

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4383378号  
(P4383378)

(45) 発行日 平成21年12月16日(2009.12.16)

(24) 登録日 平成21年10月2日(2009.10.2)

(51) Int.Cl.	F I
HO4R 3/00 (2006.01)	HO4R 3/00 310
HO4R 27/00 (2006.01)	HO4R 27/00 B
GO6F 3/14 (2006.01)	GO6F 3/14 320A
GO6F 13/00 (2006.01)	GO6F 13/00 357A
GO6F 13/14 (2006.01)	GO6F 13/14 330B

請求項の数 31 外国語出願 (全 83 頁) 最終頁に続く

(21) 出願番号	特願2005-93304 (P2005-93304)	(73) 特許権者	592051453
(22) 出願日	平成17年3月28日 (2005.3.28)		ハーマン インターナショナル インダストリーズ インコーポレイテッド
(65) 公開番号	特開2005-323344 (P2005-323344A)		アメリカ合衆国 カリフォルニア州 91329 ノースリッジ パルポア ブールヴァード 8500
(43) 公開日	平成17年11月17日 (2005.11.17)	(74) 代理人	100078282
審査請求日	平成17年5月18日 (2005.5.18)		弁理士 山本 秀策
(31) 優先権主張番号	60/557, 117	(74) 代理人	100062409
(32) 優先日	平成16年3月26日 (2004.3.26)		弁理士 安村 高明
(33) 優先権主張国	米国 (US)	(74) 代理人	100113413
(31) 優先権主張番号	60/644, 967		弁理士 森下 夏樹
(32) 優先日	平成17年1月18日 (2005.1.18)		
(33) 優先権主張国	米国 (US)		
前置審査			

最終頁に続く

(54) 【発明の名称】 オーディオ関連システムのリンク管理

(57) 【特許請求の範囲】

【請求項1】

複数のパフォーマンス機器マネージャを用いて、複数のオーディオ関連デバイス上の設定に対する変化を追跡する方法であって、該複数のオーディオ関連デバイスは、第1のオーディオ関連デバイスを含み、該第1のオーディオ関連デバイスは、第1の状態変数と、該第1の状態変数の値を変化させるように動作可能なユーザ・インターフェースとを有し、

該方法は、

第1のパフォーマンス機器マネージャが、該第1のオーディオ関連デバイスに関連付けられた第1のコントロール・パネル・ビューを構築することであって、該第1のコントロール・パネル・ビューは、該第1の状態変数を表示するように動作可能である、ことと、

該第1のパフォーマンス機器マネージャが、該第1の状態変数の値に対する変化を追跡することと、

該第1の状態変数の値が変化したことに応答して、該第1のオーディオ関連デバイスが、該第1の状態変数に対して、該第1のオーディオ関連デバイスと該第1のパフォーマンス機器マネージャとの間に確立された第1のサブクリプションに従って、該第1の状態変数の値が変化したことを該第1のパフォーマンス機器マネージャに通知することと、

該第1の状態変数の値が変化したという第1の通知に応答して、該第1のパフォーマンス機器マネージャが、該変化した第1の状態変数を表示するように該第1のコントロール・パネル・ビューを更新することと、

第2のパフォーマンス機器マネージャが、該第1の状態変数に対して、該第2のパフォーマンス機器マネージャの第2のコントロール・パネル・ビューと該第1のコントロール・パネル・ビューとの間で第2のサブスクリプションを確立することと、

該第1の状態変数の値が変化したという該第1の通知を受信したことに応答して、該第1のパフォーマンス機器マネージャが、該第2のサブスクリプションに従って、第2の通知を該第2のパフォーマンス機器マネージャに送信することと、

該第2のパフォーマンス機器マネージャが、該変化した第1の状態変数を表示するように該第2のコントロール・パネル・ビューを更新することとを含む、方法。

【請求項2】

10

前記方法は、第3のパフォーマンス機器マネージャが、前記第1の状態変数に対して、前記第2のコントロール・パネル・ビューと該第3のパフォーマンス機器マネージャの第3のコントロール・パネル・ビューとの間で第3のサブスクリプションを確立することをさらに含む、請求項1に記載の方法。

【請求項3】

前記第2のパフォーマンス機器マネージャが、前記第1の状態変数の値が変化したことに応答して、前記第3のサブスクリプションに従って、前記変化した値を前記第3のパフォーマンス機器マネージャに通知することをさらに含む、請求項2に記載の方法。

【請求項4】

前記第2のパフォーマンス機器マネージャは、前記第1の状態変数の値が変化するという前記第2の通知に応答して、該第1の状態変数の値について前記第1のオーディオ関連デバイスに問い合わせる、請求項3に記載の方法。

20

【請求項5】

前記第1のコントロール・パネル・ビューは、前記第1の状態変数の値を変化させるように構成されたユーザ・インターフェースを含む、請求項1に記載の方法。

【請求項6】

前記第1のコントロール・パネル・ビューが前記第1の状態変数の値を変化させたことに応答して、前記第1のパフォーマンス機器マネージャが、該第1のコントロール・パネル・ビューが該第1の状態変数の値を変化させたことを前記第1のオーディオ関連デバイスに通知することをさらに含む、請求項5に記載の方法。

