

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-152902

(P2008-152902A)

(43) 公開日 平成20年7月3日(2008.7.3)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G 1 1 B 20/12 (2006.01)</b>	G 1 1 B 20/12	5 C 0 5 2
<b>G 1 1 B 20/10 (2006.01)</b>	G 1 1 B 20/10 3 2 1 Z	5 C 0 5 3
<b>H O 4 N 5/85 (2006.01)</b>	G 1 1 B 20/12 1 0 3	5 D 0 4 4
<b>H O 4 N 5/93 (2006.01)</b>	H O 4 N 5/85 Z	5 D 0 7 7
<b>G 1 1 B 27/10 (2006.01)</b>	H O 4 N 5/93 Z	5 D 1 1 0

審査請求 有 請求項の数 10 O L (全 63 頁) 最終頁に続く

(21) 出願番号	特願2007-292333 (P2007-292333)	(71) 出願人	000005821 松下電器産業株式会社
(22) 出願日	平成19年11月9日 (2007.11.9)		大阪府門真市大字門真1006番地
(62) 分割の表示	特願2005-289495 (P2005-289495) の分割	(74) 代理人	100090446 弁理士 中島 司朗
原出願日	平成16年11月9日 (2004.11.9)	(74) 代理人	100072442 弁理士 松村 修治
(31) 優先権主張番号	特願2003-380464 (P2003-380464)	(74) 代理人	100125597 弁理士 小林 国人
(32) 優先日	平成15年11月10日 (2003.11.10)	(72) 発明者	岩本 啓明 大阪府門真市大字門真1006番地 松下電器産業株式会社内
(33) 優先権主張国	日本国 (JP)	(72) 発明者	池田 航 大阪府門真市大字門真1006番地 松下電器産業株式会社内
(31) 優先権主張番号	特願2004-261376 (P2004-261376)		
(32) 優先日	平成16年9月8日 (2004.9.8)		
(33) 優先権主張国	日本国 (JP)		

最終頁に続く

(54) 【発明の名称】 記録媒体、再生装置。

(57) 【要約】 (修正有)

【課題】記録媒体に対する制御を実現するようなアプリケーションが異常終了した場合、又は、起動に失敗した場合の、Fail Safeを提供する。

【解決手段】 BD-ROMには、AVC lip及びプレイリスト情報からなるプレイリストと、アプリケーションと、BD-J Objectとが記録されている。アプリケーションは、仮想マシン向けプログラミング言語で記述されたプログラムであり、仮想マシンによる実行が可能となる生存区間が、予め規定されている。前記BD-J Objectは、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグを含み、前記再生継続フラグは所定の値に設定されることによりアプリケーションと同時になされた再生制御を、アプリケーションの異常終了後も再生装置に継続させる旨を示す。

【選択図】 図 2 0

(a) Application Management Table (AMT)

life_cycle
apli_id_ref
run_attribute
run_priority

(b) Application Management Table (AMT)

life_cycle	title,Pl_Chapterを用いて表されたアプリケーションの"生存区間"	apli_id_ref	JARのファイル名に付与された5桁の連続ZZZZZである"アプリケーションID"の参照値	run_attribute	・ Auto Run ・ Present(無指定) ・ Suspend の何れかである"起動属性"	run_priority	0~255の値をとる"起動優先度"
------------	---	-------------	--	---------------	--	--------------	-------------------

## 【特許請求の範囲】

## 【請求項 1】

アプリケーションと、デジタルストリームと、前記アプリケーション及び前記デジタルストリームに関連するタイトルと、管理情報とが記録された記録媒体であって、

前記アプリケーションは、

仮想マシン向けプログラミング言語で記述されたプログラムであり、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシンのワークメモリ上に前記アプリケーションが読み出され、前記仮想マシンによる前記アプリケーションの実行が可能となる生存区間が、予め規定されており、

前記管理情報は、

前記生存区間において、前記アプリケーションの実行とともに行うべき前記タイトルに関連するデジタルストリームの再生制御を示す情報を含み、

管理情報は更に、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグを含み、

前記再生継続フラグは所定の値に設定されることにより

前記アプリケーションの実行とともに行うべき前記タイトルに関連するデジタルストリームの再生制御を、アプリケーションの異常終了後も再生装置に継続させる旨を示すことを特徴とする記録媒体。

## 【請求項 2】

記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録されたデジタルストリームであって、前記記録媒体に記録されたタイトルに関連するデジタルストリームを再生する再生制御エンジン部と、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に実行させ、前記仮想マシン部による実行とともに、前記記録媒体に記録された管理情報に基づき前記タイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に実行させるアプリケーションマネージャ部とを備え、

前記管理情報は更に、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグを含み、

前記アプリケーションマネージャ部は更に、前記再生継続フラグが所定の値に設定されているか否かの判定を行い、前記再生継続フラグが所定の値に設定されていれば、前記アプリケーションとともに行うべき前記タイトルに関連するデジタルストリームの再生制御を、アプリケーションの異常終了後も再生制御エンジン部に継続させる

ことを特徴とする再生装置。

## 【請求項 3】

記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録されたデジタルストリームであって、前記記録媒体に記録されたタイトルに関連するデジタルストリームを再生する再生制御エンジン部と

を有したコンピュータが読み込むことができるプログラムであって、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に実行させるステップと、

前記仮想マシンによる実行とともに、前記記録媒体に記録された管理情報に基づき前記タイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に実行させるステップと、

10

20

30

40

50

前記記録媒体に記録された管理情報に含まれる再生継続フラグであって、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグが所定の値に設定されているか否かの判定を行い、前記再生継続フラグが所定の値に設定されていれば、前記アプリケーションとともに行うべき前記タイトルに関連するデジタルストリームの再生制御を、アプリケーションの異常終了後も再生制御エンジン部に継続させるステップとをコンピュータに実行させることを特徴とするプログラム。

【請求項 4】

記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録されたデジタルストリームであって、前記記録媒体に記録されたタイトルに関連するデジタルストリームを再生する再生制御エンジン部とを有したコンピュータに対する再生方法であって、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に実行させるステップと、

前記仮想マシン部による実行とともに、前記記録媒体に記録された管理情報に基づき前記タイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に実行させるステップと、

前記記録媒体に記録された管理情報に含まれる再生継続フラグであって、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグが所定の値に設定されているか否かの判定を行い、前記再生継続フラグが所定の値に設定されていれば、前記アプリケーションとともに行うべき前記タイトルに関連するデジタルストリームの再生制御を、アプリケーションの異常終了後も再生制御エンジン部に継続させるステップとをコンピュータに実行させることを特徴とする再生方法。

【請求項 5】

記録媒体に記録されたデジタルストリームであって、前記記録媒体に記録されたタイトルに関連するデジタルストリームを再生する再生装置に組み込むことができるシステム集積回路であって、

前記記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に実行させ、前記仮想マシン部による実行とともに、前記記録媒体に記録された管理情報に基づき前記タイトルに関連するデジタルストリームの再生を、実行させるアプリケーションマネージャ部とを備え、

前記記録媒体に記録された管理情報に含まれる再生継続フラグであって、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグが所定の値に設定されているか否かの判定を行い、前記再生継続フラグが所定の値に設定されていれば、前記アプリケーションとともに行うべき前記タイトルに関連するデジタルストリームの再生制御を、アプリケーションの異常終了後も再生制御エンジン部に継続させることを特徴とするシステム集積回路。

【請求項 6】

アプリケーションと、デジタルストリームと、前記アプリケーション及び前記デジタルストリームに関連するタイトルと、管理情報とが記録された記録媒体であって、

前記アプリケーションは、

仮想マシン向けプログラミング言語で記述されたプログラムであり、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシンのワークメモリ上に前記アプリケーションが読み出され、前記仮想マシンによる前記アプリケーション

10

20

30

40

50

の実行が可能となる生存区間が、予め規定されており、

前記管理情報は、

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、前記記録媒体に記録されたタイトルの再生区間における再生経路を規定する複数の再生経路情報と、

前記生存区間において、前記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に関する情報を含む

ことを特徴とする記録媒体。

【請求項 7】

記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録された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、前記記録媒体に記録されたタイトルの再生区間における再生経路を規定する再生経路情報に従って前記記録媒体に記録されたデジタルストリームを再生する再生制御エンジン部と、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に実行させ、前記実行中のアプリケーションが選択した再生経路情報に従って前記記録媒体に記録されたタイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に実行させるアプリケーションマネージャ部とを備える

ことを特徴とする再生装置。

【請求項 8】

記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録された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、前記記録媒体に記録されたタイトルの再生区間における再生経路を規定する再生経路情報に従って前記記録媒体に記録されたデジタルストリームを再生する再生制御エンジン部と

を有したコンピュータが読み込むことができるプログラムであって、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に実行させるステップと、

前記実行中のアプリケーションが選択した再生経路情報に従って前記記録媒体に記録されたタイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に実行させるステップと

をコンピュータに実行させることを特徴とするプログラム。

【請求項 9】

記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録された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、前記記録媒体に記録されたタイトルの再生区間における再生経路を規定する再生経路情報に従って前記記録媒体に記録されたデジタルストリームを再生する再生制御エンジン部と

を有したコンピュータが読み込むことができる再生方法であって、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーシ

10

20

30

40

50

ョンを前記ワークメモリに読み出して、前記仮想マシン部を実行させるステップと、  
前記実行中のアプリケーションが選択した再生経路情報に従って前記記録媒体に記録されたタイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に行わせるステップと

をコンピュータに実行させることを特徴とする再生方法。

【請求項10】

記録媒体に記録された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、前記記録媒体に記録されたタイトルの再生区間における再生経路を規定する再生経路情報に従って前記記録媒体に記録されたデジタルストリームを再生する再生装置に組み込むことができるシステム集積回路であって、

10

前記記録媒体からワークメモリ上に読み出されたアプリケーションを実行する仮想マシン部と、

前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシン部のワークメモリ上に前記アプリケーションが読み出され、前記仮想マシン部による前記アプリケーションの実行が可能となる生存区間が到来すれば、前記記録媒体に記録されたアプリケーションを前記ワークメモリに読み出して、前記仮想マシン部に行わせる、前記実行中のアプリケーションが選択した再生経路情報に従って前記記録媒体に記録されたタイトルに関連するデジタルストリームの再生を、前記再生制御エンジン部に行わせるアプリケーションマネージャ部とを備える

ことを特徴とするシステム集積回路。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、仮想マシンを対象にしたアプリケーションの実行を制御する、アプリケーション制御の技術分野に属する発明であり、本制御技術をBD-ROM等の映画作品頒布のための記録媒体、BD-ROMを対象とした再生装置等に応用する場合の応用技術に深く係る。

【背景技術】

【0002】

Java（登録商標）プログラミング等、仮想マシンを対象にしたアプリケーション制御技術は、パソコンソフトの業界において深く浸透している。パソコンソフトから更に発展して、現在、Blu-ray Disc Read Only Memory(BD-ROM)用再生装置における再生制御をJava（登録商標）プログラミングで実現するための様々な工夫が検討されている。

30

同様の再生装置についての先行技術としては、以下の特許文献1に記載されたものが知られている。

【特許文献1】特許2813245号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

ところで、Java（登録商標）プログラミングで作成されたアプリケーションの動作というのは、リソースの使用状況やCPUの負荷により変動を受ける不安定なものである。リソースが枯渇していたため、アプリケーションの起動に失敗したり、アプリケーションが異常終了したりして装置がブラックアウトするのは、パソコンソフトの業界では”よくあること”で済まされるかもしれないが、BD-ROM再生装置の製造に携わる民生機器の分野では品質問題になりかねない。品質問題への波及がありうるので、メーカ各社は、Java（登録商標）プログラミングによる装置制御の実現に二の足を踏むことが多かった(注；ブラックアウトとは、装置のソフトウェアがフリーズして表示画面が真っ暗になる状態をいう)。

40

【0004】

本発明の目的は、記録媒体に対する制御を実現するようなアプリケーションが異常終了した場合、又は、起動に失敗した場合の、Fail Safeを実現することができる記録媒体を

50

提供する。

【課題を解決するための手段】

【0005】

上記目的を達成するため、本発明に係る記録媒体は、アプリケーションと、デジタルストリームと前記アプリケーション及び前記デジタルストリームに関連するタイトルと、管理情報とが記録された記録媒体であって、前記アプリケーションは、仮想マシン向けプログラミング言語で記述されたプログラムであり、前記記録媒体に記録されたタイトルの再生区間において、前記仮想マシンのワークメモリ上に前記アプリケーションが読み出され、前記仮想マシンによる前記アプリケーションの実行が可能となる生存区間が、予め規定されており、前記管理情報は、前記生存区間において、前記アプリケーションの実行とともに  
10 行うべき前記タイトルに関連するデジタルストリームの再生制御を示す情報を含み、前記管理情報は更に、アプリケーションの異常終了時に行うべきリカバリー処理を規定する再生継続フラグを含み、前記再生継続フラグは所定の値に設定されることにより前記アプリケーションの実行とともに  
10 行うべき前記タイトルに関連するデジタルストリームの再生制御を、アプリケーションの異常終了後も再生装置に継続させる旨を示すことを特徴としている。

【発明の効果】

【0006】

本発明に係る記録媒体によれば、アプリケーション実行と同時に  
20 行うべきデジタルストリームの再生制御が生存区間毎に規定されている。そのため、アプリケーションの起動に失敗したり、アプリケーションの実行の途中でたとえアプリケーションが異常終了したとしても、同時になされているデジタルストリーム再生を継続することにより、“何かが写っている状態”をもたらすことができる。これにより、装置がブラックアウトしてしまうという最悪ケースを回避することができるので、装置を製造するメーカーに、最低限度の安心感を与えることができる。

【0007】

こうした安心感が与えられるので、品質問題への波及に神経を尖らせているメーカー各社  
30 に対しても、Java（登録商標）アプリケーションによる再生装置制御の開発を強く後押しすることができる。こうした強い後押しにより、再生装置の低価格化、多様化が進めば、BD-ROMコンテンツの充実化を図ることができるので、コンテンツ関連産業の発達を強く牽引することができる。

【発明を実施するための最良の形態】

【0008】

(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。  
40

【0009】

このうちリモコン400には、再生開始(Play)、再生停止(Stop)、一時停止(Pause On)、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(Backward Play(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)を受け付けるキーや、メニュー操作時にあたってのフォーカス移動操作を受け付けるMoveUpキー、MoveDownキー、MoveRightキー、MoveLeftキー、メニュー表示操作を受け付けるPop-upキー、数値入力を受け付けるNumericキーが存在する。

【0010】

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。  
50

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。

【0011】

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,BD-J Object.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、BDJAディレクトリと呼ばれる4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,00003mpls)がある。

10

【0012】

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。

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,00003jar)がある。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。

【0013】

20

本図において拡張子.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には、MainClip、SubClipといった種別がある。MainClipは、ビデオストリーム、オーディオストリーム、字幕を構成するプレゼンテーショングラフィクスストリーム(PGストリーム)、メニューを構成するインタラクティブグラフィクスストリーム(IGストリーム)というような複数エレメンタリストリームを多重化することで得られたデジタルストリームである。

【0014】

SubClipは、オーディオストリーム、グラフィクスストリーム、テキスト字幕ストリーム(TextSTStream)等、1つのエレメンタリストリームのみにあたるデジタルストリームである。

30

拡張子"clpi"が付与されたファイル(00001.clpi,00002.clpi,00003.clpi・・・)は、AVClipのそれぞれに1対1に対応する管理情報である。管理情報故に、Clip情報は、AVClipにおけるストリームの符号化形式、フレームレート、ビットレート、解像度等の情報や、GOPの先頭位置を示すEP\_mapをもっている。

【0015】

拡張子"mpls"が付与されたファイル(00001.mpls,00002.mpls,00003.mpls・・・)は、プレイリスト情報を格納したファイルである。プレイリスト情報は、AVClipを参照してプレイリストを定義する情報である。図3は、PL情報の構成を示す図であり、本図の左側に示すように、プレイリスト情報は、『MainPath情報』、『PLMark情報』、『SubPath情報』から構成される。

40

【0016】

MainPath情報(MainPath())は、破線の矢印mp1に示すように複数のPlayItem情報(PlayItem())からなる。PlayItemとは、1つ以上のAVClip時間軸上において、In\_Time,Out\_Timeを指定することで定義される再生区間である。PlayItem情報を複数配置させることで、複数再生区間からなるプレイリスト(PL)が定義される。図中の破線mp2は、PlayItem情報の内部構成をクローズアップしている。本図に示すようにPlayItem情報は、対応するAVClipを示す『Clip\_information\_file\_name』と、『In\_time』と、『Out\_time』とからなる。図4は、AVClipと、PLとの関係を示す図である。第1段目はAVClipがもつ時間軸を示し、第2段目は、PLがもつ時間軸を示す。PL情報は、PlayItem#1,#2,#3という3つのPlayItem情報を含んでおり、これらPlayItem#1,#2,#3のIn\_time,Out\_timeにより、3つの再生区間

50

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2段目に示すPL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。

#### 【0017】

AVClipに対する指定は、原則1つであるが、複数AVClipに対する一括指定もあり得る。この一括指定は、PlayItem情報における複数のClip\_Information\_file\_nameによりなされる。図5は、4つのClip\_Information\_file\_nameによりなされた一括指定を示す図である。本図において第1段目～第4段目は、4つのAVClip時間軸(AVClip#1,#2,#3,#4の時間軸)を示し、第5段目は、PL時間軸を示す。PlayItem情報が有する、4つのClip\_Information\_file\_nameにて、これら4つの時間軸が指定されている。こうすることで、PlayItemが有するIn\_time,Out\_timeにより、択一的に再生可能な4つの再生区間が定義されることになる。これにより、PL時間軸には、切り換え可能な複数アングル映像からなる区間(いわゆるマルチアングル区間)が定義されることになる。

10

#### 【0018】

PLmark情報(PLmark())は、PL時間軸のうち、任意の区間を、チャプターとして指定する情報である。図6は、PLmark情報の内部構成を示す図であり、本図の引き出し線pm1に示すようにPLmark情報は、『ref\_to\_PlayItem\_Id』と、『Mark\_time\_stamp』とを含む。図7は、PLmarkによるチャプター定義を示す図である。本図において第1段目は、AVClip時間軸を示し、第2段目はPL時間軸を示す。図中の矢印pk1,2は、PLmarkにおけるPlayItem指定(ref\_to\_PlayItem\_Id)と、一時点の指定(mark\_time\_stamp)とを示す。これらの指定によりPL時間軸には、3つのチャプター(Chapter#1,#2,#3)が定義されることになる。以上がPLmarkについての説明である。続いてSubPath情報について説明する。

20

#### 【0019】

SubPath情報(SubPath())は、SubClipの時間軸上にIn\_Time,Out\_Timeを指定することで1つ以上の再生区間を定義する情報であり、図8に示す内部構成を有している。本図に示すようにSubPath情報は、破線の引き出し線sh1に示すように複数のSub-PlayItem情報(SubPlayItem())からなる。破線sh2を用いてクローズアップしているように、SubPlayItem情報は、『Clip\_information\_file\_name』と、『In\_time』と、『Out\_time』と、『Sync\_PlayItem\_Id』と、『Sync\_start\_Pts\_of\_PlayItem』とからなる。SubClipの時間軸上に対する、In\_Time,Out\_Timeの指定は、『Clip\_information\_file\_name』、『In\_time』、『Out\_time』によりなされる。『Sync\_PlayItem\_Id』及び『Sync\_start\_Pts\_of\_PlayItem』は、SubClip時間軸上の再生区間と、PL時間軸とを同期させるという同期指定をなす。この同期指定により、PL時間軸と、SubPlayItem時間軸とは同期して進行することになる。

