

(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2006-59435
(P2006-59435A)

(43) 公開日 平成18年3月2日(2006.3.2)

(51) Int. Cl.	F I	テーマコード (参考)
G 1 1 B 27/10 (2006.01)	G 1 1 B 27/10 A	5 C 0 5 3
G 1 1 B 20/10 (2006.01)	G 1 1 B 20/10 3 2 1 Z	5 D 0 4 4
G 1 1 B 20/12 (2006.01)	G 1 1 B 20/12	5 D 0 7 7
G 1 1 B 27/00 (2006.01)	G 1 1 B 27/00 D	5 D 1 1 0
H O 4 N 5/93 (2006.01)	H O 4 N 5/93 G	

審査請求 未請求 請求項の数 17 O L (全 58 頁)

(21) 出願番号	特願2004-239347 (P2004-239347)	(71) 出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22) 出願日	平成16年8月19日 (2004.8.19)	(71) 出願人	395015319 株式会社ソニー・コンピュータエンタテインメント 東京都港区南青山二丁目6番21号
		(74) 代理人	100082762 弁理士 杉浦 正知
		(74) 代理人	100123973 弁理士 杉浦 拓真
		(74) 代理人	100120640 弁理士 森 幸一

最終頁に続く

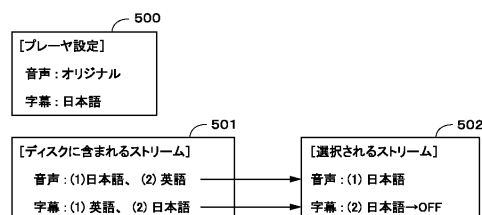
(54) 【発明の名称】 再生装置、再生方法および再生プログラム、ならびに、記録媒体

(57) 【要約】

【課題】 ディスクの再生時に、音声及び字幕を適切に自動選択することができるようにする。

【解決手段】 プレーヤ側に、再生する音声ストリームの言語としてオリジナル言語を設定可能とする。記録媒体側では、オリジナル言語による音声ストリームをクリップ情報ファイルにおいて最も先頭側に記述する。また、音声及び字幕の自動設定の結果、両者の言語が一致した場合、字幕を非表示とする。例えば、プレーヤ設定500は音声オリジナル言語、字幕が日本語となっており、ディスクに含まれるストリーム501は、音声ストリームは日本語(オリジナル言語)と英語、字幕は英語と日本語である場合、自動設定により音声はオリジナル言語の日本語が選択され、字幕も日本語が選択される。音声と字幕とで言語が一致するので、最終的に選択されるストリーム502は、音声は日本語が選択され、字幕は日本語が選択されるが非表示とされる。

【選択図】 図38



【特許請求の範囲】**【請求項 1】**

円盤状記録媒体に記録されたコンテンツデータを再生する再生装置において、
少なくとも、ビデオストリーム、該ビデオストリームにそれぞれ対応する 1 または複数の音声ストリームおよび 1 または複数の字幕ストリームを含むコンテンツデータと、該コンテンツデータの再生を制御する再生制御プログラムとが記録された記録媒体からデータを読み出す読み出し手段と、

上記再生制御プログラムに従い上記コンテンツデータを再生するプレーヤ手段と、

上記プレーヤ手段に対して上記コンテンツデータの再生時に上記 1 または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択する第 1 のモードを設定する第 1 のモード設定手段と
を備えることを特徴とする再生装置。

10

【請求項 2】

請求項 1 に記載の再生装置において、

上記プレーヤ手段に対して上記コンテンツデータの再生時に上記字幕ストリームを再生するか否かを選択する第 2 のモードを設定する第 2 のモード設定手段をさらに備えることを特徴とする再生装置。

【請求項 3】

請求項 2 に記載の再生装置において、

上記 1 または複数の音声ストリームから 1 の音声ストリームを自動選択するか否かを設定する音声ストリーム自動選択設定手段と、

20

上記 1 または複数の字幕ストリームから 1 の字幕ストリームを自動選択するか否かを設定する字幕ストリーム自動選択設定手段と
をさらに備え、

上記音声ストリーム自動選択設定手段の設定に基づき自動選択された上記音声ストリームの言語と、上記字幕ストリーム自動選択設定手段の設定に基づき自動選択された上記字幕ストリームの言語とが一致した場合、該選択された字幕ストリームを再生しないように上記第 2 のモード手段の上記第 2 のモードを自動設定するようにしたことを特徴とする再生装置。

【請求項 4】

30

請求項 3 に記載の再生装置において、

上記音声ストリーム自動選択設定手段による上記自動選択は、上記モード設定手段により上記第 1 のモードが設定されていれば、上記 1 または複数の音声ストリームからオリジナル言語による音声ストリームを選択し、上記モード設定手段により上記第 1 のモードが設定されていなければ、上記 1 または複数の音声ストリームから上記プレーヤ手段に設定された言語による音声ストリームを選択するようにしたことを特徴とする再生装置。

【請求項 5】

請求項 3 に記載の再生装置において、

上記字幕ストリーム自動設定手段による上記自動選択は、上記 1 または複数の字幕ストリームから上記プレーヤ手段に設定された言語による字幕ストリームを選択するようにしたことを特徴とする再生装置。

40

【請求項 6】

請求項 1 に記載の再生装置において、

上記記録媒体には、少なくとも上記 1 または複数の音声ストリームのそれぞれを特定する情報が記述されるストリーム情報がさらに記録され、

上記プレーヤ手段は、上記 1 または複数の上記音声ストリームのうち上記ストリーム情報内で最も先頭側に記述された上記音声ストリームを上記オリジナル言語による音声ストリームであると判断するようにしたことを特徴とする再生装置。

【請求項 7】

請求項 6 に記載の再生装置において、

50

上記ストリーム情報には、さらに、上記音声ストリームのチャンネル数情報が記述され

、
上記プレーヤ手段は、上記 1 または複数の上記音声ストリームのうち、上記オリジナル言語による音声ストリームと同一の言語で、チャンネル数が最大の音声ストリームを自動的に選択するようにしたことを特徴とする再生装置。

【請求項 8】

請求項 7 に記載の再生装置において

上記プレーヤ手段は、上記 1 または複数の上記音声ストリームのうち、上記オリジナル言語による音声ストリームと同一の言語で、チャンネル数が同一の音声ストリームが複数、存在するときは、上記ストリーム情報内でより先頭側に記述された上記音声ストリームを選択するようにしたことを特徴とする再生装置。

10

【請求項 9】

請求項 1 に記載の再生装置において、

上記記録媒体には、少なくとも上記 1 または複数の字幕ストリームのそれぞれを特定する情報が記述されるストリーム情報がさらに記録され、

上記プレーヤ手段は、上記 1 または複数の字幕ストリームのうち、同一の言語の字幕ストリームが複数、存在するときは、上記ストリーム情報内でより先頭側に記述された上記字幕ストリームを選択するようにしたことを特徴とする再生装置。

【請求項 10】

円盤状記録媒体に記録されたコンテンツデータを再生する再生方法において、

20

少なくとも、ビデオストリーム、該ビデオストリームにそれぞれ対応する 1 または複数の音声ストリームおよび 1 または複数の字幕ストリームを含むコンテンツデータと、該コンテンツデータの再生を制御する再生制御プログラムとが記録された記録媒体からデータを読み出す読み出しのステップと、

上記再生制御プログラムに従い上記コンテンツデータを再生するコンテンツ再生のステップと、

上記コンテンツ再生のステップに対して上記コンテンツデータの再生時に上記 1 または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択するモードを設定するモード設定のステップと

を備えることを特徴とする再生方法。

30

【請求項 11】

円盤状記録媒体に記録されたコンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムにおいて、

上記再生方法は、

少なくとも、ビデオストリーム、該ビデオストリームにそれぞれ対応する 1 または複数の音声ストリームおよび 1 または複数の字幕ストリームを含むコンテンツデータと、該コンテンツデータの再生を制御する再生制御プログラムとが記録された記録媒体からデータを読み出す読み出しのステップと、

上記再生制御プログラムに従い上記コンテンツデータを再生するコンテンツ再生のステップと、

40

上記コンテンツ再生のステップに対して上記コンテンツデータの再生時に上記 1 または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択するモードを設定するモード設定のステップと

を備えることを特徴とする再生プログラム。

【請求項 12】

少なくとも、

ビデオストリームと、

当該ビデオストリームにそれぞれ対応する 1 または複数の音声ストリームおよび 1 または複数の字幕ストリームを含むコンテンツデータと、

該コンテンツデータの再生を制御する再生制御プログラムと

50

少なくとも上記 1 または複数の音声ストリームのそれぞれを特定する情報が、オリジナル言語として用いる上記音声ストリームを特定する該情報を最も先頭側に配して記述されるストリーム情報と

が記録されることを特徴とする記録媒体。

【請求項 13】

請求項 12 に記載の記録媒体において、

上記再生制御プログラムは、上記コンテンツデータを再生する再生装置に対して、上記ストリーム情報に基づき上記オリジナル言語を自動選択するように設定できることを特徴とする記録媒体。

【請求項 14】

請求項 13 に記載の記録媒体において、

上記再生制御プログラムは、上記再生装置に対して、上記 1 または複数の音声ストリームから 1 の音声ストリームを自動選択するように設定できると共に、上記 1 または複数の字幕ストリームから 1 の字幕ストリームを自動選択するように設定でき、上記再生装置に対し、上記設定に基づき自動選択された上記音声ストリームの言語と上記字幕ストリームの言語とが一致した場合、該選択された字幕ストリームを再生しないように設定させるようにしたことを特徴とする記録媒体。

【請求項 15】

請求項 12 に記載の記録媒体において、

上記ストリーム情報には、さらに、上記音声ストリームのチャンネル数情報が記述され

、上記コンテンツデータを再生する再生装置に対して、上記 1 または複数の上記音声ストリームのうち、上記オリジナル言語による音声ストリームと同一の言語で、チャンネル数が最大の音声ストリームを自動的に選択させるようにしたことを特徴とする記録媒体。

【請求項 16】

請求項 15 に記載の記録媒体において

上記 1 または複数の上記音声ストリームのうち、上記オリジナル言語による音声ストリームと同一の言語で、チャンネル数が同一の音声ストリームが複数、存在するときは、選択させたい上記音声ストリームを、上記ストリーム情報内でより先頭側に記述するようにしたことを特徴とする記録媒体。

【請求項 17】

請求項 12 に記載の記録媒体において、

上記ストリーム情報には、さらに、少なくとも上記 1 または複数の字幕ストリームのそれぞれを特定する情報が記述され、

上記 1 または複数の字幕ストリームのうち、同一の言語の字幕ストリームが複数、存在するときは、選択させたい上記字幕ストリームを、上記ストリーム情報内でより先頭側に記述するようにしたことを特徴とする記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、大容量の記録媒体に記録されたプログラムの再生時の音声および字幕ストリームを適切に自動選択できるようにした再生装置、再生方法および再生プログラム、ならびに、記録媒体に関する。

【背景技術】

【0002】

ランダムアクセスおよび着脱が可能な記録媒体として、DVD (Digital Versatile Disc) が出現して久しいが、近年では、この DVD よりも大容量のディスク状記録媒体や、DVD より携帯に便利な小型のディスク状記録媒体の開発が進められている。

【0003】

従来から、このような記録媒体に対して映画やドラマ、音楽コンサートといった映像お

10

20

30

40

50

よび音声からなるコンテンツを収録したものが、セルパッケージとして販売されている。このようなセルパッケージにおいては、一つのコンテンツに対して言語の異なる複数の音声や字幕を収録することが可能である。一例として、DVDビデオ規格では、映像ストリームと、言語の異なる複数の音声ストリームおよび複数の字幕ストリームとが多重化され、1本のMPEG2 (Moving Pictures Experts Group 2)プログラムストリームが形成される。

【0004】

例えば、海外（例えばアメリカ）で制作された映画コンテンツに対し、原語（オリジナル言語）である英語による音声ストリームおよび日本語の吹き替えによる音声ストリームによる言語の異なる複数の音声ストリームと、英語表記の字幕ストリームおよび日本語表記の字幕ストリームによる言語の異なる複数の字幕ストリームとが収録される。

10

【0005】

DVDビデオ規格では、このように、言語の異なる複数の音声ストリームおよび複数の字幕ストリームが収録されている場合、再生させるストリームをユーザが選択できるようにされている。例えば、音声ストリームは、日本語による吹き替え版を選択し、字幕ストリームは、英語版を選択するようにできる。別の例として、音声ストリームは、日本語による吹き替え版を選択し、字幕ストリームを表示しないように選択することができる。

【0006】

また、DVDプレーヤ装置では、初期設定で優先して選択すべき言語が指定されており、ディスクの再生時に、その言語指定情報を参照して、ユーザによる指定が無くても再生させる音声および字幕ストリームを自動的に選択する機能を有している。さらにまた、プレーヤ装置の初期設定以外にも、コンテンツの再生歴や再生経路によって、次に再生するストリームを自動的に選択する機能も、DVDプレーヤには備わっている。

20

【0007】

ここで、DVDビデオ規格に基づく従来の音声ストリーム選択機能について、概略的に説明する。上述したように、一般的にDVDプレーヤ装置は、優先的に選択および再生する音声を、初期設定で指定することができる。例えば、日本人が使用するDVDプレーヤ装置であれば、当該プレーヤ装置の初期設定において、音声を「日本語」の設定にすることが多い。このような設定にすることで、例えばアメリカの映画を再生する場合でも、フランスの映画を再生する場合でも、そのディスクに日本語音声による音声ストリームが収録されていれば、その日本語による吹き替え版の音声が優先的に自動選択される。

30

【0008】

一方、常にそのコンテンツの制作時のオリジナルの言語で音声を聞きたいと考えるユーザもいる。例えば、音声は、コンテンツの制作時のオリジナルの言語で再生し、字幕は、当該ユーザの母国語で翻訳したものを表示させて、映画コンテンツなどを視聴したいと考えているユーザは、多いと思われる。一例として、ユーザが日本人でありアメリカで制作された映画コンテンツを視聴する場合に、音声をコンテンツの制作時のオリジナル言語である英語で再生し、字幕をユーザの母国語である日本語で表示させる。このように音声および言語を選択して視聴することにより、コンテンツの再生時に、よりオリジナルに近い雰囲気を楽しむことができる。

40

【0009】

特許文献1には、再生する音声言語を選択し字幕を表示しないモードと、再生する音声言語をディスクのデフォルトに設定し字幕言語を選択するモードと、音声、字幕およびメニュー言語をそれぞれ選択するモードとを提供し、言語設定を容易に行うことができるようにした再生装置が開示されている。

【特許文献1】特開2003-46951号公報

【発明の開示】

【発明が解決しようとする課題】

【0010】

しかしながら、従来のDVDビデオ規格では、収録されている音声ストリームがオリジ

50

ナルの言語によるものであるか否かを表す仕組みが存在しなかった。そのため、ディスクの再生時にオリジナルの言語を自動的に選択するように設定することができず、ユーザは、再生するディスク毎に、音声の設定を変更するしかなかったという問題点があった。

【0011】

例えば、オリジナル言語が日本語以外（例えばフランス語とする）のコンテンツが収録されたディスクを再生する際には、当該ディスクに収録されているコンテンツのオリジナル言語が何語であるかを、パッケージ印刷や解説文などから探し出す必要があった。そして、オリジナル言語が何語であるかが判明した後に、そのオリジナル言語（フランス語）が選択されるまで、プレーヤ装置の音声切り換えボタンを何度も押す必要があり、手間がかかっていた。

10

【0012】

この場合でも、例えばプレーヤ装置の初期設定で、再生される音声をフランス語に予め設定しておけば、当該ディスクの再生に限って、再生音声としてオリジナルの言語であるフランス語が自動的に選択されることになる。しかしながら、オリジナル言語が他の言語、例えば英語のディスクを再生する際には、ユーザがプレーヤ装置の初期設定を一々変更しなければならず、非常に不便であった。

【0013】

したがって、この発明の目的は、ディスクの再生時に、音声および字幕を適切に自動選択することができる再生装置、再生方法および再生プログラム、ならびに、記録媒体を提供することにある。

20

【課題を解決するための手段】

【0014】

この発明は、上述した課題を解決するために、円盤状記録媒体に記録されたコンテンツデータを再生する再生装置において、少なくとも、ビデオストリーム、ビデオストリームにそれぞれ対応する1または複数の音声ストリームおよび1または複数の字幕ストリームを含むコンテンツデータと、コンテンツデータの再生を制御する再生制御プログラムとが記録された記録媒体からデータを読み出す読み出し手段と、再生制御プログラムに従いコンテンツデータを再生するプレーヤ手段と、プレーヤ手段に対してコンテンツデータの再生時に1または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択する第1のモードを設定する第1のモード設定手段とを備えることを特徴とする再生装置である。

30

【0015】

また、この発明は、円盤状記録媒体に記録されたコンテンツデータを再生する再生方法において、少なくとも、ビデオストリーム、ビデオストリームにそれぞれ対応する1または複数の音声ストリームおよび1または複数の字幕ストリームを含むコンテンツデータと、コンテンツデータの再生を制御する再生制御プログラムとが記録された記録媒体からデータを読み出す読み出しのステップと、再生制御プログラムに従いコンテンツデータを再生するコンテンツ再生のステップと、コンテンツ再生のステップに対してコンテンツデータの再生時に1または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択するモードを設定するモード設定のステップとを備えることを特徴とする再生方法である。

40

【0016】

また、この発明は、円盤状記録媒体に記録されたコンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムにおいて、再生方法は、少なくとも、ビデオストリーム、ビデオストリームにそれぞれ対応する1または複数の音声ストリームおよび1または複数の字幕ストリームを含むコンテンツデータと、コンテンツデータの再生を制御する再生制御プログラムとが記録された記録媒体からデータを読み出す読み出しのステップと、再生制御プログラムに従いコンテンツデータを再生するコンテンツ再生のステップと、コンテンツ再生のステップに対してコンテンツデータの再生時に1または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択するモードを

50

設定するモード設定のステップとを備えることを特徴とする再生プログラムである。

【0017】

また、この発明は、少なくとも、ビデオストリームと、ビデオストリームにそれぞれ対応する1または複数の音声ストリームおよび1または複数の字幕ストリームを含むコンテンツデータと、コンテンツデータの再生を制御する再生制御プログラムと少なくとも1または複数の音声ストリームのそれぞれを特定する情報が、オリジナル言語として用いる音声ストリームを特定する情報を最も先頭側に配して記述されるストリーム情報とが記録されることを特徴とする記録媒体である。

【0018】

上述したように、この発明は、記録媒体には、少なくとも、ビデオストリーム、ビデオストリームにそれぞれ対応する1または複数の音声ストリームおよび1または複数の字幕ストリームを含むコンテンツデータと、コンテンツデータの再生を制御する再生制御プログラムとが記録され、再生装置において、記録媒体から再生された再生制御プログラムに従いコンテンツデータを再生する際に、1または複数の音声ストリームからオリジナル言語による音声ストリームを自動的に選択するモードを設定するようにしているため、ユーザは、コンテンツで用いられているオリジナル言語を調べて一々再生装置に対して設定しなくても、コンテンツをオリジナル言語により再生させ楽しむことができる。

10

【0019】

また、この発明は、少なくとも、ビデオストリームと、ビデオストリームにそれぞれ対応する1または複数の音声ストリームおよび1または複数の字幕ストリームを含むコンテンツデータと、コンテンツデータの再生を制御する再生制御プログラムと少なくとも1または複数の音声ストリームのそれぞれを特定する情報が、オリジナル言語として用いる音声ストリームを特定する情報を最も先頭側に配して記述されるストリーム情報とが記録媒体に記録されるため、この記録媒体を再生する再生装置は、ストリーム情報中の音声ストリームを特定する情報の並び順を調べることで、オリジナル言語として用いる音声ストリームを特定することができる。

20

【0020】

またその際、オリジナル言語として用いる音声ストリームや当該音声ストリームを特定する情報に対して当該音声ストリームがオリジナル言語である旨を示す情報を付加しなくても、オリジナル言語として用いる音声ストリームを特定することができる。

30

【発明の効果】

【0021】

上述したように、この発明によれば、言語が異なる複数の音声ストリームおよび字幕ストリームを、ディスクの再生時に適切に自動選択できるようになる効果がある。

【0022】

また、この発明の実施の一形態によれば、プレーヤの音声に関する初期設定として、「オリジナル言語」属性を指定できるようにしているため、コンテンツが制作された最初の言語を自動的に選択できるようになり、オリジナルの雰囲気損なわずにコンテンツを楽しむようになる効果がある。

40

【0023】

さらに、この発明の実施の一形態によれば、音声ストリームおよび字幕ストリームの自動選択の結果、音声と字幕とで同一言語が選択されてしまった場合でも、字幕ストリームを自動的に非表示にできるようにしている。そのため、ユーザは、不要な字幕表示を手動で非表示にする必要が無く、使い勝手が向上する効果がある。

【発明を実施するための最良の形態】

【0024】

以下、この発明の実施の一形態について、下記の順序に従い説明する。

1. UMDビデオ規格について
2. UMDビデオ規格のプレーヤモデルについて
3. ムービープレーヤのイベントモデルについて

50

4. ムービープレーヤオブジェクトについて
5. スクリプトプログラムの例
6. ファイルの管理構造について
7. ディスク再生装置について
8. 音声および字幕ストリームの自動選択について

【0025】

1. UMDビデオ規格について

先ず、理解を容易とするために、この実施の一形態に適用可能なシステムについて概略的に説明する。この発明の実施の一形態では、ECMAスクリプトと呼ばれるスクリプト言語を用いてプレーヤモデルを記述している。ECMAスクリプトは、ECMA (European Computer Manufacturers Association)により定められた、JavaScript (登録商標) に基づいたクロスプラットフォーム用のスクリプト言語である。ECMAスクリプトは、HTML文書との親和性が高いことと、独自のオブジェクトの定義が可能であるため、この発明によるプレーヤモデルに用いて好適である。

【0026】

また、以下では、このECMAスクリプトを元にしたスクリプト言語を用いた、この発明の実施の一形態に基づく規格を、UMD (Universal Media Disc: 登録商標) ビデオ規格と呼ぶ。また、UMDビデオ規格のうち、特にスクリプトに関する部分をUMDビデオスクリプト規格と呼ぶ。

【0027】

UMDビデオ規格について、概略的に説明する。図1は、UMDビデオ規格のレイヤ構成を示す。UMDビデオ規格では、スクリプトレイヤ、プレイリストレイヤおよびクリップレイヤの3層のレイヤ構造が定義され、この構造に基づきストリーム管理がなされる。

【0028】

UMDビデオ規格においては、デジタル符号化されたビデオ、オーディオおよび字幕を、MPEG2 (Moving Pictures Experts Group 2) のエレメンタリストリームとして多重化したMPEG2ストリームとして扱う。このビデオ、オーディオおよび字幕のエレメンタリストリームが多重化されたMPEG2ストリームを、クリップAVストリーム (Clip AV Stream) と呼ぶ。クリップAVストリームは、クリップAVストリームファイルに格納される。クリップAVストリームファイルの記録時に、当該クリップAVストリームファイルに1対1に対応して、クリップインフォメーションファイル (Clip Information File) が同時に作成される。これらクリップインフォメーションファイルと、対応するクリップAVストリームファイルとからなる組を、クリップ (Clip) と呼ぶ。

【0029】

クリップは、ディスクへの記録の単位ともいふべきものであり、再生時にどのような順序でクリップを再生するかは、クリップの上位のレイヤであるプレイリストレイヤで管理する。プレイリストレイヤは、クリップの再生経路を指定するレイヤであり、1または複数のプレイリスト (PlayList) を含む。プレイリストは、プレイアイテムの (PlayItem) の集合からなる。プレイアイテムには、クリップの再生範囲を示した一組のイン (In) 点およびアウト (Out) 点が含まれており、プレイアイテムを連ねることによって、任意の順序でクリップを再生することができるようになる。プレイアイテムは、クリップを重複して指定することができる。クリップAVストリームファイルのイン点およびアウト点は、タイムスタンプ (クリップ内時刻) で指定され、タイムスタンプは、クリップインフォメーションファイルの情報によってクリップAVストリーム上のバイト位置に変換される。

【0030】

プレイリストは、クリップの全部または一部を指すプレイアイテムを順序に従って再生していく構造だけを有しており、プレイリストのみを用いて再生順の分岐や、ユーザとの双方向性を実現することは、できない。この発明の実施の一形態では、複数のプレイリストが1つのファイル "PLAYLIST.DAT" にまとめられている。

【0031】

10

20

30

40

50

スクリプトレイヤは、言語仕様のECMAスクリプトを拡張した、UMDビデオスクリプトによって構築されるレイヤである。UMDビデオスクリプトは、ECMAスクリプトを基本として、UMDビデオに特有な機能を実現するための拡張を加えたスクリプトである。

【0032】

スクリプトレイヤは、プレイリストレイヤの上位のレイヤであり、プレイリストの再生指示や、プレーヤ設定を行うコマンド列から構成される。スクリプトレイヤのコマンドにより、複数の言語用に用意されたストリームのうち何れを選択する、ある条件に従って選択されるプレイリストに再生の流れが変化する、というような、条件分岐を伴うプレイリスト再生を実現することができる。このような条件分岐を伴うプレイリスト再生が用いられるアプリケーションの例としては、マルチストーリーが挙げられる。このスクリプトレイヤにより、ユーザとの双方向性機能（インタラクティブ機能）が導入されることになる。

