音声をもつAVClipと、モノラル音声をもつAVClipとを別々にBD-ROMに記録しておく必要がない。

5.1Chのサラウンド音声と、モノラル音声とが多重化されたAVClipを1つだけ、BD-ROMに記録しておき、5.1Chのサラウンド音声のオーディオストリームの再生が許可しているPlayItem情報と、5.1Chのサラウンド音声のオーディオストリームの再生が許可していないPlayItem情報とを記録しておけば、クリックオンのミキシングを意図するようなタイトルと、ミキシングを意図しないようなタイトルとを、1つのBD-ROMにて供給することができる

10

20

30

40

50

。

【 0 1 2 2 】

AVClipの数を多くすることなく、クリックオンのミキシングを意図せず、音響面の充実を意図したタイトル、クリックオンのミキシングを意図して、対話面の充実を意図したタイトルの双方を制作して、ユーザに供給することができるので、制作スタジオの手間が少なく、映画制作の効率化を図ることができる。

<Movie Object、Java（登録商標）アプリとの関係>

HDMVモードにおけるMovie Objectに、クリック音の発音を意図したようなナビゲーションコマンドが組み込まれておらず、BD-JモードにおけるJava（登録商標）アプリケーションに、クリック音の発音を意図しているようなバイトコードが組み込まれていない場合、つまり、アプリケーションがクリック音の再生を行わないことが明らかである場合、Mixing_Onフラグは、0(OFF)に設定される。逆にクリック音の発音を意図したようなナビゲーションコマンドやバイトコードが存在する場合、Mixing_Onフラグは、1(ON)に設定される。以上のような設定を行うことで、クリック音の再生を行わないアプリケーションと、オーディオの再生を行うようなPlayItemとを同時実行する場合、再生装置におけるミキシングを無効化することができ、クリック音の再生を行うアプリケーションと、オーディオの再生を行うようなPlayItemとを同時実行する場合、再生装置におけるミキシングを有効化することができる。

【 0 1 2 3 】

以上が記録媒体 についての説明である。続いて本発明に係る再生装置について説明する。

図 3 3 は、本発明に係る再生装置の内部構成を示す図である。本発明に係る再生装置は、本図に示す内部に基づき、工業的に生産される。本発明に係る再生装置は、主としてシステムLSIと、ドライブ装置という2つのパーツからなり、これらのパーツを装置のキャビネット及び基板に実装することで工業的に生産することができる。システムLSIは、再生装置の機能を果たす様々な処理部を集積した集積回路である。こうして生産される再生装置は、BD-ROMドライブ 1、リードバッファ 2、デマルチプレクサ 3、ビデオデコーダ 4、ビデオプレーン 5、サウンドプロセッサ 6、サウンドプロセッサ 7、ミキサー 8、サウンドコントローラ 9、D/Aコンバータ 10、InteractiveGraphicsデコーダ 11、Interactive Graphicsプレーン 12、Presentation Graphicsデコーダ 13、PresentationGraphicsプレーン 14、JPEGデコーダ 15、Stillプレーン 16、合成部 17、STC-delta付加部 18、ATC-delta付加部 19、ローカルストレージ 20、命令ROM 21、ユーザイベント処理部 22、PSRセット 23、CPU 24、シナリオメモリ 25、ローカルメモリ 26 から構成される。

【 0 1 2 4 】

先ず初めに、BD-ROMに記録されたAVClip再生に係る構成要素(BDドライブ 1 ~ オーディオデコーダ 6)について説明する。

BD-ROMドライブ 1 は、BD-ROMのローディング/イジェクトを行い、BD-ROMに対するアクセスを実行する。

リードバッファ 2 は、FIFOメモリであり、BD-ROMから読み出されたTSパケットが先入れ先出し式に格納される。

【 0 1 2 5 】

デマルチプレクサ(De-MUX) 3 は、リードバッファ 2 からSourceパケットを取り出して、このSourceパケットを構成するTSパケットをPESパケットに変換する。そして変換により得られたPESパケットのうち、STN_Tableに記載されたからPIDをもつものをビデオデコーダ 4、オーディオデコーダ 6、InteractiveGraphicsデコーダ 11、Presentation Graphicsデコーダ 13 のどれかに出力する。

【 0 1 2 6 】

ビデオデコーダ 4 は、デマルチプレクサ 3 から出力された複数PESパケットを復号して非圧縮形式のピクチャを得てビデオプレーン 5 に書き込む。

ビデオプレーン 5 は、非圧縮形式のピクチャを格納しておくためのプレーンである。プレーンとは、再生装置において一画面分の画素データを格納しておくためのメモリ領域である。ビデオプレーン 5 における解像度は1920×1080であり、このビデオプレーン 5 に格納されたピクチャデータは、16ビットのYUV値で表現された画素データにより構成される。ビデオプレーン 5 では、ビデオストリームにおける一フレーム毎の再生映像を、スケーリングすることができる。スケーリングとは、一フレーム毎の再生画像をビデオプレーン 5 全体の1/4(クォータという)、1/1(フルスケールという)のどちらかに変化させることである。かかるスケーリングを、BD-JモードにおいてCPU 2 4 からの指示に従い実行することで、ビデオストリームの再生画像を、画面の隅に追いやったり、全面的に出すという画面演出が可能になる。

10

【 0 1 2 7 】

サウンドプロセッサ 6 は、オーディオストリームを構成するPESパケットをPIDフィルタ 3 が出力した際、このPESパケットを格納しておくデコードバッファ(DB) 6 aと、このバッファに格納されたPESパケットをデコードして、PCM状態のオーディオデータを出力するオーディオデコーダ 6 bとを含む。

サウンドプロセッサ 7 は、BD-ROMから読み出されたファイルsound.bdmvをプリロードしておくプリロードメモリ 7 aと、プリロードされたファイルsound.bdmvにおける複数のサウンドデータのうち、CPU 2 4 により指示されたものをデコードして、PCM状態のオーディオデータを出力するオーディオデコーダ 7 bとを含む。プリロードバッファ 7 aへのプリロードは、BD-ROMの装填時やタイトル切替時に行うことが望ましい。何故なら、AVClipの再生中にファイルsound.bdmvを読み出そうとすると、AVClipとは別のファイルを読み出すための光ピックアップのシークが発生するからである。一方、BD-ROMの装填時やタイトル切替時には、AVClipの再生が継続していることは希なので、かかるタイミングにファイルsound.bdmvを読み出すことにより、AVClip再生が途切れないことを、保証することができる。

20

【 0 1 2 8 】

サウンドミキサ 8 は、サウンドプロセッサ 6、及び、サウンドプロセッサ 7 から出力されるPCM状態のサウンドデータをミキシングする。ミキシングにあたってサウンドミキサ 8 はサンプリング周波数とチャンネル数を合わせるため、サウンドプロセッサ 6 から出力された音声の属性を、サウンドプロセッサ 7 から出力される音声属性に変換するとの処理を行う。このサウンドミキサ 8 によるミキシングは、クリック音の発音を意図したようなナビゲーションコマンド、又は、クリック音の発音を意図したようなバイトコードを、CPU 2 4 が解読することでなされる。

30

【 0 1 2 9 】

サウンドコントローラ 9 は、サウンドミキサ 8 から出力された展開された状態のオーディオデータ、及び、サウンドプロセッサ 6 を介さずに圧縮された状態のオーディオデータのうち、どちらを出力するか切り換える。この音声出力は、S/PDIF又はHDMIを介して、テレビ 4 0 0 又はアンプ 5 0 0 に出力される。圧縮された状態のオーディオデータを出力する場合は、その出力先のテレビ 4 0 0 及びアンプ 5 0 0 においてデコードされる。

40

【 0 1 3 0 】

D/A変換部 1 0 は、サウンドミキサ 8 から出力されるデジタルのオーディオデータをD/A変換して、アナログ音声出力する。

I-Graphicsデコーダ(IGデコーダ) 1 1 は、BD-ROM又はローカルストレージ 2 0 から読み出されたIGストリームをデコードして、非圧縮グラフィクスをInteractiveGraphicsプレーン 1 2 に書き込む。

【 0 1 3 1 】

Interactive Graphics(IG)プレーン 1 2 は、HDMVモードにおいてI-Graphicsデコーダ 1 0 によるデコードで得られた非圧縮グラフィクスが書き込まれる。またBD-Jモードにおいて、アプリケーションにより描画された文字やグラフィクスが書き込まれる。

P-Graphicsデコーダ 1 3 は、BD-ROM又はローカルストレージ 2 0 から読み出されたPGス

50

トリームをデコードして、非圧縮グラフィクスをPresentationGraphicsプレーン 1 1 に書き込む。P-Graphicsデコーダ 1 3 によるデコードにより、字幕が画面上に現れることになる。

【 0 1 3 2 】

Presentation Graphicsプレーン 1 4 は、一画面分の領域をもったメモリであり、一画面分の非圧縮グラフィクスを格納することができる。

JPEGデコーダ 1 5 は、BD-ROM又はローカルストレージ 2 0 に記録されているJPEGデータをデコードして、Stillプレーン 1 6 に書き込む。

Stillプレーン 1 6 は、JPEGデータを展開することで得られた非圧縮のグラフィクスデータが格納されるプレーンである。このグラフィクスデータは、Java（登録商標）アプリが描画する、GUIフレームワークのいわゆる“壁紙”として用いられる。

10

【 0 1 3 3 】

合成部 1 7 は、Interactive Graphicsプレーン 1 2 の格納内容と、Presentation Graphicsプレーン 1 1 の格納内容と、ビデオプレーン 5 の格納内容と、Stillプレーン 1 6 の格納内容とを合成した合成画像を得る。

STC_delta付加部 1 8 は、System Time Clock(STC)を生成する。そしてSTC_Sequenceの切り換わり時において、それまでのSTC_SequenceにおけるSTC値(STC1)に、STC_deltaと呼ばれるオフセット値を加算することにより、新しいSTC_SequenceのSTC値(STC2)を求めて、それまでのSTC_SequenceにおけるSTC値(STC1)と、新しいSTC_SequenceのSTC値(STC2)とを連続した値にする。

20

【 0 1 3 4 】

先行STC_Sequenceにおいて最後に再生されるピクチャの表示開始時刻をPTS1(1stEND)、ピクチャの表示期間をTppとし、後続STC_Sequenceにおいて最初に表示されるピクチャの開始時刻をPTS2(2ndSTART)とした場合、STC_deltaは

$$\text{STC_delta} = \text{PTS1}(1\text{stEND}) + \text{Tpp} - \text{PTS2}(2\text{ndSTART})$$

として表現される。以上のようにSTC_deltaを求め、これが足し合わされたクロックの計数値を各デコーダに出力する。これにより各デコーダは、2つのSTC_Sequenceにあたるストリームを途切れなく再生してゆくことができる。以上により、1つのAVClipの中に、2以上のSTC_Sequenceが存在したとしても、また、連続して再生されるべき2以上のAVClipのそれぞれが、異なるSTC_Sequenceをもっていたとしても、これらのSTC_Sequence間のデコード処理を、シームレスに実行することができる。

30

【 0 1 3 5 】

また、バッファリングの連続性をみたすには、以下の1),2)を満たせば良い。

1) STC2(2ndSTART) > STC2(1stEND)をみたすこと、

ここでSTC2(1stEND)は、STC1(1stEND)を、STC2の時間軸に投影した値であり、 $\text{STC2}(1\text{stEND}) = \text{STC1}(1\text{stEND}) - \text{STC_delta}$ という計算式で与えられる。

2) TS1からのTSパケットの取り出しと、TS2からのTSパケットの取り出しとが、同じ時間軸に投影されたSTC1と、STC2とにより定義され、バッファのアンダーフローや、オーバーフローをもたらさないこと。

【 0 1 3 6 】

40

ATC_delta付加部 1 9 は、Arrival Time Clock(ATC)を生成する。そしてATC_Sequenceの切り換わり時において、それまでのATC_SequenceにおけるATC値(ATC1)に、ATC_deltaと呼ばれるオフセット値を加算することにより、それまでのATC_SequenceにおけるATC値(ATC1)と、新しいATC_SequenceのATC値(ATC2)とを連続した値にする。この加算により、 $\text{ATC2} = \text{ATC1} + \text{ATC_delta}$ になる。ATC_deltaとは、これまで読み出されているトランスポートストリーム(TS1)の最後のTSパケットの入力時点T1から、新たに読み出されたトランスポートストリーム(TS2)の最初のTSパケットの入力時点T2までのオフセット値をいい、“ $\text{ATC_delta} = \text{N1} / \text{TS_recording_rate}$ ”という計算式で与えられる。ここで入力時点T2は、TS2の最初のTSパケットの入力時点、TS1の時間軸上に投影した時点を意味する。またN1は、TS1の最後のビデオPESパケットに後続する、TSパケットのパケット数である。BD-ROMにおいてか

50

かるATC_deltaは、Clip情報に記述されるので、これを用いることにより、ATC_deltaを計算することができる。以上の計算により、これまでのATC_SequenceがもっているATC値(ATC1)と、新たなATC_SequenceがもっているATC値(ATC2)とを、連続した値にすることができる。ATC_deltaが足し合わされたクロックの計数値をデマルチプレクサ(De-MUX)3に出力することで、シームレスなバッファ制御を実現することができる。