30

#### 【0020】

図9は、SubPlayItem時間軸上の再生区間定義と、同期指定を示す図である。本図において第1段目は、PL時間軸を示し、第2段目はSubPlayItem時間軸を示す。図中のSubPlayItem.IN\_timeは再生区間の始点を、SubPlayItem.Out\_timeは再生区間の終点をそれぞれ示す。これによりSubClip時間軸上にも再生区間が定義されていることがわかる。矢印Sn1においてSync\_PlayItem\_Idは、PlayItemに対する同期指定を示し、矢印Sn2においてsync\_start\_PTS\_of\_PlayItemは、PL時間軸におけるPlayItem上の一時点の指定を示す。

40

#### 【0021】

複数AVClipの切り換えを可能とするマルチアングル区間や、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PL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。

#### 【0022】

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

50

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変わることをいう。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Movie Objectと呼ばれ、管理情報により定義される。一方BD-Jモードを想定した動的シナリオはBD-J Objectと呼ばれる。

#### 【 0 0 2 3 】

まず初めにMovie Objectについて説明する。

< Movie Object >

Movie Objectは、MovieObject.bdmvというファイルに格納される。図 1 0 は、MovieObject.bdmvの内部構成を示す図である。本図の左端に示すようにMovieObject.bdmvは、コード列 "MOBJ" を示す『type\_indicater』と、『version\_number』と、1つ以上のMovieObjectである『MovieObjects()』とからなる。図中の図中の引き出し線vh1はMovieObjectsの内部構成をクローズアップしている。MovieObjects()は、自身のデータ長である『length』と、自身に含まれるMovieObjectの個数である『number\_of\_mobjs』と、number\_of\_mobjs個のMovieObjectである『MovieObjects』とからなる。これらnumber\_of\_mobjs個のMovieObjectは、識別子mobj\_idをもって識別される。図中の引き出し線vh2は、識別子mobj\_idにより特定される任意のMovieObject[mobj\_id]()の内部構成をクローズアップしている。

10

20

#### 【 0 0 2 4 】

この引き出し線に示すように、MovieObjectは、MenuCallがなされた際、MenuCall後の再生再開を意図しているか否かを示す『resume\_intention\_flag』、MenuCallをマスクするか否かを示す情報『menu\_call\_mask』、タイトルサーチ機能をマスクするかを示す『title\_search\_flag』、ナビゲーションコマンドの個数である『number\_of\_navigation\_command』、number\_of\_navigation\_command個の『ナビゲーションコマンド』からなる。

#### 【 0 0 2 5 】

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。

30

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PLを指定することができる。第2引数は、そのPLに含まれるPlayItemや、そのPL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 2 6 】

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PlayPLat-PlayItem( )、

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Play-PLatChapter( )

40

、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PlayPLat-Specified Time()という。

#### 【 0 0 2 7 】

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。

50

## 【 0 0 2 8 】

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。Movie Object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 2 9 】

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 >

10

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る。

## 【 0 0 3 0 】

図 1 1 は、BD-J Object.bdmvの内部構成を示す図である。本図の左端に示すようにBD-J Object.bdmvは、コード列 "BOBJ" を示す『type\_indicater』と、『version\_number』と、1つ以上のBD-J Objectである『BD-J Objects()』とからなる。図中の図中の引き出し線bh1はBD-J Objectsの内部構成をクローズアップしている。BD-J Objects()は、自身のデータ長である『length』と、自身に含まれるBD-J Objectの個数である

『number\_of\_bobj\_s』と、number\_of\_bobj\_s個のBD-J Objectである『BD-J Objects』とからなる。これらnumber\_of\_bobj\_s個のBD-J Objectは、識別子bobj\_idをもって識別される。図中の引き出し線bh2は、識別子bobj\_idにより特定される任意のBD-JObject[bobj\_id]()の内部構成をクローズアップしている。

20

## 【 0 0 3 1 】

この引き出し線に示すように、本図に示すようにBD-J Objectは、『re-sume\_intention\_flag[bobj\_id]』と、『menu\_call\_mask[bobj\_id]』と、『title\_search\_flag[bobj\_id]』と、『Application\_Management\_Table[bobj\_id]』と、『Playlist\_Management\_Table[bobj\_id]』とからなる。『resume\_intention\_flag』、『menu\_call\_mask』、『title\_search\_flag』を含んでいる点においてBD-J ObjectはMovie Objectとほぼ同じである。

## 【 0 0 3 2 】

Movie Object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-J Objectでは、Java（登録商標）アプリケーションに対する指定を『Application\_Management\_Table[bobj\_id]』上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。

30

## 【 0 0 3 3 】

またMovieObjectにおけるPL再生は、PL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ていたが、BD-J ObjectにおけるPL再生は、PL再生手順を示す『Playlist\_Management\_Table[bobj\_id]』をBD-J Objectに組み込むことで記述が可能になる。それだけではなくアプリケーション管理テーブルから参照されるアプリケーションに、PL再生手順を組み込むことでも記述が可能になる。つまりプレイリスト管理テーブルの記述、アプリケーションにおけるPL再生手順の記述のどちらであっても、プレイリスト再生の手順を組み込んでおくことができる。

40

## 【 0 0 3 4 】

ここでJava（登録商標）アプリケーションについて説明する。Java（登録商標）アプリケーションは、仮想マシンのヒープ領域(ワークメモリとも呼ばれる)にロードされた1つ以上のxletプログラムからなる。このワークメモリにロードされたxletプログラム、及び、データから、アプリケーションは構成されることになる。以上がJava（登録商標）アプリケーションの構成である。

## 【 0 0 3 5 】

50

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、図12を参照しながら説明する。

#### < 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る。図12(a)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）ディレクトリ、image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）ディレクトリにクラスファイル(aaa.class,bbb.class)が、imageディレクトリに、menu.jpg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。

10

20

#### 【0036】

本図におけるクラスファイル(図中のaaa.class,bbb.class)は、上述したxletプログラムに対応するクラスファイルである。BD-Jモードにおける再生手順は、このクラスファイルのインスタンスにあたるxletプログラムにより規定される。

xletプログラムとは、JMF(Java（登録商標）Media FrameWork)方式のインターフェイスを利用することができるJava（登録商標）プログラムであり、JMF等の方式に従って、キーイベントに基づく処理を行う。xletプログラムは、JMF方式の処理が可能であるので、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ことができる。他にもxletプログラムでは、ファンクションAPIのコールを記述することにより、BD-ROM再生装置特有の処理を実行させることができる。

30

#### 【0037】

更にxletプログラムは、WWWサイトをアクセスしてコンテンツをダウンロードするという手順を実行することもできる。これによりダウンロードコンテンツと、プレイリスト再生とを交えた斬新な作品を再生させることができる。

xletプログラムのクラスファイルについて説明する。図12(b)は、クラスファイルの内部構成を示す図である。本図に示すようにクラスファイルは、通常のクラスファイル同様、『コンスタントプール』、『インターフェイス』、『メソッド1,2,3・・・n』からなる。クラスファイルにおけるメソッドには、拳動のトリガになるキーイベントが予め登録されているメソッド(EventListner)と、JMFの再生手順を命じるメソッド(JMFプレーヤインスタンスのメソッド)、BD-ROM再生装置側のファンクションAPIをコールするメソッドがある。これらのメソッドは、自身に割り当てられたローカル変数や、自身をコールする際の引数を用いることにより、演算等の手順が記述されている。以上がJava（登録商標）アーカイブファイルについての説明である。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い。

40

#### 【0038】

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。

#### < BD-ROMにおける状態遷移 >

50

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、トップメニューを中核とした構造になっている。そのトップメニューから、個々の著作物へと分岐して再生を行い、その後再び、TopMenu Titleに戻るという独特の状態遷移をなす。図13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る。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。BD-ROMのような映画作品頒布用の記録媒体の業界では、動的商標を、ローディング時に再生するという慣習が定着している。この動的商標は、映画作品の制作者や頒布者を表徴するものであり、FirstPlayTitleは、BD-ROMのローディングされた際、なによりも先に、この動的商標を再生させるという役割分担を担う。

10

#### 【0039】

そしてユーザによるメニューに対する選択操作があれば、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。

かかる状態遷移をなすTitleは、HDMVモードの動的シナリオ、BD-Jモードの動的シナリオから構成される。図14は、HDMVモードの動的シナリオから構成される2つのTitleを示す図である。本図の第1段目は、識別子title\_idにより識別される任意のTitle(title\_id)を示す。第2段目は、そのTitleを構成している1つ以上のMovieObjectからなるMovieObject列を示す。第3段目は、MovieObjectを構成するナビゲーションコマンドを示す。

20

#### 【0040】

他のTitleへのJumpを再生装置に命じるナビゲーションコマンド(JumpTitleコマンド)をMovieObjectに記述しておくことにより、図13に示したような、あるTitleから別のTitleへの分岐が実現されることになる。またPL再生を命じるナビゲーションコマンド(PlayPLコマンド)をMovieObjectに記述しておくことにより、本図第4段目に示すPLが、Titleに帰属することになる。

#### 【0041】

PLを帰属させることにより、HDMVモードにおけるTitleでは、動画再生を伴う映画作品を定義することができる。これがHDMVモードの動的シナリオにより定義されるTitleの構成である。

30

続いてBD-Jモードの動的シナリオにより構成されるTitleの内部構成について説明する。図15は、BD-Jモードの動的シナリオ(BD-J Object)により構成されるTitleの内部構成を示す図である。

#### 【0042】

第1段目は、識別子title\_idにより識別される任意のTitleを示し、第2段目は、そのTitleを構成する唯一のBD-J Objectを示す。第3段目は、そのBD-J Objectの内部にある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示す。第4段目は、第3段目のアプリケーション管理テーブルにより動作することになるアプリケーションを示す。このアプリケーションは、第5段目に示すように、他のTitleへのJumpを再生装置に命じるメソッド(JumpTitleAPIをコールするメソッド)を含んでいるので、JumpTitleAPIのコールメソッドにより、図13に示した他のTitleへの分岐が実現することになる。一方、第3段目にはプレイリスト管理テーブルが記述されているので、第4段目にあたっては、アプリケーションの実行と共にPLが再生されることになる。

40

#### 【0043】

BD-J Objectには、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が含まれているので、第4段目に示すように、アプリケーション実行と、PL再生とが同時になされることになる。このようなアプリケーション実行と、PL再生との同時実行がBD-Jモード

50

におけるTitleの特徴である。

プレイリスト管理テーブルは、全てのBD-J Objectに存在する訳ではない。プレイリスト管理テーブルは任意的な構成要素であり、プレイリスト管理テーブルをもつBD-J Objectもあるし、プレイリスト管理テーブルをもたないBD-J Objectもある。図 1 6 は、プレイリスト管理テーブルをもたないTitleを示す図である。アプリケーション管理テーブルが存在しないため、プレイリスト管理テーブルをもたないようなBD-J Objectでは、第4段目に示すように、アプリケーション動作のみが規定される。以上によるアプリケーション動作の規定により、PL再生を伴わない、制御手順のみのTitleが規定されることになる。

【 0 0 4 4 】

また図 1 4 ではHDMVモードのTitleから、HDMVモードのTitleへの分岐が記述されていたが、図 1 7 に示すようにHDMVモードのTitleから、BD-JモードのTitleへの分岐もありえる。同様に図 1 5 ではBD-JモードのTitleから、BD-JモードのTitleへの分岐が記述されていたが、図 1 8 に示すようにBD-JモードのTitleから、HDMVモードのTitleへの分岐もありえる。

【 0 0 4 5 】

上述したTitleの内部構成において、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のが、図 2 に示したindex.bdmvである。以降index.bdmvについて説明する。

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 4 6 】

図 1 9 は、index.bdmvの内部構成を示す図である。本図に示すようにindex.bdmvは、値 " INDX " を示す値をもつ『type\_indicator』と、『version\_number』と、本ファイルの先頭からIndexesまでの相対アドレスを示す『Indexes\_start\_address』と、『Indexes()』とからなる。『Indexes』は、各Titleに対するインデックスであり、破線の引き出し線ix1でクローズアップしているように、『Indexes』は、『length』、『First-Playback()FirstPlayback\_mobj\_id\_ref』、『TopMenu()TopMenu\_mobj\_id\_ref』、『number\_of\_Titles』、『Title[0]() ~ Title[number\_of\_Titles-1]()』からなる。

【 0 0 4 7 】

『FirstPlayback()FirstPlayback\_mobj\_id\_ref』は、FirstPlayTitleに対するIndexであり、FirstPlayTitleを構成するMovieObject識別子の参照値(FirstPlayback\_mobj\_id\_ref)が格納される。

『TopMenu()TopMenu\_mobj\_id\_ref』は、Top-MenuTitleに対するIndexであり、Top-menu Titleを構成するMovieObject識別子の参照値(TopMenu\_mobj\_id\_ref)が格納される。

【 0 0 4 8 】

『Title[0]() ~ Title[number\_of\_Titles-1]()』は、FirstPlayTitle、Top-MenuTitle以外のTitleに対するIndexであり、number\_of\_Title個存在している。これらは識別子title\_idをもって識別される。

ここで識別子title\_idにより特定されるインデックスをTitle[title\_id]()とする。図中の引き出し線ix2は、Title[title\_id]()に対する内部構成をクローズアップしている。

【 0 0 4 9 】

本図に示すように " Title[title\_id]() " は、 " Title[title\_id]が、分岐をもっているか否か等、Titleの再生の類型を示す『Title\_Playback\_Type[title\_id]』と、このTitleに対するサーチ機能の実行が許可されているか否かを示す『Title\_access\_Flag[title\_id]』と、Titleを構成するMovieObjectを一意に示す『title\_mobj\_id\_ref[title\_id]』とからなる。ここでTitleを構成する動的シナリオがBD-J Objectであれば、『title\_bobj\_id\_ref[title\_id]』が

『title\_mobj\_id\_ref[title\_id]』の代わりになる。『title\_bobj\_id\_ref[title\_id]』は、Titleを構成するBD-J Objectを一意に示す。

【 0 0 5 0 】

10

20

30

40

50

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

<アプリケーション管理テーブル>

BD-J Objectに含まれる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は、本実施形態の主眼となる構成要素であり、その詳細について説明する。先ずアプリケーション管理テーブル(AMT)について説明する。

【0051】

図20(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。 10

図20(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。

【0052】

『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ことを示す。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。 20

【0053】

『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、Susped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。

【0054】

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、ワークメモリに読み出され、仮想マシンによる実行が可能になっている状態をいう。 30

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えるかという“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。

【0055】

それでは、図13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の生存区間はどのように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13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21(a)は、ディスク全体の時間軸を示す図であり、図21(b)は、この時間軸における構成を示す。図21(b)に示すように、ディスク全体の時間軸は、FirstPlay Title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こと、Titleは、1つ以上のMovieObject、又は、唯一のBD-J Object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。 40

【0056】

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MovieObject又はBD-J Objectのうち、どれかがカレントMovieObject、又は、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22(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-J Object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。

#### 【0057】

同様に、識別子mobj\_idにより特定されるMovieObjectが、1つのTitleを構成しているなら、その識別子mobj\_idにより特定されるMovieObject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。

ここでMovieObject、BD-J ObjectがActivateされている期間の終期は、Title分岐(Jump Title)がなされるまでである。つまり、Title分岐(JumpTitle)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MovieObject及びカレントBD-JObjectとして扱われるから、そのMovieObject又はBD-J ObjectにおいてJumpTitleが発生するまでの1つの区間を、Title再生区間として扱う。

#### 【0058】

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PL再生手順を記述することができる。PL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22(a)の一例においてBD-J Object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22(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 - PL - 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。

#### 【0059】

図23は、図22(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。

#### 【0060】

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。

#### 【0061】

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

10

20

30

40

50

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24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。

【0062】

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24の一例においてapplication#3は、title#1、title#2の双方で起動される。

10

【0063】

図24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25(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ため、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ため、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ため、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に帰属しているため、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25(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。

20

【0064】

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。

30

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。

【0065】

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。

40

「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。

50

## 【 0 0 6 6 】

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。

10

## 【 0 0 6 7 】

「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 6 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。

20

## 【 0 0 6 8 】

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 6 9 】

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。

30

## 【 0 0 7 0 】

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。

40

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。

## 【 0 0 7 1 】

また起動優先度は、再生中PL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

50

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PLをポーズを行う。

#### 【0072】

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證することができる。

10

#### <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進行す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Fail Safe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。

20

#### 【0073】

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。

30

図27(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。

#### 【0074】

図27(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、PL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PLがどれであることを示す。PL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PLがどれであることを示す。

40

#### 【0075】

『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PL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。PL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

50

に起動されることになる。

【0076】

『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PL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、分岐元Title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、分岐先Title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。この場合分岐先Titleでは、再生属性がPresentに設定されているので、分岐先Title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間の分岐があった場合でも、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。

10

【0077】

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。

20

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」のいずれかが否かを判定する。「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。

30

【0078】

図28は、分岐先Title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尚且つAutoPlay(ii)、プレイリスト管理テーブル有り尚且つPresent(iii))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。

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、“分岐先Title = プレイリスト管理テーブル有り、尚且つ、分岐先Title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PLの再生は、自動的に開始することになる。

40

【0079】

また“直前状態 = 再生中状態”と、“分岐先Title = 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PLの再生は、自動的に停止することになる。

そしてこれら2つの組合せ以外は全て、分岐元Title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。

50

## 【0080】

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29(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29(b)は、図29(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。

10

## 【0081】

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。

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。

20

## 【0082】

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。

## 【0083】

また図29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。

30

## 【0084】

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30(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30(b)は、図30(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a

40

50

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。

【 0 0 8 5 】

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。

【 0 0 8 6 】

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。

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。

【 0 0 8 7 】

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。

【 0 0 8 8 】

次にプレイリスト管理テーブルによりタイトル再生区間がどのように規定されるかを、図 3 1 を参照しながら説明する。

図 3 1 ( a ) は、再生属性がAutoPlayに設定されたタイトルのタイトル再生区間を示す図である。再生属性がAutoPlayを示すよう設定されれば、タイトルの再生開始と同時に、AutoPlayPLの再生を開始する。ここでアプリケーションが正常に動作し、正常終了したとしても、このタイトル再生区間は、PL時間軸を基準にして定められる。

【 0 0 8 9 】

図 3 1 ( b ) は、プレイリスト管理テーブルにおいて再生属性が " AutoPlay " を示すよう設定され、アプリケーションが異常終了した場合を示す。かかる異常終了により、どのアプリケーションも動作してない状態になるが、AutoPlayPLの再生は継続する。この場合も、AutoPlayPLのPL時間軸がタイトル再生区間になる。

図 3 1 ( c ) は、プレイリスト管理テーブルにおいて再生属性が " AutoPlay " を示すよう設定され、メインアプリの起動に失敗したケースを示す。この場合も、AutoPlayPL再生は、アプリケーションの起動失敗とは関係なしに行われるので、AutoPlayPLの時間軸がタイトル再生区間になる。

【 0 0 9 0 】

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。

## 【0091】

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。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。

## 【0092】

図32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、P-Graphicsデコーダ6、Presentation Graphicsプレーン7、合成部8、フォントゼネレータ9、I-Graphicsデコーダ10、スイッチ11、Interactive Graphicsプレーン12、合成部13、CLUT部14、CLUT部15、オーディオデコーダ16、Network Device17、Local Storage18、リードバッファ19、デマルチプレクサ20、命令ROM21、ユーザイベント処理部22、PSRセット23、CPU24、シナリオメモリ25、ローカルメモリ26、スイッチ27から構成される。

## 【0093】

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16)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。

## 【0094】

デマルチプレクサ(De-MUX)3は、リードバッファ2からTSパケットを取り出して、このTS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、CPU24から設定されたPIDをもつものをビデオデコーダ4、P-Graphicsデコーダ6、I-Graphicsデコーダ10、オーディオデコーダ16のどれかに出力する。

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。

## 【0095】

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CPU24からの指示に従い実行するので、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。

## 【0096】

P-Graphicsデコーダ6は、BD-ROMから読み出されたプレゼンテーショングラフィクスストリームをデコードして、非圧縮グラフィクスをPresentation Graphicsプレーン7に書き込む。グラフィクスストリームのデコードにより、字幕が画面上に現れることになる。

Presentation Graphicsプレーン7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。本プレーンにおける解像度は1920×10

