(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2004-187272
(P2004-187272A)
(43)公開日　平成16年7月2日(2004.7.2)

(51)Int.Cl.[7]	F I		テーマコード（参考）
H04N 5/91	H04N 5/91	N	5B069
G06F 3/14	G06F 3/14	340B	5B082
G06F 12/00	G06F 12/00	520E	5C053

審査請求 未請求　請求項の数 20　ＯＬ　外国語出願　（全 85 頁）

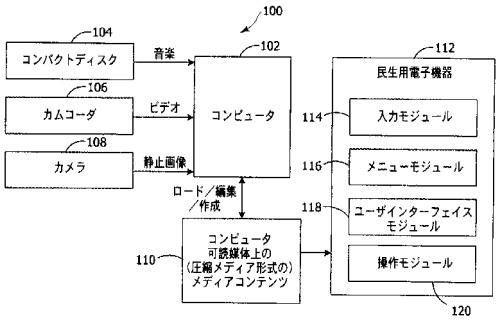
(21)出願番号　特願2003-336557(P2003-336557)
(22)出願日　平成15年9月26日(2003.9.26)
(31)優先権主張番号　60/418,973
(32)優先日　平成14年10月16日(2002.10.16)
(33)優先権主張国　米国(US)
(31)優先権主張番号　10/273,411
(32)優先日　平成14年10月17日(2002.10.17)
(33)優先権主張国　米国(US)

(71)出願人　391055933
マイクロソフト　コーポレイション
ＭＩＣＲＯＳＯＦＴ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ　（番地なし）
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　イアン　キャメロン　マーサー
アメリカ合衆国　９８０７５　ワシントン州　サマミッシュ　２０８　アベニュー　サウスイースト　２６０３

最終頁に続く

(54)【発明の名称】メディアプレーヤーの適応型メニューシステム

(57)【要約】
【課題】　メディアファイルの適応型メニュー構造を作成し、表示すること。
【解決手段】　本発明は、ユーザがメディアプレーヤーを使ってメディアファイルを操作できるようにメニュー構造を作成するオーサリングソフトウェアを含む。本発明はまた、メディアプレーヤーに関連したメディアタイプに応じて、メディアプレーヤーに関連したディスプレイ上にメニュー情報を表示するように作成したメニュー構造を適合させるフィルタリングソフトウェアも含む。一実施形態において、本発明は、機能の豊富なパーソナルコンピュータのメディアプレーヤーならびに低コストのメディアプレーヤー上での再生のために設計されたコンパイル済みのバイナリ形式でメタデータ、メニュー、および再生リストが入っている一組の小さなファイルを有する圧縮メディア形式の一部として動作可能である。
【選択図】　図１



【特許請求の範囲】
【請求項１】
ユーザがメディアプレーヤーを使ってメディアファイルを操作できるようにするために１つまたは複数の前記メディアファイルを編成する方法であって、前記メディアファイルそれぞれは関連するメディアタイプを有しており、かつ前記メディアタイプに応じて前記メディアプレーヤーによるレンダリングのために適合されており、前記メディアプレーヤーもまた関連するメディアタイプを有しており、前記方法は、
メニューヘッダを作成し、
前記作成したメニューヘッダに１つまたは複数の選択されたメディアファイルに対する参照を設定し、
前記選択されたメディアファイルのそれぞれに関連するメディアタイプを識別し、
前記メニューヘッダに前記識別したメディアタイプを設定し、前記設定したメニューヘッダはメディアプレーヤーに関連するメディアタイプに応じて前記選択されたメディアファイルをフィルタするために前記メディアプレーヤーにより使用されること
を備えることを特徴とする方法。
【請求項２】
前記メニューヘッダと他のメニューヘッダとの間の操作を可能にするために前記メニューヘッダに前記他のメニューヘッダに対する参照を設定することをさらに備えることを特徴とする請求項１に記載の方法。
【請求項３】
前記メディアプレーヤーに関連する前記メディアタイプに応じて前記他のメディアヘッダに対する前記参照をフィルタすることによって前記メニューヘッダを適合させることをさらに備えることを特徴とする請求項２に記載の方法。
【請求項４】
前記メディアプレーヤーと互換性があるコンピュータ可読媒体上に前記選択されたメディアファイルと前記設定したメニューヘッダとを格納することをさらに備えることを特徴とする請求項１に記載の方法。
【請求項５】
前記選択されたメディアファイルに関連する前記メディアタイプおよび前記メディアプレーヤーに関連する前記メディアタイプは、オーディオ、ビデオ、および画像の１つまたは複数を備えることを特徴とする請求項１に記載の方法。
【請求項６】
前記メニューヘッダに設定する前記参照は、前記選択されたメディアファイルに対するメモリ位置のオフセット、前記選択されたメディアファイルを参照する再生リストに対するポインタ、または前記選択されたメディアファイルに対するポインタの１つを備えることを特徴とする請求項１に記載の方法。
【請求項７】
前記設定したメニューヘッダを読み取り、
前記メディアプレーヤーに関連する前記メディアタイプに応じて前記選択されたメディアファイルに対する前記参照をフィルタすることによって前記読み取ったメニューヘッダを適合させ、
前記メディアプレーヤーに関連するディスプレイ上でユーザに対して前記適合させたメニューヘッダからのメニュー情報を表示すること
をさらに備えることを特徴とする請求項１に記載の方法。
【請求項８】
１つまたは複数のコンピュータ可読媒体は、請求項１に記載の前記方法を実行するためのコンピュータ実行可能命令を有することを特徴とする請求項１に記載の方法。
【請求項９】
メディアオーサリングツールにおいて、ユーザがメディアプレーヤーを用いてメディアファイルを操作できるようにするために、１つまたは複数のコンピュータ可読媒体が１つ

または複数の前記メディアファイルを編成するためのコンピュータ実行可能コンポーネントを有しており、前記メディアファイルそれぞれは関連するメディアタイプを有しており、前記メディアタイプに応じて前記メディアプレーヤーによるレンダリングのために適合されており、前記メディアプレーヤーもまた関連するメディアタイプを有しており、前記コンポーネントは、

メニューヘッダを作成するための初期化モジュールと、

前記初期化モジュールからの前記作成したメニューヘッダに１つまたは複数のメディアファイルに対する参照を設定するための再生リストモジュールと、

前記再生リストモジュールからの前記選択されたメディアファイルのそれぞれに関連するメディアタイプを識別するための要約モジュールと、

前記メニューヘッダに前記要約モジュールからの前記識別したメディアタイプを設定するための伝搬モジュールであって、前記設定したメニューヘッダはメディアプレーヤーに関連するメディアタイプに応じて前記選択されたメディアファイルをフィルタするために前記メディアプレーヤーによって使用される前記伝搬モジュールと

を備えることを特徴とするコンピュータ可読媒体。

【請求項１０】

前記メニューヘッダと他のメニューヘッダとの間の操作を可能にするために前記メニューヘッダに前記他のメニューヘッダに対する参照を設定するためのメニューモジュールをさらに備えることを特徴とする請求項９に記載のコンピュータ可読媒体。

【請求項１１】

前記メディアプレーヤーと互換性があるコンピュータ可読媒体上に前記選択されたメディアファイルと前記設定したメニューヘッダとを格納するための書き込みモジュールをさらに備えることを特徴とする請求項９に記載のコンピュータ可読媒体。

【請求項１２】

前記選択されたメディアファイルに関連する前記メディアタイプと前記メディアプレーヤーに関連する前記メディアタイプは、オーディオ、ビデオ、および画像の１つまたは複数を備えることを特徴とする請求項９に記載のコンピュータ可読媒体。

【請求項１３】

前記メニューヘッダに設定する前記参照は、前記選択されたメディアファイルに対するメモリ位置オフセット、選択されたメディアファイルを参照する再生リストの対するポインタ、または前記選択されたメディアファイルに対するポインタの１つを備えることを特徴とする請求項９に記載のコンピュータ可読媒体。

【請求項１４】

前記設定したメニューヘッダを読み取るための入力モジュールと、

前記メディアプレーヤーに関連する前記メディアタイプに応じて前記選択されたメディアファイルに対する前記参照をフィルタすることによって前記入力モジュールからの前記読み取ったメニューヘッダを適合させるためのメニューモジュールと、

前記メニューモジュールからの前記適合させたメニューヘッダからメニュー情報を前記メディアプレーヤーに関連するディスプレイ上でユーザに表示するためのユーザインターフェイスモジュールと

をさらに備えることを特徴とする請求項９に記載のコンピュータ可読媒体。

【請求項１５】

１つまたは複数のメディアファイルを編成する１つまたは複数のコンピュータ可読媒体であって、前記メディアファイルそれぞれは関連するメディアタイプを有しており、かつ前記メディアタイプに応じてメディアプレーヤーによるレンダリングのために適合されており、前記メディアプレーヤーは前記メディアプレーヤーに関連するディスプレイ上でメニュー情報を表示し、前記メニュー情報は、ユーザが前記メディアプレーヤーを用いて前記１つまたは複数のメディアファイルを操作することができるようにし、前記メディアプレーヤーもまた関連するメディアタイプを有しており、前記コンピュータ可読媒体は、

１つまたは複数の選択されたメディアファイルに対する参照と前記選択されたメディア

ファイルのそれぞれ各１つに関連するメディアタイプを有するメニューヘッダを読み取るための入力モジュールと、
メディアプレーヤーに関連するメディアタイプに応じて前記選択されたメディアファイルに対する前記参照をフィルタすることによって前記入力モジュールからの前記読み取ったメニューヘッダを適合させるためのメニューモジュールと、
前記メニューモジュールからの適合させたメニューヘッダからメニュー情報を前記メディアプレーヤーに関連するディスプレイ上でユーザに表示するためのユーザインターフェイスモジュールと
を備えることを特徴とするコンピュータ可読媒体。
【請求項１６】
前記メニューヘッダは、該メニューヘッダと他のメニューヘッダとの間の操作を可能にするために前記他のメニューヘッダに対する参照と該他のメニューヘッダに関連するメディアタイプとを含み、前記メディアプレーヤーに関連する前記メディアタイプに応じて前記他のメニューヘッダに対する前記参照をフィルタすることによって前記読み取ったメニューヘッダを適合させるための操作モジュールをさらに備えることを特徴とする請求項１５に記載のコンピュータ可読媒体。
【請求項１７】
前記選択されたメディアファイルに対する前記参照は、前記選択されたメディアファイルに対するメモリ位置のオフセット、前記選択されたメディアファイルを参照する再生リストに対するポインタ、または前記選択されたメディアファイルに対するポインタの１つを備えることを特徴とする請求項１５に記載のコンピュータ可読媒体。
【請求項１８】
前記選択されたメディアファイルに対する前記参照を格納する再生リストフィールドと、
前記再生リストフィールドにおいて参照した前記選択されたメディアファイルの前記メディアタイプを格納する要約フィールドと
をさらに備えることを特徴とする請求項１５に記載のコンピュータ可読媒体。
【請求項１９】
コンテンツに関連したデータに対応する値を格納するメタデータフィールドをさらに備えており、前記コンテンツに関連したデータは、前記メディアプレーヤーに応じて、前記１つまたは複数の選択されたメディアファイルのコンテンツのタイトル、作曲家、演者、ジャンル、スタジオ、ディレクタ、格付け、アーティスト、および説明の内の１つまたは複数から選択されていることを特徴とする請求項１５に記載のコンピュータ可読媒体。
【請求項２０】
前記選択されたメディアファイルに関連するメタデータを識別する値を格納する地域フィールドであって、前記メタデータは特定の言語に対応する地域フィールドをさらに備えることを特徴とする請求項１５に記載のコンピュータ可読媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、デジタルメディアコンテンツの分野に関する。詳細には、本発明はメディアプレーヤーと共に使用するために適応型のメニューシステムを作成し、読み出すことに関する。
【背景技術】
【０００２】
技術分野における近年の進展により、コンピュータのユーザは、現在、パーソナルコンピュータまたはラップトップコンピュータ上に様々なメディアおよびマルチメディアコンテンツを再生するなど、より一層のユーザ体験を提供する多くの機能を楽しむことができる。例えば、今日の大部分のコンピュータは、コンパクトディスク（ＣＤ）を再生することができるので、ユーザはコンピュータで作業をしながら、好みの音楽アーティストを聴くことができる。多くのコンピュータはまた、デジタル多用途ディスク（ＤＶＤ）ドライ

ブを装備し、ユーザが映画を観ることができるようにしている。
【発明の開示】
【発明が解決しようとする課題】
【０００３】
ポータブルＣＤプレーヤーや、ＤＶＤプレーヤーおよびカーレシーバ（Ｃａｒ　Ｒｅｃｅｉｖｅｒ）などの民生用電子機器は、メディア再生に関して非常に様々な機能およびユーザインターフェイス機能を有している。これらの装置には、画像（イメージ）およびビデオデータを表示できるものもあれば、オーディオデータしか再生できないものもある。ＤＶＤプレーヤーなど一部の装置は、グラフィカルなディスプレイ装置を有し、背景画像およびサムネイルと共に複雑なメニューを表示することができるが、ポータブルＣＤプレーヤーなどの他の装置は、グラフィック機能のない単一行の液晶ディスプレイ（ＬＣＤ）と共に「再生」、「次」、「前」、「停止」のような簡単なボタンしか有していない。そのような次第で、ユーザがコンピュータ可読媒体上にＭＰ３（Ｍｏｖｉｎｇ　Ｐｉｃｔｕｒｅ　Ｅｘｐｅｒｔｓ　Ｇｒｏｕｐ　ａｕｄｉｏ　ｌａｙｅｒ－３）ファイルや、ＷＭＡ（ＷＩＮＤＯＷＳ　ＭＥＤＩＡ（登録商標）　ｔｅｃｈｎｏｌｏｇｉｅｓ　ａｕｄｉｏ）ファイルや、またはＪＰＥＧ（Ｊｏｉｎｔ　Ｐｈｏｔｏｇｒａｐｈｉｃ　Ｅｘｐｅｒｔｓ　Ｇｒｏｕｐ）ファイルなどの圧縮されたメディアファイルを格納する場合、ユーザはこれら格納した圧縮メディアファイルをレンダリングするために使用されることになる再生装置のタイプすべてについて知らないことが多い。例えば、ユーザは、現在のメディアプレーヤーと異なる機能を備える新しい再生装置をその後に購入し、以前作成したコンピュータ可読媒体を再生したい場合がある。
【０００４】
こうした理由で、１つまたは複数のこれらおよびその他の不都合に対処するために、適応型のメニュー構造のためのシステムが望まれる。
【課題を解決するための手段】
【０００５】
本発明は、１つまたは複数のメディアファイルと関連付けられた適応型メニュー構造を作成するためのソフトウェアおよびデータ構造を含む。詳細には、本発明は、メニュー項目がメディアプレーヤーの機能に応じて選択的に隠されることによる適応型のメニューシステムを含む。本発明のメニューシステムは、テレビの画面上でサムネイル画像を備える鮮やかなグラフィック表示にすることもできるし、単一行のＬＣＤディスプレイ上で一度に１つのオプションのような簡単なメニューとして表すこともできる。一実施形態において、本発明は、機能豊富なパーソナルコンピュータのメディアプレーヤーならびに低コストのメディアプレーヤー上での再生のために設計されたコンパイル済みのバイナリ形式のメタデータ、メニュー、および再生リストが入っている一組の小さなファイルを有する圧縮メディア形式の一部として動作可能である。この形式は、様々な形式でのオーディオ、静止画像、およびビデオを包含する。
【０００６】
本発明のひとつの態様によれば、この方法は、１つまたは複数のファイルを編成し、ユーザがメディアプレーヤーを使ってメディアファイルを操作できるようにする。メディアファイルはそれぞれ、そのメディアファイルと関連したメディアタイプを有し、メディアタイプに応じてメディアプレーヤーによるレンダリングのために適合させられる。メディアプレーヤーもまた、そのメディアプレーヤーと関連したメディアタイプを有している。この方法には、メニューヘッダを作成することが含まれる。この方法にはまた、作成したメニューヘッダに１つまたは複数の選択されたメディアファイルに対する参照を設定することも含まれる。この方法にはまた、選択されたファイルのそれぞれに関連したメディアタイプを識別することと、識別したメディアタイプをメニューヘッダに設定することも含まれる。メディアプレーヤーは、設定したメニューヘッダを使い、メディアプレーヤーに関連したメディアタイプに応じて選択されたメディアファイルをフィルタする。
【０００７】

本発明の他の態様によれば、メディアオーサリングツールにおける１つまたは複数のコンピュータ可読媒体は、１つまたは複数のメディアファイルを編成するためのコンピュータ実行可能コンポーネントを有し、ユーザがメディアプレーヤーを使ってメディアファイルを操作できるようにする。メディアファイルはそれぞれ、そのメディアファイルと関連したメディアタイプを有し、このメディアタイプに応じてメディアプレーヤーによるレンダリングのために適合させられる。メディアプレーヤーもまた、そのメディアプレーヤーと関連したメディアタイプを有している。これらのコンポーネントには、メニューヘッダを作成するための初期化モジュールが含まれる。これらコンポーネントにはまた、初期化モジュールからの作成したメニューヘッダに選択されたメディアファイルに対する参照を設定するための再生リストモジュールも含まれる。これらコンポーネントにはまた、再生リストモジュールからの選択されたメディアファイルのそれぞれに関連したメディアタイプを識別するための要約モジュールも含まれる。これらコンポーネントにはまた、要約モジュールからの識別したメディアタイプをメニューヘッダに設定するための伝搬モジュールも含まれる。メディアプレーヤーは、設定したメニューヘッダを使い、このメディアプレーヤーに関連したメディアタイプに応じて選択されたメディアファイルをフィルタする。

【０００８】

本発明のさらに他の態様によれば、この方法は、メディアプレーヤーに関連したディスプレイ上でメニュー情報を表示する。メニュー情報は、ユーザがメディアプレーヤーを使って１つまたは複数のメディアファイルを操作できるようにする。メディアファイルはそれぞれ、そのメディアファイルに関連したメディアタイプを有し、このメディアタイプに応じてメディアプレーヤーによるレンダリングのために適合させられる。メディアプレーヤーもまた、そのメディアプレーヤーに関連したメディアタイプを有している。この方法は、選択されたメディアファイルに対する参照と、選択されたメディアファイルのそれぞれ各１つに関連したメディアタイプとを有するメニューヘッダを読み取る。この方法はまた、メディアプレーヤーに関連したメディアタイプに応じて、選択されたメディアファイルに対する参照をフィルタすることにより読み取ったメニューヘッダを適合させる。この方法はまた、適合させたメニューヘッダからのメニュー情報をメディアプレーヤーに関連したディスプレイ上でユーザに表示する。

【０００９】

本発明のさらに他の態様によれば、メディアプレーヤーは、そのメディアプレーヤーに関連したディスプレイ上にメニュー情報を表示する。メニュー情報は、ユーザがメディアプレーヤーを使って１つまたは複数のメディアファイルを操作できるようにする。メディアファイルはそれぞれ、そのメディアファイルに関連したメディアタイプを有し、そのメディアタイプに応じてメディアプレーヤーによるレンダリングのために適合させられる。このメディアプレーヤーもまた、そのメディアプレーヤーと関連したメディアタイプを有する。このメディアプレーヤーには、１つまたは複数の選択されたメディアファイルに対する参照と、選択されたメディアファイルのそれぞれ各１つに関連したメディアタイプを有するメニューヘッダを読み取るための入力モジュールが含まれる。メディアプレーヤーと関連したメディアタイプに応じて、選択されたメディアファイルに対する参照をフィルタすることにより、入力モジュールからの読み取ったメニューヘッダを適合させるためのメニューモジュールも、このメディアプレーヤーに含まれる。メニューモジュールからの適合させたメニューヘッダからのメニュー情報をメディアプレーヤーに関連したディスプレイ上でユーザに表示するためのユーザインターフェイスモジュールも、このメディアプレーヤーに含まれる。

【００１０】

本発明のさらに他の態様によれば、コンピュータ可読媒体が、１つまたは複数のメディアファイルの編成を表すデータ構造をその上に格納している。メディアファイルはそれぞれ、そのメディアファイルに関連したメディアタイプを有し、メディアタイプに応じてメディアプレーヤーによるレンダリングのために適合させられる。メディアプレーヤーは、

そのメディアプレーヤーに関連したメディアタイプを有する。このデータ構造は、ユーザがメディアプレーヤーを使ってメディアファイルを操作できるようにする。このデータ構造には、１つまたは複数の選択されたメディアファイルに対する参照を格納する再生リストフィールドが含まれる。選択されたメディアファイルはそれぞれ、その選択されたメディアファイルに関連したメディアタイプを有する。このデータ構造にはまた、再生リストフィールドにおいて参照した、選択されたメディアファイルの１つに対するメディアタイプを格納する要約フィールドも含まれる。メディアプレーヤーは、データ構造を使い、このメディアプレーヤーに関連したメディアタイプに応じて、選択されたメディアファイルをフィルタする。

【００１１】

代替として、本発明は、他の様々な方法および装置を備えることもある。

その他の特徴は、一部には明らかであり、一部には下文において指摘することになるであろう。

【発明を実施するための最良の形態】

【００１２】

本発明は、メニュー構造を作成し、ユーザがメディアプレーヤーを使って１つまたは複数のメディアファイルを操作できるようにするためのオーサリングソフトウェアを含む。本発明はまた、作成したメニュー構造を適合させ、メディアプレーヤーに関連したメディアタイプに応じてメディアプレーヤーに関連したディスプレイ上にメニュー情報を表示するためのフィルタリングソフトウェアを含む。一実施形態において、本発明のソフトウェアは、機能の豊富なパーソナルコンピュータならびに低コストのメディアプレーヤー上での再生のために設計されたコンパイル済みのバイナリ形式のメタデータ、メニュー、および再生リストが入っている一組の小さなファイルを有する圧縮メディア形式（図１０参照）の一部として動作可能である。例えば、典型的な低価格帯のポータブルＣＤプレーヤーでは、ほんの百キロバイトの作業メモリと、わずか１メガヘルツで稼動している８ビットの中央処理装置と、最大シーク時間がほとんど５秒かかる非常に低速のＣＤ駆動機構しか有していないことがある。

【００１３】

（メディア環境）

ここで図面を参照すると、図１は、本発明を使用することのできる例示的なマルチメディア環境を示している。システム１００は、１つまたは複数のコンピュータ１０２を、メディアコンテンツを提供する１つまたは複数の装置に連結させている。例えば、これらの装置には、ＣＤ１０４、カムコーダ１０６、またはカメラ１０８を含むことができる。

【００１４】

一実施形態において、コンピュータ１０２は、民生用電子機器１１２と関連したメディアプレーヤープログラムによる使用のためにコンピュータ可読媒体１１０上にメディアコンテンツを格納する。民生用電子機器１１２は、ユーザが媒体１１０上に具現されているコンテンツを体験できるようにデジタルメディアをレンダリングするために構成される任意の適切なレンダリングフィルタ、またはメディアプレーヤーもしくは装置を含む。例えば、適切なメディアプレーヤーの用途には、ＣＤメディアプレーヤーやＤＶＤメディアプレーヤーが含まれる。