10

【0033】

なお、この発明の実施の一形態では、スクリプトレイヤは、1つのファイル"SCRIPT.DAT"から構成され、リソースとして管理される。ファイル"SCRIPT.DAT"は、実際のECMAスクリプトに基づき記述されるスクリプトデータ、ボタン操作の際の効果音などを出力するためのサウンドデータ、メニュー画面の背景画像などに用いる画像データからなるスクリーンデザイン、ならびに、ボタン画像などのGUI部品を表示させるための画像データ（ビットマップデータ）が含まれる。

20

【0034】

2. UMDビデオ規格のプレーヤモデルについて

次に、UMDビデオ規格に従ったデータを再生する再生装置（プレーヤ）のモデル、すなわち、プレーヤモデルについて説明する。プレーヤは、先ず、ディスクからスクリプトプログラム、プレイリストおよびクリップインフォメーションファイルを読み出し、次に、これらにより定められている再生順序に従って、クリップAVストリームファイルを読み出し、ビデオ、オーディオおよび字幕などを再生する。

【0035】

スクリプトプログラムの言語仕様においては、プレイリストを再生する機能ブロックを、スクリプトプログラム内のオブジェクトとして実装する。このプレイリスト再生を行うオブジェクトを、UMDビデオ規格では、ムービープレーヤ(Movie Player)オブジェクトと呼ぶ。プレイリストの再生指示や、プレーヤ設定を行うコマンドは、このムービープレーヤオブジェクトが有するメソッドとなる。ムービープレーヤオブジェクトは、スクリプトレイヤからのメソッドによって制御される。このとき、ムービープレーヤオブジェクトからスクリプトレイヤに対して、状態の変化や再生位置などを通知する機能が必要となる。これは、ムービープレーヤオブジェクトがスクリプトプログラムに対してイベントを発生することに対応し、そのイベントに対応した処理は、イベントハンドラとして記述される。

30

【0036】

このように、ムービープレーヤオブジェクトからスクリプトプログラムへの情報伝達は、イベントにより行い、スクリプトプログラムからムービープレーヤオブジェクトに対する制御をメソッドにより行うモデルを構築することにより、クリップAVストリームの再生をスクリプトプログラムで制御できるようになる。

40

【0037】

図2は、上述した、この発明の実施の一形態による一例のプレーヤモデルを模式的に示す。ムービープレーヤ300は、UMDビデオ規格においてビデオ、オーディオおよび字幕の再生を司るモジュールである。上述したムービープレーヤオブジェクトは、ムービーオブジェクトをスクリプトプログラムから操作するためにスクリプトプログラム内のオブジェクトとしたものである。換言すれば、ムービープレーヤオブジェクトは、ムービープレーヤの機能を実現するためのスクリプトプログラムそのものである。

50

【 0 0 3 8 】

なお、ムービープレーヤ 3 0 0 とムービープレーヤオブジェクトは、実質的に同一の対象を表すと考えられるので、以下、これらを同一の符号を付して説明する。

【 0 0 3 9 】

図 2 において、ムービープレーヤ 3 0 0 は、ユーザ入力 3 1 0 などにより引き起こされる下位レイヤ（図 2 の例ではネイティブ実装プラットフォーム 3 0 1）や、上位レイヤであるスクリプトレイヤ 3 0 2 からのメソッドに従って、プレイリストおよびクリップインフォメーションのデータベースに基づき、クリップ A V ストリームファイルの読み出し、読み出されたクリップ A V ストリームのデコードおよび表示を行う。

【 0 0 4 0 】

ムービープレーヤオブジェクト 3 0 0 の内部は、UMD ビデオを再生する UMD ビデオプレーヤの実装に依存するものであって、スクリプトレイヤ 3 0 2 からは、ブラックボックス化されたオブジェクトとして、メソッドやプロパティといった A P I (Application Programming Interface) が提供される。ここで、UMD ビデオプレーヤは、ムービープレーヤを実装した実際の機器を指す。全ての UMD ビデオプレーヤは、UMD ビデオ規格の制約を守ってムービープレーヤを実装しており、再生互換を有する。

【 0 0 4 1 】

図 2 に示されるように、ムービープレーヤオブジェクト 3 0 0 は、ネイティブ実装プラットフォーム 3 0 1 からの制御コマンド 3 1 1 を受け付けるパス、スクリプトレイヤ 3 0 2 に対してイベント 3 1 2 を通知するパス、スクリプトレイヤ 3 0 2 からのメソッド 3 1 3 を受け付けるパスの、3 本の入出力パスを有する。

【 0 0 4 2 】

制御コマンド 3 1 1 は、ネイティブ実装のプラットフォーム 3 0 1 からの、ムービープレーヤオブジェクト 3 0 0 の動作を制御するコマンドである。ネイティブ実装プラットフォーム 3 0 1 は、例えば、実際の機器としての UMD ビデオプレーヤにおける、機器に固有の部分とムービープレーヤ 3 0 0 とのインターフェイスである。イベント 3 1 2 は、ムービープレーヤ 3 0 0 からスクリプトレイヤ 3 0 2 に対するスクリプトイベントである。メソッド 3 1 3 は、スクリプトレイヤ 3 0 2 のスクリプトプログラムからムービープレーヤ 3 0 0 に指示されるメソッドである。

【 0 0 4 3 】

ムービープレーヤオブジェクト 3 0 0 は、内部に、UMD ビデオ規格のプレイリストおよびクリップインフォメーションのデータベース 3 2 0 を有する。ムービープレーヤオブジェクト 3 0 0 は、このデータベース 3 2 0 を用いて、ユーザ入力 3 1 0 に対する無効化 (mask) や、時刻で指定された再生位置をクリップ A V ストリーム内のバイト位置に変換する処理などを行う。

【 0 0 4 4 】

ムービープレーヤオブジェクト 3 0 0 内のプレイバックモジュール 3 2 1 は、ビデオ、オーディオおよび字幕が多重された M P E G 2 P S (Program Stream) であるクリップ A V ストリームのデコードを行う。プレイバックモジュール 3 2 1 は、プレイ、ストップおよびポーズの 3 状態を持ち、制御命令やメソッドによって、この 3 状態の間を遷移する（図 3 参照）。

【 0 0 4 5 】

スクリプトレイヤ 3 0 2 は、UMD ビデオスクリプト規格に基づくスクリプトプログラムを実行し、ムービープレーヤオブジェクト 3 0 0 の制御や、画面表示を行うレイヤである。このスクリプトレイヤ 3 0 2 は、コンテンツ制作者側の意図したシナリオを実現する役割を果たす。スクリプトレイヤ 3 0 2 は、ムービープレーヤオブジェクト 3 0 0 に対してメソッド 3 1 3 を発行し、ムービープレーヤオブジェクト 3 0 0 からは、イベント 3 1 2 を受け取る。スクリプトレイヤ 3 0 2 は、ネイティブ実装プラットフォーム 3 0 1 との間で、ユーザ入力 3 1 0 に応じたキーイベント 3 1 4 や、画面描画などをネイティブ実装プラットフォーム 3 0 1 に対して指示するメソッド 3 1 5 などのやりとりを行う。

10

20

30

40

50

【0046】

例えば、メニュー画面上に配置されるボタンは、スクリプトレイヤ302のスクリプトプログラムからネイティブ実装プラットフォーム301に渡されるメソッド315に基づき、ネイティブ実装プラットフォーム301により描画される。ユーザがそのボタンに対して選択や決定などの操作を行ったときには、ユーザ入力310に応じたキーイベント314がネイティブ実装プラットフォーム301からスクリプトレイヤ302に通知され、スクリプトレイヤ302内のスクリプトプログラムは、キーイベント314に基づきキー入力310に応じた処理を行う。

【0047】

このように、ビデオ、オーディオおよび字幕のデコードや表示制御はムービープレーヤ300が司り、ボタンなどのGUI (Graphical User Interface)を構成するための部品画像(以下、GUI部品と呼ぶ)の配置や表示、ならびに、GUI部品に対して選択や決定などの操作がなされたときの処理は、スクリプトレイヤ302が行うというように、役割分担がなされている。

【0048】

ネイティブ実装プラットフォーム301は、ムービープレーヤオブジェクト300やスクリプトプログラムが動作するための基盤となるプラットフォームであって、例えば、実際のUMDビデオプレーヤがハードウェアである場合、ハードウェアとプレーヤモデルとの間の処理を仲介する役割を果たすように、ハードウェアに固有に実装される。

【0049】

例えば、ネイティブ実装プラットフォーム301は、ユーザからのユーザ入力310を受け付け、受け付けたユーザ入力310がムービープレーヤ300に対する命令なのか、スクリプトレイヤ302で描画および表示しているボタンに対する命令なのかを判定する。ネイティブ実装プラットフォーム301は、ユーザ入力310がムービープレーヤ300に対する命令であると判定されれば、ユーザ入力310をムービープレーヤ300に対する内部制御命令である制御コマンド311に変換し、ムービープレーヤ300に対して制御命令を発する。

【0050】

一方、ネイティブ実装プラットフォーム301は、ユーザ入力310がスクリプトレイヤ302で描画および表示しているGUI部品に対する命令であると判定されれば、ユーザ入力310に応じたキーイベント314をスクリプトレイヤ302に通知する。そして、このキーイベント314に応じてスクリプトレイヤ302から指示されたメソッド315に基づき、例えば画面上にボタン画像を表示させることができる。すなわち、ネイティブ実装プラットフォーム301とスクリプトレイヤ302とは、ムービープレーヤ300を介さずに、直接的にイベントおよびメソッドの受け渡しを行うことができる。

【0051】

次に、ムービープレーヤ300についてより詳細に説明する。図3は、ムービープレーヤ300の一例の内部構成を示す。上述したように、ムービープレーヤ300は、データベース320およびプレイバックモジュール321とから構成される。データベース320は、ディスクから読み取ったプレイリストの情報と、クリップの情報すなわちクリップインフォメーションとを格納する領域である。

【0052】

プレイバックモジュール321は、デコーダエンジン322と、プレイバックモジュール321の状態を表す値であるプロパティ323とからなる。プロパティ323は、例えば言語コードのように、ムービープレーヤ300の初期設定で値が決まるプロパティ323A(リードオンリーパラメータ)と、プレイバックモジュール321の状態によって値が変化するプロパティ323B(プレーヤステータス)の2種類がある。

【0053】

初期設定で値が決まるプロパティ323Aは、ネイティブなシステム、例えば実際の機器によって値がセットされ、プレイリストやクリップインフォメーション、スクリプトプ

10

20

30

40

50

プログラムから値を変更されることがない。プロパティ 3 2 3 A は、スクリプトプログラムから値を読み出すことのみが可能とされる。一方、プレイバックモジュール 3 2 1 の状態を表すプロパティ 3 2 3 B は、スクリプトプログラムから値を読み出すことができると共に、一部のスクリプトプログラムから値を書き込むことが可能とされる。

【 0 0 5 4 】

なお、この動作モデルにおいては、プレイリストおよびクリップインフォメーションは、クリップ A V ストリームの再生前にディスクからプリロードされていることを想定している。これに限らず、他の実装であっても、ムービープレーヤモデルで定めた動作を実現できていればよい。

【 0 0 5 5 】

ムービープレーヤオブジェクト 3 0 0 は、スクリプトレイヤ 3 0 2 またはネイティブ実装プラットフォーム 3 0 1 からの指示に従い、指定されたプレイリストを再生する。例えば、ムービープレーヤ 3 0 0 は、データベース 3 2 0 を参照し、指定されたプレイリストに対応するクリップ A V ストリームの再生位置をファイル中のバイト位置で得る。プレイバックモジュール 3 2 1 において、デコーダエンジン 3 2 2 は、この再生位置情報に基づき、クリップ A V ストリームのデコードを制御する。

【 0 0 5 6 】

ムービープレーヤ 3 0 0 は、図 4 に示されるように、プレイリストの再生状況に応じてプレイ(play)、ストップ(stop)およびポーズ(pause)の 3 状態を持つ。プレイ状態は、プレイリストの再生を行っており、時間が経過している状態を指す。通常再生の他、2 倍速、1 / 2 倍速といった変速再生や、順方向早送りおよび逆方向早送りがプレイ状態に含まれる。ポーズ状態は、プレイリストの再生を行っている状態で、時間軸が停止している状態である。再生をフレーム単位で進めたり戻したりする所謂コマ送り再生は、ポーズ状態とプレイ状態とを繰り返している状態である。ストップ状態は、プレイリストを再生していない状態である。

【 0 0 5 7 】

例えば、ムービープレーヤ 3 0 0 の状態は、ムービープレーヤ 3 0 0 内のデコーダエンジン 3 2 2 におけるプレイ、ポーズおよびストップの状態遷移に伴うもので、デコーダエンジン 3 2 2 の状態遷移に応じてプロパティ 3 2 3 B の値が更新される。

【 0 0 5 8 】

リジュームインフォメーション 3 2 4 は、ストップ状態直前の状態を記憶する。例えば、ムービープレーヤ 3 0 0 があるプレイリストをデコードしプレイ状態となっているときに、状態がストップ状態に遷移すると、ストップ状態の直前の状態を記憶する。また、リジュームインフォメーション 3 2 4 は、ハードウェアとしてのプレーヤが持つ不揮発性メモリに、ディスクのタイトル毎に識別可能なように複数を記憶させることができる。例えば、ディスクは、ディスクのタイトル毎にユニークな識別情報(タイトル ID と呼ぶ)を有し、リジュームインフォメーション 3 2 4 をこの識別情報と関連付けて記憶する。こうすることで、リジュームインフォメーション 3 2 4 の情報に基づき、識別情報が対応するタイトルのディスクが次にストップ状態からプレイ状態に遷移したときに、当該ディスクの再生を、以前ストップ状態になった直前の位置から開始させることができる。

【 0 0 5 9 】

3 . ムービープレーヤのイベントモデルについて

ムービープレーヤ 3 0 0 のイベントモデルについて説明する。ムービープレーヤ 3 0 0 は、プレイリストを再生するプレイ状態で、様々なイベントを発生する。このイベントは、イベントハンドラと呼ばれる、スクリプトで記述された処理プログラムの実行を引き起こす。イベントハンドラは、イベント発生によって呼び出されるメソッドである。このイベント発生によって処理プログラムの実行を開始するプログラム実行モデルを、イベントドリブンモデルと呼ぶ。イベントドリブンモデルでは、不定期なイベントが発生し、イベント発生をきっかけに用意しておいたプログラムが実行される。この実施の一形態においては、スクリプトプログラムは、イベントハンドラ群によってムービープレーヤオブジェ

10

20

30

40

50

クト300の動作を制御する。

【0060】

図5は、この発明の実施の一形態によるムービープレーヤ300のイベントモデルを模式的に示す。図5において、イベントハンドラonEventA()、onEventB()およびonEventC()は、インターフェイスであって、それぞれのイベントハンドラの内容は、スクリプトで記述されている。イベントハンドラの内容は、例えばコンテンツ制作者側で作成され実装される。UMDビデオスクリプト規格においては、ムービープレーヤオブジェクト300からスクリプトプログラムに通知されるイベント毎に、イベントハンドラが用意される。図5の例では、イベントAが発生したときに実行される処理プログラムは、イベントハンドラonEventA()に決められている。イベントBおよびイベントCについても同様に、イベントBの発生時には対応するイベントハンドラonEventB()が実行され、イベントCの発生時には対応するイベントハンドラonEventC()が実行される。

10

【0061】

イベント発生に応じて呼び出されるイベントハンドラは、システム側で選択されるため、コンテンツ制作者側では、どのイベントが発生したかを判断する処理を、スクリプトプログラム内に記述しておく必要が無い。

【0062】

図6は、プレイリストの再生中に発生する一例のイベントを示す。プレイリストPlayListの先頭には、チャプタマークChapterMarkが設定されているので、プレイリストの先頭からの再生開始時には、先ず、チャプタマークに対応したイベントChapterが発生する。さらに、チャプタが変わる度にイベントChapterがスクリプトレイヤ302に通知され、対応するイベントハンドラonChapterが実行される。また、イベントマークEventMarkが設定されている時刻に再生が到達すると、対応するマークイベントが発生する。そして、プレイリストの最後まで再生が到達すると、プレイリストの最後で再生が一時停止し、イベントPlayListEndがムービープレーヤ300からスクリプトレイヤ302に通知される。スクリプトレイヤ302側では、対応するイベントハンドラonPlayListEnd()内で、別のプレイリストの再生開始が指示される。このようにして、コンテンツ制作者が意図した順序で、一連のプレイリスト再生が継続されていく。

20

【0063】

このように、プレーヤの動作中にはさまざまなイベントが発生するものとし、イベント発生を上位プログラムに伝えることで、上位プログラムはプレーヤの状態を把握できるようになる。上位プログラムの方では、各イベント発生通知時に実行されるプログラム(イベントハンドラ)を用意しておくことで、各種イベント発生に対処する。イベントおよびイベントハンドラの詳細については、後述する。

30

【0064】

イベントハンドラがコンテンツ制作者によって記述されていない場合には、規格で規定されているプレーヤ組み込みの動作(デフォルトのイベントハンドラ)を実行するか、あるいは、そのイベントが無視され何も実行されない。何も処理を行う必要がないときには、イベントに対応したイベントハンドラを記述しないようにすることで、積極的にイベントを無視することができる。

40

【0065】

イベントモデルとしては、上述の他に、あるイベントに対応するリスナをオブジェクトがプレーヤオブジェクトに登録し、プレーヤオブジェクト内で発生したイベントが登録されたイベントであれば、プレーヤオブジェクトから当該イベントを登録したオブジェクトにイベントを送信し、当該オブジェクトで対応するメソッドを実行するようにしたイベントリスナのモデルや、どのようなイベントが発生しても一つのメソッドを呼び出すようにした単一メソッドのモデルなどが考えられる。

【0066】

この実施の一形態によるイベントモデルは、イベント登録、イベント登録削除といった処理が必要なイベントリスナのモデルよりも簡単である。また、単一メソッドのモデルは

50

、どのイベントが発生したかを知り、イベント毎に用意してある処理ルーチンを切り替えるという前処理を、そのメソッドの中に記述しておく必要がある。メソッドは、コンテンツ制作者側が実装するものであるから、モデルとしては簡単でも、コンテンツ制作者側の負担が大きくなる。さらに、大きな一つの処理プログラム（メソッド）がイベントの発生毎に呼ばれることになり、メモリの領域を多く占有し、実行速度も遅くなると考えられる。この発明の実施の一形態による、イベント毎に処理プログラム（イベントハンドラ）を用意するモデルでは、このような点について有利であるといえる。

【0067】

4. ムービープレーヤオブジェクトについて

次に、ムービープレーヤオブジェクト300の外部的な仕様について説明する。一般に、ECMAScript言語仕様に従う言語により定義されたオブジェクトは、プロパティとメソッドとを持つ。この実施の一形態によるムービープレーヤオブジェクト300も、図2および図3を用いて既に説明したように、同様にプロパティとメソッドとを有する。プロパティは、外部のオブジェクトから、対象となるオブジェクト名とプロパティ名とを指定することで、直接的に読み書きすることが可能である。これに限らず、プロパティ値の設定を行うメソッドsetXXX()（「XXX」は、対象のプロパティ名）や、プロパティ値の読み出しを行うメソッドgetXXX()を定義することで、他のオブジェクトのプロパティの読み書きを、メソッドで行うことが可能となる。

【0068】

図7は、ムービープレーヤオブジェクト300が有する一例のプロパティを一覧して示す。これは、図3におけるプロパティ323に対応する。図3におけるリードオンリーパラメータ323Aに属するプロパティは、以下の通りである。プロパティscriptVersionは、UMDビデオスクリプトのバージョンを示す。プロパティlanguageCodeは、UMDビデオプレーヤに設定された、メニュー表示言語の言語コードを示す。プロパティaudioLanguageCodeは、UMDビデオプレーヤに設定された、オーディオ言語の言語コードを示す。プロパティsubtitleLanguageCodeは、UMDビデオプレーヤに設定された、字幕（サブタイトル）言語の言語コードを示す。

【0069】

ディスクが装填された際には、このリードオンリーパラメータ323Aに設定されたプロパティlanguageCodeに示される言語コードに基づき、ディスクから読み出すスクリプトファイルが決められる。装填されたディスクに、当該言語に対応するスクリプトファイルがない場合は、デフォルトのスクリプトファイルが読み出される。例えば、複数のスクリプトファイルのうち、ディスク上で最も先頭側に配置されるファイルがデフォルトのスクリプトファイルとして読み出される。

【0070】

図3におけるプレーヤステータス323Bに属するプロパティは、以下の通りである。プロパティplaylistNumberは、現在再生中のプレイリストの番号を示す。プロパティchapterNumberは、現在再生中のチャプタの番号を示す。プロパティvideoNumberは、現在再生中のビデオストリームの番号を示す。プロパティaudioNumberは、現在再生中のオーディオストリームの番号を示す。プロパティsubtitleNumberは、現在再生中の字幕ストリームの番号を示す。プロパティplaylistTimeは、プレイリスト先頭を0としたときの時刻を示す。プロパティaudioFlagは、オーディオ再生のON/OFFおよびデュアルモノLRの指定を示す。プロパティsubtitleFlagは、字幕表示のON/OFFを示す。

【0071】

なお、デュアルモノは、ステレオオーディオの左右（L、R）チャンネルを、互いに独立したモノラルオーディオチャンネルとして用いるモードである。

【0072】

このプレーヤステータス323Bに属する各プロパティは、ムービープレーヤ300が再生または一時停止状態のときに、これらの情報が存在する。停止状態に遷移した場合、その時点でプレーヤステータス323Bに属する各プロパティは、リジュームインフォメ

ーション 3 2 4 としてバックアップされる。このとき、プレーヤステータス 3 2 3 B の内容をクリアしてもよい。

【 0 0 7 3 】

図 8 は、ムービープレーヤオブジェクト 3 0 0 が有する一例のメソッドを一覧して示す。これは、図 2 におけるメソッド 3 1 3 に対応する。メソッド play() は、ビデオを再生する。メソッド playChapter() は、チャプタを指定してビデオを再生する。メソッド stop() は、ビデオの再生を停止する。メソッド pause() は、ビデオの再生を一時停止する。メソッド playStep() は、ビデオをコマ送り再生する。メソッド changeStream() は、ビデオストリーム、オーディオストリームおよび / または字幕ストリームを変更する。メソッド getPlayerStatus() は、ムービープレーヤ 3 0 0 における再生、停止、一時停止などの状態を取得する。メソッド reset() は、ビデオの再生を停止し、リジュームインフォメーション 3 2 4 の内容をクリアする。

10

【 0 0 7 4 】

UMD ビデオ規格では、表示画面上の一部分にビデオを表示することができるようになっている。以下の 4 つのメソッドは、この場合のビデオ表示に関するメソッドである。メソッド setPos() は、ビデオの表示位置を設定する。メソッド getPos() は、ビデオの表示位置を取得する。メソッド setSize() は、ビデオの表示サイズを設定する。メソッド getSize() は、ビデオの表示サイズを取得する。

【 0 0 7 5 】

なお、実際には、ムービープレーヤ 3 0 0 とネイティブ実装プラットフォーム 3 0 1 とは、一体的に構成される。すなわち、実際にディスクが装填され、これを再生するハードウェアとしての UMD プレーヤと、UMD プレーヤを制御するソフトウェアとの関係に対応付けられ、どの部分をハードウェアで行い、どの部分をソフトウェアで行うかは、実装時の構成に依存する。例えば、UMD プレーヤをパーソナルコンピュータなどで構成する場合は、ディスクドライブ以外は、ソフトウェア的に構成することができる。また、単体の UMD プレーヤとして構成する場合は、ディスクドライブ以外に、例えばビデオデコーダやオーディオデコーダなどをハードウェア的に構成することができる。したがって、ムービープレーヤ 3 0 0 とネイティブ実装プラットフォーム 3 0 1 との間でなされるメソッドやコマンド、イベントは、図 2 に一例が示されるような明示的なやりとりに限られない。

20

30

【 0 0 7 6 】

一方、ユーザからのキー入力については、図 2 を用いて既に説明したように、ユーザ入力 3 1 0 をネイティブ実装プラットフォーム 3 0 1 が先ず、受け取る。つまり、ネイティブ実装プラットフォーム 3 0 1 は、ユーザからのキー入力をユーザ入力 3 1 0 として受け取り、ユーザ入力 3 1 0 がムービープレーヤ 3 0 0 に対するコマンドなのか、スクリプトレイヤ 3 0 2 のスクリプトプログラムに対するイベントなのかを判定し、判定結果に応じて、制御コマンド 3 1 1 またはキーイベント 3 1 4 を発生し、対応する上位レイヤ (ムービープレーヤ 3 0 0 またはスクリプトレイヤ 3 0 2) に通知する。

【 0 0 7 7 】

図 9 および図 1 0 は、ユーザ入力 3 1 0 による一例のキー入力を示す。なお、図 9 および図 1 0 に示される「VK」で始まる各キーは、実装に依存しない抽象化した仮想的なキーである。図 9 は、ムービープレーヤ 3 0 0 の操作に関する一例のキー入力を示す。キー VK_POWER は、電源キーに対応する機能を提供する。キー VK_POWER_ON は、電源 ON キーに対応する機能を提供する。キー VK_POWER_OFF は、電源 OFF キーに対応する機能を提供する。キー VK_MENU は、メニューを表示させるメニューキーに対応する機能を提供する。キー VK_ENTER は、「決定」を指示する決定キーに対応する機能を提供する。キー VK_RETURN は、処理のステップを一つ戻す戻すキーに対応する機能を提供する。

40

【 0 0 7 8 】

キー VK_PLAY は、再生を指示する再生キーに対応する機能を提供する。キー VK_STOP は、再生の停止を指示する停止キーに対応する機能を提供する。キー VK_PAUSE は、再生の一時

