

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4332195号  
(P4332195)

(45) 発行日 平成21年9月16日(2009.9.16)

(24) 登録日 平成21年6月26日(2009.6.26)

(51) Int.Cl.

F 1

G 1 1 B 27/10	(2006.01)	G 1 1 B 27/10	A
G 1 1 B 20/10	(2006.01)	G 1 1 B 20/10	3 2 1 Z
G 1 1 B 27/00	(2006.01)	G 1 1 B 27/00	D
H 04 N 5/93	(2006.01)	H 04 N 5/93	Z

請求項の数 3 (全 45 頁)

(21) 出願番号	特願2007-504195 (P2007-504195)	(73) 特許権者	000005821 パナソニック株式会社 大阪府門真市大字門真1006番地
(86) (22) 出願日	平成17年7月21日 (2005.7.21)	(74) 代理人	100090446 弁理士 中島 司朗
(65) 公表番号	特表2008-507795 (P2008-507795A)	(74) 代理人	100072442 弁理士 松村 修治
(43) 公表日	平成20年3月13日 (2008.3.13)	(74) 代理人	100125597 弁理士 小林 国人
(86) 國際出願番号	PCT/JP2005/013785	(72) 発明者	田中 敬一 大阪府門真市大字門真1006番地 松下 電器産業株式会社内
(87) 國際公開番号	W02006/009305	(72) 発明者	大戸 英隆 大阪府門真市大字門真1006番地 松下 電器産業株式会社内
(87) 國際公開日	平成18年1月26日 (2006.1.26)		
審査請求日	平成20年7月8日 (2008.7.8)		
(31) 優先権主張番号	特願2004-214916 (P2004-214916)		
(32) 優先日	平成16年7月22日 (2004.7.22)		
(33) 優先権主張国	日本国 (JP)		
(31) 優先権主張番号	特願2004-268512 (P2004-268512)		
(32) 優先日	平成16年9月15日 (2004.9.15)		
(33) 優先権主張国	日本国 (JP)		

最終頁に続く

(54) 【発明の名称】アプリケーション連動再生を行う再生装置、再生方法、及び再生プログラム

## (57) 【特許請求の範囲】

## 【請求項 1】

実行中のアプリケーションに連動させながらデジタルストリームを再生する再生装置であって、

記録媒体に記録されているファイルと、リライタブル記録媒体に記録されているファイルを、前記リライタブル記録媒体に記録されている、前記記録媒体に組み合わせるためのファイルを示すマージ管理情報に従って組み合わせることにより、仮想パッケージを生成すること、および前記仮想パッケージの生成に用いるマージ管理情報が示すファイルを読み取り専用属性に変更することを行なうパッケージ管理手段と、

前記仮想パッケージから、再生可能な複数のタイトルを検出し、そのうち1つをカレントタイトルとして選択する選択手段と、

前記カレントタイトルに関連するアプリケーションを実行するアプリケーション実行手段と、

前記仮想パッケージに含まれ、前記カレントタイトルにおいて示されるプレイリスト情報に従って、デジタルストリームの再生を制御する再生制御手段とを備え、

前記仮想パッケージの生成に用いたマージ管理情報は、前記リライタブル記録媒体に記録されている第1のファイルの名前および当該第1のファイルへのファイルパスと、当該第1のファイルが前記生成した仮想パッケージにおいて利用されるときのファイルの名前および前記生成した仮想パッケージにおいて利用されるファイルへのファイルパスとを含むファイル名マッピング情報を含んでおり、

10

20

実行された前記アプリケーションが、前記仮想パッケージの生成に用いたマージ管理情報とは異なり、かつ前記リライタブル記録媒体に記録されている新たなマージ管理情報を指定することで、前記パッケージ管理手段に前記生成した仮想パッケージの更新を要求すると、

前記パッケージ管理手段は、

前記仮想パッケージの更新を行う前に、前記指定した新たなマージ管理情報が示す、前記リライタブル記録媒体に記録されているファイルを読み取り専用属性に変更し、

前記選択手段による新たなカレントタイトルの選択により、前記カレントタイトルにおいて示されるプレイリスト情報に従うデジタルストリームの再生を再生制御手段が停止した時点で、前記指定した新たなマージ管理情報に従って、前記記録媒体に記録されているファイルと、前記リライタブル記録媒体に記録されているファイルを組み合わせ、新たな仮想パッケージを生成するとともに、

実行された前記アプリケーションが前記生成した仮想パッケージ上のファイルパスおよびファイルの名前を指定してアクセスを要求すると、前記仮想パッケージの生成に用いたマージ管理情報に含まれるファイル名マッピング情報に基づいて、当該指定した名前のファイルが第1のファイルである場合には、前記第1のファイルの格納場所により特定される前記リライタブル記録媒体に記録されている第1のファイルへアクセスさせる

ことを特徴とする再生装置。

#### 【請求項2】

実行中のアプリケーションに連動させながらデジタルストリームを再生する再生方法であって、

記録媒体に記録されているファイルと、リライタブル記録媒体に記録されているファイルを、前記リライタブル記録媒体に記録されている、前記記録媒体に組み合わせるためのファイルを示すマージ管理情報に従って組み合わせることにより、仮想パッケージを生成する仮想パッケージ生成ステップと、

前記仮想パッケージの生成に用いるマージ管理情報が示すファイルを読み取り専用属性に変更する属性変更ステップと、

前記仮想パッケージから、再生可能な複数のタイトルを検出し、そのうち1つをカレントタイトルとして選択する選択ステップと、

前記カレントタイトルに関連するアプリケーションを実行するアプリケーション実行ステップと、

前記仮想パッケージに含まれ、前記カレントタイトルにおいて示されるプレイリスト情報に従って、デジタルストリームの再生を制御する再生制御ステップとを備え、

前記仮想パッケージの生成に用いたマージ管理情報は、前記リライタブル記録媒体に記録されている第1のファイルの名前および当該第1のファイルへのファイルパスと、当該第1のファイルが前記生成した仮想パッケージにおいて利用されるときのファイルの名前および前記生成した仮想パッケージにおいて利用されるファイルへのファイルパスとを含むファイル名マッピング情報を含んでおり、

実行された前記アプリケーションが、前記仮想パッケージの生成に用いたマージ管理情報とは異なり、かつ前記リライタブル記録媒体に記録されている新たなマージ管理情報を指定することにより、前記マージ管理情報の更新を要求すると、

前記仮想パッケージの更新を行う前に、前記指定した新たなマージ管理情報が示す、前記リライタブル記録媒体に記録されているファイルは、前記属性変更ステップにより読み取り専用属性に変更され、

前記選択ステップにおける新たなカレントタイトルの選択により、前記カレントタイトルにおいて示されるプレイリスト情報に従うデジタルストリームの再生が再生制御ステップにより停止された時点で、前記仮想パッケージ生成ステップにおいて、前記指定した新たなマージ管理情報に従って、前記記録媒体に記録されているファイルと、前記リライタブル記録媒体に記録されているファイルを組み合わせ、新たな仮想パッケージが生成されるとともに、

10

20

30

40

50

実行された前記アプリケーションが前記生成した仮想パッケージ上のファイルパスおよびファイルの名前を指定してアクセスを要求すると、前記仮想パッケージの生成に用いたマージ管理情報に含まれるファイル名マッピング情報に基づいて、当該指定した名前のファイルが第1のファイルである場合には、前記第1のファイルの格納場所により特定される前記リライタブル記録媒体に記録されている第1のファイルへアクセスさせることを特徴とする再生方法。

【請求項3】

実行中のアプリケーションに連動させながらデジタルストリームの再生をコンピュータに実行させるための再生プログラムであって、

記録媒体に記録されているファイルと、リライタブル記録媒体に記録されているファイルを、前記リライタブル記録媒体に記録されている、前記記録媒体に組み合わせるためのファイルを示すマージ管理情報に従って組み合わせることにより、仮想パッケージを生成する仮想パッケージ生成ステップと、

前記仮想パッケージの生成に用いるマージ管理情報が示すファイルを読み取り専用属性に変更する属性変更ステップと、

前記仮想パッケージから、再生可能な複数のタイトルを検出し、そのうち1つをカレントタイトルとして選択する選択ステップと、

前記カレントタイトルに関連するアプリケーションを実行するアプリケーション実行ステップと、

前記仮想パッケージに含まれ、前記カレントタイトルにおいて示されるプレイリスト情報に従って、デジタルストリームの再生を制御する再生制御ステップとを備え、

前記仮想パッケージの生成に用いたマージ管理情報は、前記リライタブル記録媒体に記録されている第1のファイルの名前および当該第1のファイルへのファイルパスと、当該第1のファイルが前記生成した仮想パッケージにおいて利用されるときのファイルの名前および前記生成した仮想パッケージにおいて利用されるファイルへのファイルパスとを含むファイル名マッピング情報を含んでおり、

実行された前記アプリケーションが、前記仮想パッケージの生成に用いたマージ管理情報とは異なり、かつ前記リライタブル記録媒体に記録されている新たなマージ管理情報を指定することにより、前記マージ管理情報の更新を要求すると、

前記仮想パッケージの更新を行う前に、前記指定した新たなマージ管理情報が示す、前記リライタブル記録媒体に記録されているファイルは、前記属性変更ステップにより読み取り専用属性に変更され、

前記選択ステップにおける新たなカレントタイトルの選択により、前記カレントタイトルにおいて示されるプレイリスト情報に従うデジタルストリームの再生が再生制御ステップにより停止された時点で、前記仮想パッケージ生成ステップにおいて、前記指定した新たなマージ管理情報に従って、前記記録媒体に記録されているファイルと、前記リライタブル記録媒体に記録されているファイルを組み合わせ、新たな仮想パッケージが生成されるとともに、

実行された前記アプリケーションが前記生成した仮想パッケージ上のファイルパスおよびファイルの名前を指定してアクセスを要求すると、前記仮想パッケージの生成に用いたマージ管理情報に含まれるファイル名マッピング情報に基づいて、当該指定した名前のファイルが第1のファイルである場合には、前記第1のファイルの格納場所により特定される前記リライタブル記録媒体に記録されている第1のファイルへアクセスさせることを特徴とする再生プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、デジタルストリームとアプリケーション連動再生の技術分野に属する発明である。

【背景技術】

10

20

30

40

50

## 【0002】

アプリケーション連動再生とは、映像を再生する再生装置等において、Java（登録商標、以下同様）アプリケーションを実行しながら、デジタルストリームの再生を行う技術であり、民生機器の製造分野において今後大いに注目される技術分野である。連動すべきアプリケーション及びデジタルストリームは、“タイトル”と呼ばれる、再生単位において対応づけられる。以降、従来の再生装置について説明する。従来の再生装置は、仮想ファイルシステム部、モジュールマネージャ、再生部、プラットフォーム部から構成される。

## 【0003】

仮想ファイルシステム部は、BD-ROM、DVD-ROMなどの光記録媒体（以後光記録媒体の一例としてBD-ROMを例に説明をする）やハードディスク等の磁気記録媒体といったような、再生装置がアクセスすることができる複数の記録媒体を管理し、これらの記録内容を一體的に示すパッケージ情報を生成する。

この生成されたパッケージ情報はVirtual Packageと呼ばれるものであり、これらの記録媒体に記録されているデジタルストリームやアプリケーションプログラム（以下、「アプリケーション」という。）を、あたかも仮想的な一つのパッケージに記録されているかのごとく再生や実行に供するものである。

## 【0004】

モジュールマネージャは、複数のタイトルのうち、1つをカレントタイトルとして選択する。

再生部は、パッケージ情報に示されているデジタルストリームのうち、カレントタイトルを構成するデジタルストリーム（光記録媒体または磁気記録媒体に記録されているもの）を再生する。

## 【0005】

プラットフォーム部は、パッケージ情報に示されているアプリケーションのうち、カレントタイトルを構成するアプリケーション（光記録媒体または磁気記録媒体に記録されているもの）を実行する。

以上の構成により、再生装置はBD-ROMやハードディスク等、別々の記録媒体に記録されているデジタルストリームやアプリケーションを、仮想的な一つのパッケージとして取り扱うことができ、再生装置は、タイトルの単位で再生制御を実現することができる。なお、複数の記録媒体上のファイルをマージして構成されるパッケージメディアに関しては、以下の特許文献1に記載されている先行技術がある。

【特許文献1】特開2001-517836号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0006】

ところで、上述した構成においてタイトルは、BD-ROMに記録されているデジタルストリームのみならず、ハードディスクに記録されているデジタルストリームをも再生の対象にしている。ハードディスクは、書き換え可能な記録媒体であるので、新たに取得したデジタルストリーム等でパッケージ情報を新たに生成する場合など、タイトルの構成要素が、再生処理の途中で入れ代わってしまうことがあり得る。

## 【0007】

かかる入れ代わりが生じた結果、再生部が異常動作を行い、再生映像のブラックアウト等、復旧困難な状態を招くことが大いに考えられる。

再生の客体が、再生の途中で入れ代わるようなケースを事前に察知するには、再生装置内の再生部が、ストリームや再生制御情報等、再生対象の同一性を逐次チェックすればよい。しかしチェックすべき情報の種類は多岐にわたるため、かかるチェックは、再生部に大きな負担をもたらす。

## 【0008】

装置の安定動作に万全を期するには、ハードディスクにおける記録内容に変動があった際、ソフトウェアのインストール時のように、装置自体の再起動を行うのが望ましい。し

10

20

30

40

50

かし装置自体の再起動は、数分、数秒という処理時間がかかるので、やすやすと行う訳にはいかない。

本発明の目的は、再生処理の客体に変動があったとしても、再生処理の安定を実現することができる再生装置を提供することである。

【課題を解決するための手段】

【0009】

上記課題を解決するために、本発明は、実行中のアプリケーションに連動させながらデジタルストリームを再生する再生装置であって、記録媒体に記録されているファイルと、リライタブル記録媒体に記録されているファイルを、前記リライタブル記録媒体に記録されている、前記記録媒体に組み合わせるためのファイルを示すマージ管理情報に従って組み合わせることにより、仮想パッケージを生成すること、および前記仮想パッケージの生成に用いるマージ管理情報が示すファイルを読み取り専用属性に変更することを行なうパッケージ管理手段と、前記仮想パッケージから、再生可能な複数のタイトルを検出し、そのうち1つをカレントタイトルとして選択する選択手段と、前記カレントタイトルに関連するアプリケーションを実行するアプリケーション実行手段と、前記仮想パッケージに含まれ、前記カレントタイトルにおいて示されるプレイリスト情報に従って、デジタルストリームの再生を制御する再生制御手段とを備え、前記仮想パッケージの生成に用いたマージ管理情報は、前記リライタブル記録媒体に記録されている第1のファイルの名前および当該第1のファイルへのファイルパスと、当該第1のファイルが前記生成した仮想パッケージにおいて利用されるときのファイルの名前および前記生成した仮想パッケージにおいて利用されるファイルへのファイルパスとを含むファイル名マッピング情報を含んでおり、実行された前記アプリケーションが、前記仮想パッケージの生成に用いたマージ管理情報とは異なり、かつ前記リライタブル記録媒体に記録されている新たなマージ管理情報を指定することで、前記パッケージ管理手段に前記生成した仮想パッケージの更新を要求すると、前記パッケージ管理手段は、前記仮想パッケージの更新を行う前に、前記指定した新たなマージ管理情報が示す、前記リライタブル記録媒体に記録されているファイルを読み取り専用属性に変更し、前記選択手段による新たなカレントタイトルの選択により、前記カレントタイトルにおいて示されるプレイリスト情報に従うデジタルストリームの再生を再生制御手段が停止した時点で、前記指定した新たなマージ管理情報に従って、前記記録媒体に記録されているファイルと、前記リライタブル記録媒体に記録されているファイルを組み合わせ、新たな仮想パッケージを生成するとともに、実行された前記アプリケーションが前記生成した仮想パッケージ上のファイルパスおよびファイルの名前を指定してアクセスを要求すると、前記仮想パッケージの生成に用いたマージ管理情報に含まれるファイル名マッピング情報に基づいて、当該指定した名前のファイルが第1のファイルである場合には、前記第1のファイルの格納場所により特定される前記リライタブル記録媒体に記録されている第1のファイルへアクセスさせることを特徴とする再生装置である。

【発明の効果】

【0010】

本発明は、上記の構成を備えることにより、実行された前記アプリケーションが前記パッケージ管理手段に対し、新たなマージ管理情報を指定することにより、前記マージ管理情報の更新を要求した後、前記パッケージ管理手段は、前記パッケージ情報の更新を行う前に、前記仮想パッケージの更新を行う前に、前記指定した新たなマージ管理情報が示す、前記リライタブル記録媒体に記録されているファイルを読み取り専用属性に変更するので、実行された前記アプリケーションが前記マージ管理情報の更新を要求した後からカレントタイトルの変更までの間に新たに指定したマージ管理情報から参照されるファイルの内容が変更されないことを保証するとともに、カレントタイトル変更によりデジタルストリームの再生が停止した時点で、仮想パッケージを更新すれば、たとえ再生の客体であるデジタルストリームやアプリケーションが更新により入れ代わったとしても、再生装置の異常動作を誘発することはない。