【0137】

以上がAVClip再生に係る構成要素である。続いてBD-Jモードでの動作に係る構成要素(ローカルストレージ20～ローカルメモリ26)について説明する。

ローカルストレージ20は、webサイトからダウンロードされたコンテンツ等、BD-ROM以外の記録媒体、通信媒体から供給されたコンテンツを、メタデータと共に格納しておくためのハードディスクである。このメタデータは、ダウンロードコンテンツをローカルストレージ20にバインドして管理するための情報であり、このローカルストレージ20をアクセスすることで、BD-Jモードにおけるアプリケーションは、ダウンロードコンテンツ長さを利用した様々な処理を行うことができる。

10

【0138】

続いて、再生装置における統合制御を実現する構成要素(命令ROM21～ローカルメモリ26)について説明する。

命令ROM21は、再生装置の制御を規定するソフトウェアを記憶している。

ユーザイベント処理部22は、リモコンや再生装置のフロントパネルに対するキー操作に応じて、その操作を行うユーザイベントをCPU24に出力する。

20

【0139】

PSRセット23は、再生装置に内蔵されるレジスタであり、64個のPlayer Status Register(PSR)と、4096個のGeneral Purpose Register(GPR)とからなる。Player Status Registerの設定値(PSR)のうち、PSR4～PSR8は、現在の再生時点を表現するのに用いられる。

PSR4は、1～100の値に設定されることで、現在の再生時点が属するタイトルを示し、0に設定されることで、現在の再生時点がトップメニューであることを示す。

【0140】

PSR5は、1～999の値に設定されることで、現在の再生時点が属するチャプター番号を示し、0xFFFFに設定されることで、再生装置においてチャプター番号が無効であることを示す。

30

PSR6は、0～999の値に設定されることで、現在の再生時点が属するプレイリスト(カレントPL)の番号を示す。

【0141】

PSR7は、0～255の値に設定されることで、現在の再生時点が属するPlayItem(カレントPlayItem)の番号を示す。

PSR8は、0～0xFFFFFFFFの値に設定されることで、45KHzの時間精度を用いて現在の再生時点(カレントPTM(PresentationTime))を示す。以上のPSR4～PSR8により、図18(a)におけるBD-ROM全体の時間軸において、現在の再生時点はどこであることを特定することができる。

【0142】

40

CPU24は、命令ROM21に格納されているソフトウェアを実行して、再生装置全体の制御を実行する。この制御の内容は、ユーザイベント処理部22から出力されたユーザイベント、及び、PSRセット23における各PSRの設定値に応じて動的に変化する。

シナリオメモリ25は、カレントのPL情報やカレントのClip情報を格納しておくためのメモリである。カレントPL情報とは、BD-ROMに記録されている複数プレイリスト情報のうち、現在処理対象になっているものをいう。カレントClip情報とは、BD-ROMに記録されている複数Clip情報のうち、現在処理対象になっているものをいう。

【0143】

ローカルメモリ26は、BD-ROMからの読み出しは低速である故、BD-ROMの記録内容を一時的に格納しておくためのキャッシュメモリである。かかるローカルメモリ26が存在す

50

ることにより、BD-Jモードにおけるアプリケーション実行は、効率化されることになる。

以上が、本実施形態に係る再生装置のハードウェア構成である。続いて本実施形態に係る再生装置のソフトウェア構成について説明する。

【0144】

図34は、ROM24に格納されたソフトウェアと、ハードウェアとからなる部分を、レイア構成に置き換えて描いた図である。本図に示すように、再生装置のレイア構成は、以下のa),b),c)からなる。つまり、

a)BD Player Deviceの第1階層、

b)BD Player Modelの第2階層、

c)Application Runtime Enviromentの第3階層からなる。

10

【0145】

これらの階層のうち図34に示した再生装置のハードウェア構成は、第1階層に属することになる。本図の第1階層“BD Player Device”には、図34に示したハードウェア構成のうちビデオデコーダ4、オーディオデコーダ6、IGデコーダ11、PGデコーダ13にあたる“デコーダ”と、ビデオプレーン5、IGプレーン12、PGプレーン14にあたる“プレーン”、BD-ROM及びそのファイルシステム、ローカルストレージ20及びそのファイルシステムを含む。

【0146】

第2階層“BD Player Model”は、以下のb1),b2)の層からなる。つまり、

b2)Playback Control Engine 32の層

20

b1)Virtual File System 30及びPresentation Engine 31の層

からなり、自身より上位の階層に対し、ファンクションAPIを提供する。

第3階層“Application Runtime Enviroment”は、以下のc1),c2)の階層からなる。つまり、

c1)モジュールマネージャ34が存在する層、

c2)BD-Jプラットフォーム35が存在する層

からなる。

【0147】

先ず初めに、第2層に属するVirtual File System 30～モジュールマネージャ34について説明する。

30

Virtual File System 30は、ローカルストレージ20に格納されたダウンロードコンテンツを、BD-ROMにおけるディスクコンテンツと一体的に取り扱うための仮想的なファイルシステムである。ここでローカルストレージ20に格納されたダウンロードコンテンツは、SubClip、Clip情報、プレイリスト情報を含む。このダウンロードコンテンツにおけるプレイリスト情報はBD-ROM及びローカルストレージ20のどちらに存在するClip情報であっても、指定できる点で、BD-ROM上のプレイリスト情報と異なる。この指定にあたって、Virtual File System 30上のプレイリスト情報は、BD-ROMやローカルストレージ20におけるファイルをフルパスで指定する必要はない。BD-ROM上のファイルシステムやローカルストレージ20上のファイルシステムは、仮想的な1つのファイルシステム(Virtual File System 30)として、認識されるからである。故に、PlayItem情報におけるClip_Information_file_nameは、Clip情報の格納したファイルのファイルポディにあたる5桁の数値を指定することにより、Virtual File System 30、BD-ROM上のAVClipを指定することができる。Virtual File System 30を介してローカルストレージ20の記録内容を読み出し、BD-ROMの記録内容と動的に組み合わせることにより、様々な再生のバリエーションを産み出すことができる。ローカルストレージ20と、BD-ROMとを組合せてなるディスクコンテンツは、BD-ROMにおけるディスクコンテンツと対等に扱われるので、本願における“BD-ROM”は、ローカルストレージ20+BD-ROMの組合せからなる仮想的な記録媒体をも含むことにする。

40

【0148】

Presentation Engine 31は、AV再生ファンクションを実行する。再生装置のAV再生フ

50

アクションとは、DVDプレーヤ、CDプレーヤから踏襲した伝統的な機能群であり、再生開始(Play)、再生停止(Stop)、一時停止(PauseOn)、一時停止の解除(Pause Off)、Still機能の解除(still off)、速度指定付きの早送り(Forward Play(speed))、速度指定付きの巻戻し(BackwardPlay(speed))、音声切り換え(Audio Change)、副映像切り換え(Subtitle Change)、アングル切り換え(AngleChange)といった機能である。AV再生ファンクションを実現するべく、Presentation Engine 3 1は、リードバッファ 2 上に読み出されたAVClipのうち、所望に時刻にあたる部分のデコードを行うよう、ビデオデコーダ 4、P-Graphicsデコーダ 1 3、I-Graphicsデコーダ 1 0、オーディオデコーダ 6を制御する。所望の時刻としてPSR8(カレントPTM)に示される箇所のデコードを行わせることにより、AVClipにおいて、任意の時点を再生を可能することができる。

10

【 0 1 4 9 】

再生制御エンジン(Playback Control Engine(PCE)) 3 2は、プレイリストに対する再生制御ファンクション(i)、PSRセット 2 3における状態取得 / 設定ファンクション(ii)といった諸機能を実行する。プレイリストに対する再生制御ファンクションとは、PresentationEngine 3 1が行うAV再生ファンクションのうち、再生開始や再生停止を、カレントPL情報及びClip情報に従って行わせることをいう。これら機能(i)~(ii)は、HDMVモジュール 3 3~BD-Jプラットフォーム 3 5からのファンクションコールに応じて実行する。

【 0 1 5 0 】

モジュールマネージャ 3 4は、BD-ROMから読み出されたIndex.bdmvを保持して、分岐制御を行う。この分岐制御は、カレントタイトルを構成する動的シナリオにTerminateイベントを発行し、分岐先タイトルを構成する動的シナリオにActivateイベントを発行することとされる。

20

以上がPresentation Engine 3 1~モジュールマネージャ 3 4についての説明である。続いてBD-Jプラットフォーム 3 5について説明する。

【 0 1 5 1 】

BD-Jプラットフォーム 3 5は、いわゆるJava(登録商標)プラットフォームであり、Java(登録商標)仮想マシン 3 6を中核にした構成になっている。BD-Jプラットフォーム 3 5は、上述したJava(登録商標)2Micro_Edition(J2ME)Personal Basis Profile(PBP 1.0)と、Globally Executable MHPspecification(GEM[1.0.2])for package media targetsに加え、BD-J Extentionを実装している。BD-JExtentionは、GEM[1.0.2]を越えた機能を、BD-Jプラットフォームに与えるために特化された、様々なパッケージを含んでいる。

30

【 0 1 5 2 】

先ず初めに、BD-Jプラットフォーム 3 5の中核となるJava(登録商標)仮想マシン 3 6について説明する。

< Java(登録商標)仮想マシン 3 6 >

図 3 5は、Java(登録商標)仮想マシン 3 6の内部構成を示す図である。本図に示すようにJava(登録商標)仮想マシン 3 6は、図 3 3に示したCPU 2 4と、ユーザクラスローダ 5 2、メソッドエリア 5 3、ワークメモリ 5 4、スレッド 5 5 a,b・・・n、Java(登録商標)スタック 5 6 a,b・・・nとから構成される。

【 0 1 5 3 】

40

ユーザクラスローダ 5 2は、BDJAディレクトリのJava(登録商標)アーカイブファイルにおけるクラスファイルをローカルメモリ 2 6等から読み出してメソッドエリア 5 3に格納する。このユーザクラスローダ 5 2によるクラスファイル読み出しは、ファイルパスを指定した読み出しをアプリケーションマネージャ 3 7がユーザクラスローダ 5 2に指示することとされる。ファイルパスがローカルメモリ 2 6を示しているなら、ユーザクラスローダ 5 2は、アプリケーションを構成するJava(登録商標)アーカイブファイルにおけるクラスファイルを、ローカルメモリ 2 6からワークメモリ 5 4に読み出す。ファイルパスがVirtualFile System 3 0上のディレクトリを示しているなら、ユーザクラスローダ 5 2は、アプリケーションを構成するJava(登録商標)アーカイブファイルにおけるクラスファイルを、BD-ROM又はローカルストレージ 2 0からワークメモリ 5 4に読み出す。アプ

50

リケーションの起動制御は、このユーザクラスローダ52によるクラスファイル読み出しにより実現される。読み出しが指示されたクラスファイルがローカルメモリ26にない場合、ユーザクラスローダ52は読み出し失敗をアプリケーションマネージャ37に通知することになる。

【0154】

メソッドエリア53は、ユーザクラスローダ52によりローカルメモリ26から読み出されたクラスファイルが格納される。

ワークメモリ54は、いわゆるヒープエリアであり、様々なクラスファイルのインスタンスが格納される。図34に示したアプリケーションマネージャ37は、このワークメモリ54に常駐するレジデントアプリケーションである。ワークメモリ54には、これらレジデント型のインスタンスの他に、メソッドエリア53に読み出されたクラスファイルに対応するインスタンスが格納される。このインスタンスが、アプリケーションを構成するxletプログラムである。かかるxletプログラムをワークメモリ54に配置することによりアプリケーションは実行可能な状態になる。

【0155】

図34のレイアモデルでは、このワークメモリ54上のアプリケーションマネージャ37を、Java(登録商標)仮想マシン36上に描いていたが、これはわかり易さを意図した配慮に過ぎない。アプリケーションマネージャ37及びアプリケーションはインスタンスとしてスレッド55a,b,...nにより実行されるというのが、現実的な記述になる。

スレッド55a,b,...nは、ワークメモリ54に格納されたメソッドを実行する論理的な実行主体であり、ローカル変数や、オペランドスタックに格納された引数をオペランドにして演算を行い、演算結果を、ローカル変数又はオペランドスタックに格納する。図中の矢印ky1,ky2,kynは、ワークメモリ54からスレッド55a,b,...nへのメソッド供給を象徴的に示している。物理的な実行主体がCPU唯1つであるのに対し、論理的な実行主体たるスレッドは、最大64個Java(登録商標)仮想マシン36内に存在し得る。この64個という数値内において、スレッドを新規に作成することも、既存のスレッドを削除することも可能であり、スレッドの動作数は、Java(登録商標)仮想マシン36の動作中において増減し得る。スレッドの数は適宜増やすことができるので、複数スレッドにより1つのインスタンスの並列実行を行い、インスタンスの高速化を図ることもできる。本図ではCPU24と、スレッドとの対応関係は、1対多の関係にしているが、CPUが複数ある場合、CPUとスレッドとの対応関係は多対多の関係になりうる。スレッド55a,b,...nによるメソッド実行は、メソッドをなすバイトコードを、CPU24のネイティブコードに変換した上、CPU24に発行することとされる。このネイティブコード変換については、本願の主眼から外れるため、説明を省く。