50

停止を指示する一時停止キーに対応する機能を提供する。キーVK_FAST_FORWARDは、早送り再生を指示する早送りキーに対応する機能を提供する。キーVK_FAST_REVERSEは、早戻し再生を指示する早戻しキーに対応する機能を提供する。キーVK_SLOW_FORWARDは、順方向のスロー再生を指示するスロー（順方向）キーに対応する機能を提供する。キーVK_SLOW_REVERSEは、逆方向のスロー再生を指示するスロー（逆方向）キーに対応する機能を提供する。キーVK_STEP_FORWARDは、順方向のコマ送り再生を指示するコマ送り（順方向）キーに対応する機能を提供する。キーVK_STEP_REVERSEは、逆方向のコマ送り再生を指示するコマ送り（逆方向）キーに対応する機能を提供する。

【0079】

図10は、メニュー操作に関する一例のキー入力を示す。キーVK_NEXTは、「次」を意味する値を入力する次指定キーに対応する機能を提供する。キーVK_PREVIOUSは、「前」を意味する値を入力する前指定キーに対応する機能を提供する。例えば、キーVK_NEXTおよびキーVK_PREVIOUSを用いて、前後のチャプタへの移動を指示することができる。

10

【0080】

キーVK_UPは、「上」を意味する値を入力する上方向指定キーに対応する機能を提供する。キーVK_DOWNは、「下」を意味する値を入力する下方向指定キーに対応する機能を提供する。キーVK_RIGHTは、「右」を意味する値を入力する右方向指定キーに対応する機能を提供する。キーVK_LEFTは、「左」を意味する値を入力する左方向指定キーに対応する機能を提供する。キーVK_UP_RIGHTは、「右上」を意味する値を入力する右上方向指定キーに対応する機能を提供する。キーVK_UP_LEFTは、「左上」を意味する値を入力する左上方向指定キーに対応する機能を提供する。キーVK_DOWN_RIGHTは、「右下」を意味する値を入力する右下方向指定キーに対応する機能を提供する。キーVK_DOWN_LEFTは、「左下」を意味する値を入力する左下方向指定キーに対応する機能を提供する。これらの方向キーを用いることで、例えば画面上のカーソル表示の移動を指示することができる。

20

【0081】

キーVK_ANGLEは、マルチアングルのビデオに対するアングル切り替えを指示するアングル切り替えキーに対応する機能を提供する。キーVK_SUBTITLEは、英語字幕、日本語字幕、字幕表示/非表示などを切り替える字幕切り替えキーに対応する機能を提供する。キーVK_AUDIOは、サラウンドやバイリンガルなどオーディオ設定を切り替えるオーディオ切り替えに対応する機能を提供する。キーVK_VIDEO_ASPECTは、ビデオのアスペクト比切り替えを指示するアスペクト切り替えキーに対応する機能を提供する。キーVK_COLORED_KEY_1は、色つきファンクションキー1、キーVK_COLORED_KEY_2は、色つきファンクションキー2、キーVK_COLORED_KEY_3は、色つきファンクションキー3、キーVK_COLORED_KEY_4は、色つきファンクションキー4、キーVK_COLORED_KEY_5は、色つきファンクションキー5、キーVK_COLORED_KEY_6は、色つきファンクションキー6にそれぞれ対応する機能を提供する。

30

【0082】

上述の図9に示したキー入力と図10に示したキー入力とは役割が異なるため、通知先をネイティブ実装プラットフォーム301で振り分ける必要がある。上述したように、図9に示されるキー入力により、ビデオ、オーディオおよび字幕の再生に関する指示がなされる。ネイティブ実装プラットフォーム301は、ユーザ入力310として図9に示されるキー入力を受け取ると、受け取ったキー入力を、図11に示すコマンドに変換してムービープレーヤ300に通知する。

40

【0083】

一方、図10に示されるキー入力は、GUIに対するユーザ入力310であるので、このユーザ入力は、画面構成やボタンを配置するスクリプトレイヤ302に通知されて処理される必要がある。ネイティブ実装プラットフォーム301は、ユーザ入力310として図10に示されるキー入力を受け取ると、図2におけるイベント314に変換してスクリプトレイヤ302に通知する。図12は、このキー入力に対応する一例のイベント314を示す。

50

【 0 0 8 4 】

なお、上述した図 9 および図 1 0 には、キー VK_ANGLE、キー VK_SUBTITLE、キー VK_AUDIO という、ストリーム切り替えに関するキー入力も含まれているが、これらは、スクリプトプログラムからムービープレーヤ 3 0 0 に対するストリーム切り替えのメソッドと同様の機能を実現するためのキー入力である。

【 0 0 8 5 】

上述した図 1 1 のコマンドについて、より詳細に説明する。コマンド `uo_timeSearch(playListTime)` は、再生中のプレイリストの指定時刻からの再生を指示する。引数 `playListTime` は、プレイリストの先頭を 0 としたときの時刻を表す。このコマンドでは、プレイリスト番号の指定はできないため、引数 `playListTime` で表される時刻は、現在再生中のプレイリストの範囲内での指定時刻となる。コマンド `uo_play()` は、1 倍速での再生開始を指示する。開始位置は、リジュームインフォメーション 3 2 4 に基づき決められる。リジュームインフォメーション 3 2 4 に対応する情報が無い場合は、このユーザ操作は無効とされる。このコマンドは、プレイリスト番号の指定の無いメソッド `play()` を実行したときに対応する。また、このコマンドにおいて、ユーザ操作ではプレイリスト番号を指定できない。

【 0 0 8 6 】

コマンド `uo_playChapter(chapterNumber)` は、再生中のプレイリストの、引数 `chapterNumber` で指定されたチャプタからの再生開始を指示する。チャプタの指定がない場合には、現在再生中のチャプタの先頭からの再生開始を指示する。これは、チャプタ番号の指定の無いメソッド `playChapter()` に対応する。コマンド `uo_playPrevChapter()` は、現在より一つ前のチャプタからの再生開始を指示する。コマンド `uo_playNextChapter()` は、現在の次のチャプタからの再生開始を指示する。コマンド `uo_stop()` は、再生の停止を指示する。

【 0 0 8 7 】

コマンド `uo_jumpToEnd()` は、プレイリストの最後へのジャンプを指示する。このコマンドは、ムービープレーヤ 3 0 0 に対して、現在の再生を中止してイベント `playListEnd` を発生させるように指示するユーザ操作に対応する。このコマンドに対応して、スクリプトレイヤ 3 0 2 では、イベントハンドラ `onPlayListEnd` が実行される。コマンド `uo_forwardScan(speed)` は、引数 `speed` で指定された再生速度での順方向再生を指示する。コマンド `uo_backwardScan(speed)` は、引数 `speed` で指定された再生速度での逆方向再生を指示する。これらコマンド `uo_forwardScan(speed)` およびコマンド `uo_backwardScan(speed)` における引数 `speed` は、UMD ビデオプレーヤの実装に依存する。

【 0 0 8 8 】

コマンド `uo_playStep(forward)` は、順方向のコマ送り再生を指示する。コマンド `uo_playStep(backward)` は、逆方向のコマ送り再生を指示する。コマンド `uo_pauseOn()` は、ユーザ操作に基づき再生の一時停止を指示する。コマンド `uo_pauseOff()` は、ユーザ操作に基づき再生の一時停止状態を解除する。

【 0 0 8 9 】

コマンド `uo_changeAudioChannel(value)` は、オーディオのチャンネル切り換えまたはデュアルモノ再生時の片チャンネル切り替えを指示する。このコマンドの実行時に、フラグ `audioFlag` の値も対応した内容に変更する。コマンド `uo_setAudioEnabled(boolean)` は、オーディオストリームの ON / OFF を指定する。このコマンドの実行時に、フラグ `audioFlag` の値も対応した内容に変更する。コマンド `uo_setSubtitleEnabled(boolean)` は、字幕ストリームの ON / OFF を指定する。このコマンドの実行時に、フラグ `subtitleFlag` の値も対応した内容に変更する。コマンド `uo_angleChange()` は、表示アングルの変更を指示する。このコマンドによるユーザ操作がムービープレーヤ 3 0 0 に伝えられると、ムービープレーヤ 3 0 0 は、スクリプトレイヤ 3 0 2 に対してイベント `angleChange` を通知する。コマンド `uo_audioChange(audioStreamNumber)` は、再生するオーディオストリームの変更を指示する。コマンド `uo_subtitleChange(subtitleStreamNumber)` は、再生する字幕ス

トリームの変更を指示する。

【0090】

上述した図12に示すイベントおよびイベントのムービープレーヤ300のメソッドとの関係について、より詳細に説明する。イベントmenuは、メニューにジャンプする。このイベントは、ムービープレーヤ300に対してではなく、ネイティブ実装プラットフォーム301からスクリプトレイヤ302に通知される。このイベントmenuがスクリプトレイヤ302に受け取られると、スクリプトレイヤ302は、イベントハンドラonMenuを実行する。イベントexitは、ネイティブ実装プラットフォーム301がUMDビデオアプリケーションを終了させる際に、ネイティブ実装プラットフォーム301から発せられるイベントである。このイベントexitがスクリプトレイヤ302に受け取られると、スクリプト

10

【0091】

イベントup、イベントdown、イベントleft、イベントright、イベントfocusIn、イベントfocusOut、イベントpushおよびイベントcancelは、画面に表示されているGUI部品であるボタン画像にフォーカスが当たっている場合に発生するイベントである。このイベントは、ムービープレーヤ300に対してではなく、ネイティブ実装プラットフォーム301からスクリプトレイヤ302に通知される。なお、ボタン画像にフォーカスが当たった場合とは、例えば、画面上の位置を指示するためのカーソルがボタン画像の表示座標を示し、当該ボタン画像が選択可能となっているような状態である。イベントup、イベントdown、イベントleftおよびイベントrightは、ボタン画像に対するフォーカスが、それぞれ

20

【0092】

イベントautoPlayおよびイベントcontinuePlayは、スクリプトレイヤ302におけるスクリプトの実行開始を指示するイベントである。イベントautoPlayは、ディスクの装填時に自動的にスクリプトの実行を開始するように指示するイベントである。イベントcontinuePlayは、ディスク装填時に、例えばリジュームインフォメーション324に基づき、以前中止された時点からのスクリプトの実行再開を指示する。

30

【0093】

図12で示したイベントに対しては、イベントが発生したときに実行されるプログラムが存在する。このイベントに対応したプログラムをイベントハンドラと称する。イベントとイベントハンドラとは、例えば名前に対応関係をつけることができる。一例として、イベント名の先頭に「on」を付加したものがイベントハンドラ名となる。図13および図14は、一例のイベントハンドラを示す。イベントハンドラの内容をコンテンツ制作者が記述することにより、UMDビデオプレーヤにコンテンツ制作者が意図する様々な動作を実行させることが可能になる。

【0094】

図13は、ムービープレーヤオブジェクト300が持つ一例のイベントの一部と、対応するイベントハンドラとを示す。この図13のイベントは、上述した図2のイベント312に対応し、ムービープレーヤ300からスクリプトレイヤ302に通知される。イベントハンドラは、一種のインターフェイスであって、その内容は、例えばコンテンツ制作者がスクリプト言語を用いて実装する。イベントハンドラをこのように構成することで、イベント発生時に、コンテンツ制作者の意図する動作を実現することができる。

40

【0095】

イベントmarkおよびイベントハンドラonMark()は、イベントマークが検出された際に実行される。イベントマークは、例えば、プレイリスト中に埋め込まれ、プレイリストの再生中にムービープレーヤ300により検出される。ムービープレーヤ300によりイベン

50

トマークが検出されると、ムービープレーヤ300からスクリプトレイヤ302に対してイベントmarkが通知される。スクリプトレイヤ302は、このイベントmarkに対応するイベントハンドラonMark()を実行する。同様にして、イベントpalyListEndおよびイベントハンドラonPlayListEnd()は、プレイリストが終了した際に実行される。イベントchapterおよびイベントハンドラonChapter()は、チャプタマーク検出時に実行される。チャプタマークは、例えば、プレイリスト中に埋め込まれ、プレイリストの再生中にムービープレーヤ300により検出される。

【0096】

イベントangleChangeおよびイベントハンドラonAngleChange()は、ユーザ操作によりアングル変更が指示されたときに実行される。例えば、ユーザ操作に応じてキー入力VK_ANGLEがユーザ入力310としてネイティブ実装プラットフォーム301に入力されると、ネイティブ実装プラットフォーム301は、当該ユーザ入力310をコマンドuo_angleChange()に変換してムービープレーヤ300に渡す。ムービープレーヤ300は、このコマンドuo_angleChange()に応じてイベントangleChangeを発生させ、スクリプトレイヤ302に渡す。スクリプトレイヤ302は、このイベントangleChangeに対応したイベントハンドラonAngleChange()を実行する。同様にして、イベントaudioChangeおよびイベントハンドラonAudioChange()は、ユーザ操作によりオーディオの変更が指示されたときに実行される。イベントsubtitleChangeおよびイベントハンドラonSubtitleChange()は、ユーザ操作により字幕変更が指示されたときに実行される。

【0097】

図14は、システムオブジェクトが有する一例のイベントハンドラの一部を示す。この図14に示されるイベントハンドラは、ネイティブ実装プラットフォーム301が予め持っているイベントハンドラであり、ネイティブ実装プラットフォーム301からスクリプトレイヤ302に通知される。

【0098】

イベントmenuおよびイベントハンドラonMenu()は、メニューにジャンプする。イベントmenuは、例えば、ユーザ操作などでメニューキーが押下されたときに、ネイティブ実装プラットフォーム301からスクリプトレイヤ302に通知されるイベントである。スクリプトレイヤ302は、このイベントを受けて、対応するイベントハンドラonMenu()を実行し、イベントハンドラonMenu()内でメニュー画面を構成するGUI部品の配置や表示などを行う。イベントexitおよびイベントハンドラonExit()は、ネイティブ実装プラットフォーム301がUMDビデオアプリケーションを終了させる際に、ネイティブ実装プラットフォーム301から発せられるイベントおよび対応するイベントハンドラである。

【0099】

イベントexitは、例えば、ユーザ操作などによりUMDビデオプレーヤの動作の終了が指示された際に、ネイティブ実装プラットフォーム301からスクリプトレイヤ302に通知される。スクリプトレイヤ302のスク립トは、通知されたイベントexitを受けて、イベントハンドラonExit()内で終了処理を行うことができる。イベントautoPlayおよびイベントハンドラonAutoPlay()、ならびに、イベントcontinuePlayおよびイベントハンドラonContinuePlay()は、それぞれスク립トの実行を開始する。

【0100】

なお、システムオブジェクトのイベントハンドラ以外に、ボタンに関するイベントハンドラがある。このボタンに関するイベントハンドラは、この発明と関連性が低いので、説明を省略する。

【0101】

図15のフローチャートを用いて、ユーザ入力イベントをきっかけとして、用意されたプログラムが実行される一例の処理について、概略的に説明する。図15は、UMDビデオプレーヤにおいてディスクを通常再生中に、ユーザにより、次のチャプタを再生することを指示するための"next"キーが押されたときに、このキー入力に対応して、次のチャプタにジャンプして再生を開始すると共に、用意されたメッセージを画面上に表示する例で

10

20

30

40

50

ある。

【0102】

例えば、UMDビデオプレーヤによりディスクを通常再生中に、ユーザがUMDビデオプレーヤのリモートコントロールコマンドを用いてキー"next"を押下すると(ステップS10)、ネイティブ実装プラットフォーム301に対するユーザ入力310として、キーVK_NEXTが渡される。ネイティブ実装プラットフォーム301では、このユーザ入力310に対応してユーザコマンドuo_playNextChapter()が発生する(ステップS11)。このユーザコマンドuo_playNextChapter()は、ムービープレーヤ300に通知される。

【0103】

このコマンドuo_playNextChapter()を受け取ったムービープレーヤ300は、データベース320を検索し、プレイリスト情報から現在再生している位置を基準として、次のチャプタマークの位置を取得する(ステップS12)。ステップS13で、次のチャプタマークが存在するか否かが判断され、若し、存在しないと判断された場合、チャプタジャンプを行わず、現在の再生が継続される。

【0104】

一方、ステップS13で、次のチャプタマークが存在すると判断されれば、処理はステップS14に移行する。ステップS14では、ムービープレーヤ300は、現在の再生を中止し、次のチャプタマークが指し示す、クリップAVストリームファイル内のバイト位置を、データベース320のクリップインフォメーションファイルの特徴点情報から取得する。そして、ステップS15で、取得されたファイル内バイト位置にアクセスし、その位置からストリームの読み込みを開始して再生を開始する。

【0105】

ステップS16以下は、チャプタが切り替わったことを知らせるメッセージを画面上に表示するための一連の手順である。チャプタが切り替わりチャプタの先頭からの再生が開始されると、チャプタイベントが発生する(ステップS16)。例えば、チャプタの先頭に設けられたチャプタマークがムービープレーヤ300に検出され、イベントchapterが発生される。このチャプタイベントは、ムービープレーヤ300からスクリプトレイヤ302に通知される。ムービープレーヤ300は、このイベントの通知時に、ジャンプするチャプタのチャプタ番号も共に、スクリプトレイヤ302に対して通知する。スクリプトレイヤ302は、通知されたイベントに対応するイベントハンドラ、例えばイベントハンドラonChapter()の実行を開始する(ステップS17)。

【0106】

この例では、イベントハンドラ内には、チャプタが切り替わった際に画面上にその旨を知らせるメッセージを表示する動作が記述されているものとする。スクリプトレイヤ302のスクリプトは、このイベントハンドラを実行し、イベント発生時にムービープレーヤ300から通知されたジャンプ先のチャプタ番号を取得し(ステップS18)、ネイティブ実装プラットフォーム301に対して、例えば取得したチャプタ番号のチャプタの先頭であるなど、所定のメッセージを画面上に表示する指示を出す。ネイティブ実装プラットフォーム301は、この指示に応じて、画面上にメッセージを表示し(ステップS19)、イベントハンドラによる処理が終了される(ステップS20)。

【0107】

上述のような処理により、ユーザが次のチャプタの再生開始を指示するキー"next"を操作することによりチャプタジャンプが行われ、ジャンプ先である次のチャプタの再生開始時にチャプタの先頭であることを示すメッセージが画面上に表示されることになる。

【0108】

このように、ユーザ入力イベントは、ムービープレーヤ300の状態を変化させ、また、新たなイベントを発生させる契機ともなり、新たに発生したイベントを利用して様々な処理を行わせることができる。

【0109】

図16は、UMDビデオプレーヤにディスクがロードされてからイジェクトされるまで

10

20

30

40

50

の処理を概略的に示す。なお、図 16 中、斜線を付したブロックで記述された処理は、スクリプトが実行されている状態を示す。

【0110】

先ず、ユーザにより UMD ビデオプレーヤにディスクが装填されると、UMD ビデオプレーヤは、所定の動作によりディスクをロードし、再生可能な状態にする（ステップ S30）。ディスクがロードされると、ネイティブ実装プラットフォーム 301 によりリジュームインフォメーション 324 が参照され、当該ディスクに対応する続き再生情報がロードされる（ステップ S31）。

【0111】

次に、当該ディスクに対応するリジュームインフォメーション 324 内が参照され、続き再生情報が存在するか否かが判断され（ステップ S32）、存在すれば、ネイティブ実装プラットフォーム 301 からスクリプトレイヤに対してイベント continuePlay が通知される。スクリプトレイヤ 302 は、通知されたイベント continuePlay に対応するイベントハンドラ onContinuePlay を実行する（ステップ S33）。ステップ S32 で、当該ディスクに対応する続き再生情報が存在しないと判断されれば、処理はステップ S34 に移行し、ネイティブ実装プラットフォーム 301 からスクリプトレイヤ 302 に対してイベント autoPlay が通知され、スクリプトレイヤ 302 は、対応するイベントハンドラ onAutoPlay を実行させる。

【0112】

ステップ S35 では、イベントハンドラ onAutoPlay やイベントハンドラ onContinuePlay の記述内容に基づきディスクの再生動作などが行われ、ディスクの再生動作に伴い発生されたイベントや、当該イベントに対応するイベントハンドラが実行される。

【0113】

ここで、ネイティブ実装プラットフォーム 301 からイベント exit が発生されると、ステップ S36 で、スクリプトレイヤ 302 において対応するイベントハンドラ onExit が実行され、UMD ビデオアプリケーションを終了させるための処理が実行される。イベント exit は、例えばリモートコントロールコマンドに対する所定の操作に応じたユーザ入力 310 に基づき、ネイティブ実装プラットフォーム 301 で発生される。

【0114】

イベントハンドラ onExit に基づくスクリプト処理が終了すると、ネイティブ実装プラットフォーム 301 に処理が移る。そして、ステップ S37 で、ムービープレーヤ 300 において、再生動作を停止する処理が実行される。このとき、停止された直前の状態が続き再生情報としてリジュームインフォメーション 324 に記憶される。そして、ディスクの再生が終了され（ステップ S38）、同じディスクを再び再生しない際には（ステップ S39）、ステップ S40 で、ネイティブ実装プラットフォーム 301 によりディスクがイジェクトされ、一連の処理が終了される。また、同じディスクを再び再生する際には、処理はステップ S31 に戻される。

【0115】

図 17 は、スクリプトファイルの構成例を示す。図 1 を用いて既に説明したように、スクリプトファイルは、スクリプトレイヤ 302 を構成するファイル "SCRIPT.DAT" 内のファイルとして存在する。スクリプトファイルは、イベントハンドラ群とメイン処理部とからなる。イベントハンドラ群は、1 または複数のイベントハンドラが並べられる。イベントの発生がスクリプトレイヤ 302 に通知される毎に、通知されたイベントに対応したイベントハンドラが検索され、実行される。メイン処理部は、例えば各イベントハンドラに共通して用いられるグローバル変数などの定義が記述され、通常、最初に 1 回だけ実行される。

【0116】

図 18 は、イベントハンドラ onAutoPlay() を実行する一例の手順を示す。UMD ビデオプレーヤにディスクを装填する際に、ユーザにより、始めから再生を行うように、ムービープレーヤ 300 に対して再生指示がなされた場合（ステップ S50）に、この処理が行

われる。ネイティブ実装プラットフォーム 301 は、ステップ S51 で、スクリプト中にイベントハンドラ `onAutoPlay()` が存在するか否かが調べられる。若し、存在すれば、ネイティブ実装プラットフォーム 301 は、イベント `autoPlay` をスクリプトレイヤ 302 に対して通知する (ステップ S52)。これを受けて、ステップ S54 で、スクリプトレイヤ 302 は、イベントハンドラ `onAutoPlay()` を実行する。これにより、装填されたディスクが自動的に再生開始される。

【0117】

一方、ステップ S51 で、スクリプト中にイベントハンドラ `onAutoPlay()` が存在しないとされれば、処理はステップ S53 に移行し、ネイティブ実装プラットフォーム 301 は、イベント `exit` をスクリプトレイヤ 302 に対して通知する。この場合、例えば、メニューキーなどを操作して、ネイティブ実装プラットフォーム 301 に実装されているメニュー画面から再生指示を与えることで、ディスクの再生を開始することができる。スクリプトレイヤ 302 がイベントハンドラ `onExit()` を持っている場合、このイベントハンドラ `onExit()` が実行される。

10

【0118】

図 19 は、イベントハンドラ `onContinuePlay()` を実行する一例の手順を示す。UMD ビデオプレーヤにディスクを装填する際に、ユーザにより、続き再生を行うように、ムービープレーヤ 300 に対して再生指示がなされた場合 (ステップ S60) に、この処理が行われる。ネイティブ実装プラットフォーム 301 は、ステップ S61 で、装填されたディスクに対応するリジュームインフォメーション 324 が存在するか否かが調べられる。若し、存在しなければ、処理はステップ S62 に移行し、先頭からの再生となる。

20

【0119】

装填されたディスクに対応するリジュームインフォメーション 324 が存在する場合には、処理はステップ S63 に移行し、スクリプト中にイベントハンドラ `onContinuePlay()` が存在するか否かが調べられる。若し、存在すれば、ネイティブ実装プラットフォーム 301 は、イベント `continuePlay` をスクリプトレイヤ 302 に対して通知する。これを受けて、スクリプトレイヤ 302 は、イベントハンドラ `onContinuePlay()` を実行する (ステップ S64)。これにより、装填されたディスクが、イベントハンドラ `onContinuePlay()` に従い再生が再開される。

【0120】

一方、ステップ S63 で、スクリプト中にイベントハンドラ `onContinuePlay()` が存在しないとされれば、処理はステップ S65 に移行し、デフォルトのイベントハンドラ `onContinuePlay()` が実行される。デフォルトのイベントハンドラ `onContinuePlay()` は、例えば、リジュームインフォメーション 324 の情報に基づき前回の再生終了位置から、単純に再生を開始する。

30

【0121】

なお、これらの、イベントハンドラ `onAutoPlay` およびイベントハンドラ `onContinuePlay` によるユーザインターフェイスは、上述の例に限られず、様々な方法が考えられる。例えば、上述の図 19 においては、ステップ S60 でユーザにより続き再生が指示されてから、装填されたディスクに対応するリジュームインフォメーション 324 が存在するか否かを調べているが、これは、順序を逆にし、リジュームインフォメーション 324 が存在するか否かを先に調べ、存在する場合に、続き再生を行うか否かの選択をユーザに促してもよい。

40

【0122】

図 20 は、再生終了時の一例の処理を示す。ディスクの再生中に、例えばユーザにより、ムービープレーヤ 300 に対して再生を終了する指示がなされた場合 (ステップ S70) に、この処理が行われる。再生終了を指示するユーザ入力 310 がネイティブ実装プラットフォーム 301 に対して入力されると、ネイティブ実装プラットフォーム 301 は、終了処理を開始する (ステップ S71)。終了処理は、例えば下記の 3 つの処理である。