【0011】

10

20

30

40

50

よって、再生装置の安定動作を保障しつつ、記録媒体に記録されているファイルとリライタブル記録媒体に記録されているファイルとを動的に組み合わせ、生成した仮想パッケージを用いて、再生制御を実現することができるので、映画作品の表現の幅が広がる。

アプリケーション側から見れば仮想パッケージ上のファイルへのアクセスをする場合、そのファイルが記録媒体上に置かれているのかリライタブル記録媒体上に置かれているのかを認識する必要は無く、生成した仮想パッケージをアクセス先としてファイルが格納されている媒体として指定し、パッケージ管理手段が実際にファイルを格納する媒体（記録媒体／リライタブル記録媒体）へと切り換えるため、アプリケーションを作成する側から見れば、アクセスするべきファイルが記録媒体またはリライタブル記録媒体のいずれに格納されているのかを認識してアプリケーションプログラムを作成する必要がなくなるので、プログラムの作成の負担が軽減する。

10

【発明を実施するための最良の形態】

【0013】

（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、本発明に係る再生装置の実施行為のうち、使用行為についての形態を説明する。図1は、本発明に係る再生装置の、使用行為についての形態の一例を示す図である。図1において、本発明に係る再生装置は、再生装置200である。この再生装置200は例えば、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、映画作品を供給するという用途に供される。

20

【0014】

以上が本発明に係る再生装置の使用形態についての説明である。続いて本発明に係る再生装置が再生の対象としている、記録媒体について説明する。本発明に係る再生装置により、再生されるのは、この例では光記録媒体であるBD-ROM100である。図2は、BD-ROM100の内部構成を示す図である。

本図の第4段目にBD-ROM100を示し、第3段目にBD-ROM上のトラックを示す。本図のトラックは、BD-ROMの内周から外周にかけて螺旋状に形成されているトラックを、横方向に引き伸ばして描画している。このトラックは、リードイン領域と、ボリューム領域と、リードアウト領域とからなる。本図のボリューム領域は、物理層、ファイルシステム層、応用層というレイヤモデルをもつ。ディレクトリ構造を用いてBD-ROMの応用層フォーマット（アプリケーションフォーマット）を表現すると、図中の第1段目のようになる。この第1段目においてBD-ROMには、Rootディレクトリの下に、BDMVディレクトリがある。

30

【0015】

BDMVディレクトリの配下には、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、BOBJディレクトリと呼ばれる5つのサブディレクトリが存在し、BDMVディレクトリにはINDEX.BDMVというファイルが配置されている。

STREAMディレクトリには、いわばデジタルストリーム本体となるファイルを格納しているディレクトリであり、拡張子M2TSが付与されたファイル（00001.M2TS）が存在する。

【0016】

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）が存在する

40

。

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）が存在する

。

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）が存在する。

【0017】

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

以下、これらのファイルについて説明する。

<AVClip>

先ず初めに、拡張子.M2TSが付与されたファイルについて説明する。図3は、拡張子.M2TSが付与されたファイルがどのように構成されているかを模式的に示す図である。拡張子

50

.M2TSが付与されたファイル(00001.M2TS)は、AVClipを格納している。AVClipは(中段)、複数のビデオフレーム(ピクチャpj1,2,3)からなるビデオストリーム、複数のオーディオフレームからなるオーディオストリームを(上1段目)、PESパケット列に変換し(上2段目)、更にTSパケットに変換し(上3段目)、同じく字幕系のプレゼンテーショングラフィクスストリーム(下1段目のPGストリーム)及び対話系のインタラクティブグラフィクスストリーム(下1段目のIGストリーム)をTSパケットに変換して(下3段目)、これらを多重化することで構成される。

#### 【0018】

AVClipは、図3に示したように多重化がなされているストリームの他に、多重化がなされていないものもある。かかるAVClipは、SubClipと呼ばれ、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(TextSStream)等を構成するAVClipが存在する。

#### <Clip情報>

拡張子“CLPI”が付与されたファイル(00001.CLPI)は、AVClipのそれぞれに1対1に対応するClip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。

#### 【0019】

#### <PlayList情報>

拡張子“MPLS”が付与されたファイル(00001.MPLS)は、PlayList(PL)情報を格納したファイルである。PL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4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、PL情報は、『MainPath情報』、『PLMark情報』、『SubPath情報』から構成される。

#### 【0020】

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情報を複数配置することで、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5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。

#### 【0021】

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6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、逐一的に再生可能な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。

#### 【0022】

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7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

10

20

30

40

50

8 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。

#### 【 0 0 2 3 】

続いてSubPath情報について説明する。

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9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SubPlayItem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、SubClip時間軸とは同期して進行することになる。

#### 【 0 0 2 4 】

図10は、SubClip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SubClip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。

#### 【 0 0 2 5 】

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PL情報の特徴である。以上のClip情報及びPL情報は、“静的シナリオ”に分類される。

続いて“動的なシナリオ”について説明する。“動的に”というのは、再生装置200における状態変化やユーザからのキーイベントにより再生制御の中身がかわることをいう。BD-ROMでは、Javaアプリケーションと同様の記述により、この再生制御を記述することができる。つまりBD-ROMでは、Javaアプリケーションが、動的シナリオとしての役割を担うことになる。

#### 【 0 0 2 6 】

##### < Javaアプリケーション >

Javaアプリケーションについて説明する。Java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アプリケーションの構成である。

#### 【 0 0 2 7 】

このJava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アーカイブファイル(00001.JAR)である。以降、Javaアーカイブファイルについて、図11を参照しながら説明する。

##### < Javaアーカイブファイル >

Javaアーカイブファイル(図2の00001.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11(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ディレ

10

20

30

40

50

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アーカイブファイルは、これらをJava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アーカイブファイルのファイル名における"00001"という5桁の数値は、JavaアーカイブファイルのIDを示し、BD-Jオブジェクトは、この値を用いてJavaアーカイブファイルを参照する。本Java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。

【0028】

10

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。Java動作環境による動作モード(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 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。

【0029】

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。

20

xletプログラムのクラスファイルについて説明する。図11(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стан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、挙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(EventListener)と、BD-ROM再生装置200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アーカイブファイルについての説明である。

【0030】

30

続いて、拡張子BOBJを付したファイルについて説明する。拡張子BOBJを付したファイルは、BD-Jオブジェクトを格納したファイルである。BD-Jオブジェクトは、PlayList情報により定義されるAVClip列と、アプリケーションとの関連付けにより、タイトルを定義する情報である。図12は、BD-Jオブジェクトの内部構成を示す図である。本図に示すように、BD-Jオブジェクトは、“アプリケーション管理テーブル”と、“PlayList情報に対する参照値”とを示す。アプリケーション管理テーブルは、アプリケーションの識別子(アプリケーションID)とそのアプリケーションに属するJavaアーカイブファイルのIDを羅列することで、このタイトルを生存区間とするアプリケーションを示すテーブルである。つまり、一つのアプリケーションは一つ以上のJavaアーカイブファイルで構成される。

【0031】

40

“PlayList情報に対する参照値”は、このタイトルの開始時に、同時に再生すべきPlayList情報を示す。

以上が拡張子BOBJを付したファイルについての説明である。

続いてINDEX.BDMVについて説明する。

INDEX.BDMVは、BD-ROM全体に関する管理情報であり、映画作品のプロバイダを特定する識別子であるorganizationIDや、プロバイダが提供するBD-ROMのそれぞれに割り当てられた識別子であるdiscID等の情報を持ち、再生装置へのディスク挿入後に、INDEX.BDMVが最初に読み出されることで、再生装置においてディスクが一意に認識される。加えて、INDEX.BDMVにはBD-ROMにおいて再生可能となる複数のタイトルと、個々のタイトルを規定するBD-Jオブジェクトとを対応付けて示すテーブルが含まれる。以降、BD-ROMに記録され得るタイトルの類型について説明する。BD-ROMに記録され得るタイトルには、『FirstPlayT

50

title』、『Top\_menuTitle』、『Title#1,#2,#3』といったものがある。

【0032】

『FirstPlayTitle』は、BD-ROMが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。このFirstPlayTitleにより、映画作品の制作者や頒布者を表徴する動的商標を、ローディング時に再生するという慣習が実現されることになる。

『Top\_menuTitle』は、BD-ROMにおけるメニュー階層において、最上位に位置するメニューを再生させる、AVClip及びアプリケーションからなるTitleである。

【0033】

『Title#1,#2,#3』とは、一般的な映画作品にあたるTitleである。これら『FirstPlayTitle』、『Top\_menuTitle』、『Title#1,#2,#3』といったタイトルと、個々のBD-Jオブジェクトとの対応付けを示すのがINDEX.BDMVである。

図13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、『FirstPlayTitle情報』、『Top\_menuTitle情報』、『Title#1情報』、『Title#2情報』、『Title#3情報』といったTitle情報からなる。個々のTitle情報は、TitleのIDと、そのタイトルを規定するBD-Jオブジェクトとの対応付けを示す。以上のTitle情報により、タイトルを規定するBD-Jオブジェクトを特定することができ、このBD-Jオブジェクトから、連動すべきPlayList情報及びアプリケーションを導くことができる。以上がBD-ROMについての説明である。

【0034】

本発明に係る再生装置が対象としている記録媒体は、BD-ROMだけではない。再生装置に内蔵されているハードディスク(Local Storage)も再生の対象となる。このLocal Storageに記録されているデータについて説明する。

図14は、Local Storageにおけるディレクトリ構造を示す図である。本図のディレクトリ構造においてROOTディレクトリの配下には、「organization#1」というサブディレクトリがあり、その配下に、「disc#1」、「disc#2」というサブディレクトリがある。ディレクトリ「organization#1」とは、映画作品の特定のプロバイダに割り当てられたディレクトリである。「disc#1」、「disc#2」は、そのプロバイダが提供したBD-ROMのそれぞれに割り当てられたディレクトリである。これらのディレクトリ名には、各BD-ROMのINDEX.BDMVで示されるorganizationID、及びdiscIDの値が用いられる。

【0035】

特定のプロバイダに対応するディレクトリに、各BD-ROMに対応するディレクトリを設けることにより、各BD-ROMについてのダウンロードデータが個別に格納される。このサブディレクトリの配下に、BD-ROMに格納されていたのと同様、PlayList情報、Clip情報、AVClipが格納されており、この他にJavaアーカイブファイル、BD-J Object、マージ管理情報ファイルや、署名情報ファイルが存在する。

【0036】

ここでBD-ROM上のPlayList情報は、BD-ROM上のAVClipのみを参照しているのに対し、Local Storage上のPlayList情報(Virtual Packageとして新たに追加されたPlayList情報。具体的には図14に示すPlay List情報#2)は、BD-ROM上のAVClipと、Local Storage 18上のAVClipとを参照する情報を含む。

ここでLocal Storage上のPlayList情報が、PlayItem情報#1～PlayItem情報#4という4つのPlayItem情報から構成されているものとする。このうち、先頭のPlayItem情報#1は、BD-ROM上のClip情報を参照するものであり、残りの3つ(PlayItem情報#2～#4)が、Local Storage上のAVClip情報を参照するものである場合、このPlayList情報は、図15に示すように、BD-ROM上のAVClipと、Local Storage上のAVClipとから1つのストリームシーケンスを定義することができる。

【0037】

図15は、ローカルストレージに格納されたPL情報により、どのようなPlayList再生時間軸が定義されるかを示す図である。第1段目は、BD-ROMに記録されたAVClipにおける再

10

20

30

40

50

生時間軸を示し、第2段目は、ローカルストレージに格納されたPL情報により定義されるPlayList再生時間軸を示す。第3段目は、ローカルストレージに格納されたAVClip#2における再生時間軸を、第4段目は、ローカルストレージに格納されたAVClip#3における再生時間軸を、第5段目は、ローカルストレージに格納されたAVClip#4における再生時間軸を示す。

#### 【0038】

PlayList情報におけるPlayItemのうち、PlayItem情報#2、PlayItem情報#3、PlayItem情報#4が、AVClip#2、AVClip#3、AVClip#4を再生区間として指定するなら、Local Storage 18上のPlayList情報は、BD-ROM上のAVClip及びLocal Storage 18上のAVClipを1つのストリームシーケンスとして規定することができる。

以上のように、BD-ROM上のAVClip及びLocal Storage 18上のAVClipを1つのストリームシーケンスとして規定することができ、これをBD-ROM又はLocal Storage上のアプリケーションと組み合わせることにより、BD-ROM又はLocal Storageに記録されているアプリケーションと、AVClipとから、1つのタイトルを構成することができる。図16(a)に示すように、BD-ROMにAVClip#1が記録されており、Local StorageにAVClip#2～AVClip#4、及びアプリケーションが記録されている場合、これらのAVClip#1～AVClip#4、アプリケーションを、図16(b)に示すように、1つのタイトルとして扱うことができる。

#### 【0039】

続いて、マージ管理情報について説明する。このマージ管理情報は、Local Storageのdisc#1,#2ディレクトリに存在するファイルのうち、Virtual Packageを構成するものを一意に示しておく情報であり、このマージ管理情報は所定のファイルに格納されている(以後このマージ管理情報を格納するファイルをマージ管理情報ファイルと称す)。図17は、マージ管理情報ファイルの内部構成の一例を示す図である。マージ管理情報ファイルは、Virtual Packageを構成するLocal Storage上の各ファイルの格納位置情報からなる。各格納位置情報は、Virtual Packageとして各ファイルにアクセスするための“識別子”と、Local Storageにおけるそのファイルの格納位置を示す“ファイルパス”とからなる。

#### 【0040】

続いて、署名情報ファイルについて説明する。署名情報ファイルは、マージ管理情報ファイルに対するプロバイダの電子的な署名を示すファイルである。電子的な署名は、一般的に改ざん防止が必要な情報に対してハッシュ値を計算し、ハッシュ値を何らかの秘密鍵を用いて暗号化したものを用いる。

以上が記録媒体の説明である。

#### 【0041】

続いて本発明に係る再生装置の実施形態について説明する。図18は、再生装置のハードウェア構成を示す図である。再生装置は、BD-ROMドライブ1、Read Buffer 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、Read Buffer 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、Local Memory26、スイッチ27とから構成される。

#### 【0042】

先ず初めに、BD-ROM100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-ROM100に対するアクセスを実行する。

Read Buffer 2は、FIFOメモリであり、BD-ROM100又はLocal Storage18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。

#### 【0043】

10

20

30

40

50

デマルチプレクサ(De-MUX) 3 は、Read Buffer 2 からTSパケットを取り出して、取り出した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、CPU 2 4 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 1 0、オーディオデコーダ 1 6 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。

#### 【 0 0 4 4 】

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 $1920 \times 1080$ 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 $1/4$ (クオータという)、 $1/1$ (フルスクリーンという)のどちらかに変化させることである。かかるスケーリングを、BD-JモードにおいてCPU 2 4 からの指示に従い実行するので、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。

#### 【 0 0 4 5 】

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 $1920 \times 1080$ 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。

#### 【 0 0 4 6 】

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 1 0 は、DVDライクモードにおいてBD-ROM又はLocal Storage 1 8 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Interactive Graphicsプレーン 1 2 に書き込む。

#### 【 0 0 4 7 】

スイッチ 1 1 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 1 2 は、I-Graphicsデコーダ 1 0 によるデコードで得られた非圧縮グラフィクスが書き込まれる。またInteractive Graphicsプレーン 1 2 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 R G B のフルカラーで書き込まれる。

#### 【 0 0 4 8 】

合成部 1 3 は、Interactive Graphicsプレーン 1 2 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 1 0 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レイして、表示することができる。

#### 【 0 0 4 9 】

10

20

30

40

50

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、DVDライクモード（従来のDVDの再生のようなデジタルストリームを再生するモード）で動作しているときには、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。BD-Jモード（Javaアプリケーションとデジタルストリームとを連動して再生するモード）で動作しているときには、RGBのフルカラーを、Y,Cr,Cbに変換する。なお、Javaアプリケーションは、タイトルにバウンドされていてもいなくても良く、更にはディスクにバウンドしていてもいなくても良い。

#### 【0050】

10

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パケットを復号して、非圧縮形式のオーディオデータを出力する。

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コネクション等を確立する。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アプリケーションに行わせる。

#### 【0051】

20

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に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を利用した様々な処理を行うことができる。また、マージ管理情報ファイルを保持している。