【0156】

Java(登録商標)スタック56a,b,...nは、スレッド55a,b,...nと1対1の比率で存在しており、プログラムカウンタ(図中のPC)と、1つ以上のフレームとを内部に持つ。"プログラムカウンタ"は、インスタンスにおいて、現在どの部分が実行されているかを示す。"フレーム"はメソッドに対する1回のコールに対して割り当てられたスタック式の領域であり、その1回のコール時の引数が格納される"オペランドスタック"と、コールされたメソッドが用いる"ローカル変数スタック(図中のローカル変数)"とからなる。フレームは、コールが1回なされる度にJava(登録商標)スタック56a,b,...n上に積み上げられるのだから、あるメソッドが自身を再帰的に呼び出す場合も、このフレームは、1つ積み上げられることになる。

【0157】

以上がJava(登録商標)仮想マシンについての説明である。

<アプリケーションマネージャ37>

アプリケーションマネージャ37は、Java(登録商標)仮想マシン36内のワークメモリ上で動作するシステムソフトウェアであり、タイトル分岐が発生する度に、分岐前タイトルでは実行されていないが、新たなタイトルではAutoRunの起動属性を有するアプリケ

10

20

30

40

50

ーションを起動するようJava（登録商標）仮想マシン36に指示する。それと共に、分岐前タイトルでは実行されていたが、新たなタイトルを生存区間としないアプリケーションを終了させる。これら起動制御及び終了制御は、カレントBD-JObjectにおけるアプリケーション管理テーブルを参照した上でなされる。

【0158】

図36は、BD-J Objectにおけるアプリケーション管理テーブルに基づく、アプリケーションマネージャ37の処理を示す図である。

図36における1, 2, 3は、アプリケーション管理テーブル参照(1)、Java（登録商標）仮想マシン36に対するアプリケーション起動指示(2)、Java（登録商標）仮想マシン36による、Java（登録商標）アーカイブファイルに対する読み出し指示(3)、Java（登録商標）アプリケーションを定義するクラスファイルのクラスロード(4,5,6)という一連の過程を模式化して示す。この起動指示によりJava（登録商標）仮想マシン36は、ローカルメモリ26からワークメモリにxletプログラムを読み出す。

【0159】

図37は、BD-J ObjectにおけるPLMTに基づく、アプリケーションマネージャ37の処理を示す図である。1は、BD-J ObjectにおけるPLMTの参照を示し、2は、Presentation Engine 31に対するプレイリスト情報の読み出し指示を示す。

図37における1,2,3,4は、Virtual File System 30経由のプレイリスト情報読み出し(1)、プレイリスト情報を構成するPlayItem情報の解読(2)、Virtual File System 30経由のClip情報読み出し(3)、Clip情報の解読(4)を模式化したものである。以上の過程を経てClip情報、プレイリスト情報が解読されれば、AVClipを構成するTSパケットを、Virtual File System 30を通じてPresentation Engine 31に引き渡す。このようにしてPresentation Engine 31にTSパケットが順次渡れば、Presentation Engine 31はAVClipを構成するTSパケットをデコーダに出力して、プレーンに表示させる。図中の1,2,3,4は、AVClipを構成するTSパケットの読み出し(1,2)、Virtual File System 30からPresentation Engine 31へのTSパケット引き渡し(3)、デコーダへのTSパケット投入(4)、デコーダから各種プレーンへのデコード結果出力(5)を模式的に示している。

【0160】

図38は、BD-J Objectにおけるサウンド管理テーブルに基づく、アプリケーションマネージャ37の処理を示す図である。図38における0.1,2は、アプリケーションマネージャ37によるBD-JObject内のサウンド管理テーブルの参照(0)、アプリ生存区間毎のサウンドミキシングの有効化、無効化の指示(1)。操作されたボタンに対応するサウンドデータの、出力指示又はデコード指示(2)を模式的に示したものである。

【0161】

以降、ソフトウェアによるアプリケーションマネージャ37の実装について説明する。図39は、アプリケーションマネージャ37による処理手順を示すフローチャートである。本図における処理手順は、ステップS1 - ステップS2 - ステップS3 - ステップS4からなるメインループを有している。ステップS1は、TitleJumpがなされたか否かの判定であり、もしなされればTitle切り換えを行う(ステップS7)。

【0162】

ステップS8は、カレントタイトルに対応するBD-JObjectに、PLMTが存在するか否かの判定である。もし存在しないのであれば、前のタイトルでは、PLMTに記載されていたプレイリストの再生を停止させる(ステップS9)。

もし存在するのであれば、前のタイトルではPLMTに記載されていなかったが、カレントタイトルでは、PLMTに記載されており、AutoPlay属性が付与されているプレイリストの再生を開始させる(ステップS10)。

【0163】

ステップS11では、カレントTitleに対応するBD-JObjectに、サウンド管理テーブルが存在するか否かの判定を行う。ステップS12は、サウンド管理テーブルが存在する場合に実行される判定ステップであり、そのサウンド管理テーブルにおけるMixing_Onフラ

10

20

30

40

50

グが1であるか否かの判定を行う。もしMixing_Onフラグが1であるなら、サウンドミキシングを有効化しようサウンドコントローラ9に指示し(ステップS 1 3)、もしMixing_Onフラグが0であるなら、サウンドミキシングを無効化しようサウンドコントローラ9に指示する(ステップS 1 4)。

【0164】

ステップS 1 5は、カレントTitleに対応するBD-JObjectにアプリケーション管理テーブルが存在するか否かの判定であり、もし存在するなら、前のTitleを生存区間としていないが、カレントTitleを生存区間としているJava(登録商標)アプリケーションであって、AutoRun属性を有するものを起動する(ステップS 1 6)。もし存在しないなら、前のTitleを生存区間としているが、カレントTitleを生存区間としていないアプリを停止する(ステップS 1 7)。

10

【0165】

その後、Java(登録商標)アプリケーションの起動に成功したか否かの判定を行い(ステップS 1 8)、起動に成功すれば(ステップS 1 8でYes)、AutoPlayPLの再生画像をクォータ(1/4)に変換する(ステップS 1 9)という手順を実現するものである。

一方このステップS 1 8がNoであればステップS 2 3、S 2 4、S 1 6、S 1 8からなるループ処理を実行することになる。本ループ処理における制御変数は、再起動カウンタである。再起動カウンタは、アプリケーションの再起動回数を規定するカウンタである。本再起動カウンタは、本フローチャートの起動時にリセットされ、ステップS 2 3において、0か否かの判定がなされる。0でない場合、ステップS 2 4において再起動カウンタはデクリメントされる。以上のステップS 2 3、S 2 4、S 1 6、S 1 8～ステップS 1 9からなるループ処理により、再起動カウンタが0でない限り、AutoRunアプリケーションの起動は繰り返されることになる。かかる繰り返しにより、アプリケーションの起動が保証されることになる。

20

【0166】

ステップS 2は、メインとなるアプリケーションが実行されてない状態かどうかの判定であり、もしそうであれば、ステップS 5の判定を行う。ステップS 5は、アプリケーションが正常終了したかの判定である。もし異常終了していればステップS 2 1、ステップS 2 2の処理を実行する。正常終了していればステップS 2 1～ステップS 2 2を実行せず、ステップS 1～ステップS 4からなるメインループに戻る。

30

【0167】

ステップS 2 1は、AutoPlayPLの再生中であるか否かの判定であり、もし再生中なら、AutoPlayPLの再生画像をフルスクリーン化しようPlaybackControl Engine 3 2に指示する(ステップS 2 2)。その後、ステップS 2 3に移行する。ステップS 2 3への移行により、異常終了時においてもステップS 1 4～ステップS 1 7からなるループ処理が実行されることになる。これにより再起動カウンタの回数が0になるまで、アプリケーションの再起動は繰り返されることになる。

【0168】

ステップS 4は、BDドライブ1にBD-ROMが存在しているか否かの判定であり、もしBD-ROMが存在しなければ、全てのアプリケーションに対し、終了指示を発する(ステップS 6)。

40

以上のように本実施形態によれば、マルチチャネルでの音声出力を意図するような再生制御の実行時にはサウンドミキシングを無効化しておき、マルチチャネルでの音声出力を意図しないような再生制御の実行時にはサウンドミキシングを有効化しておくことができる。

【0169】

これにより、映画の制作スタジオは、マルチチャネルでの再生を意図している場合は、クリック音を禁止し、代わりにクリック音での再生を意図している場合は、代わりにマルチチャネルによる音声出力を禁止するという調整が可能になる。

クリック音の導入により“マルチチャネルの音声出力が途切れるのではないか”という

50

不安の呪縛から、制作スタジオを解放することができるので、映画作品作成にあたってのクリック音の導入に弾みをつけることができる。

【 0 1 7 0 】

尚、第 1 実施形態では、マルチチャンネルが5.1CHのサラウンド音声であり、オーディオストリームがこのマルチチャンネルの属性を有している場合、Mixing_OnフラグをOFFにすると説明したが、マルチチャンネルが2CHのステレオ音声を意味し、オーディオストリームがステレオ属性を有している場合、Mixing_OnフラグをOFFにしてもよい。

ステレオ音声を展開した後のデータ量はそれ程大きい訳ではなく、LPCM状態に展開した後に、サウンドミキシングを行っても、デジタル出力が可能になることもある。このように、ステレオ音声の展開後のデータ量はそれ程大きくないと期待でき、デジタル出力が可能であることは確認し得た場合、マルチチャンネルをONに設定してもよい。

【 0 1 7 1 】

(第 2 実施形態)

第 1 実施形態では、プレイリストの自動再生がなされたアプリケーションの生存区間毎にMixing_Onフラグを設けたが、これは見方を変えたと、プレイリストと、アプリケーション生存区間とが等価であり、プレイリストを構成する再生区間毎に、Mixing_Onフラグによる制御が及ぶと考えることができる。

【 0 1 7 2 】

これらに鑑み、本実施形態ではMixing_Onフラグをプレイリスト情報に設ける技術を提案する。

図 4 0 (a) は、第 2 実施形態に係るプレイリスト情報の内部構成を示す図である。本図が、図 8 (a) に示したプレイリスト情報の内部構成と異なるのは、PlayItem情報内にMixing_Onフラグが設けられている点である。図 4 0 (b) は、PlayItem情報内に設けられた、Mixing_Onフラグの内容を示す図である。

【 0 1 7 3 】

Mixing_Onフラグが1に設定された場合、ファイルsound.bdmvによるクリック音を、PlayItemの再生音声にミキシングする旨を示す。

Mixing_Onフラグが0に設定された場合、ファイルsound.bdmvによるクリック音を、PlayItemの再生音声にミキシングしない旨を示す。

ファイルsound.bdmvによるクリック音をミキシングするかどうかを、プレイリストの単位で規定することができる。そして、このMixing_Onフラグをどのような値に設定するかは、そのPlayItem情報が有するSTN_Tableの設定内容に応じたものとなる。

【 0 1 7 4 】

以上が、第 2 実施形態に係る記録媒体 の改良である。続いて第 2 実施形態に係る再生装置の改良について説明する。第 2 実施形態に係る再生装置の改良点は、Playback Control Engine 3 2 に存在する。

以降図 4 1 のフローチャートを参照して、Playback Control Engine 3 2 による具体的な制御手順を説明する。

【 0 1 7 5 】

図 4 1 は、Playback Control Engine 3 2 によるプレイリスト再生手順を示すフローチャートである。この再生手順は、PresentationEngine 3 1 に対する制御(ステップ S 1 0 6)と、BD-ROMドライブ 1 又はローカルストレージ 2 0 に対する制御(ステップ S 1 0 8)とを主に含む。本フローチャートにおいて処理対象たるPlayItemをPlayItem#xとする。本フローチャートは、カレントPL情報(.mpls)の読み込みを行い(ステップ S 1 0 1)、その後、ステップ S 1 0 2 ~ ステップ S 1 1 0 の処理を実行するというものである。ここでステップ S 1 0 2 ~ ステップ S 1 1 0 は、ステップ S 1 0 9 がYesになるまで、カレントPL情報を構成するそれぞれのPI情報について、ステップ S 1 0 3 ~ ステップ S 1 1 0 の処理を繰り返すというループ処理を構成している。このループ処理において処理対象となるPlayItemを、PlayItem#x(PI#x)とよぶ。このPlayItem#xは、カレントプレイリストの先頭のPlayItemに設定されることにより、初期化される(ステップ S 1 0 2)。上述したループ処

理の終了要件は、このPlayItem#xがカレントプレイリストの最後のPlayItemになることであり(ステップS 1 0 9)、もし最後のPlayItemでなければ、カレントプレイリストにおける次のPlayItemがPlayItem#xに設定される(ステップS 1 1 0)。

【 0 1 7 6 】

ループ処理において繰り返し実行されるステップS 1 0 3 ~ ステップS 1 1 0 は、PlayItem#xのClip_information_file_nameで指定されるClip情報をシナリオメモリ25に読み込み(ステップS 1 0 3)、PlayItem#xのIn_timeを、カレントClip情報のEPmapを用いて、Iピクチャアドレスuに変換し(ステップS 1 0 4)、PlayItem#xのOut_timeを、カレントClip情報のEP_mapを用いて、Iピクチャアドレスvに変換して(ステップS 1 0 5)、これらの変換で得られたアドレスvの次のIピクチャを求めて、そのアドレスの1つ手前をアドレスwに設定し(ステップS 1 0 7)、そうして算出されたアドレスwを用いて、IピクチャアドレスuからアドレスwまでのTSパケットの読み出しをBD-ROMドライブ1又はローカルストレージ20に命じるというものである(ステップS 1 0 8)。