10

20

30

40

50

80であり、Presentation Graphicsプレーン7中の非圧縮グラフィクスの各画素は8ビットのインデックスカラーで表現される。CLUT(Color Lookup Table)を用いてかかるインデックスカラーを変換することにより、Presentation Graphicsプレーン7に格納された非圧縮グラフィクスは、表示に供される。

【0097】

合成部8は、ビデオプレーン5に格納された非圧縮状態のピクチャデータ(i)を、Presentation Graphicsプレーン7の格納内容と合成する。

フォントゼネレータ9は、文字フォントを用いてtextSTストリームに含まれるテキストコードをビットマップに展開してPresentation Graphicsプレーン7に書き込む。

I-Graphicsデコーダ10は、HDMVモードにおいてBD-ROM又はLocal Storage18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12に書き込む。

【0098】

スイッチ11は、フォントゼネレータ9が生成したフォント列、P-Graphicsデコーダ6のデコードにより得られたグラフィクスの何れかを選択的にPresentation Graphicsプレーン7に書き込むスイッチである。

Interactive Graphicsプレーン12は、I-Graphicsデコーダ10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Interactive Graphicsプレーン12には、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。

【0099】

合成部13は、Interactive Graphicsプレーン12の格納内容と、合成部8の出力である合成画像(非圧縮状態のピクチャデータと、Presentation Graphicsプレーン7の格納内容とを合成したもの)とを合成する。かかる合成により、アプリケーションがI-Graphicsデコーダ10に書き込んだ文字・グラフィクスを、非圧縮状態のピクチャデータ上にオーバーレイして、表示することができる。

【0100】

CLUT部14は、ビデオプレーン5に格納された非圧縮グラフィクスにおけるインデックスカラーを、Y,Cr,Cb値に変換する。

CLUT部15は、Interactive Graphicsプレーン12に格納された非圧縮グラフィクスにおけるインデックスカラーを、Y,Cr,Cb値に変換する。

オーディオデコーダ16は、デマルチプレクサ3から出力されたPESパッケージを復号して、非圧縮形式のオーディオデータを出力する。

【0101】

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(Network Device17~De-mux20)について説明する。

Network Device17は、再生装置における通信機能を実現するものであり、BD-JモードにおいてURL指定がJava(登録商標)アプリケーションから与えられれば、そのURLにあたるwebサイトとのTCPコネクション、FTPコネクション等確立する。かかるコネクション確立によりwebサイトからのダウンロードをJava(登録商標)アプリケーションに行わせる。

【0102】

Local Storage18は、Network Device17により確立されたコネクションを通じて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Local Storage18にバインドして管理するための情報であり、このLocal Storage18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。

【0103】

10

20

30

40

50

リードバッファ 19 は、FIFOメモリであり、Local Storage 18 に格納されたダウンロードコンテンツに、SubClipが含まれている場合、このSubClipを構成するTSパケットを、先入れ先出し式に格納する。

デマルチプレクサ(De-MUX) 20 は、リードバッファ 19 からTSパケットを取り出して、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、所望のPIDをもつものをフロントゼネレータ9、I-Graphicsデコーダ10、オーディオデコーダ16に出力する。

#### 【0104】

以上のNetwork Device 17 ~ De-mux 20により、Java(登録商標)アプリケーションがネットワークを通じてダウンロードしたコンテンツを、BD-ROMに記録されたコンテンツ同様再生させることができる。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ROM 21 ~ スイッチ 27)について説明する。

命令ROM 21 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。

#### 【0105】

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CPU 24に出力する。

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示するのに用いられる。

#### 【0106】

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。

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。

#### 【0107】

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PL(カレントPL)の番号を示す。

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(Presentation Time))を示す。以上のPSR4 ~ PSR8により、図 21(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かを特定することができる。

#### 【0108】

CPU 24 は、命令ROM 21 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PL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。

#### 【0109】

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。

スイッチ 27 は、BD-ROM及びLocal Storage 18 から読み出された各種データを、リードバッファ 2、リードバッファ 19、シナリオメモリ 25、ローカルメモリ 26 のどれかに選択的に投入するスイッチである。

#### 【0110】

10

20

30

40

50

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。

図33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。

【0111】

これらの階層のうち図32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32に示したハードウェア構成のうちビデオデコーダ4、P-Graphicsデコーダ6、I-Graphicsデコーダ10、オーディオデコーダ16からなる"デコーダ"と、ビデオプレーン5、Presentation Graphicsプレーン7、Interactive Graphicsプレーン12からなる"プレーン"、BD-ROM及びそのファイルシステム、Local Storage18及びそのファイルシステムを含む。

10

【0112】

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 Engine32の層
- b1)Virtual File System30及びPresentation Engine31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。

20

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)HDMVモジュール33、BD-Jモジュール35が存在する層

からなる。図33のレイアモデルにおいてこのモジュールマネージャ34が最上位の階層に位置するが、モジュールマネージャ34は、HDMVモジュール33及びBD-Jモジュール35をバイパスしてPlayback Control Engine32に至る迂回路ur1を有している。この迂回路によりモジュールマネージャ34は、図33のレイアモデルにおいて"L"字形を逆さまにした形状をなし、User Event Manager37を内蔵した構成になっている。

30

【0113】

BD-Jモジュール35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38を中核にした構成になっている。Java(登録商標)仮想マシン38内のワークメモリは、様々なシステムプログラムやアプリケーションが動作している。Java(登録商標)仮想マシン38の上位に描いた、アプリケーションマネージャ36、Event Listner Manager39(Default Operation Manager40)は、そうしたシステムプログラムの1つである。このうちアプリケーションマネージャ36の中にはPLMTPrcessor41が存在している。またBD-Jモジュール35と、Playback Control Engine32との間には、Permission Controller42が介在している。

【0114】

先ず初めに、第2層に属するVirtual File System30~モジュールマネージャ34について説明する。図34は、Virtual File System30~モジュールマネージャ34による処理を模式化した図である。

40

Virtual File System30は、Local Storage18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Local Storage18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Local Storage18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 System30上のプレイリスト情報は、BD-ROMやLocal Storage18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Local Storage18上のファ

50

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 3 0)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及びSubPlayItem情報の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 3 0、BD-ROM上のAVClipを指定することができる。Virtual File System 3 0を介してLocal Storage 1 8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。Local Storage 1 8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、Local Storage 1 8 + 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。

10

#### 【0115】

Presentation Engine 3 1は、AV再生ファンクションを実行する。再生装置のAV再生ファンクションとは、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(Pause On)、一時停止の解除(PauseOff)、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(Backward Play(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(Angle Change)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6、I-Graphicsデコーダ10、オーディオデコーダ16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。

20

#### 【0116】

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。PLに対する再生制御ファンクションとは、Presentation Engine 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モジュール3 5からのファンクションコールに応じて実行する。

30

#### 【0117】

つまり、PL再生を命じるファンクションコールがあれば、Playback Control Engine 3 2は、再生が命じられたプレイリスト情報を、Virtual File System 3 0経由でBD-ROM又はLocal Storage 1 8から読み出す。そのように読み出されたプレイリスト情報内のPlayItem情報を参照し、そのPlayItem情報においてClip\_Information\_file\_nameに記載されているClip情報を、Virtual File System 3 0経由でBD-ROM又はLocal Storage 1 8から読み出す。図3 4における1,2,3,4は、Virtual File System 3 0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 3 0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 3 0を通じてPresentation Engine 3 1に引き渡す。このようにしてPresentation Engine 3 1にTSパケットが順次渡れば、Presentation Engine 3 1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 3 0からPresentation Engine 3 1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。

40

#### 【0118】

HDMVモジュール3 3は、HDMVモードの実行主体であり、mobj\_idにより分岐先MovieObjectを指定したactivate要求(activate(mobj\_id))がモジュールマネージャ3 4からなされれば、MovieObject(mobj\_id)をローカルメモリ2 6に読み出して、このMovieObjectに記述されたナビゲーションコマンドを解読し、解読結果に基づきPlaybackControl Engine 3

50

2 に対するファンクションコールを実行する。図 2 8 において 2, 3, 4 が付された矢印は、モジュールマネージャ 3 4 による activate(mobj\_id)(2)、MovieObject に記述されたナビゲーションコマンドの解読(3)、Playback Control Engine 3 2 に対するファンクションコール(4)を、模式的に示している。

【 0 1 1 9 】

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でなされる。title\_id を指定した JumpTitle コマンド (JumpTitle(title\_id)) を MovieObject が実行した場合、カレントタイトルを構成する MovieObject に Terminate イベントを発行し、title\_id に対応するタイトルを構成する Movie Object を activate するよう、activate(mobj\_id) イベントが発行する。図中の 0, 1, 2 が付された矢印は、JumpTitle コマンドの実行( 0)、モジュールマネージャ 3 4 による Index.bdmv 参照( 1)、分岐先 Title を構成する Movie Object の activate( 2) の通知を模式的に示している。JumpTitleAPI (JumpTitle(title\_id)) を BD-J Object がコールした場合も同様であり、カレントタイトルを構成する BD-J Object に Terminate イベントを発行し、その title\_id に対応するタイトルを構成する BD-J Object を activate するよう、activate(bobj\_id) を BD-J モジュール 3 5 に発行する。

【 0 1 2 0 】

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。続いてアプリケーションマネージャ 3 6 について、図 3 5 を参照しながら説明する。図 3 5 は、アプリケーションマネージャ 3 6 を示す図である。

アプリケーションマネージャ 3 6 は、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ーションを起動するよう Java (登録商標) 仮想マシン 3 8 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-J Object 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。ここでタイトル分岐があった場合、モジュールマネージャ 3 4 から activate(bobj\_id) が通知される。かかる通知があれば、bobj\_id に対応する BD-J Object をカレント BD-J Object にして、そのカレント BD-J Object におけるアプリケーション管理テーブルを参照することで、上述した自動起動すべきアプリケーション及び自動終了すべきアプリケーションを特定する。図 3 5 における

0, 1, 2, 3 は、TitleJump の発生( 0)、activate(bobj\_id) の通知( 1)、アプリケーション管理テーブル参照( 2)、Java (登録商標) 仮想マシン 3 8 に対するアプリケーション起動指示( 3) 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 3 8 は、ローカルメモリ 2 6 からワークメモリに xlet プログラムを読み出す( 4,5)。

【 0 1 2 1 】

以上がアプリケーションマネージャ 3 6 についての説明である。続いて User Event Manager 3 7 ~ Default Operation Manager 4 0 について、図 3 6 を参照しながら説明する。

User Event Manager 3 7 は、ユーザイベント処理部 2 2 が受け付けたユーザイベントを、再生制御のためのユーザイベントと、キーイベントとに分離する。再生制御のためのユーザイベントとは、再生開始(Play)、再生停止(Stop)、一時停止(Pause On)、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(Backward Play(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(Angle Change) のどれかを命じるユーザイベントである。一方キーイベントとは、Pop-up キー, MoveUp キー, MoveDown キー, MoveRight キー, MoveLeft キー, Numeric キーのどれかの押下を示すユーザイベントである。そして再生制御のためのユーザイベントに基づき、再生制御ファンクションを Playback Control Engine 3 2 に実行させるためのファンクションコールを行う。このファンクション

コールは、U0(User Operation)と呼ばれ、HDMVモジュール 3 3、BD-Jモジュール 3 5を介することなく、モジュールマネージャ 3 4における迂回路にあるU0コントローラ 3 7aを用いてなされる。これにより再生開始(Play)、再生停止(Stop)、一時停止(Pause On)、一時停止の解除(Pause Off)というような再生制御は、遅延なくなされることになる。図中の 1,2,3は、User Event Manager 3 7による再生制御ユーザイベントと、キーイベントとの分離( 1,2)、再生制御ユーザイベントに基づくPlayback Control Engine 3 2に対するファンクションコール( 3)を模式的に示している。

#### 【 0 1 2 2 】

Java (登録商標) 仮想マシン 3 8は、アプリケーションを構成するxletプログラムをワークメモリにロードして、xletプログラムを解読し、解読結果に従って、下位層に対する制御を行う。下位層への制御は、JMFメソッドを図示しないBDミドルウェアに発行して、BD再生装置が対応しているファンクションコールに置き換させ、置換後のファンクションコールをPlayback Control Engine 3 2に発行することでなされる。

10

#### 【 0 1 2 3 】

Event Listener Manager 3 9は、キーイベントを解析し、イベントの振り分けを行う。図中の実線矢印 1, 2は、このEvent Listener Manager 3 9による振り分けを模式的に示す。xletプログラム内のEvent Listenerに登録されたキーイベントなら、BD-J Objectにより間接参照されているxletプログラムにかかるイベントを振り分ける。xletプログラムのEvent Listenerには、JMFに対応したキーイベントが登録されているので、本キーイベントによりxletプログラムの起動が可能になる。キーイベントがEvent Listener非登録のキーイベントである場合、本キーイベントをDefault Operation Manager 4 0に振り分ける。BD-ROM再生装置において生じるキーイベントには、Event Listenerに登録されていない多様なものがあり、これらのキーイベントが生じたとしても、漏れの無い処理を実行するためである。

20

#### 【 0 1 2 4 】

Default Operation Manager 4 0は、xletプログラム内のEvent Listenerに登録されていないキーイベントがEvent Listener Manager 3 9から振り分けられると、そのEvent Listener非登録イベントに対応するファンクションコールをPlayback Control Engine 3 2に対して実行する。このDefault Operation Manager 4 0によるファンクションコールを模式的に示したのが、図中の矢印 3である。

30

#### 【 0 1 2 5 】

PLMTプロセッサ 4 1は、アプリケーションマネージャ 3 6の一構成要素であり、モジュールマネージャ 3 4からactivate(bobj\_id)がなされれば、そのbobj\_idにより特定されるBD-J Objectのプレイリスト管理テーブルを参照する。そして再生属性 = AutoPlayのPLが、そのBD-J Objectのプレイリスト管理テーブル内に存在すれば、そのAutoPlayPLを再生するようPlayback Control Engine 3 2に出力する。一方、PLによる再生終了を示すNotifyイベントをPlayback Control Engine 3 2が発行すれば、その発行された時点とTitle終了時点として解釈する。図中の 1,2は、Playback Control Engine 3 2に対するPlayPLファンクションのコール( 1)、Playback Control Engine 3 2からのNotifyイベント出力( 2)を模式化している。

40

#### 【 0 1 2 6 】

以上でBD-Jモジュール 3 5内のレイヤ構成についての説明を終える。尚、Permission Controller 4 2については、本実施形態における説明を割愛する。Permission Controller 4 2の処理内容は、本実施形態よりも第3実施形態で詳述するのが望ましいからである。

< Java (登録商標) 仮想マシン 3 8の内部構成 >

Java (登録商標) 仮想マシン 3 8の内部構成について説明する。図 3 7は、Java (登録商標) 仮想マシン 3 8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 8は、図 3 2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。

50

## 【0127】

ユーザクラスローダ52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26等から読み出してメソッドエリア53に格納する。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36がユーザクラスローダ52に指示することでなされる。ファイルパスがローカルメモリ26を示しているなら、ユーザクラスローダ52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26からワークメモリ54に読み出す。ファイルパスがVirtual File System30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52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Local Storage18からワークメモリ54に読み出す。図35に示したアプリケーション起動制御(3,4,5)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36に通知することになる。

10

## 【0128】

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33に示したアプリケーションマネージャ36、Event Listener Manager39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。図33、図35、図36のレイアモデルでは、このワークメモリ54をJava(登録商標)仮想マシン38上位層に記述していた。これはワークメモリ54上のアプリケーションマネージャ36及びEvent Listener Manager39を、わかり易く記述するための配慮に過ぎない。アプリケーションマネージャ36及びEvent Listener Manager39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。

20

30

## 【0129】

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38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38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に発行することでなされる。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。

40

## 【0130】

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

50

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。JMFプレーインスタンスの再生メソッドをコールする場合、又は、JumpTitleAPIのコールをコールする場合にも、これらのコールに対応するフレームがJava(登録商標)スタック56a,b,...n上に積み上げられる。そしてこれらのフレームのオペランドスタックには、再生メソッドによる再生の対象となるMPLSファイルのファイル名、又は、JumpTitleAPIのコールのジャンプ先を示すtitle\_id等が引数として格納される。

10

#### 【0131】

以上のように構成されたJava(登録商標)仮想マシン38の内部構成において、アプリケーションマネージャ36及びその構成要素たるPLMT Processor 41がどのような処理を行うかを、詳細に説明する。

bobj\_idのBD-J Objectのactivateを要求するイベント(activatred[bobj\_id])をモジュールマネージャ34が出力すれば、ワークメモリ54上のインスタンスであるアプリケーションマネージャ36は、そのbobj\_idを有するBD-J Objectを新たにカレントBD-J Objectにする。そしてアプリケーションマネージャ36は、分岐元タイトルにおける実行状態と、カレントBD-J Objectにおけるアプリケーションの起動属性とを照合して、自動起動すべきアプリケーション(i)、自動終了すべきアプリケーション(ii)を判定する。

20

#### 【0132】

(i)自動起動すべきアプリケーションの判定は、分岐元タイトルにおいて非実行状態であるが、カレントBD-J ObjectにおいてAutoPlay属性を有するアプリケーションのapli\_id\_refをカレントBD-J Objectのアプリケーション管理テーブルから検索することでなされる。かかるapli\_id\_refがあれば、アプリケーション[apli\_id\_ref]についてのJava(登録商標)アーカイブファイルに属するクラスファイルを、メソッドエリア53に読み出ようユーザクラスロード52に指示し、そのクラスファイルに対するインスタンスをワークメモリ54上に生成させる。こうしてカレントタイトルを生存区間としているアプリケーションが実行可能な状態になる。その後、アプリケーションにおけるメソッドをスレッド55a,b,...nに実行させることで、アプリケーションを実行させる。

30

#### 【0133】

(ii)自動終了すべきアプリケーションの判定は、分岐元タイトルにおいて実行状態であるが、カレントタイトルにおいて生存区間を有しないアプリケーションのapli\_id\_refをカレントBD-J Objectのアプリケーション管理テーブルから検索することでなされる。かかるapli\_id\_refがあれば、かかるアプリケーションを構成するxletプログラムを終了させる。これにより当該アプリケーションがワークメモリ54上で占めていた領域や、そのアプリケーションのメソッド実行のためのJava(登録商標)スタック56a,b,...nにおけるフレーム等のリソースが回収されることになる。

40

#### 【0134】

アプリケーションマネージャ36の構成要素たるPLMT Processor 41は、分岐元タイトルにおける再生状態と、カレントタイトルにおけるプレイリストの再生属性とを照合して、自動再生すべきプレイリスト(i)、自動終了すべきプレイリスト(ii)を判定する。

(i)自動再生すべきプレイリストの判定は、分岐元タイトルにおいて非再生状態であり、カレントタイトルにおいてAutoPlay属性に設定されているプレイリストがプレイリスト管理テーブル内に存在するか否かの検索をプレイリスト管理テーブルに対して行うことでなされる。もしあれば、再生すべきプレイリストのPL\_id\_refを引数としたプレイリスト再生のファンクションコールを実行する。このコールにより、引数たるPL\_id\_refをオペランドスタックに格納したフレームがJava(登録商標)スタック56a,b,...nに生成される。そしてプレイリスト再生のファンクションコールをスレッド55a,b,...nが実行

50

する。このファンクションコール実行によりPresentation Engine 3 1はプレイリスト再生を開始する。

【0135】

(ii)自動終了すべきプレイリストの判定は、分岐元タイトルにおいて再生中状態であり、カレントタイトルのプレイリスト管理テーブルに記述されていないプレイリストが存在するか否かの検索をプレイリスト管理テーブルに対して行うことでなされる。もしあれば、そのプレイリストの再生停止を命じるファンクションコールを実行して、そのプレイリストの再生のためのファンクションコールに対応するフレームをJava(登録商標)スタック56a,b・・・nから削除する。