(1) 新たなイベント発生を抑止

50

(2) キューに溜まったイベントハンドラの破棄

(3) ムービープレーヤ 300 に対する制御コマンド `uo_stop()` の発行

【0123】

ステップ S71 の処理が実行され、現在実行されているイベントハンドラが終了すると (ステップ S72)、次のステップ S73 で、ネイティブ実装プラットフォーム 301 からスクリプトレイヤ 302 に対して、イベント `exit` が通知される。スクリプトレイヤ 302 は、これを受けて、スクリプトレイヤ 302 は、イベントハンドラ `onExit()` を実行する (ステップ S74)。イベントハンドラ `onExit()` により、例えば、再生終了時の所定の後処理や、ユーザによる設定データを記憶するメソッド `setUserData` などが実行される。

【0124】

そして、次のステップ S75 で、ネイティブ実装プラットフォーム 301 により、終了処理がなされる。この終了処理では、例えば、不揮発性メモリに対する続き情報の保存 (すなわち、再生終了直前の状態のリジュームインフォメーション 324 に対するバックアップ) や、システムメニューへの遷移などが行われる。

【0125】

以上のようなプレーヤモデルにより、ビデオ、オーディオおよび字幕の再生が可能となる。また、コンテンツ制作者が予め設定しておいた、再生中のある時刻にあるイベントを発生させて、コンテンツ制作者が予め用意しておいたイベントハンドラを実行するようにしているため、コンテンツ制作者が意図する動作を実現できる。さらに、UMD ビデオプレーヤによるディスクの再生中にユーザ操作があった場合は、ネイティブ実装プラットフォーム 301 からムービープレーヤ 300 に対して、ユーザ操作に応じたコマンドが通知され、ユーザの意図する通りにプレーヤの状態を変化させることができる。さらにまた、ユーザ操作によるユーザ入力を受けたネイティブ実装プラットフォームが、スクリプトレイヤ 302 に対してユーザ入力に対応するイベントを通知することにより、ユーザ操作に応じてコンテンツ制作者が用意した動作を実現することが可能となる。このようにプレーヤモデルを構築することで、ビデオ、オーディオおよび字幕の再生と、インタラクティブな操作とをユーザに提供することが可能となる。

【0126】

5. スクリプトプログラムの例

次に、スクリプトレイヤ 302 のスクリプトプログラムの例について説明する。まず、図 21 に示されるようなコンテンツ再生の流れが、コンテンツ制作者により作られているものとする。図 21 に示されるコンテンツは、表示される要素としては、プレイリスト 400 および 401、トップメニュー 402、ならびに、メッセージ 403 から構成される。プレイリスト 400 は、ディスクが装填されると自動的に表示される警告文画面を表示するためのものである。プレイリスト 401 は、例えばこのコンテンツの主眼である映画の本編である。トップメニュー画面 402 は、プレイリスト 401 の再生を指示できるように、ボタンなどの GUI 部品が配置される。また、メッセージ 403 は、プレイリスト 401 の再生中の任意の時刻に表示される。

【0127】

さらに、この図 21 の構成では、幾つかのイベントハンドラが用意されている。イベントハンドラ `onAutoPlay()` は、ディスクが UMD プレーヤに装填されると、プレイリスト 400 を自動的に再生し、警告文を表示させる。イベントハンドラ `onPlayListEnd()` は、プレイリストの再生が終了すると呼び出されるイベントハンドラで、この図 21 の例では、プレイリスト 400 やプレイリスト 401 の終了で呼び出される。すなわち、イベントハンドラ `onPlayListEnd()` は、どのプレイリストが終了したかを判定し、プレイリスト 400 の再生が終了した場合には、プレイリスト 401 の再生開始を指示する。また、プレイリスト 401 の再生が終了した場合には、トップメニュー画面 402 を呼び出す。

【0128】

イベントハンドラ `onMenu()` は、ユーザがメニューキーを操作したときに呼び出され、トップメニュー 402 を呼び出して画面に表示する。イベントハンドラ `onMark()` は、再生中

10

20

30

40

50

にマークMarkが指し示す時刻に到達したときに実行される。この図21の例では、プレイリスト401に対してマークMarkが設定されており、プレイリスト401の再生がマークMarkの指し示す時刻に到達すると、画面上にメッセージ403が表示されるようになっている。

【0129】

すなわち、図21の例では、UMDビデオプレーヤにディスクが装填されると、イベントハンドラonAutoPlayが呼び出されてプレイリスト400が再生され、警告画面が表示される。プレイリスト400の再生時間が経過し、プレイリスト400の最後に到達すると、イベントハンドラonPlayListEndが呼び出され、プレイリスト400が最後まで再生されたことが判定され、次のプレイリスト401が再生される。ここで、プレイリスト401の再生中に、ユーザによりメニューキーが操作されると、イベントハンドラonMenuが呼び出され、トップメニュー画面402が表示される。また、イベントハンドラonMenuにより、トップメニュー画面402に対する所定の操作に応じて、プレイリスト401の先頭から再生が開始される。さらに、プレイリスト401の再生時刻がマークMarkが指し示す時刻に到達したら、イベントハンドラonMarkが呼び出され、メッセージ403が画面上に表示される。プレイリスト401が最後まで再生されると、イベントハンドラonPlayListEndが呼び出され、プレイリスト401が最後まで再生されたことが判定され、トップメニュー画面402が表示される。

10

【0130】

図22は、この図21に示すような動作を実現するための一例のスクリプトプログラムを示す。上述したように、スクリプトプログラムは、イベントハンドラが並べられ、イベントの発生に応じて対応するイベントハンドラが実行されるようになっている。スクリプトプログラムは、後述するファイル"SCRIPT.DAT"に格納される。

20

【0131】

ムービープレーヤ300に対してプレイリストの再生を指示するメソッドは、「movieplayer.play()」である。括弧内には、引数として、再生するプレイリストの番号を記述する。プレイリストの再生が終了すると、イベントplayListEndが発生する。このイベントplayListEndが発生すると、スクリプトからイベントハンドラmovieplayer.onPlayListEnd()が呼び出される。このとき、スクリプトには、イベントplayListEndと共に、オブジェクトevent_infoが渡される。オブジェクトevent_infoには、どのプレイリストが終了したかを表すプレイリスト番号などが格納される。スクリプトでは、このオブジェクトevent_infoの内容により、次の動作を変えることができる。

30

【0132】

6. ファイルの管理構造について

次に、UMDビデオ規格に適用されるファイルの管理構造について、図23を用いて説明する。ファイルは、ディレクトリ構造により階層的に管理されて、ディスク上に記録される。ディスクのファイルシステムは、ISO(International Organization for Standardization)-9660あるいはUDF(Universal Disk Format)などで規定されたファイルシステムを適用することができる。

【0133】

ルートディレクトリの下に、ファイル"TITLEID.DAT"およびディレクトリ"VIDEO"が置かれる。ディレクトリ"VIDEO"の下には、さらに、ディレクトリ"RESOURCE"、ディレクトリ"CLIP"およびディレクトリ"STREAM"、ならびに、ファイル"PLAYLIST.DAT"が置かれる。

40

【0134】

ファイル"TITLEID.DAT"は、タイトル(コンテンツの種類)毎に異なるタイトル識別子が格納されるファイルである。1つのディスクに対し、1つのファイル"TITLEID.DAT"を有する。

【0135】

ディレクトリ"RESOURCE"の下には、ファイル"SCRIPT.DAT"が置かれる。このファイル"SCRIPT.DAT"は、上述したように、スクリプトレイヤ302を構成するスクリプトプログラ

50

ムが格納される。ディレクトリ"RESOURCE"の下には、通常、1個のファイル"SCRIPT.DAT"が置かれる。これに限らず、ディレクトリ"RESOURCE"の下に、複数のファイル"SCRIPT.DAT"を置くこともできる。このときは、例えばファイル名の一部をそれぞれ変更し、互いに重複しないようにする。複数のファイル"SCRIPT.DAT"は、例えば、表示言語の異なる複数のメニューなどを用意する際に、言語毎に1つファイル"SCRIPT.DAT"が用いられる。この場合でも、実際に使用されるファイル"SCRIPT.DAT"は、1つとされる。

【0136】

ディレクトリ"CLIP"の下には、1以上のクリップインフォメーションファイルが置かれる。クリップインフォメーションファイルは、ファイル名を、デリミタであるピリオドの前が「00001」などの5文字乃至数文字からなる文字列（この例では数字）、ピリオドの後ろの拡張子が「CLP」とされる。拡張子「CLP」により、当該ファイルがクリップインフォメーションファイルであることを識別できる。

10

【0137】

ディレクトリ"STREAM"の下には、1以上のクリップAVストリームファイルが置かれる。クリップAVストリームファイルは、ファイル名を、デリミタであるピリオドの前が「00001」などの5文字乃至数文字からなる文字列（この例では数字）、ピリオドの後ろの拡張子が「PS」とされる。拡張子「PS」により、当該ファイルがクリップAVストリームファイルであることを識別できる。この実施の一形態では、クリップAVストリームファイルは、ビデオストリーム、オーディオストリームおよびサブタイトル（字幕）ストリームが多重化され、MPEG2 (Moving Pictures Experts Group 2)のプログラムストリームとして、上述の拡張子「PS」で識別されるファイルに格納される。

20

【0138】

上述したように、クリップAVストリームファイルは、ビデオデータおよびオーディオデータを圧縮符号化および時分割多重して得られるファイルであって、このファイルを読み込み、デコード処理を行うことで、ビデオデータおよびオーディオデータが得られる。また、クリップインフォメーションファイルは、このクリップAVストリームファイルの性質などが記述されるファイルであって、クリップAVストリームファイルと対応する。この実施の一形態では、クリップインフォメーションファイルと対応するクリップAVストリームファイルとで、ファイル名における、拡張子の前の、5文字乃至数文字からなる文字列を一致させておくことで、両者の対応関係を容易に把握できる。

30

【0139】

ファイル"SCRIPT.DAT"は、上述したように、スクリプトプログラムが記述されたスクリプトファイルであり、この実施の一形態が適用されるディスクの再生形態をインタラクティブなものとするために用いるプログラムが格納されている。ファイル"SCRIPT.DAT"は、ディスクに格納される他のファイルに先立って読み出される。

【0140】

ファイル"PLAYLIST.DAT"は、クリップAVストリームの再生順を指定するプレイリストが記述されたプレイリストファイルである。図24～図26を用いて、ファイル"PLAYLIST.DAT"の内部構造について説明する。図24は、ファイル"PLAYLIST.DAT"の全体構造を表す一例のシンタクスを示す。ここでは、シンタクスをコンピュータ装置などのプログラムの記述言語として用いられるC言語の記述法に基づき示す。これは、他のシンタクスを表す図において、同様である。

40

【0141】

フィールドname_lengthは、8ビットのデータ長を有し、このプレイリストファイルに付された名称の長さを示す。フィールドname_stringは、255バイトのデータ長を有し、このプレイリストファイルに付された名称を示す。フィールドname_stringは、その先頭から、フィールドname_lengthが表すバイト長までが、有効な名称として使用される。例えば、フィールドname_lengthが値"10"を持つ場合には、フィールドname_stringの先頭から10バイト分が有効な名称として解釈される。

【0142】

50

フィールドnumber_of_PlayListsは、16ビットのデータ長を有し、続けて記述されるブロックPlayList()の個数を示す。次行のforループによりフィールドnumber_of_PlayListsに示される回数分だけ、当該個数のブロックPlayList()が記述される。ブロックPlayList()は、プレイリストそのものである。

【0143】

ブロックPlayList()の一例の内部構造について説明する。ブロックPlayList()の先頭には、フィールドPlayList_data_lengthが配される。フィールドPlayList_data_lengthは、32ビットのデータ長を有し、当該フィールドPlayList_data_lengthを含むブロックPlayList()のデータ長を示す。続いて、15ビットのデータ長を有するフィールドreserved_for_word_alignmentと、1ビットのデータ長を有するフラグcapture_enable_flag_PlayListとが配される。フィールドreserved_for_word_alignmentは、データ長が1ビットのフラグcapture_enable_flag_PlayListと組み合わせて、ブロックPlayList()内での配置を16ビットの位置に揃えるために用いられる。

【0144】

フラグcapture_enable_flag_PlayListは、当該capture_enable_flag_PlayListを含むブロックPlayList()に属する動画像の二次利用を許可するか否かを示すフラグである。例えば、このフラグcapture_enable_flag_PlayListの値が"1"であれば、当該PlayList()に属する動画像の、再生機内での2次利用を許可することを示す。

【0145】

なお、上述では、フラグcapture_enable_flag_PlayListを1ビットのフラグとしたが、これはこの例に限定されない。例えば、フラグcapture_enable_flag_PlayListを複数ビット構成として、2次利用の段階的な許可を記述するようにしてもよい。一例として、フラグcapture_enable_flag_PlayListを2ビット構成とし、値が"0"の場合には2次利用を完全禁止とし、値が"1"の場合には例えば64画素×64ラインなど、所定の解像度以下に圧縮符号化した場合のみ2次利用を可能とする。また、値が"2"であれば、制限無く2次利用を許可するといった利用が考えられる。これに限らず、2ビット構成のうちビット0が値"1"の場合にはコンテンツ再生アプリケーションでの2次使用を許可し、ビット1が値"1"の場合には同一筐体内の他のアプリケーション(例えば壁紙画像やスクリーンセーバ)での2次使用を許可する。この場合には、ビット0およびビット1の値を組み合わせて用いることができる。

【0146】

フィールドPlayList_name_lengthは、8ビットのデータ長を有し、このブロックPlayList()に付された名称の長さを示す。フィールドPlayList_name_stringは、255ビットのデータ長を有し、このブロックPlayList()に付された名称を示す。フィールドPlayList_name_stringは、その先頭から、フィールドPlayList_name_stringが表すバイト長までが、有効な名称として使用される。

【0147】

フィールドnumber_of_PlayItemsは、16ビットのデータ長を有し、続けて記述されるブロックPlayItem()の個数を示す。次行のforループによりフィールドnumber_of_PlayItem2に示される回数分だけ、当該個数のブロックPlayItem()が記述される。ブロックPlayItem()は、プレイアイテムそのものである。

【0148】

ブロックPlayList()内の各ブロックPlayItem()には、識別情報(ID)が付与される。例えば、ブロックPlayList()内の最初に記述されるブロックPlayItem()は、0番とされ、以降、ブロックPlayItem()の出現順に、1番、2番、・・・と通し番号が付される。この通し番号が各ブロックPlayItem()の識別情報として用いられる。ブロックPlayItem()の個数だけ繰り返されるforループの引数iを、対応するブロックPlayItem()の識別情報として用いることができる。ブロックPlayItem()の次に、ブロックPlayListMark()が配置される。

【0149】

10

20

30

40

50

図 2 5 を用いて、ブロック PlayItem() の一例の内部構造について説明する。ブロック PlayItem() の先頭には、フィールド length が配される。フィールド length は、16 ビットのデータ長を有し、当該ブロック PlayItem() の長さを示す。続いて、フィールド Clip_Information_file_name_length が配される。フィールド Clip_Information_file_name_length は、16 ビットのデータ長を有し、このブロック PlayItem() に対応するクリップインフォメーションファイルの名称の長さを示す。フィールド Clip_Information_file_name は、バイト単位で可変長のデータ長を有し、このブロック PlayItem() に対応するクリップインフォメーションファイルの名称を示す。フィールド Clip_Information_file_name は、その先頭から、フィールド Clip_Information_file_name_length が表すバイト長までが、有効な名称として使用される。フィールド Clip_Information_file_name でクリップインフォメーションファイルが指定されると、上述したファイル名の対応関係により、当該クリップインフォメーションファイルに対応するクリップ AV ストリームファイルが特定できる。

10

【0150】

フィールド IN_time およびフィールド OUT_time は、それぞれ 32 ビットのデータ長を有し、ブロック PlayItem() 内においてフィールド Clip_Information_file_name で指定したクリップインフォメーションファイルに対応するクリップ AV ストリームファイルの再生開始位置および再生終了位置を指定する時刻情報である。これらフィールド IN_time およびフィールド OUT_time の情報を用いることで、クリップ AV ストリームファイルの先頭以外の部分からの再生開始を指定することができる。同様に、クリップ AV ストリームファイルの後端以外の再生終了を指定することができる。

20

【0151】

図 2 6 を用いて、ブロック PlaylistMark() の一例の内部構造について説明する。ブロック PlaylistMark() の先頭には、フィールド length が配される。フィールド length は、32 ビットのデータ長を有し、当該ブロック PlaylistMark() の長さを示す。続いて、フィールド number_of_PlayList_marks が配される。フィールド number_of_PlayList_marks は、16 ビットのデータ長を有し、続くブロック Mark() の個数を示す。次行の for ループによりフィールド number_of_PlayList_marks に示される回数分だけ、当該個数のブロック Mark() が記述される。

【0152】

ブロック Mark() の一例の内部構造について説明する。ブロック Mark() は、先頭にフィールド mark_type が配される。フィールド mark_type は、8 ビットのデータ長を有し、当該フィールド mark_type を含むブロック Mark() の種類を示す。この実施の一形態では、図 2 7 に一例が示されるように、チャプタマーク、インデックスマークおよびイベントマークの 3 種類のマークが規定されている。チャプタは、プレイリスト (ブロック Playlist()) を分割する頭出し単位であり、インデックスは、チャプタをさらに分割する頭出し単位である。チャプタマークおよびインデックスマークは、それぞれ、これらチャプタ位置およびインデックス位置を時刻情報で示す。イベントマークは、マークイベントを発生させるマークである。

30

【0153】

フィールド mark_name_length は、8 ビットのデータ長を有し、このブロック Mark() に付された名称の長さを示す。ブロック Mark() の最下行に配されるフィールド mark_name_string は、このブロック Mark() に付された名称を示す。フィールド mark_name_string は、その先頭から、フィールド mark_name_length が表すバイト長までが、有効な名称として使用される。

40

【0154】

フィールド ref_to_PlayItem_id、フィールド mark_time_stamp、フィールド entry_ES_stream_id およびフィールド entry_ES_private_stream_id の 4 要素は、ブロック Playlist() 上で定義されるブロック Mark() を、クリップ AV ストリームファイルと対応付ける。すなわち、フィールド ref_to_PlayItem_id は、16 ビットのデータ長を有し、ブロック PlayItem() の識別情報を示す。これにより、クリップインフォメーションファイルと、クリップ

50

A Vストリームファイルとが特定される。

【0155】

フィールドmark_time_stampは、32ビットのデータ長を有し、クリップA Vストリームファイル内でのマークの時刻を指定するために用いられる。図28を用いて、概略的に説明する。図28において、プレイリストは、番号0、1および2がそれぞれ指定された3つのプレイアイテム(PlayItem(#0)、PlayItem(#1)およびPlayItem(#2))からなり、プレイリスト上の時刻 t_0 は、番号1のプレイアイテム(PlayItem(#1))に含まれるものとする。また、番号0、1および2の各プレイアイテムは、それぞれ対応するクリップインフォメーションファイルを介してクリップA Vストリームファイルのプログラムストリーム(Program Sream)A、BおよびCにそれぞれ対応しているものとする。

10

【0156】

このような場合において、プレイリスト上の時刻 t_0 にマークを指定する場合、フィールドref_to_PlayItem_idの値を、時刻 t_0 を含むプレイアイテムを示す"1"とし、さらに、対応するクリップA VストリームファイルB上で時刻 t_0 に相当する時刻を、フィールドmark_time_stampに記述する。

【0157】

図26の説明に戻り、フィールドmark_time_stampに続けてフィールドentry_ES_stream_idおよびフィールドentry_ES_private_stream_idが配される。フィールドentry_ES_stream_idおよびフィールドentry_ES_private_stream_idは、それぞれ8ビットのデータ長を有し、当該ブロックMark()が特定のエレメンタリストリームに関連付けられている場合に、そのエレメンタリストリームを特定するために用いられる。フィールドentry_ES_stream_idおよびフィールドentry_ES_private_stream_idは、それぞれ該当するエレメンタリストリームが多重化されているパケット(packet())のストリームID(stream_id)と、プライベートパケットヘッダ(private_packet_header())のプライベートストリームID(private_stream_id)を示す。

20

【0158】

なお、これらパケット(packet())のストリームID(stream_id)、プライベートパケットヘッダ(private_packet_header())のプライベートストリームID(private_stream_id)は、例えばMPEG2システムのプログラムストリームの規定に基づく。

【0159】

これらフィールドentry_ES_stream_idおよびフィールドentry_ES_private_stream_idは、例えば、クリップA Vストリーム#0とクリップA Vストリーム#1とで異なるチャプタ構成である場合などに用いられる。該当するブロックMark()が特定のエレメンタリストリームに関連付けられていない場合には、これら2つのフィールドの値がそれぞれ"0"とされる。

30

【0160】

次に、図29～図33を用いて、クリップインフォメーションファイルの内部構造について説明する。クリップインフォメーションファイル"XXXXX.CLP"は、上述したように、ディレクトリ"STREAM"の下に置かれた、対応するクリップA Vストリームファイル"XXXXX.PS"の性質などを記述する。

40

【0161】

図29は、クリップA Vストリームファイル"XXXXX.CLP"の全体構造を表す一例のシンタクスを示す。クリップA Vストリームファイル"XXXXX.CLP"は、先頭に、フィールドpresentation_start_timeおよびフィールドpresentation_end_timeがそれぞれ配される。フィールドpresentation_start_timeおよびフィールドpresentation_end_timeは、それぞれ32ビットのデータ長を有し、対応するクリップA Vストリームファイルの先頭と後端の時刻を示す。時刻情報は、MPEG2システムにおけるPTS(Presentation Time Stamp)を用いることができる。PTSは、90kHzの精度を有する。

【0162】

次に、7ビットのデータ長を有するフィールドreserved_for_word_alignmentと、1ビ

50

ットのデータ長を有するフラグcapture_enable_flag_Clipとが配される。フィールドreserved_for_word_alignmentは、データ長が1ビットのフラグcapture_enable_flag_Clipと組み合わせて、ファイル"XXXXX.CLP"内での配置を16ビットの位置に揃えるために用いられる。フラグcapture_enable_flag_Clipは、当該ファイル"XXXXX.CLP"に対応するクリップAVストリームファイルに含まれる動画の2次利用を許可するか否かを示すフラグである。例えば、このフラグcapture_enable_flag_Clipの値が"1"であれば、当該ファイル"XXXXX.CLP"に対応するクリップAVストリームファイルの動画の、再生機内での2次利用を許可することを示す。

【0163】

フィールドnumber_of_streamsは、8ビットのデータ長を有し、続くブロックStreamInfo()構造の個数を示す。フィールドnumber_of_streamsの次から、forループによりフィールドnumber_of_streamsで示される回数分だけ、ブロックStreamInfo()が記述される。forループの後には、ブロックEP_map()が配される。

【0164】

ブロックStreamInfo()の一例の内部構造について説明する。ブロックStreamInfo()の先頭には、フィールドlengthが配される。フィールドlengthは、16ビットのデータ長を有し、当該ブロックStreamInfo()の長さを示す。続いて、それぞれ8ビットのデータ長を有するフィールドstream_idおよびフィールドprivate_stream_idが配され、図30に一例が示されるように、当該ブロックStreamInfo()をエレメンタリストリームに関連付けている。この図30の例では、当該ブロックStreamInfo()は、フィールドstream_idが値"0xE0"~値"0xEF"でビデオストリームに関連付けられ、値"0xBD"でATRA C (Adaptive Transform Acoustic Coding)オーディオストリーム、LPCM (Linear Pulse Code Modulation)オーディオストリームまたは字幕ストリームと関連付けられる。また、当該ブロックStreamInfo()は、フィールドprivate_stream_idが値"0x00"~値"0x0F"、値"0x10"~値"0x1F"および値"0x80"~値"0x9F"で、ATRA Cオーディオストリーム、LPCMオーディオストリームおよび字幕ストリームにそれぞれ関連付けられる。

【0165】

なお、図30での値の表記において、「0x」は、後続する数値が16進表記であることを示す。これは、以下の同様な表現において、共通である。

【0166】

ここで、ブロックStreamInfo()は、大別して、ストリーム中で変化しない情報とストリーム中で変化する情報との2種類の情報が記述されている。ストリーム中で変化しない情報は、ブロックStaticInfo()に記述される。一方、ストリーム中で変化する情報は、変化点を時刻情報で指定して、ブロックDynamicInfo()に記述される。

【0167】

ブロックStreamInfo()において、ブロックStaticInfo()の後ろにバイト位置を揃えるための、8ビットのデータ長を有するフィールドreserved_for_word_alignmentが配され、その次に、フィールドnumber_of_DynamicInfoが配される。フィールドnumber_of_DynamicInfoは、8ビットのデータ長を有し、ブロックStreamInfo()内にその後に記述されるブロックDynamicInfo()の個数が示される。forループにより、フィールドnumber_of_DynamicInfoで示される回数分だけ、フィールドpts_change_pointおよびブロックDynamicInfo()が記述される。

【0168】

フィールドpts_change_pointは、32ビットのデータ長を有し、対応するブロックDynamicInfo()の情報が有効になる時刻をPTSにより示す。ストリーム毎に先頭となる時刻も、フィールドpts_change_pointで示され、これは、ファイル"XXXXX.CLP"内で定義される、上述したフィールドpresentation_start_timeと等しくなる。