10

【 0 1 7 7 】

一方、Presentation Engine 31 に対しては、カレントPLMarkのmark_time_stampからPlayItem#xのOut_timeまでの出力を命じる(ステップS 1 0 6)。以上のステップS 1 0 5 ~ ステップS 1 0 8 により、AVClipにおいて、PlayItem#xにより指示されている部分の再生がなされることになる

その後、PlayItem#xがカレントプレイリストの最後のPIであるかの判定がなされる(ステップS 1 0 9)。

20

【 0 1 7 8 】

PlayItem#xがカレントプレイリストの最後のPIでなければ、カレントプレイリストにおける次のPlayItemを、PlayItem#xに設定して(ステップS 1 1 0)、ステップS 1 0 3 に戻る。以上のステップS 1 0 3 ~ ステップS 1 1 0 を繰り返すことにより、プレイリストを構成するPIは順次再生されることになる。

本フローチャートにおいて、ステップS 1 1 5 ~ ステップS 1 1 7 は、ステップS 1 0 3 ~ ステップS 1 1 0 が一巡される毎に実行されるステップである。ステップS 1 1 5 は、PlayItem#xにおけるMixing_Onフラグが1であるか否かの判定を行う。もしMixing_Onフラグが1であるなら、サウンドミキシングを有効化しようサウンドコントローラ9に指示し(ステップS 1 1 6)、もしMixing_Onフラグが0であるなら、サウンドミキシングを無効化しようサウンドコントローラ9に指示する(ステップS 1 1 7)。

30

【 0 1 7 9 】

以上のように本実施形態によれば、HDMVモードにおいても、BD-Jモードにおいても、利用されるプレイリスト情報にMixing_Onフラグを設けるので、HDMVモードにおいて、IGストリームを用いてGUIが実現される場合であっても、クリック音のサウンドミキシングの有効化、無効化を規定しておくことができる。

またクリック音に限らず、WWWサイトからダウンロードした他のオーディオストリームとのサウンドミキシングを実現する場合であっても、そのサウンドミキシングの有効化/無効化をMixing_Onフラグに規定しておくことができる。

【 0 1 8 0 】

40

(第3実施形態)

第1実施形態においてJava(登録商標)仮想マシンにおけるプレイリスト再生は、BD-J Object内のプレイリスト管理テーブルを用いて規定することができた。ここで問題になるのがプレイリスト管理テーブルである。つまりプレイリストを再生してよいか否かは、BD-J Object毎のプレイリスト管理テーブルに記述されているから、あるTitleでは再生可能であるが、別のTitleでは再生不可能になることがある。またプレイリスト再生は可能であったとしても、著作権保護の観点から、ある種のアプリケーションからの再生を禁じたい場合がある。かかるプレイリスト再生の制限を実現するため、第3実施形態では、Java(登録商標)プラットフォーム35にパーミッションコントローラが設けられている。

【 0 1 8 1 】

50

パーミッションコントローラは、どれかのアプリケーションがプレイリスト再生を要求した場合、そのアプリケーションと相互認証を行い、要求元アプリケーションにプレイリストの再生権限があるかどうかを判定する。もしあれば、当該再生をPlaybackControl Engine 3 2に要求し、なければ不許可を示す応答イベントを要求元アプリケーションに出力する。このパーミッションコントローラによる許可判定により、ある配給会社の配給にかかるプレイリストを、別の配給会社の配給にかかるアプリケーションが要求したとしても、かかる要求を不許可にすることができる。そのため、正当権限なきアプリケーションによるプレイリストの無断引用を避けることができる。許可とすべきプレイリストとアプリケーションとの組合せ、不許可とすべきプレイリストとアプリケーションとの組合せは、別途BD-ROMに記録されたPermissionファイルに規定されており、パーミッションコントローラによる判定はこれに基づく。かかるファイルの詳細は本願の主眼ではないので説明を省略する。

【0182】

第3実施形態においてアプリケーションマネージャ37は、現在の再生時点において再生可能なプレイリストを、アプリケーションからの要求に応じて通知する。図42は、このアプリケーションマネージャ37による通知処理の処理手順を示すフローチャートである。本フローチャートでは、アプリケーションの起動中において、再生可能なプレイリストの通知を求める要求(GetPL)をアプリケーションが発行したか否かの監視を行っている(ステップS45)。もし発行されれば、現在の再生時点が属しているTitleを構成するBD-J Objectに、プレイリスト管理テーブルが存在するか否かを判定する(ステップS46)。プレイリストの記述があれば、プレイリスト管理テーブルに記述されたプレイリストを再生可能なプレイリストとして要求元のアプリケーションに通知する(ステップS47)。

【0183】

プレイリストの記述がなければ、プレイリスト再生が不可能な旨を発行元アプリケーションに通知する(ステップS48)。以上が第3実施形態に係るアプリケーションマネージャ37の処理手順である。

続いてプレイリスト再生が要求された場合のアプリケーションマネージャ37の処理について説明する。第3実施形態に係るアプリケーションマネージャ37は、図43のフローチャートに従って処理を行う。

【0184】

図43においてアプリケーションマネージャ37は、プレイリスト再生を要求したアプリケーションが存在するか否かの判定を行っている(ステップS51)。どれかのアプリケーションがプレイリスト再生を要求すれば、要求元アプリケーションにプレイリスト再生の権利が存在するか否かの認証をパーミッションコントローラに行わさせる(ステップS52)。もし再生する権利があれば、PlaybackControl Engine 3 2に再生開始を指示して(ステップS53)、Playback Control Engine 3 2からのサクセス応答を待つ(ステップS54)。

【0185】

かかる再生要求があるとPlayback Control Engine 3 2は、プレイリスト情報の正当性をチェックする。かかる正当性チェックには、プレイリスト情報、Clip情報、AVClipが存在するBD-ROM及びローカルストレージ20において、正当なプレイリストを構成しているかというチェックやプレイリスト情報におけるclip_Information_file_nameにより指定されるClip情報及びAVClipがBD-ROM及びローカルストレージ20に現存するか否かというチェックがある。clip_Information_file_nameにより正しいファイルが参照されていない場合、又は、BD-ROM及びローカルストレージ20から構成される仮想的なパッケージに矛盾があり、正しプレイリストを構成することができない場合、PlaybackControl Engine 3 2はfalseを示す応答を返すことになる。また要求元アプリケーションより、高い起動優先度をもつアプリケーションがそのプレイリストを再生しており、プレイリスト再生を実現するリソースにおいて競合が生じている場合最もPlaybackControl Engine 3 2は、falseを示す応答を返す。

【0186】

以上の過程を経てサクセス応答があれば、プレイリスト再生成功を示すイベントを要求元アプリケーションに出力する(ステップS55)。

サクセス応答がなければ、プレイリスト再生失敗を示すイベントを要求元アプリケーションに出力する(ステップS56)。一方、ステップS52において要求元アプリケーションを再生する権利が要求元アプリケーションになれば、プレイリスト再生不可を示すイベントを要求元アプリケーションに出力する(ステップS57)。

【0187】

以上のように本実施形態によれば、プレイリスト再生可否が各Title毎でバラバラであり、プレイリスト再生の権限をもったアプリケーションやそうでないアプリケーション等様々なアプリケーションがあったとしても、適切なプレイリスト再生を、アプリケーションからの要求に応じて実行することができる。そのため、アプリケーション実行と、プレイリスト再生とを組み合わせた、多彩なコンテンツ表現が可能になる。

【0188】

(第4実施形態)

第1実施形態では、Title開始時において、再生を開始したいプレイリストにAutoPlayを示す再生属性を付与して、AutoPlayPLの再生を再生装置に命じていた。これに対し本実施形態では、アンバウンダリーアプリケーションをBD-ROMに記録しておき、Title開始時にあたって、自動的に再生を開始すべきTitleをアンバウンダリーアプリケーションに選択させる改良に関する。

【0189】

アンバウンダリーアプリケーションは、Playback Control Engine 3.2のような、再生装置におけるレジデントアプリケーションと対等な立場にあるアプリケーションであり、プレイリスト管理テーブルに記述されている複数プレイリスト情報の中から再生装置側のPSR設定値に合致するものを選び、通知するとの処理をPlaybackControl Engine 3.2からの要求に応じて実行する。

【0190】

プレイリスト選択をアンバウンダリーアプリケーションに行わせる場合、かかる選択が必要なTitleにあたっては、プレイリスト管理テーブルにおける再生属性を、全て無指定に設定しておく。これは、“全てが無指定”であることを合図に、PlaybackControl Engine 3.2にプレイリスト選択をタイトルアンバウンダリーアプリケーションに要求させるためである。

【0191】

このアンバウンダリーアプリケーションによる選択は、オーサリング時に規定された選択アルゴリズムに基づく。図44(a)～(c)は、アンバウンダリーアプリケーションに組み込まれた選択アルゴリズムの内容を表形式に示す図である。この表は、PSRの値が取り得る値の範囲と、PSRがそれらの値になった際、再生すべきプレイリストとを対応付けて示している。このうち図44(a)は、パレンタルレベルに基づく選択アルゴリズムの内容を示す。ここでパレンタルレベルは、再生装置においてPSR(14)に示されている。具体的にいうと、PSR(14)には、ユーザの年齢を示す整数値が設定されており、これを再生装置はパレンタルレベルとして解釈する。図44(a)において、PSR(14)がとりうる値は、14歳未満、14歳以上18歳未満、18歳以上という3つの範囲に分けられている。そしてこれらの範囲毎に、再生すべきプレイリストが対応づけられている。アンバウンダリーアプリケーションがかかる選択アルゴリズムによる選択を行えば、PSRの設定値が14歳未満ならPlayList#1が、14以上18歳未満ならPlayList#2が、18歳以上ならPlayList#3がそれぞれ選択されることになる。

【0192】

図44(b)は、Language for Audioに基づく選択アルゴリズムの内容を示す。ここでLanguage for Audioは、再生装置においてPSR(16)に示されている。具体的にいうと、PSR(16)には、整数値が設定されており、これを再生装置は音声再生用の言語設定として解釈

する。図44(b)において、PSR(16)がとりうる値は、英語を示す値、日本語を示す値、その他の値という3つの範囲に分けられている。そしてこれらの範囲毎に、再生すべきプレイリストが対応づけられている。アンバウンダリーアプリケーションがかかる選択アルゴリズムによる選択を行えば、PSR(16)の設定値が英語を示すならPlayList#1が、日本語を示すならPlayList#2が、英語、日本語以外の値ならPlayList#3がそれぞれ選択されることになる。

【0193】

図44(c)は、Player Configuration for Videoに基づく選択アルゴリズムの内容を示す。ここでPlayer Configuration for Videoは、再生装置においてPSR(14)に示されている。具体的にいうと、PSR(14)には、整数値が設定されており、これを再生装置は映像再生用の環境設定として解釈する。図44(c)において、PSR(14)がとりうる値は、解像度525×600TVsystem LetterBox 解像度525×600 TVsystem、1920×1080 TVsystemという3つの範囲に分けられている。そしてこれらの範囲毎に、再生すべきプレイリストが対応づけられている。アンバウンダリーアプリケーションがかかる選択アルゴリズムに従って選択を行えば、PSR(14)の設定値が解像度525×600TVsystem LetterBoxを示すものならPlayList#1が、解像度525×600を示すものならPlayList#2が、TVsystem、1920×1080TVsystemを示すものならPlayList#3がそれぞれ選択されることになる。図44(a)～(c)に示したような選択アルゴリズムは、これらの図に示される条件分岐を、コンピュータ記述言語で記述することにより作成することができる。

【0194】

以上が本実施形態に係る記録媒体の改良である。続いて本実施形態に係る再生装置の改良について説明する。本実施形態での改良点は、主としてアプリケーションマネージャ37、PlaybackControl Engine 32にある。

アプリケーションマネージャ37は、Titleの分岐が生じ、プレイリスト管理テーブルを参照した際、そのプレイリスト管理テーブルにAutoPlayPLが存在するか否かを判定する。AutoPlayPLが無ければ、プレイリスト管理テーブルをPlaybackControl Engine 32に引き渡して、このプレイリスト管理テーブルに記載されているプレイリストのうち、どれかを自動的に再生するようPlaybackControl Engine 32に要求する。

【0195】

PlaybackControl Engine 32は、プレイリスト管理テーブルの引き渡しを受ければ、アンバウンダリーアプリケーションに対し、プレイリスト選択を行うよう要求する。この要求に応じてアンバウンダリーアプリケーションから再生可能なプレイリストのリストが通知されれば、そのPLリストに記載されたプレイリストのうち、PlayItemから引き渡されたプレイリスト管理テーブルに存在するものを判定する。アンバウンダリーアプリケーションにより選択されたプレイリストの中に、プレイリスト管理テーブルに記載されたものがあれば、その再生を自動的に開始する。

【0196】

図45は、タイトルアンバウンダリーアプリケーションにPL選択を行わせる過程を模式的に描いた図である。本図の左側は再生装置におけるソフトウェアのレイヤ構成を示し、本図の右側に、BD-ROMの記録内容を示す。図中の1,2,3,4は、AutoPlayがないプレイリスト管理テーブルを発見した場合の、アプリケーションマネージャ37からの通知(1)、PlaybackControl Engine 32による再生可能プレイリストの問合せ(2)、タイトルアンバウンダリーアプリケーションによるPSR設定値取得(3)、タイトルアンバウンダリーアプリケーションからPlaybackControl Engine 32への再生可能なプレイリストの通知(4)を模式的に描いている。