【0136】

ワークメモリ54上でのアプリケーション終了には、4つの態様がある。図38は、アプリケーション終了の4態様を示す図である。1つ目は、リソースが枯渇したためアプリケーションマネージャ36によるTerminateイベント発行によりアプリケーションが終了するとの態様(1)、2つ目は、他のアプリケーションが発行したTerminateイベントを、アプリケーションマネージャ36を介して受け付けて、終了するとの態様(2)、3つ目は、アプリケーション管理テーブルに記述された生存区間が終了したため、アプリケーションマネージャ36が発行したTerminateイベントにより終了するとの態様(3)、4つ目は、アプリケーションが自身の終了プロセスを起動することにより終了するとの態様である(4)。

これらの4つの態様のうち3つものが、アプリケーションマネージャ36によりなされているため、アプリケーションマネージャ36は、アプリケーション動作制御の中樞を掌握することがわかる。仮に、Terminateイベントの発行でもアプリケーションが終了しない場合、アプリケーションマネージャ36は、そのアプリケーションを強制終了させて、使用しているリソースを明け渡させることができる。強権的なリソース解放を担うのも、アプリケーションマネージャ36の1つの特徴である。

【0137】

以上がBD-Jモジュール35の構成要素である。

(フローチャートの説明)

以上のアプリケーションマネージャ36についての説明は、その概要に触れたに過ぎない。アプリケーションマネージャ36の処理を更に詳しく示したのが図39、図40のフローチャートである。以降、これらのフローチャートを参照してアプリケーションマネージャ36の処理手順についてより詳しく説明する。

【0138】

図39は、アプリケーションマネージャ36による処理手順を示す図である。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切り換えを行い(ステップS7)、分岐先Titleのアプリケーション管理テーブルを参照することにより、分岐元Titleで起動中で、分岐先Titleで非生存のアプリケーションを終了させる(ステップS8)。一方、分岐先Titleのプレイリスト管理テーブルを参照することにより、分岐元Titleで再生中で、分岐先Titleで非生存となるPLの再生を停止させる(ステップS9)。

【0139】

そして、分岐元Titleでは非再生状態、分岐先TitleにおいてAutoPlay属性になっているPLをPLMT Processor 41に判定させる(ステップS10)。もし存在すればPLMT Processor 41はAutoPlayPLを再生するようPlayback Control Engine 32に指示する(ステップS11)。もし存在しないなら、AutoPlayPLは再生しない。

続くステップS12~ステップS18からなる一連の手順は、分岐先Titleを生存区間とするアプリケーションの起動を実現する。この手順は、AutoRunアプリケーションに対して起動指示を行い(ステップS14)、起動に成功すれば(ステップS15でYes)、AutoPlayPLの再生画像をクォータ(1/4)に変換する(ステップS18)という手順を実現するものである。

10

20

30

40

50

## 【 0 1 4 0 】

一方このステップ S 1 5 が No であればステップ S 1 4 ~ ステップ S 1 7 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、ステップ S 1 2 においてリセットされ、ステップ S 1 6 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 1 7 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 1 4、ステップ S 1 5、ステップ S 1 6、ステップ S 1 7 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。

10

## 【 0 1 4 1 】

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い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 1 9、ステップ S 2 0 の処理を実行する。正常終了していればステップ S 1 9 ~ ステップ S 2 0 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。

## 【 0 1 4 2 】

ステップ S 1 9 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 Playback Control Engine 3 2 に指示する(ステップ S 2 0)。その後、ステップ S 1 6 に移行する。ステップ S 1 6 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。これによりステップ S 1 2 において設定された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。

20

## 【 0 1 4 3 】

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)。

図 4 0 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の具体例を示す。図 4 0 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 PL 再生の進行、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。一方、application # 1 は、時点 t1 で動作状態になるので、PL 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画面であり、Interactive Graphics プレーン 1 2 に対する描画処理を Java (登録商標) アプリケーションが実行することによってなされる。Java (登録商標) アプリケーションがかかる描画処理を行うには、グラフィクス描画や文字描画のためのライブラリが必要になる。以降アプリケーション実行と PL 再生とが同時になされる限り、親子画面の表示は継続する。

30

40

## 【 0 1 4 4 】

その後 application # 1 が異常終了してしまい、時点 t2 においてアプリケーションマネージャ 3 6 がこの異常終了を検知したとする。矢印 br1 は、この検知を象徴的に示す。そうするとステップ S 2 0 においてアプリケーションマネージャ 3 6 は、PL の再生画像をフルスクリーンにする。図中の時点 t3 は、このフルスクリーン化のタイミングを示す。図中の第 1 段目の右側に、フルスクリーン化で得られた画像 gj3 を示す。

## 【 0 1 4 5 】

50

以上のように本実施形態によれば、プレイリスト管理テーブルの再生属性を、“Auto-Play”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。

【0146】

また、アプリケーションの起動に失敗したり、アプリケーションが異常したとしても、プレイリスト管理テーブルで規定されたプレイリストの再生を継続することにより、“とりあえず何かが写っている状態”になる。これにより、装置がブラックアウトしてしまうという最悪ケースを回避することができるので、装置を製造するメーカーに、最低限度の安心感を与えることができる。

10

【0147】

（第2実施形態）

第2実施形態は、エラー終了時におけるリカバリー処理を、オーサリング時に規定しておくための改良に関する。かかるリカバリー処理を規定するため、本実施形態に係る記録媒体では、1つのBD-J Object内にエラー管理テーブルを設けている。図41(a)は、第2実施形態に係るBD-J Objectの内部構成を示す図である。本図に示すように、BD-J Objectには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の他に、エラー管理テーブル(Error Management Table[bobj\_id])が追加されている。図41(b)は、エラー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、エラー管理テーブルは、Number\_of\_recovery個のリカバリー情報(recovery())からなる。識別子recovery\_idで特定される、任意のリカバリー情報の内部構成を、引き出し線em1にてクローズアップしている。この引き出し線によるとリカバリー情報は、リカバリー情報に対応するアプリケーションの識別子を、一意に特定する参照値『Apli\_id\_ref』、『Restart\_Application\_flag』、『Continuous\_Playback\_flag』、『Select\_Title\_flag』、『Notify\_Event\_flag』、『Reboot\_flag』という5つのflagからなる。図42は、これら5つのフラグの意味内容を示す。以降エラー管理テーブルにおける各フラグの意味内容について説明する。

20

【0148】

『Restart\_Application\_flag』は、0に設定されることにより異常終了時に再起動しないことを示す。0以外の整数値nに設定されることにより、再起動をn回繰り返すことを示す。デフォルト値として本flagは、0に設定されている。

30

『Continuous\_Playback\_flag』は、0に設定されることにより異常終了時にPL再生を継続しないことを示す。整数値1に設定されることにより、異常終了時にPL再生を継続することを示し、2に設定されることにより、異常終了時に、フルスクリーン・一倍速でPL再生を継続することを示す。デフォルト値として本flagは、0に設定されている。

【0149】

『Select\_Title\_flag』は、0に設定されることにより異常終了時にTitle分岐を行わないことを示す。0以外の整数値nに設定された場合、Title番号nにより特定されるTitleにJumpすることを示す。デフォルト値として本flagは、0に設定されている。

40

『Notify\_Event\_flag』は、0に設定されることにより異常終了時に、イベントを出力しないことを示す。0以外の整数値nに設定された場合、イベント番号nのイベントを出力することを示す。デフォルト値として本flagは、1に設定されている。

【0150】

『Reboot\_flag』は、0に設定されることにより異常終了時に、再生装置のブートストラップを行わないことを示す。整数値1に設定された場合、ブートストラップを行うことを示す。

以上のflagが存在するので、アプリケーションが異常終了した際、どのようなリカバリー処理を実行するかをオーサリング時に予め規定しておくことができる。エラー管理テーブル記述の具体例について説明する。図43(a)は、エラー管理テーブルが記述された2

50

つのTitle(title#1、title#2)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が記述されており、title#1のエラー管理テーブルにはこのapplication#1が異常終了した際に用いられるべきRecovery情報が記述されている。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が記述されている。

【0151】

title#2側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#2が記述されており、title#2側のエラー管理テーブルには、このapplication#2に対するRecovery情報が記述されている。

図43(b)は、図43(a)のように記載されたアプリケーション管理テーブル、エラー管理テーブル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。application#1についてのRecovery情報は、Continuous\_Playback\_Flag=2と記述されているため、application#1の異常終了時にはプレイリスト再生が一倍速、フルスクリーンにより継続することになる。

【0152】

一方application#2についてのRecovery情報は、Notify\_Event\_Flag=2と記述されているので、application#2の異常終了には、番号=2のイベントを出力することになる。

以上のようなRecovery情報の記述により、アプリケーションが異常終了した際の処理を、Title毎、アプリケーション毎に変化させることができる。

BD-J Objectにおいて上述したエラー管理テーブルが追加されたため、本実施形態に係るアプリケーションマネージャ36は図44、図45のフローチャートに従って処理を行う。図44は、第2実施形態に係るアプリケーションマネージャ36の処理手順を示すフローチャートである。本フローは、図39同様、ステップS1～ステップS4をメインループにしている。このメインアプリにおいてタイトルが選択された際の処理は、ステップS21～ステップS27に示されるものとなる。

【0153】

ステップS21では、分岐先Titleにプレイリスト管理テーブルが存在するか否かの判定をPLMT Processor 41に行わせる。もし存在すれば、分岐元Titleにおいて非再生状態だが、分岐先TitleにおいてAutoPlay属性になっているPLの再生をPlaybackControl Engine 32に開始させて(ステップS22)、その再生が成功したかどうかを判定する(ステップS23)。もし成功すればステップS25～ステップS28の処理を行う。成功しなければ図45のフローチャートに移行する。

【0154】

分岐先Titleにプレイリスト管理テーブルがない場合は、分岐元Titleにおいて再生状態であったPLの再生をPlayback Control Engine 32に停止させて(ステップS24)、ステップS25～ステップS28の処理を行う。

ステップS25は、分岐先Title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る。もし存在すれば、分岐先アプリケーションにおけるAutoRunアプリケーションを起動して(ステップS26)、ステップS27において起動に成功したか否かを判定する。成功すればステップS1～ステップS4のループ処理に戻る。成功しなければ図45のフローチャートに移行する。

【0155】

図45のフローチャートは、アプリケーションが異常終了した場合の処理手順を示すフローチャートである。ステップS30は、異常終了したアプリケーションが属するTitleに、エラー管理テーブルが存在するか否かの判定であり、もし存在しなければステップS1～ステップS4からなるループ処理に戻る。

存在すれば、ステップS31～ステップS44の処理を経た上でステップS1～ステップS4からなるループ処理に戻る。ステップS31は、エラー管理テーブルにおけるRestart\_Application\_flagが0でないかの判定である。0でなければ、ステップS40～ステップS44からなるループ処理を実行する。このループ処理は、Restart\_Application\_flag

10

20

30

40

50

の値nを再起動カウンタに設定して(ステップS 4 0)、ステップS 4 1～ステップS 4 4のループ処理を実行するというものである。このループ処理の制御変数は再起動カウンタであり、この再起動カウンタが0になること(ステップS 4 1でYes)、及び、アプリケーションの起動に成功したこと(ステップS 4 4でYes)が本ループ処理の終了条件である。本ループ処理では、これらのステップS 4 1、ステップS 4 4がNoである限り、再起動カウンタをデクリメントして(ステップS 4 2)、AutoRunアプリケーションを起動するという(ステップS 4 3)を繰り返す。こうした繰り返しにより異常終了において、アプリケーションの再起動がなされるのである。Restart\_Application\_flagが0であれば、ステップS 3 2を実行する。

【0156】

ステップS 3 2は、Continuous\_Playback\_flagが0であるか、1であるか、2であるかを判定するステップである。もし2であれば、AutoPlayPLの再生画像をフルスクリーン表示にして(ステップS 3 3)、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。

もし1であれば、AutoPlayPLの再生画像をクォータ表示にしたまま(ステップS 3 4)、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。

【0157】

もし0であれば、ステップS 3 5に移行する。

ステップS 3 5は、エラー管理テーブルのselevt\_title\_flagが0でないか否かの判定であり、もし0であるならステップS 3 7に移行する。0でないなら、selevt\_title\_flagの値nを分岐先Titleにして(ステップS 3 6)、図4 4のステップS 7に移行する。

ステップS 3 7は、エラー管理テーブルのNotify\_Event\_flagが0でないか否かの判定であり、もし0であるならステップS 3 9に移行する。0でないなら、Notify\_Event\_flagの値nから特定されるイベントnを発生して(ステップS 3 8)、図4 4のステップS 1～ステップS 4のメインループに移行する。ステップS 3 9は、エラー管理テーブルのBoot\_flagが0でないか否かの判定であり、もし0であるならステップS 1～ステップS 4からなるメインループに移行する。0でないなら、図4 4の先頭に移行して、再生装置のブートストラップを実行する。

【0158】

以上のように本実施形態によれば、アプリケーションがエラー終了した際、再生装置はどのような処置をとるべきかを、装置側の製造者ではなく、オーサリング担当者が規定しておくことができる。

尚、エラー管理テーブルをもたないタイトルの再生時においてアプリケーションの異常終了が発生した場合に、リカバリー処理を行うようなプログラムを再生装置に組み込んでいてもよい。

【0159】

またRestart\_Application\_Flag～Rboot\_Flagのどれかを指定するような引数をTitle-JumpAPIに設けておき、この引数に応じたリカバリー処理をアプリケーションマネージャ3 6に実行させてもよい。

(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PL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。これに対し本実施形態では、アプリケーションのJMFメソッドによるPL再生を主眼としている。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PL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PL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PL再生の制限を実現するため、第3実施形態では、Permission Controller 4 2、アプリケーションマネージャ3 6が以下の処理を行う。

【0160】

10

20

30

40

50

パーミッションコントローラ 4 2 は、どれかのアプリケーションが PL 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 PL 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 Playback Control 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 Permission Controller 4 2 による許否判定により、ある配給会社の配給にかかる PL 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 PL の無断引用を避けることができる。許可とすべき PL とアプリケーションとの組合せ、不許可とすべき PL 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、Permission Controller 4 2 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。

10

## 【 0 1 6 1 】

第 3 実施形態においてアプリケーションマネージャ 3 6 は、現在の再生時点において再生可能な PL を、アプリケーションからの要求に応じて通知する。図 4 6 は、このアプリケーションマネージャ 3 6 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 PL 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 S 4 5)。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 S 4 6)。PL 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 PL を再生可能な PL として要求元のアプリケーションに通知する (ステップ S 4 7)。

20

## 【 0 1 6 2 】

PL の記述がなければ、PL 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 S 4 8)。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 3 6 の処理手順である。

続いて PL 再生が要求された場合のアプリケーションマネージャ 3 6 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 3 6 は、図 4 7 のフローチャートに従って処理を行う。

## 【 0 1 6 3 】

図 4 7 においてアプリケーションマネージャ 3 6 は、PL 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 S 5 1)。どれかのアプリケーションが PL 再生を要求すれば、要求元アプリケーションに PL 再生の権利が存在するか否かの認証を Permission Controller 4 2 に行わさせる (ステップ S 5 2)。もし再生する権利があれば、Playback Control Engine 3 2 に再生開始を指示して (ステップ S 5 3)、Playback Control Engine 3 2 からのサクセス応答を待つ (ステップ S 5 4)。

30

## 【 0 1 6 4 】

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 Local Storage 1 8 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 Local Storage 1 8 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 Local Storage 1 8 から構成される仮想的なパッケージに矛盾があり、正しいプレイリストを構成することができない場合、Playback Control 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 PL を再生しており、PL 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 Playback Control Engine 3 2 は、false を示す応答を返す。

40

## 【 0 1 6 5 】

以上の過程を経てサクセス応答があれば、PL 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 S 5 5)。

50

サクセス応答がなければ、PL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、PL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)。

【0166】

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。

10

【0167】

(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。

【0168】

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Playback Control Engine 3.2からの要求に応じて実行する。

20

【0169】

PL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、Playback Control Engine 3.2にPL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。

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48(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PLとを対応付けて示している。このうち図48(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48(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PL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。

30

40

【0170】

図48(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48(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PL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

50

る。

【 0 1 7 1 】

図 4 8 ( 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 4 8 ( c )において、PSR(14)がとりうる値は、解像度525×600 TVsystem 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PL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525×600 TVsystem 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

10

×1080 TVsystem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 4 8 ( 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。

【 0 1 7 2 】

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 3 6、Playback Control Engine 3 2にある。

アプリケーションマネージャ 3 6 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Playback Control Engine 3 2 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PLのうち、どれかを自動的に再生するようPlayback Control Engine 3 2 に要求する。

20

【 0 1 7 3 】

Playback Control Engine 3 2 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、PL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PLのリストが通知されれば、そのPLリストに記載されたPL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PL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。

30

【 0 1 7 4 】

図 4 9 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 3 6 からの通知( 1)、Playback Control Engine 3 2 による再生可能PL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Playback Control Engine 3 2 への再生可能なPLの通知( 4)を模式的に描いている。

40

【 0 1 7 5 】

尚、図 4 9 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 3 8 内のワークメモリ 5 4 においてインスタンスとしてスレッド 5 5 により実行されるとというのが、現実的な記述になる。

【 0 1 7 6 】

以上のように本実施形態によれば、Titleのバウンダリー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Playback Control Engine 3 2 は、BD-ROMにおける複数PLのうち、再生装置側の状態設定に応じたものはどれであ

50

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PL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。

【 0 1 7 7 】

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)

第4実施形態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、PSRの設定値に応じて再生すべきPLを選択するという選択アルゴリズムを有していたが、本実施形態に係るタイトルアンバウンダリーアプリケーションは、1つのPL内にマルチアングル区間がある場合、当該マルチアングル区間における複数アングルのどれかを選ぶとの処理を、タイトルアンバウンダリーアプリケーションに行わせる改良に関する。本実施形態に係るタイトルアンバウンダリーアプリケーションは、PSRがとりうる値の複数の範囲と、それら範囲毎に再生すべきアングルとを対応づけたものである。本実施形態において現在の再生時点がマルチアングル区間であるとPlayback Control Engine 3 2はタイトルアンバウンダリーアプリケーションに対し、何れのアングルを再生するかの問い合わせを行う。かかる問い合わせがあれば、タイトルアンバウンダリーアプリケーションは現在におけるPSR設定値を取得し、選択アルゴリズムを実行して、取得した設定値に応じたアングルを選択する。その選択結果をPlayback Control Engine 3 2に通知してそのアングルの再生をPlayback Control Engine 3 2に行わせる。

【 0 1 7 8 】

以上のように本実施形態によれば、PSRがどの値であるとき、どのアングルを選択するというアルゴリズムを、オーサリング担当者が規定しておくことができるので、アングルを応用した様々なアプリケーションをオーサリング担当者は作り出すことができる。

(第6実施形態)

BD-Jモードにおいて、PL再生との同期をどのように実現するかという改良に関する。Playback Control Engine 3 2は、PlayPLAPIファンクションがコールされれば、PL情報に基づく処理手順を実行する。PLが2時間という再生時間を有するならば、この2時間の間、上述した処理は継続することになる。ここで問題になるのは、Java(登録商標)仮想マシン3 8がサクセス応答を返す時間と、Playback Control Engine 3 2が実際に処理を終える時間とのギャップである。Java(登録商標)仮想マシン3 8は、イベントドリブンの処理主体であるためコール直後に再生成功か、再生失敗かを示す応答を返すが、Playback Control Engine 3 2による実際の処理終了は2時間経過後であるので、サクセス応答をアプリケーションに返す時間を基準にしたのでは、2時間経過後にあたる処理終結を感知しえない。PL再生において早送り、巻戻し、Skipが行われると、この2時間という再生期間は2時間前後に変動することになり、処理終結の感知は更に困難になる。