30

【請求項7】

前記第1のコントロール・パネル・ビューが前記第1の状態変数の値を変化させたという通知に応答して、前記第1のオーディオ関連デバイスが、該第1の状態変数の値について前記第1のパフォーマンス機器マネージャに問い合わせる、請求項6に記載の方法。

【請求項8】

前記方法は、第2のオーディオ関連デバイスが、該第2のオーディオ関連デバイスと前記第1のオーディオ関連デバイスの前記第1の状態変数との間で第3のサブスクリプションを確立することをさらに含む、該第2のオーディオ関連デバイスは、第2の状態変数を含む、請求項1に記載の方法。

【請求項9】

40

前記第1のオーディオ関連デバイスが、前記第1の状態変数の値が変化したことに応答して、前記第3のサブスクリプションに従って、前記第2のオーディオ関連デバイスに通知することをさらに含む、請求項8に記載の方法。

【請求項10】

前記第1のオーディオ関連デバイスが前記第2の状態変数に対して第4のサブスクリプションを確立した場合には、前記第2のオーディオ関連デバイスが、該第4のサブスクリプションに従って、該第2の状態変数に対する変化を該第1のオーディオ関連デバイスに通知することをさらに含む、請求項9に記載の方法。

【請求項11】

前記第1の状態変数の値についての前記第2のパフォーマンス機器マネージャによる前

50

記第 1 のオーディオ関連デバイスに対する問い合わせに回答して、前記第 1 のパフォーマンス機器マネージャが、前記第 1 のオーディオ関連デバイスの処理の利用可能性を決定することをさらに含む、請求項 1 に記載の方法。

【請求項 1 2】

前記第 1 のオーディオ関連デバイスが十分な処理の利用可能性を有していない場合には、前記問い合わせを前記第 1 のパフォーマンス機器マネージャから第 3 のパフォーマンス機器マネージャに転送することをさらに含む、該第 3 のパフォーマンス機器マネージャは、該第 1 のオーディオ関連デバイスが通信するように構成されている別のデバイスに含まれている、請求項 1 1 に記載の方法。

【請求項 1 3】

前記第 1 の状態変数に関する転送された問い合わせを前記第 1 のパフォーマンス機器マネージャから受信したことに回答して、前記第 3 のパフォーマンス機器マネージャが、該第 1 の状態変数の値と共に前記第 2 のパフォーマンス機器マネージャに回答することをさらに含む、請求項 1 2 に記載の方法。

【請求項 1 4】

前記第 1 のオーディオ関連デバイスおよび前記第 2 のオーディオ関連デバイスが、該第 1 のオーディオ関連デバイスおよび該第 2 のオーディオ関連デバイスのメモリにそれぞれ前記第 1 のコントロール・パネル・ビューおよび前記第 2 のコントロール・パネル・ビューを格納することをさらに含む、請求項 1 に記載の方法。

【請求項 1 5】

前記第 1 のオーディオ関連デバイスおよび前記第 2 のオーディオ関連デバイスが、該第 1 のオーディオ関連デバイスおよび該第 2 のオーディオ関連デバイスのメモリにそれぞれ前記第 1 の状態変数の値および前記第 2 の状態変数の値を格納することをさらに含む、請求項 8 に記載の方法。

【請求項 1 6】

前記第 1 のパフォーマンス機器マネージャが、前記第 1 の状態変数の値を調整するために、前記第 1 のコントロール・パネル・ビューのユーザ・インターフェースの第 1 のタイプのグラフィカル・ユーザ・インターフェース・エレメントと該第 1 のコントロール・パネル・ビューのユーザ・インターフェースの第 2 のタイプのグラフィカル・ユーザ・インターフェース・エレメントとの間でユーザが選択することを可能にすることをさらに含む、請求項 5 に記載の方法。

【請求項 1 7】

前記第 1 のコントロール・パネル・ビューは、前記第 1 の状態変数の値および前記第 2 の状態変数の値を変化させるように動作可能なユーザ・インターフェースを含む、請求項 8 に記載の方法。

【請求項 1 8】

前記第 1 のパフォーマンス機器マネージャが、前記第 1 の状態変数の値について前記第 1 のオーディオ関連デバイスに問い合わせることをさらに含む、請求項 1 に記載の方法。

【請求項 1 9】

前記第 1 の状態変数の値に対する変化を追跡することは、前記第 1 のオーディオ関連デバイスの処理パワーが制限されているために該第 1 の状態変数に対して該第 1 のオーディオ関連デバイスにサブスクライブされている第 2 のオーディオ関連デバイスに含まれている第 3 のパフォーマンス機器マネージャを用いて、該第 1 の状態変数に対する変化を追跡することを含み、該第 2 のオーディオ関連デバイスは、該第 1 のオーディオ関連デバイスよりも相対的に大きな処理パワーを有し、

前記第 1 のパフォーマンス機器マネージャに通知することは、該第 3 のパフォーマンス機器マネージャが、該第 1 の状態変数の値が変化したことを該第 1 のパフォーマンス機器マネージャに通知することを含む、請求項 1 に記載の方法。