【００１５】

本発明において、メディアプレーヤー、民生用電子機器１１２、またはその種の他のものは、メディアプレーヤーの機能により３つの例示的なレベルに体系付けることができる。各メディアプレーヤーは、そのメディアプレーヤーがレンダリングすることのできるメディアのタイプを識別するメディアタイプを有する。例えば、メディアタイプ（再生リスト要約タイプ、メニュー要約タイプ、などとも称される）は、オーディオ、ビデオ、静止画像の内、１つまたは複数のものを含む。レベル１のメディアプレーヤーには、ポータブルＣＤプレーヤーや、カーレシーバ、ＤＶＤプレーヤーのようなオーディオのみのプレーヤーが含まれる。レベル１のメディアプレーヤーに加えて、レベル２のメディアプレーヤ

ーには、オプションで画像をオーディオと同時にレンダリングすることのできるポータブルおよびセットトップ型のＤＶＤプレーヤーを含めて、オーディオと静止画像の機能の付いたプレーヤーが含まれる。レベル３のメディアプレーヤーには、レベル２のメディアプレーヤーと、オーディオ、静止画像、およびビデオの機能の付いたプレーヤーとが含まれる。例示的な民生用電子機器１１２には、これらに限られないが、ポータブルＣＤプレーヤー、ＭＰ３プレーヤー、自動車のオーディオシステム、携帯情報端末、携帯電話、またはその種の他のものが含まれる。

【００１６】

本発明のソフトウェアおよびデータ構造は、これらに限られないが、メディアプレーヤーおよびメディアオーサリングシステムを含むシステムにおいて動作可能である。本発明によるメディアプレーヤーは、そのメディアプレーヤーに関連したディスプレイ上でメニュー情報を表示するソフトウェアを含む。例示的な民生用電子機器１１２またはメディアプレーヤーには、入力モジュール１１４、メニューモジュール１１６、ユーザインターフェイスモジュール１１８、および操作モジュール１２０が含まれる。入力モジュール１１４は、選択されたメディアファイルに対する参照と、その選択されたメディアファイルのそれぞれ各１つに関連したメディアタイプとを有するメニューヘッダを読み取る。例えば、メディアファイルおよびメニューヘッダは、コンピュータ可読媒体上に格納される。メディアファイルは、メディアタイプに応じてメディアプレーヤーによるレンダリングのために適合させられる。メニューモジュール１１６は、メディアプレーヤーに関連したメディアタイプに応じて、選択されたメディアファイルに対する参照をフィルタすることによって、読み取ったメニューヘッダを適合させる。ユーザインターフェイスモジュール１１８は、適合させたメニューヘッダからのメニュー情報をメディアプレーヤーに関連したディスプレイ上でユーザに表示する。一実施形態において、メニューヘッダは、少なくとも１つの他のメニューヘッダに対する参照と、そうした他のメニューヘッダに関連したメディアタイプとを含み、これらメニューヘッダ間での操作を可能にする。操作モジュール１２０は、メディアプレーヤーに関連したメディアタイプに応じて、他のメニューヘッダに対する参照をフィルタすることによって、読み取ったメニューヘッダを適合させる。

【００１７】

（再生リスト）

再生リストは、コンピュータ可読媒体上のオーディオ、ビデオ、および画像ファイルのグループを編成するのに便利な方法である。再生リストには、これらに限られないが、メディアファイル、オーディオファイルのグループ、ビデオファイルのグループ、時間動作の画像シーケンスのグループ、および画像のオーディオまたはビデオとの様々な複合並行的な組み合わせのグループの内の１つまたは複数のものが含まれる。例えば、ユーザは、異なる演者、または異なる種類の音楽もしくはビデオの再生リストを作成することができる。ユーザはまた、再生リストをシャッフルしたり、繰り返したりすることによって作成した再生リストを操作することもできる。再生リストは、ユーザがメディアファイルのリストを容易に閲覧し、ソートし、サーチし、そして素早く操作することができるようにする。

【００１８】

（オーサリングソフトウェア）

本発明のソフトウェアは、各種のプレーヤーおよびメディアタイプにわたって動作可能である。例として、オーディオと写真の入っているコンピュータ可読媒体は、オーディオのみのポータブルＣＤプレーヤーにおいてオーディオの再生機能を提供する。この機能をサポートするためにアクセラレータファイル内の各メニュー項目が、そのメニュー項目に含まれるコンテンツのタイプ（例えば、オーディオ、静止画像、ビデオ）でタグ付けされている。媒体の作成中に、これらのタグがメニューツリーを通して上へ伝搬され、そのためにどのレベルにおいても、あらゆる項目がタグ付けされて、その媒体に入っているすべての種類のメディアを示す。このタグを用いて、プレーヤーは、その機能に応じてメニューをフィルタすることができ、そのプレーヤーがレンダリングできるメニュー項目および

再生リストのみを表示することができる。

【００１９】

次に図２を参照すると、フローチャートにより、本発明によるオーサリングソフトウェアの例示的な動作が示されている。図２の例示的な実施形態において、オーサリングソフトウェアは、ユーザがメディアプレーヤーを使ってファイルを操作できるように１つまたは複数のメディアファイルを編成するためのコンピュータ実行可能コンポーネントを含んでいる。コンポーネントには、初期化モジュール、再生リストモジュール、要約モジュール、伝搬モジュール、メニューモジュール、および書き込みモジュールが含まれる。初期化モジュールは、２０４においてメニューヘッダを作成する。再生リストモジュールは、２０６において、この作成したメニューヘッダに再生リストとしてカプセル化された選択されたメディアファイルに対する参照を設定する。要約モジュールは、２０８において、それぞれの再生リスト内でそれぞれの選択されたメディアファイルに関連したメディアタイプを識別する。伝搬モジュールは、２１０において、メニューヘッダにこの識別したメディアタイプを設定する。メディアプレーヤーは、この設定したメニューヘッダを使い、このメディアプレーヤーに関連したメディアタイプに応じて、選択されたメディアファイルをフィルタする。メニューモジュールは、メニューヘッダ間での操作を可能にするために、２１２においてメニューヘッダに他のメニューヘッダ（例えば、少なくとも１つ）に対する参照を設定する。書き込みモジュールは、選択されたメディアファイルおよび設定したメニューヘッダを、２１４において、このメディアプレーヤーに関連したコンピュータ可読媒体上に格納する。メニューヘッダに設定する参照には、これらに限られないが、選択されたメディアファイルに対するメモリ位置のオフセット、選択されたファイルを参照している再生リストに対するポインタ（例えば、数値識別子）、または選択されたメディアファイル自体に対するポインタが含まれてもよい。一実施形態においては、１つまたは複数のコンピュータ可読媒体が、図２に示した動作を実行するためのコンピュータ実行可能命令を格納する。別の実施形態において、メニュー構造には、単一のメディアファイルを有する単一の再生リストが含まれる。

【００２０】

オーサリングソフトウェアの実行後、メニュー構造は、図３に示したようになる。本発明によるメニュー構造中の各要素は、タグまたはその要素に関連したメディアタイプを示すその他のデータを格納する。図３の例において、最上位のメニューは、ビデオデータを格納しているサブメニューとオーディオデータを格納しているサブメニューとを有している。オーディオデータを格納しているサブメニューは、オーディオデータを体系付けている更なるサブメニューを有している。ユーザは、メニュー構造の経路３０２を通ってビデオコンテンツへ、あるいはメニュー構造の経路３０４を通ってオーディオコンテンツへ行くことができる。

メニューおよび再生リストの作成に関して次にさらに詳述する。

【００２１】

（再生リストおよびメニューの作成）

このセクションでは、選択されたメディアファイルから再生リストを作成し、再生リストを階層構造において編成するオーサリングソフトウェアによって実行される例示的な方法を説明する。この方法の動作は、選択されたファイルのタイプと現行の編成に基づいている。選定されたメディアファイルが単一層のディレクトリにある場合、オーサリングソフトウェアは、メディアファイルをファイル名でアルファベット順にディレクトリ名別にグループ化する「ディレクトリ別のすべてのメディアファイル」のような再生リストを作成する。二層のディレクトリがあり、平均的な最上位のディレクトリには１つ半未満のサブディレクトリが入っていて、７５％を超える第２レベルのディレクトリには、５～２５のオーディオファイルが入っている場合（例えば、各アーティストにつき２～３のアルバムのみの典型的なアーティストアルバムの構成）、このオーサリングソフトウェアは、再生リスト「ディレクトリ別のすべてのメディアファイル」を作成し、この再生リストの中でメディアファイルをアルファベット順に＜ディレクトリ名１＞そして＜ディレクトリ名

２＞として＜ディレクトリ名１＋「‐」ディレクトリ名２＞別にグループ化する。

【００２２】

三層以上のディレクトリがあるか、または階層を安易に平坦化するには二層では「枝葉」が多すぎる（例えば、ディレクトリごとに過度に多くのサブディレクトリが入っている）場合、このオーサリングソフトウェアは、最初にこのファイルシステムの全幅を横断する際に（アルファベット順ではなく）ディレクトリがこのファイルシステム中で見つかる順番にメディアファイルを＜最下層のディレクトリ名＞別にグループ化する再生リスト「ディレクトリ別のすべてのメディアファイル」を作成する。ヌルでない再生リストファイルが１つでもあれば、オーサリングソフトウェアは、存在する各再生リストにつき再生リストを作成する。このような再生リストが６つ以上ある場合、このような再生リストは「再生リスト」メニューの下に格納される。２～６つの再生リストが存在する場合、オーサリングソフトウェアは、「その他」と題するメニュー項目の下に残りのすべてのメニューを作成する。

【００２３】

（オーディオメディアファイルの再生リスト）

オーサリングソフトウェアは、作成した再生リストを編成するための階層メニュー構造を作成する。オーディオメディアファイルでの例示的な実施形態において、オーサリングソフトウェアは、アーティスト、作曲家、アルバム、アーティストとアルバム、ジャンル、年、およびディレクトリに従ってグループおよび対応するメニューを作成する（例えば、選択されたメディアファイルの既存の構造に対応して）。

【００２４】

階層メニュー構造のルートにおいて、「アーティスト」と題するメニューは、メディアファイルをアルファベット順でアーティスト別にグループ化する「アーティスト別のすべての曲」と題するサブメニューを有する。オーサリングソフトウェアは、アーティストに関連するメディアファイルをアルファベット順にアルバム別に、またはアルバムのリリース日に従ってグループ化するアーティストごとのサブメニューを作成する。作曲家の情報が一部の選択されたメディアファイルについて利用可能であれば（例えば、２５％超）、オーサリングソフトウェアは、メディアファイルをアルファベット順に作曲家別にグループ化する「作曲家別のすべての曲」と題するサブメニューを有するメニュー「作曲家」をルートに作成する。オーサリングソフトウェアは、作曲家に関連するメディアファイルをアルファベット順にアルバム別にグループ化する、またはアルバムのリリース日に従ってグループ化する作曲家ごとのサブメニューを作成する。作曲家の情報を持たないメディアファイルは、省略される。

【００２５】

オーサリングソフトウェアはまた、メディアファイルをアルファベット順にアルバム別に、またはアルバムのリリース日に従ってグループ化する「アルバム別のすべての曲」と題するサブメニューを有するメニュー「アルバム」をルートに作成する。オーサリングソフトウェアはさらに、メディアファイルをアーティストとアルバムのペア（即ち、アーティストとアルバムの組み合わせごとに単一のグループを作成する）に従ってグループ化するメニュー「アーティスト‐アルバム」をルートに作成する。オーサリングソフトウェアはまた、メディアファイルをアルファベット順にジャンル別に（即ち、ジャンルごとに単一のグループを作成する）グループ化するサブメニュー「ジャンル別のすべての曲」を有するメニュー「ジャンル」を作成する。メニュー「年」には、これらに限られないが、サブメニュー「年代別のすべての曲」、「年別のすべての曲」、および「ディレクトリ」が含まれる。サブメニュー「年代別のすべての曲」は、「年なし」の曲をグループの最後にグループ化するか、または省略して、昇順で年代別にグループ化した（例えば、年代ごとに１つのグループ）メディアファイルを含む。サブメニュー「年別のすべての曲」は、「年なし」の曲をグループの最後にグループ化するか、または省略して、昇順で年別にグループ化した（例えば、年ごとに１つのグループ）メディアファイルを含む。サブメニュー「ディレクトリ」は、選択されたメディアファイルの既存のディレクトリ構造と平行する

グループを有する。すなわち、サブメニュー「ディレクトリ」は、ディレクトリごとに１つのグループを有する（例えば、「パーティのお気に入り」、「運転中の音楽」）。

【００２６】

当業者は、ここに記載した再生リストおよび階層構造が単に例示にすぎないことを理解するであろう。本発明には、オーディオデータ、ビデオデータ、または静止画データを含めて、他の再生リスト、階層構造、その他同種類のものが含まれると発明者に考えられている。

【００２７】

（画像メディアファイルの再生リスト）

選択されたメディアファイルが静止画像を含む場合、オーサリングソフトウェアは、（例えば、デフォルトの５秒間で）表示するためのあらゆる選択された画像を含めて、メニュー「すべてのピクチャ」をルートに作成する。一実施形態において、この時間は、オーサリングソフトウェアに関連したユーザインターフェイスの上級オプションダイアログを介して変更することができる。さらに、このオーサリングソフトウェアは、現行のディレクトリにおいて選択された画像および選択された音楽ファイルの相互一致を検査する。画像（例えば、２００×２００ピクセルより大きい）を有するディレクトリのすべてが音楽も有する場合、オーソリングソフトウェアは、ディレクトリ別に配列された「すべての曲を伴うすべてのピクチャ（エンドレス）」と題する再生リストを作成し、各ディレクトリはその付随する音楽を伴って再生を行う。一実施形態において、各画像の表示時間は、ディレクトリ中の曲の長さの合計を画像の数で割ったものに等しい。例えば、各画像は、最短で５秒間そして最長で３０秒間、表示することができる。この最長および最短の設定は、上級オプションダイアログにおいてユーザ調整可能である。ディレクトリ中の音楽がディレクトリにあるすべての画像には短すぎる場合、この技法を用いると一部の画像は表示されないことがある。

【００２８】

代替として、再生リスト「すべての曲を伴うすべてのピクチャ（エンドレス）」は、すべての画像と並行して再生するすべての曲を含む。一実施形態において、各画像の表示時間は、選択されたメディアファイルにあるすべての曲の長さの合計を画像の数で割ったものに等しい。例えば、各画像は、最短で５秒間そして最長で３０秒間、表示される。この最長および最短の設定は、上級オプションダイアログでユーザ調整可能である。画像のシーケンスは、音楽の長さに合うように必要に応じて繰り返されるか、あるいは音楽が再生リストの中で繰り返されて、合計の画像の長さに合うように十分な音楽があるようにする。画像は、例えば、最下位のディレクトリ名別にグループ化される。

【００２９】

（ビデオメディアファイルの再生リスト）

選択されたメディアファイルがビデオを含む場合、オーサリングソフトウェアは、階層構造のルートにメニュー「ビデオ」を作成する。このメニューは、各ビデオファイルの再生リスト、ならびに英数字のシーケンスで連結したビデオファイルを有する「すべてのビデオの再生」と題する全ビデオファイルの再生リストを含む。オーサリングソフトウェアはまた、以下のセクションで説明するように、ユーザ入力に従ってビデオメディアファイルをグループ化する。

【００３０】

（自動再生リストに対するマニュアル調整）

ユーザは、階層構造および／またはデフォルトの再生リストを調整することを選んでもよい。オーサリングソフトウェアに関連したユーザインターフェイスは、コンピュータ可読媒体上で含めるための全オプションが事前にチェックマークを付けられた状態で、階層構造および再生リストのダイアグラムを表示する。例えば、メニュー／再生リストの構造は、各メニューまたは再生リスト項目の隣のチェックボックスと共にスクロールするツリー表示として表示される。チェックボックスをクリアすることにより、ツリーのその枝全体を消去する（とはいえ、その下の項目の状態は、ユーザがそれを元に戻す場合に備えて

保持されているが）。

【００３１】

オーサリングソフトウェアに関連したユーザインターフェイスは、様々な特徴を実装したり、あるいは無効にしたりすることができる。例えば、昇格および降格ボタンは、現在の選択されている項目を昇格したり、降格したりできなければ、選択できないようにしてもよい（例えば、グレー化して）。項目をどこに降格するかについてユーザには選択の機会が与えられておらず、必要に応じて見出しを付け加えて、自動的に正しい見出しの下に移動する。例えば、最上位のメニューに３つのビデオファイルがあり、「ビデオ」メニューのエントリがなければ、初回の降格では見出し「その他のビデオ」を作成し、２回目の降格ではそれに追加し、３回目の降格では見出しを「ビデオ」に名称変更する。メニューもしくは再生リストの順番を変更したり、またはそれらの名称を変えたりすることについて選択肢は与えられていない。すべての名称は、ファイル中のメタデータから導き出されるか、または「その他のビデオ」のように名称を自動的に生成する。個別の再生リストが、その収容しているカテゴリーと同じレベルに昇格されると、この収容カテゴリーは、名称を「その他の．．．」（例えば、「その他のビデオ」）に変更する。

【００３２】

オーサリングソフトウェアに関連したユーザインターフェイスの他のオプションには、これらに限られないが、「最短の画像持続時間」、「デフォルトの画像持続時間」、「最長の画像持続時間」、および「永久に繰り返し（キオスクモード）」を調整することが含まれる。「最短の持続時間」オプションは、ユーザが画像の最短持続時間を変更できるようにする。最短の持続時間は、画像とオーディオの再生リストを並行して作成する際に、最長値と共に使われる。「デフォルトの画像持続時間」オプションは、ユーザが画像のデフォルトの持続時間を変更できるようにする。デフォルトの持続時間は、画像シーケンスの一部として再生する画像に使われる。「最長の画像持続時間」オプションは、ユーザが画像の最長持続時間を変更できるようにする。このオプションは、画像とオーディオが一緒に提示されるときに使われる。「永久に繰り返し（キオスクモード）」オプションは、各再生リストの繰り返しカウントフィールドをゼロに設定させる。このオプションがチェック付けされていなければ、繰り返しカウントは、各再生リストにつき「１」に設定される。一実施形態において、再生リストを１つにまとめたり、自動的に作成されたメニュー名およびメニュー構造のさらに高度な編集を行ったりするオプションをユーザに与えることもできる。ユーザはまた、オーサリングソフトウェアを実行する前に再生リストファイルを作成することもできる。当業者は、オーサリングソフトウェアが多かれ少なかれここに記載された機能を含むことができて、それでもなお本発明の適用範囲内であることを察するであろう。

【００３３】

階層構造および／またはデフォルトの再生リストに何らかのマニュアル調整が為された後、図４の例示的なメニュー構造に示したようなメニュー構造になる。図４において、メニューヘッダ４０２は、「オーディオ」４０４、「ビデオ」４０６、および「画像」４０８と題するメニュー項目を含む。「オーディオ」メニュー項目４０４は、「ジャンル別のすべて」の再生リスト４１２、「再生リストA」の再生リスト４１４、および「アーティストメニュー」のメニュー項目４１６を含むメニューヘッダ４１０を参照している。「アーティストメニュー」のメニュー項目４１６は、更なるメニュー項目または再生リスト項目を格納する別のメニューヘッダ４１８を参照している。各メニューヘッダは、親のメニューヘッダへの参照を含む。例えば、メニューヘッダ４１８は、メニューヘッダ４１０に対する参照を有している。同様に、メニューヘッダ４１０は、メニューヘッダ４０２に対する参照を有している。この例示的なメニュー構造のルートにあるメニューヘッダ４０２は、親のメニューヘッダを有していない。

【００３４】

一実施形態において、オーサリングソフトウェアは、１つの子のみを有するメニューをサーチして、作成されたメニュー構造を横断するメニュー簡易化機能（menu simplifier

）を含む。メニュー簡易化機能は、この種のメニューすべてを取り除き、この取り除いたメニューに代えて階層中のこの唯一の子を昇格させる。このようにして、メニュー簡易化機能は、メニュー構造の複雑さを低減し、ユーザによる操作を簡易化する。

【００３５】

オーサリングソフトウェアは、デフォルト設定、およびユーザのどのようなマニュアル設定を共に記録して、コンピュータ可読媒体にセットアップファイルを（例えば、拡張マークアップ言語で）書き込む。ユーザが後で同じ媒体（例えば、再書き込み可能な媒体）にさらにメディアファイルを追加する場合、オーサリングソフトウェアは、セットアップファイルを調べて、こうした同一の設定を識別し、適用する。例えば、ユーザがデフォルトの再生リストのすべてを削除し、「最下位のディレクトリ名別のすべての曲」のオプションのみを選択する場合、ユーザがそのコンピュータ可読媒体にメディアファイルを追加しようとするたびに、この種のオプションがこの媒体のデフォルトになるであろう。オーサリングソフトウェアは、ユーザにとって特定の媒体のこのようなカスタム化されたオプションを識別する。

【００３６】

（フィルタリングソフトウェア）

次に図５を参照すると、フローチャートにより、本発明によるフィルタリングソフトウェアの例示的な動作が示されている。フィルタリングソフトウェアは、メディアプレーヤーに関連したディスプレイ上にメニュー情報（例えば、メタデータ）を表示する。メニュー情報は、ユーザがメディアプレーヤーを使って１つまたは複数のメディアファイルを操作できるようにする。それぞれのメディアファイルは、メディアタイプを有し、このメディアタイプに応じてメディアプレーヤーによるレンダリングのために適合可能である。メディアプレーヤーもまた、メディアタイプを有している。フィルタリングソフトウェアは、５０２において、選択されたメディアファイルに対する参照と、選択されたメディアファイルのそれぞれ各１つに関連したメディアタイプを有するメニューヘッダを読み取る。フィルタリングソフトウェアは、５０４において、メディアプレーヤーに関連したメディアタイプに応じて、選択されたメディアファイルに対する参照をフィルタすることによって、読み取ったメニューヘッダを適合させる。一実施形態において、フィルタリングソフトウェアは、ユーザインターフェイスで選択することのできないメニュー項目に陰影をつけて（例えば、グレーに変えて）、これらのメニュー項目を選択可能なメニュー項目から区別する。フィルタリングソフトウェアは、５０６において、適合させたメニューヘッダからのメニュー情報をメディアプレーヤーに関連したディスプレイ上でユーザに表示する。一実施形態において、メニューヘッダは、別のメニューヘッダ対する参照と、この別のメニューヘッダに関連したメディアタイプを含んで、メニューヘッダ間の操作を可能にする。フィルタリングソフトウェアはまた、５０８において、メディアプレーヤーに関連したメディアタイプに応じてこの別のメニューヘッダに対する参照をフィルタすることによって読み取ったメニューヘッダを適合させる。選択されたメディアファイルのそれぞれ各１つに関連したメディアタイプと、メディアプレーヤーに関連したメディアタイプは、オーディオ、ビデオ、および／または画像とを備えている。選択されたメディアファイルに対する参照と、別のメニューヘッダに対する参照には、これらに限られないが、選択されたメディアファイルに対するメモリ位置のオフセット、選択されたメディアファイルを参照する再生リストに対するポインタ、または選択されたメディアファイルに対するポインタが含まれる。一実施形態において、１つまたは複数のコンピュータ可読媒体は、図５に示した方法を行うためのコンピュータ実行可能命令を有する。