#### 【0052】

Read Buffer19は、FIFOメモリであり、BD-ROM又はLocal Storage18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。

30

デマルチプレクサ(De-MUX)20は、Read Buffer19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をもつものをフォントゼネレータ9、P-Graphicsデコーダ6、オーディオデコーダ16に出力する。

#### 【0053】

以上のNetwork Device17～De-mux20により、Java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ROM21～スイッチ27)について説明する。

命令ROM21は、再生装置の制御を規定するソフトウェアを記憶している。

40

#### 【0054】

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

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

#### 【0055】

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

50

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。

#### 【0056】

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Play Item)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23(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。

#### 【0057】

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

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

#### 【0058】

Local Memory26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Local Memory26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 Storage18から読み出された各種データを、Read Buffer2、Read Buffer19、シナリオメモリ25、Local Memory26のどれかに選択的に投入するスイッチである。

#### 【0059】

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200におけるソフトウェア構造について説明する。

図19は、命令ROM21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、レイヤ構成に置き換えて描いた図である。本図に示すように、再生装置のレイヤ構成は、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 Environmentの第3階層からなる。

#### 【0060】

これらの階層のうち図18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18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と、ネットワークデバイス17とを含む。

#### 【0061】

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 System38及びPresentation Engine31の層

10

20

30

40

50

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。

【0062】

このうち図18に示したPSRセット23やシナリオメモリ25は、Playback Control Engine32内に存在することになる。

第3階層“Application Runtime Environment”は、以下のc1)c2)のスタック階層からなる。つまり、

c2)モジュールマネージャ33が存在する層

10

c1)DVDライクモジュール29a、Javaプラットフォーム29bが存在する層  
からなる。

【0063】

以降、このソフトウェア構造における各構成要素について説明する。

<DVDライクモジュール29a、Javaプラットフォーム29b>

DVDライクモジュール29aは、ナビゲーションコマンドを解読し、解読結果に基づきPlayback Control Engine32に対するファンクションコールを実行する。

【0064】

20

Javaプラットフォーム29bは、いわゆるJavaプラットフォームであり、

d1-1)Java仮想マシン30

d1-2)Java仮想マシンが動作するためのミドルウェア

を階層化した構成になっている。

【0065】

<Java仮想マシン30>

Java仮想マシン30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、メソッドを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えさせ、置換後のファンクションコールをPlayback Control Engine32に発行することでなされる。

30

【0066】

(Java仮想マシン30の内部構成)

ここでJava仮想マシン30の内部構成について説明する。図20は、Java仮想マシン30の内部構成を示す図である。本図に示すようにJava仮想マシン30は、CPU24と、ユーザクラスローダ52、メソッドエリア53、ワークメモリ54、スレッド55a,b···n、Javaスタック56a,b···nとから構成される。

【0067】

40

ユーザクラスローダ52は、BDJAディレクトリのJavaアーカイブファイルにおけるクラスファイルをLocal Memory26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モジュールマネージャ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Local Memory26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アーカイブファイルにおけるクラスファイルを、Local Memory26からワークメモリ54に読み出す。ファイルパスがFile System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。

50

## 【0068】

メソッドエリア53は、ユーザクラスローダ52によりLocal Memory26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。ワークメモリ54には、常駐型のレジデントアプリケーションやメソッドエリア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。

## 【0069】

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仮想マシン30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仮想マシン30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。

## 【0070】

Javaスタック56a,b・・・nは、スレッド55a,b・・・nと1対1の比率で存在しており、プログラムカウンタ(図中のPC)と、1つ以上のフレームとを内部に持つ。“プログラムカウンタ”は、インスタンスにおいて、現在どの部分が実行されているかを示す。“フレーム”はメソッドに対する1回のコールに対して割り当てられたスタック式の領域であり、その1回のコール時の引数が格納される“オペランドスタック”と、コールされたメソッドが用いる“ローカル変数スタック(図中のローカル変数)”とからなる。フレームは、コールが1回なされる度にJava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。

## 【0071】

以上がJava仮想マシンの内部構成である。このように構成されたJava仮想マシンは、イベントにより駆動される実行主体(イベントドリブンの実行主体)となる。以上がJava仮想マシンについての説明である。

## &lt; Presentation Engine 31 &gt;

Presentation Engine 31は、AV再生ファンクションを実行する。再生装置のAV再生ファンクションとは、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(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)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 31は、Read Buffer 2上に読み出されたAVClipのうち、所望の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、任意の時点を再生可能にすることができる。

## 【0072】

## &lt; Playback Control Engine 32 &gt;

再生制御エンジン(Playback Control Engine 32 (PCE))32は、プレイリストに対する

10

20

30

40

50

再生制御ファンクション(i)、PSRセット23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 Engine31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AVClip情報に従って行わせることをいう。これら機能(i)～(ii)は、DVDライクモジュール29a～Javaプラットフォーム29bからのファンクションコールに応じて実行する。

#### 【0073】

ここでPlayback Control Engine32による処理と、Java仮想マシンによる処理との同期について説明する。Playback Control Engine32は、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。再生しようとするAVClipが15分、30分という再生時間を有するなら、これらの時間、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仮想マシン30がサクセス応答を返す時間と、Playback Control Engine32が実際に処理を終える時間とのギャップである。Java仮想マシン30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 Engine32がAVClip及びPlayItemの再生を終了するのは、15分、30分経過後であるので、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、15分、30分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、巻戻しが行われると、この15分、30分という再生期間は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。そこでPlayback Control Engine32は、PlayItemの再生終了時、AVClipの再生終了時に、PlayItemやAVClipの再生が終了したことを示すイベントをアプリケーションに出力する。かかる出力により、Playback Control Engine32がPlayItemやAVClipの再生を終結した時点を、アプリケーションは知得することができる。

#### 【0074】

##### <モジュールマネージャ33>

モジュールマネージャ33は、INDEX.BDMVを読み出して、このINDEX.BDMVにおける複数のタイトル情報のうち、1つをカレントタイトル情報として選ぶ。そしてそのカレントタイトル情報に示されるBD-Jオブジェクトを読み出す。そしてこのBD-Jオブジェクトに記述されているPlayList情報に基づき、再生制御を行うよう、再生制御エンジンを制御する。またBD-Jオブジェクトに記述されるJavaアーカイブファイルを読み出して、実行するようJava仮想マシンを制御する。

#### 【0075】

ここでPlayList情報に基づくデジタルストリーム再生とアプリケーションの実行が終了した場合、又は、ユーザがメニュー呼出を行った場合、モジュールマネージャは、別のタイトルを定義するタイトル情報を読み出し、別のタイトル情報をカレントタイトル情報として選ぶ。デジタルストリームの再生や、ユーザによるメニュー呼出に応じて、別のタイトル情報をカレントのタイトル情報として選ぶ処理を、“タイトル切り換え”と呼ぶ。

#### 【0076】

かかる“タイトル切り換え”を繰り返すことにより、図21に示すような状態遷移を実現することができる。本図における楕円枠は、Title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。

#### 【0077】

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Top\_MenuTitleに戻るという処理を、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。上述したモジュールマネージャ33の制御により、かかる状態遷移は実現される。

以上が、Java仮想マシン、Presentation Engine31、Playback Control Engine32、モジュールマネージャ33の説明である。Playback Control Engine32に対する、Java

10

20

30

40

50

仮想マシンによる制御は、Virtual Packageを通じてなされる。Virtual Packageを通じた、Playback Control Engine 3 2に対する制御を実現するため、再生装置には、以下に示す構成要素(ネットワーク管理モジュール3 7、Virtual FileSystem部3 8)が存在する。以降これらの構成要素について説明する。

【0078】

<ネットワーク管理モジュール3 7>

ネットワーク管理モジュール3 7は、アプリケーションからのメソッド呼出に従って、Virtual Package構築に必要なデータを、映画作品のプロバイダが運営しているWWWサイトからダウンロードする。Virtual Package構築に必要なデータには、マージ管理情報ファイル、署名情報ファイル、及びBD-ROM上のファイルを置き換え、もしくは追加するファイル(PlayList情報、Clip情報、AVClip、Java アーカイブファイルなど)がある。Work Memory 5 4上のアプリケーションによるダウンロード要求がなされると、ネットワーク管理モジュール3 7は、ネットワークを通じて、Virtual Package構築に必要なデータをダウンロードし、Local Storage 1 8に書き込んでゆく。

10

【0079】

<Virtual FileSystem部3 8>

Virtual FileSystem部3 8は、第2階層に属する構成要素の1つであり、アプリケーションからのメソッド呼出に従い、Virtual Packageを構築する。Virtual Packageの構築処理には、Virtual Packageを構成する各AVClipの状態管理と、Virtual Package情報の生成処理とが含まれる。

20

【0080】

1.Virtual Package情報

Virtual Package情報は、BD-ROMにおけるボリューム管理情報を拡張した情報である。ここでボリューム管理情報は、ある記録媒体上に存在するディレクトリ・ファイル構造を規定する情報であり、ディレクトリについてのディレクトリ管理情報、ファイルについてのファイル管理情報とからなる。

【0081】

Virtual Package情報とは、BD-ROMのディレクトリ・ファイル構造を示すボリューム管理情報に、新たなファイル管理情報を追加することにより、BD-ROMにおけるディレクトリ・ファイル構造の拡張を図ったものである。ここでBDボリューム管理情報に追加されるファイル管理情報とは、Local Storage 1 8に存在するPlayList情報、Clip情報、AVClip、及びJavaアーカイブファイルについてのファイル管理情報であり、かかるファイル管理情報を追加したVirtual Package情報を作成し、Playback Control Engine 3 2に引き渡すことにより、Playback Control Engine 3 2は、Local Storage 1 8上にあるPlayList情報、Clip情報、AVClip、及びJavaアーカイブファイルがBD-ROMに存在するものとして認識することができる。図2 2は、Virtual FileSystem部3 8によるVirtual Package情報構築の一例を示す図である。本図の左上は、BD-ROMにおけるディレクトリ・ファイル構造であり、図2に示したものと同じである。本図の左下は、Local Storage 1 8におけるディレクトリ・ファイル構造であり、図1 4に示したものと同じである。このLocal Storage 1 8上のPlayList情報、Clip情報、AVClip、Javaアーカイブファイルについてのファイル管理情報を、BD-ROMのボリューム管理情報に追加する。

30

【0082】

具体的には、

i) Local Storage 1 8上のPlayList(00002.MPLS)についてのファイル管理情報を、BDボリューム管理情報におけるPLAYLISTディレクトリのディレクトリ管理情報に追加し、

40

ii) Local Storage 1 8上のClip情報#2、Clip情報#3、Clip情報#4(00002.CLPI, 00003.CLPI, 00004.CLPI)についてのファイル管理情報を、BDボリューム管理情報におけるCLIPINFディレクトリのディレクトリ管理情報に追加し、

50

iii) Local Storage 1 8 上のAVClip#2、AVClip#3、AVClip#4(00002.M2TS,00003.M2TS,0004.M2TS)についてのファイル管理情報を、BDボリューム管理情報におけるSTREAMディレクトリのディレクトリ管理情報に追加し、

iv) Local Storage 1 8 上のJavaアーカイブファイル(00002.JAR)についてのファイル管理情報を、BDボリューム管理情報におけるBDJAディレクトリのディレクトリ管理情報に追加する。

#### 【 0 0 8 3 】

こうすることにより、Virtual Package情報が得られる。この追加により得られたボリューム管理情報が、Virtual Package情報である。

こうして作成されたVirtual Package情報を、Playback Control Engine 3 2 に引き渡す。これにより、Playback Control Engine 3 2 は、Local Storage 1 8 上のPlayList情報、Clip情報、AVClip、Javaアーカイブファイルを、BD-ROM上のPlayList情報、Clip情報、AVClip、Javaアーカイブファイルと対等に扱うことができる。以上がVirtual Package情報生成についての説明である。

#### 【 0 0 8 4 】

続いて、このVirtual Package情報のアップデートのタイミングについて説明する。

図 2 1 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 2 3 ( a ) は、ディスク全体の時間軸を示す図であり、図 2 3 ( b ) は、この時間軸における構成を示す。図 2 3 ( 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、Top\_Menu 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と、Titleは、唯一のBD-J Objectから構成されるので、どれかの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。このTitleの再生区間の狭間、つまりあるTitleから、別のTitleへの切り換わる区間(Ti

#### 【 0 0 8 5 】

続いて、Javaアプリケーションによるマージ管理情報ファイル、署名情報ファイル及びコンテンツファイルのダウンロード手順について図 2 4 を用いて説明する。

まず、Javaアプリケーションは、現在のマージ管理情報ファイルをサーバに送信する(ステップ S 1 1 )。これによりダウンロードを要求する。サーバからデータを受信したか否かを判断する(ステップ S 1 2 )。ダウンロードすると、Javaアプリケーションは、対応するディスクディレクトリに新規ディレクトリを作成して、ダウンロードしたマージ管理情報ファイルと署名情報ファイルをその新規ディレクトリに書き込む(ステップ S 1 3 )。なお、ディスクディレクトリにある既存のマージ管理情報ファイル、署名情報ファイルと新たにダウンロードするマージ管理情報ファイル、署名情報ファイルのファイル名が重ならなければ、新規ディレクトリを作成せずに、既存ディレクトリ(disc#1ディレクトリ)の直下にダウンロードしてもよい。続いて、ダウンロードしたAVClip、Clip情報、PlayList情報、Javaアーカイブファイルを対応するディレクトリに書き込む(ステップ S 1 4 )。新規のマージ管理情報ファイルのファイルパスと署名情報ファイルのファイルパスを引数にして、request Updateメソッドを呼び出す(ステップ S 1 5 )。返り値がFalseか否かを判断し(ステップ S 1 6 )、Falseなら処理は終了する。Falseでないなら、更新されたVirtual Package情報を用いて処理を実行する(ステップ S 1 7 )。

#### 【 0 0 8 6 】

なお、ダウンロードを要求する際に、Javaアプリケーションは、現在のマージ管理情報ファイルをサーバに送信したが、マージ管理情報ファイルのIDのみを送信するとしても良い。

続いて、request UpdateのPreparingの処理手順について図 2 5 を用いて説明する。

10

20

30

40

50

まず、メソッド呼出時の引数となるファイルパスを用いて、新たなマージ管理情報ファイル、及び新たな署名情報ファイルを読み出す（ステップS21）。次に、新たなマージ管理情報ファイルが改ざんされているか否かをチェックするための署名を検証する（ステップS22）。異常があれば例外終了する。正常であれば、呼出元アプリの権限をチェックする（ステップS23）。権限がなければ例外終了する。権限があれば、新たなマージ管理情報ファイルにより指定されているファイルがローカルストレージに実在するか否かを判断する（ステップS24）。実在しなければ例外終了する。実在すれば、新たなマージ管理情報ファイル、新たな署名情報ファイル及び新たなマージ管理情報ファイルが参照しているローカルストレージ上の全ファイルを読み取り専用属性に変更する（ステップS25）。

10

#### 【0087】

続いて図26は、Virtual Packageの更新処理を示すフローチャートである。まず、挿入されているBD-ROMに対応するディスクディレクトリのマージ管理情報ファイル及び署名情報ファイルを、Javaアプリケーションがrequest Updateメソッド呼び出し時に引数のファイルパスで指定した新たなマージ管理情報ファイルと署名情報ファイルに置き換える（ステップS31）。そして、Local Storage18においてマージ管理情報ファイルで指定されているPlaylist情報のファイル管理情報を、BDボリューム管理情報におけるPLAYLISTディレクトリのディレクトリ管理情報に追加する（ステップS32）。その後、ステップS33～ステップS37のループ処理を実行する。このループ処理は、Local Storage18に存在するそれぞれのClip情報について、ステップS34～ステップS36の処理を繰り返すものである。ここで、ループ処理の対象となる1つのClip情報をClip情報xとする。本ループ処理において、Clip情報xに対応するAVClipを特定する（ステップS34）。Local Storage18においてマージ管理情報ファイルで指定されているClip情報のファイル管理情報を、BDボリューム管理情報におけるCLPINFディレクトリのディレクトリ管理情報に追加する（ステップS35）。Local Storage18においてマージ管理情報ファイルで指定されているAVClipxのファイル管理情報を、BDボリューム管理情報におけるSTREAMディレクトリのディレクトリ管理情報に追加する（ステップS36）。以上の処理を全てのClip情報及びAVClipについて繰り返せば、Clip情報、AVClipについてのファイル管理情報が、BDボリューム管理情報に追加されることになる。こうした追加により得られたBDボリューム管理情報がVirtual Package情報になる。このVirtual Package情報をVirtual Package呼出の呼出元のアプリケーションに引き渡し（ステップS38）、処理を終える。