【 0 1 7 9 】

Playback Control Engine 3 2は、アプリケーションとスタンドアローンで動作するため、アプリケーションマネージャ3 6は、PL再生の終了時点を正確に解釈することができない。そこで本実施形態では、アプリケーションが終了しようがいまいが、ワークメモリにJMFプレーヤインスタンスがある限り、つまり、Presentation Engine 3 1の制御権をBD-Jモジュール3 5が掌握している間、Playback Control Engine 3 2からNotifyイベントを待つ。そしてNotifyイベントがあれば、タイトルが終了したと解釈して、次のタイトルへの分岐を行うようモジュールマネージャ3 4に通知する。こうすることにより、Playback Control Engine 3 2がPL再生を終結した時点を、タイトルの終端とすることができる。

【 0 1 8 0 】

以降図5 0 ~ 図5 4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。

図50は、Playback Control Engine 32によるPL再生手順を示すフローチャートである。この再生手順は、Presentation Engine 31に対する制御(ステップS106)と、BD-ROMドライブ1又はLocal Storage 18に対する制御(ステップS108)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(.mpis)の読み込みを行い(ステップS101)、その後、ステップS102～ステップS110の処理を実行するというものである。ここでステップS102～ステップS110は、ステップS109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S103～ステップS110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PL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S102)。上述したループ処理の終了要件は、このPlayItem#xがカレントPLの最後のPlayItemになることであり(ステップS109)、もし最後のPlayItemでなければ、カレントPLにおける次のPlayItemがPlayItem#xに設定される(ステップS110)。

10

20

30

40

50

#### 【0181】

ループ処理において繰り返し実行されるステップS103～ステップS110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S103)、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S104)、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S105)、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S107)、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Local Storage 18に命じるというものである(ステップS108)。

#### 【0182】

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S106)。以上のステップS105～ステップS108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PLの最後のPIであるかの判定がなされる(ステップS109)。

#### 【0183】

PlayItem#xがカレントPLの最後のPIでなければ、カレントPLにおける次のPlayItemを、PlayItem#xに設定して(ステップS110)、ステップS103に戻る。以上のステップS103～ステップS110を繰り返すことにより、PLを構成するPIは順次再生されることになる。

図51は、アングル切り換え手順及びSkipBack, SkipNextの手順を示すフローチャートである。本フローチャートは、図50の処理手順と並行してなされるものであり、ステップS111～S112からなるループ処理を繰り返すというものである。本ループにおけるステップS111は、アングル切り換えを要求するAPIが、Java(登録商標)仮想マシン38からコールされたか否かの判定であり、アングル切り換えAPIのコールがあれば、カレントClip情報を切り換えるという操作を実行する。

#### 【0184】

図51のステップS115は、判定ステップであり、PlayItem#xのis\_multi\_anglesがオンであるか否かの判定を行う。is\_multi\_anglesとは、PlayItem#xがマルチアングルに対応しているか否かを示すフラグであり、もしステップS115がNoであるならステップS113に移行する。ステップS115がYesであるなら、ステップS116～ステップS119を実行する。ステップS116～ステップS119は、切り換え後のアングル番号を変数yに代入して(ステップS116)、PlayItem#xにおけるy番目のClip\_informatio

n\_file\_nameで指定されているClip情報をシナリオメモリ21に読み出し(ステップS117)、カレントPTMを、カレントClip情報のEP\_mapを用いてIピクチャアドレスuに変換し(ステップS118)、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に変換する(ステップS119)というものである。こうしてIピクチャアドレスu,vを変化した後、同時実行されている図50側の処理を停止させた上でステップS106に移行する。ステップS106への移行により、別のAVClipからTSパケットが読み出されるので、映像内容が切り換わることになる。

【0185】

一方、図51のループにおけるステップS112は、SkipBack/SkipNextを意味するAPIがJava(登録商標)仮想マシン38からコールされたか否かの判定であり、もしコールされれば、図52のフローチャートの処理手順を実行する。図52は、SkipBack,SkipNextAPIがコールされた際の処理手順を示すフローチャートである。SkipBack,SkipNextを実行するにあたっての処理手順は多種多様なものである。ここで説明するのはあくまでも一例に過ぎないことに留意されたい。

10

【0186】

ステップS121は、PSRで示されるカレントPI番号、及び、カレントPTMを変換することにより、カレントMark情報を得る。ステップS122は、押下されたのがSkipNextキーであるか、SkipBackキーであるかの判定であり、SkipNextキーであるならステップS123において方向フラグを+1に設定し、SkipBackキーであるならステップS124において方向フラグを-1に設定する。

20

【0187】

ステップS125は、カレントPLMarkの番号に方向フラグの値を足した番号を、カレントPLMarkの番号として設定する。ここでSkipNextキーであるなら方向フラグは+1に設定されているのでカレントPLMarkはインクリメントされることになる。SkipBackキーであるなら方向フラグは-1に設定されているので、カレントPLMarkはデクリメントされることになる。

【0188】

ステップS126では、カレントPLMarkのref\_to\_PlayItem\_Idに記述されているPIを、PlayItem#xに設定し、ステップS127では、PlayItem#xのClip\_information\_file\_nameで指定されるClip情報を読み込む。ステップS128では、カレントClip情報のEP\_mapを用いて、カレントPLMarkのmark\_time\_stampを、Iピクチャアドレスuに変換する。一方ステップS129では、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に変換する。ステップS130は、カレントPLMarkのmark\_time\_stampからPlayItem#xのOut\_timeまでの出力をPresentation Engine 31に命じて、同時実行されている図50側の処理を停止させた上で、図50のステップS107に移行する。こうしてIピクチャアドレスu,vを変化して、別の部分の再生を命じた上でステップS107への移行するので、別のAVClipからTSパケットが読み出されることになり、映像内容が切り換えが実現する。

30

【0189】

図53は、Presentation Engine 31による処理手順の詳細を示すフローチャートである。本フローチャートは、IピクチャのPTSをカレントPTMに設定した後で(ステップS131)、ステップS132~ステップS137からなるループ処理を実行するものである。

40

続いてステップS132~ステップS137におけるループ処理について説明する。このループ処理は、カレントPTMにあたるピクチャ、オーディオの再生出力と、カレントPTMの更新とを繰り返すものである。本ループ処理におけるステップS136は、ループ処理の終了要件を規定している。つまりステップS136は、カレントPTMがPI#xのOut\_timeであることをループ処理の終了要件にしている。

【0190】

ステップS133は、早送りAPI、又は、早戻しAPIがJava(登録商標)仮想マシン38からコールされたか否かの判定である。もしコールされれば、ステップS138において

50

早送りか早戻しかの判定を行い、早送りであるなら、次のIピクチャのPTSをカレントPTMに設定する(ステップS 1 3 9)。このようにカレントPTMを、次のIピクチャのPTSに設定することで、1秒飛びにAVClipを再生してゆくことができる。これにより、2倍速等でAVClipは順方向に早く再生されることになる。早戻しであるなら、カレントPTMがPlayItem# xのOut\_timeに到達したかを判定する(ステップS 1 4 0)。もし到達していないのなら、1つ前のIピクチャのPTSをカレントPTMに設定する(ステップS 1 4 1)。このように読出先アドレスAを、1つ前のIピクチャに設定することで、AVClipを後方向に1秒飛びに再生してゆくことができる。これにより、2倍速等でAVClipは、逆方向に再生されることになる。尚、早送り、巻戻しを実行するにあたっての処理手順は多種多様なものである。ここで説明するのはあくまでも一例に過ぎないことに留意されたい。

10

## 【0191】

ステップS 1 3 4は、メニューコールAPIがコールされたか否かの判定であり、もしコールされれば、現在の再生処理をサスペンドして(ステップS 1 4 2)、メニュー処理用のメニュープログラムを実行する(ステップS 1 4 3)。以上の処理により、メニューメニューコールがなされた場合は、再生処理を中断した上で、メニュー表示のための処理が実行されることになる。

## 【0192】

ステップS 1 3 5は、sync\_PlayItem\_idにより、PlayItem# xを指定したSub-PlayItem# yが存在するか否かの判定であり、もし存在すれば、図5 4のフローチャートに移行する。図5 4は、SubPlayItemの再生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、先ずステップS 1 4 6において、カレントPTMはSubPlayItem# yのsync\_start\_PTS\_of\_playItemであるか否かを判定する。もしそうであれば、ステップS 1 5 3においてSubPlayItem# yに基づく再生処理を行うようPlayback Control Engine 3 2に通知する。

20

## 【0193】

ステップS 1 3 6がYesと判定された場合、ステップS 1 4 4、ステップS 1 4 5を実行する。ステップS 1 4 4は、Virtual File System 3 0からNotify End Of Fileイベントが出力され、尚且つ、デコーダからNotify End Of Decodingイベントが出力されたかを判定する。もし出力されれば、Notify End Of StreamイベントをPlayback Control Engine 3 2に出力する。

## 【0194】

図5 4のステップS 1 4 7～ステップS 1 5 2は、SubPlayItem# yに基づく再生処理を示すフローチャートである。

30

ステップS 1 4 7では、SubPlayItem# yのClip\_information\_file\_nameで指定されるClip情報を読み込む。ステップS 1 4 8では、カレントClip情報のEP\_mapを用いて、SubPlayItem# yのIn\_timeを、アドレス に変換する。一方ステップS 1 4 9では、SubPlayItem# yのOut\_timeを、カレントClip情報のEP\_mapを用いて、アドレス に変換する。ステップS 1 5 0は、SubPlayItem# yのIn\_timeからSubPlayItem# yのOut\_timeまでの出力をデコーダに命じる。これらの変換で得られたアドレス 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 に設定し(ステップS 1 5 1)、そして算出されたアドレス を用いて、SubClip# zにおけるアドレス からアドレス までのTSパケットの読み出しをBD-ROMドライブ1又はLocal Storage 1 8に命じるというものである(ステップS 1 5 2)。

40

## 【0195】

また図5 0に戻ってPlayback Control Engine 3 2の処理の説明の続きを行う。ステップS 1 1 3はPresentation Engine 3 1による再生制御が完了したかの判定であり、最後のPlayItem# xに対して、図5 3のフローチャートの処理が行われている限り、ステップS 1 1 3がNoになる。図5 3のフローチャートの処理が終了して初めて、ステップS 1 1 3はYesになりステップS 1 1 4に移行する。ステップS 1 1 4は、Java(登録商標)仮想マシン3 8へのNotifyイベントの出力であり、この出力により、2時間という再生時間の経過をJava(登録商標)仮想マシン3 8は知ることができる。

50

## 【0196】

以上のように本実施形態によれば、2時間という再生時間の経過時点アプリケーションマネージャ36は把握することができるので、プレイリストの再生終了と同期した処理をJava(登録商標)仮想マシン38に実行させることができる。

(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。

10

## 【0197】

(A)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、本発明の光ディスクは、記録される動的シナリオ、Index Tableに特徴があり、この特徴は、BD-ROMの物理的性質に依存するものではない。動的シナリオ、Index Tableを記録しうる記録媒体なら、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。また、コンパクトフラッシュ(登録商標)カード、スマートメディア、メモリスティック、マルチメディアカード、PCM-CIAカード等の半導体メモリカードであってもよい。フレキシブルディスク、SuperDisk,Zip,Clik!等の磁気記録ディスク(i)、ORB,Jaz,SparQ,SyJet,EZFly,マイクロドライブ等のリムーバルハードディスクドライブ(ii)であってもよい。更に、機器内蔵型のハードディスクであってもよい。

20

## 【0198】

(B)全ての実施形態における再生装置は、BD-ROMに記録されたAVClipをデコードした上でTVに出力していたが、再生装置をBD-ROMドライブのみとし、これ以外の構成要素をTVに具備させてもよい、この場合、再生装置と、TVとをIEEE1394で接続されたホームネットワークに組み入れることができる。また、実施形態における再生装置は、テレビと接続して利用されるタイプであったが、ディスプレイと一体型となった再生装置であってもよい。更に、各実施形態の再生装置において、処理の本質的部分をなす部分のみを、再生装置としてもよい。これらの再生装置は、何れも本願明細書に記載された発明であるから、これらの何れの態様であろうとも、各実施形態に示した再生装置の内部構成を元に、再生装置を製造する行為は、本願の明細書に記載された発明の実施行為になる。各実施形態に示した再生装置の有償・無償による譲渡(有償の場合は販売、無償の場合は贈与になる)、貸与、輸入する行為も、本発明の実施行為である。店頭展示、カタログ勧誘、パンフレット配布により、これらの譲渡や貸渡を、一般ユーザに申し出る行為も本再生装置の実施行為である。

30

## 【0199】

(C)各フローチャートに示したプログラムによる情報処理は、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、上記フローチャートに処理手順を示したプログラムは、単体で発明として成立する。全ての実施形態は、再生装置に組み込まれた態様で、本発明に係るプログラムの実施行為についての実施形態を示したが、再生装置から分離して、各実施形態に示したプログラム単体を実施してもよい。プログラム単体の実施行為には、これらのプログラムを生産する行為(1)や、有償・無償によりプログラムを譲渡する行為(2)、貸与する行為(3)、輸入する行為(4)、双方向の電子通信回線を介して公衆に提供する行為(5)、店頭展示、カタログ勧誘、パンフレット配布により、プログラムの譲渡や貸渡を、一般ユーザに申し出る行為(6)がある。

40

## 【0200】

(D)各フローチャートにおいて時系列に実行される各ステップの「時」の要素を、発明を特定するための必須の事項と考える。そうすると、これらのフローチャートによる処理手順は、再生方法の使用形態を開示していることがわかる。各ステップの処理を、時系列に行うことで、本発明の本来の目的を達成し、作用及び効果を奏するよう、これらのフロ

50

- チャ - トの処理を行うのであれば、本発明に係る記録方法の実施行為に該当することは  
いうまでもない。

【0201】

(E)BD-ROMに記録するにあたって、AVClipを構成する各TSパケットには、拡張ヘッダを  
付与しておくことが望ましい。拡張ヘッダは、TP\_extra\_headerと呼ばれ、  
『Arrival\_Time\_Stamp』と、『copy\_permission\_indicator』とを含み4バイトのデー  
タ長を有する。TP\_extra\_header付きTSパケット(以下EX付きTSパケットと略す)は、32個毎  
にグループ化されて、3つのセクタに書き込まれる。32個のEX付きTSパケットからなるグ  
ループは、6144バイト(=32×192)であり、これは3個のセクタサイズ6144バイト(=2048×3  
)と一致する。3個のセクタに収められた32個のEX付きTSパケットを " Aligned Unit " とい  
う。

10

【0202】

IEEE1394を介して接続されたホームネットワークでの利用時において、再生装置200  
は、以下のような送信処理にてAligned Unitの送信を行う。つまり送り手側の機器は、A  
ligned Unitに含まれる32個のEX付きTSパケットのそれぞれからTP\_extra\_headerを取り外  
し、TSパケット本体をDTCP規格に基づき暗号化して出力する。TSパケットの出力にあっ  
ては、TSパケット間の随所に、isochronousパケットを挿入する。この挿入箇所は、TP\_ex  
tra\_headerのArrival\_Time\_Stampに示される時刻に基づいた位置である。TSパケットの  
出力に伴い、再生装置200はDTCP\_Descriptorを出力する。DTCP\_Descriptorは、TP\_ext  
ra\_headerにおけるコピー許否設定を示す。ここで「コピー禁止」を示すようDTCP\_Desc  
riptionerを記述しておけば、IEEE1394を介して接続されたホームネットワークでの利用時にお  
いてTSパケットは、他の機器に記録されることはない。

20

【0203】

(F)各実施形態において、記録媒体に記録されるデジタルストリームはAVClipであつた  
が、DVD-Video規格、DVD-Video Recording規格のVOB(Video Object)であつてもよい。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/IEC1  
3818-1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方式であつてもよい。更にオーディオストリームは、Linear-PCM方式、DoIb  
y-AC3方式、MP3方式、MPEG-AAC方式、Dts、WMA(Windows(登録商標)media audio)であつ  
てもよい。

30

【0204】

(G)各実施形態における映像作品は、アナログ放送で放送されたアナログ映像信号をエ  
ンコードすることにより得られたものでもよい。デジタル放送で放送されたトランスポー  
トストリームから構成されるストリームデータであつてもよい。

またビデオテープに記録されているアナログ/デジタルの映像信号をエンコードしてコ  
ンテンツを得ても良い。更にビデオカメラから直接取り込んだアナログ/デジタルの映像  
信号をエンコードしてコンテンツを得ても良い。他にも、配信サーバにより配信されるデ  
ジタル著作物でもよい。

【0205】

(H)BD-Jモジュール35は、衛星放送受信のために機器に組み込まれたJava(登録商標  
)プラットフォームであつてもよい。BD-Jモジュール35がかかるJava(登録商標)プラ  
ットフォームであれば、本発明に係る再生装置は、MHP用STBとしての処理を兼用するこ  
とになる。

40

更に携帯電話の処理制御のために機器に組み込まれたJava(登録商標)プラットフォー  
ムであつてもよい。かかるBD-Jモジュール35がかかるJava(登録商標)プラットフォー  
ムであれば、本発明に係る再生装置は、携帯電話としての処理を兼用することになる。

【0206】

(I)レイアモデルにおいて、BD-Jモードの上にHDMVモードを配置してもよい。特にHDMV  
モードでの動的シナリオの解釈や、動的シナリオに基づく制御手順の実行は、再生装置に  
対する負担が軽いので、HDMVモードをBD-Jモード上で実行させても何等

50

問題は生じないからである。また再生装置や映画作品の開発にあたって、動作保証が1つのモードで済むからである。

【0207】

更にBD-Jモードだけで再生処理を実行してもよい。第5実施形態に示したように、BD-JモードでもPLの再生と同期した再生制御が可能になるから、強いてHDMVモードを設けなくてもよいという理由による。

(J)AVClipに多重化されるべきインタラクティブグラフィクスストリームにナビゲーションコマンドを設けて、あるPLから別のPLへの分岐を実現しても良い。(K)第1実施形態では、1つのBD-ROMに属する複数タイトルの全てを生存区間とするタイトルとしてタイトルアンバウンダリーアプリケーションを規定した。この他にも、複数のBD-ROMに属するタイトル

10

【0208】

(L)第1実施形態でアプリケーション管理テーブルを作成するにあたって、同時実行し得るアプリケーションの数は、例えば4個以下に制限することが望ましい。

アプリケーションの同時実行数を4個以下に制限する理由は以下の通りである。BD-ROMの再生装置は、デジタル放送のチューナ機能を具備しているものが多く、そのチューナ機能を実現するアプリケーションがメモリに常駐(レジデント)していることも多い。かかるレジデントアプリケーションが動作する余地を生むため、アプリケーションの数は4つ以下に制限されている。4つのアプリケーションの内訳として、1つ目をタイトルアンバウンダリーアプリケーション、2つ目をタイトルバウンダリーアプリケーション、3つ目をチャプターバウンダリーアプリケーションとしておくことが望ましい。

20

【0209】

(M)第2実施形態においてエラー管理テーブルは、1つのアプリケーションが異常終了した際、1つのリカバリー処理を実行するよう規定されていた。しかし1つのアプリケーションが異常終了した際、複数のリカバリー処理を実行するようにしてもよい。つまりあるアプリケーションが異常終了した際、プレイリスト再生を継続と、アプリケーションと再起動と、イベント出力とを再生装置に実行させてもよい。

【0210】

またアプリケーション毎ではなく、1つのタイトルにつき1つのリカバリー処理を規定するようエラー管理テーブルを構成してもよい。

30

(N)AVClipには、メニューを表示して対話的な操作を受け付けるためのインタラクティブグラフィクスストリームを多重化しておくことができるため、トップメニューを表示して対話的な操作を受け付けるようなAVClipを、単に再生させるようなナビゲーションコマンドをMovieObjectに記述しておくだけで、トップメニュータイトルを製作してもよい。

【0211】

(O)各実施形態におけるディレクトリ・ファイル構成及びファイル内のデータ構造は一例であり、本発明の特徴である管理情報は、ディレクトリ・ファイル構成及びファイル内のデータ構造に依存しない。例えばBD-Jモードの動作シナリオであるBD-J Objectは、識別子bobj\_idと、識別子BD-Jが付与されたファイル(ZZZZZ.BD-J)として、BDJAディレクトリに配置し、BD-J Object.bdmvのBD-J Object[n]()には、識別子bobj\_idのみを格納してよい。