【００３７】

（メニュー表示）

本発明のソフトウェアによって作成された媒体上のメニューは、サブメニューおよび再生リストの単一階層構造を含む。メニューツリーは、例えば、総ノード数９，９９９の全体としての上限の範囲内で、任意に深く、任意に広くすることができる。他の実施形態では、メニューツリーに関して他のまたは異なる規制を設けることもできる。各メニューが

９，９９９エントリまで収容することができるので、メディアプレーヤーは、スクロールするまたはページ化したメニューの一覧を表示する。テレビ接続されたプレーヤーでは、サムネイルメニューまたは簡単なリストメニューを表示してもよい（例えば、現在のメニュー内の再生リストまたはメニューのようなエントリがサムネイルを有しているか否かに基づいて）。一実施形態において、テレビ接続されたプレーヤーは、現在のメニューのあらゆるエントリにサムネイルがあるとき（即ち、ゼロでないエントリ）に限り、サムネイルメニューを表示する。

【００３８】

画面上のサムネイルのレイアウトおよびページごとに表示されるサムネイルの数は、プレーヤーに依存する。一部のプレーヤーでは、途切れなくスクロールするリストを表示するようにすることもできるし、他のプレーヤーでは、メニューをページ割りし、次／前の選択を用意するようにすることもできる。プレーヤーは、４：３のテレビ受像機上では、通例、グリッド３×２のサムネイルを表示するであろうが、１６：９のテレビ受像機上では、代わりに４×２のグリッドを表示するようにすることもできる。

【００３９】

プレーヤーはまた、再生リストまたはメニュー名をその関連したサムネイルと共に表示する。プレーヤーがすべてのサムネイルを瞬時に表示することができない場合には、テキストのラベルを先ず表示し、サムネイルがデコードされるに従ってサムネイルを１つずつ付加する。プレーヤーは、ユーザがある所与の項目に素早く操作できるようにするために、サムネイルが付加されている間でさえも、次または前のページへスクロールできるようにしてもよい。

【００４０】

（メニュー操作）

メニュー表示は、プレーヤーに依存する。つまり、各製造業者がメニューをどのように表示するかを決めることができるということである。ポータブルＣＤプレーヤー上では、例えば、選択を通して操作するのに使われる「次」、「前」、「再生／選択」、「停止」ボタンと共に、メニューを一度に一行でメニュー表示することもある。ほとんどのプレーヤーは、媒体が挿入されたときに最上位のメニューを表示するであろう。しかし、カーレシーバは、起動時にメニューをバイパスし、ディスク上の最初の再生リストで直ちに再生を始めることもできるし、あるいは、前回使われた時に再生されていた曲および再生リストで再生を再開することもできる。ユーザは次に、メニューをもたらすために「メニュー」キーを押す必要があり、そして「次」、「前」、および「再生／選択」を使い操作するであろう。

【００４１】

ＤＶＤプレーヤー上では、背景画像およびオプションのサムネイルを備えたフルグラフィック表示を示すことができるであろう。一部のＤＶＤプレーヤーでは、メニュー項目を番号付けし、リモコン上の１～９のキーを使うようにすることもできるし、一部では、単にリストまたはグリッド形式でメニュー項目を提示し、操作と選択にカーソルキーを使うこともできる。

【００４２】

図６は、操作に利用可能なメディアタイプを示す本発明のフィルタリングソフトウェアの例示的なユーザインターフェイス６０２のスクリーンショットを示している。図６のこのユーザインターフェイス例６０２では、ユーザによる選択のために利用可能なアイコンには、ビデオファイルのアイコン６０４、オーディオファイルのアイコン６０６、画像ファイルのアイコン６０８、およびプレゼンテーションのアイコン６１０が含まれる。

【００４３】

（メニューのデータ構造）

次に図７を参照すると、例示的なブロック図として、ファイルヘッダ７０２と、メニュー＃１およびメニュー＃２のような１つまたは複数のメニュー７０４を有するメニュー構造７００が示されている。各メニュー７０４は、メニューヘッダ＃１およびメニューヘッ

ダ＃２のようなメニューヘッダ７０６を含む。コンピュータ可読媒体が、１つまたは複数のメディアファイルの編成を表すメニューヘッダ７０６を格納している。メディアファイルはそれぞれ、メディアタイプを有し、このメディアタイプに応じてメディアプレーヤーによるレンダリングのために適合可能である。メディアプレーヤーもまた、このメディアプレーヤーがレンダリングすることのできる少なくとも１つのメディアタイプを有している。図７に示したメニューヘッダ７０６は、ユーザがメディアプレーヤーを使ってメディアファイルを操作することができるようにしている。メニューヘッダ７０６は、再生リストフィールド（例えば、再生リスト項目７０８）および要約フィールド（例えば、再生リスト要約タイプ）を備えている。再生リストフィールドは、１つまたは複数の選択されたメディアファイルに対する参照を格納する。参照には、これらに限られないが、選択されたメディアファイルに対するメモリ位置のオフセット、選択されたメディアファイルを参照している再生リストに対するポインタ、または選択されたファイルに対するポインタが含まれる。それぞれの選択されたメディアファイルは、関連したメディアタイプを有している。要約フィールドは、再生リストの中で参照されている選択されたメディアファイルのメディアタイプを格納する。メディアプレーヤーは、メニューヘッダ７０６を使って、このメディアプレーヤーに関連したメディアタイプに応じて選択されたメディアファイルをフィルタする。

【００４４】

一実施形態において、図７に示したメニューヘッダ７０６は、別のデータ構造（例えば、メニュー＃２のような別のメニューヘッダ７０６）に対する参照を格納しているメニューフィールド（例えば、メニュー項目７１０）を含み、データ構造間で（例えば、メニューヘッダ＃１とメニューヘッダ＃２との間で）操作を可能にする。さらに、メニューヘッダ７０６は、オーサリングソフトウェアに応じて、選択されたメディアファイルのタイトル、作曲家、演者、ジャンル、およびコンテンツの説明の内の１つまたは複数から選択されたメタデータに対応する値を格納しているメタデータフィールド（例えば、メニュー名、メニューの中の再生リスト名）を含むこともできる。

【００４５】

図７に示した１つまたは複数のデータ構造を収容しているファイル（例えば、MENU.HMT）は、選択されたメディアファイルを格納しているコンピュータ可読媒体に書き込まれる。このファイルは、ファイル中のデータ構造間に隙間がないように書き込まれる。オーサリングソフトウェアおよびフィルタリングソフトウェアによる使用のための例示的なデータ構造の説明に移る。例示的なメニューファイルヘッダ７０２のフィールドを表１に示す。

【００４６】

【表１】

表１－メニューファイルヘッダ

オフセット	長さ	フィールド名
0	8	識別子
8	2	バージョン
10	4	MENU.HMTのサイズ
14	4	ＬＣＩＤ
18	2	最初のメニューのオフセット
20	2	メニュータイトルの長さ
22	可変	メニュータイトル

【００４７】

「識別子」フィールドは、ＡＳＣＩＩで「MENU_HMT」を収容する８バイトのエントリである。「バージョン」フィールドは、このファイルを作成するのに使われた仕様のバージョンを表す２バイトのエントリである。例えば、バージョン１．２は、０ｘ７８（１０進法で１２０）として格納されるであろう。「MENU.HMTのサイズ」フィールドは、バイトで

MENU.HMTのサイズを収容する４バイトのエントリである。「ＬＣＩＤ」フィールドは、このMENU.HMTファイルの言語ＩＤのための４バイトのエントリである。「最初のメニューのオフセット」フィールドは、MENU.HMTの先頭から最初の「メニューヘッダ」７０６までのバイトオフセットを表す２バイトのエントリである。「メニュータイトルの長さ」フィールドは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、メニュータイトルのバイト長を収容する２バイトのエントリである。「メニュータイトル」フィールドは、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了したメニュータイトルを格納する。このエントリの最大長は、ヌルのＵＣＳ－２の１文字を含めて、ＵＣＳ－２の６５文字である。プレーヤーは、これを使用して、メニュー全体のタイトルを表示する。空の文字列（「NULL」の１文字）は、表示するタイトルがないか、オーサリングソフトウェアが背景画像の上にタイトル、例えば、「私のハワイの休日」、「２００２スクールパーティ」、をレンダリングしたことを表している。

【００４８】

メニュー７０４は、背景画像、無地の背景色、またはプレーヤーのデフォルトの様式のいずれかをサポートする。背景画像または背景色が定義されると、テキスト色も定義される。背景画像、背景色、およびテキスト色のエントリがゼロだと、プレーヤーはそのデフォルトの様式を使う。各サブメニューは、単一の親メニューによって参照され、厳密な階層メニュー構造を形成している。例示的なメニューヘッダ７０６のフィールドを表２に示す。

【００４９】

【表２】

表２－メニューヘッダ

オフセット	長さ	フィールド名
	4	メニューヘッダのサイズ
	4	親メニューに対するオフセット
	4	背景画像ＩＤ（４：３）
	4	背景画像ＩＤ（１６：９）
	4	背景色
	4	テキストの色
	2	項目の数
	2	メニューサブタイトルの長さ
	可変	メニューサブタイトル
	可変	メニューまたは再生リスト項目＃１
		・・・
	可変	メニューまたは再生リスト項目＃ｎ

「注」ｎはメニュー項目の数を表す

【００５０】

「メニューヘッダのサイズ」フィールドは、バイトで「メニュー項目」７１０および「再生リスト項目」７０８を含む「メニューヘッダ」７０６のサイズを表す４バイトのエントリである。「親メニューに対するオフセット」フィールドは、MENU.HMTの先頭から「親メニュー」までのバイトオフセットを表す４バイトのエントリである。この値は、それが最上位レベルのメニュー７０４であれば、ゼロである。「背景画像ＩＤ（４：３）」フィールドは、４：３のディスプレイ上でこのメニュー７０４の背景として表示すべき画像を定義する４バイトのエントリである。画像は、HMT拡張子を有する６４０×４８０のＪＰＧファイルとする。ゼロの値は、背景画像がないことを表す。この値は、CONTENTS.HMT中の「メニュー画像」ファイルのインデックスである（図９参照）。プレーヤーは、アスペクト比を保持してディスプレイ上の中心に背景画像を表示する。プレーヤーは、背景色があればこれを使い、何ら覆われていない領域を埋める。「背景画像ＩＤ（１６：９）」フ

ィールドは、１６：９のディスプレイ上でこのメニューの背景色として表示すべき画像を定義する４バイトのエントリである。画像は、HMT拡張子を有する８５２×４８０のＪＰＧファイルとする。ゼロの値は、背景色がないことを表す。この値がゼロでなければ、有効な「背景画像ＩＤ（４：３）」も同じようにあるべきである。この値は、CONTENTS.HMT中の「メニュー画像」ファイルのインデックスである。プレーヤーは、アスペクト比を保持してディスプレイ上の中心に背景画像を表示する。

【００５１】

「背景色」フィールドは、このメニューをディスプレイ上にレンダリングするときに使われるべき背景色を定義する４バイトのエントリである。それはＲＧＢ値として、０ｘＦＦＲＲＧＧＢＢのバイト順で設定される。「背景画像ＩＤ」が定義されると、背景色は、この背景画像によって覆われていないディスプレイの領域があれば、その領域に見えるのみである。ゼロの値は、背景色がないことを表す。プレーヤーが色のレンダリングができなければ、このフィールドは無視してもよい。「テキスト色」フィールドは、このメニュー７０４上のテキストをディスプレイ上でレンダリングすべき色を定義する４バイトのエントリである。それはＲＧＢ値として、０ｘＦＦＲＲＧＧＢＢのバイト順で設定される。「背景画像ＩＤ」または「背景色」が定義されると、このエントリにはゼロでない値が入るはずである。ゼロの値は、プレーヤーがそのデフォルトの様式を使うべきであるということを意味するように定義されている。プレーヤーが色のレンダリングができなければ、このフィールドは無視してもよい。「項目の数」フィールドは、このメニュー７０４におけるメニュー項目７１０または「再生リスト項目」７０８の数を定義する２バイトのエントリである。「メニューサブタイトルの長さ」フィールドは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、メニューサブタイトルのバイト長を収容する２バイトのエントリである。「メニューサブタイトル」フィールドは、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了したメニューサブタイトルである。このエントリの最大長は、ヌルのＵＣＳ－２の１文字を含めて、ＵＣＳ－２の６５文字である。プレーヤーは、これを使用して、このメニュー７０４のサブタイトルを表示する。空の文字列（「NULL」の１文字）は、表示するサブタイトルがないか、オーサリングソフトウェアが背景画像上にタイトルをレンダリングしたことを表している。「メニューまたは再生リスト項目」フィールドは、「メニュー項目」７１０または「再生リスト項目」７０８のいずれかを表す可変サイズのエントリである。

例示的なメニュー項目７１０のフィールドを表３に示す。

【００５２】

【表３】

表３－メニュー項目

オフセット	長さ	フィールド名
0	1	エントリのタイプ
1	1	メニュー要約タイプ
2	4	サムネイルＩＤ（サムネイルがない場合は、０）
6	4	選択された状態のサムネイルＩＤ
10	4	メニューに対するオフセット
14	2	メニュー名の長さ
16	可変	メニュー名

【００５３】

「エントリのタイプ」フィールドは、これがメニュー項目７１０のデータ構造か、あるいは再生リスト項目７０８のデータ構造かを定義する１バイトのエントリである。メニュー項目７１０については、この値は、表４において定義した「MENU」の値となる。

【００５４】
【表４】
表４－エントリのタイプ

エントリのタイプ	値
0	UNUSED
1	MENU
2	PLAYLIST
3～255	RESERVED

【００５５】
「メニュー要約タイプ」フィールドは、このメニュー項目を介してアクセス可能な再生リストのタイプを定義するために使われる。「サムネイルＩＤ」フィールドは、CONTENTS.HMTの中のこのメニュー項目７１０のための「サムネイルＩＤ」を表す４バイトのエントリであり、このメニュー項目７１０にサムネイルがなければ、この値はゼロである。「選択された状態のサムネイルＩＤ」フィールドは、CONTENTS.HMTの中のこのメニュー項目７１０のための選択された状態を表す「サムネイルＩＤ」を定義する４バイトのエントリである。ゼロの値は、色を使って選択を指示するためにプレーヤーが境界線のある長方形（ｂｏｕｎｄｉｎｇ　ｒｅｃｔａｎｇｌｅ）または他の強調表示を生成することを表している。「メニューに対するオフセット」フィールドは、MENU.HMTの先頭から「メニュー」７０４までのバイトオフセットを定義する４バイトのエントリである。「メニュー名の長さ」フィールドは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、メニュー名のバイト長を収容する２バイトのエントリである。「メニュー名」フィールドは、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了した「メニュー」７０４の名称である。このエントリの最大長は、ヌルのＵＣＳ－２の１文字を含めて、ＵＳＣ－２の６５文字である。
例示的な再生リスト項目７０８のフィールドを表５に示す。
【００５６】
【表５】
表５－再生リスト項目

オフセット	長さ	フィールド名
0	1	エントリのタイプ
1	1	再生リスト要約タイプ
2	4	サムネイルＩＤ（サムネイルがない場合は、０）
6	4	選択された状態のサムネイルＩＤ
10	4	再生リストＩＤ
14	4	グループインデックスの開始
18	4	ファイルインデックスの開始
22	2	メニュー中の再生リスト名の長さ
24	可変	メニュー中の再生リスト名

【００５７】
「エントリのタイプ」フィールドは、これがメニュー項目７１０であるか、あるいは再生リスト７０８であるかを定義する１バイトのエントリである。再生リスト項目７０８については、この値は、「PLAYLIST」の値となる。「再生リスト要約タイプ」フィールドは、このメニュー項目７１０が参照する再生リストのタイプを定義するのに使われる。「サムネイルＩＤ」フィールドは、CONTENTS.HMTの中のこのメニュー項目７１０のための「サムネイルＩＤ」を定義する４バイトのエントリである。このメニュー項目にサムネイルがなければ、この値はゼロである。サムネイルＩＤは、再生リストヘッダ中のサムネイルＩＤと同じであってもよい。「選択された状態のサムネイルＩＤ」フィールドは、CONTENTS.H

MTの中のこのメニュー項目７１０のための選択された状態を表す「サムネイルＩＤ」を定義する４バイトのエントリである。ゼロの値は、色を使って選択を指示するためにプレーヤーが境界線のある長方形または他の強調表示を生成することを表している。「再生リストＩＤ」フィールドは、CONTENTS.HMTの中のこのメニュー項目７１０のための再生リストのＩＤを定義する４バイトのエントリである。「グループインデックス開始」フィールドは、再生を開始するために再生リストファイルの中の「グループ」のインデックスを定義する４バイトのエントリである。１の値は、再生リストの中の最初のグループを表す。「ファイルインデックス開始」フィールドは、再生を開始するためにグループの中の「ファイル」のインデックスを定義する４バイトのエントリである。１の値は、グループの中の最初のファイルを表す。さらに、この値は、「グループ」が「並行」再生リストグループ（Ｐａｒａｌｌｅｌ　ｐｌａｙｌｉｓｔ　ｇｒｏｕｐ）（ＰＩＡ）の場合、１である。

【００５８】

「グループインデックス開始」および「ファイルインデックス開始」は共に、１つの再生リストをメニュー７０４の中で複数回参照できるようにする。例えば、メニュー７０４は、ディスク上のあらゆる画像についてサムネイルを示すことができ、各サムネイルは選択した画像で始まる全画像の再生リストのルーピングをもたらすであろう。「メニュー中の再生リスト名の長さ」フィールドは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、メニュー７０４の再生リストのバイト長を収容する２バイトのエントリである。「メニュー中の再生リスト名」フィールドは、「メニュー」７０４に現れるとおりの「再生リスト」の名称である。名称は、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了する。このエントリの最大長は、ヌルのＵＣＳ－２の１文字を含めて、ＵＣＳ－２の６５文字である。

【００５９】

「メニュー画像ファイルテーブル」は、MENU.HMTファイルの中で使われるサムネイルおよび背景画像のすべてをリストするために使われる（補足説明Ａ参照）。「メニュー画像ファイルテーブル」は、ファイル名テーブル（表７参照）が次に続く「メニュー画像ファイルエントリ」のリストを含む。各「メニュー画像ファイル」につき１つの「メニュー画像ファイルエントリ」が存在する。各「メニュー画像」ファイルエントリは、表６に示すように設定される。

【００６０】

【表６】

表６－メニュー画像ファイルエントリ形式

オフセット	長さ	フィールド名
0	4	ディレクトリ番号に対するオフセット
4	2	ファイルタイプ
6	2	特別フラグ

【００６１】

【表７】

表７－ファイル名テーブル

オフセット	長さ	フィールド名
	4	ディレクトリ番号ＣＩＤ＃ｎ＋１
	2	ファイル名の長さＣＩＤ＃ｎ＋１
	可変	ファイル名ＣＩＤ＃ｎ＋１
		・・・
	4	ディレクトリ番号ＣＩＤ＃ｍ
	2	ファイル名の長さＣＩＤ＃ｍ
	可変	ファイル名ＣＩＤ＃ｍ

「注」ｍ－ｎは「メニュー画像」ファイルの数を表す

【００６２】

「ディレクトリ番号に対するオフセット」フィールドは、CONTENTS.HMTの先頭からこのエントリのディレクトリ番号までのバイトオフセットを表す４バイトのエントリである。「ファイルタイプ」フィールドは、「ファイル」タイプ（例えば、データのエンコーディング形式）を表す２バイトのエントリである。有効な値には、サムネイル、メニューの背景（４×３）、およびメニューの背景（１６×９）が含まれる。「特別フラグ」フィールドは、特別なフラグを保持する２バイトのエントリである。「ディレクトリ番号」フィールドは、このファイルが入っているディレクトリのディレクトリテーブルの中のインデックスを表す４バイトのエントリである。「ファイル名の長さ」フィールドは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、ファイル名のバイト長を収容する２バイトのエントリである。「ファイル名」フィールドは、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了したファイル名である。このエントリの最大長は、ヌルのＵＣＳ－２の１文字を含めて、ＵＣＳ－２の１１１文字である。このエントリは、「ビッグエンディアン」のワードレイアウトを使用する。

【００６３】

（多言語のサポート）

図７に示したデータ構造（例えば、メニューヘッダ）７０６は、選択されたメディアファイルに関連したメタデータを識別する値を格納している地域フィールドを含む。メタデータは、特定の言語に対応している。地域フィールドは、単一の記憶媒体上でメニューおよびテキストデータの多言語のサポートを提供する。例示的な地域識別子（ＬＣＩＤ）８０２の構造を図８に示す。補足説明Ａに説明する例示的な圧縮メディア形式において、TEXT.HMTおよびMENU.HMTのファイルヘッダは、それらの言語を表すＬＣＩＤ８０２を収容し、これらは一致するものでなければならない。CONTENTS.HMTは、この記憶媒体上の言語についてのＬＣＩＤのリストを収容している。

【００６４】

メディアプレーヤーは、デフォルトの言語として、ＬＣＩＤ８０２のリストの中の最初の項目を使用する。２つ以上の言語が定義されている場合、プレーヤーは、（例えば、このメディアプレーヤーの地域、または言語に基づいて）リストから再生するためにＬＣＩＤ８０２を選ぶはずである。プレーヤーは、TEXT.HMTおよびMENU.HMTの「ディレクトリ番号」を使用することによって、対応するこれらのファイルをロードする。最初の（デフォルトの）ＬＣＩＤ８０２のエントリは、CONTENTS.HMTの位置である「ディレクトリ番号」を有している。即ち、MENU.HMTおよびTEXT.HMTのデフォルト言語は、CONTENTS.HMTと同じディレクトリにある。

【００６５】

ＬＣＩＤ８０２それ自体は、いくつかの部分を有している。最初の１０ビットは主言語ＩＤ８０４であり、その言語それ自体が入っている。次の６ビットには、副言語ＩＤ８０６が入っており、同じ主言語ＩＤ８０４を共有している地域を区別するために使われるこ

とがよくある。その次の４ビットは、ソートＩＤ８０８を表し、同じ言語および地域に使われる可能性のある代替のソーティング順を区別することができる。残りの１２ビットは、将来の使用のために予約されており、常にゼロにする。

【００６６】

例えば、ＬＣＩＤ８０２は、下記に説明するCONTENTS.HMTのようなファイルに格納してもよい。

【００６７】

（CONTENTS.HMTファイルの構造）

図９は、例示的なCONTENTS.HMTファイルの構造を示している。例示的なCONTENTS.HMTファイルには、ファイルヘッダ９０２、ディレクトリテーブル９０４、再生リストファイルエントリ９０６、オーディオファイルエントリおよびオーディオファイル名テーブル９０８、メニュー画像ファイルエントリおよびメニュー画像ファイル名テーブル９１０、画像ファイルエントリおよび画像ファイル名テーブル９１２、そしてビデオファイルエントリおよびビデオファイル名テーブル９１４が含まれる。アクセラレータファイル中のオフセットは、ファイルの先頭からのバイトオフセットである。「オーディオ」９０８、「メニュー画像」９１０、「画像」９１２および「ビデオ」９１４のファイルのテーブルは、２Ｋの境界上で開始する。ディレクトリテーブル９０４は、余分なパディングなしで、ＬＣＩＤテーブルを含めて、ファイルヘッダ９０２の後に書き込まれる。ある所与のタイプの利用可能なファイルがなければ、対応するテーブルは空である。このことにより、ファイルの数およびこのテーブルに対するオフセットが共にゼロであるものとして作成される。表８は、例示的なファイルヘッダ９０２を示している。