【請求項 2 0】

オーディオ関連デバイス上の設定に対する変化を追跡するシステムであって、

10

20

30

40

50

該システムは、

第1のパフォーマンス機器マネージャと変化可能な値を含む状態変数とを含むように構成されたオーディオ関連デバイスと、

該オーディオ関連デバイスの第1のユーザ・インターフェースであって、該状態変数の該変化可能な値の表示を選択的に含むように構成された第1のコントロール・パネル・ビューを含む第1のユーザ・インターフェースと、

該オーディオ関連デバイスと通信する第2のデバイスであって、該第2のデバイスは、第2のパフォーマンス機器マネージャと第2のユーザ・インターフェースとを含み、該第2のユーザ・インターフェースは、該状態変数の該変化可能な値の表示を選択的に含むように構成された第2のコントロール・パネル・ビューを含む、第2のデバイスと

を備え、

該第1のコントロール・パネル・ビューは、該オーディオ関連デバイスにおける該状態変数に第1のサブスクリプションを介してサブスクライブされており、該第2のコントロール・パネル・ビューは、該第1のコントロール・パネル・ビューを用いて選択的に表示される該状態変数の該変化可能な値に第2のサブスクリプションを介してサブスクライブされており、

該変化可能な値に対する変化に関する第1の変化通知は、該第1のパフォーマンス機器マネージャを介して、該第1のサブスクリプションに従って該第1のコントロール・パネルに通信され、該第1のパフォーマンス機器マネージャは、該第1の変化通知にตอบสนองして、該第2のパフォーマンス機器マネージャを介して、該第2のサブスクリプションに従って該第2のコントロール・パネル・ビューに第2の変化通知を通信するように構成されており、該第2の変化通知は、該変化可能な値に対する変化を示す、システム。

【請求項21】

前記第1の変化通知は、前記変化可能な値が前記第1のコントロール・パネル・ビューに現在表示中であるというコンファメーションにตอบสนองして、前記オーディオ関連デバイスから該第1のコントロール・パネル・ビューに通信され、前記第2の変化通知は、該変化可能な値が前記第2のコントロール・パネル・ビューに現在表示中であるというコンファメーションにตอบสนองして、前記第1のパフォーマンス機器マネージャによって、該第2のコントロール・パネル・ビューに通信される、請求項20に記載のシステム。

【請求項22】

前記オーディオ関連デバイスは、第1のオーディオ関連デバイスであり、前記システムは、第2のオーディオ関連デバイスをさらに含み、該第2のオーディオ関連デバイスもまた、前記第1のコントロール・パネル・ビューを用いて選択的に表示される前記状態変数の前記変化可能な値に第3のサブスクリプションを介してサブスクライブされており、該変化可能な値に対する変化を示す第2の変化通知もまた、該第3のサブスクリプションに従って該第2のオーディオ関連デバイスに通信され、該第2の変化通知にตอบสนองして、該第2のオーディオ関連デバイスは、該変化可能な値の格納された値を更新するように構成されている、請求項20に記載のシステム。

【請求項23】

その他の複数のオーディオ関連デバイスが、前記第1のコントロール・パネル・ビューを用いて選択的に表示される前記状態変数の前記変化可能な値にそれぞれのサブスクリプションを介してサブスクライブされており、前記変化可能な値に対する変化を示す前記第2の変化通知もまた、前記第1のパフォーマンス機器マネージャによって、該それぞれのサブスクリプションに従って該その他の複数のオーディオ関連デバイスに通信される、請求項20に記載のシステム。

【請求項24】

前記第1のパフォーマンス機器マネージャは、前記第1の変化通知にตอบสนองして、前記変化可能な値に対する変化について前記オーディオ関連デバイスに問い合わせ、該変化可能な値に対する変化を前記第2のオーディオ関連デバイスに通信するようにさらに構成されている、請求項22に記載のシステム。

## 【請求項 25】

前記第2のパフォーマンス機器マネージャは、前記第2の変化通知に応答して、前記変化可能な値に対する変化について前記オーディオ関連デバイスに問い合わせるように構成されている、請求項20に記載のシステム。

## 【請求項 26】

前記第2のパフォーマンス機器マネージャは、前記第2の変化通知に応答して、前記変化可能な値に対する変化について前記第1のパフォーマンス機器マネージャに問い合わせるように構成されている、請求項20に記載のシステム。

## 【請求項 27】

オーディオ関連デバイス上の設定に対する変化を追跡するプログラムであって、該プログラムは、第1のパフォーマンス機器マネージャのメモリ装置に格納されており、

該プログラムは、

ディスプレイにおいてコントロール・パネル・ビューを生成する手順と、

状態変数に対して第1のサブスクリプションを介してサブスクライブする手順であって、該状態変数は、該オーディオ関連デバイスの変化可能なオーディオ・パラメータを含み、該変化可能なオーディオ・パラメータは、該オーディオ関連デバイスの動作中に、手で、または、自動で、または、論理的に更新されるように構成されている、手順と、

該第1のサブスクリプションに従って該オーディオ関連デバイスにおける該変化可能なオーディオ・パラメータに対する更新を示す第1の変化通知メッセージを受信する手順と

、  
該受信された変化可能なオーディオ・パラメータを表示するように該コントロール・パネル・ビューを更新する手順と、

第2のパフォーマンス機器マネージャを介して、該コントロール・パネル・ビューにおける該変化可能なオーディオ・パラメータに第2のサブスクリプションを介してサブスクライブされている別のオーディオ関連デバイスによって受信されるための第2の変化通知メッセージを生成し、該第2のサブスクリプションに従って、該第2の変化通知メッセージを該別のオーディオ関連デバイスに送信する手順であって、該第2の変化通知メッセージは、該受信された変化可能なオーディオ関連パラメータを表示するように該コントロール・パネル・ビューが更新されたことを示す、手順と