20

#### 【0088】

続いて図27は、モジュールマネージャ33の処理手順を示すフローチャートである。まず、FirstPlayTitleをカレントTitleに選ぶ（ステップS41）。カレントTitleに対応するBD-JオブジェクトをカレントBD-Jオブジェクトとして特定する（ステップS42）。カレントBD-Jオブジェクトに記載されているPlaylist情報に基づくPL再生をPCEに実行させる（ステップS43）。カレントBD-JオブジェクトのAMTにおいて、カレントTitleを生存区間としているJavaアプリをJavaプラットフォームに実行させる（ステップS44）。カレントBD-JオブジェクトのAMTにおいて、カレントTitleを生存区間としていないJavaアプリをJavaプラットフォームに終了させる（ステップS45）。次に、Playlist情報に基づくPL再生が完了したか否かを判断する（ステップS46）。完了していれば、次のTitleを特定し（ステップS47）、次のTitleをカレントTitleに選ぶ（ステップS48）。完了していなければ、Titleコールが発生したか否かを判断する（ステップS49）。発生していれば、ステップS47に移行する。発生していなければ、Titleジャンプが発生したか否かを判断する（ステップS50）。発生していれば、ステップS47に移行する。発生していなければ、Titleのメインとなるアプリが終了したか否かを判断する（ステップS51）。終了していれば、ステップS47に移行する。終了していなければ、ステップS46に移行する。

30

#### 【0089】

続いて図28は、Playback Control Engine32による再生手順を示すフローチャート

40

50

である。本フローチャートは、PlayList情報における先頭のPlayItem情報をPlayItem iとして(ステップS 6 1)、ステップS 6 2～ステップS 6 8のループ処理を実行するものである。ステップS 6 2～ステップS 6 8のループ処理における制御変数は、変数iであり、ステップS 6 2～ステップS 6 6の処理を実行して、その後、制御変数iをインクリメントするという処理を(ステップS 6 8)、変数iが、PlayItem数(Number Of PlayItem)を越えるまで実行するものである(ステップS 6 7)。

#### 【0090】

このステップS 6 2～ステップS 6 6の処理について説明する。これらの処理は、PlayItem情報iのClip\_information\_file\_nameに記述されているAVClipを、AVClipjとして(ステップS 6 2)、再生の対象とする。AVClipjのうち、PlayItem.In\_timeからPlayItem.Out\_timeまでの部分を再生するよう、ドライブ装置及びデコーダに指示する(ステップS 6 3)。PlayItem情報iをSync\_PlayItem\_idに指定しているSubPlayItemkが存在するか否かを判定する(ステップS 6 4)。もし存在しなければ、そのままステップS 6 7に移行する。もし存在すれば、SubPlayItemkのClip\_information\_file\_nameに記述されているAVClipをAVCliphにする(ステップS 6 5)。AVCliphのうち、Sync\_Start PTS\_of\_PlayItemから、Out\_timeまでを再生するよう、ドライブ装置及びデコーダに指示して(ステップS 6 6)、ステップS 6 7に移行する。

#### 【0091】

以上の処理を、PlayList情報を構成する全てのPlayItem情報について繰り返すことにより、PlayList情報にて規定されるストリームシーケンスの再生がなされる。

続いて図29は、Title Changeにおいて、Virtual Package情報アップデートがどのように行われるかを示す図である。

第一段は、タイトルの再生区間を示す。第二段は、Javaアプリケーションの再生区間を示す。第三段は、ストリームを示す。第四段は、バーチャルファイルシステムの状態を示す。

#### 【0092】

Javaアプリケーションが、バーチャルファイルシステムに更新要求を送信すると、バーチャルファイルシステムは、Preparing状態になる。Preparing状態では図25で示した処理が行われる。処理が終了するとタイトル切り換えが発生するまで待つ。これがprepared状態である。タイトル切り換えが起こると、図26で示した処理を実行し、更新を行う。再びstableの状態になれば、例えば、Top MenuTitleから再度Title#1が選択された場合には、更新後の内容でTitle#1は再生される。

#### 【0093】

ここで、タイトルの切り換えは、ストリーム列の再生終了や、ユーザによるメニュー呼び出しにて、別のタイトルをモジュールマネージャがカレントタイトルとして選んだことにより発生する。

続いて、上記処理を、図30から図33を用いて模式的に説明する。

図30は、Javaアプリケーションが、現在のマージ管理情報ファイルをサーバに送信する様子を示した図である。ここで、ROOT以下に示されるファイルは、ローカルストレージ上のファイルであり、BDMV以下に示されるファイルは、Virtual Package上のファイルである。

#### 【0094】

続いて図31は、Javaアプリケーションが、コンテンツファイルと新しいマージ管理情報ファイル、署名情報ファイルをダウンロードする様子を示した図である。00012.clpiと00012.M2TSは、ダウンロードしたコンテンツファイルであり、newMFディレクトリに格納されているマージ管理情報ファイル及び署名情報ファイルは、ダウンロードした新しいマージ管理情報ファイル、署名情報ファイルである。

#### 【0095】

続いて図32は、Javaアプリケーションが、バーチャルファイルシステムに対し、マージ管理情報ファイルと署名情報ファイルを新しいマージ管理情報ファイル、署名情報ファ

10

20

30

40

50

イルに更新するように要求する様子を示した図である。図32に示すように更新要求は、新しいマージ管理情報ファイルと新しい署名情報ファイルをファイルパスで指定することにより行う。

【0096】

続いて図33は、マージ管理情報ファイル及び署名情報ファイルの置き換え、及びマッピングの様子を示した図である。図33の左側は、古いマージ管理情報ファイル、署名情報ファイルが、Updating中に、新しいマージ管理情報ファイル、署名情報ファイルに置き換えられる様子を示している。また、図33の右側は、タイトル切り換え後におけるマッピングの様子を示している。

【0097】

なお、古いマージ管理情報ファイルから参照されていて、新しいマージ管理情報ファイルから参照されていないファイルは、読み取り専用属性が解除され、Javaアプリケーションから読み書き可能になる。

マージ管理情報ファイルには、ローカルストレージ上の追加コンテンツの位置を指示する情報が含まれている。追加コンテンツの位置を指示する情報としては、追加コンテンツID、追加コンテンツが格納されているディレクトリまでのファイルパス、または追加コンテンツの各ファイルへのファイルパスなどが考えられる。さらに、それらのファイルをVirtual Package上のファイルとしてマッピングする時に、Virtual Package上では別名ファイルとしてアクセスできるように、マージ管理情報ファイルにファイル名マッピング情報を記載するとしても良い。その場合は、ファイル名マッピング情報には、ローカルストレージ上の各ファイル名（ファイルパスを含む）と対応するVirtual Package上のファイル名（ファイルパスを含む）が含まれる。

【0098】

マージ管理情報ファイルにファイル名マッピング情報が記載されていた場合、Virtual Packageは、BD-ROM上のファイル構成にファイル名マッピング情報に記載されているVirtual Package上のファイル名を加えたファイルを構成要素とする仮想的なパッケージメディアとして、バーチャルファイルシステム部により構築される。JavaアプリケーションはVirtual Package上のファイルへアクセスする場合、BD-ROM上のファイルまたはローカルストレージ上のファイルといったような指定をするのではなく、Virtual Package上のファイルとして指定する。バーチャルファイルシステム部は、Virtual Package上のファイルに対するJavaアプリケーションからのアクセスを検出すると、ファイル名マッピング情報を元に、ファイルのアクセス先をローカルストレージもしくはBD-ROMへ切り換える。アプリケーションからのアクセス要求があったファイルがファイル名マッピング情報に記載されている場合にはアクセス要求のあったVirtual Package上のファイルに対応するローカルストレージ上のファイルへアクセス先を変更する。ファイル名マッピング情報に記載されていないファイルへのアクセスの場合は、BD-ROM上のファイルへアクセスする。

【0099】

即ちJavaアプリケーション側から見ればVirtual Package上のファイルへのアクセスをする場合、そのファイルがBD-ROM上に置かれているのかローカルストレージ上に置かれているのかを認識する必要は無く、Virtual Packageをアクセス先としてファイルが格納されている媒体として指定し、バーチャルファイルシステム部が実際にファイルを格納する媒体（BD-ROM / ローカルストレージ）へと切り換えるため、Javaアプリケーションを作成する側から見れば、アクセスするべきファイルがBD-ROMまたはローカルストレージのいずれに格納されているのかを認識してアプリケーションプログラムを作成する必要がなくなるので、プログラムの作成の負担が軽減する。

【0100】

以上のように本実施形態によれば、タイトル切り換え時に、Virtual Packageの更新を行うので、再生の客体が入れ代わったとしても、再生装置の異常動作を誘発することはない。

本実施形態は、タイトル呼出時における改良に関する。タイトル呼出とは、カレントタイトルの再生を一時的に停止(suspend)して、呼出先のタイトルの再生を行い、そのタイトルの実行終了後、元のタイトルの再生を再開(resume)させるものである。

#### 【0101】

かかる再生再開を前提にしているので、タイトル呼出時にあたってPlayback Control Engineは、PSRに格納されている、再生制御のためのシステムパラメータをバックアップPSRに退避し、呼出先のタイトルの再生後、そのバックアップPSRに退避したパラメータをPSRに復帰するとの処理を行う。

以下は、PSRに格納されるシステムパラメータの一覧である。PSR(0)からPSR(12)までは、再生状態を示すシステムパラメータを格納する。PSR(13)からPSR(19)までは、プリファレンスとしてプレーヤに設定されるシステムパラメータを格納している。PSR(20)からPSR(32)までは、バックアップ用のPSRである。

#### 【0102】

PSR(0)	:	IGストリーム番号	
PSR(1)	:	音声ストリーム番号	
PSR(2)	:	PGストリーム/テキスト字幕ストリーム番号	
PSR(3)	:	アングル番号	
PSR(4)	:	現在再生中のタイトル番号	
PSR(5)	:	現在再生中のチャプター番号	10
PSR(6)	:	現在再生中のプレイリスト識別子	
PSR(7)	:	現在再生中のプレイアイテム識別子	
PSR(8)	:	再生時刻情報	
PSR(9)	:	ナビゲーションタイマー	
PSR(10)	:	選択キー情報	
PSR(11)	:	IGストリームにおける現在のページ識別子	
PSR(12)	:	PGストリーム、テキスト字幕ストリームにおけるユーザースタ	
イル識別子			
PSR(13)	:	パレンタルレベル	
PSR(14)	:	字幕サポート情報	
PSR(15)	:	プレーヤ設定値(オーディオ)	20
PSR(16)	:	音声ストリーム用言語コード	
PSR(17)	:	PGストリーム、テキスト字幕ストリーム用の言語コード	
PSR(18)	:	メニュー用言語コード	
PSR(19)	:	プレーヤのバージョン情報	
PSR(20)	:	PSR(0)のバックアップ	
PSR(21)	:	PSR(1)のバックアップ	
PSR(22)	:	PSR(2)のバックアップ	
PSR(23)	:	PSR(3)のバックアップ	
PSR(24)	:	PSR(4)のバックアップ	
PSR(25)	:	PSR(5)のバックアップ	30
PSR(26)	:	PSR(6)のバックアップ	
PSR(27)	:	PSR(7)のバックアップ	
PSR(28)	:	PSR(8)のバックアップ	
PSR(29)	:	PSR(9)のバックアップ	
PSR(30)	:	PSR(10)のバックアップ	
PSR(31)	:	PSR(11)のバックアップ	
PSR(32)	:	PSR(12)のバックアップ	

ここで、かかるタイトル呼出時にVirtual Package情報のアップデートがあると、タイトル呼出の前後で、Virtual Package情報の入れ代わりが生じる。

## 【0103】

呼出元のタイトルに復帰したときはVirtual Package情報がすでに変更されているため、バックアップしておいた値を使って呼出元タイトルを再生しようとすると、以前と同じ値ではエラーが起こる可能性がある。そのため、JavaアプリケーションがVirtual Package情報のアップデート要求を行うときは、バックアップPSRをクリアすることで、この問題を回避する。ただし、Virtual Package情報におけるマージ管理情報ファイルの内容によって、変更の影響を受けないケースも有りうるため、システムパラメータの値をクリアするかはJavaアプリケーションに任せてもよい。

## 【0104】

図34(a)は、タイトル呼出にあたって、カレントタイトルの再生を一時的に停止(suspend)する際のPlayback Control Engineの処理手順を示すフローチャートである。図34(b)は、呼出先のタイトルの再生終了後、元のタイトルの再生を再開(resume)する際のPlayback Control Engineの処理手順を示すフローチャートである。

カレントタイトルの再生を一時的に停止(suspend)する際には、PSR(0)からPSR(1)2)をPSR(20)からPSR(32)に退避する(ステップS71)。

## 【0105】

呼出先のタイトルの再生終了後、元のタイトルの再生を再開(resume)する際には、まず、Virtual Package情報の更新があるか否かを判断する(ステップS81)。更新がない場合は、PSR(20)からPSR(32)をPSR(0)からPSR(12)に復帰する(ステップS83)。更新があると、PSR(20)からPSR(32)を初期化し(ステップS82)、ステップS83の処理を行う。

## 【0106】

以上のように本実施形態によれば、タイトル呼出時において、Virtual Package情報のアップデートがあった際、バックアップPSRの内容を一旦初期化するので、その呼出前のPSRに基づき、Playback Control Engineが誤った再生処理をする恐れはない。Playback Control Engineの安定動作の実現が可能になる。

なおVirtual Package情報のアップデートを伴うときに、バックアップPSRの値をクリアするかどうかをJavaアプリケーションに任せず、システム側で強制的にシステムパラメータの値をクリアしてもよい。

## 【0107】

## (第3実施形態)

本実施形態は、マージ管理情報のバージョン管理と再生装置内のレジデントアプリケーションからマージ対象の追加コンテンツを指定する方法について述べる。図35は、本実施形態に係るマージ管理情報ファイルの具体的な内容の一例を示す図である。第1実施形態におけるマージ管理情報ファイル(より詳しくはマージ管理情報ファイルに格納されるマージ管理情報)の更新は、古いマージ管理情報へ上書きすることにより更新がなされていたが、この場合、古いマージ管理情報はマージ管理情報ファイルの中からは消去されてしまう。本実施形態では、同じディスクIDにおいても古いマージ管理情報への上書きは行わず、新しいマージ管理情報がマージ情報管理ファイルへ次々と追加される形となる。Virtual Packageの構築をキャンセルして、オリジナルのBD-ROMに変えた場合も、その状態に対応する情報をマージ管理情報ファイルに保持する。オリジナルのBD-ROMに変更した場合は、マージ対象のディレクトリの欄は空欄もしくは、オリジナルBD-ROMであることを示す文字列等が記載される。

## 【0108】

このようにマージ管理情報ファイルを更新する際に、古いマージ管理情報に上書きしないようにすると、過去のマージ管理情報(の履歴)をマージ情報管理ファイルに保持しておくことが可能となり、古いバージョンのVirtual Packageを構築したい場合に、過去のマージ管理情報を参照して、古いバージョンのVirtual Packageを構築することが出来る。また、Javaアプリケーションだけでなく、再生装置内のレジデントアプリケーションからも、マージ管理情報ファイル(詳しくはマージ管理情報ファイルに格納された過去のマ

10

20

30

40

50

ージ管理情報)を参照して、ユーザが過去に作成したVirtual Packageを構築することができる。

【0109】

過去のマージ管理情報をレジデントアプリが用いるその他の例として、ユーザが過去に作成した追加コンテンツ一覧を表示し、ユーザが不要と思った追加コンテンツをレジデントアプリから削除することも考えられる。マージ管理情報ファイルから追加コンテンツが保存されているディレクトリを判別することが可能であるため、追加コンテンツを保存したJavaアプリケーションではないレジデントアプリケーションからも追加コンテンツのディレクトリを探し出し削除することができる。

【0110】

図36は、本実施形態に係るJavaアプリケーションがVirtual Packageの更新を要求している様子を示す図である。第1実施形態と本図との違いは、対象となるディスクIDのマージ管理情報が存在する場合でも上書きを行わず、次々と追加されていく点である。そのときに、マージ管理情報ファイル内のどれが最も新しいマージ管理情報かわかるように、JavaアプリケーションがVirtual Package更新要求を行ったときの日時情報も付加しておく。なお、更新日時情報は日付に限らず、単なる連番等の情報でもよい。