【００６８】

【表８】

表８－ファイル目録ヘッダ

オフセット	長さ	フィールド名
0	8	識別子
8	2	バージョン
10	8	HMT生成
18	4	CONTENTS.HMTのサイズ
22	4	ディレクトリの数
26	4	ディレクトリテーブルに対するオフセット
30	4	再生リストファイルの数
34	2	再生リストエントリのサイズ
36	4	再生リストファイルテーブルに対するオフセット
40	4	オーディオファイルの数
44	2	オーディオエントリのサイズ
46	4	オーディオファイルテーブルに対するオフセット
50	4	メニュー画像ファイルの数
54	2	メニュー画像エントリのサイズ
56	4	メニュー画像ファイルテーブルに対するオフセット
60	4	画像ファイルの数
64	2	画像エントリのサイズ
66	4	画像ファイルテーブルに対するオフセット
70	4	ビデオファイルの数
74	2	ビデオエントリのサイズ
76	4	ビデオファイルテーブルに対するオフセット
80	2	ＬＣＩＤエントリの数
82	4	ディレクトリ番号１
86	4	ＬＣＩＤ１
		・・・
82+8*(n-1)	4	ディレクトリ番号ｎ
86+8*(n-1)	4	ＬＣＩＤｎ

【００６９】

「識別子」フィールドは、ＡＳＣＩＩで「INFO_HMT」を収容する８バイトのエントリである。「バージョン」フィールドは、このファイルを作成するために使用した仕様書のバージョンを表す２バイトのエントリである。例えば、バージョン１．２０は、０ｘ７８（十進法で１２０）として格納されるであろう。「HMT生成」フィールドは、このCONTENTS.HMTに合致するLSN.HMTの生成を表す８バイトのエントリである。このCONTENTS.HMTは、同じ「HMT生成」番号が入っているLSN.HMTと共にのみ使用される。「HMT生成」の値が合致しなければ、プレーヤーは、このLSN.HMTファイルを無視する。ゼロの値は、LSN.HMTファイルがないことを表す。「CONTENTS.HMTのサイズ」フィールドは、バイトでCONTENTS.HMTのサイズを収容する４バイトのエントリである。「ディレクトリの数」フィールドは、ディレクトリテーブル９０４の中のディレクトリの数を収容する４バイトのエントリである。「ディレクトリテーブルに対するオフセット」フィールドは、CONTENTS.HMTの先頭からディレクトリテーブル９０４までのバイトオフセットを表す４バイトのエントリである。「再生リストファイルの数」フィールドは、再生リストファイルテーブルの中の再生リストファイルの数を収容する４バイトのエントリである。少なくとも１つの再生リストファイルがある。「再生リストエントリのサイズ」フィールドは、バイトで「再生リストエントリ」のサイズを表す２バイトのエントリである。「再生リストファイルテーブルに対するオフセット」フィールドは、CONTENTS.HMTの先頭から再生リストファイルテーブルまでの

バイトオフセットを表す４バイトのエントリである。「オーディオファイルの数」フィールドは、オーディオファイルテーブル９０８の中のオーディオファイルの数を収容する４バイトのエントリである。「オーディオエントリのサイズ」フィールドは、バイトで「オーディオファイルエントリ」のサイズを表す２バイトのエントリである。「オーディオファイルテーブルに対するオフセット」フィールドは、CONTENTS.HMTの先頭からオーディオファイルテーブル９０８までのバイトオフセットを表す４バイトのエントリである。このオフセットは、２Ｋの倍数であり、「再生リストファイルテーブル」の終わりと「オーディオファイルテーブル」９０８の始まりの間のどのような隙間も無視される。ゼロの値は、CONTENTS.HMTの中に「オーディオエントリ」がないことを表している。

【００７０】

「メニュー画像ファイルの数」フィールドは、「メニュー画像」ファイルテーブル９１０の中の「メニュー画像」ファイルの数を収容する４バイトのエントリである。「メニュー画像エントリのサイズ」フィールドは、バイトで「メニュー画像ファイルエントリ」のサイズを表す２バイトのエントリである。「メニュー画像ファイルテーブルに対するオフセット」フィールドは、CONTENTS.HMTの先頭から「メニュー画像」ファイルテーブル９１０までのバイトオフセットを表す４バイトのエントリである。このオフセットは、２Ｋの倍数であり、「オーディオファイルテーブル」９０８の終わりと「メニュー画像ファイルテーブル」９１０の始まりの間のどのような隙間も無視される。ゼロの値は、CONTENTS.HMTの中に「メニュー画像エントリ」がないことを表す。「画像ファイルの数」フィールドは、「画像」ファイルテーブル９１２の中の「画像」ファイルの数を収容する４バイトのエントリである。「画像エントリのサイズ」フィールドは、バイトで「画像ファイルエントリ」のサイズを表す２バイトのエントリである。「画像ファイルテーブルに対するオフセット」フィールドは、CONTENTS.HMTの先頭から「画像」ファイルテーブル９１２までのバイトオフセットを表す４バイトのエントリである。このオフセットは、２Ｋの倍数であり、「メニュー画像ファイルテーブル」９１０の終わりと「画像ファイルテーブル」９１２の始まりの間のどのような隙間も無視される。ゼロの値は、CONTENTS.HMTの中に「画像エントリ」がないことを表す。「ビデオファイルの数」フィールドは、ビデオファイルテーブル９１４の中のビデオファイルの数を収容する４バイトのエントリである。「ビデオエントリのサイズ」フィールドは、バイトで「ビデオファイルエントリ」のサイズを表す２バイトのエントリである。「ビデオファイルテーブルに対するオフセット」フィールドは、CONTENTS.HMTの先頭からビデオファイルテーブル９１４までのバイトオフセットを表す４バイトのエントリである。このオフセットは、２Ｋの倍数であり、「画像ファイルテーブル」９１２の終わりと「ビデオファイルテーブル」９１４の始まりの間のどのような隙間も無視される。ゼロの値は、CONTENTS.HMTの中に「ビデオエントリ」がないことを表す。「ＬＣＩＤエントリの数」フィールドは、媒体上の言語の数を表す２バイトのエントリである。この値は、少なくとも１である。「ディレクトリ番号」フィールドは、この地域識別子のためのTEXT.HMTおよびMENU.HMTが入っているディレクトリのディレクトリテーブル９０４の中のインデックスを表す４バイトのエントリである。これは、ディレクトリ名としてＬＣＩＤをもつ\HIGHMATのサブディレクトリである。しかし、\HIMATそれ自体に対する参照は、デフォルトの言語について有効である。「ＬＣＩＤｎ」フィールドは、地域識別子を表す４バイトのエントリである。

【００７１】

本発明のデータ構造は、現在のメニューだけがメディアプレーヤーに関連したメモリ中に必要とされるように編成されている。本発明の柔軟性のあるメニューシステムは、最小構成の装置（例えば、ポータブルＣＤプレーヤー）から大型の１６：９テレビディスプレイ（例えば、ＤＶＤプレーヤーのメニュー）まで拡張性を有している。媒体は、メニューの階層を指定するが、レイアウト、サムネイル（一般消費者向け画像、ビデオのサムネイル、またはこのような所定のアート）、および背景画像は、メディアプレーヤーごとに様々である。一実施形態において、メニュー名または再生リスト名は、ユーザ入力あるいは自動生成される。メニューは、フィルタされて、プレーヤーが表示または再生することの

できるそうしたメニュー項目だけを表示する（例えば、オーディオのみのプレーヤーは、すべての画像およびビデオメニューのエントリを取り除いて、純粋にオーディオの選択のみを表示する）。

【００７２】

（例示的な動作環境）

図１０に示すように、本発明は、機能の豊富なＰＣメディアプレーヤーならびに低コストのメディアプレーヤー上での再生のために設計されたコンパイル済みのバイナリ形式でメタデータ、メニュー、および再生リストが入っている一組の小さなファイルを有する例示的な圧縮メディア形式１００２の一部として動作可能である。例示的な圧縮メディア形式の一部として使用するためのアクセラレータファイルの一例については、補足説明Ａを参照されたい。本発明の例示的な圧縮メディア形式は、様々な形式でのオーディオ、静止画像、およびビデオメディアファイル１００４を包含する。本発明と共に使用するための圧縮メディア形式は、光ディスク（例えば、ＣＤ－ＲＯＭ、ＣＲ－Ｒ、ＣＤ－ＲＷ、ＤＶＤ－ＲＡＭ、ＤＶＤ－Ｒ、ＤＶＤ－ＲＷ、ＤＶＤ＋ＲＷ、ＤＶＤ＋Ｒ、ＤＶＤ－ＲＯＭ）、フラッシュメモリ（例えば、ＣＯＭＰＡＣＴＦＬＡＳＨ（登録商標）ブランド、セキュアデジタル（ｓｅｃｕｒｅ　ｄｉｇｉｔａｌ）、ＭＥＭＯＲＹ　ＳＴＩＣＫブランド）、磁気媒体（例えば、ハードディスク）、その他同種類のものなどのコンピュータ可読媒体１００８上のファイルシステム１００６にある圧縮されたメディアファイル１００４を編成するための論理形式を定義している。

【００７３】

図１１は、コンピュータ１３０の形態の汎用コンピューティング装置の一例を示している。本発明の一実施形態において、コンピュータ１３０のようなコンピュータは、本明細書で説明され、図示されたその他の図における使用に適している。コンピュータ１３０は、１つまたは複数のプロセッサまたは処理ユニット１３２およびシステムメモリ１３４を有している。図示した実施形態において、システムバス１３６は、システムメモリ１３４からプロセッサ１３２までを含む様々なシステム構成要素を連結している。バス１３６は、メモリバスまたはメモリコントローラ、周辺バス、ＡＧＰ（Ａｃｃｅｌｅｒａｔｅｄ　Ｇｒａｐｈｉｃｓ　Ｐｏｒｔ）、および任意の種々のバスアーキテクチャを使用したプロセッサまたはローカルバスを含めて、数種のバス構造の任意の１つまたは複数を表している。限定ではなく、例として、この種のアーキテクチャには、ＩＳＡ（Ｉｎｄｕｓｔｒｙ　Ｓｔａｎｄａｒｄ　Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＭＣＡ（Ｍｉｃｒｏ　Ｃｈａｎｎｅｌ　Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＥＩＳＡ（Ｅｎｈａｎｃｅｄ　ＩＳＡ）バス、ＶＥＳＡ（Ｖｉｄｅｏ　Ｅｌｅｃｔｒｏｎｉｃｓ　Ｓｔａｎｄａｒｄｓ　Ａｓｓｏｃｉａｔｉｏｎ）ローカルバス、およびメザニン（Ｍｅｚｚａｎｉｎｅ）バスとしても知られるＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ　Ｃｏｍｐｏｎｅｎｔ　Ｉｎｔｅｒｃｏｎｎｅｃｔ）バスが含まれる。

【００７４】

コンピュータ１３０は、通例、少なくとも何らかの形態のコンピュータ可読媒体を有している。コンピュータ可読媒体は、揮発性および不揮発性の媒体、取り外し可能および取り外し不可の媒体の両方を含み、コンピュータ１３０によってアクセスすることのできる任意の利用可能な媒体であってよい。限定ではなく、例として、コンピュータ可読媒体は、コンピュータ記憶媒体および通信媒体を備える。コンピュータ記憶媒体は、コンピュータ可読命令、データ構造、プログラムモジュールまたはその他のデータのような情報の保存のための任意の方法または技術で実現される揮発性および不揮発性、取り外し可能および取り外し不可の媒体を含む。例えば、コンピュータ記憶媒体には、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリもしくは他のメモリ技術、ＣＤ－ＲＯＭ、デジタル多用途ディスク（ＤＶＤ）もしくは他の光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、またはコンピュータ１３０によってアクセスすることができ所望の情報を格納するために使うことのできる他の任意の媒体が含まれる。通信媒体は、通例、コンピュータ可読命令、データ構造、プログラムモジュール、ま

たは他のデータを搬送波や他のトランスポート機構のような変調されたデータ信号中に具現しており、任意の情報配信媒体を含む。当業者は、変調されたデータ信号に通じており、この信号は、情報をその信号中にエンコードするような様態でその特徴の１つまたは複数が設定され、または変更されている。有線ネットワークまたは直接結線接続（ｄｉｒｅｃｔ－ｗｉｒｅｄ　ｃｏｎｎｅｃｔｉｏｎ）のような有線メディア、ならびに音響、ＲＦ、赤外線、および他の無線媒体のような無線メディアは、通信媒体の例である。上記の任意の組み合わせもまた、コンピュータ可読媒体の適用範囲内に含まれる。

【００７５】

システムメモリ１３４は、取り外し可能および／または取り外し不可、揮発性および／または不揮発性メモリの形態のコンピュータ記憶媒体を含む。図示した実施形態において、システムメモリ１３４は、読み出し専用メモリ（ＲＯＭ）１３８およびランダムアクセスメモリ（ＲＡＭ）１４０を含む。起動期間などでコンピュータ１３０内の要素間で情報を転送する助けとなる基本ルーチンが入っている、基本入力／出力システム１４２（ＢＩＯＳ）は、通例、ＲＯＭ１３８に格納されている。ＲＡＭ１４０には、通例、処理ユニット１３２によって直ぐにアクセス可能であり、かつ／または現在作動中であるデータおよび／またはプログラムモジュールを収容している。限定ではなく、例示として、図１１は、オペレーティングシステム１４４、アプリケーションプログラム１４６、その他のプログラムモジュール１４８、およびプログラムデータ１５０を図示している。

【００７６】

コンピュータ１３０はまた、他の取り外し可能／取り外し不可、揮発性／不揮発性のコンピュータ記憶媒体を含んでもよい。例えば、図１１は、取り外し不可の、不揮発性の磁気媒体から読み出したり、または書き込んだりするハードディスクドライブ１５４を図示している。図１１はまた、取り外し可能の不揮発性磁気ディスク１５８から読み出したり、または書き込んだりする磁気ディスクドライブ１５６、ＣＤ－ＲＯＭまたは他の光媒体のような取り外し可能の不揮発性光ディスク１６２から読み出したり、または書き込んだりする光ディスクドライブ１６０も示している。例示的な動作環境で使うことができる他の取り外し可能／取り外し不可の揮発性／不揮発性のコンピュータ記憶媒体には、これらに限られないが、磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、固体素子ＲＡＭ、固体素子ＲＯＭ、その他同種類のものが含まれる。ハードディスクドライブ１５４、そして磁気ディスクドライブ１５６および光ディスクドライブ１６０は、インターフェイス１６６のような、不揮発性メモリインターフェイスによってシステムバス１３６に接続されるのが通例である。

【００７７】

上記で論じ、図１１に図示した装置または他の大容量記憶装置およびこれらに付属のコンピュータ記憶媒体は、コンピュータ１３０のコンピュータ可読命令、データ構造、プログラムモジュールおよび他のデータの記憶機能を提供する。例えば、図１１において、ハードディスクドライブ１５４は、オペレーティングシステム１７０、アプリケーションプログラム１７２、他のプログラムモジュール１７４、およびプログラムデータ１７６を格納しているように図示されている。これらのコンポーネントは、オペレーティングシステム１４４、アプリケーションプログラム１４６、他のプログラムモジュール１４８、およびプログラムデータ１５０と同じでも、または異なっていてもよいことに留意されたい。オペレーティングシステム１７０、アプリケーションプログラム１７２、他のプログラムモジュール１７４、およびプログラムデータ１７６は、ここでは異なる番号が付与され、少なくとも、これらが異なるコピーであることを示している。

【００７８】

ユーザは、コマンドおよび情報をコンピュータ１３０の中に、キーボード１８０およびポインティング装置１８２（例えば、マウス、トラックボール、ペン、またはタッチパッド）のような入力装置またはユーザインターフェイス選択装置を通して入力する。他の入力装置（図示せず）には、マイクロフォン、ジョイスティック、ゲームパッド、衛星受信アンテナ、スキャナ、またはその種の他のものが含まれてもよい。これらおよびその他の

入力装置は、システムバス１３６に連結されているユーザ入力インターフェイス１８４を通じて処理ユニット１３２に連結されているが、パラレルポート、ゲームポート、またはＵＳＢ（Ｕｎｉｖｅｒｓａｌ　Ｓｅｒｉａｌ　Ｂｕｓ）のような、他のインターフェイスおよびバス構造によって接続されてもよい。モニタ１８８または他種のディスプレイ装置もまた、ビデオインターフェイス１９０のような、インターフェイスを介してシステムバス１３６に接続される。モニタ１８８に加えて、コンピュータは、プリンタおよびスピーカのような他の周辺出力装置（図示せず）を含む場合が多く、これらは出力周辺インターフェイス（図示せず）を通して接続することができる。

【００７９】

コンピュータ１３０は、リモートコンピュータ１９４のような、１つまたは複数のリモートコンピュータへの論理接続を用いて、ネットワーク化された環境で動作することもできる。リモートコンピュータ１９４は、パーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピア装置または他の共通ネットワークノードであってよく、コンピュータ１３０に関連して上述した要素の多くまたはすべてを含むのが通例である。図１１に描かれた論理接続には、ＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）１９６およびＷＡＮ（Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）１９８が含まれるが、他のネットワークを含んでもよい。このようなネットワーク環境は、オフィス、企業全体のコンピュータネットワーク、イントラネット、および地球規模のコンピュータネットワーク（例えば、インターネット）において一般的である。

【００８０】

ローカルエリアネットワーキング環境で使われる場合、コンピュータ１３０は、ネットワークインターフェイスまたはアダプタ１８６を通してＬＡＮ１９６に接続される。ワイドエリアネットワーキング環境で使われる場合、コンピュータ１３０は、インターネットのようなＷＡＮ１９８を介して通信を確立するためのモデム１７８または他の手段を含むのが通例である。モデム１７８は、内蔵でも外付けでもよく、ユーザ入力インターフェイス１８４、またはその他の適切な機構を介してシステムバス１３６に接続される。ネットワーク化された環境において、コンピュータ１３０に関連して描かれたプログラムモジュール、またはその一部は、リモートメモリ記憶装置（図示せず）に格納してもよい。限定ではなく、例として、図１１は、リモートアプリケーションプログラム１９２をこのメモリ装置上に常駐しているように図示している。示されたネットワーク接続は、例示であり、これらのコンピュータ間の通信リンクを確立するその他の手段を使うことができるということが理解されるであろう。

【００８１】

一般的に、コンピュータ１３０のデータプロセッサは、コンピュータの様々なコンピュータ可読記憶媒体にいろいろな時間に格納された命令によってプログラムされている。プログラムおよびオペレーティングシステムは、通例、例えば、フロッピー（登録商標）ディスクまたはＣＤ－ＲＯＭ上に分散されている。そこから、それらはコンピュータの二次メモリにインストール、またはロードされる。実行時に、これらはコンピュータの一次電子メモリに少なくとも部分的にロードされる。これらおよび他の様々な種類のコンピュータ可読記憶媒体がマイクロプロセッサまたはその他のデータプロセッサに関連して下述されるステップを実施するための命令またはプログラムを収容する場合、本明細書に説明した本発明には、このような媒体が含まれる。本発明にはまた、本明細書で説明した方法および技術に従ってプログラムされた場合、コンピュータそれ自体も含まれる。

【００８２】

説明のために、オペレーティングシステムのような、プログラムおよび他の実行可能プログラムコンポーネントを個別ブロックとして本明細書に図示している。しかし、このようなプログラムおよびコンポーネントは、コンピュータのいろいろな記憶装置コンポーネントに様々な時間に常駐し、コンピュータのデータプロセッサによって実行されるということが認められる。

【００８３】

コンピュータ１３０を含めて、例示的なコンピューティングシステム環境に関連して説明したが、本発明は、数多くの他の汎用または特定用途のコンピューティングシステム環境または構成で動作できる。このコンピューティングシステム環境は、本発明の使用の範囲または機能の適用範囲についてどのような限定も示唆することを目的としていない。さらに、このコンピューティングシステム環境は、例示的な動作環境において図示した構成要素のどのような１つまたは組み合わせに関連するいかなる依存性または要求条件を有するものとして解釈されるべきではない。本発明と共に使用するのに適する可能性のあるよく知られたコンピューティングシステム、環境、および／または構成の例には、これらに限られないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドまたはラップトップ装置、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラム可能な民生用電子機器、ネットワークＰＣ、ミニコンピュータ、メインフレームコンピュータ、任意の上記システムまたは装置を含む分散コンピューティング環境、その他同種類のものが含まれる。

【００８４】

本発明は、１つまたは複数のコンピュータまたは他の装置によって実行される、プログラムモジュールのような、コンピュータ実行可能命令の一般的なコンテキストで説明することができる。一般的に、プログラムモジュールには、これらに限定されないが、特定のタスクを実行したり、特定の抽象データタイプを実装したりするルーチン、プログラム、オブジェクト、コンポーネント、およびデータ構造が含まれる。本発明はまた、通信ネットワークを通してリンクされるリモート処理装置によってタスクが実行される分散コンピューティング環境において実施されてもよい。分散コンピューティング環境においては、プログラムモジュールを、メモリ記憶装置を含むローカルおよびリモートのコンピュータ記憶媒体の両方に配置してもよい。

【００８５】

動作において、コンピュータ１３０は、図２に示したもののようなコンピュータ実行可能命令を実行して、適応型のメニュー構造においてメディアファイルを編成する。さらに、メディアプレーヤーのコンピュータ１３０は、図５に示したもののようなコンピュータ実行可能命令を実行して、メディアプレーヤーの機能に従ってメニュー構造をフィルタする。

【００８６】

本発明の要素またはその実施形態を紹介する際に、冠詞「ある」、「この」、「その」および「前記」は、１つまたは複数の要素があることを意味することが意図されている。用語「備える」、「収容する」、「含む」および「有する」は、包含的であることが意図され、これら記載されたものの他に付加的な要素がある可能性があることを意味している。

【００８７】

上記に鑑みて、本発明のいくつかの目的が達成され、その他の有利な結果も得られることが理解されるであろう。

【００８８】

本発明の適用範囲から逸脱することなく上記の構成、製品、および方法において、様々な変更を為すことができるので、上記の説明に含まれ、添付の図面において示されたすべての事項は、説明として解釈されるべきであって、限定の意味で解釈されるべきではない。

【００８９】

（補足説明Ａ）

図１２は、本発明のソフトウェアと共に使用するための例示的な圧縮メディア形式におけるファイルシステムを示している。アクセラレータファイルは、媒体上の「HIGHMAT」と呼ばれる最上位ディレクトリの下にあり、次のファイル、CONTENTS.HMT、nnnnnnnn.HMT、MENU.HMT、およびTEXT.HMTを含む。CONTENTS.HMTファイルは、この媒体上にあるすべてのメディアファイルについての情報を収容している。それには、ディレクトリテーブルが

入っており、次にそれぞれのサポートされたファイルタイプ（「オーディオ」、「ビデオ」、「画像」、および「再生リスト」）のファイルエントリテーブルが続く。媒体上の各再生リストにつき１つの「nnnnnnnn.HMT」と呼ばれるファイルがあり、ここでnnnnnnnnは、１６進数の再生リストファイルの識別子である。これらの再生リストファイルは、「PLAYLIST」サブディレクトリに作成される。MENU.HMTファイルは、メニュー構造を収容している。TEXT.HMTファイルは、再生期間中に必要とされるすべてのテキスト形式の情報を収容している。