をプロセッサに実行させる、プログラム。

## 【請求項 28】

前記プログラムは、

前記受信された変化可能なオーディオ・パラメータに対する別の変化を前記別のオーディオ関連デバイスに通知する第3の変化通知メッセージであって、該別の変化は、該コントロール・パネル・ビューを用いて引き起こされたものである、第3の変化通知メッセージと、

該オーディオ関連デバイスから第3の問い合わせを受信したことに応答して、該オーディオ関連デバイスによって受信されるために、該受信された変化可能なオーディオ・パラメータに対する該別の変化を送信するためのコンピュータ・コードと

を生成する手順をプロセッサにさらに実行させる、請求項27に記載のプログラム。

## 【請求項 29】

前記プログラムは、

前記コントロール・パネル・ビュー内の前記変化可能なオーディオ・パラメータに第3のサブスクリプションを介してサブスクライブされている別のコントロール・パネル・ビューによって受信されるための第3の変化通知メッセージであって、該第3の変化通知メッセージは、前記受信された変化可能なオーディオ・パラメータを表示するように該コントロール・パネル・ビューが更新されたことを示す、第3の通知メッセージと、

第3の問い合わせを受信したことに応答して、該別のコントロール・パネル・ビューによって受信されるために、該受信された変化可能なオーディオ・パラメータを送信するためのコンピュータ・コードと

10

20

30

40

50

を生成し、該第3のサブスクリプションに従って、該第3の変化通知メッセージを該別のコントロール・パネル・ビューに送信する手順をプロセッサにさらに実行させる、請求項27に記載のプログラム。

【請求項30】

前記プログラムは、

前記第1の変化通知メッセージの受信にตอบสนองして、前記オーディオ関連デバイスから前記変化可能なオーディオ・パラメータを送信することを要求する第1の問い合わせと、

前記第1の問い合わせにตอบสนองして、該変化可能なオーディオ・パラメータを受信するためのコンピュータ・コードと

を生成する手順をプロセッサにさらに実行させる、請求項27に記載のプログラム。

10

【請求項31】

前記プログラムは、

前記別のオーディオ関連デバイスによって受信されるために、前記コントロール・パネル・ビューから前記受信された変化可能なオーディオ・パラメータを送信するように、該別のオーディオ関連デバイスから第2の問い合わせを受信したことにตอบสนองする手順をプロセッサにさらに実行させる、請求項30に記載のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、オーディオ・システムに関し、より詳細には、オーディオ関連の機器の管理のためのシステムに関する。

20

【背景技術】

【0002】

オーディオ信号を増幅するためにオーディオ・システムを使用することは周知である。一般的に、オーディオ・システムは、増幅器を用いてオーディオ信号を増幅し、1つ以上のスピーカを駆動するために動作する。このようなオーディオ・システムは、小売衣料品店で用いられるといった2つ、3つのコンポーネントを有する比較的シンプルな構成から、コンサートホールに用いられるといった大きく複雑なシステムに及び得る。

【0003】

オーディオ・システムの構成は通常、オーディオ・システムを作り上げるデバイスの物理的配置および配線を伴う。加えて、デバイスのいくつか(例えば、増幅器)は、一般的に、ユーザによる構成可能な設定を含む。この設定は、所望のオーディオ・システムの性能を実現するためにユーザの好みで構成され得る。オーディオ・システムのサイズおよび複雑さが増すにつれて、配線および設定の数もまた増加する。加えて、オーディオ・システムにおけるデバイスは、地理的に、より分散し得る。従って、システムの操作、メンテナンス、およびトラブルシューティングは、より困難になる。このことは、特に、ツアー中の音楽グループの場合といった、さまざまな構成で繰り返し分解され、再構築されるオーディオ・システムに当てはまる。

30

【0004】

加えて、いくつかのオーディオ関連デバイスを包含するオーディオ・システムをセットアップし、かつ、制御するために用いられるグラフィカル・ユーザ・インターフェース(GUI)は、すぐに用いるために複雑で、難しくなり得る。このシステムにおける特定のオーディオ関連デバイスを構成するために、例えば、ユーザは、所望のデバイスに対する制御を見つけ出す前に、相当大きな階層をトラバースすることを必要とされるかもしれない。加えて、オーディオ関連デバイスとGUIにおけるコントロール・パネル・ビューとの間の通信は、高いネットワーク・トラフィックを有する傾向がある。いくつかの場合では、急激な変化は、他のデバイスに対する通信を遅くするネットワーク上の障害を生み出し得る。

40

【発明の開示】

【課題を解決するための手段】

50

## 【 0 0 0 5 】

( 要 約 )

本発明は、以下を提供する：

( 項 目 1 )

オーディオ関連デバイス上の設定に対する変化を追跡するための方法であって、

第 1 の状態変数を有する第 1 のオーディオ関連デバイスを提供することであって、該第 1 のオーディオ関連デバイスは、該第 1 の状態変数の値を変化させるように動作可能なユーザ・インターフェースを含む、ことと、