40

【産業上の利用可能性】

【0212】

本発明に係る記録媒体及び再生装置は、ホームシアターシステムでの利用のように、個人的な用途で利用されることがありうる。しかし本発明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明らかなので、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、産業上の利用可能性を有する。

【図面の簡単な説明】

50

【 0 2 1 3 】

【 図 1 】 本発明に係る再生装置の使用行為についての形態を示す図である。

【 図 2 】 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。

【 図 3 】 PL情報の構成を示す図である。

【 図 4 】 AVClip時間軸と、PL時間軸との関係を示す図である。

【 図 5 】 4つのClip\_Information\_file\_nameによりなされた一括指定を示す図である。

【 図 6 】 PLmark情報の内部構成を示す図である。

【 図 7 】 PLmarkによるチャプター定義を示す図である。

【 図 8 】 SubPath情報の内部構成を示す図である。

【 図 9 】 SubPlayItem時間軸上の再生区間定義と、同期指定とを示す図である。

10

【 図 1 0 】 Movie Objectの内部構成を示す図である。

【 図 1 1 】 BD-J Objectの内部構成を示す図である。

【 図 1 2 】 ( a ) Java ( 登録商標 ) アーカイブファイルに収められているプログラム、データを示す図である。( b ) クラスファイルの内部構成を示す図である。

【 図 1 3 】 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 1 4 】 HDMVモードの動的シナリオから構成されるTitleを示す図である。

【 図 1 5 】 BD-Jモードの動的シナリオ(BD-J Object)により構成されるTitleの内部構成を示す図である。

【 図 1 6 】 プレイリスト管理テーブルをもたないTitleを示す図である。

【 図 1 7 】 HDMVモードのTitleから、BD-JモードのTitleへの分岐を示す図である。

20

【 図 1 8 】 BD-JモードのTitleから、HDMVモードのTitleへの分岐を示す図である。

【 図 1 9 】 index.bdmvの内部構成を示す図である。

【 図 2 0 】 ( 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。

【 図 2 1 】 ( 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 2 2 】 ( a ) ( b ) 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-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 2 3 】 図 2 2 ( 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 2 4 】 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。

30

【 図 2 5 】 ( 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。

【 図 2 6 】 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 7 】 ( 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 8 】 分岐先Title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て尚且つPresent、プレイリスト管理テーブル有りて尚且つ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。

40

【 図 2 9 】 ( 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 9 ( 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 3 0 】 ( 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 3 0 ( 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 3 1 】 ( a ) ~ ( d ) プレイリスト時間軸と、タイトルの再生区間との関係を示す図

50

である。

【図32】本発明に係る再生装置の内部構成を示す図である。

【図33】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。

【図34】Presentation Engine 31～モジュールマネージャ34による処理を模式化した図である。

【図35】アプリケーションマネージャ36による処理を模式化した図である。

【図36】User Event Manager 37～Default Operation Manager 40を示す図である。

【図37】Java(登録商標)仮想マシン38の内部構成を示す図である。

【図38】アプリケーション終了の4態様を示す図である。

10

【図39】アプリケーションマネージャ36による処理手順を示す図である。

【図40】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の具体例を示す図である。

【図41】(a)第2実施形態に係るBD-J Objectの内部構成を示す図である。(b)エラー管理テーブルの内部構成を示す図である。

【図42】エラー管理テーブルにおける5つのフラグの意味内容を示す。

【図43】(a)エラー管理テーブルが記述された2つのTitle(title#1、title#2)を示す図である。(b)図43(a)のように記載されたアプリケーション管理テーブル、エラー管理テーブル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。

20

【図44】第2実施形態に係るアプリケーションマネージャ36の処理手順を示すフローチャートである。

【図45】第2実施形態に係るアプリケーションマネージャ36の処理手順を示すフローチャートである。

【図46】アプリケーションマネージャ36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47】第3実施形態に係るアプリケーションマネージャ36の処理手順を示す図である。

【図48】(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。

30

【図49】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。

【図50】Playback Control Engine 32によるPL再生手順を示すフローチャートである。

【図51】アングル切替、SkipBack、SkipNextの受付手順を示すフローチャートである。

【図52】SkipBack、SkipNextAPIがコールされた際の処理手順を示すフローチャートである。

【図53】Presentation Engine 31による処理手順の詳細を示すフローチャートである。

40

【図54】SubPlayItemの再生手順を示すフローチャートである。

【符号の説明】

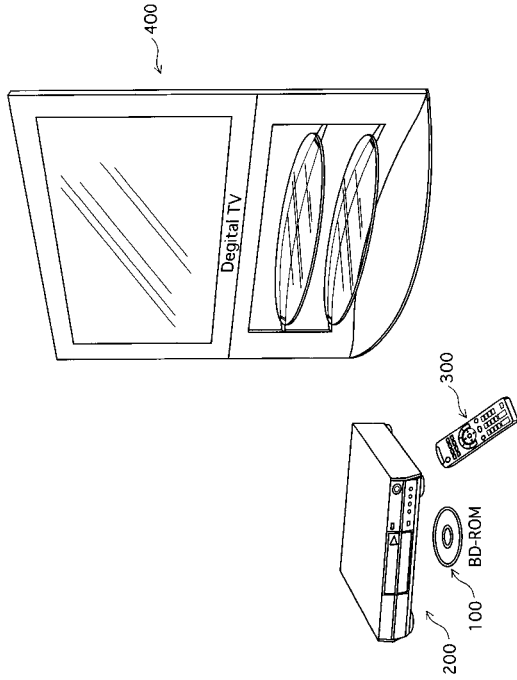
【0214】

- 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 P-Graphicsデコーダ
- 7 Presentation Graphicsプレーン

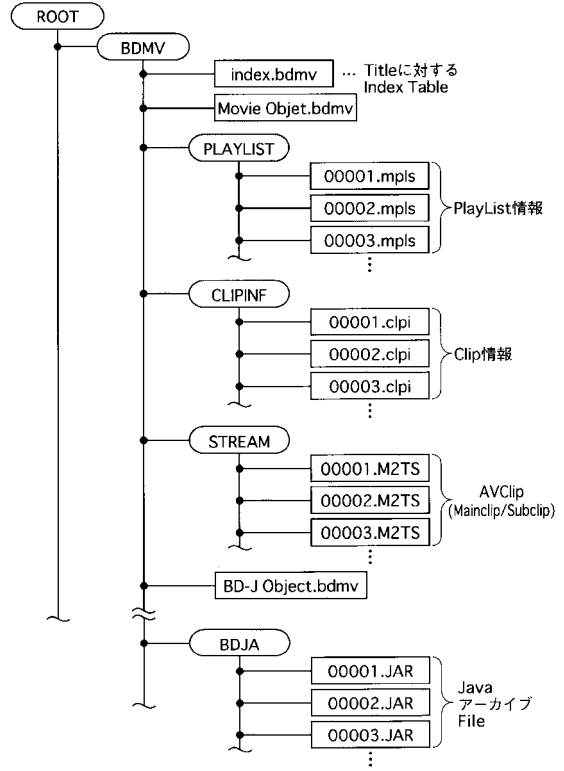
50

8 合成部	
9 フォントゼネレータ	
1 0 I-Graphicsデコーダ	
1 1 スイッチ	
1 2 Interctive Graphicsプレーン	
1 3 合成部	
1 4 CLUT部	
1 5 CLUT部	
1 6 オーディオデコーダ	
2 2 ユーザイベント処理部	10
2 3 PSRセット	
2 4 CPU	
2 5 シナリオメモリ	
2 6 ローカルメモリ	
3 3 HDMVモジュール	
3 4 モジュールマネージャ	
3 5 BD-Jモジュール	
3 6 アプリケーションマネージャ	
3 7 UOコントローラ	
3 8 Java (登録商標) 仮想マシン	20
4 1 PLMTプロセッサ	
4 2 パーミッションコントローラ	
5 2 ユーザクラスローダ	
5 3 メソッドエリア	
5 4 ワークメモリ	
5 5 a,b,...nスレッド	
5 6 a,b,...d Java (登録商標) スタック	

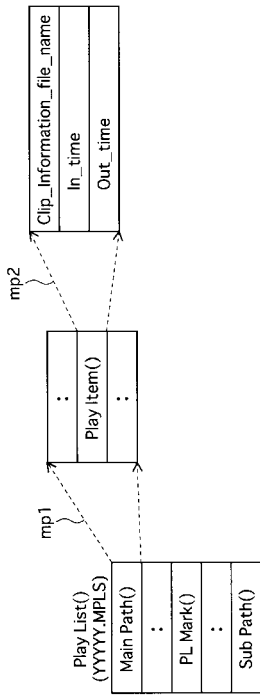
【 図 1 】



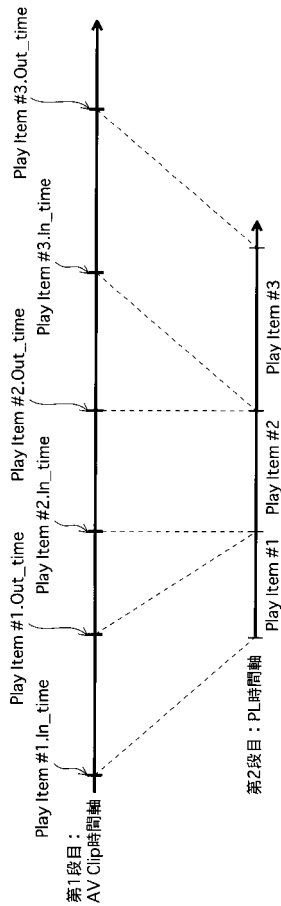
【 図 2 】



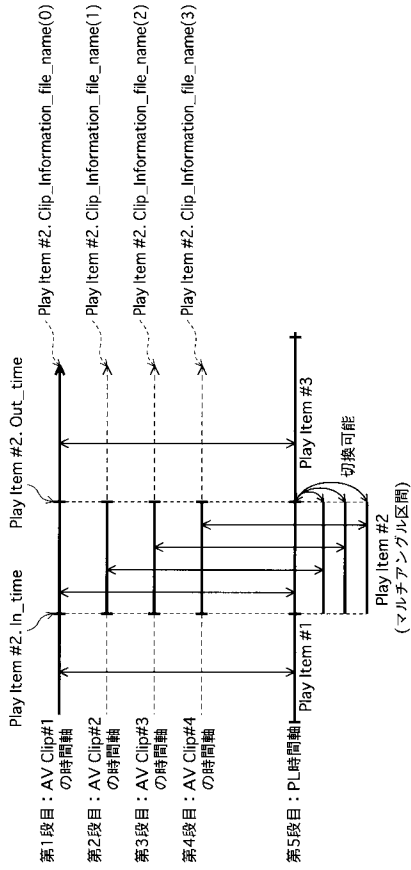
【 図 3 】



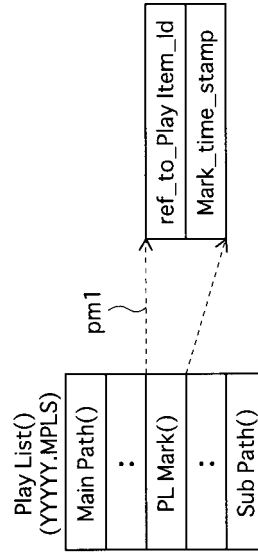
【 図 4 】



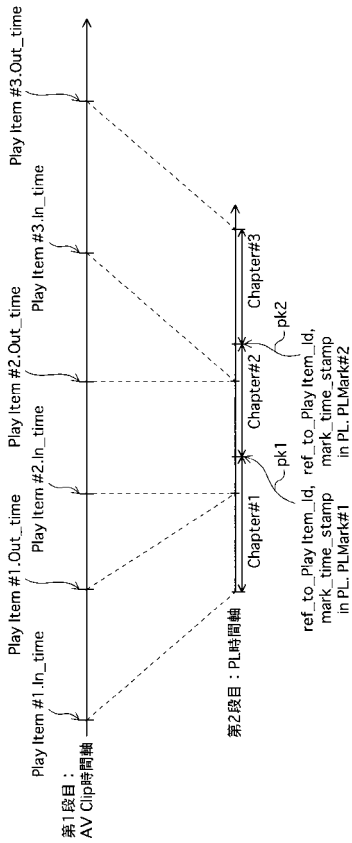
【 図 5 】



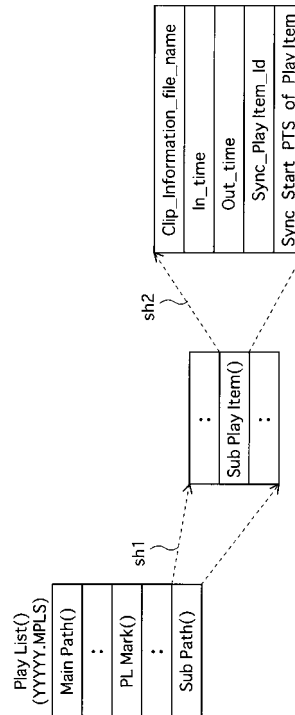
【 図 6 】



【 図 7 】

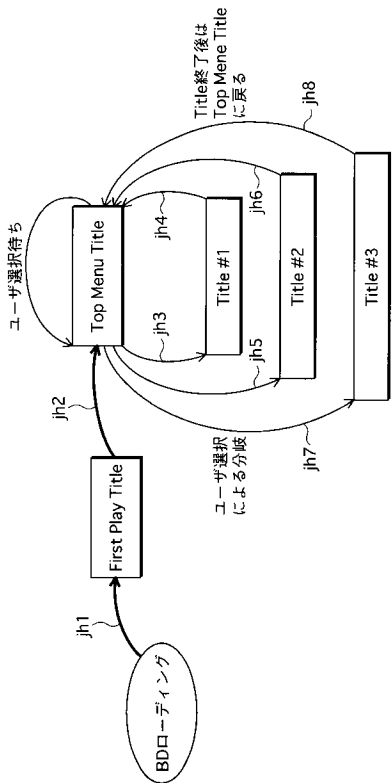


【 図 8 】

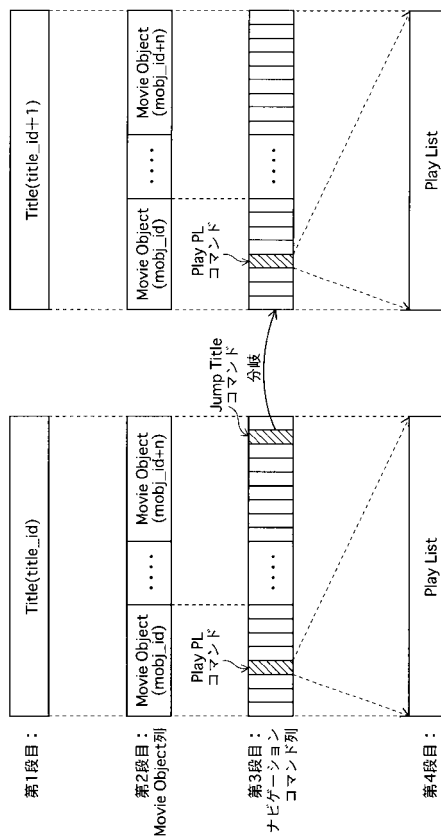




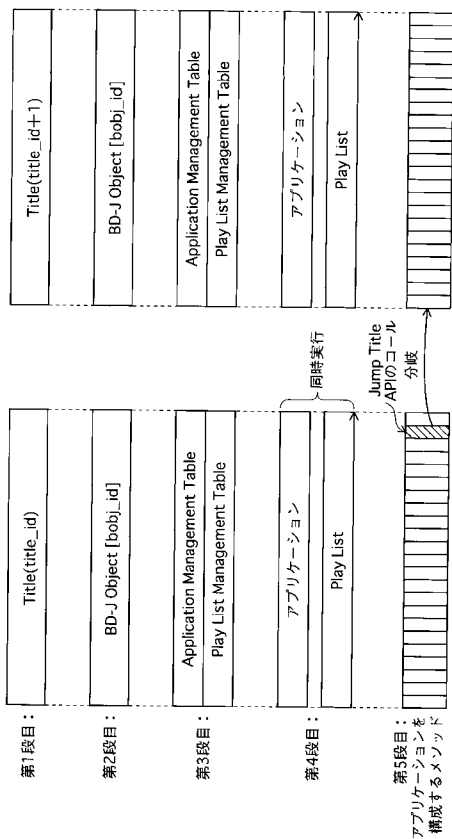
【 図 1 3 】



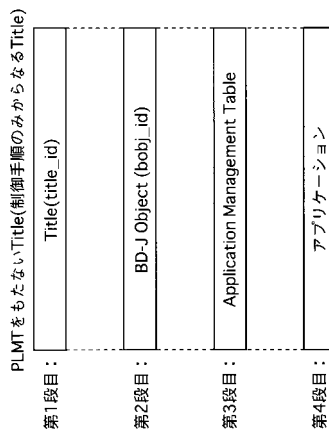
【 図 1 4 】



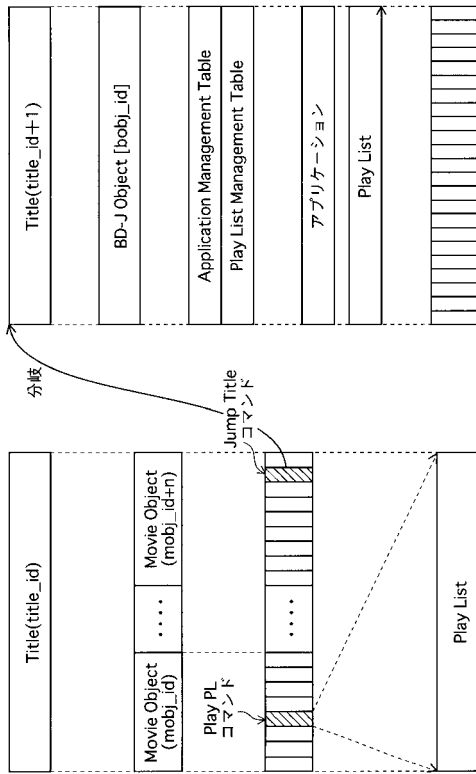
【 図 1 5 】



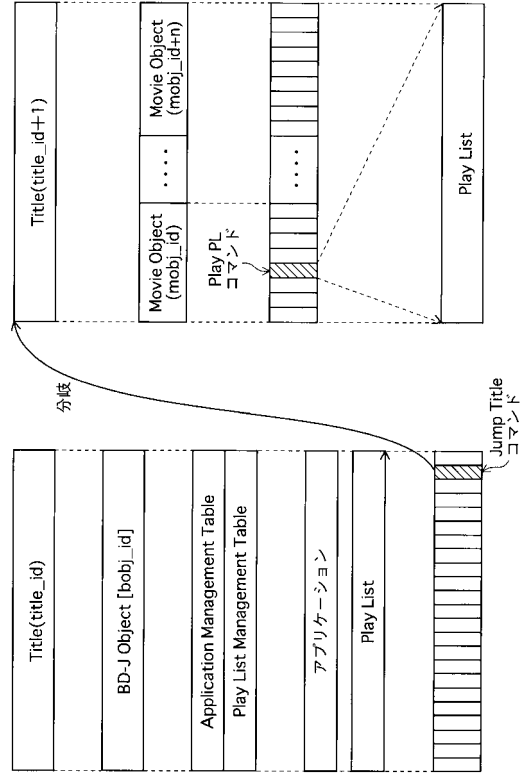
【 図 1 6 】



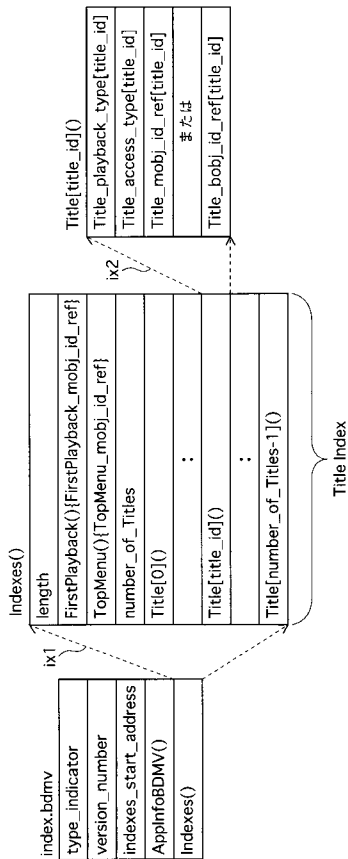
【 図 1 7 】



【 図 1 8 】



【 図 1 9 】



【 図 2 0 】

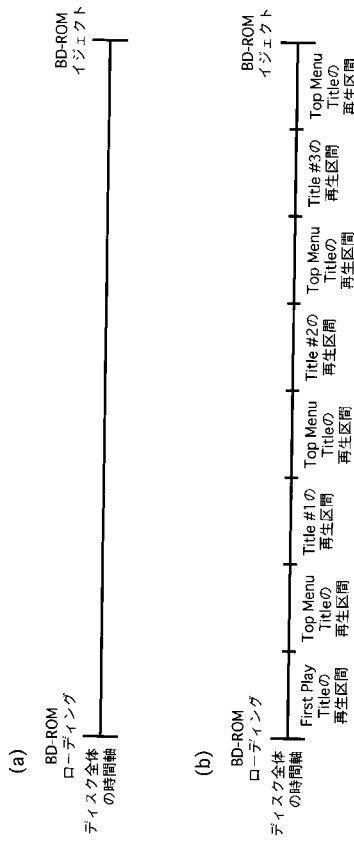
(a) Application Management Table (AMT)