【0197】

尚、図45においてタイトルアンバウンダリーアプリケーションは、BD-ROM上に記述していたが、これはタイトルアンバウンダリーアプリケーションを、わかり易く記述するための配慮に過ぎない。タイトルアンバウンダリーアプリケーションは、Java(登録商標)アプリケーションであるので、Java(登録商標)仮想マシン36内のワークメモリ54に

においてインスタンスとしてスレッド55により実行されるというのが、現実的な記述になる。

【0198】

以上のように本実施形態によれば、Titleのパウンダリで生存しているようなアプリケーションに、上述したような判定を行わせるので、再生装置側のPlaybackControl Engine 32は、BD-ROMにおける複数プレイリストのうち、再生装置側の状態設定に応じたものはどれであるかを、Title開始時の早い段階で知得することができる。再生属性=AutoPlayのアプリケーションを決めておかなくても、Title開始時に再生を開始すべきプレイリストを決定することができるので、ラングエージクレジットやパレンタルロックといった再生制御をBD-Jモードにおいても実現することができる。

10

【0199】

尚、本実施形態における選択アルゴリズムは、PSRがとり得る値をプレイリストに対応づけたが、再生装置におけるPSRの設定値が、想定外の値であった場合に、再生装置に再生させるプレイリストを予め規定しておいてもよい。

(第5実施形態)

本実施形態は、タイトル切替時において、サウンドミキシングをあらかじめ無効化しておく改良に関する。タイトル切替時には、ファイルsound.bdmvの読み出しが発生することがあり得るので、かかる切替時には、デフォルトとして、サウンドミキシングをあらかじめ無効化しておく。こうすることにより、ファイルsound.bdmvがプリロードバッファ7b内に存在しないことによる、クリックオン再生の途切れを防止することができる。(第6

20

実施形態) 本実施形態は、xletプログラムがJMF(Java(登録商標) Media FrameWork)方式のインターフェイスを利用して、mplsファイルに対するインスタンス(JMFプレーヤインスタンス)を生成することにより、プレイリスト再生を再生装置に命じる場合の実施形態である。

【0200】

mplsファイルとは、プレイリスト情報の実体面を規定するファイルであるので、第2実施形態に示したように、このプレイリスト情報に、Mixing_Onフラグを設けておく。Mixing_Onフラグを、プレイリストに設けた場合、Java(登録商標)アプリケーションがプレイリスト情報の再生をPlaybackControl Engine 32に要求することにより、指定されたプレイリスト情報内にあるMixing_Onフラグに応じて、サウンドミキシングを有効化、あるいは、無効化することができる。

30

【0201】

(第7実施形態)

第1実施形態では、サウンドミキシングの有効化あるいは無効化等をサウンドコントローラ9に指示するための機能をアプリケーションマネージャ37にもたせたが、本実施形態は、かかる機能をJava(登録商標)アプリケーションにもたせる実施形態である。

そこで本実施形態では、かかる機能の呼出をJava(登録商標)アプリケーションから受け付けるためのAPI(Application Interface)をJava(登録商標)プラットフォーム35にもたせる。そうして、サウンドミキシングの有効化あるいは無効化を要求する機能呼出をJava(登録商標)アプリケーションが実行した場合、サウンドミキシングの有効化あるいは無効化等をサウンドコントローラ9に指示する。

40

【0202】

尚、かかる有効化、無効化を指示する機能を、モジュールマネージャ34にもたせてもよい。

(第8実施形態)

再生装置にPCMでのサラウンド音声の伝送が可能な出力形態が存在し、その出力形態がユーザにより選択されている場合、ミキシング音声の再エンコードは不要であるため、そのまま出力することが可能である場合がある。かかる出力形態には、HDMIでの出力形態、アナログでの出力形態がある。

【0203】

50

再生装置にPCMでのサラウンド音声の伝送が可能な出力形態が存在するのか、そして、その出力形態が選択されているのかを再生装置が判断するために、どのような音声出力形態をユーザーが望んでいるか、もしくは再生装置のI/Fの接続形態からどのような音声出力が可能であるかを、本実施形態ではPSRセット23に記述しておく。

以降、本実施形態にかかるPSRの内容について説明する。本実施形態では、PSRセット23のうち、21番目のPSR(PSR21)、22番目のPSR(PSR22)に以下のような改良が加えられている。

【0204】

PSR21は、ミキシング(例えば、2chの追加音声のミキシング)が可能か否かを示す“Audio Mixing Capability”、ミキシング可能なチャンネル数(もしくは、ミキシング可能な個々のチャンネル名)を示す“Audio Mixing Channel Number”、現在ユーザーが選択している音声出力形態に対する最大の出力ch数を示す“Audio Output Channel Number”を示す。

10

【0205】

例えば、音声出力形態としてS/PDIFを選択している場合、サラウンド音声は帯域上の問題から圧縮符号化しないと出力できない。そのためミキシング後に再エンコードする必要があるが、高ビットレートに対応したデジタルI/FであるHDMI等を選択している場合、非圧縮(LPCM)の形式でもそのままサラウンド音声のまま送ることが可能である。したがって、S/PDIF接続時には、ミキシング後にエンコードするか否かでAudio Output Channel Numberの値が変わる。

20

【0206】

PSR22には、再生装置の一属性として、ビデオの出力を行わない再生装置であることを示すAudio Only Player、所定のプロファイルに準じた再生装置であることを示すProfile 1 Player、Profile 2 Playerがある。例えば、Profile 1には対応しながらも、Profile 2に対応しない再生装置はProfile 1 Playerを有効に設定し、Profile 2 Playerを無効に設定する。これらの値は、日本語Lから参照可能なため再生装置のプロファイルに応じた動的な再生経路選択や、メニューの有無などが選択できる。特に、Audio Only Playerが有効になっていた場合には、日本語LでCD再生装置と同等のユーザー操作性を与えるようにプレイリストを再生するようタイトルを作成することが可能であり、この場合、ユーザーはBD-ROMを音楽CDと全く変わりなく操作できる。勿論、同一のディスクでありながら、Audio Only Playerが無効になっている再生装置に対しては、今まで通り映像出力を仮定してGUIベースのメニューなどを表示し、インタラクティブな再生が可能である。このような想定はBDを採用したカーオーディオシステムなどで有効と考えられる。

30

【0207】

以上のようなPSRが追加されたため、本実施形態に係るPlayback Control Engine 32は、PlayItem情報又はプレイリスト情報に対応するMixing_Onフラグが、“ミキシング=有効”を示している場合、PSR21からミキシング可能なかを判定する。そしてPSR21がミキシング可能な場合に、PSR21からミキシング可能なch数を取得し、そのch数以下のch数を持つオーディオストリーム又はサウンドデータを、選択的にサウンドミキサ8、サウンドコントローラ9に出力して、ミキシングを行わせる。

40

【0208】

以上のように本実施形態によれば、どのような音声出力形態をユーザーが望んでいるか、もしくは再生装置のI/Fの接続形態からどのような音声出力が可能であるかを、PSRに記述しておくので、ミキシング出力の可否を、更に詳細に規定しておくことができる。

(備考)

以上の説明は、本発明の全ての実施行為の形態を示している訳ではない。下記(A)(B)(C)(D)・・・の変更を施した実施行為の形態によっても、本発明の実施は可能となる。本願の請求項に係る各発明は、以上に記載した複数の実施形態及びそれらの変形形態を拡張した記載、ないし、一般化した記載としている。拡張ないし一般化の程度は、本発明の技術分野の、出願当時の技術水準の特性に基づく。

50

【0209】

(A)各実施形態に係るBD-ROMは、以下の工程を順次実行することにより、作ることができる。

先ず初めに、BD-ROMをどのような筋書きで再生させるかを定めるかを企画して(企画工程)、動画収録、音声収録等の素材作成を行い(素材作成工程)、企画工程において作成された筋書きから、ボリューム構成情報を作成する(シナリオ作成工程)。

【0210】

ボリューム構成情報とは、抽象的な記述にて、光ディスクの応用層のフォーマットを示す情報である。

その後、ビデオ素材、オーディオ素材、字幕素材、メニュー素材のそれぞれをエンコードすることにより、エレメンタリストリームを得る(素材エンコード工程)。その後、複数のエレメンタリストリームの多重化を行う(多重化工程)。

【0211】

こうして多重化がなされれば、多重化されたストリーム及びボリューム構成情報を、BD-ROMの応用層フォーマットに適合させる作業を行い、BD-ROMのボリューム領域に記録すべきデータの全体像(一般にボリュームデータという)を得る(フォーマッティング工程)。

ここで本発明に係る記録媒体の応用層フォーマットは、プログラミング言語で記述されたクラス構造体のインスタンスであり、BD-ROM規格、BD-J規格に規定された構文に基づいて、クラス構造体のインスタンスを記述することで、BD-Jobject,Clip情報,PlayList情報等を作成することができる。この場合、テーブル形式のデータは、プログラミング言語のfor文を用いて定義することができ、その他、特定の条件下のみ、必要になるようなデータは、if文を用いて定義することができる。

【0212】

こうした適合処理の後、ボリュームデータが得られれば、ボリュームデータを再生してみてシナリオ作成工程の結果が正しいか否かの確認を行う(エミュレーション工程)。このエミュレーション工程では、BD-ROMプレーヤモデルのバッファ状態のシミュレートを行うのが望ましい。

最後にプレス工程を行う。

【0213】

このプレス工程では、ボリュームイメージを物理データ列に変換して、この物理データ列を用いて原盤カッティングを行い、ディスク原盤を作成する。さらにプレス装置によって作成された原盤から、BD-ROMを製造する。この製造は主に、基板成形、反射膜成膜、保護膜コーティング、張り合わせ、レーベルの印刷といった諸工程からなる。

以上の工程を経て、各実施形態に示した記録媒体(BD-ROM)を作ることができる。

【0214】

(B)各実施形態に示したフローチャートや、機能的な構成要素による情報処理は、ハードウェア資源を用いて具体的に実現されていることから、自然法則を利用した技術的思想の創作といえ、“プログラムの発明”としての成立要件を満たす。

・本発明に係るプログラムの生産形態

本発明に係るプログラムは、以下のようにして作ることができる。先ず初めに、ソフトウェア開発者は、プログラミング言語を用いて、各フローチャートや、機能的な構成要素を実現するようなソースプログラムを記述する。この記述にあたって、ソフトウェア開発者は、プログラミング言語の構文に従い、クラス構造体や変数、配列変数、外部関数のコールを用いて、各フローチャートや、機能的な構成要素を具現するソースプログラムを記述する。

【0215】

具体的には、フローチャートにおける繰り返し処理は、上記構文に規定されたfor文等を用いて記述する。判定処理は、上記構文に規定されたif文,switch文等を用いて記述する。デコーダに対する再生制御や、ドライブ装置のアクセス制御等、ハードウェアについての制御は、ハードウェアの製造元から供給される外部関数を呼び出すことにより、記述す

10

20

30

40

50

る。

【0216】

記述されたソースプログラムは、ファイルとしてコンパイラに与えられる。コンパイラは、これらのソースプログラムを翻訳してオブジェクトプログラムを生成する。

コンパイラによる翻訳は、構文解析、最適化、資源割付、コード生成といった過程からなる。構文解析では、ソースプログラムの字句解析、構文解析および意味解析を行い、ソースプログラムを中間プログラムに変換する。最適化では、中間プログラムに対して、基本ブロック化、制御フロー解析、データフロー解析という作業を行う。資源割付では、ターゲットとなるプロセッサの命令セットへの適合を図るため、中間プログラム中の変数をターゲットとなるプロセッサのプロセッサが有しているレジスタまたはメモリに割り付ける。コード生成では、中間プログラム内の各中間命令を、プログラムコードに変換し、オブジェクトプログラムを得る。

10

【0217】

ここで生成されたオブジェクトプログラムは、各実施形態に示したフローチャートの各ステップや、機能的構成要素の個々の手順を、コンピュータに実行させるような1つ以上のプログラムコードから構成される。ここでプログラムコードは、プロセッサのネイティブコード、JAVA（登録商標）バイトコードというように、様々な種類がある。プログラムコードによる各ステップの実現には、様々な態様がある。外部関数を利用して、各ステップを実現することができる場合、この外部関数をコールするコール文が、プログラムコードになる。また、1つのステップを実現するようなプログラムコードが、別々のオブジェクトプログラムに帰属することもある。命令種が制限されているRISCプロセッサでは、算術演算命令や論理演算命令、分岐命令等を組合せることで、フローチャートの各ステップを実現してもよい。

20

【0218】

オブジェクトプログラムが生成されるとプログラマはこれらに対してリンクを起動する。リンクはこれらのオブジェクトプログラムや、関連するライブラリプログラムをメモリ空間に割り当て、これらを1つに結合して、ロードモジュールを生成する。こうして生成されるロードモジュールは、コンピュータによる読み取りを前提にしたものであり、各フローチャートに示した処理手順や機能的な構成要素の処理手順を、コンピュータに実行させるものである。以上の処理を経て、本発明に係るプログラムを作ることができる。