該第 1 のオーディオ関連デバイスに関連した第 1 のコントロール・パネル・ビューを構築することであって、該第 1 のコントロール・パネル・ビューは、該第 1 の状態変数を表示するように動作可能である、ことと、

該第 1 の状態変数の該値に対する変化を追跡することと、

該第 1 の状態変数の該値における変化にตอบสนองして、該第 1 の状態変数の該値が変化したことを該第 1 のコントロール・パネル・ビューに通知することと、

該第 1 の状態変数の該値が変化した通知にตอบสนองして、該第 1 のコントロール・パネル・ビューは、該第 1 の状態変数の該値に対して該第 1 のオーディオ関連デバイスにクエリすることと

を包含する、方法。

## 【 0 0 0 6 】

( 項 目 2 )

第 2 のコントロール・パネル・ビューと前記第 1 の状態変数との間でサブスクリプションを確立することをさらに包含する、項目 1 に記載の方法。

## 【 0 0 0 7 】

( 項 目 3 )

前記第 1 の状態変数の前記値における変化にตอบสนองして、前記第 2 のコントロール・パネル・ビューに通知することをさらに包含する、項目 2 に記載の方法。

## 【 0 0 0 8 】

( 項 目 4 )

前記第 2 のコントロール・パネル・ビューは、前記第 1 の状態変数の前記値が変化した通知にตอบสนองして、該第 1 の状態変数の該値に対して前記第 1 のオーディオ関連デバイスにクエリする、項目 3 に記載の方法。

## 【 0 0 0 9 】

( 項 目 5 )

前記第 1 のコントロール・パネル・ビューは、前記第 1 の状態変数の前記値を変化させるように適合されたユーザ・インターフェースを包含する、項目 1 に記載の方法。

## 【 0 0 1 0 】

( 項 目 6 )

前記第 1 のコントロール・パネル・ビューによる前記状態変数の前記値における変化にตอบสนองして、該状態変数の該値は、該第 1 のコントロール・パネル・ビューによって変化されたことを前記第 1 のオーディオ関連デバイスに通知することをさらに包含する、項目 5 に記載の方法。

## 【 0 0 1 1 】

( 項 目 7 )

前記第 1 のコントロール・パネル・ビューが、前記第 1 の状態変数の前記値を変化させた通知にตอบสนองして、前記第 1 のオーディオ関連デバイスは、該第 1 の状態変数の該値に対して該第 1 のコントロール・パネル・ビューにクエリする、項目 6 に記載の方法。

## 【 0 0 1 2 】

( 項 目 8 )

第 2 のオーディオ関連デバイスと前記第 1 のオーディオ関連デバイスの前記第 1 の状態変数との間でサブスクリプションを確立することをさらに包含し、該第 2 のオーディオ関

10

20

30

40

50

連デバイスは、第 2 の状態変数を包含する、項目 1 に記載の方法。

【 0 0 1 3 】

( 項目 9 )

前記第 1 の状態変数の前記値における変化に応答して、前記第 2 のオーディオ関連デバイスに通知することをさらに包含する、項目 8 に記載の方法。

【 0 0 1 4 】

( 項目 1 0 )

前記第 1 のオーディオ関連デバイスが、前記第 2 の状態変数に対すサブスクリプションを確立したとき、該第 2 の状態変数に対する変化を該第 1 のオーディオ関連デバイスに通知することをさらに包含する、項目 9 に記載の方法。

10

【 0 0 1 5 】

( 項目 1 1 )

前記第 1 の状態変数の該値に対する前記第 1 のコントロール・パネル・ビューの前記クエリに応答して、前記第 1 のオーディオ関連デバイスの処理の利用可能性を決定することをさらに包含する、項目 1 に記載の方法。

【 0 0 1 6 】

( 項目 1 2 )

前記第 1 のオーディオ関連デバイスが十分な処理の利用可能性を有しないとき、該第 1 のオーディオ関連デバイスが通信するように構成されている別のデバイスに前記第 1 のコントロール・パネル・ビューからの前記クエリを転送することをさらに包含する、項目 1

20

1 に記載の方法。

【 0 0 1 7 】

( 項目 1 3 )

前記第 1 のオーディオ関連デバイスからの前記第 1 の状態変数に関する転送されたクエリを受け取ることに応答して、該第 1 の状態変数の前記値を有する前記第 1 のコントロール・パネル・ビューに回答することをさらに包含する、項目 1 2 に記載の方法。

【 0 0 1 8 】

( 項目 1 4 )

メモリに前記第 1 のコントロール・パネル・ビューと前記第 2 のコントロール・パネル・ビューとを格納することをさらに包含する、項目 8 に記載の方法。

30

【 0 0 1 9 】

( 項目 1 5 )

メモリに前記第 1 の状態変数の前記値と前記第 2 の状態変数の前記値とを格納することをさらに包含する、項目 8 に記載の方法。

【 0 0 2 0 】

( 項目 1 6 )