【0169】

図31を用いて、ブロックStaticInfo()の一例の内部構造について説明する。ブロック

StaticInfo()は、対応するエレメンタリストリームの種類により内容が異なる。対応するエレメンタリストリームの種類は、図30を用いて説明した、フィールドstream_idおよびフィールドprivate_stream_idの値に基づき判断できる。図31では、ブロックStaticInfo()が対応するエレメンタリストリームの種類がビデオストリーム、オーディオストリームおよびサブタイトル(字幕)ストリームの何れであるかを、if構文を用いてそれぞれ記述している。以下、ブロックStaticInfo()について、エレメンタリストリーム毎に説明する。

【0170】

エレメンタリストリームがビデオストリームであった場合、ブロックStaticInfo()は、それぞれ4ビットのデータ長を有するフィールドpicture_sizeおよびフィールドframe_rate、1ビットのデータ長を有するフラグcc_flagからなる。フィールドpicture_sizeおよびフィールドframe_rateは、当該ビデオストリームの画像のサイズおよびフレーム周波数をそれぞれ示す。フラグcc_flagは、当該ビデオストリームがクローズドキャプションを含むか否かを示す。例えば、フラグcc_flagの値が"1"で、当該ビデオストリームがクローズドキャプションを含む。フィールドreserved_for_word_alignmentは、データ配置を16ビットに揃えるために用いられる。

【0171】

エレメンタリストリームがオーディオストリームであった場合、ブロックStaticInfo()は、16ビットのデータ長を有するフィールドaudio_language_code、8ビットのデータ長を有するフィールドchannel_configuration、1ビットのデータ長を有するフラグlfe_existanceおよび4ビットのデータ長を有するフィールドsampling_frequencyからなる。フィールドaudio_language_codeは、当該オーディオストリームに含まれている言語を表すコードを示す。フィールドchannel_configurationは、モノラル、ステレオ、マルチチャンネルなど、オーディオデータのチャンネル属性を示す。フィールドlfe_existanceは、低域強調チャンネルが含まれているか否かを示し、例えば値が"1"で、含まれていることを示す。フィールドsampling_frequencyは、オーディオデータのサンプリング周波数を示す。フィールドreserved_for_word_alignmentは、データ配置を16ビットに揃えるために用いられる。

【0172】

エレメンタリストリームがサブタイトル(字幕)ストリームであった場合、ブロックStaticInfo()は、16ビットのデータ長を有するフィールドsubtitle_language_code、4ビットのデータ長を有するフィールドsubtitle_presentation_typeおよび1ビットのデータ長を有するフラグconfigurable_flagからなる。フィールドsubtitle_language_codeは、当該字幕ストリームに含まれている言語を表すコードを示す。フィールドsubtitle_presentation_typeは、当該字幕ストリームが通常の字幕であるか、コメンタリ用字幕(映像の説明用などの特殊な字幕)であるかを示す。フラグconfigurable_flagは、当該字幕ストリームを行事する際に、文字の大きさや位置の変更を許可するか否かを示し、例えば値が"1"で、許可することを示す。フィールドreserved_for_word_alignmentは、データ配置を16ビットに揃えるために用いられる。

【0173】

図32を用いて、ブロックDynamicInfo()の一例の内部構造について説明する。ブロックDynamicInfo()は、先頭に、8ビットのデータ長を有するフィールドreserved_for_word_alignmentが配される。続く内容は、対応するエレメンタリストリームの種類により異なる。対応するエレメンタリストリームの種類は、図30を用いて説明した、フィールドstream_idおよびフィールドprivate_stream_idの値に基づき判断できる。図32では、ブロックDynamicInfo()が対応するエレメンタリストリームの種類がビデオストリーム、オーディオストリームおよびサブタイトル(字幕)ストリームの何れであるかを、if構文を用いてそれぞれ記述している。以下、ブロックDynamicInfo()について、エレメンタリストリーム毎に説明する。

【0174】

エレメンタリストリームがビデオストリームであった場合、ブロックDynamicInfo()は、4ビットのデータ長を有するフィールドdisplay_aspect_ratioからなる。フィールドdisplay_aspect_ratioは、ビデオの表示出力アスペクト比が16:9か4:3かを示す。フィールドreserved_for_word_alignmentは、データ配置を16ビットに揃えるために用いられる。

【0175】

エレメンタリストリームがオーディオストリームであった場合、ブロックDynamicInfo()は、4ビットのデータ長を有するフィールドchannel_assignmentからなる。フィールドchannel_assignmentは、当該オーディオストリームが2チャンネルで構成されている場合に、出力がステレオかデュアルモノかを示す。デュアルモノは、例えば2ヶ国語の音声を再生可能とする際に用いられる。フィールドreserved_for_word_alignmentは、データ配置を16ビットに揃えるために用いられる。

10

【0176】

エレメンタリストリームが字幕ストリームであった場合、ブロックDynamicInfo()は、データ配置を16ビットに揃えるために用いられる、フィールドreserved_for_word_alignmentで構成される。すなわち、字幕ストリームに関しては、動的に変化する属性が定義されていない。

【0177】

図33を用いて、ブロックEP_map()の一例の内部構造について説明する。ブロックEP_map()は、エレメンタリストリーム毎に、ビットストリーム内のデコード開始可能位置(エントリポイント)を、時刻情報と位置情報とを用いて表したものである。位置情報は、例えばエレメンタリストリームが記録される記録媒体における、アクセスの最小単位を用いることができる。各エレメンタリストリームは、ブロックEP_map()で示された位置からのデコード処理が可能であるものとする。

20

【0178】

固定レートのストリームでは、デコード開始可能位置を計算で求めることができるので、このブロックEP_map()のような情報は、不要である。一方、可変レートのストリームや、MPEG系のビデオの圧縮符号化方式のようにアクセスユニット毎にデータのサイズが変わるようなストリームの場合には、ランダムアクセスを行うために重要な情報となる。

【0179】

ブロックEP_map()は、先頭に、配置を16ビットに揃えるために、8ビットのデータ長を有するフィールドreserve_for_word_alignmentが配される。続いて、フィールドnumber_of_stream_id_entriesが配される。フィールドnumber_of_stream_id_entriesは、8ビットのデータ長を有し、このブロックEP_map()に記述されているエレメンタリストリームの数を示す。第1のforループにより、フィールドstream_id、フィールドprivate_stream_idおよびフィールドnumber_of_EP_entriesが、フィールドnumber_of_stream_id_entriesで示される回数分だけ、記述される。さらに、第1のforループの1回の記述毎に、第2のforループにより、フィールドnumber_of_EP_entriesで示される回数分だけ、フィールドPTS_EP_startおよびフィールドRPN_EP_startが配される。

30

【0180】

第1のforループ内において、最初に、それぞれ8ビットのデータ長を有するフィールドstream_idおよびフィールドprivate_stream_idが配され、図30に一例が示されるようにして、エレメンタリストリームを特定している。次に、配されるフィールドnumber_of_EP_entriesは、32ビットのデータ長を有し、当該エレメンタリストリームに対して記述されているエントリポイントの数を示す。その後、第2のforループにて、フィールドnumber_of_EP_entriesが示す数だけ、フィールドPTS_EP_startおよびフィールドRPN_EP_startがそれぞれ配される。

40

【0181】

フィールドPTS_EP_startおよびフィールドRPN_EP_startは、それぞれ32ビットのデータ長を有し、エントリポイント自体を表す。フィールドPTS_EP_startは、エントリポイン

50

トのクリップAVストリームファイル内での時刻をPTSで示す。一方、フィールドRPN_EP_startは、エントリポイントのクリップAVストリームファイル内での位置を例えば2048バイト単位で示す。

【0182】

この実施の一形態においては、ディスク状のアクセス単位である1セクタが2048バイトとされる。そのため、エントリポイントのクリップAVストリームファイル内での位置は、フィールドRPN_EP_startにより、セクタ単位で示されることになる。

【0183】

ここで、ビデオストリームの再生開始可能位置の直前には、必ず、パケットprivate_stream_2が配される。このパケットprivate_stream_2は、ビデオストリームをデコードするために利用可能な情報が格納されるパケットである。そのため、ビデオストリームのエントリポイントの位置は、当該パケットprivate_stream_2が格納されるパックpack()の位置とされる。

【0184】

ブロックEP_map()は、上述のようにして、クリップAVストリーム上の時刻と、クリップAVストリームファイル内での位置とを対応付けている。これにより、クリップAVストリームへのアクセスポイントの時刻情報(タイムスタンプ)が与えられたときに、クリップAVストリームファイルの中でデータの読み出しを開始すべきデータアドレスを検索することが容易となり、ディスクのランダムアクセスをスムーズに行うことができる。

【0185】

なお、この実施の一形態では、ブロックEP_map()において、エレメンタリストリーム毎の時刻情報と位置情報との組(第2のforループ内のフィールドPTS_EP_startとフィールドRPN_EP_startとの組)は、フィールドPTS_EP_startおよびRPN_EP_startの両方に対して昇順(または降順)に予め並べて登録するようにしている。換言すれば、時刻情報と位置情報とは、予め所定の方向に並べ替えられている。このため、このままのデータに対して二分検索を実行することが可能である。

【0186】

なお、この発明の実施の一形態では、ビデオのエレメンタリストリームは、MPEG2-Videoの規格に基づくエレメンタリストリームであるとして説明したが、これはこの例に限定されない。例えば、ビデオのエレメンタリストリームは、MPEG4-Visualや、MPEG4-AVCによるものでもよい。また、オーディオのエレメンタリストリームは、ATRACオーディオのエレメンタリストリームであるとして説明したが、これもこの例に限らず、例えばMPEG1/2/4オーディオにも適用可能である。

【0187】

7. ディスク再生装置について

次に、この発明の実施の一形態を適用可能なディスク再生装置について説明する。図34は、この発明を適用可能なディスク再生装置100の一例の構成を概略的に示す。バス111に対して、CPU(Central Processing Unit)112、メモリ113、ドライブインターフェイス114、入力インターフェイス115、ビデオデコーダ116、オーディオデコーダ117、ビデオ入出力インターフェイス118およびオーディオ入出力インターフェイス119がそれぞれ接続される。このディスク再生装置100の各部は、バス111を介してビデオストリーム、オーディオストリーム、各種コマンドやデータなどを互いにやりとりできるようになっている。

【0188】

ドライブインターフェイス114には、さらに、ディスクドライブ102が接続される。ディスクドライブ102は、ドライブインターフェイス114を介してバス111とデータやコマンドのやりとりを行う。

【0189】

CPU112は、ROM(Read Only Memory)およびRAM(Random Access Memory)を有し(図示しない)、ROMに予め記憶されたプログラムやデータに従い、バス111を介

10

20

30

40

50

してこのディスク再生装置 100 の各部とデータやコマンドのやりとりを行い、このディスク装置 100 の全体を制御する。RAM は、CPU 112 のワークメモリとして用いられる。

【0190】

入力インターフェイス 115 は、ユーザにより実際に入力操作が行われる入力装置からの入力信号が供給される。入力装置は、例えば、赤外線信号などで遠隔的にディスク再生装置 100 を操作するリモートコントロールコマンドや、このディスク再生装置 100 に直接的に設けられたキーなどである。入力インターフェイス 115 は、これらの入力装置から供給された入力信号を、CPU 112 に対する制御信号に変換して出力する。

【0191】

ディスク 101 は、図 23 以降で説明したようなフォーマットで以て、プレイリスト、スクリプトプログラム、クリップインフォメーションファイル、クリップ AV ストリームファイルなどが記録されている。ディスク 101 がディスクドライブ 102 に装填されると、自動再生またはユーザの入力操作に従いディスク 101 が再生される。ディスク 101 から読み出されたスクリプトファイルやプレイリストファイル、クリップインフォメーションファイルは、CPU 112 に供給され、例えば CPU 112 が有する RAM に記憶される。CPU 112 は、RAM に記憶されたこれらのデータやスクリプトプログラムに基づき、ディスク 101 からクリップ AV ストリームファイルを読み出す。

【0192】

ディスク 101 から読み出されたクリップ AV ストリームファイルは、メモリ 113 に一旦格納される。ビデオデコーダ 116 は、CPU 112 の命令に基づき、メモリ 113 に格納されたクリップ AV ストリームファイルのビデオストリームや字幕ストリームをデコードする。デコードされたビデオデータや字幕データは、例えば CPU 112 によりそれぞれ拡大、縮小処理などの画像処理を施されると共に、合成、加算処理を施され、1本のビデオデータとされる。これらの画像処理は、これに限らず、ビデオデコーダ 116 やビデオ出力インターフェイス 118 において行うこともできる。このビデオデータは、メモリ 113 にバッファリングされ、ビデオ出力インターフェイス 118 に供給される。ビデオ出力インターフェイス 118 は、例えば、供給されたビデオデータをアナログビデオ信号に変換して、ビデオ出力端子 120 に導出する。

【0193】

同様に、オーディオデコーダ 117 は、CPU 112 の命令に基づき、メモリ 113 に格納されたクリップ AV ストリームファイルのオーディオストリームをデコードする。デコードされたオーディオデータは、メモリ 113 にバッファリングされ、オーディオ出力インターフェイス 119 に供給される。オーディオ出力インターフェイス 119 は、供給されたオーディオデータを、例えばアナログオーディオ信号に変換してオーディオ出力端子 121 に導出する。

【0194】

なお、ここでは、図 34 に示される各部がそれぞれ独立したハードウェアで構成されているように説明したが、これはこの例に限定されない。例えば、ビデオデコーダ 116 および/またはオーディオデコーダ 117 は、CPU 112 上で動作するソフトウェアにより構成することができる。

【0195】

図 35 は、図 34 に示したディスク再生装置 100 における動作をより詳細に説明するための機能ブロック図である。ディスク再生装置 100 は、概略的には、オペレーションシステム 201 と、ビデオコンテンツ再生部 210 とからなる。ビデオコンテンツ再生部 210 は、実質的には、オペレーションシステム 201 上で動作するソフトウェアプログラムである。これに限らず、ビデオコンテンツ再生部 210 は、ソフトウェアとハードウェアが統合的に動作するものとしてもよい。以下では、ビデオコンテンツ再生部 210 がソフトウェアであるものとして説明する。なお、図 35 では、ディスクドライブ 102 は、省略されている。

10

20

30

40

50

【0196】

オペレーションシステム201は、ディスク再生装置100に電源が投入されるとCPU112において最初に起動し、各部の初期設定など必要な処理を行い、アプリケーションプログラム(ここではビデオコンテンツ再生部210)をROMから呼び出す。オペレーションシステム201は、ビデオコンテンツ再生部210の動作中に、ビデオコンテンツ再生部210に対して、ディスク101からのファイルの読み出しやファイルシステムの解釈といった、基本的なサービスを提供する。例えば、オペレーションシステム201は、ビデオコンテンツ再生部210から渡されたファイル読み出しリクエストに応じて、ドライブインターフェイス114を介してディスクドライブ102を制御し、ディスク101に記録されているデータを読み出す。読み出されたデータは、オペレーションシステム201の制御により、ビデオコンテンツ再生部210に渡される。

【0197】

また、オペレーションシステム201は、マルチタスク処理機能を備え、複数のソフトウェアモジュールを、例えば時分割制御により見かけ上並列的に制御することができる。すなわち、図35に一例が示される、ビデオコンテンツ再生部210を構成する各モジュールは、オペレーションシステム201のマルチタスク処理機能により、全て、並列的な動作が可能である。

【0198】

以下、ビデオコンテンツ再生部210の動作について、より具体的に説明する。ビデオコンテンツ再生部210は、内部にさらに幾つかのモジュールを有しており、下記の機能を実現する。

(1) 装填されたディスク101がUMDビデオの規格に準じたディスク(以下、UMDビデオディスクと呼ぶ)であるか否かを判断する。

(2) 装填されたディスク101がUMDビデオディスクであると判断した場合、ディスク101からスクリプトファイルを読み出して、スクリプト制御モジュール211に渡す。

(3) 装填されたディスク101がUMDビデオディスクであると判断した場合、さらに、データベースを構成するファイル(プレイリストファイル、クリップインフォメーションファイルなど)を読み出して、プレーヤ制御モジュール212に渡す。

【0199】

以下、ビデオコンテンツ再生部210の各モジュールの動作について説明する。

【0200】

スクリプト制御モジュール211は、スクリプトファイル"SCRIPT.DAT"に記述されているスクリプトプログラムを解釈して実行する。プレーヤモデルの説明で既に述べたように、メニュー画面などの画像の作成および出力や、ユーザ入力に応じたカーソル移動、メニュー画面の変更といったGUIは、スクリプトプログラムによりグラフィクス処理モジュール219を制御することで実現する。また、スクリプト制御モジュール211は、スクリプトプログラムの実行により、プレーヤ制御モジュール212の制御などが可能である。

【0201】

プレーヤ制御モジュール212は、ディスク101から読み出された、プレイリストファイル"PLAYLIST.DAT"や、クリップインフォメーションファイル"XXXXX.CLP"といったファイルに格納されたデータベース情報を参照して、ディスク101に記録されているビデオコンテンツの再生に関わる、以下のような制御を行う。

(1) プレイリストやクリップインフォメーションといったデータベース情報を解析する。

(2) コンテンツデータ供給モジュール213、デコード制御モジュール214およびバッファ制御モジュール215を制御する。

(3) スクリプト制御モジュール211または入力インターフェイス115からの指示に従い、再生、再生停止、再生一時停止といったプレーヤの状態遷移制御や、ストリーム切

10

20

30

40

50

り替えなどの再生制御処理を行う。

(4) デコード制御モジュール214から、再生中のビデオストリームについて、時刻情報を取得し、時刻表示やマークイベントの生成などを行う。

【0202】

コンテンツデータ供給モジュール213は、プレーヤ制御モジュール212の指示に従い、ディスク101からクリップAVストリームファイルといったコンテンツデータを読み出し、バッファ制御モジュール215に渡す。バッファ制御モジュール215は、渡されたコンテンツデータをバッファの実体215Aとしてのメモリ113に溜め込む。コンテンツデータ供給モジュール213は、バッファ制御モジュール215を制御し、ビデオデコーダ制御モジュール216、オーディオデコーダ制御モジュール217および字幕デコーダ制御モジュール218からの要求に従い、メモリ113に溜め込まれたコンテンツデータを、これらのモジュール216、217および218に所定に供給する。また、コンテンツデータ供給モジュール213は、バッファ制御モジュール215により溜め込まれるコンテンツデータの量を所定に制御するように、ディスク101からコンテンツデータの読み込みを行う。

【0203】

デコード制御モジュール214は、プレーヤ制御モジュール212の指示に従い、ビデオデコーダ制御モジュール216、オーディオデコーダ制御モジュール217および字幕デコーダ制御モジュール218の動作を制御する。また、デコード制御モジュール214は、内部に時計機能を有し、ビデオデータとオーディオデータとが同期的に出力されるように、各デコーダ制御モジュール216、217および218の動作を制御する。

【0204】

バッファ制御モジュール215は、バッファの実体215Aとして、メモリ113の一部を排他的に用いる。また、バッファ制御モジュール215は、データ先頭ポインタおよびデータ書き込みポインタを記憶する。バッファ制御モジュール215は、さらに、内部モジュールとしてビデオ読み出し機能、オーディオ読み出し機能および字幕読み出し機能を有する。ビデオ読み出し機能の内部には、ビデオ読み出しポインタを有する。また、ビデオ読み出し機能の内部には、アクセスユニット情報である情報au_information()を蓄積するためのレジスタを備える。オーディオ読み出し機能の内部には、オーディオ読み出しポインタを有する。字幕読み出し機能の内部には、字幕読み出しポインタと字幕読み出し機能フラグとを有する。字幕読み出し機能フラグは、書き込む値に応じて字幕読み出し機能の有効/無効を制御する。例えば、字幕読み出し機能フラグに"1"を書き込むと、字幕読み出し機能が有効とされ、"0"を書き込むと、字幕読み出し機能が無効とされる。

【0205】

バッファ制御モジュール215の内部モジュールであるビデオ読み出し機能、オーディオ読み出し機能および字幕読み出し機能は、さらに、ビデオストリーム、オーディオストリームおよび字幕ストリームが多重化されたクリップAVストリームから、それぞれのストリームを分離するデマルチプレクサ機能を有する。この発明の実施の一形態では、MPEG2システムのプログラムストリームの形式で複数のエレメンタリストリームが時分割多重されて、クリップAVストリームが形成されている。したがって、ビデオ読み出し機能、オーディオ読み出し機能および字幕読み出し機能は、MPEG2システムのプログラムストリームに対するデマルチプレクサ機能を有する。

【0206】

このため、ビデオ読み出し機能は、ストリーム内に所定に配置されるフィールドstream_id(図30参照)の値を読み取り、保持する。同様に、オーディオ読み出し機能および字幕読み出し機能は、フィールドstream_idおよびフィールドprivate_stream_id(図30参照)の値を読み取り、保持する。これらフィールドstream_idやフィールドprivate_stream_idの値は、供給されたビットストリームを解析する際に用いる。

【0207】

ビデオデコーダ制御モジュール216は、メモリ113からビデオストリームの単一の

10

20

30

40

50

ビデオアクセスユニットを読み出してビデオデコーダ 1 1 6 に供給するように、バッファ制御モジュール 2 1 5 内のビデオ読み出し機能に対して指示を出す。そして、ビデオデコーダ制御モジュール 2 1 6 は、ビデオデコーダ 1 1 6 を制御して、ビデオデコーダ 1 1 6 に供給されたビデオストリームをアクセスユニット単位でデコードする。ビデオストリームをデコードして生成されたビデオデータは、グラフィクス処理モジュール 2 1 9 に供給される。

【 0 2 0 8 】

同様に、オーディオデコーダ制御モジュール 2 1 7 は、メモリ 1 1 3 からオーディオストリームの単一のオーディオアクセスユニットを読み出してオーディオデコーダ 1 1 7 に供給するように、バッファ制御モジュール 2 1 5 内のオーディオ読み出し機能に対して指示を出す。なお、この実施の一形態では、オーディオストリームを構成するアクセスユニット（オーディオフレーム）は、既知の固定長とする。そして、オーディオデコーダ制御モジュール 2 1 7 は、オーディオデコーダ 1 1 7 を制御して、オーディオデコーダ 1 1 7 に供給されたオーディオストリームをアクセスユニット単位でデコードする。オーディオストリームをデコードして生成されたオーディオデータは、オーディオ出力モジュール 2 4 2 に供給される。

【 0 2 0 9 】

さらに、字幕デコーダ制御モジュール 2 1 8 は、メモリ 1 1 3 から字幕ストリームの単一の字幕アクセスユニットを読み出して字幕デコーダ制御モジュール 2 1 8 に供給するように、バッファ制御モジュール 2 1 5 内の字幕読み出し機能に対して指示を出す。なお、この実施の一形態では、字幕ストリームを構成する字幕アクセスユニットは、ユニットの先頭に当該ユニットの長さ情報が格納されている。字幕デコーダ制御モジュール 2 1 8 は、字幕デコード機能を有し、供給された字幕ストリームをデコードすることができる。字幕デコーダ制御モジュール 2 1 8 の字幕デコード機能により字幕ストリームがデコードされた字幕の画像データは、グラフィクス処理モジュール 2 1 9 に供給される。

【 0 2 1 0 】

グラフィクス処理モジュール 2 1 9 は、上述したように、ビデオデコーダ制御モジュール 2 1 6 の制御に基づきビデオデコーダ 1 1 6 でデコードされたビデオデータと、字幕デコーダ制御モジュール 2 1 8 によりデコードされた字幕の画像データとが供給される。グラフィクス処理モジュール 2 1 9 は、供給されたこれらビデオデータに対して字幕の画像データを所定に加算し、出力するためのビデオ信号を生成する。グラフィクス処理モジュール 2 1 9 では、さらに、スクリプト制御モジュール 2 1 1 やプレーヤ制御モジュール 2 1 2 の指示に従い、メニュー画像やメッセージ画像を生成し、出力ビデオ信号に対して合成（オーバーレイ）する。

【 0 2 1 1 】

例えば、グラフィクス処理モジュール 2 1 9 は、供給された字幕の画像データに対して、スクリプト制御モジュール 2 1 1 からの指示に従い拡大処理や縮小処理を行い、ビデオデータに対して所定に加算する。

【 0 2 1 2 】

また、グラフィクス処理モジュール 2 1 9 は、予め指定された出力ビデオデバイスのアスペクトレシオと、ディスク 1 0 1 から再生されたコンテンツ内で指定された出力アスペクトレシオとに基づき、出力信号のアスペクト変換を行う。例えば、出力ビデオデバイスのアスペクトレシオが 1 6 : 9 である場合、出力アスペクトレシオが 1 6 : 9 であれば、ビデオデータをそのまま出力し、出力アスペクトレシオが 4 : 3 であれば、出力されるビデオデータを、画像の高さが出力ビデオデバイスの画面高さに一致するようにスクイーズ（縮小）処理し、画像の左右に黒画像を挿入して出力する。出力ビデオデバイスが 4 : 3 である場合には、出力アスペクトレシオが 4 : 3 であれば、ビデオデータをそのまま出力し、出力アスペクトレシオが 1 6 : 9 であれば、出力されるビデオデータを、画像の幅が出力ビデオデバイスの画面幅に一致するようにスクイーズ処理し、画像の上下に黒画像を挿入して出力する。

10

20

30

40

50

【0213】

グラフィクス処理モジュール219は、さらに、プレーヤ制御モジュール212からの要求に応じて、現在処理中のビデオ信号をキャプチャし、プレーヤ制御モジュール212に返すような処理も行う。

【0214】

ビデオ出力モジュール241は、メモリ113の一部を排他的に占有してFIFO (First In First Out)のバッファとして用い、グラフィクス処理モジュール219により処理されたビデオデータをこのバッファに一時的に溜め込み、所定のタイミングで読み出す制御を行う。バッファから読み出されたビデオデータは、ビデオ出力インターフェイス118から出力される。