life_cycle
apli_id_ref
run_attribute
run_priority

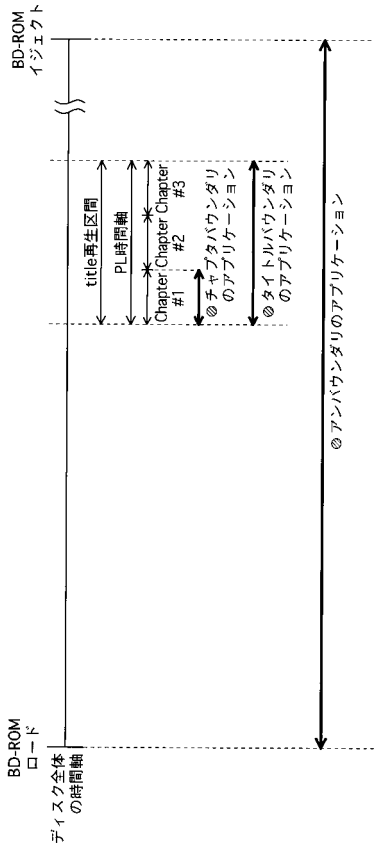
(b) Application Management Table(AMT)

life_cycle	apli_id_ref	run_attribute	run_priority
title_pl_chapterを 用いて表現された アプリケーション の“生存区間”	JARのファイル名 に付与された5桁 の整数ZZZZZである “アプリケーションID” の参照値	・Auto Run ・Present(無指定) ・Suspend の何れかである “起動属性”	0~255の値をとる “起動優先度”

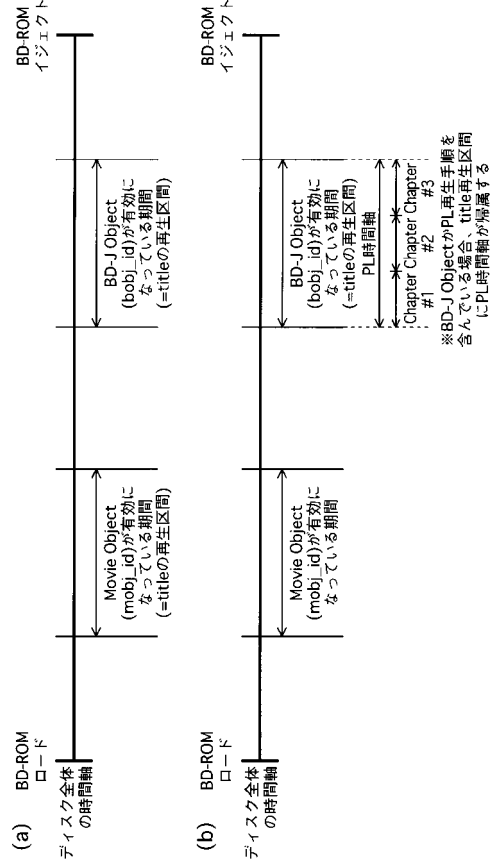
【 図 2 1 】



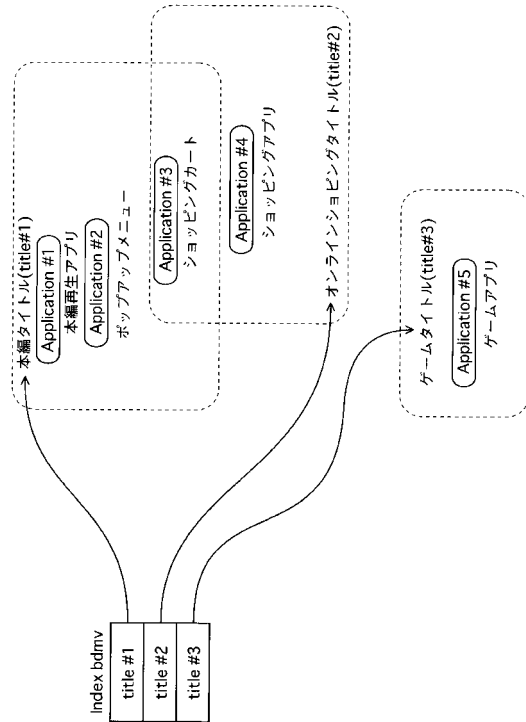
【 図 2 3 】



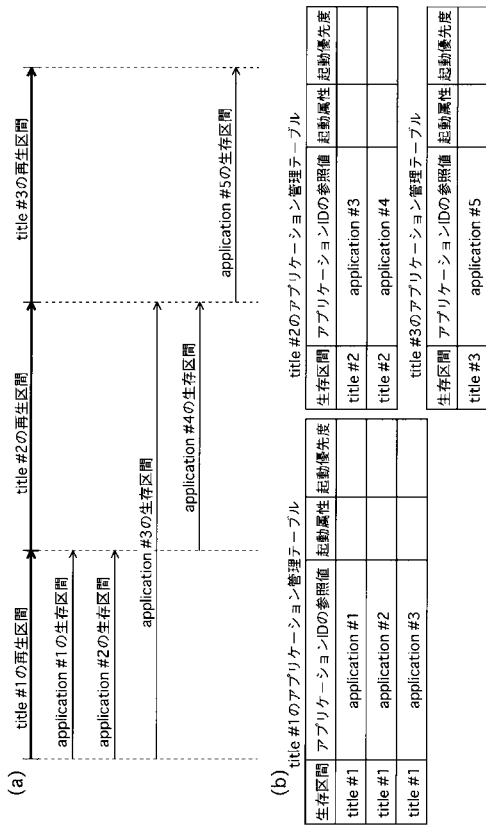
【 図 2 2 】



【 図 2 4 】



【 図 2 5 】



【 図 2 6 】

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	Suspend
非起動	何もせず、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	何もせず、状態継続
Suspend	レジュームする	レジュームする

【 図 2 7 】

(a) Play List Management Table (PLMT)

PL_id_ref
Playback_Attribute

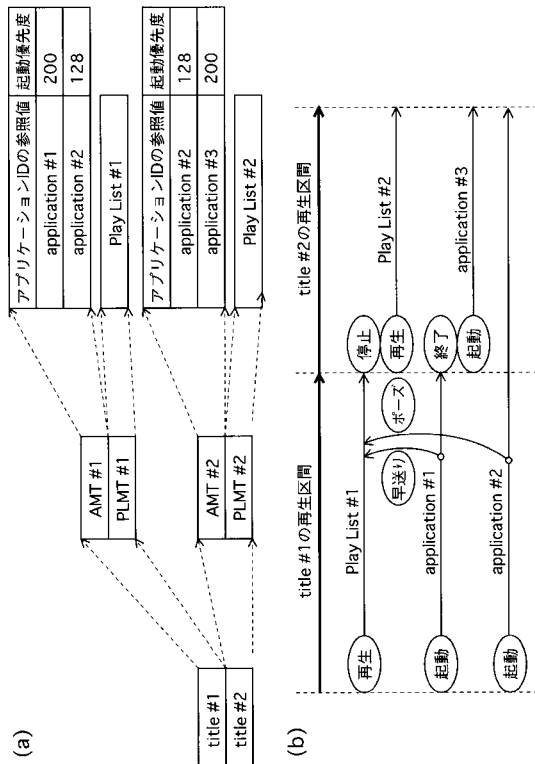
(b) Play List Management Table (PLMT)

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"	<ul style="list-style-type: none"> <li>Auto Play</li> <li>Present(無指定) の何れかである "再生属性"</li> </ul>

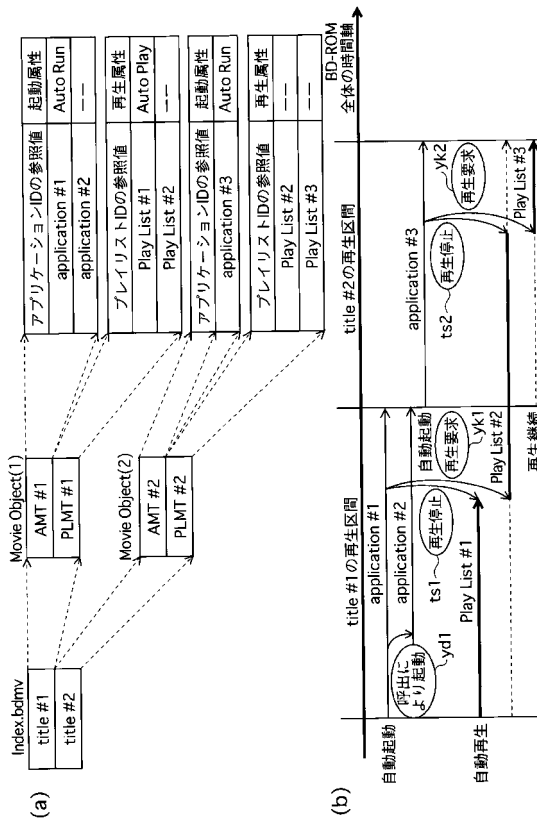
【 図 2 8 】

	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	状態継続	状態継続
分岐元titleは非再生状態	自動再生開始	状態継続