前記第 1 の状態変数の前記値を調整するために、前記第 1 のコントロール・パネル・ビューの前記ユーザ・インターフェースのグラフィカル・ユーザ・インターフェース・エレメントの第 1 のタイプと該第 1 のコントロール・パネル・ビューの該グラフィカル・ユーザ・インターフェース・エレメントの第 2 のタイプとの間でユーザが選択することを可能

40

にすることをさらに包含する、項目 5 に記載の方法。

【 0 0 2 1 】

( 項目 1 7 )

前記第 1 のコントロール・パネル・ビューは、前記第 1 の状態変数の前記値および前記第 2 の状態変数の前記値を変化させるように動作可能なユーザ・インターフェースを包含する、項目 8 に記載の方法。

【 0 0 2 2 】

( 項目 1 8 )

G U I および選択デバイスを有するオーディオ関連デバイスのディスプレイを提供し、かつ、該ディスプレイから選択するための方法であって、

50

該グラフィカル・ユーザ・インターフェースが、ベニュー・ビューまたはプロダクト・ビュー内にあるか否かを決定することと、

該ベニュー・ビュー内にあるとき、ネットワークに接続されている各オーディオ関連デバイスを表している仮想デバイスのリストを取り出し、該ディスプレイ上で該仮想デバイスのリストを表示することと、または、

該プロダクト・ビュー内にあるとき、オブジェクトに対する状態変数のリストを取り出し、該ディスプレイ上で該状態変数のリスト表示することとを包含する、方法。

【 0 0 2 3 】

( 項目 1 9 )

前記ベニュー・ビュー内にあるとき、前記仮想デバイスのリストから、選択された仮想デバイスを指し示す前記選択デバイスを示している選択信号を受け取ることとをさらに包含する、項目 1 8 に記載の方法。

【 0 0 2 4 】

( 項目 2 0 )

前記選択された仮想デバイスを選択する前記信号に応答して、該選択された仮想デバイスに対するオブジェクトのリストを取り出すことと、該選択された仮想デバイスに対する該オブジェクトのリストを表示することとをさらに包含する、項目 1 9 に記載の方法。

【 0 0 2 5 】

( 項目 2 1 )

前記オブジェクトのリストからの、選択されたオブジェクトを指し示している前記選択デバイスを示している選択信号を取り出すことと、該選択されたオブジェクトを選択する該信号に応答して、該選択されたオブジェクトに対する状態変数のリストを取り出し、該選択されたオブジェクトに対する該状態変数を表示することとをさらに包含する、項目 2 0 に記載の方法。

【 0 0 2 6 】

( 項目 2 2 )

単一のグラフィカル・ユーザ・インターフェース・エレメントを用いて、前記状態変数のリストからの第 1 の状態変数と第 2 の状態変数とを制御することとをさらに包含する、項目 2 1 に記載の方法。

【 0 0 2 7 】

( 項目 2 3 )

前記第 1 の状態変数および前記第 2 の状態変数は、異なるデータ・タイプを包含する、項目 2 2 に記載の方法。

【 0 0 2 8 】

( 項目 2 4 )

前記グラフィカル・ユーザ・インターフェース・エレメントは、スライダ、ボタン、ノブ、ボタン、ロータリ・エンコーダおよびリスト・ボックスからなるグループから選択される、項目 2 3 に記載の方法。

【 0 0 2 9 】

( 項目 2 5 )

選択されたオブジェクトに対する状態変数のリストを取り出すことと、該選択されたオブジェクトに関連した信号に応答して、該選択されたオブジェクトに対する該状態変数のリストを表示することとをさらに包含する、項目 1 8 に記載の方法。

【 0 0 3 0 】

( 項目 2 6 )

前記プロダクト・ビュー内にあるとき、メニューに対する前記状態変数に関連したオブジェクトのリストを取り出すことと、該メニューを表示することとをさらに包含する、項目 1 8 に記載の方法。

【 0 0 3 1 】

10

20

30

40

50

(項目27)

オーディオ関連デバイスとともに用いられるパフォーマンス機器マネージャであって、複数のプラグインであって、該複数のプラグインのそれぞれは、オーディオ関連デバイスに関連した、複数のプラグインと、

該オーディオ関連デバイスのノード構造にクエリするように構成されているGUIのコアと、

該GUIのコアにおける該複数のプラグインの間で通信するように構成されているプラグイン・インターフェースと、

該オーディオ関連デバイスと通信するよう構成されているネットワーク・インターフェースと

を備え、

該複数のプラグインは、該プラグイン・インターフェースとの通信をインスタンス化するように構成されている、パフォーマンス機器マネージャ。

【0032】

(項目28)

前記GUIのコアは、前記オーディオ関連デバイスとコントロール・パネル・ビューとの間のサブスクリプションを管理するように構成されているリンク・マネージャを備えた、項目27に記載のパフォーマンス機器マネージャ。

【0033】

(項目29)

前記リンク・マネージャは、前記オーディオ関連デバイスの状態変数における変化を追跡するように構成されているイベント・エンジンを備えた、項目28に記載のパフォーマンス機器マネージャ。

【0034】

(項目30)

前記イベント・エンジンによって決定される前記状態変数における変化に応答して、前記リンク・マネージャは、前記コントロール・パネル・ビューに通知するように構成されているリンク・エンジンを備えた、項目29に記載のパフォーマンス機器マネージャ。