10

【0215】

オーディオ出力モジュール242は、メモリ113の一部を排他的に占有してFIFOのバッファとして用い、オーディオデコード119から出力されたオーディオデータをこのバッファに溜め込み、所定のタイミングで読み出す制御を行う。バッファから読み出されたオーディオデータは、オーディオ出力インターフェイス119から出力される。

【0216】

また、オーディオ出力モジュール242は、コンテンツのオーディオモードがデュアルモノ（例えば2ヶ国語）であった場合、予め指定された音声出力モードに従いオーディオデータを出力する。音声出力モードが「主音声」に指定されている場合、例えばメモリ113において左チャンネルのオーディオデータを右チャンネルにもコピーして、2チャンネルの出力を両方とも左チャンネルのオーディオデータとして出力する。音声出力モードが「副音声」であった場合には、例えばメモリ113において右チャンネルのオーディオデータを左チャンネルにもコピーして、2チャンネルの出力を両方とも右チャンネルのオーディオデータとして出力する。音声出力モードが「主・副音声」である場合や、コンテンツがステレオである場合には、オーディオデータをそのまま出力する。

20

【0217】

このような音声出力モードの設定は、ビデオコンテンツ再生部210が生成するメニュー画面などにより、ユーザが対話的に行うことができるようになっている。

【0218】

不揮発性メモリ制御モジュール250は、プレーヤ制御モジュール212からの指示により、ビデオコンテンツ再生部210が終了しても消去されない領域へのデータの書き込みや、当該領域からのデータの読み出しを行う。タイトル識別ID (Title_ID)をキーとして、データSaved_Player_StatusおよびデータSaved_User_Dataの組を複数件、当該領域に記憶する機能を有する。データSaved_Player_Statusとして、プレーヤ制御モジュール212の持つデータBackup_Player_Statusが記憶される。このデータBackup_Player_Statusは、例えば上述したプレーヤステータス323Bの、プレーヤ制御モジュール212が終了する直前のデータに対応し、データSaved_Player_Statusは、リジュームインフォメーション324に対応する。また、データSaved_User_Dataとして、プレーヤ制御モジュール212が持つデータUser_Dataが記憶される。データUser_Dataは、ユーザによりプレーヤ制御モジュール212に対して設定された所定のデータである。

30

40

【0219】

例えば、ディスク再生装置100が不揮発性のメモリであるフラッシュメモリなどを有する場合、不揮発性メモリ制御モジュール250は、このフラッシュメモリの所定領域にこれらデータSaved_Player_StatusおよびデータSaved_User_Dataの組を、ディスク101のタイトルIDと関連付けて記憶する。不揮発性メモリ制御モジュール250がデータを記憶する記憶媒体は、フラッシュメモリに限らず、例えばハードディスクなどでもよい。

【0220】

8. 音声および字幕ストリームの自動選択について

次に、この発明の実施の一形態による音声および字幕ストリームの自動選択について説明する。この発明の実施の一形態では、言語の異なる複数の音声ストリームおよび言語の

50

異なる複数の字幕ストリームが多重化されている場合に、音声の自動選択によって適切な言語の音声ストリームが選択されるようにすると共に、字幕の自動選択によって適切な言語の字幕ストリームが選択されるようにする。

【0221】

また、この発明では、プレーヤの音声言語設定に、従来の、「日本語」、「英語」といった言語名による指定に加えて、「オリジナル言語」という状態を設定できるようにする。さらに、プレーヤの音声言語設定が「オリジナル言語」に設定されている場合でも、音声の自動選択によって、適切な言語の音声ストリームが選択されるようにする。さらにまた、字幕と音声とを共に自動選択した場合に、音声と字幕とで同一の言語が選択されてしまう場合を回避するようにした。

10

【0222】

なお、「オリジナル言語」とは、コンテンツが制作されたときの最初の言語のことをいう。一例として、コンテンツが日本で制作されたもので、主に日本語が用いられていれば、オリジナル言語は、日本語とされる。オリジナル言語は、コンテンツ制作者側で設定することができる。

【0223】

図36～図38を用いて、この発明の実施の一形態による音声および字幕ストリームの自動選択について、概略的に説明する。なお、図36～図38では、ディスクに含まれるストリーム501において(1)で示される音声オリジナル言語であるものとする。また、この例では、字幕についてはオリジナル言語を定義していない。

20

【0224】

第1の例として、図36を参照し、プレーヤ設定500で音声を「オリジナル言語」、字幕を「日本語」とした場合について考える。そして、ディスクに収録されているコンテンツがアメリカで制作された映画であって、ディスクに含まれるストリーム501は、音声に関しては(1)英語(=オリジナル言語)、(2)日本語の2種類、字幕に関しては(1)英語、(2)日本語の2種類が収録されているものとする。この場合は、ディスクをプレーヤで再生した直後に選択されるストリーム502は、音声としてはオリジナル言語の(1)英語、字幕としては(2)日本語となる。この図36の例では、音声および字幕において適切な選択がなされ、問題は生じない。

【0225】

第2の例として、図37を参照し、プレーヤ設定500は、第1の例と同じ音声を「オリジナル言語」、字幕を「日本語」とされている。そして、ディスクに収録されているコンテンツが日本で制作された映画であって、ディスクに含まれるストリーム501は、音声に関しては(1)日本語(=オリジナル言語)、(2)英語の2種類、字幕に関しては(1)英語、(2)日本語の2種類が収録されているものとする。この場合は、ディスクをプレーヤで再生した直後に選択されるストリーム502は、音声としてはオリジナル言語の(1)日本語、字幕としても(2)日本語となり、重複した言語が選択されてしまう。この場合、日本語の字幕は不要であり、ユーザにとって使い勝手が悪いと考えられる。

30

【0226】

この第2の例のような問題を解決するためには、図38に一例が示されるように、プレーヤ設定500において音声言語が「オリジナル言語」に設定されており、且つ、オリジナル言語に対応する言語と字幕の言語とが一致した場合は、選択されるストリーム502に例示されるように、字幕の表示をOFFにするような機能が実現されればよい。

40

【0227】

すなわち、音声オリジナル言語に指定されている場合において、自動選択された結果の音声の言語と字幕の言語とが一致した場合、字幕の表示および非表示を指示するプレーヤの属性情報(プロパティ323中のプレーヤステータス323B、図3参照)に含まれるプロパティsubTitleFlagの値を、字幕OFFを示す値とする。このルールを導入することにより、音声および字幕の言語選択時において、音声と字幕とで同一言語のストリームが出力されることを回避することができる。

50

【 0 2 2 8 】

以下、この発明の実施の一形態による音声および字幕ストリームの自動選択について、より具体的に説明する。先ず、音声および字幕を適切に自動選択するために、プレーヤ装置側が持つべき情報について説明する。ストリーム再生を司るムービープレーヤ300のプロパティ323(図3参照)は、図7を用いて既に説明した。この図7で示したプロパティ323のうち、この実施の一形態に特に関係するものは、音声については、プロパティaudioLanguageCode、プロパティaudioNumberおよびプロパティaudioFlagであり、字幕については、プロパティsubtitleLanguageCode、プロパティsubtitleNumberおよびプロパティsubtitleFlagである。

【 0 2 2 9 】

プロパティaudioLanguageCodeは、上述したように、UMDビデオプレーヤに設定された、オーディオ言語の言語コードを示す。換言すれば、このプロパティaudioLanguageCodeは、音声として選択されるべき言語を表す。プロパティaudioLanguageCodeの値は、例えばISO(International Organization for Standardization)639-1で定義された言語コードを用いることができる。ISO639-1では、例えば英語は"en"、日本語は"ja"と表される。この発明の実施の一形態では、プロパティaudioLanguageCodeの値として、さらに、ISO639-1では定義されていない値"00"を追加し、この値"00"を以て「オリジナル言語」を表すものとする。

【 0 2 3 0 】

プロパティaudioNumberは、上述したように、現在再生中のオーディオストリームの番号を示す。このプロパティaudioNumberは、音声ストリームのストリームID(stream_id)と、プライベートストリームID(private_stream_id)とが並べて表される16ビットの値である(図30参照)。なお、16ビットのうち、上位8ビットがストリームIDであり、下位8ビットがプライベートストリームIDである。クリップAVストリームファイル中の音声ストリームは、この16ビットの値で、一意に指定することができる。

【 0 2 3 1 】

プロパティaudioFlagは、音声ストリームの再生情報を表す。図39は、プロパティaudioFlagが取り得る一例の値を示す。プロパティaudioFlagの値が"0"のときは、音声再生がOFFとされ、音声再生されない。値が"1"のときは、上述したプロパティaudioNumberで指定される音声ストリームが再生される。このとき、プロパティaudioNumberで指定される音声ストリームがデュアルモノ、すなわち、ステレオにおける2チャンネルのうち、左チャンネルと右チャンネルとに別々の内容の音声収録されている音声ストリームである場合を考慮して、値が"2"のときはデュアルモノの左チャンネルのみ、値が"3"のときは右チャンネルのみをそれぞれ再生する。音声ストリームがデュアルモノで、且つ、プロパティaudioFlagの値が"1"であれば、デュアルモノの左右両チャンネルが共に再生される。

【 0 2 3 2 】

プロパティsubtitleLanguageCodeは、上述したように、UMDビデオプレーヤに設定された、字幕(サブタイトル)言語の言語コードを示す。換言すれば、このプロパティsubtitleLanguageCodeは、字幕として選択すべき言語を示す。プロパティsubtitleLanguageCodeの値は、上述の音声ストリームの場合と同様に、例えばISO(International Organization for Standardization)639-1で定義された言語コードを用いることができる。

【 0 2 3 3 】

プロパティsubtitleNumberは、上述したように、現在再生中の字幕ストリームの番号を示す。このプロパティsubtitleNumberは、字幕ストリームのストリームID(stream_id)と、プライベートストリームID(private_stream_id)とが並べて表される16ビットの値である(図30参照)。なお、16ビットのうち、上位8ビットがストリームIDであり、下位8ビットがプライベートストリームIDである。クリップAVストリームファイル中の字幕ストリームは、この16ビットの値で、一意に指定することができる。

【 0 2 3 4 】

10

20

30

40

50

プロパティ subtitleFlag は、字幕ストリームの再生状態を表す。図 4 0 は、プロパティ subtitleFlag の取り得る一例の値を示す。プロパティ subtitleFlag の値が " 0 " のときは、字幕表示が OFF とされ字幕が表示されない。値が " 1 " で字幕表示が ON とされ、字幕が表示される。

【 0 2 3 5 】

次に、音声および字幕を適切に自動選択するために、コンテンツ側が持つべき情報について説明する。コンテンツ側では、音声ストリームの属性情報と、字幕ストリームの属性情報とを、クリップインフォメーションファイルに記述しておく。上述したが、クリップインフォメーションファイルは、図 2 9 および図 3 1 などを用いて既に説明したように、対応するクリップ AV ストリーム中に多重化されている各エレメンタリストリームについて、ストリームを一意に識別するためのフィールド stream_id およびフィールド private_stream_id の情報と、図 3 1 に示される属性情報が並べられている。図 3 1 に示される属性情報のうち、この実施の一形態に特に関連性の高いものは、音声ストリームに関してはフィールド audio_language_code であり、字幕ストリームに関してはフィールド subtitle_language_code である。

【 0 2 3 6 】

なお、クリップ AV ストリームファイルには、複数の言語の音声ストリームを多重化することができる。そのため、多重化された複数言語の音声ストリームのうち、オリジナル言語がどの音声ストリームであるかを示す必要がある。この実施の一形態では、図 2 9 で示したクリップインフォメーションファイルにおいて for ループにより並べられているストリームの並び順で、最初の音声ストリームの言語がオリジナル言語であると定義する。オリジナル言語の言語コードは、当該音声ストリームに対応するフィールド audio_language_code (図 3 1 参照) で表される。

【 0 2 3 7 】

ムービープレーヤ 3 0 0 のプロパティ 3 2 3 (プレーヤステータス 3 2 3 B) は、スク립トレイヤ 3 0 2 から、図 8 を用いて既に説明したメソッドにより設定することができる。図 8 のメソッドのうち、音声ストリームと、字幕ストリームの設定に関連するメソッドは、メソッド play()、メソッド playChapter() およびメソッド changeStream() である。なお、これらのうち、メソッド changeStream() は、指定のストリームに変更するためのメソッドであり、ストリームの自動選択の仕組みはないため、ここでの説明は省略する。また、メソッド playChapter() は、以下に説明するメソッド play() の引数の一つである引数 playListTime を引数 chapterNumber に変えたものに相当するので、ここでは、メソッド play() についてのみ、説明する。

【 0 2 3 8 】

図 4 1 は、メソッド play() の一例の引数を一覧して示す。メソッド play() に対してこれらの引数を所定に与えることにより、ストリーム番号を設定した上で、ビデオストリーム、音声ストリームおよび字幕ストリームの再生を開始することができる。メソッド play() の文法は、例えば式 (1) のようになる。すなわち、引数のそれぞれをデリミタ (この例ではカンマ「 , 」) で区切り、所定の順序に並べて記述される。引数を省略する場合は、デリミタのみが記述される。

```
movieplayer.play(playListNumber,playListTime,menuMode,pauseMode,video_strm,audio_strm,audio_status,subtitle_strm,subtitle_status)    ・ ・ ( 1 )
```

【 0 2 3 9 】

引数 playListNumber は、再生するプレイリスト番号を表す。引数 playListTime は、プレイリストの先頭からの時刻を指定する。プレイリストの先頭から再生を開始する場合は、値 " 0 " を指定する。引数 menuMode は、値 " true " と値 " false " の何れかを取り、値 " true " の場合、メニューモードでの再生を指定し、値 " false " の場合、ノーマルモードでの再生を指定する。引数 pauseMode は、値 " true " と値 " false " の何れかを取り、値 " true " の場合、一時停止での待機を指定し、値 " false " の場合、通常速度での再生開始を指定する。引数 video_strm は、再生するビデオストリームを指定する。この引数 video_strm では、値 " - 1 " と

10

20

30

40

50

値"- 2"を取ることができ、値"- 1"は、再生するビデオストリームをムービープレーヤ300による自動選択で指定し、値"- 2"で再生するビデオストリームに変更が無いことを指定する。

【0240】

引数audio_strm以下が、音声ストリームおよび字幕ストリームの再生に関する引数で、この実施の一形態と関連性の強いものである。引数audio_strmは、ムービープレーヤ300のプロパティaudioNumber(図7参照)に値をセットするための引数である。引数audio_strmは、ストリームID(stream_id)およびプライベートストリームID(private_stream_id)の16ビットで表される音声ストリーム番号を、プロパティaudioNumberにセットできる。さらに、引数audio_strmは、値"- 1"および値"- 2"を引数に与えることができる。値"- 1"は、ムービープレーヤ300内で適切な音声ストリームを自動選択して、選択された音声ストリームの音声ストリーム番号をプロパティaudioNumberにセットすることを指示する。値"- 2"は、現在設定されているプロパティaudioNumberの値を変えずに再生開始することを指示する。

10

【0241】

引数audio_statusは、ムービープレーヤ300のプロパティaudioFlag(図7参照)を設定するためのパラメータである。値"0"、値"1"、値"2"および値"3"については、引数audio_statusの値がそのままプロパティaudioFlagにセットされる。引数audio_statusが値"- 2"の場合は、現在のプロパティaudioFlagの値を維持する。

【0242】

引数subtitle_strmは、ムービープレーヤ300のプロパティsubtitleNumber(図7参照)に値をセットするための引数である。引数subtitle_strmは、ストリームID(stream_id)およびプライベートストリームID(private_stream_id)の16ビットで表される字幕ストリーム番号を、プロパティsubtitleNumberにセットできる。さらに、引数subtitle_strmは、値"- 1"および値"- 2"を引数に与えることができる。値"- 1"は、ムービープレーヤ300内で適切な字幕ストリームを自動選択して、選択された字幕ストリームの字幕ストリーム番号をプロパティsubtitleNumberにセットすることを指示する。値"- 2"は、現在設定されているプロパティsubtitleNumberの値を変えずに再生開始することを指示する。

20

【0243】

引数subtitle_statusは、ムービープレーヤ300のプロパティsubtitleFlag(図7参照)を設定するためのパラメータである。値"0"および値"1"については、引数subtitle_statusの値がそのままプロパティsubtitleFlagにセットされる。引数subtitle_statusが値"- 1"の場合には、プロパティsubtitleFlagの値は、ムービープレーヤ300内で適切な値に自動設定される。引数subtitle_statusが値"- 2"の場合は、現在のプロパティsubtitleFlagの値を維持する。

30

【0244】

プロパティaudioFlagにムービープレーヤ300による自動設定が無く、プロパティsubtitleFlagに自動設定があるのは、音声と字幕が同一の言語に設定された場合、字幕の表示を自動的にOFFして字幕を表示されなくするためである。

40

【0245】

次に、音声ストリームおよび字幕ストリームの自動選択処理について、より具体的に説明する。図42は、複数種類の音声ストリームおよび字幕ストリームを自動選択する際の処理の流れを、概略的に示す。ディスクがプレーヤ装置に装填されると、先ず、ユーザにより再生するビデオストリームが選択される(ステップS80)。選択されたビデオストリームに対応する音声ストリームがステップS81で自動選択され、選択結果に応じて次のステップS82でプロパティaudioFlagが設定される。プロパティaudioFlagを参照し、ステップS83で字幕ストリームが選択される。そして、選択結果に応じて次のステップS84でプロパティsubtitleFlagが設定される。

【0246】

50

図42におけるステップS81による音声ストリーム自動選択の一例の処理を、図43および図44のフローチャートを用いてより詳細に説明する。この音声ストリームの自動選択処理においては、音声ストリームの属性情報のうち(1)言語コード、(2)チャンネル数、(3)クリップインフォメーションファイル内での並び順、の3つが用いられる。これらのうち、(1)の言語コードが最も優先順位が高く、(2)のチャンネル数が2番目の優先順位、(3)のクリップインフォメーションファイル内での並び順は、最も優先順位が低い。

【0247】

なお、図43および図44において、符号Aは、図43および図44間で対応する符号に処理が移行することを示す。

【0248】

音声ストリームの選択処理が開始されると(ステップS90)、メソッドplay()において再生する音声ストリームのストリーム番号を表す引数audio_strmに関して、値が"-1"に設定されているか、または、値が省略されているか否かが判断される(ステップS91)。上述したように、引数audio_strmが値"-1"で、音声ストリームの自動選択が指定される。

【0249】

若し、引数audio_strmに値"-1"以外の値が設定されていると判断されれば、処理はステップS92に移行され、引数audio_strmの値をプロパティaudioNumberに設定し、引数audio_strmで指定された音声ストリームを選択する。そして、図42のステップS82に処理が移行し、選択された音声ストリームに応じて、ムービープレーヤ300においてプロパティaudioFlagが所定に設定される。このとき、選択された音声ストリームのオーディオチャンネル数が5.1チャンネルの場合、例えばプロパティaudioFlagに値"1"が設定され、音声ストリームの再生を行うか否かのみを指定する。ステップS92で当該音声ストリームが選択できなければ、処理はエラーとなる。

【0250】

なお、処理がエラーとなった場合の実際の動作は、UMDビデオプレーヤの実装に依存する。例えば、エラーの場合には、この図43のフローチャートによる処理が終了され、プロパティaudioNumberは変更されないまま、次の処理に移行される。このようなエラー処理は、図43および図44における他のエラー処理についても、同様とする。

【0251】

ステップS91で、引数audio_strmが値"-1"または省略されていると判断されれば、処理はステップS93に移行される。ステップS93では、ムービープレーヤ300のプロパティaudioNumberで指定される音声ストリームが存在しないか、または、プロパティaudioNumberが不定値であるか否かが判断される。若し、プロパティaudioNumberで指定される音声ストリームが存在すれば、処理はステップS94に移行され、プロパティaudioNumberで指定された音声ストリームが選択される。そして、図42のステップS82に処理が移行し、ムービープレーヤ300において、選択された音声ストリームに応じてプロパティaudioFlagが所定に設定される。当該音声ストリームが選択できなければ、処理はエラーとなる。

【0252】

ステップS93で、ムービープレーヤ300のプロパティaudioNumberで指定される音声ストリームが存在しないか、または、プロパティaudioNumberが不定値であると判断されれば、処理はステップS95に移行される。このステップS95以下の処理が、具体的な音声ストリームの自動選択処理となる。なお、図42のステップS81による音声ストリームの自動選択の処理は、図43のフローチャートで示される処理全体を示し、図43のフローチャート中のステップS95以下の処理が、引数audio_strmが自動選択を表す値"-1"のときに実行される、具体的な音声ストリームの自動選択処理となる。

【0253】

まず、上述した(1)の言語コードに基づく自動選択処理がなされる。この処理は、プ

10

20

30

40

50

レーヤの音声言語設定と同一言語の音声ストリームを優先的に選択させるための処理である。ステップS 9 6で、ムービープレーヤ3 0 0のプロパティaudioLanguageCodeの値が「オリジナル言語」を示す値"00"となっているか否かが判断される。若し、プロパティaudioLanguageCodeの値が「オリジナル言語」を示す値となっていると判断されれば、処理は図4 4のステップS 1 0 1に移行される。

【0 2 5 4】

ステップS 1 0 1～S 1 0 4では、コンテンツ側に設定されている「オリジナル言語」が何であるかを判断する。ステップS 1 0 1で、クリップインフォメーションファイル(図2 9参照)内のブロックStreamInfo()における音声ストリームの並び順が調べられ、1番目に配置される音声ストリームの言語コードが取得される。そして、当該言語コードと同一の言語コードを持ち、オーディオチャンネル数がUMDビデオプレーヤに設定されたオーディオチャンネル数と同じかそれ以下の音声ストリームが存在するか否かが判断される。若し、存在すると判断されれば、処理はステップS 1 0 2に移行し、存在しないと判断された場合は、処理はステップS 1 0 3に移行する。

10

【0 2 5 5】

ステップS 1 0 2では、ステップS 1 0 1における条件を満たす音声ストリームのうち、オーディオチャンネル数が最も大きい音声ストリームを選択する。例えば、UMDビデオプレーヤのオーディオチャンネル設定が5 . 1チャンネルであって、ブロックStreamInfo()の1番目に配置される音声ストリームと言語コードが同一の、オーディオチャンネル数が2チャンネルの音声ストリームと、5 . 1チャンネルの音声ストリームとが存在する場合、5 . 1チャンネルの音声ストリームが選択される。オーディオチャンネル数が同一の音声ストリームが複数存在するときは、当該ブロックStreamInfo()内の並び順でより前の方に配置される音声ストリームが選択される。そして、図4 2のステップS 8 2に処理が移行し、ムービープレーヤ3 0 0において、選択された音声ストリームに応じてプロパティaudioFlagが所定に設定される。

20

【0 2 5 6】

ステップS 1 0 3では、オーディオチャンネル数が、UMDビデオプレーヤに対して設定されたオーディオチャンネル数以下の音声ストリームが存在するか否かが判断される。若し、存在すれば、オーディオチャンネル数が最も大きい音声ストリームを選択する(ステップS 1 0 4)。オーディオチャンネル数が同一の音声ストリームが複数存在するときは、当該ブロックStreamInfo()内の並び順でより前の方に配置される音声ストリームが選択される。そして、図4 2のステップS 8 2に処理が移行し、ムービープレーヤ3 0 0において、選択された音声ストリームに応じてプロパティaudioFlagが所定に設定される。

30

【0 2 5 7】

ステップS 1 0 3で、存在しないと判断された場合は、図4 3および図4 4のフローチャートによる一連の処理が終了される。この場合、音声ストリームが選択されないまま、ステップS 8 2の処理に移行し、プロパティaudioFlagが所定に設定されることになる。

【0 2 5 8】

図4 3の説明に戻り、ステップS 9 6で、若し、ムービープレーヤ3 0 0のプロパティaudioLanguageCodeの値が「オリジナル言語」を示す値となっていないと判断されれば、処理はステップS 9 7に移行される。ステップS 9 7では、ムービープレーヤ3 0 0のプロパティaudioLanguageCodeと同一の言語コードを持ち、オーディオチャンネル数がUMDビデオプレーヤに設定されたオーディオチャンネル数と同じかそれ以下の音声ストリームが存在するか否かが判断される。若し、存在すると判断されれば、処理はステップS 9 8に移行し、存在しないと判断されれば、処理はステップS 9 9に移行する。

40

【0 2 5 9】

ステップS 9 8では、上述したステップS 1 0 2と同様にして、ステップS 9 7における条件を満たす音声ストリームのうち、オーディオチャンネル数が最も大きい音声ストリームを選択する。オーディオチャンネル数が同一の音声ストリームが複数存在するときは、当該ブロックStreamInfo()内の並び順でより前の方に配置される音声ストリームが選択

50

される。そして、図 4 2 のステップ S 8 2 に処理が移行し、ムービープレーヤ 3 0 0 において、選択された音声ストリームに応じてプロパティ audioFlag が所定に設定される。

【 0 2 6 0 】

ステップ S 9 9 では、上述のステップ S 1 0 3 と同様にして、オーディオチャンネル数が、UMD ビデオプレーヤに対して設定されたオーディオチャンネル数以下の音声ストリームが存在するか否かが判断される。若し、存在すれば、オーディオチャンネル数が最も大きい音声ストリームを選択する（ステップ S 1 0 0 ）。オーディオチャンネル数が同一の音声ストリームが複数存在するときは、当該ブロック StreamInfo() 内の並び順でより前の方に配置される音声ストリームが選択される。そして、図 4 2 のステップ S 8 2 に処理が移行し、ムービープレーヤ 3 0 0 において、選択された音声ストリームに応じてプロパティ audioFlag が所定に設定される。