【0111】

図37は、レジデントアプリケーションがマージ管理情報ファイルを用いて、ユーザに対して追加コンテンツ一覧表示を行う一例である。ここで表示している追加コンテンツ一覧は、図35で用いたマージ管理情報ファイルの内容を元に、レジデントアプリケーションが追加コンテンツ一覧表示を行っている。レジデントアプリケーションからユーザに対してマージ管理情報ファイルを用いて追加コンテンツ一覧表示を行う際、追加コンテンツがどのような内容か、ユーザが理解できる情報まで表示する方が望ましい。図37では追加コンテンツ名として、追加コンテンツの内容をユーザにわかるように表示している。追加されたコンテンツの概要をレジデントアプリケーションがユーザに対して表示できるようにするため、図36ではマージ管理情報ファイル内のマージ管理情報に日付情報のみを加えているが、追加コンテンツの概要などもマージ管理情報に付加してもよい。その場合、JavaアプリケーションがVirtual Package更新要求を行う際に、追加コンテンツIDと共に追加コンテンツ概要を入力に与える。入力する追加コンテンツ概要は直接文字列を入力するだけでなく、追加コンテンツ概要が記載されたファイルへのパスを指定してもよい。マージ管理情報には、更新日時に加え、追加コンテンツ概要も保存される。レジデントアプリケーションは追加コンテンツ一覧表示を行うときに、日時情報に加え、追加コンテンツの概要を表示することが可能になる。

【0112】

なお、Javaアプリケーションが追加コンテンツ概要を指定するのではなく、追加コンテンツ自体に追加コンテンツの内容を示すメタ情報を付加しておき、レジデントアプリケーションがそのメタ情報を読み込んで、それを元に追加コンテンツ概要を表示してもよい。

また、図37に示される追加日時は、追加コンテンツが最初にBD-ROMとマージされた日時を示している。この情報もマージ管理情報から読み取ることが可能である。

【0113】

なお、追加コンテンツが最初にマージされた日時は、マージ管理情報とは別に保持していても良い。また、追加コンテンツを保存するディレクトリの作成日時から判断することも可能である。追加コンテンツ一覧表示の中にある選択ボタンが押されると、レジデントアプリはマージ管理情報ファイル内の該当するマージ管理情報として選択された追加コンテンツへのディレクトリパスとディスクID及び、選択された日時を書き込む。つまり、最後に選択された追加コンテンツが、最も新しいマージ管理情報となる。オリジナルのBD-ROMが選択された場合、マージ管理情報ファイルにおいてマージ管理情報のマージ対象のディレクトリにはオリジナルBD-ROMを示す値、もしくは空欄が挿入される。追加コンテンツ一覧表示の中にある削除ボタンが押されると、マージ管理情報ファイルから削除の対象となる追加コンテンツIDのディレクトリを読み取り、そのディレクトリの削除を行う。また

10

20

30

40

50

、マージ管理情報ファイルからも削除対象となる追加コンテンツIDに対応したマージ管理情報を削除する。

【0114】

図38は、本実施形態に係るBD-ROM挿入から再生までの処理の流れを示すフローチャートである。まず、挿入されたBD-ROMのディスクIDを確認し(ステップS91)、マージ管理情報ファイルの内容(より詳しくはマージ管理情報)を読み取る(ステップS92)。挿入されたBD-ROMのディスクIDに対応するマージ管理情報が存在するか否かを判断する(ステップS93)。対応するマージ管理情報がない場合は、オリジナルのBD-ROMで再生を行う(ステップS94)。対応するマージ管理情報が存在する場合は、最新のマージ管理情報を用いてVirtual Packageを構築する(ステップS95)。構築の際に、エラー検出があるか否かを判断する(ステップS96)。Virtual Package構築時にエラーが検出された場合は、Virtual Packageの構築の際に用いた最新のマージ管理情報より古いマージ管理情報であって、挿入されたBD-ROMのディスクIDに対応するマージ管理情報が存在するか否かを判断する(ステップS97)。最新のマージ管理情報より古いマージ管理情報が存在すれば、最新のマージ管理情報より一つ古いバージョンのマージ管理情報で、Virtual Packageの構築を行う(ステップS98)。最新のマージ管理情報より古いマージ管理情報が存在しなければ、オリジナルのBD-ROMのみを用いて再生を行う。ステップS96でエラーが検出されなければ、構築されたVirtual Packageで再生を行う(ステップS99)。検出されるエラーの例としては、最新のマージ管理情報の誤りや、プレイリストが参照している、存在するはずのストリームが存在しない場合などが考えられる。

10

20

【0115】

以上のように本実施形態によれば、マージ管理情報ファイルに過去のマージ管理情報を保持しておくことで、再生装置内のレジデントアプリから過去のマージ管理情報ファイルの内容の履歴を参照し、古いバージョンのマージ管理情報を用いてVirtual Packageを構築することができる。また、Virtual Package構築時にエラーが発生したときに、代替策として、古いバージョンのVirtual Packageを構築することで、エラーを回避することができる。

【0116】

(第4実施形態)

本実施形態は、JavaアプリケーションによるVirtual Package更新要求時に、Virtual Package有効期間を指定し、有効期間内のみVirtual Packageで再生を行う方法について述べる。

30

【0117】

図39は、有効区間を指定してVirtual Package更新要求を行っている様子を示す図である。Javaアプリケーションはマージ対象の追加コンテンツのIDを指定するとともに、Virtual Package構築を行う有効区間も同時に指定する。例えば、ディスクを取り出すまではVirtual Packageとして再生し、ディスクを取り出した後、再び挿入したときはオリジナルのBD-ROMで再生したい場合、JavaアプリケーションからのVirtual Package更新要求時に、Virtual Packageがディスク取り出しまで有効であることを示す値を引数に指定する。

40

【0118】

図40は、本実施形態に係るBD-ROM挿入、または再起動から再生までの処理の流れを示すフローチャートである。まず、挿入されたBD-ROMのディスクIDを確認し(ステップS101)、マージ管理情報ファイルの内容(より詳しくはマージ管理情報)を読み取る(ステップS102)。挿入されたBD-ROMのディスクIDに対応するマージ管理情報が存在するか否かを判断する(ステップS103)。対応するマージ管理情報がない場合は、オリジナルのBD-ROMで再生を行う(ステップS104)。対応するマージ管理情報が存在する場合、そのマージ管理情報が有効区間内か否かを判断する(ステップS105)。すでに有効区間が過ぎているのであれば、そのマージ管理情報をマージ管理情報ファイルから削除し(ステップS106)、オリジナルのBD-ROMで再生する。有効区間の範囲内であれば、

50

そのマージ管理情報を用いてVirtual Packageを構築し(ステップS107)、Virtual Packageで再生を開始する(ステップS108)。

#### 【0119】

なお、本実施形態の応用として、JavaモードのみVirtual Packageを構成するといったパターンも考えられる。JavaアプリケーションからVirtual Package構築要求時にJavaモードのみの指定がある場合、DVDライクモードからJavaモードへの遷移が行われたときに、その時点でVirtual Packageを構築し、その後Javaモードへ移行する。逆にJavaモードからDVDライクモードへ遷移するときは、Virtual Packageを解除し、オリジナルのBD-ROMに戻した後、DVDライクモードへ移行する。

#### 【0120】

以上のように本実施形態によれば、Virtual Packageによる再生の区間を指定することができ、一度切りでよいVirtual Packageの再生や、期間限定のVirtual Package等を構築することができる。

なお、本実施形態では、Virtual Packageの更新要求時に有効期間を指定したが、BD-ROM挿入においてVirtual Package構築要求の際に、有効期間を指定しても良い。

#### 【0121】

##### (第5実施形態)

本実施形態では、第一実施形態で説明した図25におけるステップS23の呼出元アプリの権限について詳細に説明する。具体的には、不正なJavaアプリケーションからのVirtual Package更新要求を拒否する方法について述べる。

#### 【0122】

図41は、パーミッションリクエストファイルによるVirtual Packageの更新要求を制限している様子を示す図である。Virtual Packageの更新はJavaアプリケーションからのVirtual Package更新要求に基づいて行われるが、不正なJavaアプリケーションからVirtual Packageの更新が行われると、視聴制限の変更や一定の条件下でしか見られない特典映像の再生などをできるようにするなど、ディスクの内容を不正に更新されてしまう恐れがある。そのため、Virtual Packageの更新要求は、要求することを許可されたことを示す情報(Virtual Package更新パーミッション)を保持するJavaアプリケーションからのみ行うことができるようになる。Virtual Package更新パーミッションを保持しているかどうかは、要求を出したJavaアプリケーションに対応するパーミッションリクエストファイルの内容を確認することで判断する。具体的には、クラスローダが、パーミッションリクエストファイルの内容に従い、Javaアプリの機能を制限する。例えば、パーミッションリクエストファイルに記載されているVirtual Package更新許可判断エレメントの属性値がtrueの場合は更新要求を受け入れ、更新許可判断エレメントの属性値がfalseとなっていれば、更新要求を拒否する。

#### 【0123】

図42は、ローカルストレージ内のマージの対象となるディレクトリに対するアクセス制限を示す図である。不正なJavaアプリケーションにマージの対象となるディレクトリの内容を変更されてしまうと、たとえVirtual Package更新要求を制限したとしても、Virtual Packageの内容を不正なものに変えることが可能になる。ローカルストレージに対する読み書きアクセスも同様にパーミッションリクエストファイルの内容に基づいてアクセスを許可する。例えば、パーミッションリクエストファイルに記載されているファイルアクセス許可判断エレメントのwrite属性値、read属性値がどちらもtrueとなっている場合は、ファイルの読み書きが可能となり、ダウンロードコンテンツ等を書き込んだり、保存したファイルをさらに編集したりすることができる。ただし、ファイルアクセス許可判断エレメントのwrite属性値、read属性値のどちらか、または両方がfalseとなっているパーミッションリクエストファイルを持つJavaアプリケーション、もしくはパーミッションリクエストファイル自体を保持していないJavaアプリケーションはファイルアクセスに対して制限を受けることになる。

#### 【0124】

10

20

30

40

50

以上のように本実施形態によれば、不正なJavaアプリケーションによるVirtual Package更新要求、及びローカルストレージ内のディレクトリ内容の変更を防ぐことが出来る。

続いて、パーミッションリクエストファイルによるVirtual Packageの更新の制限について、具体例を挙げて説明する。例えば、ローカルストレージに映画作品の特定のプロバイダに割り当てられたディレクトリが複数ある場合を考える。具体的には、ローカルストレージに、AスタジオのコンテンツとBスタジオのコンテンツとC社のコンテンツとが格納されているとする。また、BD-ROMにもAスタジオのコンテンツとBスタジオのコンテンツとC社のコンテンツとが格納されているとする。ここで、C社はディスクマガジンの提供元とする。ローカルストレージ内のコンテンツをBD-ROMにマージして使用する際に、例えば、A社のコンテンツにB社のコンテンツをマージさせてしまうと問題である。そこで、ディスクマガジンの提供者であるC社のみがパーミッションtrueとし、様々なサービスを付加できるようにする。A社及びB社は、それぞれ自社のコンテンツのみマージできるようにする。

#### 【0125】

##### (第6実施形態)

本実施形態では、タイトルにまたがって動作するJavaアプリケーションが存在した場合において、タイトル切り換え時にVirtual Packageを更新する方法について述べる。

図43は、タイトル内だけで生存するJavaアプリケーションとタイトルにまたがって生存するJavaアプリケーションのライフサイクルを示す図である。Javaアプリケーションの生存期間はアプリケーション管理情報に記載されており、モジュールマネージャがアプリケーション管理情報に従って、Javaアプリケーションの起動及び終了を管理する。Javaアプリケーションには、起動したタイトル内のみ生存可能なアプリ(タイトルバウンドアプリ)と複数のタイトルにまたがって生存可能なアプリ(タイトルアンバウンドアプリ)とが存在する。アプリケーション管理情報に、そのJavaアプリケーションがタイトルバウンドアプリか、もしくはタイトルアンバウンドアプリかを示す情報と、アプリケーションIDと、生存可能なタイトル番号とが記載されている。

#### 【0126】

図43に記載したアプリケーション管理情報の例では、Title#1においてはJavaアプリ#1はタイトルバウンドアプリ、Javaアプリ#2はタイトルアンバウンドアプリとなる。タイトルバウンドアプリであるJavaアプリ#1は、Title#1の終了と共にモジュールマネージャにより終了させられる。一方、タイトルアンバウンドアプリであるJavaアプリ#2は、Title#1が終了しても生存し続け、遷移先のタイトルにおけるアプリケーション管理情報に応じて、生存し続けるか終了するかが決定される。図43の例においては、Title#1からTitle#2への遷移の場合、どちらのタイトルにもJavaアプリ#2が生存可能であるので、Title#1からTitle#2の遷移にかけては、Javaアプリ#2は生存し続けることになる。ただしTitle#2においてはJavaアプリ#2はタイトルバウンドアプリとなっているので、Title#2の終了と共にJavaアプリ#2も終了する。

#### 【0127】

図44は、タイトル切り換え時のVirtual Package更新におけるタイトルアンバウンドアプリに対する処理を示す図である。図43で述べたように、タイトルアンバウンドアプリでタイトル切り換え前と切り換え後の両方のタイトルで生存できる場合は、タイトル切り換えが発生しても継続して動作し続ける。しかし、Virtual Package更新要求が発行されていた場合は、タイトル切り換え時にタイトルアンバウンドアプリを含む全アプリを一旦終了させる。そして、Virtual Packageの更新が完了した後、再びタイトルアンバウンドアプリ及び遷移先タイトルのタイトルバウンドアプリを起動する。

#### 【0128】

図45は、タイトルアンバウンドアプリを考慮したタイトル切り換えのフローチャートである。まず、タイトル再生が開始されると(ステップS111)、そのタイトル再生中にJavaアプリからVirtual Package更新要求が発行されたか否かを判断する(ステップS112)。要求があれば、Virtual FileSystem部はVirtual Packageの更新準備を行う(

10

20

30

40

50

ステップS113)。そして、タイトル切り換えが発生すると(ステップS114)、Virtual Packageの更新要求を受け付けていたか否かを判断する(ステップS115)。受け付け済みの場合、タイトルアンバウンドアプリを含む全アプリを終了させる(ステップS116)。そして、全アプリを終了させた後、Virtual Packageの更新作業を行い(ステップS117)、その後切り換え先のタイトル再生を行う(ステップS118)。タイトル切り換え発生以前にVirtual Package更新要求が発行されていなかった場合は、タイトル切り換え時にはタイトルバウンドアプリのみ終了させる(ステップS119)。

#### 【0129】

以上のように本実施形態によれば、Virtual Package更新処理時には全アプリが終了していることが保証されるため、更新前の古いファイルへの参照が残っていたり、キャッシュに古いファイルが残っているため新しいファイルと古いファイルが混在するなど、Virtual Package更新後に整合性が失われてしまう問題を防ぐことが出来る。10

なお、Virtual Package更新を伴うタイトル切り換え時に複数のディスクにまたがって生存可能なアプリケーション(ディスクアンバウンドアプリ)が動作中の場合、Virtual Package更新をディスクの入れ替え操作と同様と捉えることにより、ディスクアンバウンドアプリに関してはVirtual Package更新を伴うタイトル切り換えが発生しても、強制終了させずに動作し続けることにもよい。

#### 【0130】

なお、タイトル切り換えが発生した後、Virtual Package更新処理が完了するまではタイトルアンバウンドアプリを終了させず、Virtual Package更新処理完了後に更新されたアプリケーション管理情報に従って、タイトルアンバウンドアプリの起動及び終了を管理してもよい。この場合、Virtual Package更新処理が完了するまで、タイトルアンバウンドアプリからのVirtual Packageへの参照は、更新前のVirtual Packageへ参照させる。20

#### 【0131】

##### (第7実施形態)

本実施形態は、INDEX.BDMVファイルの変更を伴うVirtual Package更新についての実施の形態である。JavaアプリからのVirtual Package更新要求をバーチャルファイルシステム部が受けると、次にマージ対象となるディレクトリにINDEX.BDMVファイルが存在するか確認する。存在すれば、Virtual Packageの更新前の準備段階において、INDEX.BDMVファイルのみ先読みを行う。そして、現在有効なINDEX.BDMVファイルを無効化し、新たなINDEX.BDMVファイルを有効にする。BDプレーヤのレジデントアプリがタイトルサーチを行う場合や、Javaアプリがタイトル情報を取得する場合は、新たなINDEX.BDMVファイルを元に行う。つまり、Virtual Package更新後のタイトル構成を事前にJavaアプリやユーザに知らせることで、更新後に存在しないタイトルへの切り換えや、予期せぬタイトルへの切り換えを防ぐことが出来る。30