30

【0219】

(C)本発明に係るプログラムは、以下のようにして使用することができる。

(i)組込プログラムとしての使用

本発明に係るプログラムを組込プログラムとして使用する場合、プログラムにあたるロードモジュールを、基本入出力プログラム(BIOS)や、様々なミドルウェア(オペレーションシステム)と共に、命令ROMに書き込む。こうした命令ROMを、制御部に組み込み、CPUに実行させることにより、本発明に係るプログラムを、再生装置の制御プログラムとして使用することができる。

【0220】

(ii)アプリケーションとしての使用

40

再生装置が、ハードディスク内蔵モデルである場合は、基本入出力プログラム(BIOS)が命令ROMに組み込まれており、様々なミドルウェア(オペレーションシステム)が、ハードディスクにプレインストールされている。また、ハードディスクから、システムを起動するためのブートROMが、再生装置に設けられている。

【0221】

この場合、ロードモジュールのみを、過搬型の記録媒体やネットワークを通じて、再生装置に供給し、1つのアプリケーションとしてハードディスクにインストールする。そうすると、再生装置は、ブートROMによるブートストラップを行い、オペレーションシステムを起動した上で、1つのアプリケーションとして、当該アプリケーションをCPUに実行させ、本発明に係るプログラムを使用する。

50

【 0 2 2 2 】

ハードディスクモデルの再生装置では、本発明のプログラムを1つのアプリケーションとして使用しうるので、本発明に係るプログラムを単体で譲渡したり、貸与したり、ネットワークを通じて供給することができる。

(D)本発明に係るシステムLSIの生産・使用行為

システムLSIとは、高密度基板上にベアチップを実装し、パッケージングしたものをいう。複数個のベアチップを高密度基板上に実装し、パッケージングすることにより、あたかも1つのLSIのような外形構造を複数個のベアチップに持たせたものも、システムLSIに含まれる(このようなシステムLSIは、マルチチップモジュールと呼ばれる。)。

【 0 2 2 3 】

ここでパッケージの種別に注目するとシステムLSIには、QFP(クッド フラッド アレイ)、PGA(ピン グリッド アレイ)という種別がある。QFPは、パッケージの四側面にピンが取り付けられたシステムLSIである。PGAは、底面全体に、多くのピンが取り付けられたシステムLSIである。

これらのピンは、ドライブ装置との入出力インターフェイス、リモコン装置との入力インターフェイス、テレビとのインターフェイス、その他、IEEE1394インターフェイスやPCIバスとのインターフェイスとしての役割を担っている。システムLSIにおけるピンには、こうしたインターフェイスの役割が存在するので、システムLSIにおけるこれらのピンに、ドライブ装置等や再生装置の各種回路を接続することにより、システムLSIは、再生装置の中核としての役割を果たす。

【 0 2 2 4 】

システムLSIにパッケージングされるベアチップとは、各実施形態において内部構成図として示した各構成要素の機能を具現する命令ROMやCPU、デコーダLSI等である。

先に“組込プログラムとしての使用”で述べたように、命令ROMには、プログラムにあたるロードモジュールや、基本入出力プログラム(BIOS)、様々なミドルウェア(オペレーションシステム)が書き込まれる。本実施形態において、特に創作したのは、このプログラムにあたるロードモジュールの部分なので、プログラムにあたるロードモジュールを格納した命令ROMを、ベアチップとしてパッケージングすることにより、本発明に係るシステムLSIは生産することができる。

【 0 2 2 5 】

具体的な実装としては、SoC実装やSiP実装が望ましい。SoC(System on chip)実装とは、1チップ上に複数の回路を焼き付ける技術である。SiP(System in Package)実装とは、複数チップを樹脂等で1パッケージにする技術である。以上の過程を経て、本発明に係るシステムLSIは、各実施形態に示した再生装置の内部構成図を基に作ることができる。

尚、上述のようにして生成される集積回路は、集積度の違いにより、IC、LSI、スーパーLSI、ウルトラLSIと呼称されることもある。

【 0 2 2 6 】

さらに、各再生装置の構成要素の一部又は全てを1つのチップとして構成してもよい。集積回路化は、上述したSoC実装、SiP実装に限るものではなく、専用回路又は汎用プロセスで実現してもよい。LSI製造後に、プログラムすることが可能なFPGA(Field Programmable Gate Array)や、LSI内部の回路セルの接続や設定を再構成可能なリコンフィギュラブル・プロセッサを利用することが考えられる。更には、半導体技術の進歩又は派生する技術によりLSIに置き換わる集積回路化の技術が登場すれば、当然、その技術を用いて機能ブロックの集積回路化を行っても良い。例えば、バイオ技術の適応などが可能性としてありうる。

【 0 2 2 7 】

(E)全ての実施形態では、本発明に係る光ディスクをBD-ROMとして実施したが、どのような記録媒体であってもよい。例えば、DVD-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R、CD-RW等の光ディスク、PD、MO等の光磁気ディスクであってもよい。

(F)各実施形態におけるビデオストリームは、BD-ROM規格のAVC clipであったが、DVD-Vid

10

20

30

40

50

eo規格、DVD-Video Recording規格のVOB(VideoObject)であってもよい。VOBは、ビデオストリーム、オーディオストリームを多重化することにより得られたISO/IEC13818-1規格準拠のプログラムストリームである。またAVClipにおけるビデオストリームは、MPEG4やWMV方式であってもよい。更にオーディオストリームは、Linear-PCM方式、Dolby-AC3方式、M P3方式、MPEG-AAC方式、dts方式であってもよい。

【0228】

(G)各実施形態ではMPEG 4 -AVC (H.264やJVTとも呼ばれる) をもとに説明したが、MPEG 2 ビデオストリームであってもよく、また、その他の形式(VC-1等)の画像の場合でも単独でデコード可能な画像であれば、容易に応用可能である。

【産業上の利用可能性】

10

【0229】

本発明に係る記録媒体及び再生装置は、上記実施形態に内部構成が開示されており、この内部構成に基づき量産することが可能なので、資質において工業上利用することができる。このことから本発明に係る記録媒体及び再生装置は、産業上利用可能性を有する。

【図面の簡単な説明】

【0230】

【図1】本発明に係る記録媒体の使用行為を示す図である。

【図2】BD-ROMの内部構成を示す図である。

【図3】拡張子.m2tsが付与されたファイルがどのように構成されているかを模式的に示す図である。

20

【図4】AVClipを構成するTSパケットがどのような過程を経てBD-ROMに書き込まれるかを示す図である。

【図5】AVClipを構成するTSパケットがどのような過程を経てBD-ROMに書き込まれるかを示す図である。

【図6】Clip情報の内部構成を示す図である。

【図7】映画のビデオストリームに対するEP_map設定を示す図である。

【図8】PlayList情報のデータ構造を示す図である。

【図9】AVClipと、PlayList情報との関係を示す図である。

【図10】(a) STN_tableの構成を示す図である。(b) ~ (e) entry - attributeの詳細を示す図である。

30

【図11】PlayList情報の、PlayListMark情報の内部構成を示す図である。

【図12】PlayList情報の、PlayListMark情報によるチャプター位置の指定を示す図である。

【図13】BD-J Objectの内部構成を示す図である。

【図14】アーカイブファイルにより収められているプログラム、データを示す図である。

【図15】ファイルsound.bdmvの構成を示す図である。

【図16】(a) アプリケーション管理テーブルの内部構成を示す図である。(b) アプリケーション管理テーブルを構成する情報要素の意味内容を示す図である。

40

【図17】ディスクコンテンツにおける状態遷移を示す図である。

【図18】(a) BD-ROM全体の時間軸を示す図である。(b) BD-ROM全体の時間軸における構成を示す図である。

【図19】(a) (b) BD-ROM全体の時間軸において、BD-J Objectから特定されるタイトル再生区間を示す図である。

【図20】図19 (b) の時間軸上に規定される、生存区間の典型を示す図である。

【図21】本編タイトル、オンラインショッピングタイトル、ゲームタイトルという3つのタイトルを含むディスクコンテンツを示す図である。

【図22】(a) (b) アプリケーション管理テーブル、生存区間の一例を示す図である。

50

【図 2 3】起動属性がとり得る三態様(Present、AutoRun、Suspend)と、直前タイトルにおけるアプリケーション状態の三態様(非起動、起動中、Suspend)とがとりうる組合せを示す図である。

【図 2 4】(a) プレイリスト管理テーブルの内部構成を示す図である。(b) プレイリスト管理テーブルを構成する情報要素の意味内容を示す図である。

【図 2 5】プレイリスト管理テーブル、アプリケーション管理テーブルにより規定されるタイトルの具体例を示す。

【図 2 6】カレントタイトルがとり得る三態様(プレイリスト管理テーブル無し、プレイリスト管理テーブル有りで尚且つ無指定、プレイリスト管理テーブル有りで尚且つAutoPlay)と、直前タイトルにおけるPLの状態(非再生状態、再生中状態)とがとりうる6通りの組合せを示す図である。

10

【図 2 7】(a) プレイリスト管理テーブル及びアプリケーション管理テーブルの記述例を示す図である。(b) 図 2 7 (a) のように記述されたアプリケーション管理テーブル、プレイリスト管理テーブルによりプレイリスト再生、アプリケーション実行がどのように進行するかを示す図である。

【図 2 8】(a) プレイリスト管理テーブルの他の記述例を示す図である。(b) 図 2 8 (a) のケースに基づくアプリケーション実行及びプレイリスト再生の進行を示す図である。

【図 2 9】SMTの内部構成を示す図である。

【図 3 0】(a) (b) は、STN_Tableにおけるオーディオストリームについてのentry-attributeと、Mixing_Onフラグの設定との因果関係を示す図である。

20

【図 3 1】Mixing_OnフラグがONに設定されたタイトルにおいて、GUIフレームワーク上のボタンが操作された場合の、再生装置による音声再生を示す図である。

【図 3 2】Mixing_OnフラグがOFFに設定されたタイトルにおいて、GUIフレームワーク上のボタンが操作された場合の、再生装置による音声再生を示す図である。

【図 3 3】本発明に係る再生装置の内部構成を示す図である。

【図 3 4】ROM 2 4 に格納されたソフトウェアと、ハードウェアとからなる部分を、レイア構成に置き換えて描いた図である。

【図 3 5】Java (登録商標) 仮想マシン 3 6 の内部構成を示す図である。

【図 3 6】Presentation Engine 3 1 ~ モジュールマネージャ 3 4 による処理を模式化した図である。

30

【図 3 7】BD-J ObjectにおけるPLMTに基づく、アプリケーションマネージャ 3 7 の処理を示す図である。

【図 3 8】BD-J ObjectにおけるSMTに基づく、アプリケーションマネージャ 3 7 の処理を示す図である。

【図 3 9】アプリケーションマネージャ 3 7 による処理手順を示すフローチャートである。

【図 4 0】(a) 第 2 実施形態に係るプレイリスト情報の内部構成を示す図である。(b) は、PlayItem情報内に設けられた、Mixing_Onフラグの内容を示す図である。

【図 4 1】Playback Control Engine 3 2 によるプレイリスト再生手順を示すフローチャートである。

40

【図 4 2】アプリケーションマネージャ 3 7 による通知処理の処理手順を示すフローチャートである。

【図 4 3】第 3 実施形態に係るアプリケーションマネージャ 3 7 の処理手順を示す図である。

【図 4 4】(a) パレンタルレベルに基づく選択アルゴリズムの内容を示す。(b) Language for Audioに基づく選択アルゴリズムの内容を示す。(c) Player Configuration for Videoに基づく選択アルゴリズムの内容を示す。

【図 4 5】タイトルアンバウンダリーアプリケーションにプレイリスト選択を行わせる過程を模式的に描いた図である。

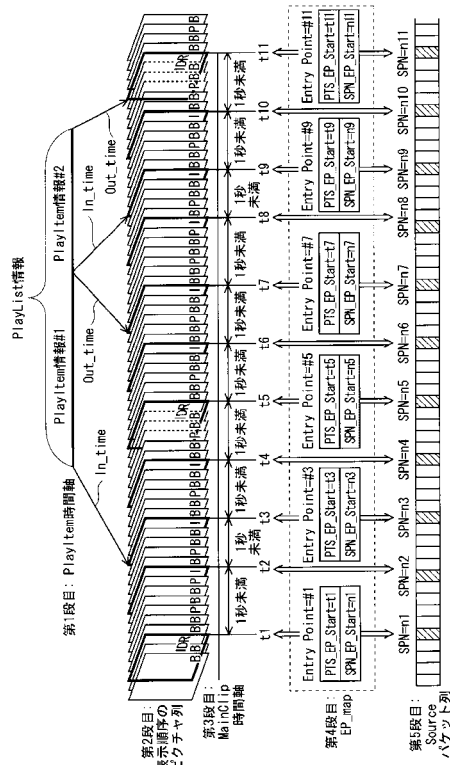
50

【符号の説明】

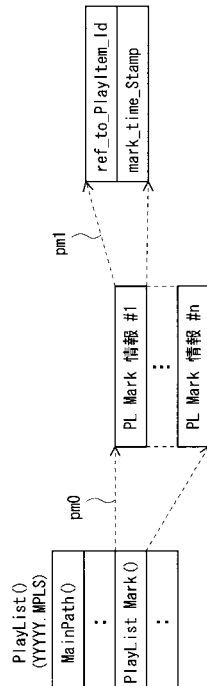
【0231】

1	BD-ROMドライブ	
2	リードバッファ	
3	デマルチプレクサ	
4	ビデオデコーダ	
5	ビデオプレーン	
6	サウンドプロセッサ	
7	サウンドプロセッサ	
8	ミキサー	10
9	サウンドコントローラ	
10	D/Aコンバータ	
11	Interactive Graphicsデコーダ	
12	Interactive Graphicsプレーン	
13	Presentation Graphicsデコーダ	
14	Presentation Graphicsプレーン	
15	JPEGデコーダ	
16	Stillプレーン	
17	合成部	
18	STC-delta付加部	20
19	ATC-delta付加部	
20	ローカルストレージ	
21	命令ROM	
22	ユーザイベント処理部	
23	PSRセット	
24	CPU	
25	シナリオメモリ	
26	ローカルメモリ	