【００９０】

他の実施形態において、次のファイル、<Thumbnail>.HMTおよび<Background>.HMTが含まれている。媒体上の各サムネイルにつき１つのファイルがある（例えば、.HMT拡張子を有する１６０×１２０ＪＰＧ形式）。オーサリングソフトウェアは、これらのファイルにしかるべき名称を付してもよい。「サムネイル」ファイルは、「IMAGES」サブディレクトリに作成される。媒体上の各メニューの背景につき１つのファイルがある。４：３のアスペクト比の背景は、.HMT拡張子を有する６４０×４８０のＪＰＧ形式のである。１６：９のアスペクト比の背景は、.HMT拡張子を有する８５２×４８０のＪＰＧ形式である。オーサリングソフトウェアは、これらのファイルにしかるべき名称を付してもよい。「背景」ファイルは、「IMAGES」サブディレクトリに作成される。

【００９１】

HIGHMATディレクトリおよびその中のすべてのファイルは、ユーザの混乱を少なくするために媒体を作成するソフトウェアによって、隠されたものとしてマークしてもよい。プレーヤーは、媒体をこれらのファイルおよびディレクトリが隠されていてもいなくても取り扱えるべきである。バイナリ構造のすべては、「リトルエンディアン」のバイト順を用いてエンコードされる。「ファイル」および「ディレクトリ」名は、「ビッグエンディアン」のバイト順を使う。プレーヤーが未知の値に出くわしたときは、デフォルトの様式では、この問題となるファイルを無視することである。例えば、「再生リストの要約タイプ」が２５５に等しければ、再生リスト全体をスキップする。

【００９２】

TEXT.HMT中のフィールドへのメタデータマッピングは、次の表で概要を説明する。

【００９３】

【表９】

表Ａ１

	オーディオファイル	画像ファイル	ビデオファイル	再生リストファイル
テキスト１	オーディオのタイトル	画像のタイトル	ビデオのタイトル	再生リストのタイトル
テキスト２	アーティスト名	作成装置名（ＤＳＣ名）	アーティスト名	なし
テキスト３	作曲家名	画像撮影日	作成日	なし
テキスト４	アルバム名	アルバム名	アルバム名	なし
テキスト５	ジャンル名	ジャンル名	ジャンル名	ジャンル名
グループ	なし	なし	なし	グループ名
追加テキスト歌詞	歌詞	注釈（コメント）	注釈（コメント）	なし
著作権	著作権	著作権	著作権	なし
アルバムのアーティスト	アルバムのアーティスト	なし	なし	なし

【００９４】

TEXT.HMTの構造を表Ａ２に示す。

【００９５】
【表１０】

表Ａ２

ファイルヘッダ
コンテンツテキストエントリテーブル
グループテキストエントリテーブル
追加テキストエントリテーブル
コンテンツテキストデータ
グループ名テキストデータ
追加テキストデータ

【００９６】
このファイルを書き込むには、ファイル中のデータ構造間に隙間がないようにする。
【００９７】
【表１１】

表Ａ３－テキストファイル情報形式

オフセット	長さ	フィールド名
0	8	識別子
8	2	バージョン
10	4	ファイルサイズ
14	4	コンテンツテキストエントリの数
18	4	グループテキストエントリの数
22	4	追加テキストエントリの数
26	4	コンテンツテキストエントリのオフセット
30	4	グループテキストエントリのオフセット
34	4	追加テキストエントリのオフセット
38	4	ＬＣＩＤ
42	2	ディスク名の長さ
44	可変	ディスク名

【００９８】
（識別子）
このフィールドは、８バイトのＡＳＣＩＩで「TEXT_HMT」であるべきである。
【００９９】
（バージョン）
この２バイトのエントリは、このファイルの作成に使われたＨＭＴ仕様のバージョンを表す。例えば、バージョン１．２０は、０ｘ７８（十進法で１２０）として格納されるであろう。
【０１００】
（ファイルサイズ）
このフィールドは、このTEXT.HMTファイルのサイズを４バイトで収容する。
【０１０１】
（コンテンツエントリの数）
この４バイトのエントリは、「コンテンツテキストエントリ」の数を定義する。
【０１０２】
（グループエントリの数）
この４バイトのエントリは、「グループテキストエントリ」の数を定義する。
【０１０３】
（追加テキストエントリの数）

この４バイトのエントリは、「追加テキストエントリ」の数を定義する。

【０１０４】

（コンテンツテキストエントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「コンテンツ」テキストエントリテーブルまでのオフセットである。

【０１０５】

（グループテキストエントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「グループ」テキストエントリテーブルまでのオフセットである。

【０１０６】

（追加テキストエントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「追加」テキストエントリテーブルまでのオフセットである。

【０１０７】

（ＬＣＩＤ）

この４バイトのエントリは、MENU.HMTファイルの「言語」ＩＤである。

【０１０８】

（ディスク名の長さ）

この２バイトのエントリは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、ディスク名のバイト長を収容する。

【０１０９】

（ディスク名）

この可変長のエントリは、ディスクの名称を表す。「ディスク名」は、ＵＣＳ－２であり、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了する。このエントリの最大長は、ヌルのＵＣＳ－２の１文字を含めて、ＵＣＳ－２の６５文字である。

【０１１０】

（テキストエントリ）

テキスト項目のすべては、次の形式を用いて格納される。

【０１１１】

【表１２】

表Ａ４－テキストエントリ

オフセット	長さ	フィールド名
0	2	長さ
2	可変	テキスト

【０１１２】

（長さ）

この２バイトのエントリは、末尾のヌルのＵＣＳ－２の１文字（ヌルの２バイト）を除いて、テキストのバイト長を収容する。

【０１１３】

（テキスト）

これは、ＵＳＣ－２「テキストエントリ」である。このテキストは、ヌルのＵＣＳ－２の１文字（ヌルの２バイト）で終了する。テキストエントリの最大長を、次の表に示す。

【０１１４】

【表１３】

表Ａ５－テキストの長さ

テキストデータのタイプ	最大長
グループ名	65
追加テキストエントリ	32,767
その他すべて	1,024

【０１１５】

この長さは、ＵＣＳ－２文字で定義されており、末尾の「ヌル」のＵＣＳ－２の１文字を含む。

【０１１６】

「コンテンツ」テキストエントリ「テーブル」は、「コンテンツテキストエントリ」（表Ａ６参照）のリストを含む。CONTENTS.HMT中の各ファイル（「再生リスト」、「オーディオ」、「メニュー画像」、「画像」および「ビデオ」）につき１つの「コンテンツテキストエントリ」があり、これらのファイルがCONTENTS.HMTにおいてリストされているのと同じ順番になっている。

【０１１７】

【表１４】

表Ａ６－コンテンツテキストエントリ

オフセット	長さ	フィールド名
0	4	コンテンツＩＤ
4	4	テキスト１エントリのオフセット
8	4	テキスト２エントリのオフセット
12	4	テキスト３エントリのオフセット
16	4	テキスト４エントリのオフセット
20	4	テキスト５エントリのオフセット
24	4	追加テキストエントリのオフセット

【０１１８】

（コンテンツＩＤ）

この４バイトのエントリは、この「コンテンツ」テキストエントリの「コンテンツ」IDを表す。

【０１１９】

（テキスト１エントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「テキスト１」エントリの最初のバイトに対するオフセットを表す。CONTENTS.HMT中の各コンテンツ項目は、「テキスト１」エントリを有していることが要求される。

【０１２０】

（テキスト２エントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「テキスト２」エントリの最初のバイトに対するオフセットを表す。コンテンツ項目が「テキスト２」データを有していなければ、このエントリは、０である。

【０１２１】

（テキスト３エントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「テキスト３」エントリの最初のバイトに対するオフセットを表す。コンテンツ項目が「テキスト３」データを有していなければ、このエントリは、０である。

【０１２２】

（テキスト４エントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「テキスト４」エントリの最初のバイトに対するオフセットを表す。コンテンツ項目が「テキスト４」データを有していなければ、このエントリは、０である。

【０１２３】

（テキスト５エントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「テキスト５」エントリの最初のバイトに対するオフセットを表す。コンテンツ項目が「テキスト５」データを有していなければ、このエントリは、０である。

【０１２４】

（追加テキストエントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「追加」テキストエントリの最初のバイトに対するオフセットを表す。コンテンツ項目が「追加」テキストデータを有していなければ、このエントリは、０である。

【０１２５】

「グループ」テキストエントリ「テーブル」は、「グループテキストエントリ」（表Ａ７）のリストを含む。CONTENTS.HMTにリストされた再生リストファイル中の各グループにつき１つの「グループテキストエントリ」がある。これらのエントリは、昇順のグループ番号順である。

【０１２６】

【表１５】

表Ａ７－グループテキストエントリ形式

オフセット	長さ	フィールド名
0	4	グループ名
4	4	グループ名テキストエントリのオフセット

【０１２７】

（グループ番号）

この４バイトのエントリは、この「グループ」テキストエントリの「グループ」番号を表す。

【０１２８】

（グループ名テキストエントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から「グループ」名テキストエントリの最初のバイトに対するオフセットを表す。各グループは、名称を有していることが要求される。

【０１２９】

追加テキストエントリテーブルは、表Ａ８に定義するようにゼロまたはそれ以上の「追加テキストエントリ」として定義される。これらのエントリは、昇順の「コンテンツ」ＩＤ順に格納される。「追加テキストエントリ」がゼロの場合、「追加テキストエントリの数」フィールド、「追加テキストエントリ」の「ベース」オフセット、「追加テキストデータ」の「ベース」オフセットはすべて、０である。

【０１３０】

【表１６】

表Ａ８－追加テキストエントリ

オフセット	長さ	フィールド名
0	4	コンテンツＩＤ
4	1	追加テキストデータの数
5	1	予約
6	2	１番目のテキストデータのタイプ
8	4	１番目のテキストエントリのオフセット
		・・・
6+6*(n-1)	2	ｎ番目のテキストデータのタイプ
8+6*(n-1)	4	ｎ番目のテキストエントリのオフセット

【０１３１】

（コンテンツＩＤ）

この４バイトのエントリは、CONTENTS.HMT中のこの「追加」テキストエントリに関連したファイルの「コンテンツ」ＩＤを表す。

【０１３２】

（追加テキストデータの数）

この１バイトのエントリは、この「追加」テキストエントリによって関連付けられる「追加」テキストデータの数を表す。

【０１３３】

（予約）

この１バイトのエントリは、将来の使用のために予約されている。

【０１３４】

（ｎ番目のテキストデータのタイプ）

この２バイトのエントリは、下記の表で定義される値の１つを収容する。

【０１３５】

【表１７】

表Ａ９－ｎ番目のテキストデータのタイプ

追加テキストタイプ	値
0	UNUSED
1	LYRICS
2	COPYRIGHT
3	ALBUM ARTIST
4 - 65,535	RESERVED

【０１３６】

（ｎ番目のテキストエントリのオフセット）

この４バイトのエントリは、TEXT.HMTの先頭から各「追加」テキストエントリの最初のバイトに対するオフセットを表す。

【０１３７】

対応する参照文字は、図面全体を通して対応する部位を示す。

【図面の簡単な説明】

【０１３８】

【図１】本発明を実施するのに適したメディア環境の一例を示すブロック図である。

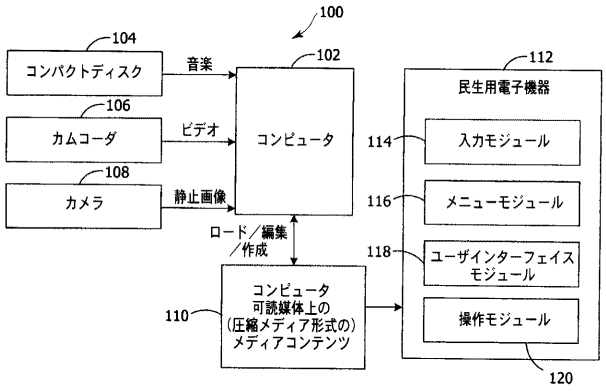
【図２】本発明によるオーサリングソフトウェアの例示的な動作を示すフローチャートである。

【図３】階層メニュー構造にメディアタイプの情報を含めることを示す階層メニュー構造

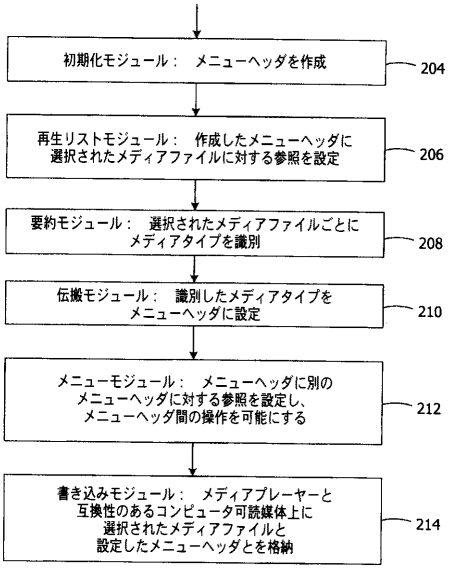
の例示的なブロック図である。
【図４】階層メニュー構造の一具体例を示す例示的なブロック図である。
【図５】本発明によるフィルタリングソフトウェアの例示的な動作を示すフローチャートである。
【図６】レンダリングに利用可能なメディアタイプを示す本発明のフィルタリングソフトウェアの例示的なユーザインターフェイスのスクリーンショットである。
【図７】１つまたは複数のメニューヘッダを有するメニュー構造を示す例示的なブロック図である。
【図８】地域情報を格納する構造を示す例示的なブロック図である。
【図９】コンピュータ可読媒体上のすべてのメディアファイルに付随したデータを格納するデータ構造の例示的なブロック図である。
【図１０】本発明のソフトウェアと共に使用するための例示的な圧縮メディア形式を示すブロック図である。
【図１１】本発明を実施するのに適したコンピューティングシステム環境の一例を示すブロック図である。
【図１２】本発明のソフトウェアと共に使用するための例示的な圧縮メディア形式におけるファイルシステムを示すブロック図である。
【符号の説明】
【０１３９】
１００　システム
１０２　コンピュータ
１０４　コンパクトディスク
１０６　カムコーダ
１０８　カメラ
１１０　コンピュータ可読媒体上の（圧縮メディア形式の）メディアコンテンツ
１１２　民生用電子機器
１１４　入力モジュール
１１６　メニューモジュール
１１８　ユーザインターフェイスモジュール
１２０　操作モジュール
３０２、３０４　経路
６０２　ユーザインターフェイス
６０４　ビデオファイル
６０６　オーディオファイル
６０８　画像ファイル
６１０　プレゼンテーション
７００　メニュー構造
７０２　MENU.HMTファイルヘッダ
７０４　メニュー＃１、メニュー＃２
７０６　メニューヘッダ＃１、メニューヘッダ＃２
７０８　再生リスト項目
７１０　メニュー項目
１３０　コンピュータ
１３２　処理ユニット
１３４　システムメモリ
１３６　システムバス
１３８　ＲＯＭ
１４０　ＲＡＭ
１４２　ＢＩＯＳ
１４４　オペレーティングシステム

１４６　アプリケーションプログラム
１４８　他のプログラムモジュール
１５０　プログラムデータ
１５４　ハードディスクドライブ
１５６　磁気ディスクドライブ
１５８　取り外し可能の不揮発性磁気ディスク
１６０　光ディスクドライブ
１６２　取り外し可能の不揮発性光ディスク
１６６　不揮発性メモリインターフェイス
１７０　オペレーティングシステム
１７２　アプリケーションプログラム
１７４　他のプログラムモジュール
１７６　プログラムデータ
１７８　モデム
１８０　キーボード
１８２　ポインティング装置
１８４　ユーザ入力インターフェイス
１８６　ネットワークインターフェイス
１８８　モニタ
１９０　ビデオインターフェイス
１９２　リモートアプリケーションプログラム
１９４　リモートコンピュータ
１９６　ローカルエリアネットワーク
１９８　ワイドエリアネットワーク

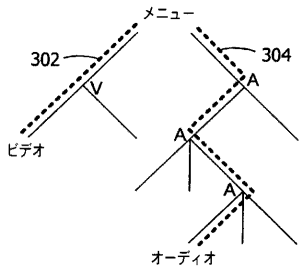
【図１】



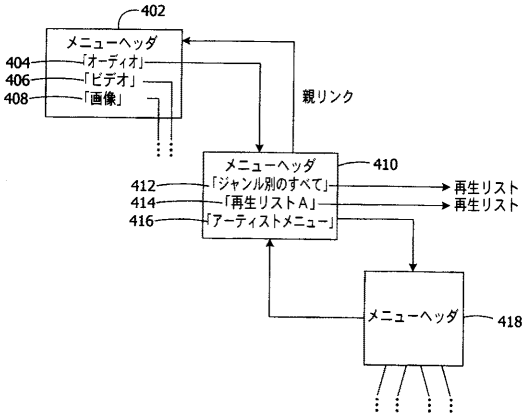
【図２】



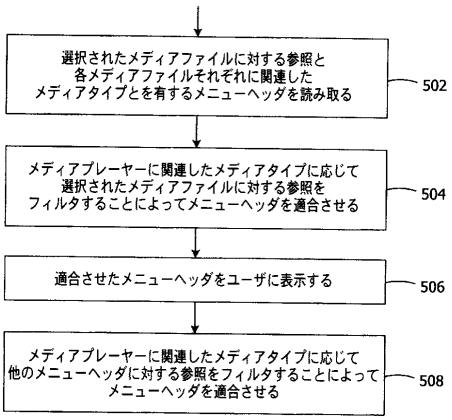
【図３】



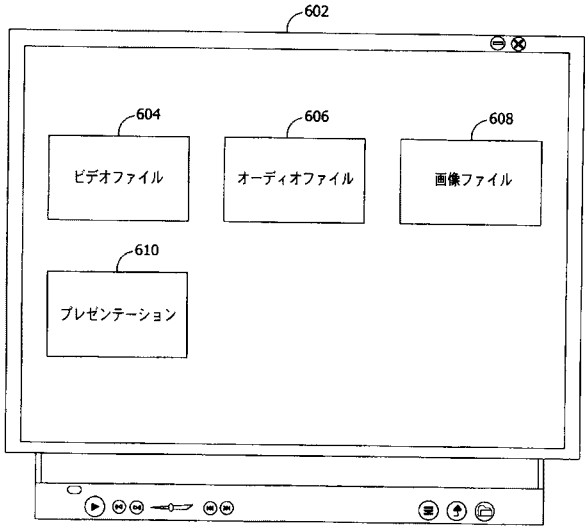
【図４】



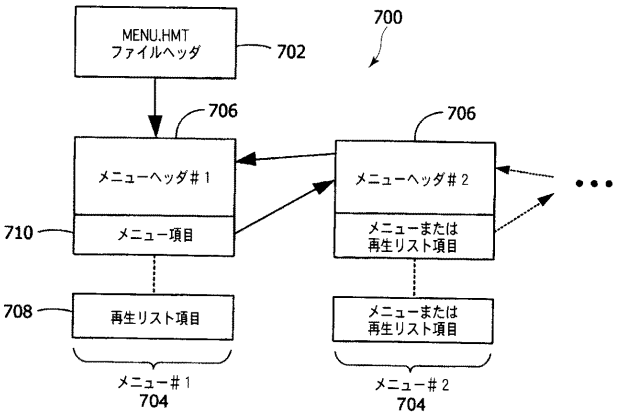
【図５】



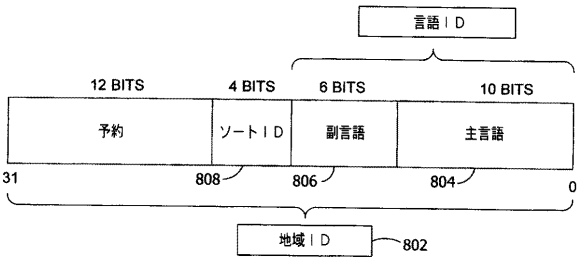
【図６】



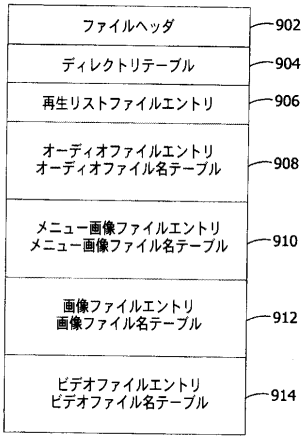
【図７】



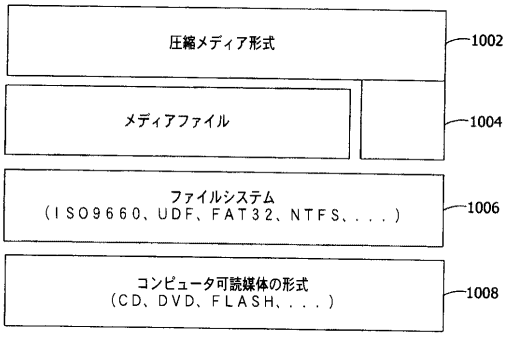
【図８】



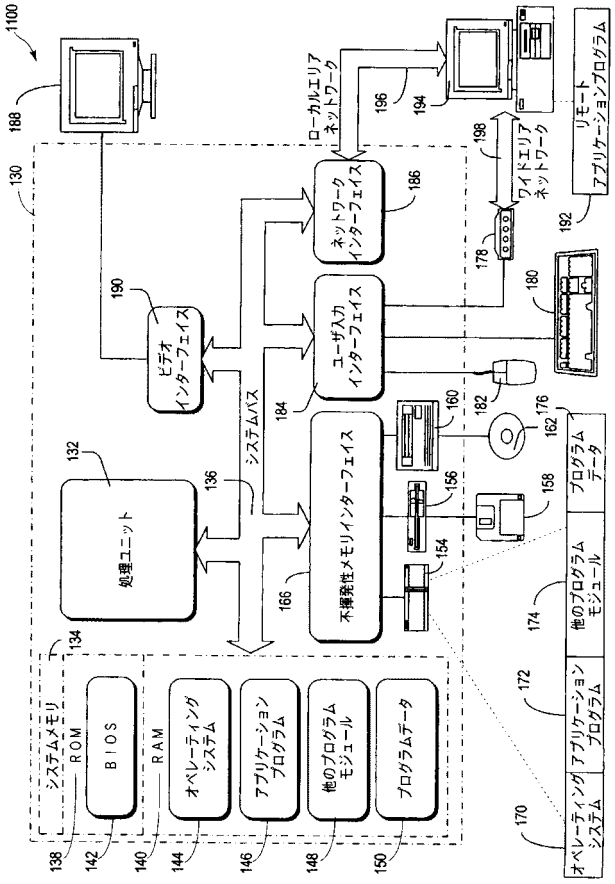
【図９】



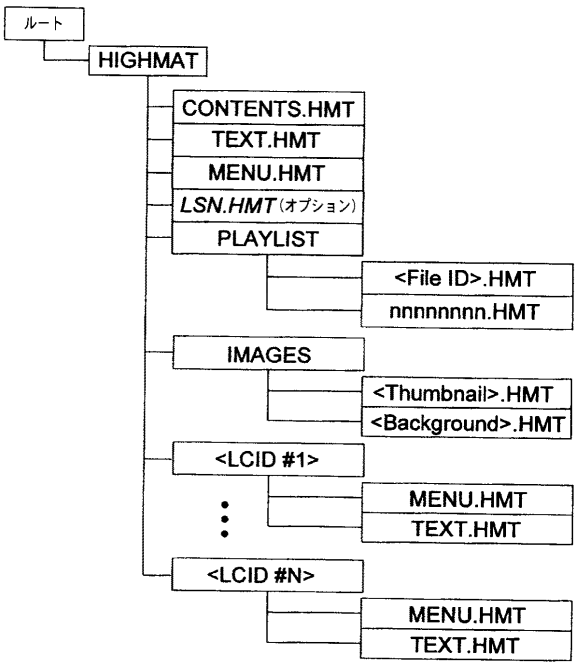
【図１０】



【図１１】



【図１２】



フロントページの続き

(72)発明者　ケビン　レイ　ラチャペル
アメリカ合衆国　９８０５３　ワシントン州　レッドモンド　２０４　ドライブ　ノースイースト　６２０３

Ｆターム(参考) 5B069 AA01 BA01 CA14 CA18
5B082 EA07 GC04
5C053 FA07 FA14 FA24 FA27 FA30 HA29 LA11 LA14

【外国語明細書】

ADAPTIVE MENU SYSTEM FOR MEDIA PLAYERS

TECHNICAL FIELD

The present invention relates to the field of digital media content. In particular, this invention relates to creating and reading an adaptive menu system for use with media players.