【0035】

本発明は、オーディオ関連機器の管理および制御のためのシステムを提供する。このシステムは、オーディオ・システムにおけるオーディオ関連機器の全てを管理、モニタ、および制御する能力を提供する共通のアーキテクチャである。従って、マイクロフォンから増幅器、ミキサー、照明装置にわたるすべてのものが、このシステムによってアクセスされ、制御され得る。

【0036】

このシステムは、さまざまな物理ネットワークおよび通信プラットフォームのいずれかを用いて、種々の複雑さを有する複数の異なるオーディオ関連デバイスをシームレスに統合するネットワーク独立プロトコル(network independent protocol)を提供する。加えて、このシステムは、未知の特定のオーディオ関連デバイスのアスペクトを発見するメッセージング機能を提供することによって、システム内のオーディオ関連デバイスのアップグレード、変更、追加が可能である。このメッセージング機能はまた、複数のオーディオ関連デバイス間でオーディオデータおよび関連情報が効率的に、および最小のネットワーク・トラフィックで送達されることを可能にする。

【0037】

1つ以上のパフォーマンス機器マネージャがこのシステムに含まれる。パフォーマンス機器マネージャは、オーディオ・システム全体のコントロール・パネル・ビューを取り出しおよび/または生成する。加えて、パフォーマンス機器マネージャは、オーディオ・システム内の各特定のデバイスに対してコントロール・パネル・ビューを取り出しおよび/または生成し得る。コントロール・パネル・ビューは、オーディオ・システム内のオーディオ関連デバイスの協調された管理、制御およびモニタを可能にする。このシステム内で

10

20

30

40

50

用いられるに通信プロトコルは、デバイス・ツー・デバイス通信と同様に、パフォーマンス機器マネージャとオーディオ関連デバイス間の標準通信を可能にする。

【0038】

このシステムには、リンク・マネージャが含まれている。このリンク・マネージャは、他の複数のオーディオ関連システムまたはコントロール・パネル・ビューへのサブスクリプションを管理するように構成され得る。管理されているデータ変数が変化するとき、リンク・マネージャ・モジュールは、変化を他のオーディオ関連デバイスおよび/またはコントロール・パネル・ビューに通知し得る。この通知にตอบสนองして、他のオーディオ関連デバイスおよびコントロール・パネル・ビューは、変化したデータ変数を有するオーディオ関連デバイスにサブスクライブし、アップデータされたデータ変数の受け取りを要求し得る。いくつかの例において、このリンク・マネージャはまた、特定のオーディオ関連デバイスに対してサブスクリプションの負荷を減らすように構成されている。従って、このリンクマネージャを用いることは、ネットトラフィックを減少させる。さらに、サブスクリプションの負荷の分配がデータをアップデートする速度を増加させる。

10

【0039】

オーディオ関連デバイスを制御するために用いられるGUIは、ベニュー・ビューとプロダクト・ビューとの間で切り替え得る。ベニュー・ビューにおいては、コントロール・パネル・ビューのGUIは、オーディオ関連デバイスにおいて、ユーザが同時に複数のデバイスを見ることを可能にする。プロダクト・ビューにおいては、単一のデバイスに対するコントロール・パネル・ビューのGUIが表示され得る。ユーザは、必要に応じて、これらのビューの間で切り替えることが可能である。例えば、ユーザは、タブレット・コンピュータ上の見られるといったタッチ・スクリーン・インターフェースを用いるとき、プロダクト・ビューを用い、マウスおよびキーボードを有するコンピュータを用いるとき、ベニュー・ビューを用い得る。

20

【0040】

本発明の他のシステム、方法、特徴および利点が、以下の図面および詳細な説明を検討することにより当業者に明らかになる。すべてのこのような追加的なシステム、方法、特徴および利点がこの記載に含まれ、本発明の範囲内であることが意図される。

【発明の効果】

【0041】

本発明によって、システムの操作、メンテナンス、およびトラブルシューティングが、より容易になる。このことは、特に、ツアー中の音楽グループの場合といった、さまざまな構成で繰り返し分解され、再構築されるオーディオ・システムにおいて有用である。また、いくつかのオーディオ関連デバイスを包含するオーディオ・システムをセットアップし、かつ、制御するために用いられるグラフィカル・ユーザ・インターフェース(GUI)が、すぐに用いるために単純で、簡単になる。加えて、オーディオ関連デバイスとGUIにおけるコントロール・パネル・ビューとの間の通信は、高いネットワーク・トラフィックを有する傾向があるが、急激な変化は、他のデバイスに対する通信を遅くするネットワーク上の障害を生み出し得るいくつかの場合、本発明によってこのような障害が解決され得る。

30

40

【発明を実施するための最良の形態】

【0042】

I. システム

図1は、コンサートホールといった場所でのミュージックコンサートといった公演のために構成された複数のオーディオ関連デバイスを有するオーディオ・システム100の例である。他の例において、オーディオ・システム100は、例えば、遊園地に恒久的に備え付けられたサウンドシステム、および例えばスタジアムのロックコンサートのために一時的に備え付けられたサウンドシステムを含む他の任意のタイプの場所での他の任意のタイプの公演のために構成され得る。この例のオーディオ・システム100に含まれるオーディオ関連デバイスは、ステージ102、オーディエンスエリア104、第1のルーム1