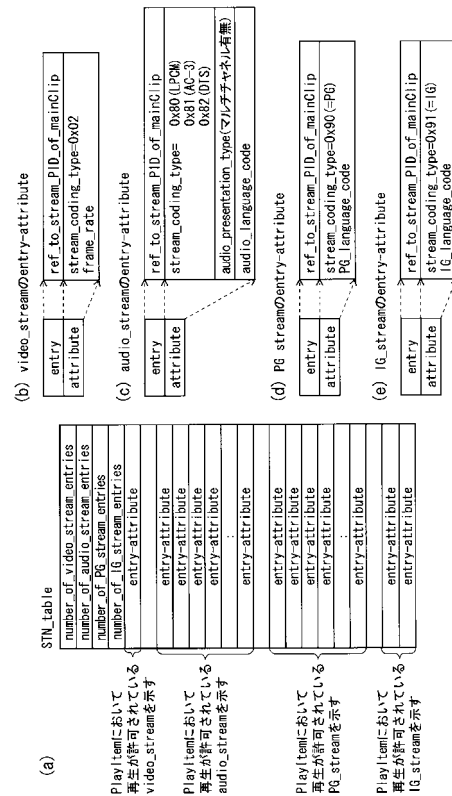
【図 9】



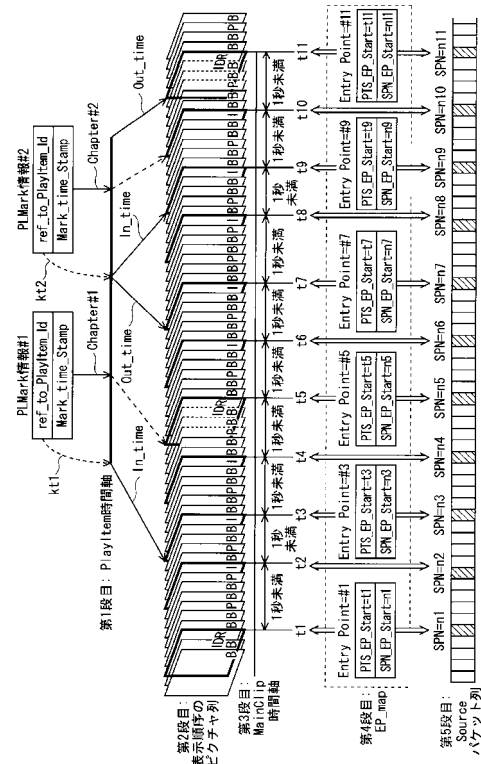
【図 11】



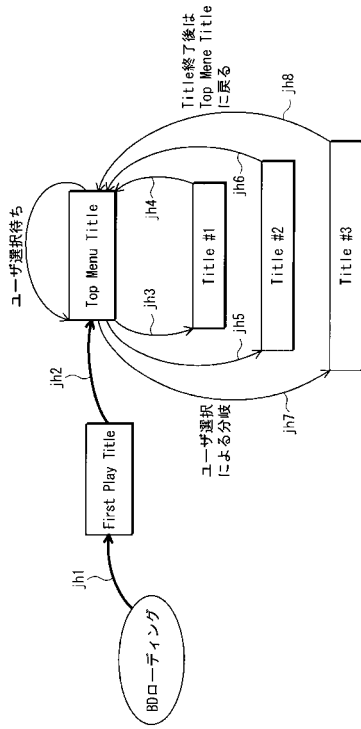
【図 10】



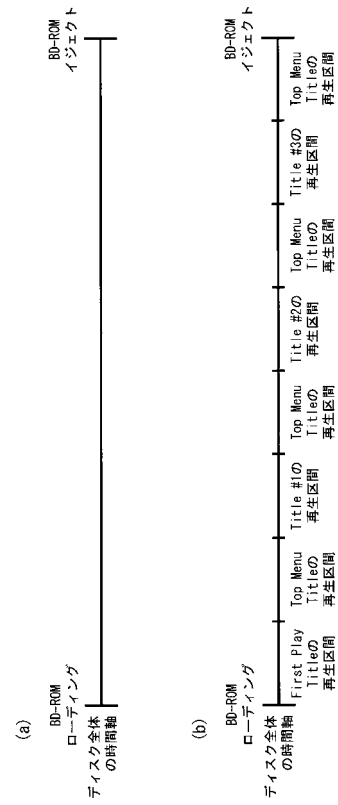
【図 12】



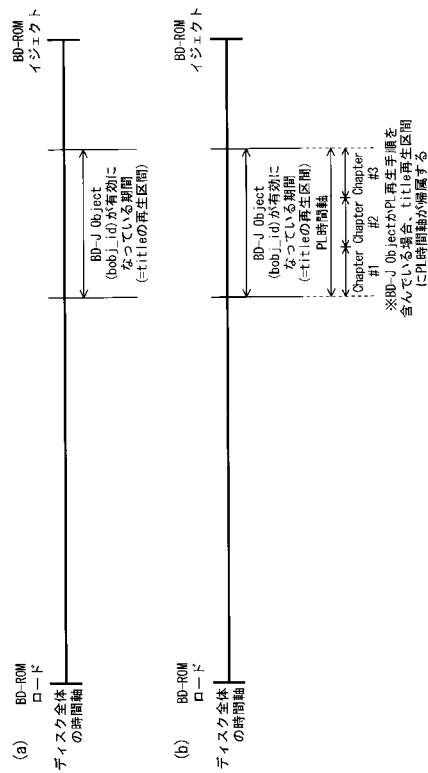
【図 17】



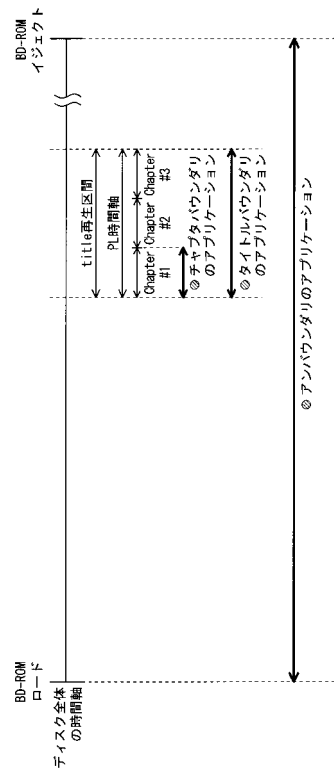
【図 18】



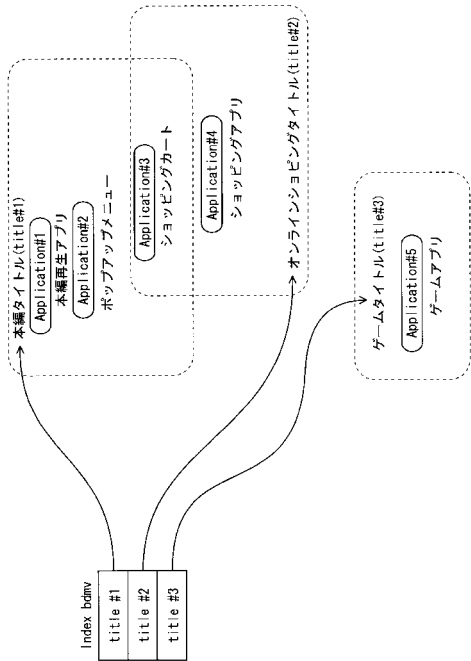
【図 19】



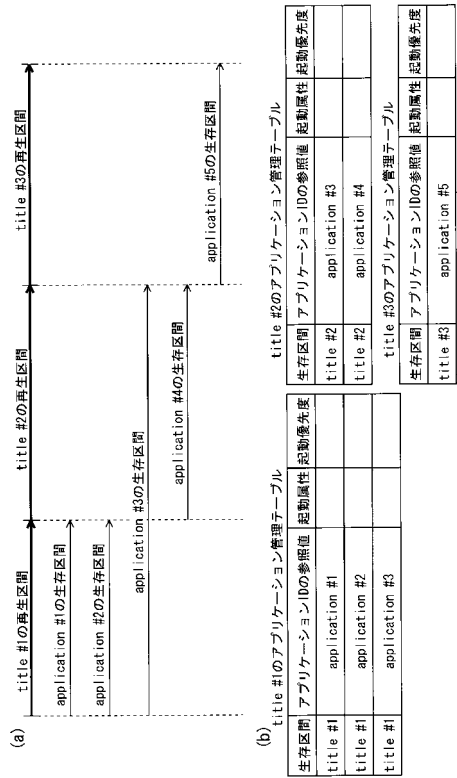
【図 20】



【図 2 1】



【図 2 2】



【図 2 3】

起動属性によるアプリケーション状態の変化			
直前タイトルにおけるアプリケーションの状態	起動属性		
	Present	AutoRun	Suspend
	非起動	アプリケーションを起動する	何もせず、状態継続
	起動中	何もせず、状態継続	サスペンドする
Suspend	レジュームする	レジュームする	何もせず、状態継続

【図 2 4】

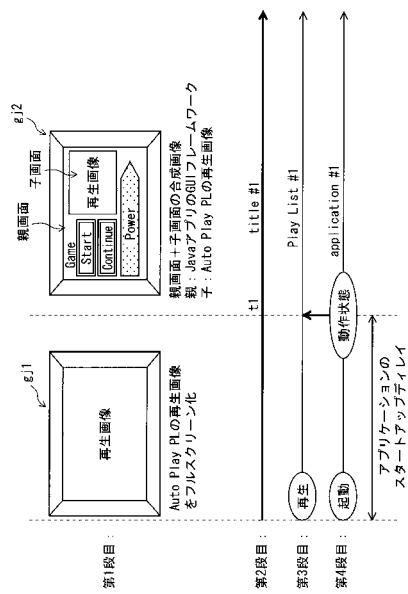
(a) Play List Management Table (PLMT)

PL_id_ref
Playback_Attribute

(b) Play List Management Table (PLMT)

PL_id_ref	Playback_Attribute
MPLSファイルのファイル名における5桁の数値である“Play List ID”	・Auto Play ・Present (無指定)の何れかである“再生属性”

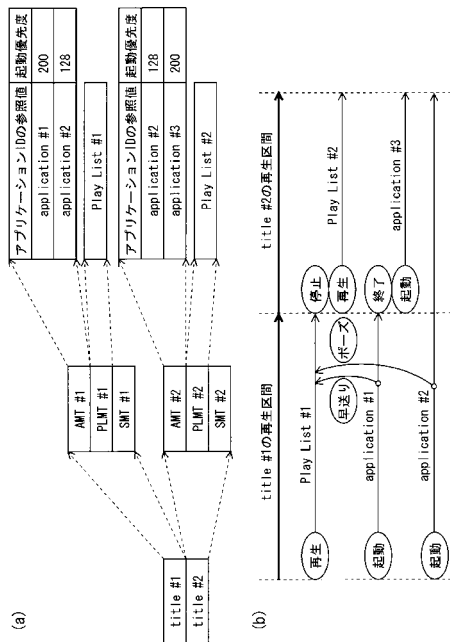
【 図 2 5 】



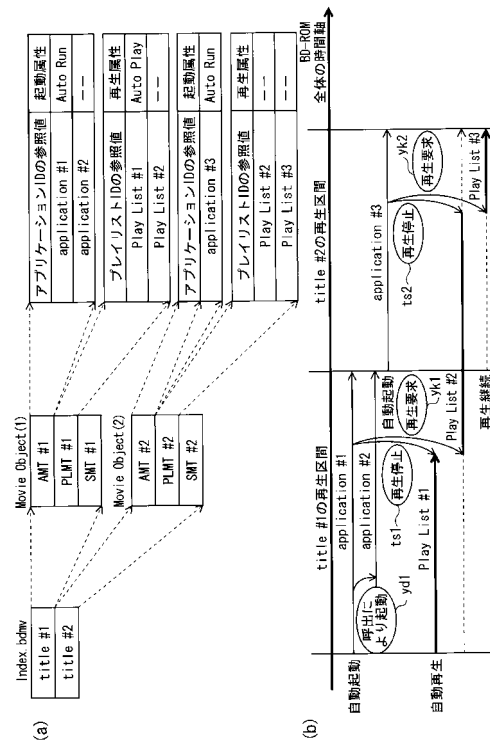
【 図 2 6 】

	分岐先titleにPLMT有り
	再生属性: Auto Play 再生属性: Present
分岐元titleは再生中状態	再生停止 状態継続
分岐元titleは非再生状態	状態継続 自動再生開始 状態継続