BACKGROUND OF THE INVENTION

Due to recent advances in technology, computer users are now able to enjoy many features that provide an improved user experience, such as playing various media and multimedia content on their personal or laptop computers. For example, most computers today are able to play compact discs (CDs) so users can listen to their favorite musical artists while working on their computers. Many computers are also equipped with digital versatile disc (DVD) drives enabling users to watch movies.

Consumer electronic devices such as portable CD players, DVD players and car receivers have widely varying capabilities in terms of media playback and user interface capabilities. Some devices are capable of displaying images and video data while some can only play back audio data. Some devices such as DVD players have graphical displays and can display complex menus with background images and thumbnails while others such as portable CD players have only simple buttons like Play, Next, Previous, and Stop with a single line liquid crystal display (LCD) with no graphical capabilities. As such, when a user stores compressed media files such as Moving Picture Experts Group audio layer-3 (MP3) files, WINDOWS MEDIA technologies audio (WMA) files, or Joint Photographic Experts Group (JPEG) files on a computer-readable medium, the user often does not know all the playback device types that will be employed to render

the stored compressed media files. For example, the user may purchase a new playback device at a later time with different capabilities from an existing media player and want to play back a previously-created computer-readable medium.

For these reasons, a system for an adaptive menu structure is desired to address one or more of these and other disadvantages.

SUMMARY OF THE INVENTION

The invention includes software and data structures for creating an adaptive menu structure associated with one or more media files. In particular, the invention includes an adaptive menu system whereby menu items are selectively hidden depending on the capabilities of a media player. The menu system of the invention can be a rich graphical display with thumbnail images on a television screen or can be represented as a simple one-option-at-a-time menu on a single line LCD display. In one embodiment, the invention is operable as part of a compressed media format having a set of small files containing metadata, menus, and playlists in a compiled binary format designed for playback on feature-rich personal computer media players as well as low cost media players. The format encompasses audio, still images, and video in various formats.

In accordance with one aspect of the invention, a method organizes one or more media files to enable a user to navigate the media files using a media player. The media files each have a media type associated therewith and are adapted for rendering by the media player as a function of the media type. The media player also has a media type associated therewith. The method includes creating a menu header. The method also includes populating the created menu header with references to one or more selected media files. The method also includes identifying a media type associated with each of the selected media files and populating the menu header with the identified media types. A media player uses the populated menu header to filter the selected media files as a function of a media type associated with the media player.

In accordance with another aspect of the invention, one or more computer-readable media in a media authoring tool have computer-executable components for organizing one or more media files to enable a user to navigate the media files using a media player. The media files each have a media type associated therewith and are

adapted for rendering by the media player as a function of the media type. The media player also has a media type associated therewith. The components include an initialization module for creating a menu header. The components also include a playlist module for populating the created menu header from the initialization module with references to selected media files. The components also include a summary module for identifying a media type associated with each of the selected media files from the playlist module. The components also include a propagation module for populating the menu header with the identified media types from the summary module. A media player uses the populated menu header to filter the selected media files as a function of a media type associated with the media player.

In accordance with yet another aspect of the invention, a method displays menu information on a display associated with a media player. The menu information enables a user to navigate one or more media files using the media player. The media files each have a media type associated therewith and are adapted for rendering by the media player as a function of the media type. The media player also has a media type associated therewith. The method reads a menu header having references to selected media files and a media type associated with each respective one of the selected media files. The method also adapts the read menu header by filtering the references to the selected media files as a function of a media type associated with a media player. The method also displays menu information from the adapted menu header to a user on a display associated with the media player.

In accordance with still another aspect of the invention, a media player displays menu information on a display associated with the media player. The menu information enables a user to navigate one or more media files using the media player. The media files each have a media type associated therewith and are adapted for rendering by the media player as a function of the media type. The media player also has a media type associated therewith. The media player includes an input module for reading a menu header having references to one or more selected media files and a media type associated with each respective one of the selected media files. The media player also includes a menu module for adapting the read menu header from the input module by filtering the references to the selected media files as a function of a media type associated with a

media player. The media player also includes a user interface module for displaying menu information from the adapted menu header from the menu module to a user on a display associated with the media player.

In accordance with yet another aspect of the invention, a computer-readable medium has stored thereon a data structure representing an organization of one or more media files. The media files each have a media type associated therewith and are adapted for rendering by a media player as a function of the media type. The media player has a media type associated therewith. The data structure enables a user to navigate the media files using the media player. The data structure includes a playlist field storing a reference to one or more selected media files. The selected media files each have a media type associated therewith. The data structure also includes a summary field storing the media type for one of the selected media files referenced in the playlist field. A media player uses the data structure to filter the selected media files as a function of a media type associated with the media player.

Alternatively, the invention may comprise various other methods and apparatuses.

Other features will be in part apparent and in part pointed out hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating one example of a suitable media environment in which the invention may be implemented.

FIG. 2 is a flow chart illustrating exemplary operation of authoring software according to the invention.

FIG. 3 is an exemplary block diagram of a hierarchical menu structure illustrating the inclusion of media type information in the structure.

FIG. 4 is an exemplary block diagram illustrating a specific example of a hierarchical menu structure.

FIG. 5 is a flow chart illustrating exemplary operation of filtering software according to the invention.

FIG. 6 is a screen shot of an exemplary user interface of filtering software of the invention illustrating the media types available for rendering.

FIG. 7 is an exemplary block diagram illustrating a menu structure having one or more menu headers.

FIG. 8 is an exemplary block diagram illustrating a structure storing locale information.

FIG. 9 is an exemplary block diagram of a data structure storing data associated with all the media files on a computer-readable medium.

FIG. 10 is a block diagram illustrating an exemplary compressed media format for use with the invention software.

FIG. 11 is a block diagram illustrating one example of a suitable computing system environment in which the invention may be implemented.

FIG. A1 is a block diagram illustrating a file system in an exemplary compressed media format for use with the invention software.

Corresponding reference characters indicate corresponding parts throughout the drawings.

DETAILED DESCRIPTION OF THE INVENTION

The invention includes authoring software to create a menu structure to enable a user to navigate one or more media files using a media player. The invention also includes filtering software to adapt the created menu structure to display menu information on a display associated with the media player as a function of a media type associated with the media player. In one embodiment, the invention software is operable as part of a compressed media format (see FIG. 10) having a set of small files containing metadata, menus, and playlists in a compiled binary format designed for playback on feature-rich personal computer media players as well as low cost media players. For example, a typical low-end portable CD player might have as little as one hundred kilobytes of working memory, an eight-bit central processing unit running at only one megahertz and a very slow CD-drive mechanism with a maximum seek time of nearly five seconds.

Media Environment

Referring now to the drawings, FIG. 1 illustrates an exemplary multimedia environment in which the invention can be used. A system 100 has one or more computers 102 coupled to one or more devices providing media content. For example, the devices may include a CD 104, a camcorder 106, or a camera 108.

In one embodiment, the computer 102 stores media content on a computer-readable medium 110 for use by a media player program associated with a consumer electronic device 112. The consumer electronic device 112 includes any suitable rendering filter or media player or device that is configured to render digital media so that the user can experience the content that is embodied on the medium 110. For example, suitable media player applications include a CD media player and a DVD media player.

In the invention, the media players, consumer electronic devices 112, or the like may be organized into three exemplary levels according to the capabilities of the media player. Each media player has a media type that identifies the type of media that the media player is capable of rendering. For example, the media type (also referred to as a playlist summary type, a menu summary type, or the like) includes one or more of the following: audio, video, and still image. Level 1 media players include audio-only players such as portable CD players, car receivers, and DVD players. In addition to Level 1 media players, level 2 media players include audio and still image capable players including portable and set-top DVD players optionally capable of rendering images with audio simultaneously. Level 3 media players include level 2 media players and audio, still image, and video capable players. Exemplary consumer electronic devices 112 include, but are not limited to, the following: a portable CD player, an MP3 player, an audio system in an automobile, a personal digital assistant, a cellular telephone, or the like.

The software and data structures of the invention are operable in systems including, but not limited to, media players and media authoring systems. A media player according to the invention includes software displaying menu information on a display associated with the media player. An exemplary consumer electronic device 112 or media player includes an input module 114, a menu module 116, a user interface

module 118, and a navigation module 120. The input module 114 reads a menu header having references to selected media files and a media type associated with each respective one of the selected media files. For example, the media files and menu header are stored on a computer-readable medium. The media files are adapted for rendering by the media player as a function of the media type. The menu module 116 adapts the read menu header by filtering the references to the selected media files as a function of a media type associated with the media player. The user interface module 118 displays menu information from the adapted menu header to the user on a display associated with the media player. In one embodiment, the menu header includes a reference to at least one other menu header and a media type associated with the other menu header to enable navigation between the menu headers. The navigation module 120 adapts the read menu header by filtering the reference to the other menu header as a function of the media type associated with the media player.

Playlists

A playlist is a convenient way to organize groups of audio, video, and image files on a computer-readable medium. The playlist may include, but is not limited to, one or more of the following: a media file, a group of audio files, a group of video files, a group of timed image sequences, and a group of various complex parallel combinations of images with audio or video. For example, a user may create playlists for different performers or different kinds of music or videos. The user also can manipulate the created playlists by shuffling or repeating the playlists. Playlists allow the user to easily view a listing of media files to sort, search, and quickly navigate.

Authoring Software

The invention software is operable across a range of players and media types. For instance, a computer-readable medium containing audio and photographs offers audio playback capabilities in an audio-only portable CD player. To support this capability each menu item within an accelerator file is tagged with the types of content that it includes (e.g., audio, still images, video). During creation of the medium, these tags are propagated up through the menu tree so that at any level, every item is tagged to

show all the kinds of media that it contains. Using this tag, the player can filter the menus according to its capabilities and display only the menu items and playlists that it can render.

Referring next to FIG. 2, a flow chart illustrates exemplary operation of authoring software according to the invention. In the exemplary embodiment of FIG. 2, the authoring software includes computer-executable components for organizing one or more media files to enable the user to navigate the files using the media player. The components include an initialization module, a playlist module, a summary module, a propagation module, a menu module, and a writer module. The initialization module creates a menu header at 204. The playlist module populates the created menu header at 206 with references to selected media files encapsulated as playlists. The summary module identifies the media type associated with each of the selected media files within each of the playlists at 208. The propagation module populates the menu header at 210 with the identified media types. A media player uses the populated menu header to filter the selected media files as a function of a media type associated with the media player. The menu module populates the menu header at 212 with references to other menu header (e.g., at least one) to enable navigation between the menu headers. The writer module stores the selected media files and the populated menu header(s) at 214 on a computer-readable medium compatible with the media player. The references populating the menu header may include, but are not limited to, a memory location offset to the selected media files, a pointer (e.g., a numerical identifier) to a playlist referencing the selected files, or a pointer to the selected media files themselves. In one embodiment, one or more computer-readable media store computer-executable instructions for performing the operations illustrated in FIG. 2. In an alternative embodiment, the menu structure includes a single playlist having a single media file.

After execution of the authoring software, a menu structure exists as illustrated on FIG. 3. Each element in a menu structure according to the invention stores tags or other data indicative of the media type associated with the element. In the example of FIG. 3, the top-level menu has a submenu storing video data and a submenu storing audio data. The submenu storing audio data has additional submenus organizing the audio data.

The user may navigate a path 302 in the menu structure to video content, or a path 304 in the menu structure to audio content.

Additional detail regarding creation of menus and playlists follows.

Playlist and Menu Creation

This section describes an exemplary method performed by the authoring software to create playlists from the selected media files and organize the playlists in a hierarchical structure. Operation of the method is based on the type and existing organization of the selected media files. If the selected media files are in a single layer of directories, the authoring software creates a playlist such as 'All media files by directory' in which the media files are grouped by directory name in alphabetical order by file names. If two layers of directories are present and the average top level directory contains less than one and a half subdirectories and more than seventy-five percent of the second level directories contains between five and twenty-five audio files (e.g., a typical artist-album structure with only a few albums for each artist), the authoring software creates a playlist 'All media files by directory' in which the media files are grouped by <directory name 1 + '-' directory name 2> in <directory name 1> then <directory name 2> in alphabetical order.

If more than two layers of directories are present or the two layers are too 'leafy' (e.g., contain too many subdirectories per directory) to readily flatten the hierarchy, then the authoring software creates a playlist 'All media files by directory' in which the media files are grouped by <lowest directory name> in the order the directories are found in the file system as the file system is traversed breadth first (not in alphabetical order). If there are any non-null playlist files present, the authoring software creates a playlist for each playlist present. Such playlists are stored under a 'Playlists' menu if there are more than five such playlists. If there are between two and six playlists present, the authoring software creates all remaining menus under a menu item entitled 'Other'.

Playlists for Audio Media Files

The authoring software creates a hierarchical menu structure for organizing the created playlists. In an exemplary embodiment with audio media files, the authoring

software creates groups and corresponding menus according to artist, composer, album, artist and album, genre, year, and directory (e.g., corresponding to the existing structure of the selected media files).

At the root of the hierarchical menu structure, a menu entitled 'Artists' has a submenu entitled 'All songs by artist' in which the media files are grouped by artist in alphabetical order. The authoring software creates a submenu for each artist in which the media files associated with the artist are grouped by album in alphabetical order or according to a release date for the album. If composer information is available for some of the selected media files (e.g., greater than twenty-five percent), the authoring software creates a menu 'Composer' at the root with a submenu entitled 'All songs by composer' in which the media files are grouped by composer in alphabetical order. The authoring software creates a submenu for each composer in which the media files associated with the composer are grouped by album in alphabetical order or according to a release date for the album. Media files without composer information are omitted.

The authoring software also creates a menu 'Albums' at the root with a submenu entitled 'All songs by album' in which the media files are grouped by album in alphabetical order or according to a release date for the album. The authoring software further creates a menu 'Artist – Album' at the root in which the media files are grouped according to artist-album pair (i.e., a single group is created for each artist-album pairing). The authoring software also creates a menu 'Genres' with a submenu 'All songs by genre' in which the media files are grouped by genre in alphabetical order (i.e., a single group is created for each genre). A menu 'Year' includes, but is not limited to, the submenus 'All songs by decade,' 'All songs by year,' and 'Directories.' The submenu 'All songs by decade' includes media files grouped by decade in ascending order (e.g., one group for each decade) with 'No year' songs grouped at the end of the groups or omitted. The submenu 'All songs by year' includes media files grouped by year in ascending order (e.g., one group for each year) with 'No year' songs grouped at the end of the groups or omitted. The submenu 'Directories' has groups that parallel an existing directory structure of the selected media files. That is, the submenu 'Directories' has one group for each directory (e.g. "Party Favorites", "Driving music").

Those skilled in the art will appreciate that the playlists and hierarchical structures described herein are merely exemplary. It is contemplated by the inventors that the invention includes other playlists, hierarchical structures, and the like, including any combination of audio data, video data, or still image data.

Playlists for Image Media Files

If the selected media files include still images, the authoring software creates a menu 'All pictures' at the root including every selected image for display (e.g., with a default duration of five seconds). In one embodiment, the duration can be changed via an advanced options dialog in a user interface associated with the authoring software. In addition, the authoring software examines the co-incidence of the selected images and selected music files in existing directories. If all of the directories that have images (e.g., larger than 200x200 pixels) also have music, the authoring software creates a playlist entitled "All pictures with all songs (endless)" arranged by directory so that each directory plays with the accompanying music. In one embodiment, the display time for each image equals the sum of the song lengths in the directory divided by the number of images. For example, each image may be displayed for a minimum of five seconds and a maximum of thirty seconds. The maximum and minimum settings are user adjustable in an advanced options dialog. Some images may not be displayed using this technique if the music in a directory is too short for all the images that are in the directory.

Alternatively, the playlists "All pictures with all songs (endless)" includes all songs playing in parallel with all the images. In one embodiment, the display time for each image equals the sum of the lengths of all the songs in the selected media files divided by the number of images. For example, each image is displayed for a minimum of five seconds and a maximum of thirty seconds. The maximum and minimum settings are user adjustable in an advanced options dialog. The image sequence is repeated as necessary to match the length of music or the music is repeated in the playlist to ensure there is enough music to match the total image length. Images are grouped, for example, by lowest directory name.

Playlists for Video Media Files

If the selected media files include video, the authoring software creates a menu 'Video' at the root of the hierarchical structure. The menu includes a playlist for each video file as well as a playlist for all video files entitled 'Play all videos' with video files concatenated in alphanumeric sequence. The authoring software also groups the video media files according to user input, as described in the following section.

Manual Adjustments to Automatic Playlists

The user may choose to adjust the hierarchical structure and/or default playlists. The user interface associated with the authoring software displays a diagram of the hierarchical structure and playlists with all options pre-checked for inclusion on the computer-readable medium. For example, the menu/playlist structure is displayed as a scrolling tree-view with checkboxes next to each menu or playlist item. Clearing a check box eliminates that entire branch of the tree (although the state of the items below it is preserved in case the user reinstates it).

The user interface associated with the authoring software may implement or disable various features. For example, promote and demote buttons may not be selectable (e.g., grayed-out) if the currently selected item cannot be promoted or demoted. No choice is given to the user where to demote an item, it automatically goes under the correct heading, adding that heading if necessary. For example, if there are three video files in the top-level menu and no 'Video' menu entry, the first demotion will create a heading 'Other video', the second demotion will add to that, the third demotion will rename the heading to 'Video.' No option is given to change the order of the menus or playlists or to change their names. All names are derived from metadata in files or are automatically generated names like 'Other video.' When an individual playlist is promoted to the same level as its containing category the containing category changes name to 'Other ...' (e.g. 'Other video').

Other options for the user interface associated with the authoring software include, but are not limited to, adjusting a 'Minimum image duration,' a 'Default image duration,' a 'Maximum image duration,' and a 'Repeat forever (kiosk mode).' The 'Minimum image duration' option allows the user to change the minimum duration for

images. The minimum duration is used with the maximum when creating parallel image and audio playlists. The 'Default image duration' option allows the user to change the default duration for images. The default duration is used for images playing as part of an image sequence. The 'Maximum image duration' option allows the user to change the maximum duration for images. This option is used when images and audio are presented together. The 'Repeat forever (kiosk mode)' option causes the repeat count field for each playlist to be set to zero. If this option is not checked, the repeat count is set to '1' for each playlist. In one embodiment, the user may be given options to merge playlists or perform more advanced editing of the menu names and menu structures created automatically. Users can also create playlist files prior to executing the authoring software. Those skilled in the art will note that the authoring software may include more or less functionality than described herein and still be within the scope of the invention.

After any manual adjustments have been made to the hierarchical structure and/or the default playlists, a menu structure exists such as illustrated in the exemplary menu structure of FIG. 4. In FIG. 4, a menu header 402 includes menu items entitled Audio 404, Video 406, and Images 408. The Audio menu item 404 references a menu header 410 including an All by Genre playlist item 412, a Playlist A playlist item 414, and an Artists Menu menu item 416. The Artists Menu menu item 416 references another menu header 418 that stores further menu items or playlist items. Each menu header includes a reference to a parent menu header. For example, menu header 418 has a reference to menu header 410. Similarly, menu header 410 has a reference to menu header 402. Menu header 402, being at the root of this exemplary menu structure, does not have a parent menu header.

In one embodiment, the authoring software includes a menu simplifier that traverses the created menu structure searching for menus that have only one child. The menu simplifier removes any such menus and promotes the only child in the hierarchy to take the place of the removed menu. In this fashion, the menu simplifier reduces the complexity of the menu structure to simplify navigation by the user.

The authoring software writes a setup file (e.g., in extensible markup language) to the computer-readable medium recording both the default settings and any manual settings of the user. If the user later adds more media files to the same medium (e.g., a

re-writeable medium), the authoring software examines the setup file to identify and apply those same settings. For example, if the user deletes all of the default playlists and selects only an 'All songs by lowest directory name' option, then such option will be the default for that computer-readable medium each time the user attempts to add media files to the medium. The authoring software identifies such customized options for the specific medium to the user.

Filtering Software

Referring next to FIG. 5, a flow chart illustrates exemplary operation of filtering software according to the invention. The filtering software displays menu information (e.g., metadata) on a display associated with the media player. The menu information enables a user to navigate one or more media files using the media player. Each of the media files has a media type and is adaptable for rendering by the media player as a function of the media type. The media player also has a media type. The filtering software reads a menu header at 502 having references to selected media files and a media type associated with each respective one of the selected media files. The filtering software adapts the read menu header at 504 by filtering the references to the selected media files as a function of a media type associated with a media player. In one embodiment, the filtering software shades the non-selectable menu items (e.g., turns to gray) in the user interface to differentiate these menu items from selectable menu items. The filtering software displays menu information at 506 from the adapted menu header to a user on a display associated with the media player. In one embodiment, the menu header includes a reference to another menu header and a media type associated with the other menu header to enable navigation between the menu headers. The filtering software also adapts the read menu header at 508 by filtering the reference to the other menu header as a function of the media type associated with the media player. The media type associated with each respective one of the selected media files and the media type associated with the media player comprise audio, video, and/or image. The references to the selected media files and the references to another menu header include, but are not limited to, a memory location offset to the selected media files a pointer to a playlist referencing the selected media files, or a pointer to the selected media files. In

one embodiment, one or more computer-readable media have computer-executable instructions for performing the method illustrated in FIG. 5.

Menu Display

The menu on a medium created by the invention software includes a single hierarchical structure of submenus and playlists. The menu tree may be arbitrarily deep and arbitrarily broad within an overall limit of 9,999 total nodes, for example. Other embodiments may implement other or different restrictions relating to the menu tree. Because each menu may contain up to 9,999 entries, media players display a scrolling or paged view of the menu. TV-connected players may display a thumbnail menu or a simple list menu (e.g., based on whether entries such as playlists or menus within the current menu have thumbnails or not). In one embodiment, TV-connected players display a thumbnail menu only when a thumbnail is present (i.e. non-zero entry) for every entry in the current menu.