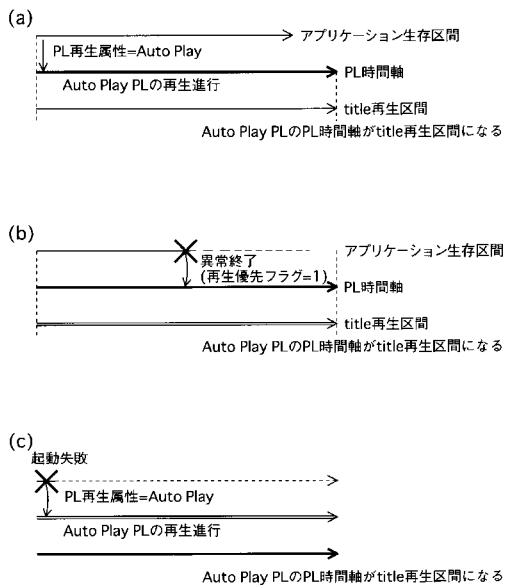
【 図 2 9 】



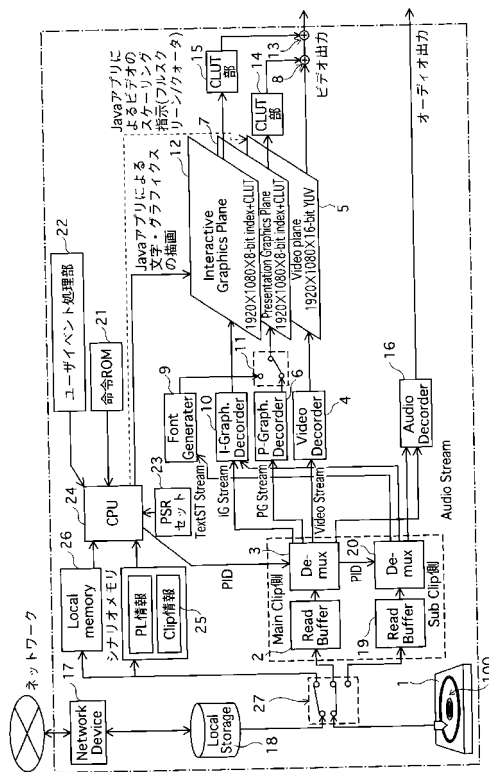
【 図 3 0 】



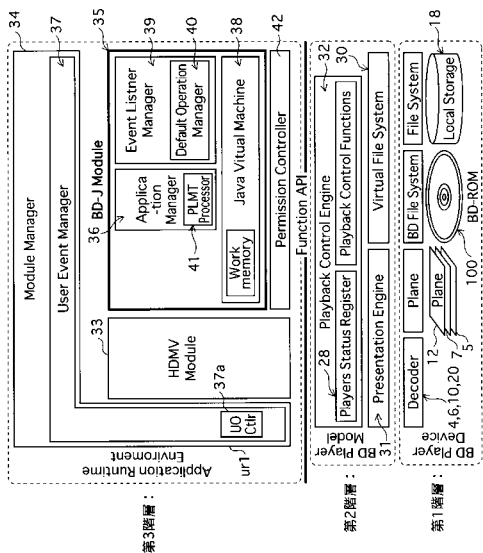
【 図 3 1 】



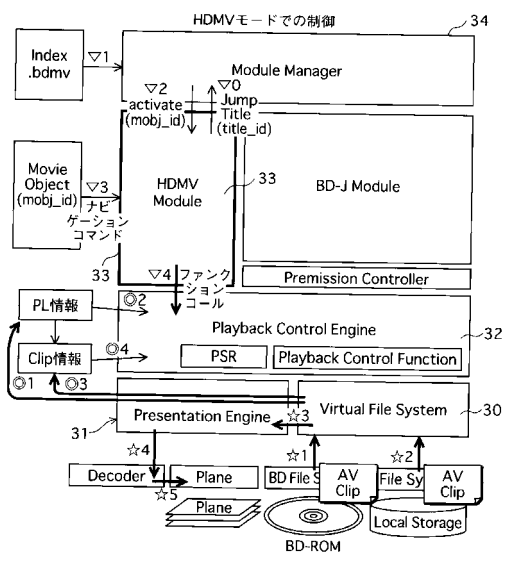
【 図 3 2 】



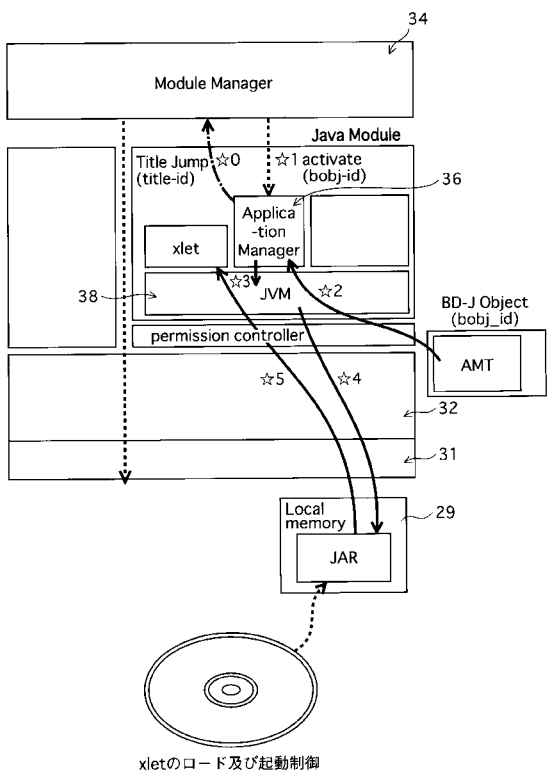
【 図 3 3 】



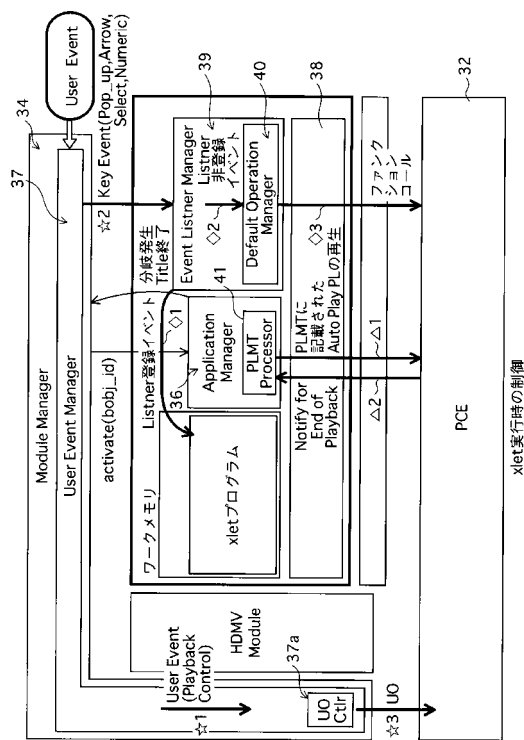
【 図 3 4 】



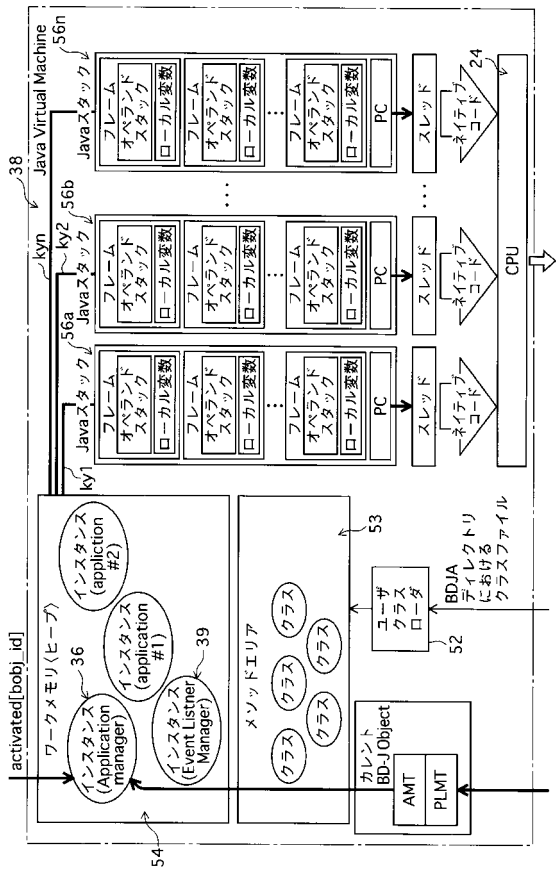
【 図 3 5 】



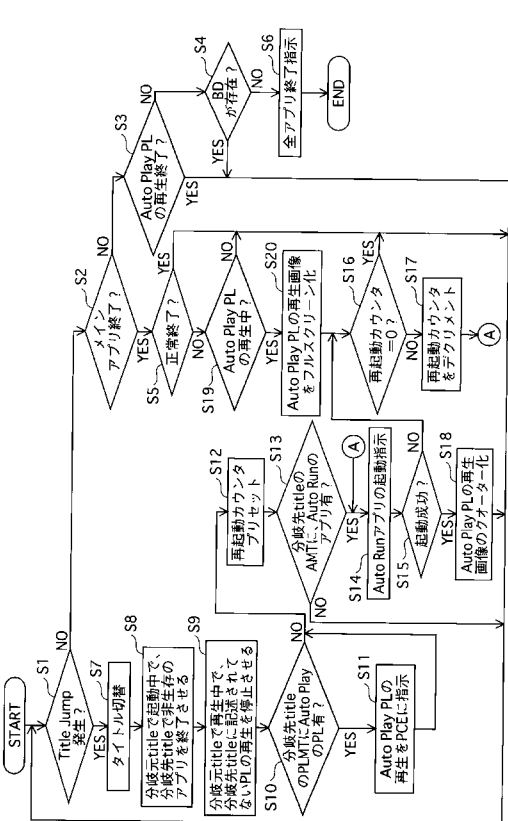
【 図 3 6 】



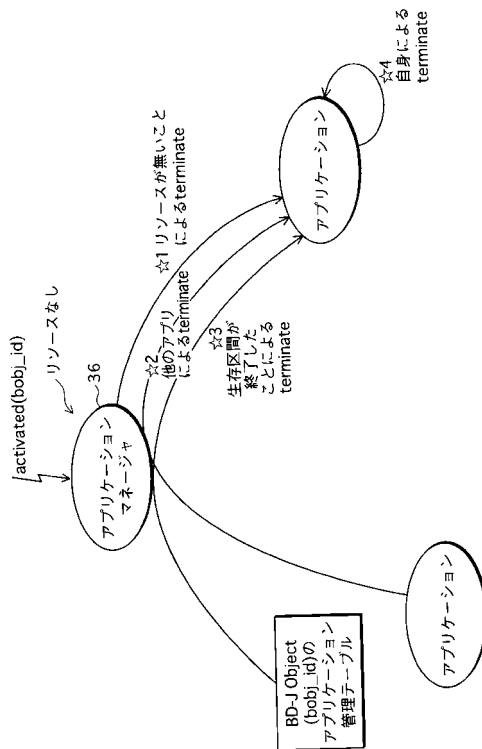
【 図 3 7 】



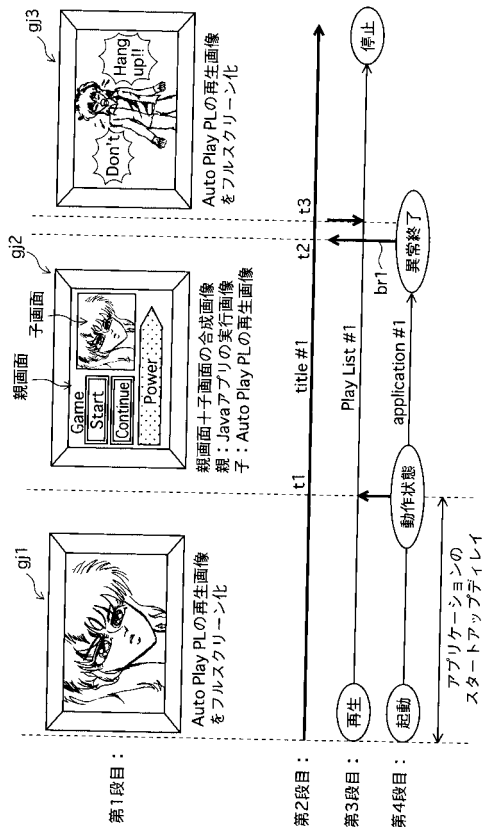
【 図 3 9 】



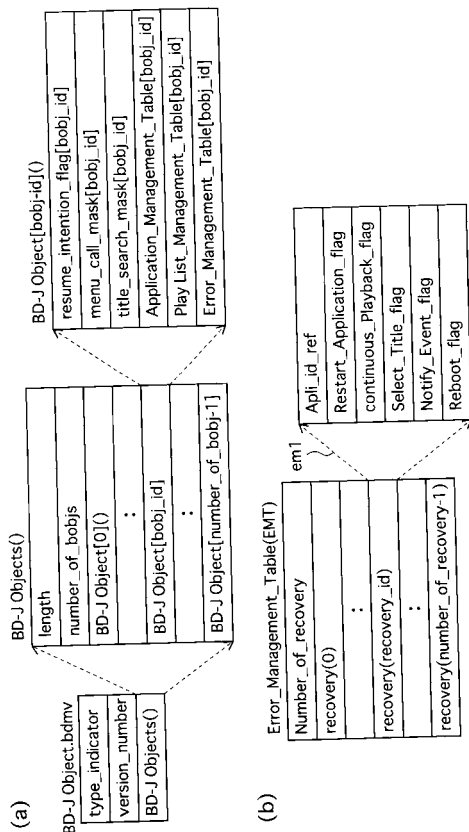
【 図 3 8 】



【 図 4 0 】



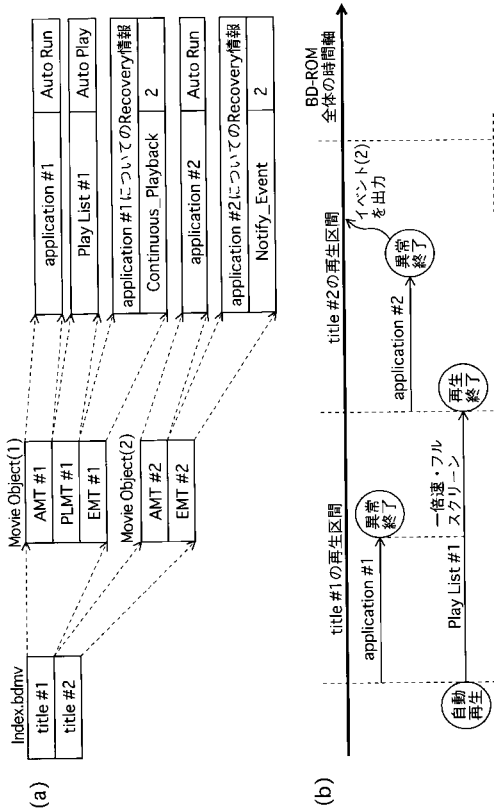
【 図 4 1 】



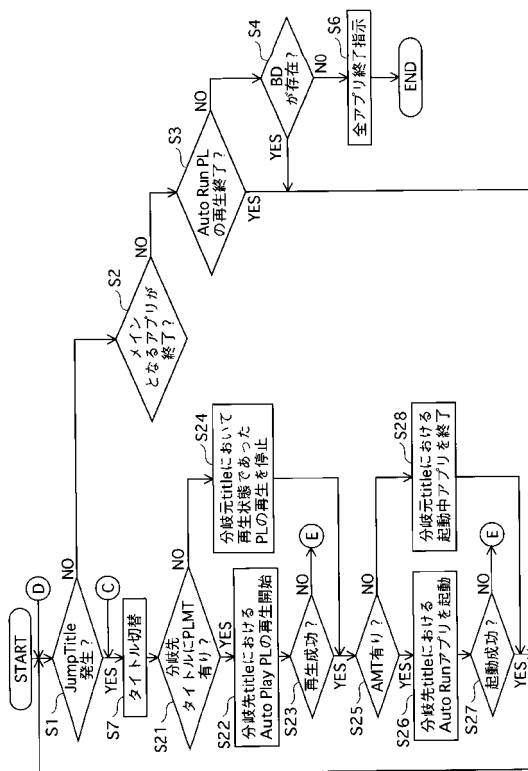
【 図 4 2 】

Recovery()	値	デフォルト
Restart_Application_Flag (アプリ再起動)	0: No n: 再起動回数	0
Continuous_Playback_Flag (プレイリスト再生継続)	0: No 1: Yes 2: フルスクリーン/一倍速による再生	0
Select_Title_Flag (タイトル選択)	0: No n: タイトル番号	0
Notify_Event_Flag (イベント通知)	0: No n: イベント番号	1
Reboot_Flag (システム再起動)	0: No 1: Yes	0

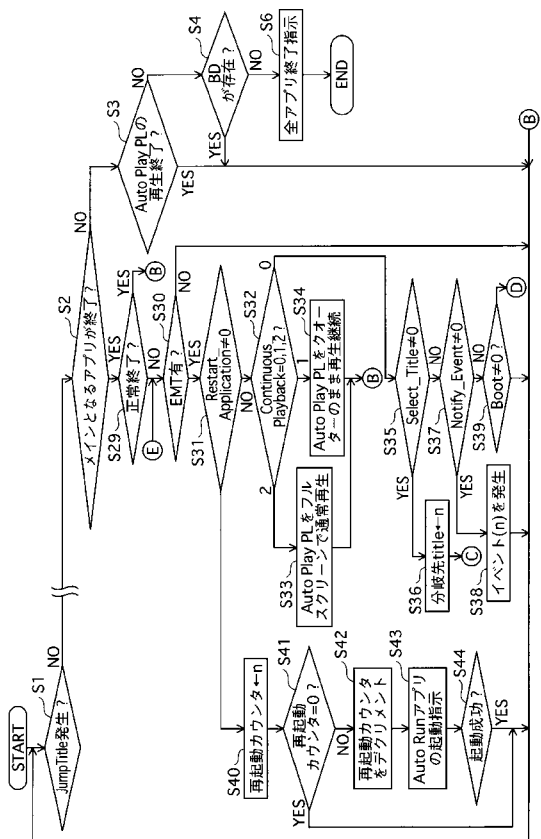
【 図 4 3 】



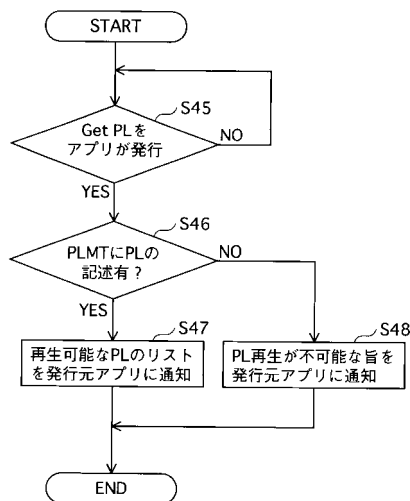
【 図 4 4 】



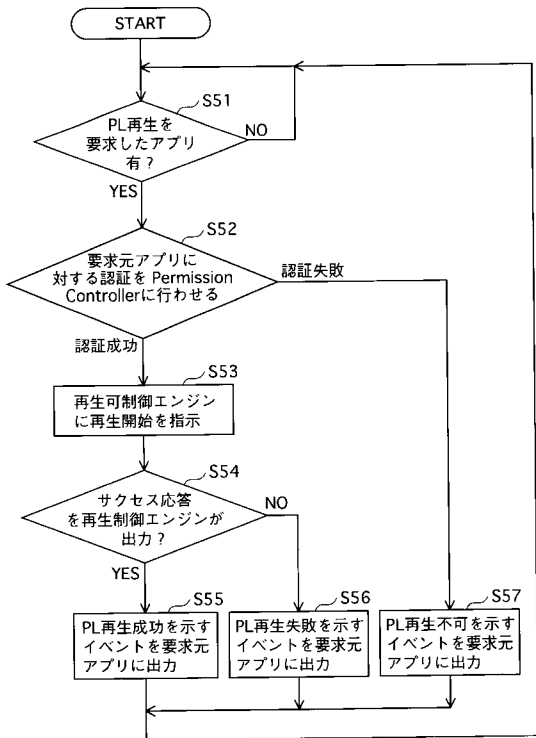
【 図 4 5 】



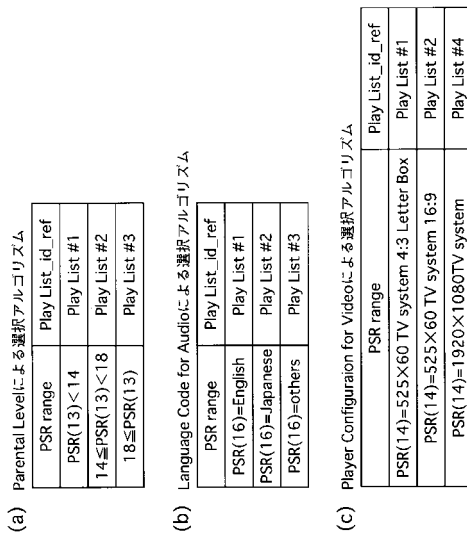
【 図 4 6 】



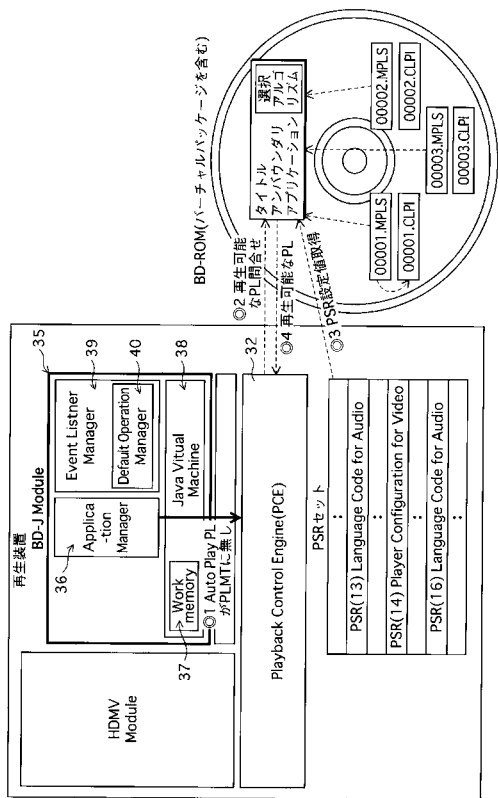
【 図 4 7 】



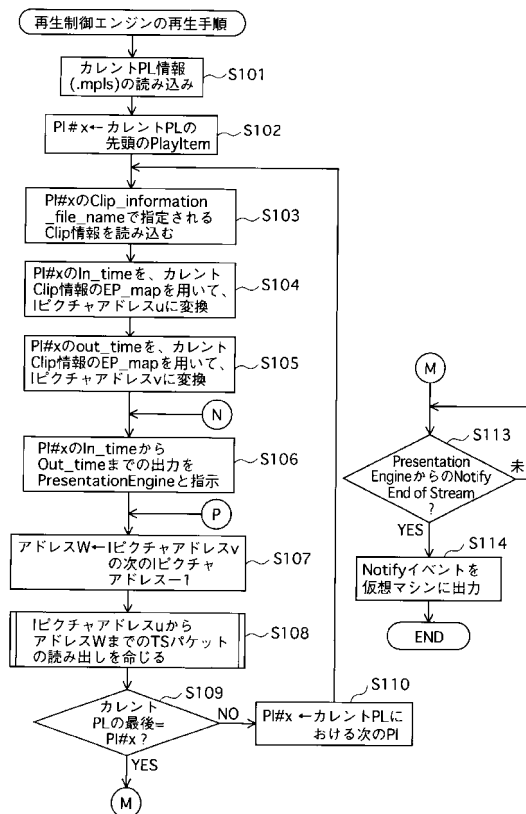
【 図 4 8 】



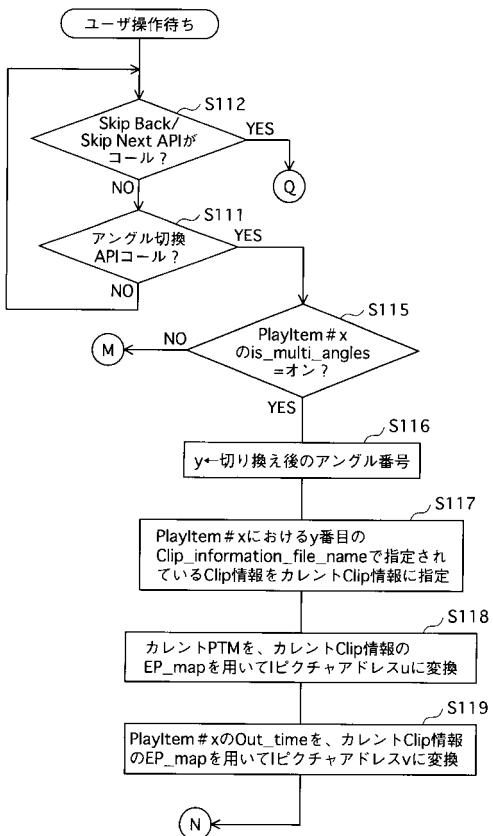
【図 49】



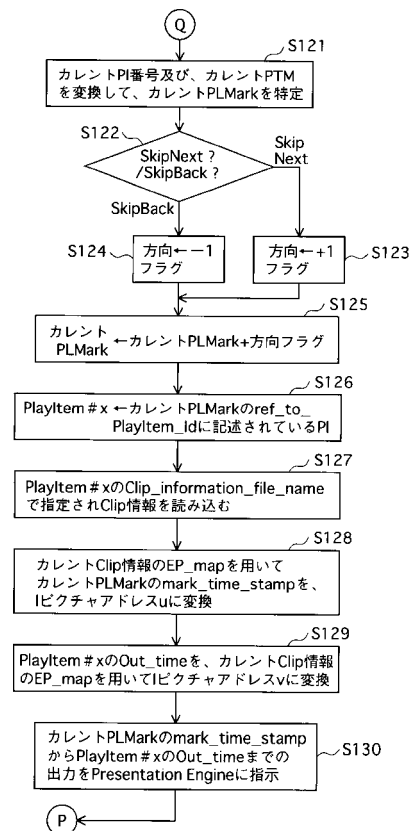
【図 50】



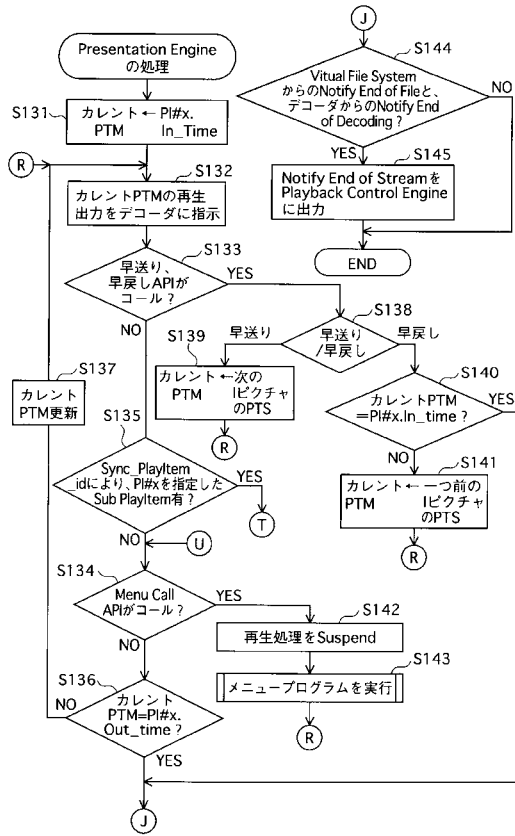
【図 51】



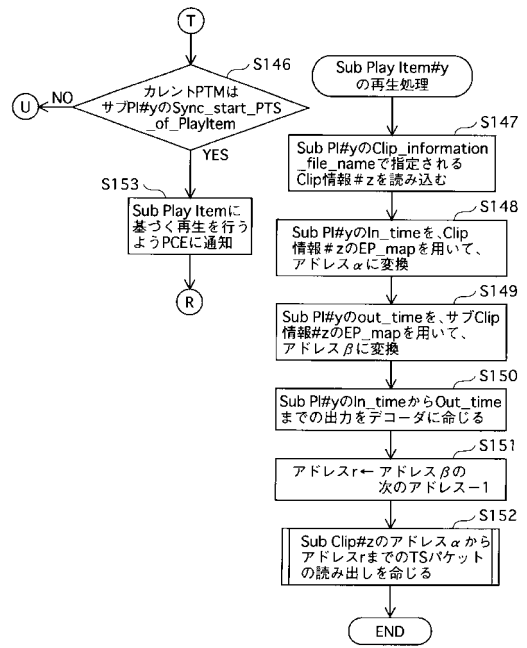
【図 52】



【 図 5 3 】



【 図 5 4 】



## フロントページの続き

(51)Int.Cl.	F I	テーマコード(参考)
<b>G 1 1 B 27/00 (2006.01)</b>	G 1 1 B 27/10	A
	G 1 1 B 27/00	D

(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 大久保 雅文

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

Fターム(参考) 5C052 AA04 AB03 AB05 DD04

5C053 FA17 FA24 GB06

5D044 AB05 AB07 BC02 CC06 DE48 DE60 FG18 GK12 GK19

5D077 AA23 BA14 BA30 EA31 EB10

5D110 AA14 AA27 AA29 DA03 DA11 DA18 DB19