#### 【0132】

図46は、INDEX.BDMVファイルの変更を伴うVirtual Package更新のフローチャートである。まず、Javaモード用のタイトルが再生されると(ステップS121)、タイトル再生中にJavaアプリケーションがVirtual Packageの更新要求を行ったか否かを判断する(ステップS122)。更新要求があると、バーチャルファイルシステム部は更新要求を受けとり、preparing処理を行う(ステップS123)。ファイル構成、ディレクトリ構成が正しいかどうかのチェックを行うと同時にINDEX.BDMVファイルが含まれているかどうかを判断する(ステップS124)。INDEX.BDMVファイルが含まれていた場合は、現在有効となっているINDEX.BDMVを無効化し、新たなINDEX.BDMVファイルを有効にする(ステップS125)。次にタイトル切り換えが発生したか否かを判断する(ステップS126)。プレーヤ内のレジデントアプリ、もしくはJavaアプリからタイトル切り換えが行われるときは、ステップS125で有効にしたINDEX.BDMVファイルを参照してタイトル切り換えを行う。タイトルの切り換えが発生すると、Updating処理を行う(ステップS127)。40

#### 【0133】

このように、Virtual Package更新要求後、タイトル切り換えが発生するまではVirtual50

Packageの更新はされないが、INDEX.BDMVファイルはVirtual Package更新後のINDEX.BDMVファイルが有効になる。そのため、Virtual Package更新要求後は、タイトル切り換え発生前に、タイトルサーチ時に表示されるタイトルリストが変化することになる。

ユーザは、変更されたタイトルリストを元にタイトルを選択するため、Virtual Package更新後に存在しないタイトルを選択してしまうことによるエラーを防ぐことができ、Virtual Package更新によりタイトル構成が変化しても、タイトル切り換え時に問題なくVirtual Packageを更新することが可能となる。

#### 【0134】

なお、INDEX.BDMVファイルの変更を伴うVirtual Package更新は、BD再起動時でも良い。

10

#### ( その他の変形例 )

以上、本発明に係る再生装置について、実施の形態に基づいて説明したが、本発明は上記の実施の形態に限られないことは勿論である。

#### 【0135】

上記実施形態では、記録媒体を再生する再生機能のみを持つ再生装置について説明したが、それに限らない。例えば、録画機能を持つ録画再生装置であっても良い。

また、ローカルストレージ上のファイルを置く構造は組み合わせる対象となるBD-ROM上のファイルとの対応関係が明確に示される構造であれば、どのような構造を用いてローカルストレージ上にファイルを置いても構わない。

20

#### 【0136】

上記実施形態では、仮想マシンのプログラミング言語としてJava(登録商標)を利用したが、Java(登録商標)ではなく、UNIX(登録商標)OSなどで使われているB-Shellや、Perl Script、ECMA Scriptなど他のプログラミング言語であっても良い。

また、上述の実施の形態ではBD-ROMを再生する再生装置について説明をしたが、書き込み可能な光記録媒体に上述の実施の形態にて説明をしたBD-ROM上の必要なデータが記録されていた場合においても上述と同様の効果を奏することはもちろんのことである。

#### 【0137】

更には光記録媒体以外にも可搬性の記録媒体(例えばSDカードやコンパクトフラッシュ(登録商標)など)に上述の実施の形態にて説明をしたBD-ROM上の必要なデータが記録されていた場合においても上述と同様の効果を奏することはもちろんのことである。

30

#### 【産業上の利用可能性】

#### 【0138】

本発明を構成する再生装置は、製造産業において経営的に、また継続的及び反復的に製造し、販売することができる。特に、映像コンテンツの制作に携わる映画産業、民生機器産業において利用できる。

#### 【図面の簡単な説明】

#### 【0139】

【図1】システム図である。

【図2】BD-ROM 100の内部構成を示す図である。

40

【図3】拡張子.M2TSが付与されたファイルがどのように構成されているかを模式的に示す図である。

【図4】PL情報の構成を示す図である。

【図5】AVClipと、PLとの関係を示す図である。

【図6】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7】PLmark情報の内部構成を示す図である。

【図8】PLmarkによるチャプター定義を示す図である。

【図9】SubPath情報の内部構成を示す図である。

【図10】SubClip時間軸上の再生区間定義と、同期指定を示す図である。

【図11】( a ) 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

50

である。(b) クラスファイルの内部構成を示す図である。

【図12】BD-Jオブジェクトの内部構成を示す図である。

【図13】INDEX.BDMVの内部構成を示す図である。

【図14】Local Storageにおけるディレクトリ構造を示す図である。

【図15】ローカルストレージに格納されたPL情報により、どのようなPlayList再生時間軸が定義されるかを示す図である。

【図16】(a) BD-ROM及びLocal Storageに記録されているAVClip、またはアプリケーションを示す図である。(b) AVClip#1～AVClip#4、アプリケーションを1つのタイトルとして扱う様子を示す図である。

【図17】マージ管理情報ファイルの内部構成の一例を示す図である。 10

【図18】再生装置のハードウェア構成を示す図である。

【図19】命令ROM21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、レイヤ構成に置き換えて描いた図である。

【図20】Java仮想マシン30の内部構成を示す図である。

【図21】タイトル切り換えによる状態遷移を示す図である。

【図22】Virtual FileSystem部38によるVirtual Package情報構築の一例を示す図である。

【図23】(a) ディスク全体の時間軸を示す図である。(b) ディスク全体の時間軸における構成を示す。 20

【図24】Javaアプリケーションによるダウンロード手順を示すフローチャートである。

【図25】Update RequestのPreparingの処理手順を示すフローチャートである。

【図26】Virtual packageのUpdatingの処理を示すフローチャートである。

【図27】モジュールマネージャの処理手順を示すフローチャートである。

【図28】PL再生手順を示すフローチャートである。

【図29】Title Changeにおいて、Virtual Package情報のアップデートがどのように行われるかを示す図である。

【図30】Javaアプリケーションが、現在のマージ管理情報ファイルをサーバに送信する様子を示した図である。

【図31】Javaアプリケーションが、コンテンツファイルと新しいマージ管理情報ファイルと署名情報ファイルをダウンロードする様子を示した図である。 30

【図32】Javaアプリケーションが、バーチャルファイルシステムに対し、更新要求する様子を示した図である。

【図33】マージ管理情報ファイルと署名情報ファイルの置き換え、及びマッピングの様子を示した図である。

【図34】(a) タイトル呼出にあたって、カレントタイトルの再生を一時的に停止(suspend)する際のPlayback Control Engineの処理手順を示すフローチャートである。(b) 呼出先のタイトルの再生終了後、元のタイトルの再生を再開(resume)する際のPlayback Control Engineの処理手順を示すフローチャートである。

【図35】第3実施形態におけるマージ管理情報を示す図である。

【図36】第3実施形態に係るJavaアプリケーションがVirtual Package構築を要求している様子を示す図である。 40

【図37】レジデントアプリケーションがマージ管理情報を用いて、ユーザに対して追加コンテンツ一覧表示を行う一例を示す図である。

【図38】第3実施形態に係るBD-ROM挿入から再生までの処理の流れを示すフローチャートである。

【図39】有効区間を指定してVirtual Package構築要求を行っている様子を示す図である。

【図40】第4実施形態に係るBD-ROM挿入、または再起動から再生までの処理の流れを示すフローチャートである。

【図41】パーミッションリクエストファイルによるVirtual Packageの構築要求を制限 50

している様子を示す図である。

【図42】ローカルストレージ内のマージの対象となるファイルに対するアクセス制限を示す図である。

【図43】タイトル内だけで生存するJavaアプリケーションとタイトルにまたがって生存するJavaアプリケーションのライフサイクルを示す図である。

【図44】タイトル切り換え時のVirtual Package更新におけるタイトルアンバウンドアプリに対する処理を示す図である。

【図45】タイトルアンバウンドアプリを考慮したタイトル切り換えのフローチャートである。

【図46】INDEX.BDMVファイルの変更を伴うVirtual Package更新のフローチャートである。

10

【符号の説明】

【0140】

- 1 BD-ROMドライブ
- 2 Read Buffer
- 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
- 8 合成部
- 9 フォントゼネレータ
- 10 I-Graphicsデコーダ
- 11 スイッチ
- 12 Interactive Graphicsプレーン
- 13 合成部
- 14 CLUT部
- 15 CLUT部
- 16 オーディオデコーダ
- 17 Network Device
- 18 Local Storage
- 19 Read Buffer
- 20 デマルチプレクサ
- 21 命令ROM
- 22 ユーザイベント処理部
- 23 PSRセット
- 24 CPU
- 25 シナリオメモリ
- 26 Local Memory
- 27 スイッチ
- 28 Players Status Register
- 29a DVDライクモジュール
- 29b Javaプラットフォーム
- 30 Java仮想マシン
- 31 Presentation Engine
- 32 Playback Control Engine
- 33 モジュールマネージャ
- 37 ネットワーク管理モジュール
- 38 Virtual File System
- 52 ユーザクラスローダ

20

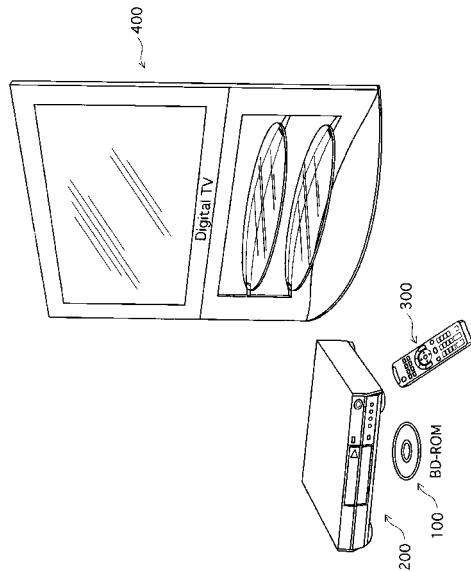
30

40

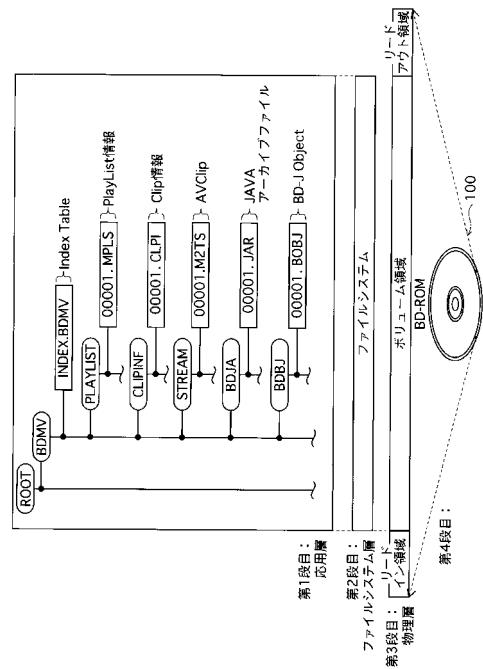
50

- 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・・・n Javaスタック  
 1 0 0 BD-ROM  
 2 0 0 再生装置  
 3 0 0 リモコン  
 4 0 0 テレビ

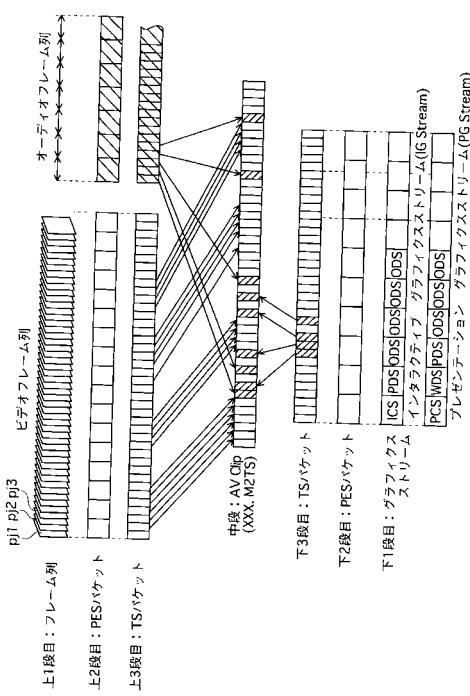
【図 1】



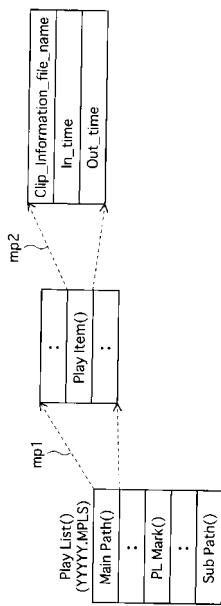
【図 2】



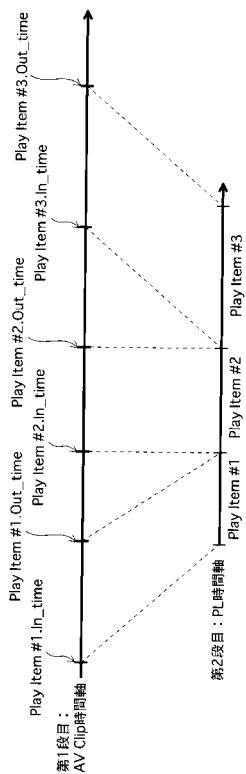
【図3】



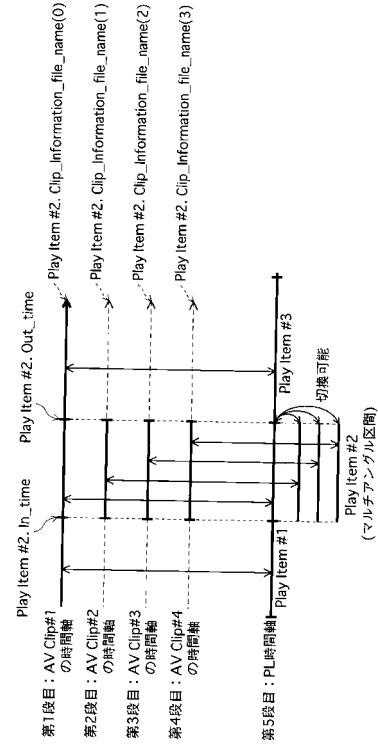
【 図 4 】



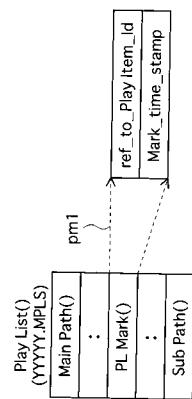
【圖 5】



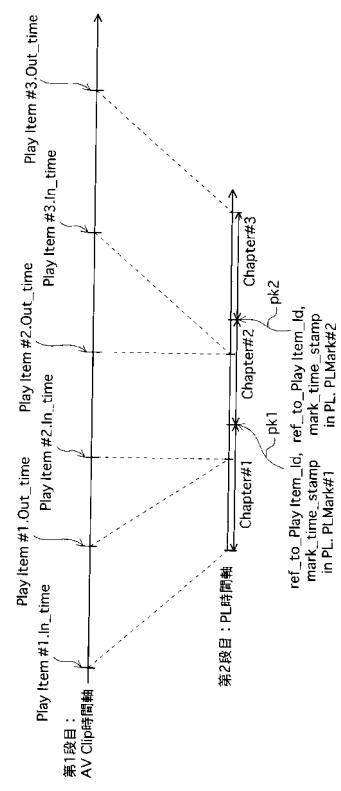
【図6】



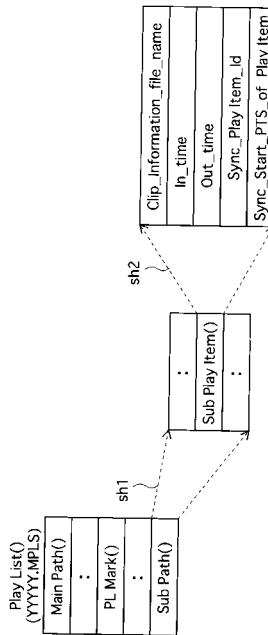
【図7】



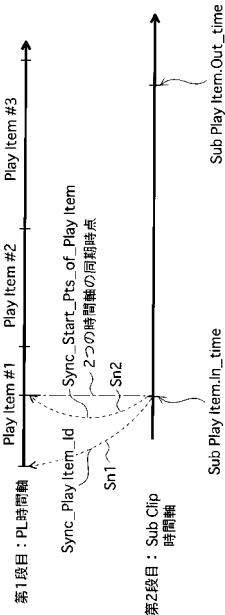
【図8】



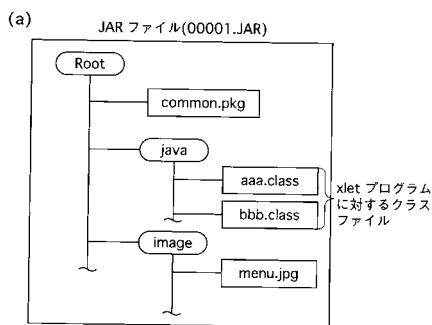
【図9】



【図10】

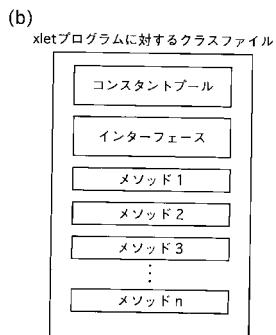


【図 1 1】

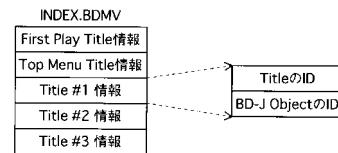


Javaアプリとストリームとを関連付ける情報 … BD-JObject  
XXXX.BOBJ

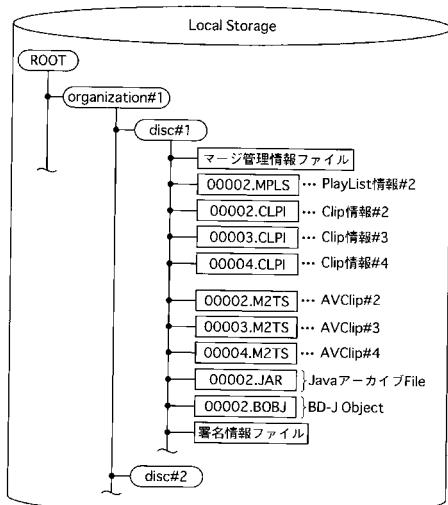
Playlist情報IDの参照値  
アプリケーション管理テーブル  
(このTitleを生存区間として  
いるJavaアプリを示す)