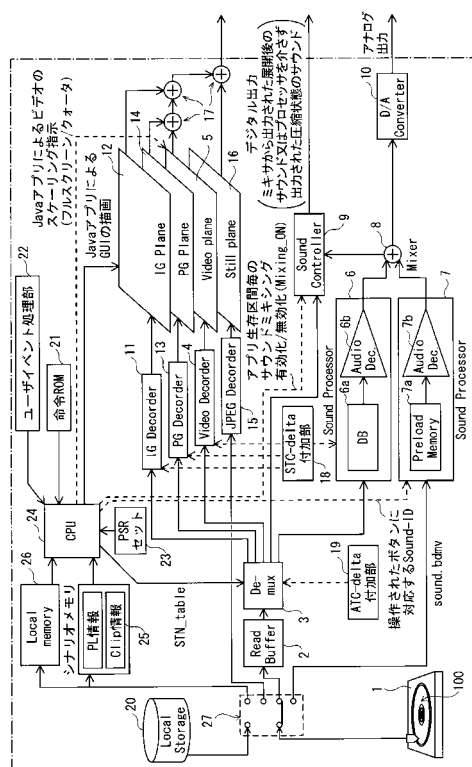
【 図 2 7 】



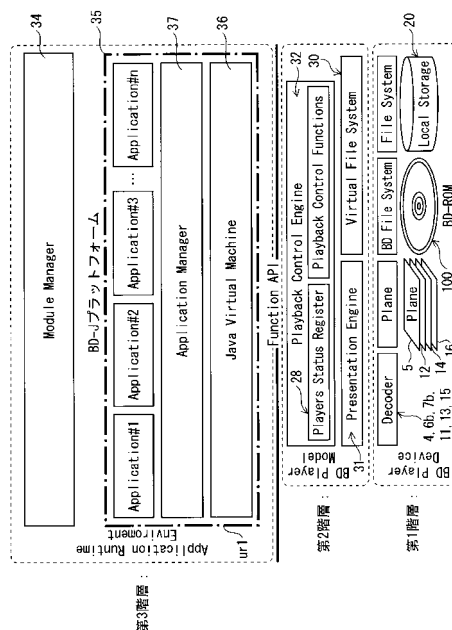
【 図 2 8 】



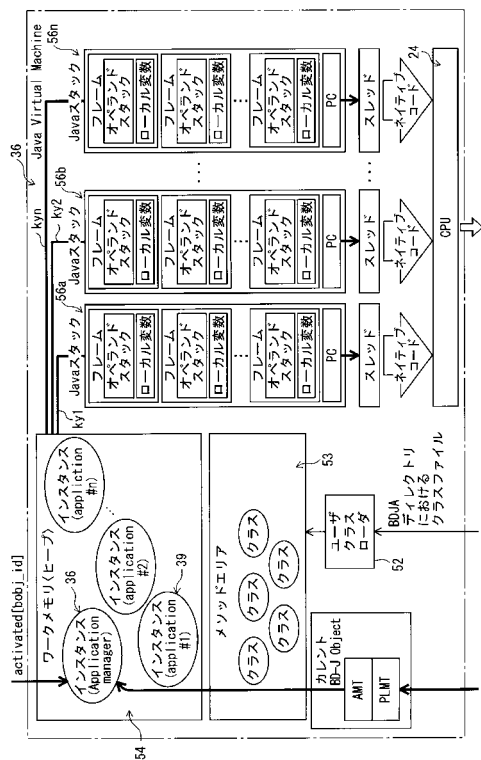
【 ㊦ 3 3 】



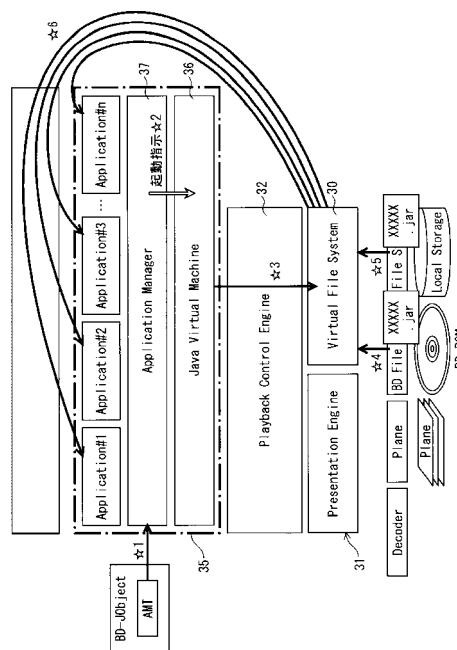
【 ㊦ 3 4 】



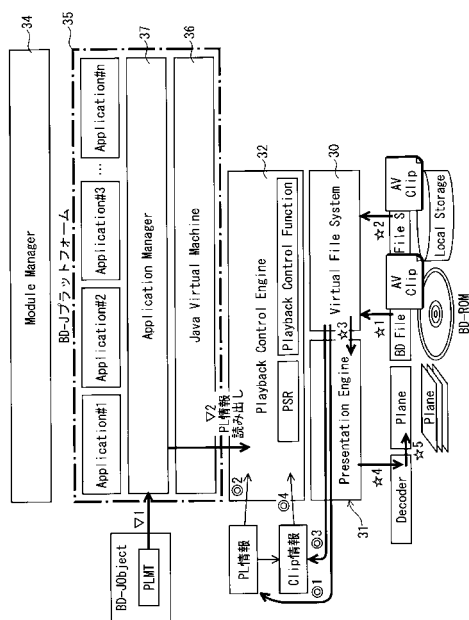
【 ㄨ 3 5 】



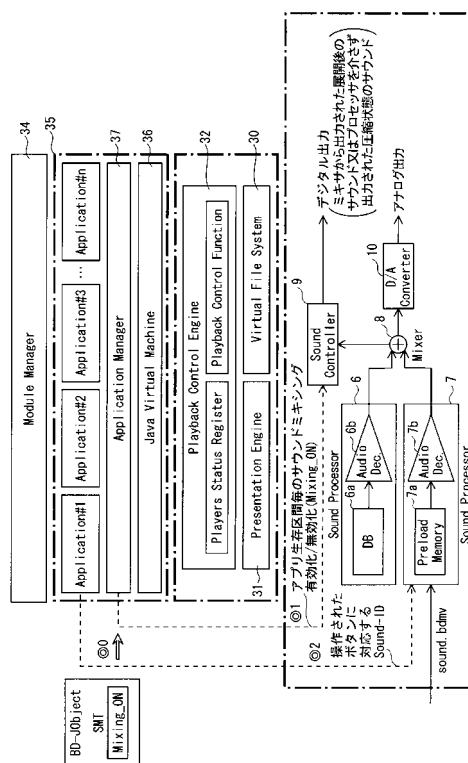
【 ㄨ 3 6 】



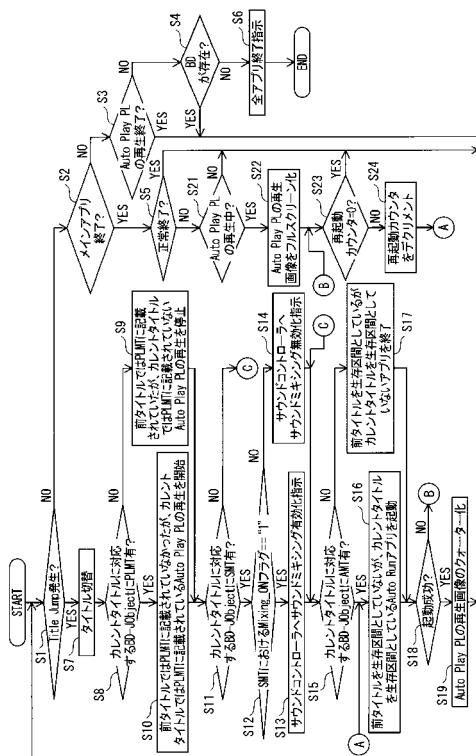
【 図 3 7 】



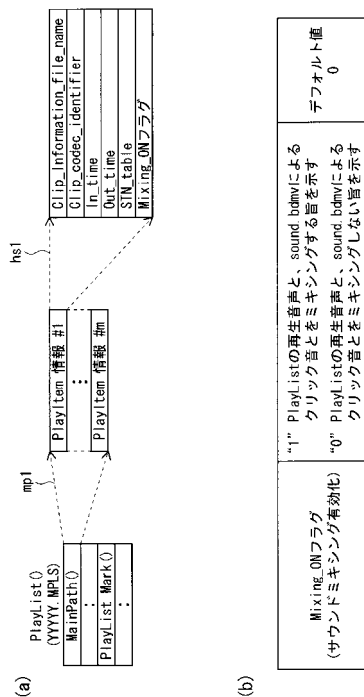
【 ㄨ 3 8 】



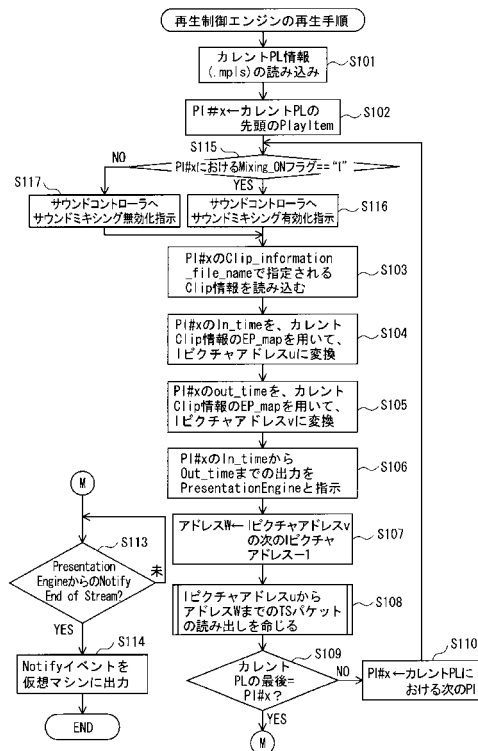
【 図 3 9 】



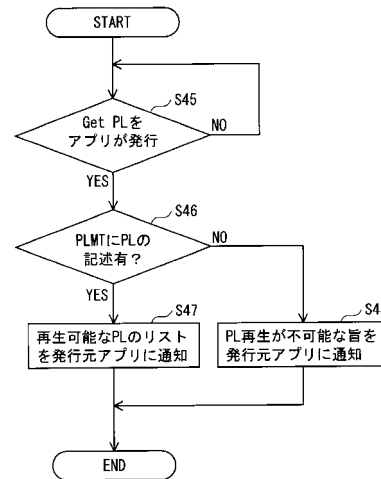
【 図 4 0 】



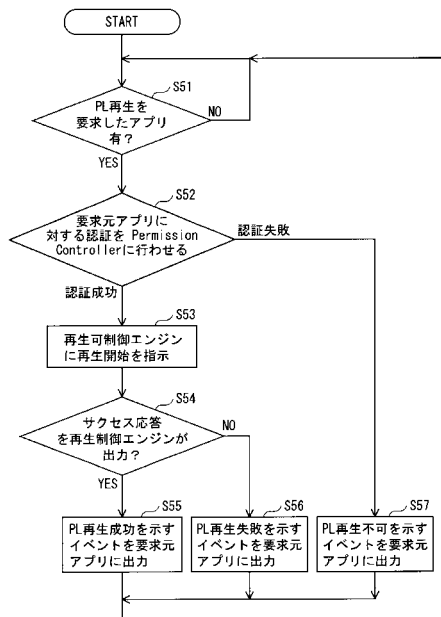
【図 4 1】



【図 4 2】



【図 4 3】



【図 4 4】

(a)

Parental Levelによる選択アルゴリズム

PSR range	Play List_id_ref
PSR (13) < 14	Play List #1
14 ≤ PSR (13) < 18	Play List #2
18 ≤ PSR (13)	Play List #3

(b)

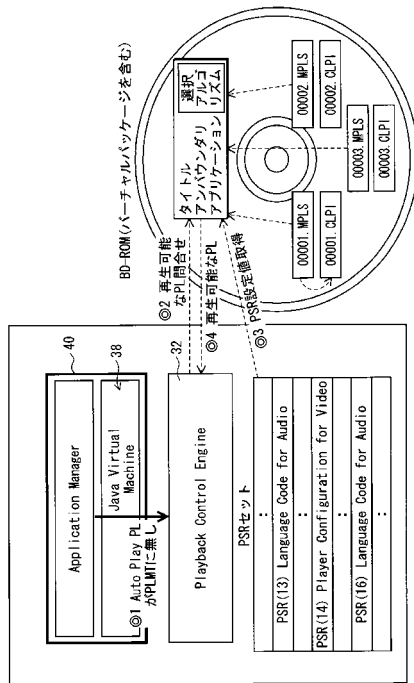
Language Code for Audioによる選択アルゴリズム

PSR range	Play List_id_ref
PSR (16)=English	Play List #1
PSR (16)=Japanese	Play List #2
PSR (16)=others	Play List #3

(c)

Player Configuration for Videoによる選択アルゴリズム

PSR range	Play List_id_ref
PSR (14)=525×60 TV system 4:3 Letter Box	Play List #1
PSR (14)=525×60 TV system 16:9	Play List #2
PSR (14)=1920×1080TV system	Play List #4



フロントページの続き

(51)Int.Cl. F I
H 0 4 N 5/85 Z

(72)発明者 岡田 智之
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

(72)発明者 矢羽田 洋
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

審査官 若林 治男

(56)参考文献 特開平 0 9 - 0 1 8 8 3 3 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G 1 1 B 2 7 / 0 0
G 1 1 B 2 7 / 0 3 4
H 0 4 N 5 / 8 5
H 0 4 N 5 / 9 1
H 0 4 N 5 / 9 3