The layout of thumbnails on the screen and the number of thumbnails displayed per page is player dependent. Some players may choose to display a continuous scrolling list, others may chose to paginate the menu and offer next/previous selections. On a 4:3 TV set, players will typically display a grid of 3x2 thumbnails while on a 16:9 TV set they may chose to display a 4x2 grid instead.

Players also display the playlist or menu name with the associated thumbnail. In the event that the player cannot display all thumbnails instantaneously, text labels are displayed first and the thumbnails added one by one as they are decoded. The player may allow scrolling to the next or previous page even while the thumbnails are being added in order to allow users to navigate quickly to a given item.

Menu Navigation

Menu display is player dependent; that is, each manufacturer may decide how to display menus. On a portable CD player, for example, the menu might be displayed one line at a time with the Next, Previous, Play/Select and Stop buttons used to navigate through the choices. Most players will display the top-level menu when a medium is inserted. Car receivers, however, may choose to bypass the menu on startup and begin

playback immediately with the first playlist on the disk or may resume playback on the song and playlist that was playing the last time they were used. The user would then need to press a 'menu' key to bring up the menu and use Next, Previous and Play/Select to navigate.

On a DVD player, a full graphical display might be shown complete with background image and optional thumbnails. Some DVD players may chose to number the menu items and use the 1-9 keys on a remote control, while some may simply present the menu items in a list or a grid format and use cursor keys for navigation and selection.

FIG. 6 illustrates a screen shot of an exemplary user interface 602 of the filtering software of the invention illustrating the media types available for navigation. In the example user interface 602 of FIG. 6, icons available for selection by the user include a video files icon 604, an audio files icon 606, an image files icon 608, and a presentations icon 610.

Data Structure for Menus

Referring next to FIG. 7, an exemplary block diagram illustrates a menu structure 700 having a file header 702 and one or more menus 704 such as menu #1 and menu #2. Each menu 704 includes a menu header 706 such as menu header #1 and menu header #2. A computer-readable medium stores the menu header 706 representing an organization of one or more media files. The media files each have a media type and are adaptable for rendering by the media player as a function of the media type. The media player also has at least one media type that the media player is capable of rendering. The menu headers 706 shown in FIG. 7 enables a user to navigate the media files using the media player. The menu header 706 comprises a playlist field (e.g., a playlist item 708) and a summary field (e.g., a playlist summary type). The playlist field stores a reference to one or more selected media files. The reference includes, but is not limited to, a memory location offset to the selected media files, a pointer to a playlist referencing the selected media files, or a pointer to the selected media files. Each of the selected media files has an associated media type. The summary field stores the media type for the selected media file referenced in the playlist field. A media player uses the menu header

706 to filter the selected media files as a function of a media type associated with the media player.

In one embodiment, the menu header 706 illustrated in FIG. 7 includes a menu field (e.g., a menu item 710) storing a reference to another data structure (e.g., another menu header 706 such as menu #2) to enable navigation between the data structures (e.g., between menu header #1 and menu header #2). Further, the menu header 706 may include a metadata field (e.g., menu name, playlist name in menu) storing a value corresponding to metadata selected from one or more of the following as a function of the authoring software: title, composer, performer, genre, and description of content of the selected media files.

A file (e.g., MENU.HMT) containing one or more of the data structures illustrated in FIG. 7 is written to the computer-readable medium storing the selected media files. The file is written such that there are no gaps between the data structures in the file. A description of exemplary data structures for use by the authoring software and the filtering software follows. Fields in exemplary menu file header 702 are shown in Table 1.

Table 1 - Menu File Header

Offset	Length	Field Name
0	8	Identifier
8	2	Version
10	4	Size of MENU.HMT
14	4	LCID
18	2	Offset of first menu
20	2	Menu Title Length
22	Variable	Menu Title

The Identifier field is an eight byte entry containing "MENU_HMT" in ASCII. The Version field is a two byte entry representing the version of the specification used to author this file. For example, version 1.20 would be stored as 0x78 (120 decimal). The Size of MENU.HMT field is a four byte entry containing the size of MENU.HMT in bytes. The LCID field is a four byte entry for the Language ID of this MENU.HMT file. The Offset of First Menu field is a two byte entry representing the byte offset from the beginning of the MENU.HMT to the first Menu Header 706. The Menu Title Length

field is a two byte entry containing the byte length of the menu title excluding the ending null UCS-2 character (two null bytes). The Menu Title field stores a menu title terminated with a null UCS-2 character (two null bytes). The maximum length for this entry is 65 UCS-2 characters including the null UCS-2 character. Players use this to display a title for the entire menu. An empty string (one NULL character) indicates that there is no title to display or that the authoring software has rendered the title onto the background image. For example: "My Hawaii Vacation", "2002 School Party".

Menus 704 support either a background image, solid background color or player default behavior. If a background image or background color is defined, then the text color is also defined. If the background image, background color and text color entries are zero, then the player uses its default behavior. Each submenu is referenced by a single parent menu forming a strict hierarchical menu structure. The fields in exemplary menu header 706 are shown in Table 2.

Table 2 - Menu Header

Offset	Length	Field Name
	4	Size of Menu Header
	4	Offset to Parent Menu
	4	Background Image ID (4:3)
	4	Background Image ID (16:9)
	4	Background Color
	4	Text Color
	2	Number of Items
	2	Menu Subtitle Length
	Variable	Menu Subtitle
	Variable	Menu or Playlist Item #1
		...
	Variable	Menu or Playlist Item #n

*n represents the number of menu items

The Size of Menu Header field is a four byte entry representing the size of the Menu Header 706 including the Menu Item 710 and Playlist Item 708 in bytes. The Offset to Parent Menu field is a four byte entry representing the byte offset from the beginning of MENU.HMT to the Parent Menu. This value is zero if this is the top level menu 704. The Background Image ID (4:3) field is a four byte entry defining the image to display as the background of this menu 704 on a 4:3 display. The image should be a 640x480 JPG file with an HMT extension. A value of zero indicates there is no

background image. The value is the index of a Menu Image file in CONTENTS.HMT (see FIG. 9). Players should display the background image centered on the display preserving the aspect ratio. Players should use a background color if any to fill in any uncovered area. The Background Image ID (16:9) field is a four byte entry defining the image to display as the background of this menu on a 16:9 display. The image should be a 852x480 JPG file with an HMT extension. A value of zero indicates there is no background image. If this value is non zero then there should be a valid Background Image ID (4:3) as well. The value is the index of the Menu Image file in CONTENTS.HMT. Players should display the background image centered on the display preserving the aspect ratio.

The Background Color field is a four byte entry defining the background color that should be used when this menu is rendered on the display. It is formatted as an RGB value with the following byte order: 0xFFRRGGBB. If a Background Image ID is defined then the background color should only be visible on areas of the display not covered by the background image if any. A value of zero indicates there is no background color. If a player is not capable of color rendering then this field may be ignored. The Text Color field is a four byte entry defining the color that the text on this menu 704 should be rendered on the display. It is formatted as an RGB value with the following byte order: 0xFFRRGGBB. If a Background Image ID or Background Color is defined then this entry should contain a non zero value. A value of zero is defined to mean that the player should use its default behavior. If a player is not capable of color rendering then this field may be ignored. The Number of Items field is a two byte entry defining the number of menu items 710 or Playlist Items 708 in this menu 704. The Menu Subtitle Length field is a two byte entry containing the byte length of the menu subtitle excluding the ending null UCS-2 character (two null bytes). The Menu Subtitle field is a menu subtitle terminated with a null UCS-2 character (two null bytes). The maximum length for this entry is 65 UCS-2 characters including the null UCS-2 character. Players should use this to display a subtitle for this menu 704. An empty string (one NULL character) indicates that there is no subtitle to display or that the authoring software has rendered the title onto the background image. The Menu or

Playlist Item field is a variable-sized entry representing either a Menu item 710 or a Playlist Item 708.

Fields in exemplary menu item 710 are shown in Table 3.

Table 3 - Menu Item

Offset	Length	Field Name
0	1	Type of entry
1	1	Menu Summary Type
2	4	Thumbnail ID (0 if no thumbnail)
6	4	Selected State Thumbnail ID
10	4	Offset to Menu
14	2	Menu Name Length
16	Variable	Menu Name

The Type of Entry field is a one byte entry defining whether this is a menu item 710 or a playlist item 708 data structure. For the menu item 710, this value should be the value of MENU as defined in Table 4.

Table 4 - Type of Entry

Type of Entry	Value
0	UNUSED
1	MENU
2	PLAYLIST
3 - 255	RESERVED

The Menu Summary Type field is used to define the type of playlists that are accessible via this menu item. The Thumbnail ID field is a four byte entry representing the Thumbnail ID for this menu item 710 in CONTENTS.HMT; if there is no thumbnail for this menu item 710 then the value is zero. The Selected State Thumbnail ID field is a four byte entry defining the Thumbnail ID that represents the selected state for this menu item 710 in CONTENTS.HMT. A value of zero indicates that the player should generate a bounding rectangle or other highlight to indicate selection using a color. The Offset to Menu field is a four byte entry defining the byte offset from the beginning of MENU.HMT to the Menu 704. The Menu Name Length field is a two byte entry containing the byte length of the menu name excluding the ending null UCS-2 character (two null bytes). The Menu Name field is the name of the Menu 704 terminated with a

null UCS-2 character (two null bytes). The maximum length for this entry is 65 UCS-2 characters including the null UCS-2 character.

Fields in exemplary playlist item 708 are shown in Table 5.

Table 5 - Playlist Item

Offset	Length	Field Name
0	1	Type of entry
1	1	Playlist Summary Type
2	4	Thumbnail ID (0 if no thumbnail)
6	4	Selected State Thumbnail ID
10	4	Playlist ID
14	4	Starting Group Index
18	4	Starting File Index
22	2	Playlist Name in Menu Length
24	Variable	Playlist Name in Menu

The Type of Entry field is a one byte entry defining whether this is a menu item 710 or playlist item 708. For the playlist item 708 this value should be the value of PLAYLIST. The Playlist Summary Type field is used to define the type of playlist that this menu item 710 references. The Thumbnail ID field is a four byte entry defining the Thumbnail ID for this menu item 710 in CONTENTS.HMT. If there is no thumbnail for this menu item then the value is 0. The thumbnail ID may be the same as the thumbnail ID in a playlist header. The Selected State Thumbnail ID field is a four byte entry defining the Thumbnail ID that represents the selected state for this menu item 710 in CONTENTS.HMT. A value of zero indicates that the player should generate a bounding rectangle or other highlight to indicate selection using a color. The Playlist ID field is a four byte entry defining the ID of the playlist for this menu item 710 in CONTENTS.HMT. The Starting Group Index field is a four byte entry defining the index of the Group in the playlist file to start playback. A value of one indicates the first group in the playlist. The Starting File Index field is a four byte entry defining the index of the File in the group to start playback. A value of one indicates the first file in the group. In addition, this value is one if the Group is a Parallel playlist group (PIA).

The Starting Group Index and Starting File Index together allow one playlist to be referenced multiple times in the menu 704. For example, a menu 704 could show thumbnails for every image on the disk and each thumbnail would take you to a looping

playlist of all images beginning with the selected image. Playlist Name in Menu Length field is a two byte entry containing the byte length of the playlist name in menu 704 excluding the ending null UCS-2 character (two null bytes). Playlist Name in Menu field is the name of the Playlist as it appears in the Menu 704. The name is terminated with a null UCS-2 character (two null bytes). The maximum length for this entry is 65 UCS-2 characters including the null UCS-2 character.

The Menu Image File Table is used to list all of the thumbnails and background images used in the MENU.HMT file (see Appendix A). The Menu Image File Table include a list of Menu Image File Entries followed by a file name table (see Table 7). One Menu Image File Entry exists for each Menu Image File. Each Menu Image file entry is formatted as shown in Table 6.

Table 6 - Menu Image File Entry Format

Offset	Length	Field Name
0	4	Offset to Directory Number
4	2	File Type
6	2	Special Flags

Table 7 - File name table

Offset	Length	Field Name
	4	Directory Number CID#n+1
	2	File Name Length CID#n+1
	variable	File Name CID#n+1
		...
	4	Directory Number CID#m
	2	File Name Length CID#m
	variable	File Name CID#m

* m-n represents the number of Menu Image files

The Offset to Directory Number field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the directory number for this entry. The File Type field is a two byte entry representing the File type (e.g., the data encoding format). Valid values include thumbnail, menu background (4x3), and menu background (16x9). The Special Flags field is a two byte entry holding special flags. The Directory Number field is a four byte entry representing the index in the directory table of the directory that contains this file. The File Name Length field is a two byte entry containing the byte length of the file name excluding the ending null UCS-2 character (two null bytes). The

File Name field is the file name terminated with a null UCS-2 character (two null bytes). The maximum length for this entry is 111 UCS-2 characters including the null UCS-2 character. This entry uses Big-Endian word layout.

Support for Multiple Languages

The data structure (e.g., menu header) 706 illustrated in FIG. 7 includes a locale field storing a value identifying metadata associated with the selected media files. The metadata corresponds to a specific language. The locale field provides support for multiple languages of menu and text data on a single storage medium. The structure of an exemplary locale identifier (LCID) 802 is illustrated in FIG. 8. In an exemplary compressed media format as described in Appendix A, the file headers of TEXT.HMT and MENU.HMT contain the LCID 802 that represents their language and they must match. CONTENTS.HMT contains a list of LCID's for the languages on this storage media.

The media players use the first item in the list of LCIDs 802 as the default language. If there is more than one language defined then the player should choose the LCID 802 to play from the list (e.g., based on the language or locale of the media player). The player loads the corresponding TEXT.HMT and MENU.HMT by using a Directory Number for those files. The first (default) LCID 802 entry should have the Directory Number that is the location of CONTENTS.HMT; i.e. the default language MENU.HMT and TEXT.HMT is in the same directory as CONTENTS.HMT.

The LCID 802 itself has several parts: the first ten bits are the primary language ID 804, which contains the language itself. The next six bits contain the sublanguge ID 806, which is often used to differentiate regions that share the same primary language ID 804. The next four bits represent the sort ID 808, which can differentiate between alternate sorting orders that might be used for the same language and region. The remaining 12 bits are reserved for future use and should always be zero.

For example, the LCIDs 802 may be stored in a file such as CONTENTS.HMT described below.

Structure of CONTENTS.HMT file

FIG. 9 illustrates the structure of an exemplary CONTENTS.HMT file. The exemplary CONTENTS.HMT file includes a file header 902, a directory table 904, playlist file entries 906, audio file entries and audio file name table 908, menu image file entries and menu image file name table 910, image file entries and image file name table 912, and video file entries and video file name table 914. The offsets in the accelerator files are byte offsets from the beginning of the file. The Audio 908, Menu Image 910, Image 912 and Video 914 file tables start on a 2K boundary. The directory table 904 is written after the file header 902 including the LCID table with no extra padding. If there are no files available of a given type, the corresponding table is empty. This is authored as both the number of files and the offset to the table being zero. Table 8 illustrates exemplary file header 902.

Table 8 - File manifest header

Offset	Length	Field Name
0	8	Identifier
8	2	Version
10	8	HMT Generation
18	4	Size of CONTENTS.HMT
22	4	Number of Directories
26	4	Offset to Directory table
30	4	Number of Playlist Files
34	2	Size of playlist Entry
36	4	Offset to Playlist file table
40	4	Number of Audio files
44	2	Size of Audio Entry
46	4	Offset to Audio file table
50	4	Number of Menu Image files
54	2	Size of Menu Image Entry
56	4	Offset to Menu Image file table
60	4	Number of Image files
64	2	Size of Image Entry
66	4	Offset to Image file table
70	4	Number of Video files
74	2	Size of Video Entry

76	4	Offset to Video file table
80	2	Number of LCID entries
82	4	Directory Number 1
86	4	LCID 1
		...
82 + 8*(n-1)	4	Directory Number n
86 + 8*(n-1)	4	LCID n

The Identifier field is an eight byte entry containing "INFO_HMT" in ASCII. The Version field is a two byte entry representing the version of the specification used to author this file. For example, version 1.20 would be stored as 0x78 (120 decimal). The HMT Generation field is an eight byte entry representing the generation of LSN.HMT that matches this CONTENTS.HMT. This CONTENTS.HMT is only used with an LSN.HMT file that contains the same HMT Generation number. If the HMT Generation values do not match, the player ignores the LSN.HMT file. A value of zero indicates that there is no LSN.HMT file. The Size of CONTENTS.HMT field is a four byte entry containing the size of CONTENTS.HMT in bytes. The Number of Directories field is a four byte entry containing the number of directories in directory table 904. The Offset to Directory Table field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the directory table 904. The Number of Playlist Files field is a four byte entry containing the number of playlist files in a playlist file table. There is at least one playlist file. The Size of Playlist Entry field is a two byte entry representing the size of a Playlist Entry in bytes. The Offset to Playlist File Table field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the playlist file table. The Number of Audio Files field is a four byte entry containing the number of audio files in audio file table 908. The Size of Audio Entry field is a two byte entry representing the size of an Audio File Entry in bytes. The Offset to Audio File Table field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the audio file table 908. The offset is a multiple of 2K, any gaps between the end of the Playlist File Table and the start of the Audio File Table 908 are ignored. A value of zero indicates that there are no Audio Entries in CONTENTS.HMT.

The Number of Menu Image Files field is a four byte entry containing the number of Menu Image files in a Menu Image file table 910. The Size of Menu Image

Entry field is a two byte entry representing the size of a Menu Image File Entry in bytes. The Offset to Menu Image File Table field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the Menu Image file table 910. The offset is a multiple of 2K, any gaps between the end of the Audio File Table 908 and the start of the Menu Image File Table 910 are ignored. A value of zero indicates that there are no Menu Image Entries in CONTENTS.HMT. The Number of Image Files field is a four byte entry containing the number of Image files in an Image file table 912. The Size of Image Entry field is a two byte entry representing the size of an Image File Entry in bytes. The Offset to Image File Table field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the Image file table 912. The offset is a multiple of 2K, any gaps between the end of the Menu Image File Table 910 and the start of the Image File Table 912 are ignored. A value of zero indicates that there are no Image Entries in CONTENTS.HMT. The Number of Video Files field is a four byte entry containing the number of video files in a video file table 914. The Size of Video Entry field is a two byte entry representing the size of a Video File Entry in bytes. The Offset to Video File Table field is a four byte entry representing the byte offset from the beginning of CONTENTS.HMT to the video file table 914. The offset is a multiple of 2K, any gaps between the end of the Image File Table 912 and the start of the Video File Table 914 are ignored. A value of zero indicates that there are no Video Entries in CONTENTS.HMT. The Number of LCID entries field is a two byte entry representing the number of languages on the medium. This value is at least 1. The Directory Number field is a four byte entry representing the index in directory table 904 of the directory that contains the TEXT.HMT and MENU.HMT for this locale identifier. This is a subdirectory of \HIGHMAT with the LCID as the directory name. However, a reference to \HIGHMAT itself is valid for the default language. The LCID n field is a four byte entry representing the locale identifier.

The data structures of the invention are organized such that only the current menu needs to be in memory associated with the media player. The flexible menu system of the invention scales from a minimal device (e.g., portable CD player) to a large, 16:9 television display (e.g., a DVD player menu). The medium specifies menu hierarchy, but the layout, thumbnails (consumer images, video thumbnails, or predefined art like these),

and background image vary per media player. In one embodiment, menu or playlist names are user entered or auto-generated. Menus are filtered to display only those menu items that the player is capable of displaying or playing (e.g. an audio-only player will remove all image and video menu entries to display only pure-audio selections).

Exemplary Operating Environment

As illustrated in FIG. 10, the invention is operable as part of an exemplary compressed media format 1002 having a set of small files containing metadata, menus, and playlists in a compiled binary format designed for playback on feature-rich PC media players as well as low cost media players. See Appendix A for an example of accelerator files for use as part of an exemplary compressed media format. The exemplary compressed media format of the invention encompasses audio, still images, and video media files 1004 in various formats. The compressed media format for use with the invention defines a logical format for organizing compressed media files 1004 in a file system 1006 on computer-readable media 1008 such as optical discs (e.g., CD-ROM, CD-R, CD-RW, DVD-RAM, DVD-R, DVD-RW, DVD+RW, DVD+R, DVD-ROM), flash memory (e.g., COMPACTFLASH brand, secure digital, MEMORY STICK brand), magnetic media (e.g., hard disks), and the like.

FIG. 11 shows one example of a general purpose computing device in the form of a computer 130. In one embodiment of the invention, a computer such as the computer 130 is suitable for use in the other figures illustrated and described herein. Computer 130 has one or more processors or processing units 132 and a system memory 134. In the illustrated embodiment, a system bus 136 couples various system components including the system memory 134 to the processors 132. The bus 136 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

The computer 130 typically has at least some form of computer readable media. Computer readable media, which include both volatile and nonvolatile media, removable and non-removable media, may be any available medium that can be accessed by computer 130. By way of example and not limitation, computer readable media comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. For example, computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile discs (DVD) or other optical disc storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store the desired information and that can accessed by computer 130. Communication media typically embody computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and include any information delivery media. Those skilled in the art are familiar with the modulated data signal, which has one or more of its characteristics set or changed in such a manner as to encode information in the signal. Wired media, such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared, and other wireless media, are examples of communication media. Combinations of the any of the above are also included within the scope of computer readable media.

The system memory 134 includes computer storage media in the form of removable and/or non-removable, volatile and/or nonvolatile memory. In the illustrated embodiment, system memory 134 includes read only memory (ROM) 138 and random access memory (RAM) 140. A basic input/output system 142 (BIOS), containing the basic routines that help to transfer information between elements within computer 130, such as during start-up, is typically stored in ROM 138. RAM 140 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 132. By way of example, and not limitation, FIG. 11 illustrates operating system 144, application programs 146, other program modules 148, and program data 150.

The computer 130 may also include other removable/non-removable, volatile/nonvolatile computer storage media. For example, FIG. 11 illustrates a hard disk drive 154 that reads from or writes to non-removable, nonvolatile magnetic media. FIG. 11 also shows a magnetic disk drive 156 that reads from or writes to a removable, nonvolatile magnetic disk 158, and an optical disc drive 160 that reads from or writes to a removable, nonvolatile optical disc 162 such as a CD-ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile discs, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 144, and magnetic disk drive 156 and optical disc drive 160 are typically connected to the system bus 136 by a non-volatile memory interface, such as interface 166.

The drives or other mass storage devices and their associated computer storage media discussed above and illustrated in FIG. 11, provide storage of computer readable instructions, data structures, program modules and other data for the computer 130. In FIG. 11, for example, hard disk drive 154 is illustrated as storing operating system 170, application programs 172, other program modules 174, and program data 176. Note that these components can either be the same as or different from operating system 144, application programs 146, other program modules 148, and program data 150. Operating system 170, application programs 172, other program modules 174, and program data 176 are given different numbers here to illustrate that, at a minimum, they are different copies.

A user may enter commands and information into computer 130 through input devices or user interface selection devices such as a keyboard 180 and a pointing device 182 (e.g., a mouse, trackball, pen, or touch pad). Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to processing unit 132 through a user input interface 184 that is coupled to system bus 136, but may be connected by other interface and bus structures, such as a parallel port, game port, or a Universal Serial Bus (USB). A monitor 188 or other type of display device is also connected to system bus 136 via an interface, such as a video interface 190. In addition to the monitor 188, computers often include

other peripheral output devices (not shown) such as a printer and speakers, which may be connected through an output peripheral interface (not shown).