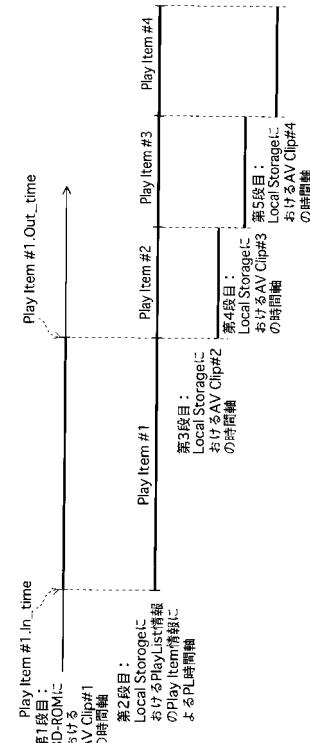
【図 1 3】



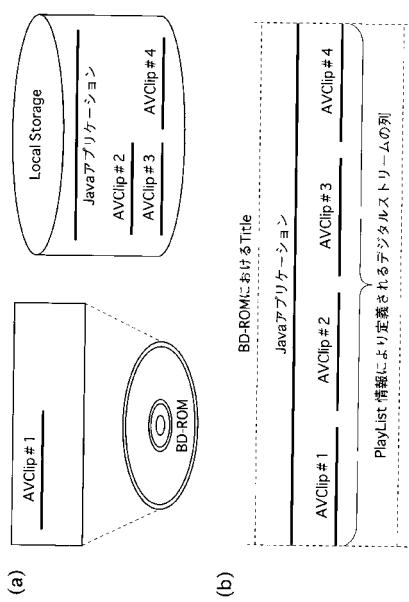
【図 1 4】



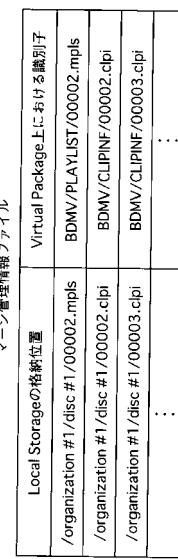
【図 1 5】



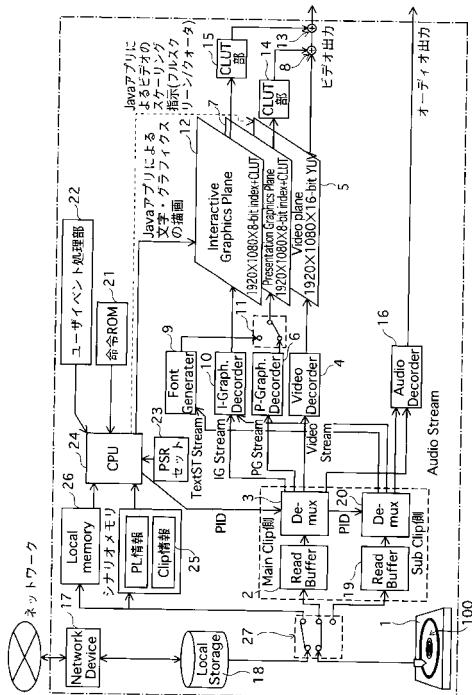
【図16】



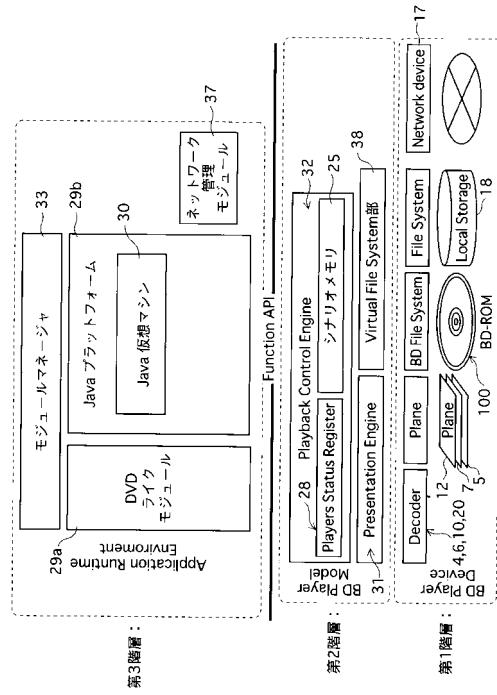
【図17】



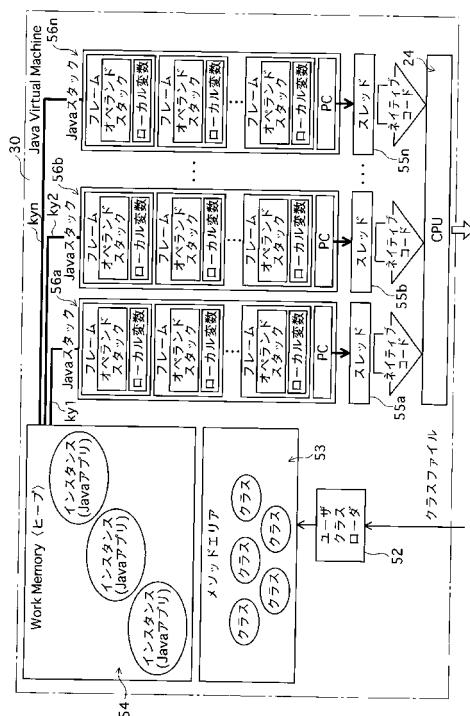
【図18】



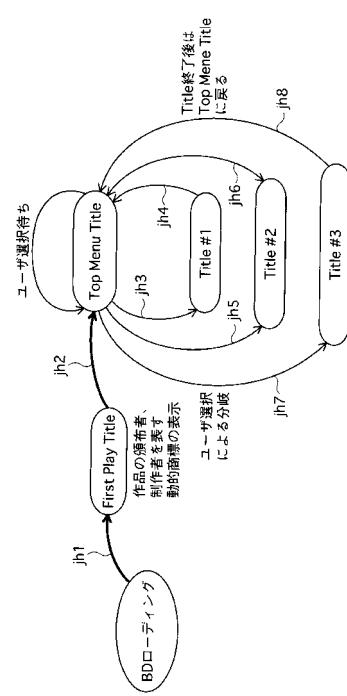
【図19】



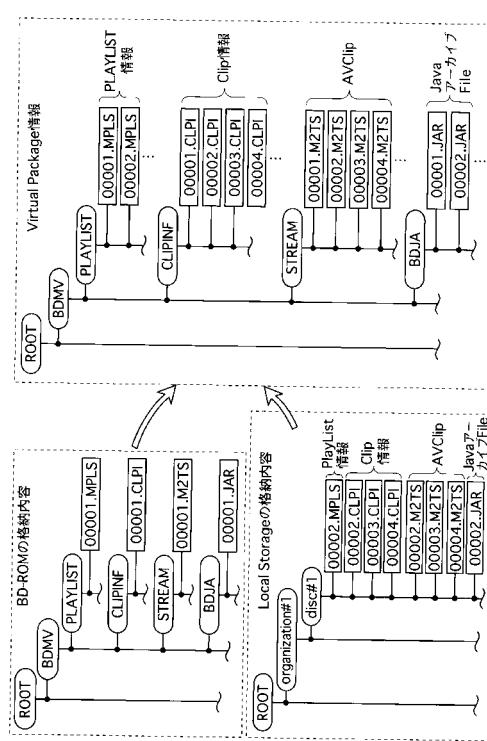
【 図 2 0 】



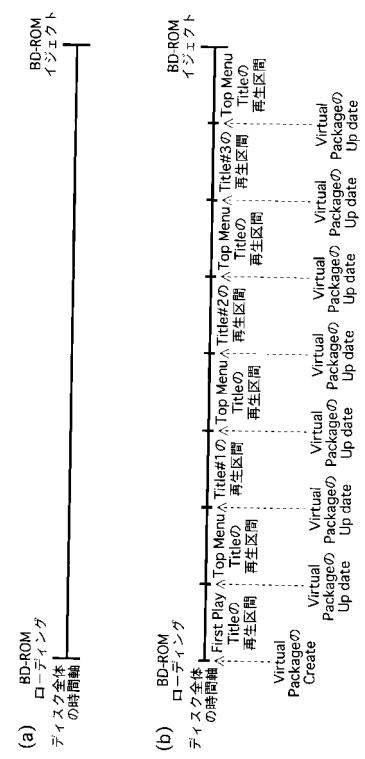
## 【図21】



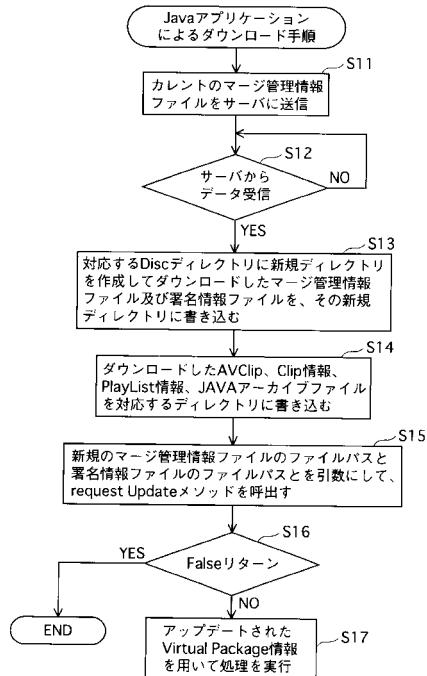
【 図 2 2 】



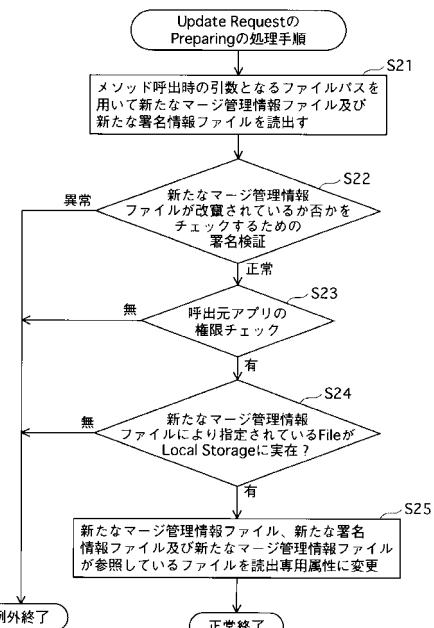
【 図 2 3 】



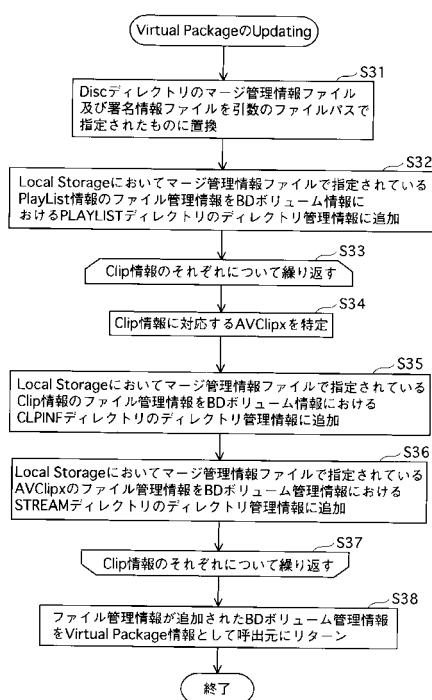
【図24】



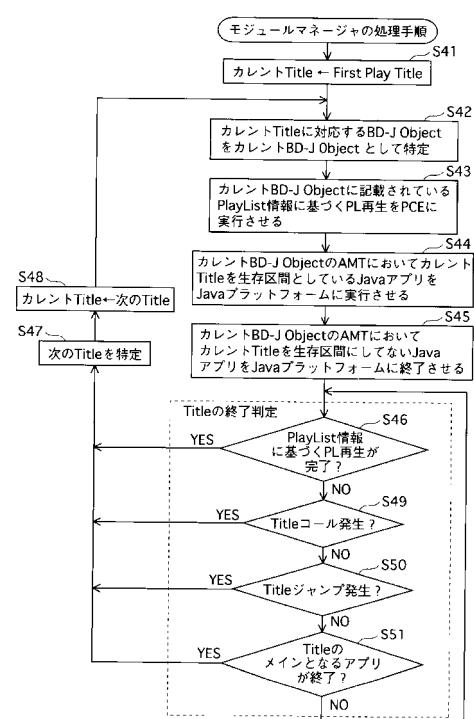
【図25】



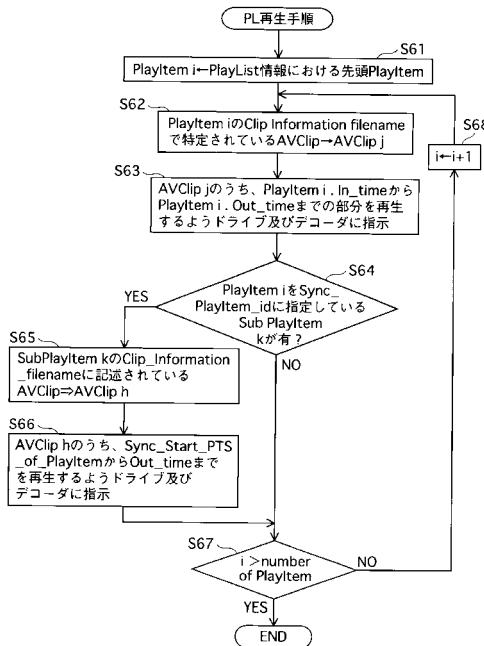
【図26】



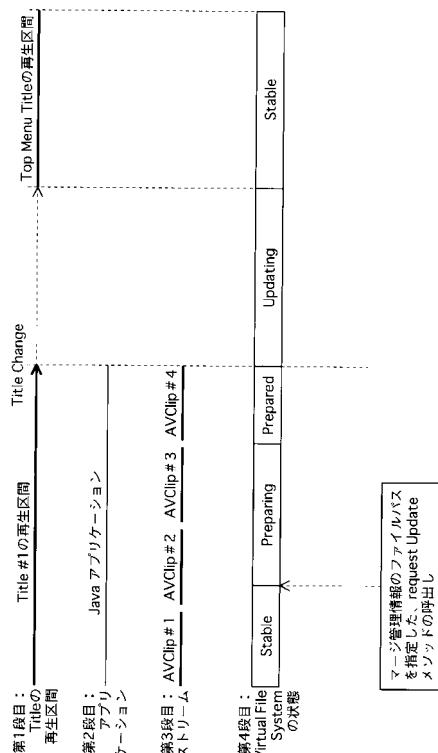
【図27】



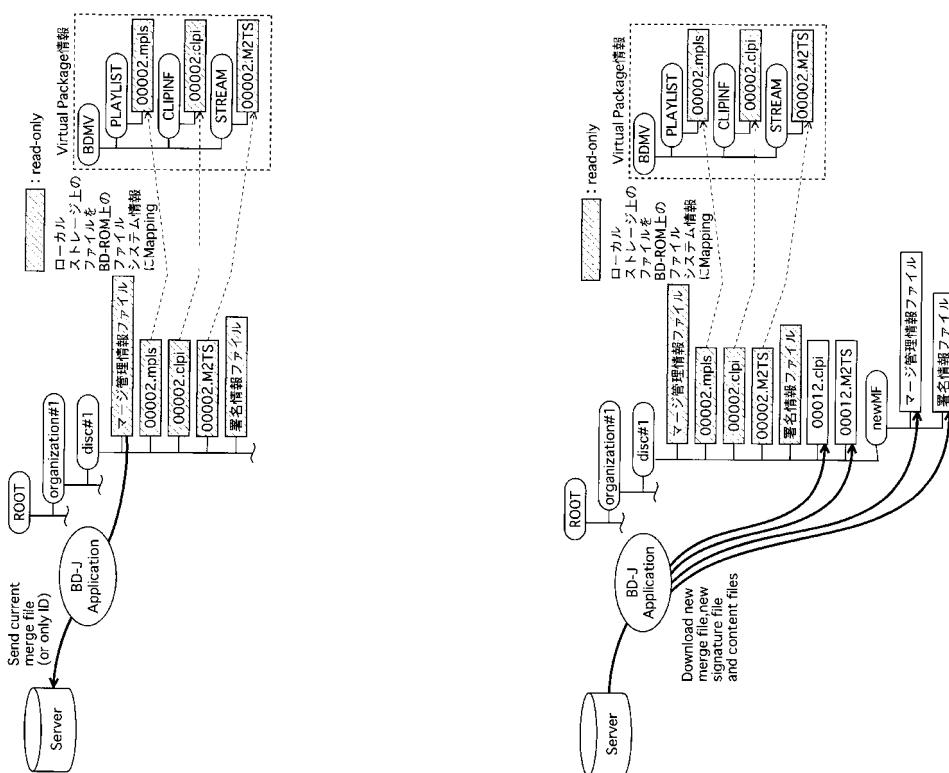
【図28】



【図29】

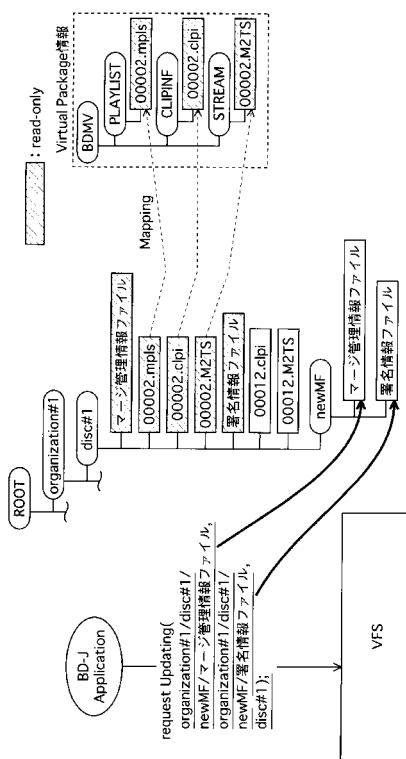


【図30】

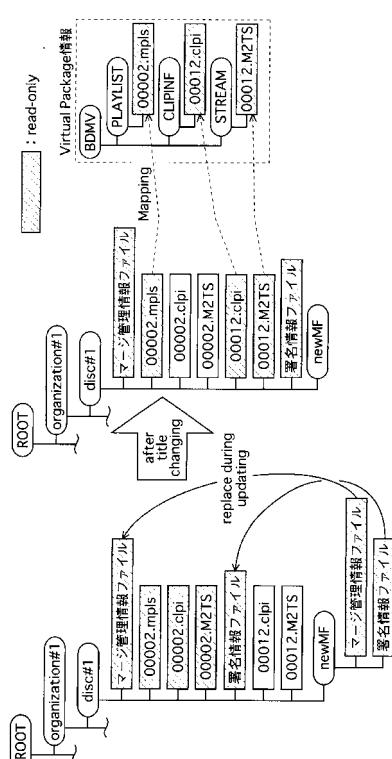


【図31】

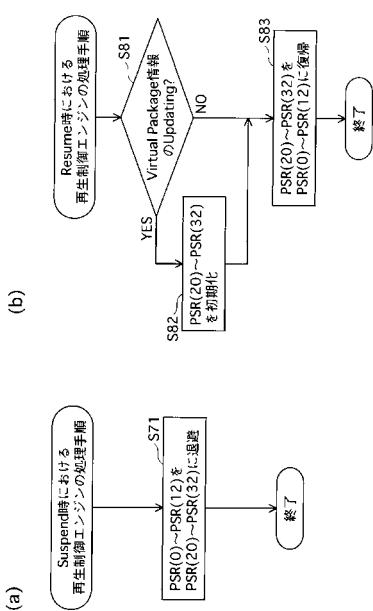
【図32】



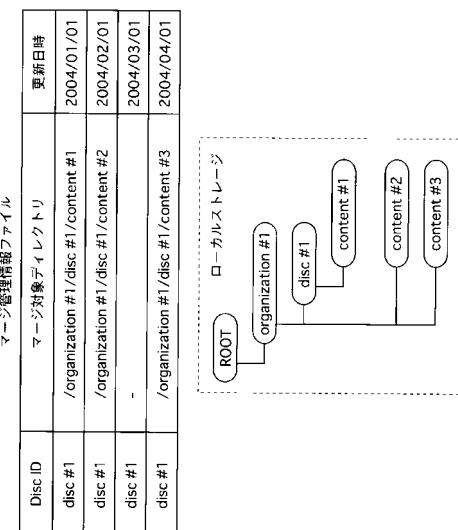
【図33】



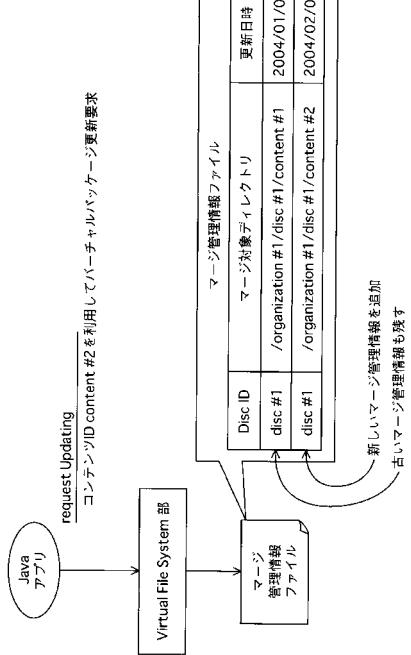
【図34】



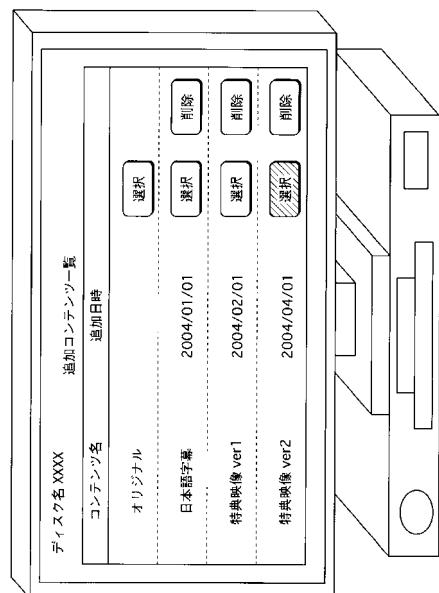
【図35】



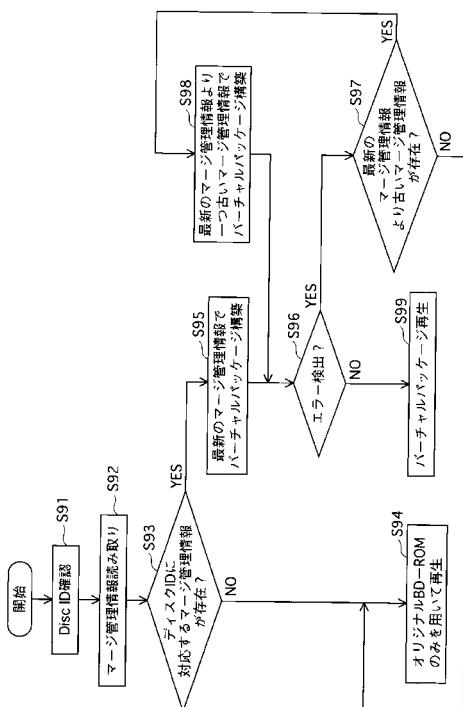