10

【 0 2 6 1 】

ステップ S 9 9 で、存在しないと判断された場合は、図 4 3 および図 4 4 のフローチャートによる一連の処理が終了される。この場合、音声ストリームが選択されないまま、ステップ S 8 2 の処理に移行し、プロパティ audioFlag が所定に設定されることになる。

【 0 2 6 2 】

図 4 2 におけるステップ S 8 3 による字幕ストリーム自動選択の一例の処理を、図 4 5 のフローチャートを用いてより詳細に説明する。字幕ストリームの選択が開始されると（ステップ S 1 1 0 ）、ステップ S 1 1 1 で、メソッド play() において再生する字幕ストリームのストリーム番号を表す引数 subtitle_strm に関して、値が "- 1" に設定されているか、あるいは、値が省略されているか否かが判断される。上述したように、引数 subtitle_strm が値 "- 1" で、字幕ストリームの自動選択が指定される。

20

【 0 2 6 3 】

若し、引数 subtitle_strm に値 "- 1" 以外の値が設定されていると判断されれば、処理はステップ S 1 1 2 に移行され、引数 subtitle_strm の値をプロパティ subtitleNumber に設定し、引数 subtitle_strm で指定された字幕ストリームを選択する。そして、図 4 2 のステップ S 8 4 に処理が移行し、ムービープレーヤ 3 0 0 においてプロパティ subtitleFlag が設定される。ステップ 1 1 2 で字幕ストリームが選択できなければ、処理はエラーとなる。

【 0 2 6 4 】

なお、処理がエラーなった場合の実際の動作は、UMD ビデオプレーヤの実装に依存する。例えば、エラーの場合には、この図 4 5 のフローチャートによる処理が終了され、字幕ストリームが選択されないまま、次の処理に移行される。このようなエラー処理は、図 4 5 における他のエラー処理についても、同様とする。

30

【 0 2 6 5 】

ステップ S 1 1 1 で引数 subtitle_strm が値 "- 1" または省略されていると判断されれば、処理はステップ S 1 1 3 に移行される。ステップ S 1 1 3 では、ムービープレーヤ 3 0 0 のプロパティ subtitleNumber で指定される字幕ストリームが存在しないか、または、プロパティ subtitleNumber の値が不定値であるか否かが判断される。若し、プロパティ subtitleNumber で指定される字幕ストリームが存在すれば、処理はステップ S 1 1 4 に移行され、プロパティ subtitleNumber で指定される字幕ストリームが選択される。そして、図 4 2 のステップ S 8 4 に処理が移行される。ステップ S 1 1 4 で字幕ストリームが選択できなければ、処理はエラーになる。

40

【 0 2 6 6 】

ステップ S 1 1 3 で、ムービープレーヤ 3 0 0 のプロパティ subtitleNumber で指定される字幕ストリームが存在しないか、または、プロパティ subtitleNumber の値が不定値であると判断されれば、処理はステップ S 1 1 5 に移行される。このステップ S 1 1 5 以降の処理が、具体的な字幕ストリームの自動選択処理となる。

【 0 2 6 7 】

次のステップ S 1 1 6 では、クリップインフォメーションファイル内のブロック Stream

50

Info()における字幕ストリームが調べられ、字幕コードの言語コードが取得される。そして、ムービープレーヤ300のプロパティsubtitleLabelCodeと同一の言語コードを持つ字幕ストリームが存在するかが判断される。若し、存在すると判断されれば、処理はステップS117に移行し、当該字幕ストリームが選択される。ステップS116での条件を満たす字幕ストリームが複数存在するときは、クリップインフォメーションファイルにおけるブロックStreamInfo()内の並び順でより前の方に配置される字幕ストリームが選択される。そして、図42のステップS84に処理が移行する。

【0268】

ステップS116で、条件を満たす字幕ストリームが存在しないと判断されれば、処理はステップS118に移行する。ステップS118では、選択可能な字幕ストリームが存在しないとして、プロパティsubtitleFlagの値を、字幕を表示しないことを指定する値"0"に設定する。

10

【0269】

図42におけるステップS84によるプロパティsubtitleFlagの自動設定の一例の処理を、図46のフローチャートを用いてより詳細に説明する。プロパティsubtitleFlagの自動設定が開始されると(ステップS120)、ステップS121で、メソッドplay()の字幕再生の状態を表す引数subtitle_statusが、自動設定を示す値"-1"になっているか、あるいは、省略されているかが判断される。若し、引数subtitle_statusに、値"-1"以外の値が設定されていれば、処理はステップS122に移行し、引数subtitle_statusに設定されている値をプロパティsubtitleFlagに対して設定する。

20

【0270】

若し、引数subtitle_statusが値"-1"になっているか、あるいは、省略されてれば処理はステップS123に移行する。ステップS123では、ムービープレーヤ300のプロパティsubtitleLabelCodeで表される言語コードと、現在選択されている音声ストリームの言語コード(ムービープレーヤ300のプロパティaudioLanguageCode)とが一致し、且つ、プロパティaudioFlagの値が"0"ではない値、すなわち、プロパティaudioFlagが音声ストリームを再生するような値に設定されているかが判断される。若し、ステップS123の条件を満たしていると判断されれば、処理はステップS124に移行する。

【0271】

ステップS124では、プロパティsubtitleFlagが値"0"に設定され、字幕表示を行わないようにされる。すなわち、ステップS123の条件を満たすことは、音声ストリームで再生される音声の言語と、字幕ストリームにより表示される字幕の言語とが同一であることを意味する。そこで、プロパティsubtitleFlagの値を"0"として、字幕の表示を抑止する。

30

【0272】

ステップS123の条件を満たしていないと判断された場合、処理はステップS125に移行する。ステップS125では、プロパティsubtitleFlagが値"1"とされ、字幕表示を行うようにする。

【0273】

なお、図46では、ステップS123、S124およびS125において、字幕ストリームの言語コードに基づき字幕表示のON/OFFを設定している。これはこの例に限らず、字幕ストリームの属性をさらに利用して字幕表示のON/OFFを設定するようにしてもよい。例えば、字幕と音声の言語が一致していても、字幕ストリームの属性が通常の子幕ではない字幕、例えば上述したコメンタリ用字幕を表す場合、字幕の表示を抑止しない(プロパティsubtitleFlagの値を"1"にする)ようにすることが考えられる。これは、字幕ストリームがコメンタリ用であれば、字幕の表示を抑制する必要が無いためである。

40

【0274】

字幕ストリームの属性は、図31で既に説明したように、クリップAVストリームファイル"XXXXX.CLP"におけるブロックStaticInfo()内のフィールドsubtitle_presentation_t

50

ypeの値により表される。

【0275】

字幕ストリームの属性を用いて字幕表示のON/OFFを設定する場合の一例の処理を、図47のフローチャートを用いて説明する。図47において、上述した図46と共通する部分には同一の符号を付し、詳細な説明を省略する。ステップS123による判断で、ムービープレーヤ300のプロパティsubtitleLabelCodeで表される言語コードと、現在選択されている音声ストリームの言語コードとが一致し、且つ、プロパティaudioFlagの値が"0"ではない値に設定されていると判断されれば、処理はステップS126に移行する。

【0276】

ステップS126では、字幕ストリームの属性が通常の子幕(Normal)の属性を示しているか否かが判断される。若し、字幕ストリームの属性が通常の子幕の属性を示していると判断されれば、処理はステップS124に移行され、プロパティsubtitleLabelが値"0"に設定され、字幕表示を行わないようにされる。また、字幕ストリームの属性が通常の子幕の属性を示していないと判断されれば、処理はステップS125に移行され、プロパティsubtitleLabelが値"1"とされ、字幕表示を行うようにする。

【0277】

以上説明したようにして、この発明の実施の一形態による、音声および字幕の適切な自動選択が実現される。

【0278】

なお、上述では、言語の異なる複数の音声ストリームを多重化するように説明したが、これはこの例に限定されない。例えば、言語が同一で内容が異なる音声ストリームを多重化することもできる。一例として、同一の言語で内容の異なる音声ストリームAと音声ストリームBとを多重化し、一方(例えば音声ストリームA)をオリジナル言語とすることができる。

【図面の簡単な説明】

【0279】

【図1】UMDビデオ規格のレイヤ構成を示す略線図である。

【図2】この発明の実施の一形態による一例のプレーヤモデルを模式的に示す略線図である。

【図3】ムービープレーヤの一例の内部構成を示す略線図である。

【図4】ムービープレーヤの一例の内部構成を示す略線図である。

【図5】この発明の実施の一形態によるムービープレーヤのイベントモデルを模式的に示す略線図である。

【図6】プレイリストの再生中に発生する一例のイベントを示す略線図である。

【図7】ムービープレーヤオブジェクトが有する一例のプロパティを一覧して示す略線図である。

【図8】ムービープレーヤオブジェクトが有する一例のメソッドを一覧して示す略線図である。

【図9】ユーザ入力による一例のキー入力を示す略線図である。

【図10】ユーザ入力による一例のキー入力を示す略線図である。

【図11】キー入力に応じた一例の制御コマンドを示す略線図である。

【図12】キー入力に対応する一例のイベントを示す略線図である。

【図13】一例のイベントハンドラを示す略線図である。

【図14】一例のイベントハンドラを示す略線図である。

【図15】ユーザ入力イベントをきっかけとして、用意されたプログラムが実行される一例の処理を示すフローチャートである。

【図16】UMDビデオプレーヤにディスクがロードされてからイジェクトされるまでの処理を概略的に示すフローチャートである。

【図17】スクリプトファイルの構成例を示す略線図である。

10

20

30

40

50

【図18】 イベントハンドラ `onAutoPlay()` を実行する一例の手順を示すフローチャートである。

【図19】 イベントハンドラ `onContinuePlay()` を実行する一例の手順を示すフローチャートである。

【図20】 再生終了時の一例の処理を示すフローチャートである。

【図21】 スクリプトプログラムの例について説明するための図である。

【図22】 一例のスクリプトプログラムを示す略線図である。

【図23】 UMDビデオ規格に適用されるファイルの管理構造を

【図24】 ファイル "PLAYLIST.DAT" の全体構造を表す一例のシンタクスを示す略線図である。

【図25】 ブロック `PlayItem()` の一例の内部構造を示す略線図である。

【図26】 ブロック `PlayListMark()` の一例の内部構造を示す略線図である。

【図27】 ブロック `Mark()` 内のフィールド `mark_type` について説明するための図である。

【図28】 クリップAVストリームファイル内でのマーク時刻の指定について説明するための図である。

【図29】 クリップAVストリームファイル "XXXXX.CLP" の全体構造を表す一例のシンタクスを示す略線図である。

【図30】 ブロック `StreamInfo()` のエレメンタリストリームに対する関連付けを説明するための図である。

【図31】 ブロック `StaticInfo()` の一例の内部構造を示す略線図である。

【図32】 ブロック `DynamicInfo()` の一例の内部構造を示す略線図である。

【図33】 ブロック `EP_map()` の一例の内部構造を示す略線図である。

【図34】 この発明を適用可能なディスク再生装置の一例の構成を概略的に示すブロック図である。

【図35】 ディスク再生装置における動作をより詳細に説明するための機能ブロック図である。

【図36】 この発明の実施の一形態による音声および字幕ストリームの自動選択について説明するための図である。

【図37】 この発明の実施の一形態による音声および字幕ストリームの自動選択について説明するための図である。

【図38】 この発明の実施の一形態による音声および字幕ストリームの自動選択について説明するための図である。

【図39】 プロパティ `audioFlag` が取り得る一例の値を示す略線図である。

【図40】 プロパティ `subtitleFlag` の取り得る一例の値を示す略線図である。

【図41】 メソッド `play()` の一例の引数を一覧して示す略線図である。

【図42】 複数種類の音声ストリームおよび字幕ストリームを自動選択する際の処理の流れを概略的に示すフローチャートである。

【図43】 音声ストリーム自動選択の一例の処理をより詳細に説明するためのフローチャートである。

【図44】 音声ストリーム自動選択の一例の処理をより詳細に説明するためのフローチャートである。

【図45】 字幕ストリーム自動選択の一例の処理をより詳細に説明するためのフローチャートである。

【図46】 プロパティ `subtitleFlag` の自動設定の一例の処理を詳細に説明するためのフローチャートである。

【図47】 プロパティ `subtitleFlag` の自動設定の他の例の処理を説明するためのフローチャートである。

【符号の説明】

【0280】

101 ディスク

10

20

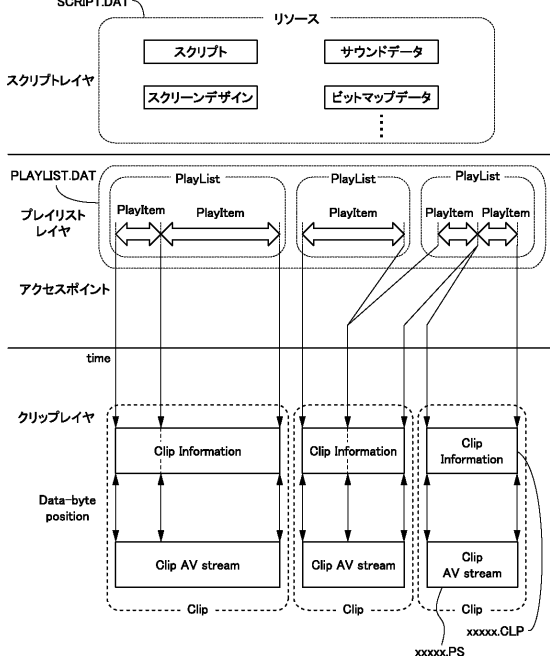
30

40

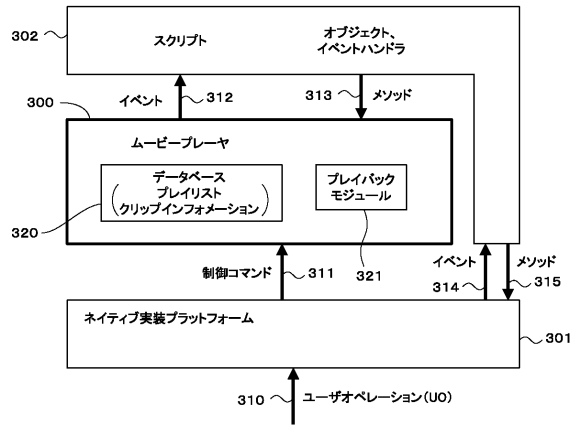
50

1 1 2	C P U	
1 1 3	メモリ	
1 1 5	入力インターフェイス	
1 1 6	ビデオデコーダ	
1 1 7	オーディオデコーダ	
1 1 8	ビデオ出力インターフェイス	
1 1 9	オーディオ出力インターフェイス	
2 0 1	オペレーションシステム	
2 1 0	ビデオコンテンツ再生部	
2 1 1	スクリプト制御モジュール	10
2 1 2	プレーヤ制御モジュール	
2 1 4	デコード制御モジュール	
2 1 5	バッファ制御モジュール	
2 1 6	ビデオデコーダ制御モジュール	
2 1 7	オーディオデコーダ制御モジュール	
2 1 8	字幕デコーダ制御モジュール	
2 1 9	グラフィクス制御モジュール	
2 4 1	ビデオ出力モジュール	
2 4 2	オーディオ出力モジュール	
2 5 0	不揮発性メモリ制御モジュール	20
3 0 0	ムービープレーヤ	
3 0 1	ネイティブ実装プラットフォーム	
3 0 2	スクリプトレイヤ	
3 1 0	ユーザ入力	
3 1 1	制御コマンド	
3 1 2	イベント	
3 1 3	メソッド	
3 2 0	データベース	
3 2 1	プレイバックモジュール	
3 2 2	デコーダエンジン	30
3 2 3	プロパティ	
3 2 4	リジュームインフォメーション	

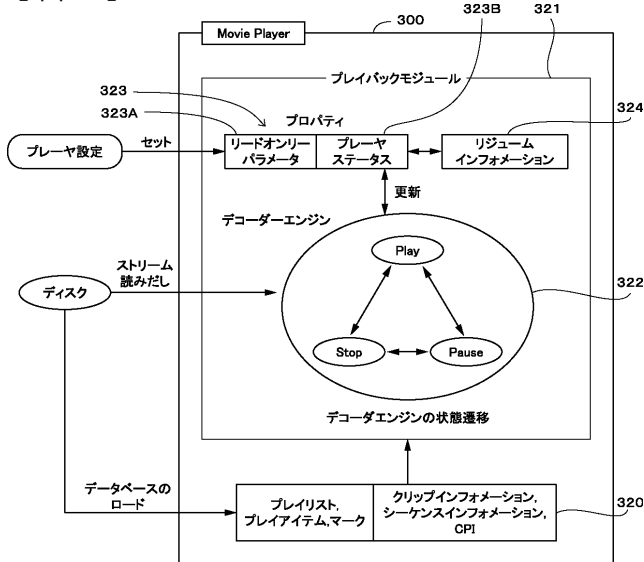
【 図 1 】



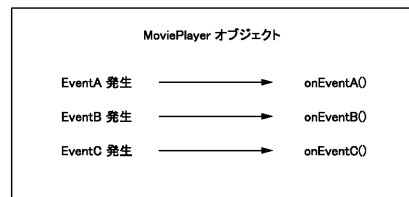
【 図 2 】



【 図 3 】



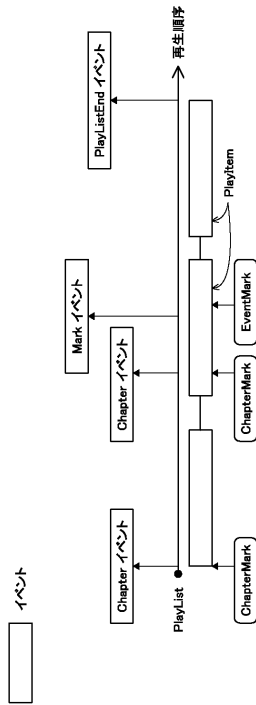
【 図 5 】



【 図 4 】

状態	説明
play	PlayList の再生を行っており、時間が経過している状態を指す。通常再生のほか、変速再生、早送り、巻き戻しも含む。コマ送り、コマ戻しは pause と play を繰り返している状態である。
pause	PlayList の再生を行っている状態で、時間軸が停止している状態。
stop	PlayList を再生していない状態。

【 図 6 】



【 図 7 】

Name	Description
scriptVersion	UMD Video Scriptのバージョン
languageCode	UMD Video Playerに設定されたオーディオ言語コード
audioLanguageCode	UMD Video Playerに設定されたオーディオ言語コード
subttlLanguageCode	UMD Video Playerに設定された字幕言語コード
playlistNumber	現在再生中のプレイリスト番号
chapterNumber	現在再生中のチャプター番号
videoNumber	現在再生中のビデオストリーム番号
audioNumber	現在再生中のオーディオストリーム番号
subttlNumber	現在再生中の字幕ストリーム番号
playlistTime	プレイリスト先頭を0とした時の時刻
audioFlag	オーディオ再生ON/OFF及Dual mono LRの指定
subttlFlag	字幕表示ON/OFF

リットオンリーパラメータ

プレーヤステータス

【 図 8 】

Name	Description
play()	再生
playChapter()	チャプター指定再生
stop()	再生の停止
pause()	再生の一時停止
playStep()	コマ送り
changeStream()	ビデオ、オーディオ、字幕ストリーム変更
getPlayerStatus()	MoviePlayerの再生、停止、一時停止などの状態を取得
reset()	再生を停止し、resumeInformationをクリア
setPos()	ビデオの表示位置の設定
getPos()	ビデオの表示位置の取得
setSize()	ビデオの表示サイズの設定
getSize()	ビデオの表示サイズの取得

【 図 9 】

キー名称	説明
VK_POWER	電源キー
VK_POWER_ON	電源 ONキー
VK_POWER_OFF	電源OFFキー
VK_MENU	メニュー
VK_ENTER	決定
VK_RETURN	戻る
VK_PLAY	再生
VK_STOP	停止
VK_PAUSE	一時停止
VK_FAST_FORWARD	早送り
VK_FAST_REVERSE	早戻し
VK_SLOW_FORWARD	スロー(順方向)
VK_SLOW_REVERSE	スロー(逆方向)
VK_STEP_FORWARD	コマ送り(順方向)
VK_STEP_REVERSE	コマ送り(逆方向)

【 図 10 】

キー名称	説明
VK_NEXT	次
VK_PREVIOUS	前
VK_UP	上
VK_DOWN	下
VK_RIGHT	右
VK_LEFT	左
VK_UP_RIGHT	右上
VK_UP_LEFT	左上
VK_DOWN_RIGHT	右下
VK_DOWN_LEFT	左下
VK_ANGLE	アングル切り替え
VK_SUBTITLE	字幕切り替え
VK_AUDIO	オーディオ切り替え
VK_VIDEO_ASPECT	ビデオのアスペクト比切り替え
VK_COLORED_KEY_1	色つきファンクションキー1
VK_COLORED_KEY_2	色つきファンクションキー2
VK_COLORED_KEY_3	色つきファンクションキー3
VK_COLORED_KEY_4	色つきファンクションキー4
VK_COLORED_KEY_5	色つきファンクションキー5
VK_COLORED_KEY_6	色つきファンクションキー6

【 図 1 1 】

ユーザ操作に起因する制御命令	説明
uo_timeSearch(playList Time)	再生中のplayListの指定時刻から再生する。playListTimeはPlayList先頭を0としたときの時刻を表す。PlayList番号は指定できない。そのため、現在再生中のPlayListの範囲内での時刻指定になる。
uo_play()	1xで再生開始する。開始位置はresumeInformationによって決められる。resumeInformationが無い場合は、このユーザ操作は無効になる。playListNumberの指定のないplay()メソッドを実行したときに対応。ユーザ操作では、PlayList番号を指定できない。
uo_playChapter(chapter Number)	再生中のplayListの指定のChapterから再生開始する。Chapterの指定がない場合には、現在再生中のChapterの先頭から再生開始する。chapterNumberの指定のないplayChapter()メソッドに対応。
uo_playPrevChapter()	前Chapterの先頭から再生開始する。
uo_playNextChapter()	次Chapterの先頭から再生開始する。
uo_stop()	再生を停止する。
uo_jumpToEnd()	PlayListの最後にjumpする。MoviePlayerに対して、現在の再生を中止して、playListEndイベントを発生させるよう指示するユーザ操作。スクリプトでは、onPlayListEndイベントハンドラが実行される。
uo_forwardScan(speed)	speedで指定された速度で順方向再生。speedはUMD Video Player実装依存。
uo_backwardScan(speed)	speedで指定された速度で逆方向再生。speedはUMD Video Player実装依存。
uo_playStep(forward)	順方向コマ送り再生
uo_playStep(backward)	逆方向コマ送り再生
uo_pauseOn()	ユーザによる一時停止
uo_pauseOff()	一時停止を解除
uo_changeAudioChannel(value)	オーディオのチャンネル数切り替えおよびdual monoの時の片チャンネル切り換え。audioFlagを変更する。
uo_setAudioEnabled(boolean)	オーディオストリームのON/OFFを指定。audioFlagを変更する。
uo_setSubtitleEnabled(boolean)	字幕ストリームのON/OFFを指定。subtitleFlagを変更する。
uo_angleChange()	表示アングルを変更する。このユーザ操作がMoviePlayerに伝えられると、MoviePlayerはスクリプトにangleChangeイベントを通知する。
uo_audioChange(audioStreamNumber)	再生するオーディオを変更する。
uo_subtitleChange(subtitleStreamNumber)	再生する字幕を変更する。

【 図 1 2 】

イベント	説明	MoviePlayerのメソッドとの関係
menu	メニューにジャンプする。	MoviePlayerではなく、スクリプトに通知されるイベント。onMenuイベントハンドラが実行される。
exit	native platformがUMD Videoアプリケーションを終了させる際にnative platformから発生されるイベント。	onExitイベントハンドラが実行される。
up,down,left,right focusIn, focusOut, push, cancel	画面に表示されているボタンにフォーカスが当たっている間に発生するイベント	MoviePlayerではなく、スクリプトに通知されるイベント。
autoPlay, continuePlay	スクリプトの実行開始を指示するイベント	

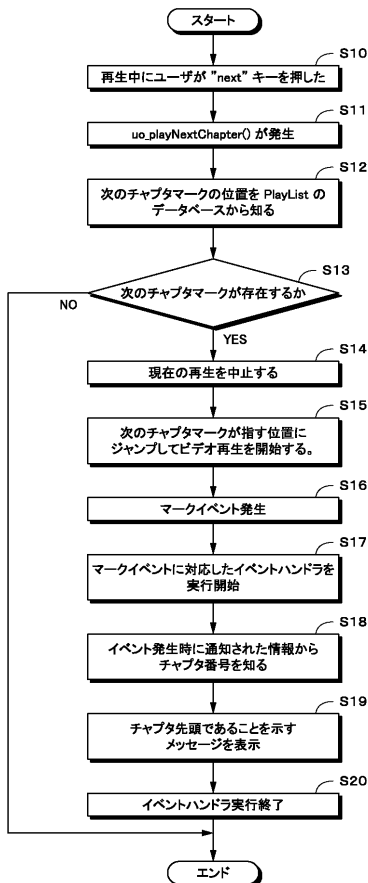
【 図 1 3 】

イベント名	対応するイベントハンドラ名	Description
mark	onMark()	Event-mark 検出時に実行される
playListEnd	onPlayListEnd()	プレイリストが終了した時に実行される
chapter	onChapter()	Chapter-mark 検出時に実行される
angleChange	onAngleChange()	ユーザ操作のアングル変更が指示された時に実行される
audioChange	onAudioChange()	ユーザ操作のオーディオ変更が指示された時に実行される
subtitleChange	onSubtitleChange()	ユーザ操作の字幕変更が指示された時に実行される