The computer 130 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 194. The remote computer 194 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 130. The logical connections depicted in FIG. 11 include a local area network (LAN) 196 and a wide area network (WAN) 198, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and global computer networks (e.g., the Internet).

When used in a local area networking environment, computer 130 is connected to the LAN 196 through a network interface or adapter 186. When used in a wide area networking environment, computer 130 typically includes a modem 178 or other means for establishing communications over the WAN 198, such as the Internet. The modem 178, which may be internal or external, is connected to system bus 136 via the user input interface 194, or other appropriate mechanism. In a networked environment, program modules depicted relative to computer 130, or portions thereof, may be stored in a remote memory storage device (not shown). By way of example, and not limitation, FIG. 11 illustrates remote application programs 192 as residing on the memory device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Generally, the data processors of computer 130 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in conjunction with

a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described herein.

For purposes of illustration, programs and other executable program components, such as the operating system, are illustrated herein as discrete blocks. It is recognized, however, that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

Although described in connection with an exemplary computing system environment, including computer 130, the invention is operational with numerous other general purpose or special purpose computing system environments or configurations. The computing system environment is not intended to suggest any limitation as to the scope of use or functionality of the invention. Moreover, the computing system environment should not be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include, but are not limited to, routines, programs, objects, components, and data structures that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

In operation, computer 130 executes computer-executable instructions such as those illustrated in FIG. 2 to organize media files in an adaptive menu structure. In

addition, computer 130 in a media player executes computer-executable instructions such as those illustrated in FIG. 5 to filter the menu structure according to the capabilities of the media player.

When introducing elements of the present invention or the embodiment(s) thereof, the articles "a," "an," "the," and "said" are intended to mean that there are one or more of the elements. The terms "comprising," "including," and "having" are intended to be inclusive and mean that there may be additional elements other than the listed elements.

In view of the above, it will be seen that the several objects of the invention are achieved and other advantageous results attained.

As various changes could be made in the above constructions, products, and methods without departing from the scope of the invention, it is intended that all matter contained in the above description and shown in the accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

Appendix A

FIG. A1 illustrates a file system in an exemplary compressed media format for use with the invention software. The accelerator files exist on a medium under a top-level directory called 'HIGHMAT' and include the following files: CONTENTS.HMT, nnnnnnnn.HMT, MENU.HMT, and TEXT.HMT. The CONTENTS.HMT file contains information about all the media files present on the medium. It contains a directory table, followed by the file entry tables for each of the supported file types (Audio, Video, Image and Playlist). There is one file for each playlist on the medium called 'nnnnnnnn.HMT' where nnnnnnnn is a hexadecimal playlist file identifier. These playlist files are created in a 'PLAYLIST' subdirectory. The MENU.HMT file contains the menu structure. The TEXT.HMT file contains all the textual information needed during playback.

In alternative embodiment, the following files are included: <Thumbnail>.HMT and <Background>.HMT. There is one file for each thumbnail on the medium (e.g., 160x120 JPG format with .HMT extension). The authoring software may name the files as appropriate. The Thumbnail files are created in an 'IMAGES' subdirectory. There is one file for each menu background on the medium. Backgrounds with a 4:3 aspect ratio are in 640x480 JPG format with .HMT extension. Backgrounds with a 16:9 aspect ratio are in 852x480 JPG format with .HMT extension. The authoring software may name the files as appropriate. The Background files are created in the 'IMAGES' subdirectory.

The HIGHMAT directory and all of the files within it may be marked as hidden by the software that creates the medium in order to lessen user confusion. Players should handle media whether these files and directories are hidden or not. All of the binary structures should be encoded using Little Endian byte ordering. File and Directory names should use Big Endian byte ordering. When the player encounters an unknown value the default behavior should be to ignore the offending file. For example, if Playlist Summary Type is equal to 255, then the entire playlist should be skipped.

The mapping of metadata to the fields in TEXT.HMT is outlined in the following table.

Table A1

	Audio Files	Image Files	Video Files	Playlist Files
Text1	Audio Title	Image Title	Video Title	Playlist Name

Text2	Artist Name	Creating Device Name (DSC Name)	Artist Name	None
Text3	Composer Name	Date Image Taken	Created Date	None
Text4	Album Name	Album Name	Album Name	None
Text5	Genre Name	Genre Name	Genre Name	Genre Name
Group	None	None	None	Group Name
Extra Text	Lyrics	Note (Comment)	Note (Comment)	None
Lyrics				
Copyright	Copyright	Copyright	Copyright	None
Album Artist	Album Artist	None	None	None

The structure of TEXT.HMT file is shown in Table A2.

Table A2

File Header
Contents Text Entry Table
Group Text Entry Table
Extra Text Entry Table
Contents Text Data
Group Name Text Data
Extra Text Data

The file should be written such that there are no gaps between the data structures in the file.

Table A3 - Text File Information Format

Offset	Length	Field Name
0	8	Identifier
8	2	Version
10	4	File size
14	4	Number of Contents Text Entries
18	4	Number of Group Text Entries
22	4	Number of Extra Text Entries
26	4	Offset of contents text entries
30	4	Offset of group text entries
34	4	Offset of extra text entries
38	4	LCID
42	2	Disk Name Length
44	Variable	Disk Name

Identifier

This field should be "TEXT_HMT" by ASCII in 8 bytes.

Version

This 2 byte entry represents the version of the HMT specification used to author this file. For example version 1.20 would be stored as 0x78 (120 decimal)

File size

This field should contain the size of this TEXT.HMT file in 4 bytes.

Number of Contents Entries

This 4 byte entry defines the number of Contents Text Entries.

Number of Group Entries

This 4 byte entry defines the number of Group Text Entries.

Number of Extra Text Entries

This 4 byte entry defines the number of Extra Text Entries.

Offset of Contents text entries

This 4 byte entry is the offset from the beginning of TEXT.HMT to the Contents text entry table.

Offset of Group text entries

This 4 byte entry is the offset from the beginning of TEXT.HMT to the Group text entry table.

Offset of Extra text entries

This 4 byte entry is the offset from the beginning of TEXT.HMT to the Extra text entry table.

LCID

This 4 byte entry is the Language ID of this MENU.HMT file.

Disk Name Length

This 2 byte entry contains the byte length of the disk name excluding the ending null UCS-2 character (two null bytes).

Disk Name

This variable length entry represents the name of the disk. The Disk Name is UCS-2 and should be terminated with a null UCS-2 character (two null bytes). The maximum length for this entry is 65 UCS-2 characters including the null UCS-2 character.

Text Entry

All of the text items are stored using the following format:

Table A4 - Text Entry

Offset	Length	Field Name
0	2	Length
2	Variable	Text

Length

This 2 byte entry contains the byte length of the text excluding the ending null UCS-2 character (two null bytes).

Text

This is the UCS-2 Text Entry. The text should be terminated with a null UCS-2 character (two null bytes). The maximum length for text entries is shown in the following table.

Table A5 - Text Length

Type of Text Data	Maximum Length
Group Name	65
Extra Text Entries	32,767
All Other	1,024

The length is defined as UCS-2 characters and includes the ending Null UCS-2 character.

The Contents text entry Table includes a list of Contents Text Entries (see Table A6). There is one Contents Text Entry for each file in CONTENTS.HMT (Playlist, Audio, Menu Image, Image and Video) in the same order as the files are listed in CONTENTS.HMT.

Table A6 - Contents Text Entry

Offset	Length	Field Name
0	4	Contents ID
4	4	Offset of Text1 entry
8	4	Offset of Text2 entry
12	4	Offset of Text3 entry
16	4	Offset of Text4 entry
20	4	Offset of Text5 entry
24	4	Offset of extra text entry

Contents ID

This 4 bytes entry represents of Contents ID for this Contents text entry.

Offset of Text1 entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Text1 entry. It is required that each contents item in CONTENTS.HMT has a Text1 entry.

Offset of Text2 entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Text2 entry. If the contents item does not have Text2 data then this entry should be 0.

Offset of Text3 entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Text3 entry. If the contents item does not have Text3 data then this entry should be 0.

Offset of Text4 entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Text4 entry. If the contents item does not have Text4 data then this entry should be 0.

Offset of Text5 entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Text5 entry. If the contents item does not have Text5 data then this entry should be 0.

Offset of extra text entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Extra text entry. If the contents item does not have Extra text data then this entry should be 0.

The Group text entry Table includes a list of Group Text Entries (Table A7). There should be one Group Text Entry for each group in the playlist files listed in CONTENTS.HMT. The entries should be in the ascending group number order.

Table A7 - Group Text Entry Format

Offset	Length	Field Name
0	4	Group number
4	4	Offset of group name text entry

Group number

This 4 bytes entry represents the Group number for this Group text entry.

Offset of Group name text entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of Group name text entry. It is required that each group has a name.

The extra text entry table is defined as zero or more Extra Text Entries as defined in Table A8. The entries should be stored in ascending Contents ID order. If there are zero Extra Text Entries the Number of Extra Text Entries field, the Base offset of Extra Text Entries and the Base offset of Extra Text Data should all be 0.

Table A8 - Extra Text Entry

Offset	Length	Field Name
0	4	Contents ID
4	1	Number of Extra text data
5	1	Reserved
6	2	Type of 1st text data
8	4	Offset of 1st text entry
		...
6+6*(n-1)	2	Type of n-th text data
8+6*(n-1)	4	Offset of n-th text entry

Contents ID

This 4 bytes entry represents of Contents ID of the file associated with this Extra text entry in CONTENTS.HMT.

Number of Extra text data

This 1 byte entry represents of number of Extra text data associated by this Extra text entry.

Reserved

This 1 byte entry is reserved for future use.

Type of n-th text data

This 2 byte entry contains one of the values defined in the table below.

Table A9 - Type of n-th Text Data

Extra Text Type	Value
0	UNUSED
1	LYRICS
2	COPYRIGHT
3	ALBUM ARTIST
4 – 65,535	RESERVED

Offset of n-th text entry

This 4 bytes entry represents the offset from the beginning of TEXT.HMT for first byte of each Extra text entry.

1. A method of organizing one or more media files to enable a user to navigate the media files using a media player, said media files each having a media type associated therewith and being adapted for rendering by the media player as a function of the media type, said media player also having a media type associated therewith, said method comprising:

creating a menu header;

populating the created menu header with references to one or more selected media files;

identifying a media type associated with each of the selected media files; and

populating the menu header with the identified media types, said populated menu header being for use by a media player to filter the selected media files as a function of a media type associated with the media player.

2. The method of claim 1, further comprising populating the menu header with a reference to another menu header to enable navigation between the menu header and said other menu header.

3. The method of claim 2, further comprising adapting the menu header by filtering the reference to the other menu header as a function of the media type associated with the media player.

4. The method of claim 1, further comprising storing the selected media files and the populated menu header on a computer-readable medium compatible with the media player.

5. The method of claim 1, wherein the media type associated with the selected media files and the media type associated with the media player comprise one or more of the following: audio, video, and image.

6. The method of claim 1, wherein the references populating the menu header comprise one of the following: a memory location offset to the selected media files, a pointer to a playlist referencing the selected media files, or a pointer to the selected media files.

7. The method of claim 1, further comprising:

reading the populated menu header;

adapting the read menu header by filtering the references to the selected media files as a function of the media type associated with the media player; and

displaying menu information from the adapted menu header to a user on a display associated with the media player.

8. The method of claim 1, wherein one or more computer-readable media have computer-executable instructions for performing the method recited in claim 1.

9. In a media authoring tool, one or more computer-readable media having computer-executable components for organizing one or more media files to enable a user to navigate the media files using a media player, said media files each having a media type associated therewith and being adapted for rendering by the media player as a function of the media type, said media player also having a media type associated therewith, said components comprising:

an initialization module for creating a menu header;

a playlist module for populating the created menu header from the initialization module with references to one or more selected media files;

a summary module for identifying a media type associated with each of the selected media files from the playlist module; and

a propagation module for populating the menu header with the identified media types from the summary module, said populated menu header being for use by a media player to filter the selected media files as a function of a media type associated with the media player.

10. The computer-readable media of claim 9, further comprising a menu module for populating the menu header with a reference to another menu header to enable navigation between the menu header and said other menu header.

11. The computer-readable media of claim 9, further comprising a writer module for storing the selected media files and the populated menu header on a computer-readable medium compatible with the media player.

12. The computer-readable media of claim 9, wherein the media type associated with the selected media files and the media type associated with the media player comprise one or more of the following: audio, video, and image.

13. The computer-readable media of claim 9, wherein the references populating the menu header comprise one of the following: a memory location offset to the selected media files, a pointer to a playlist referencing the selected media files, or a pointer to the selected media files.

14. The computer-readable media of claim 9, further comprising:

an input module for reading the populated menu header;

a menu module for adapting the read menu header from the input module by filtering the references to the selected media files as a function of the media type associated with the media player; and

a user interface module for displaying menu information from the adapted menu header from the menu module to a user on a display associated with the media player.

15. One or more computer readable media organizing one or more media files, said media files each having a media type associated therewith and being adapted for rendering by a media player as a function of the media type, said media player displaying menu information on a display associated with said media player, said menu information enabling a user to navigate the one or more media files using the media player, said media player also having a media type associated therewith, said computer-readable media comprising:

an input module for reading a menu header having references to one or more selected media files and a media type associated with each respective one of said selected media files;

a menu module for adapting the read menu header from the input module by filtering the references to the selected media files as a function of a media type associated with a media player; and

a user interface module for displaying menu information from the adapted menu header from the menu module to a user on a display associated with the media player.

16. The computer-readable media of claim 15, wherein said menu header includes a reference to another menu header and a media type associated with said other menu header to enable navigation between said menu header and the other menu header, and further comprising a navigation module for adapting the read menu header by filtering the reference to the other menu header as a function of the media type associated with the media player.

17. The computer-readable media of claim 15, wherein the references to the selected media files comprise one of the following: a memory location offset to the selected media files, a pointer to a playlist referencing the selected media files, or a pointer to the selected media files.

18. The computer-readable media of claim 15, further comprising:

a playlist field storing the references to the selected media files; and

a summary field storing the media types for the selected media files referenced in the playlist field.

19. The computer-readable media of claim 15, further comprising a metadata field storing a value corresponding to content-related data and wherein the content-related data is selected from one or more of the following as a function of the media player: title, composer, performer, genre, studio, director, rating, artist, and description of content of the one or more selected media files.

20. The computer-readable media of claim 15, further comprising a locale field storing a value identifying metadata associated with the selected media files, wherein the metadata corresponds to a specific language.

1. Abstract

Creating and displaying an adaptive menu structure for media files. The invention includes authoring software to create the menu structure to enable a user to navigate the media files using a media player. The invention also includes filtering software to adapt the created menu structure to display menu information on a display associated with the media player as a function of a media type associated with the media player. In one embodiment, the invention is operable as part of a compressed media format having a set of small files containing metadata, menus, and playlists in a compiled binary format designed for playback on feature-rich personal computer media players as well as low cost media players.

2. Representative Drawing

FIG. 1

FIG. 1

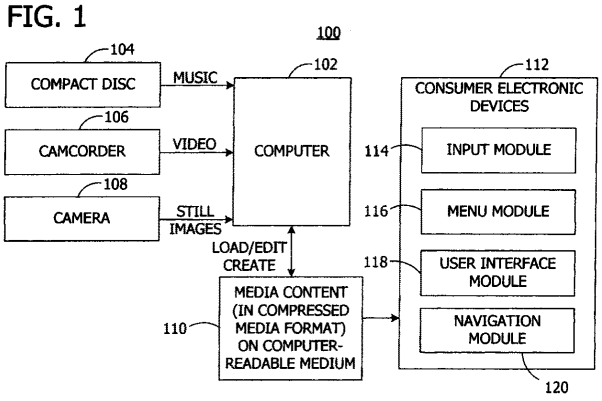


FIG. 2

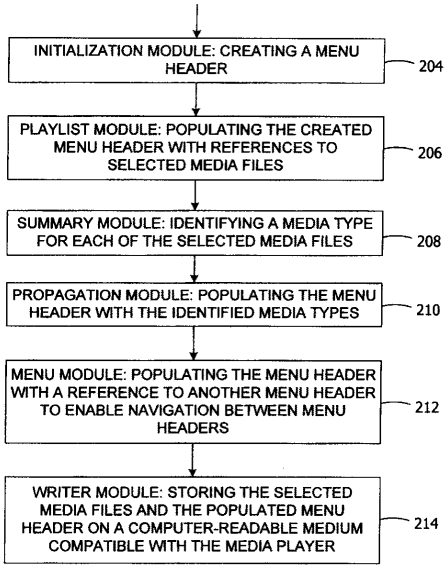


FIG. 3

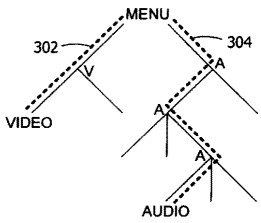


FIG. 4

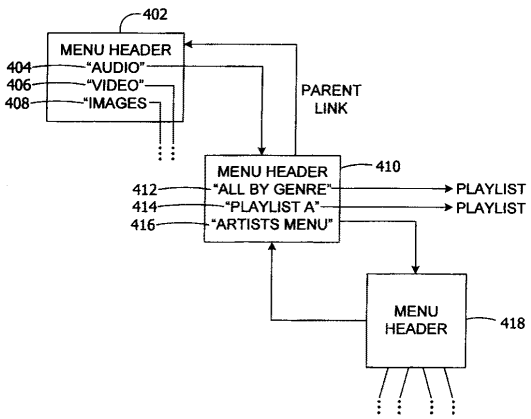


FIG. 5

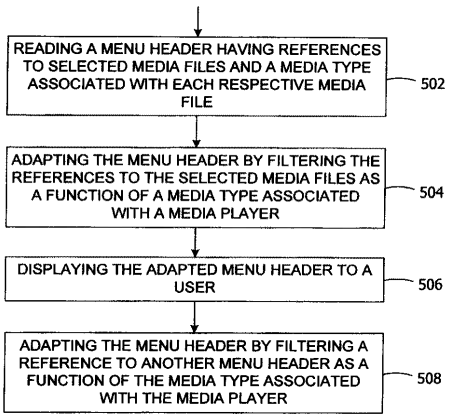


FIG. 6

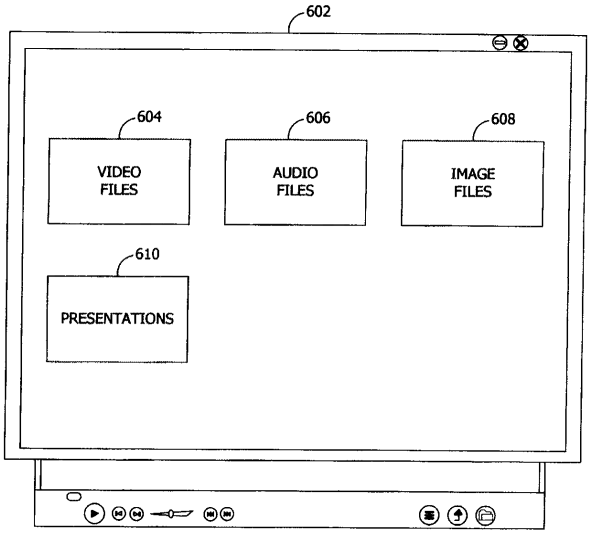


FIG. 7

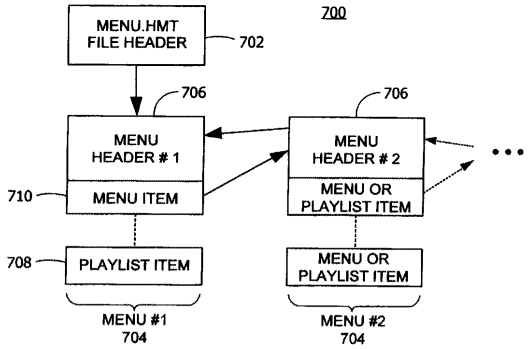


FIG. 8

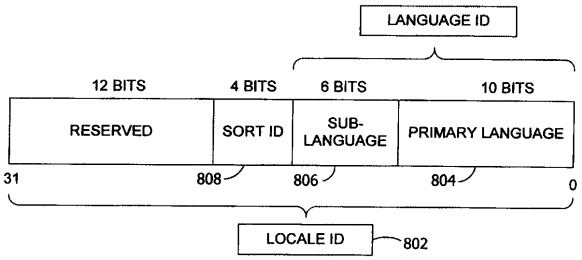


FIG. 9

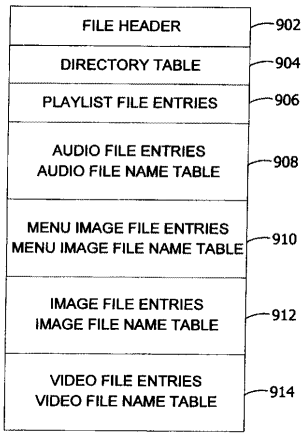
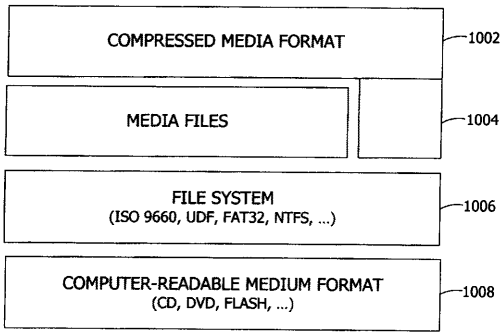


FIG. 10



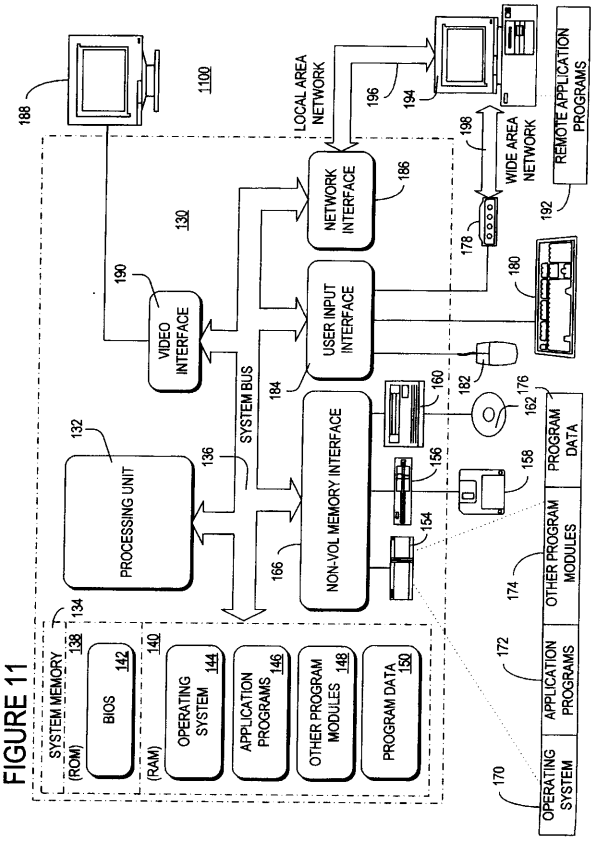


FIGURE 11

FIG. A1