【図36】



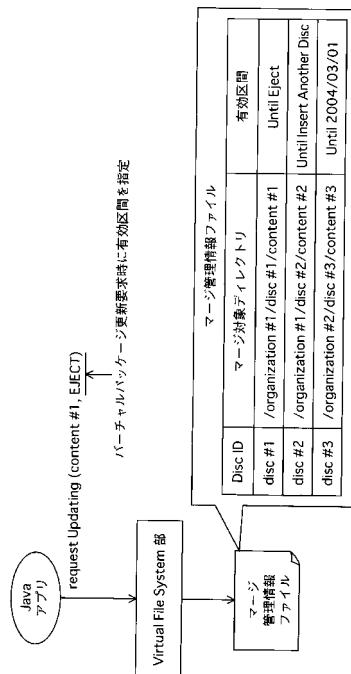
【図37】



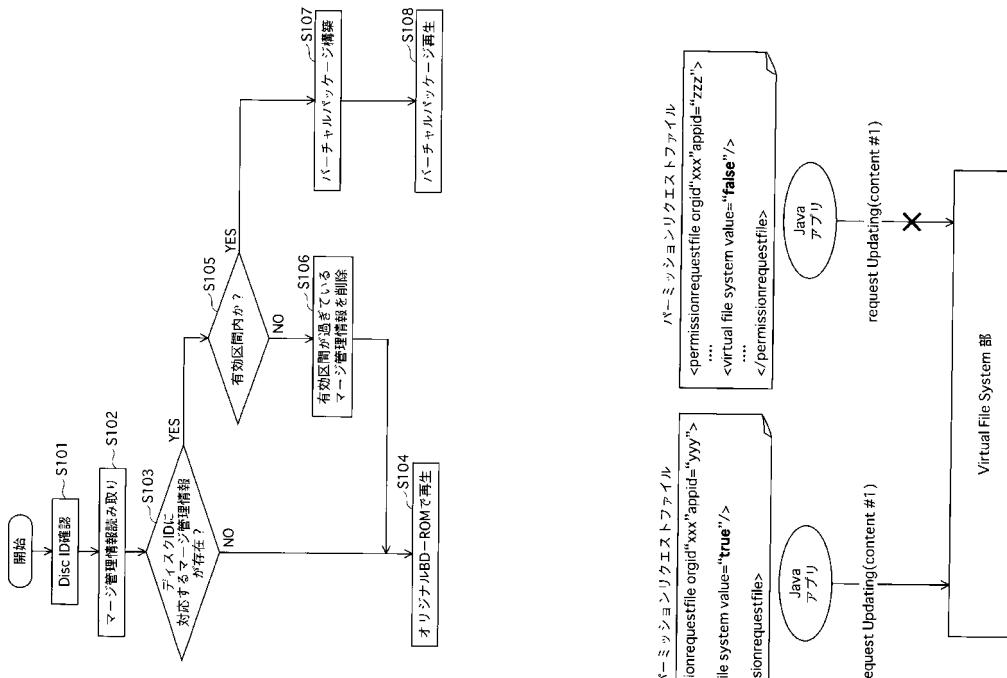
【図38】



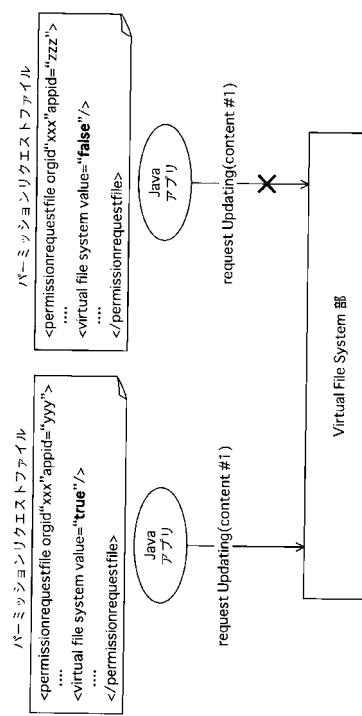
【図39】



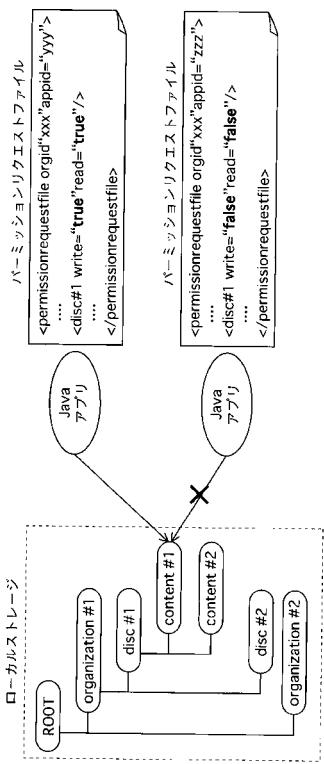
【図40】



【図41】

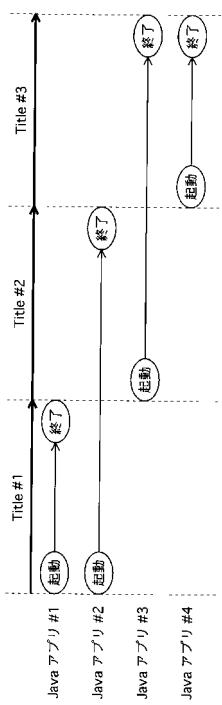


【図42】

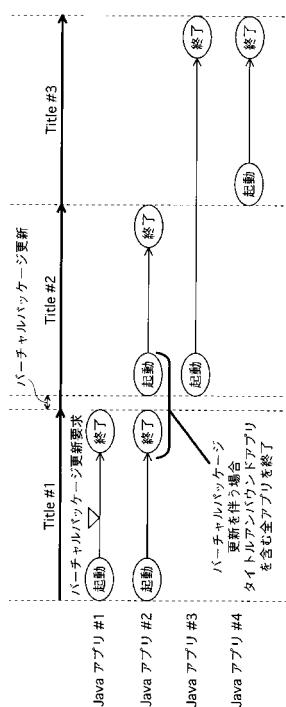


【図43】

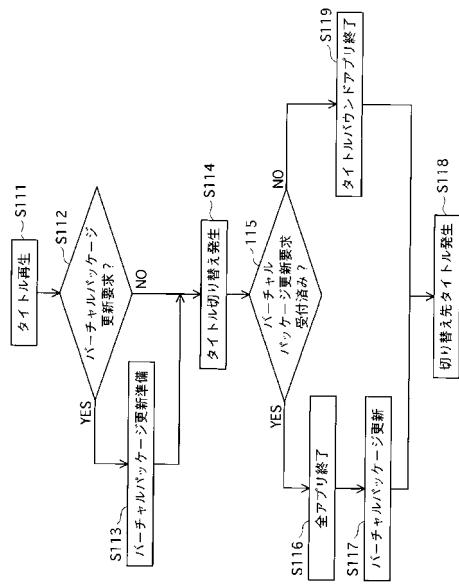
アブリケーション管理情報		
タイトル	アブリケーションID	タイトルバウンド属性
Title #1	Java アプリ #1	Bound
	Java アプリ #2	Unbound
Title #2	Java アプリ #2	Bound
	Java アプリ #3	Unbound
Title #3	Java アプリ #3	Bound
	Java アプリ #4	Bound



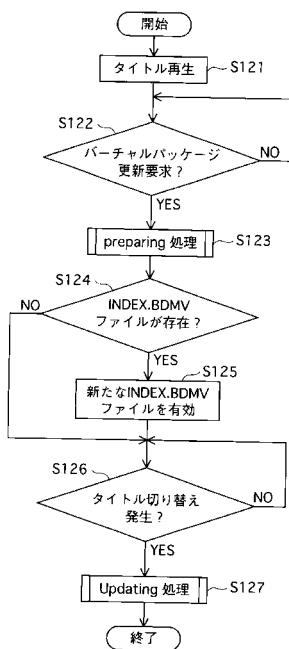
【図4-4】



【図4-5】



【図4-6】



---

フロントページの続き

(31)優先権主張番号 特願2004-293043(P2004-293043)

(32)優先日 平成16年10月5日(2004.10.5)

(33)優先権主張国 日本国(JP)

(31)優先権主張番号 特願2004-369701(P2004-369701)

(32)優先日 平成16年12月21日(2004.12.21)

(33)優先権主張国 日本国(JP)

(31)優先権主張番号 特願2005-99410(P2005-99410)

(32)優先日 平成17年3月30日(2005.3.30)

(33)優先権主張国 日本国(JP)

(72)発明者 大蘆 雅弘

大阪府門真市大字門真1006番地 松下電器産業株式会社内

審査官 金子 秀彦

(56)参考文献 特開平11-164070(JP, A)

特開2001-517836(JP, A)

特開2004-240832(JP, A)

特開2002-247526(JP, A)

国際公開第2004/030356(WO, A1)

(58)調査した分野(Int.Cl., DB名)

G11B 27/10

G11B 20/10

G11B 27/00

H04N 5/93