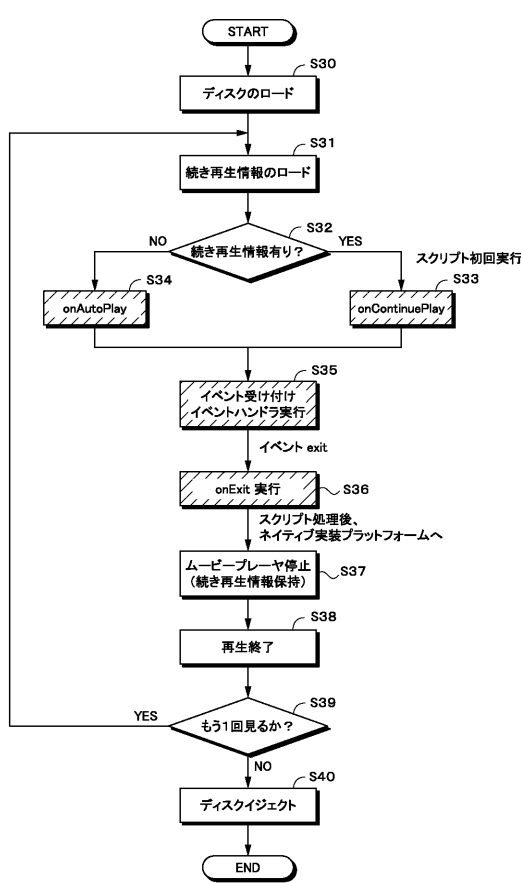
【 図 1 4 】

イベント名	対応するイベントハンドラ名	内容
menu	onMenu()	メニューにジャンプする。
exit	onExit()	native platform が UMD Video アプリケーションを終了させる際に native platform から発生されるイベント。
autoPlay, continuePlay	onAutoPlay(), onContinuePlay()	スクリプトの実行を開始する。

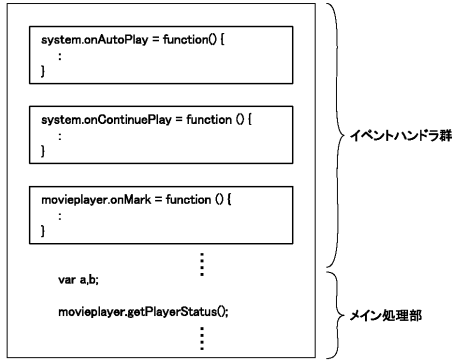
【 図 1 5 】



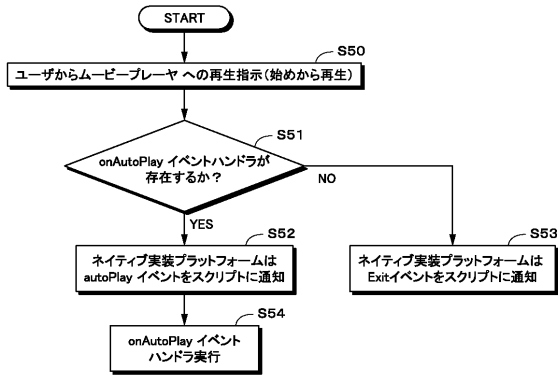
【 図 1 6 】



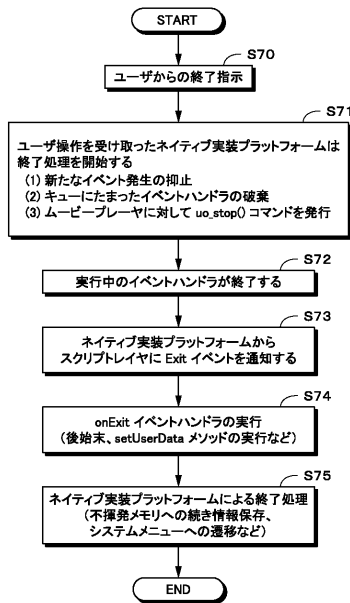
【 図 17 】



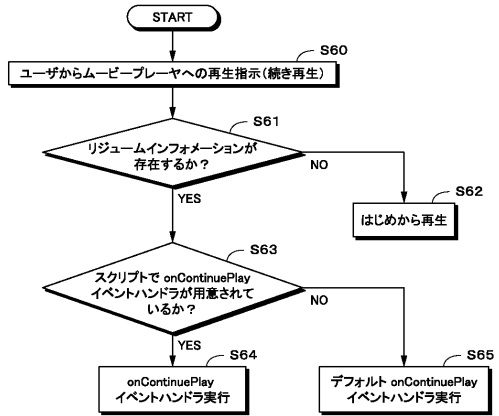
【 図 18 】



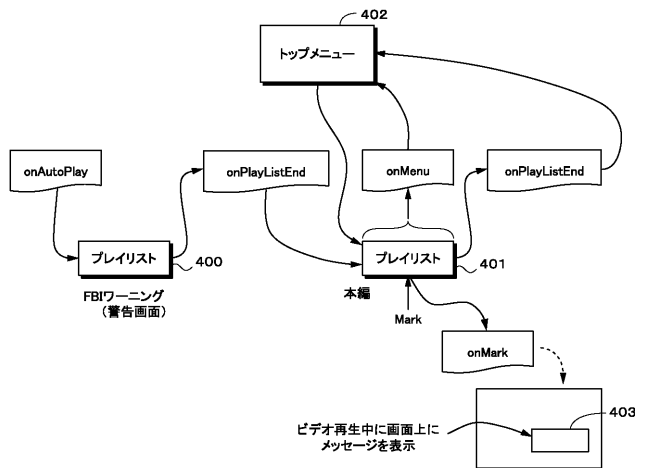
【 図 20 】



【 図 19 】



【 図 21 】



【 図 2 2 】

```

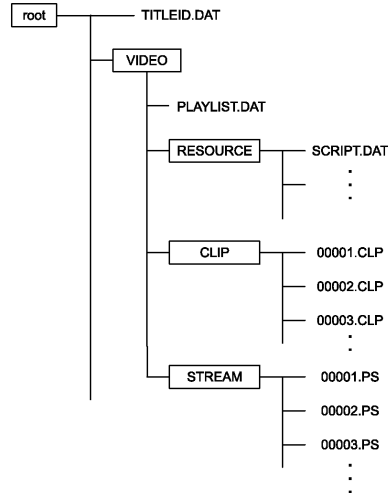
system.onAutoPlay = function(){
    //Play Playlist # 1 FBI warning.
    movieplayer.play(1);
}

movieplayersystem.onPlaylistEnd = function(event_info){
    if(event_info.playlistNumber == 1){
        // play feature film after FBI warning ends.
        movieplayer.play(2);
    }else{
        // transit to top menu after feature film ends.
        resource.pagetable["top_menu"].open();
    }
}

system.onMenu = function(){
    // transfer to top menu with display menu user
    operation.
    resource.pagetable["top_menu"].open();
}

movieplayer.onMark = function(event_info){
    //display dialog when event mark encountered.
    if(event_info.mark_data == 1){
        resource.pagetable["dialog_window_1"].open();
    }
    ...
}
    
```

【 図 2 3 】



【 図 2 4 】

Syntax	No. of bits	Mnemonic
"PLAYLIST.DAT" {		
name_length	8	uimbsf
name_string	8*255	bslbf
number_of_PlayLists	16	uimbsf
for(i=0; i<number_of_PlayLists; i++){		
Playlist(){ // A Playlist()		
Playlist_data_length	32	uimbsf
// 属性情報		
reserved_for_word_alignment	15	bslbf
capture_enable_flag_PlayList	1	bslbf
Playlist_name_length	8	uimbsf
Playlist_name_string	8*255	bslbf
//		
number_of_PlayItems	16	uimbsf
for (i=0; i<number_of_PlayItems; i++) {		
PlayItem()		
}		
PlayListMark()		
}		
}		
}		

【 図 2 5 】

Syntax	No. of bits	Mnemonic
PlayItem() {		
length	16	uimbsf
Clip_Information_file_name_length	16	uimbsf
Clip_Information_file_name	8*Clip_Infor mation_file_ name_length	bslbf
IN_time	32	uimbsf
OUT_time	32	uimbsf
}		

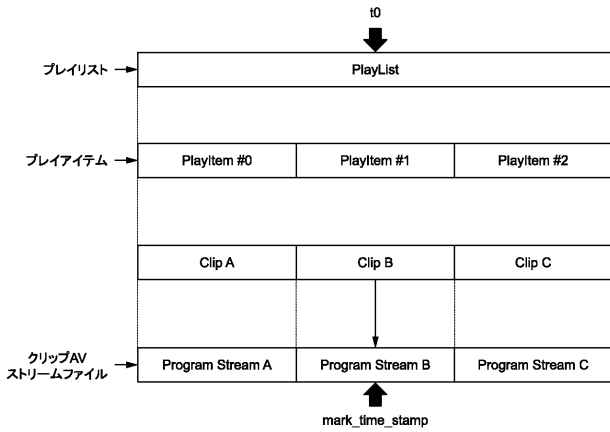
【 図 2 6 】

Syntax	No. of bits	Mnemonic
PlayListMark() {		
length	32	uimbsf
number_of_PlayList_marks	16	uimbsf
for(i=0; i<number_of_PlayList_marks; i++){		
Mark(){		
mark_type	8	uimbsf
mark_name_length	8	uimbsf
ref_to_PlayItem_id	16	uimbsf
mark_time_stamp	32	uimbsf
entry_ES_stream_id	8	uimbsf
entry_ES_private_stream_id	8	uimbsf
mark_data	32	bslbf
mark_name_string	8*24	bslbf
}		
}		
}		

【 図 2 7 】

mark_type	stream coding
0	reserved
1	Chapterマーク
2	Indexマーク
3	Eventマーク
4-255	reserved

【 図 2 8 】



【 図 2 9 】

Syntax	No. of bits	Mnemonic
XXXXX.CLP{		
presentation_start_time	32	uimsbf
presentation_end_time	32	uimsbf
reserved_for_word_alignment	7	bslbf
capture_enable_flag_Clip	1	bslbf
number_of_streams	8	uimsbf
for (i = 0; i < number_of_streams; i++) {		
StreamInfo{		
length	16	uimsbf
stream_id	8	uimsbf
private_stream_id	8	uimsbf
StaticInfo{		
reserved_for_word_alignment	8	bslbf
number_of_DynamicInfo	8	uimsbf
for (j = 0; j < number_of_DynamicInfo; j++) {		
pts_change_point	32	uimsbf
DynamicInfo{		
}		
}		
}		
}		
EP_map{		
}		

【 図 3 0 】

エレメンタリストリームの種類	stream_id	private_stream_id
ビデオ	0xE0~0xEF	(なし)
ATRACオーディオ	0xBD	0x00~0x0F
LPCMオーディオ	0xBD	0x10~0x1F
字幕	0xBD	0x80~0x9F

【 図 3 1 】

Syntax	No. of bits	Mnemonic
StaticInfo{		
if (stream == VIDEO){		
reserved_for_word_alignment	16	bslbf
picture_size	4	uimsbf
frame_rate	4	uimsbf
reserved_for_word_alignment	7	bslbf
cc_flag	1	bslbf
} else if (stream == AUDIO){		
audio_language_code	16	bslbf
channel_configuration	8	uimsbf
reserved_for_word_alignment	3	bslbf
life_existence	1	bslbf
sampling_frequency	4	uimsbf
} else if (stream == SUBTITLE){		
subtitle_language_code	16	bslbf
subtitle_presentation_type	4	uimsbf
reserved_for_word_alignment	11	bslbf
configurable_flag	1	uimsbf
}		
}		

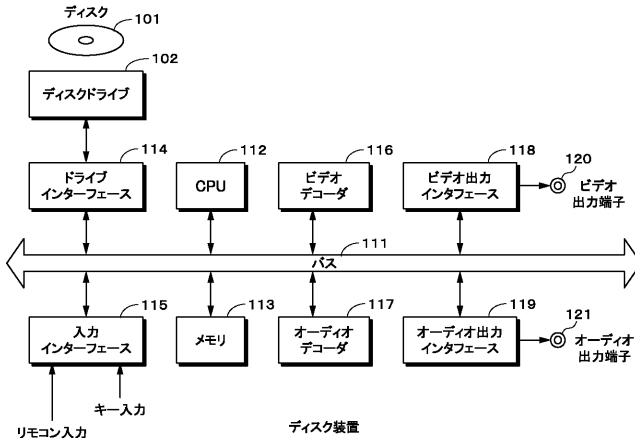
【 図 3 2 】

Syntax	No. of bits	Mnemonic
DynamicInfo(i,j){		
reserved_for_word_alignment	8	bslbf
if (stream == VIDEO){		
reserved_for_word_alignment	4	bslbf
display_aspect_ratio	4	uimsbf
} else if (stream == AUDIO){		
reserved_for_word_alignment	4	bslbf
channel_assignment	4	uimsbf
} else if (stream == SUBTITLE){		
reserved_for_word_alignment	8	bslbf
}		
}		

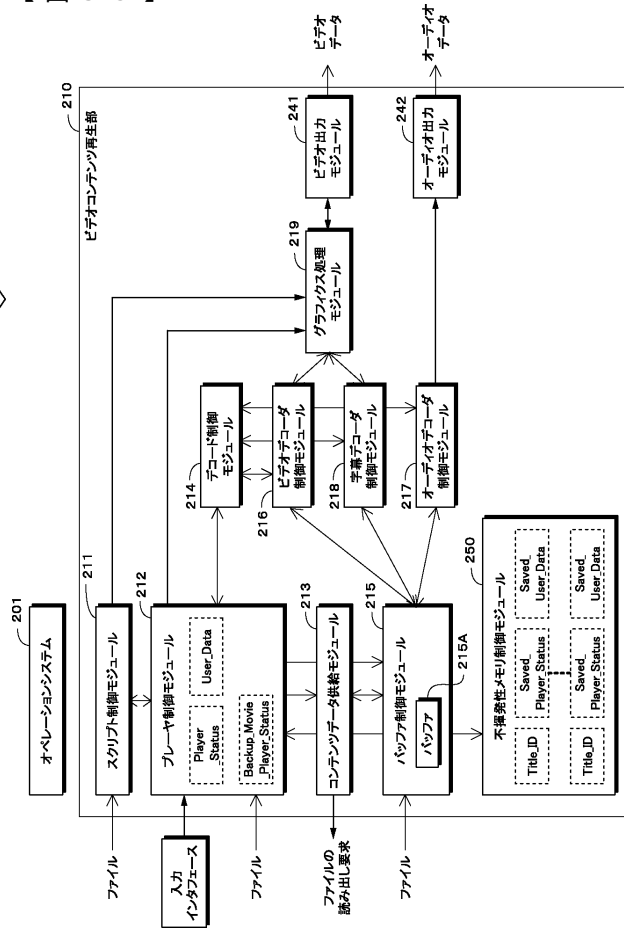
【 図 3 3 】

Syntax	No. of bits	Mnemonic
EP_map{		
reserved_for_word_alignment	8	bslbf
number_of_stream_id_entries	8	uimsbf
for (k=0; k<number_of_stream_id_entries; k++){		
stream_id	8	bslbf
private_stream_id	8	bslbf
number_of_EP_entries	32	uimsbf
for (i=0; i<number_of_EP_entries; i++){		
PTS_EP_start	32	uimsbf
RPN_EP_start	32	uimsbf
}		
}		
}		

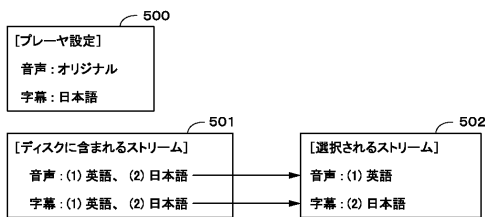
【 図 3 4 】



【 図 3 5 】



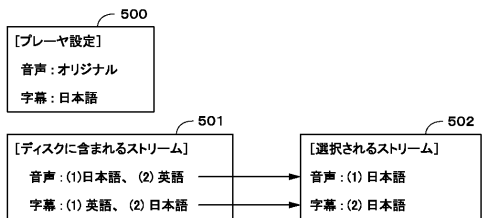
【 図 3 6 】



【 図 3 9 】

audioFlag	説明
0	音声再生 OFF
1	ON デュアルモノの場合は両チャンネル再生
2	デュアルモノの左のみ
3	デュアルモノの右のみ

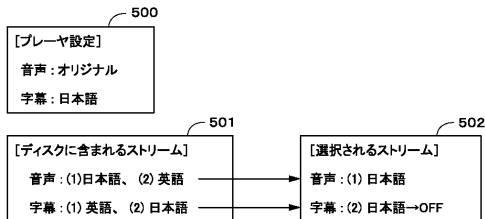
【 図 3 7 】



【 図 4 0 】

subtitleFlag	説明
0	字幕表示 OFF
1	ON

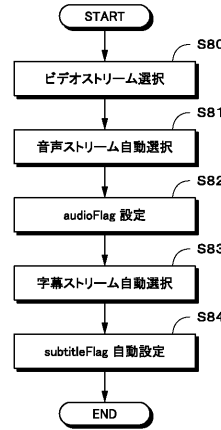
【 図 3 8 】



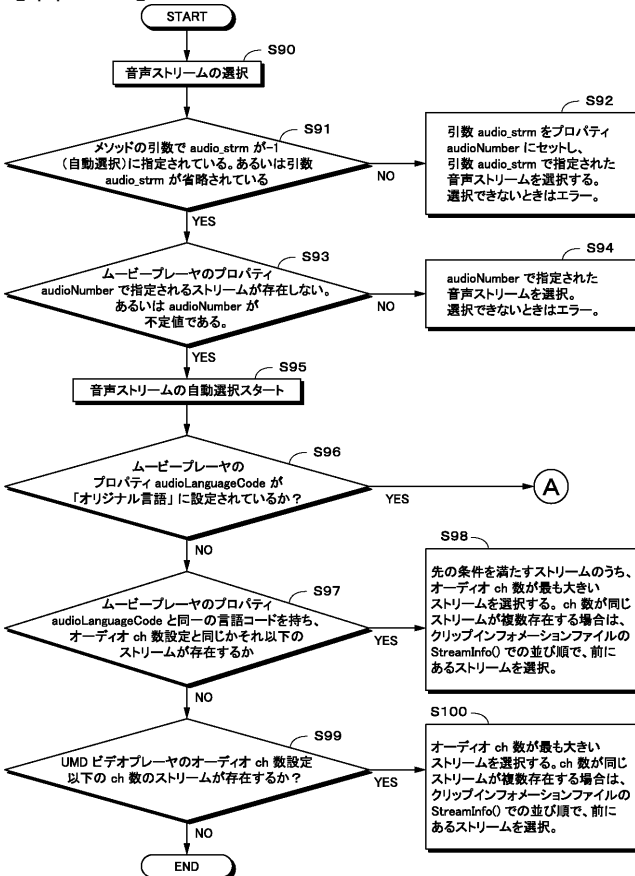
【 図 4 1 】

Value	型	Note
playListNumber	int	プレイリスト番号を表す。 -1: 現在再生中のプレイリスト番号
playListTime	int	プレイリストの先頭からの時刻を指定。 プレイリスト先頭からの再生は0を指定。
menuMode	Boolean	true の場合、メニューモードでの再生を指定。 false の場合、ノーマルモードでの再生を指定。
pauseMode	Boolean	true の場合、一時停止で待機する。 false の場合、通常速度での再生を開始。
video_strm	int	再生するビデオストリームを指定する。 -1: ムービープレーヤによる自動選択。 -2: 変更無し
audio_strm	int	再生する音声ストリームの番号を指定する。 -1: ムービープレーヤによる自動選択。 -2: 変更無し
audio_status	int	音声再生の状態を表す。 ここで指定した値がプロパティ audioFlag に反映される。 0: 音声再生 OFF。 1: ON(デュアルモノの場合は両チャンネル再生)。 2: デュアルモノの左のみを再生 3: デュアルモノの右のみを再生 -2: 変更無し
subtitle_strm	int	再生する字幕ストリームの番号を指定する。 -1: ムービープレーヤによる自動選択を指定。 -2: 変更無し。
subtitle_status	int	字幕再生の状態を表す。 ここで指定した値がプロパティ subtitleFlag に反映される。 0: 字幕表示 OFF。 1: 字幕表示 ON。 -1: 自動選択。 -2: 変更無し。

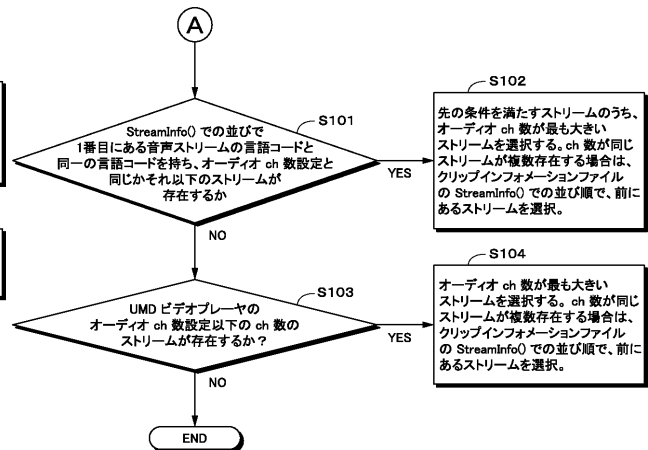
【 図 4 2 】



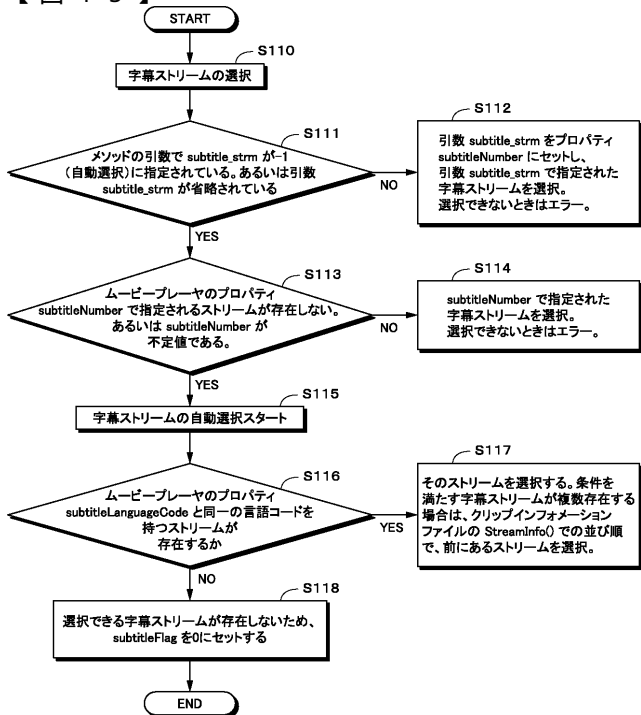
【 図 4 3 】



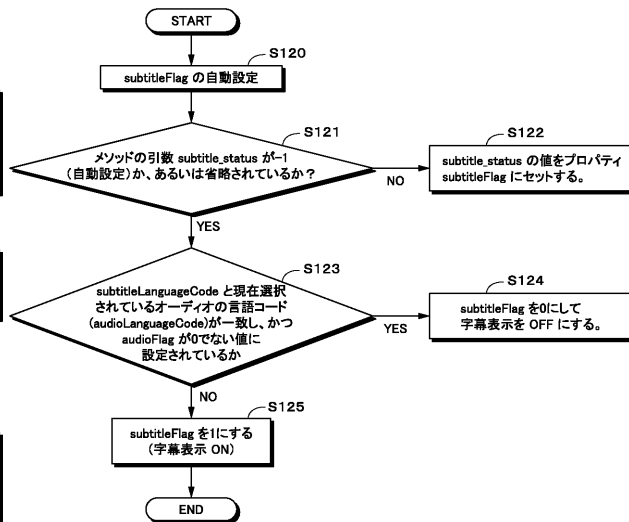
【 図 4 4 】



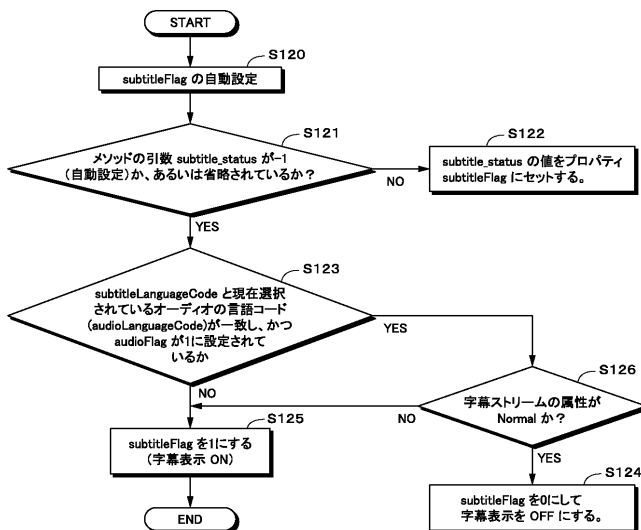
【 図 4 5 】



【 図 4 6 】



【 図 4 7 】



フロントページの続き

(72)発明者 浜田 俊也

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72)発明者 各務 辰哉

東京都港区南青山2丁目6番地21号 株式会社ソニー・コンピュータエンタテインメント内

(72)発明者 藤波 靖

東京都品川区北品川6丁目7番35号 ソニー株式会社内

Fターム(参考) 5C053 FA23 GB11 JA01 KA24

5D044 AB05 AB07 AB09 BC02 CC06 DE18 FG18 GK08 GK12

5D077 AA23 CA02 DC21 DF01 EA34

5D110 AA14 AA27 AA29 DA04 DC05 DE01