50

06および第2のルーム108を含む場所に備え付けられたものとして表される。

【0043】

オーディオ・システム100は、複数のスピーカ116から可聴音を生成するために処理されるオーディオ信号を提供するギター112およびマイクロフォン114を含む。他の例において、オーディオ信号は、他の任意のタイプのデバイスによって生成され得る。ギター112によって生成されるオーディオ信号は、オーディオ信号プリプロセッサ118によって処理され、オーディオ・システム100にまた含まれるオーディオ・ネットワーク120に提供され得る。オーディオ・ネットワーク120は、オーディオ信号およびオーディオ・システムに関連した他の任意の信号を運ぶことが可能である任意の形状の通信ハイウェイであり得る。オーディオ・ネットワーク120は、Ethernet（登録商標）、USB、RS232、RS485、FireWire、または他の任意のプロトコルといった複数のプロトコルのいずれかにおいて光ファイバ、有線、および/または無線通信能力を含み得る。

10

【0044】

マイクロフォン114からのオーディオ信号はまた、オーディオ・ネットワーク120に提供され得る。この図解された例において、オーディオ信号は、ミキサー124によって受け取られ、処理され得る。ミキサー124による処理は、フィルタ、コンプレッサ、イコライザ、エフェクト、利得、またはミキサー124に含まれる他の任意の設定といった構成可能な設定に基付き得る。ミキサー124による処理に続いて、処理されたオーディオ信号は、第1のオーディオ・プロセッサ126および第2のオーディオ・プロセッサ128に含まれる構成可能なセットを用いて更なる処理のために第1のオーディオ・プロセッサ126および第2のオーディオ・プロセッサ128によって受け取られ得る。第1のオーディオ・プロセッサ126および第2のオーディオ・プロセッサ128は、オーディオ信号に更なるエフェクトを提供する。第1のオーディオ・プロセッサ126および第2のオーディオ・プロセッサ128によって処理されるオーディオ信号は、第1の増幅器132、第2の増幅器134、第3の増幅器136および第4の増幅器138によって受け取られ得る。利得およびイコライザ（「EQ」）設定といった増幅器132、134、136、138における追加的な構成可能な設定は、オーディオ信号のさらなる処理および増幅に適用され得る。増幅され、処理されたオーディオ信号は、スピーカ116のそれぞれを駆動し得る。

20

30

【0045】

オーディオ信号はまた、第5の増幅器142によって受け取られ、処理され、かつ、第2のルーム108に含まれるスピーカ116のそれぞれを駆動するために用いられ得る。ウォール・コントローラ146は、オーディオ・ネットワーク120上で第5の増幅器142を制御するために用いられ得る。加えて、ウォール・コントローラ146は、オーディオ・システム100の動作に関連した1つ以上のパラメータであって、オーディオ・ネットワーク120上で受け取られる1つ以上のパラメータを表示することが可能であるディスプレイを含み得る。

【0046】

第1のPEM150はまた、オーディオ・システム100に含まれ得る。第1のPEM150は、メモリデバイスに格納され、オーディオ関連デバイスといった任意の形式のコンピュータデバイスにおいて実行され得る。このオーディオ・システムの例において、第1のPEM150は、GUIを有するパーソナルコンピュータ（「PC」）で実行されている。他の例において、第1のPEM150の動作が、1つ以上のGUIを有する複数のコンピュータデバイスのメモリデバイス間に配信され得る。第1のPEM150は、GUIを用いて、1つ以上のコントロール・パネル・ビューにおけるオーディオ・システム100のさまざまなオーディオ関連デバイスおよびアスペクトを、選択的に表示する能力を含み得る。あるいは、複数の並列動作するPEMは、オーディオ・システム100の所定の部分に存在するさまざまなオーディオ関連デバイスおよび機能性のコントロール・パネル・ビューを選択的に表示するために用いられ得る。従って、大きく複雑なオーディオ・シ

40

50

システムにおいて、オーディオ・システム 100 は、複数の P E M が、オーディオ・システム 100 内で異なるサブシステムとして利用され得るように分けられ得る。

【 0 0 4 7 】

ここで用いられるものとして、用語「コントロール・パネル・ビュー」または「種々のコントロール・パネル・ビュー」は、ディスプレイ上で地理的に生成されたユーザ・インターフェース、コントロール面といったハードウェアベースのユーザ・インターフェース、および/または地理的に生成されたユーザ・インターフェースならびにハードウェアベースのユーザ・インターフェースの組み合わせとして定義される。

【 0 0 4 8 】

G U I の形式のシステム・ベニュー・ビュー・ディスプレイであるコントロール・パネル・ビューは、第 1 の P E M 1 5 0 によって生成され得る。システム・ベニュー・ビュー・ディスプレイは、オーディオ・システム 100 のオーディオ関連デバイスの構成、I D、動作パラメータのシステム特定の描写であり得る。オーディオ・システム 100 のオーディオ関連デバイスは、第 1 の P E M 1 5 0 によって発見され、かつ、システム・ベニュー・ビュー・ディスプレイに含まれ得る。加えて、第 1 の P E M 1 5 0 は、オーディオ・ネットワーク 1 2 0 を用いてオーディオ・システム 100 のオーディオ関連デバイスを個々に管理、モニタ、制御し得る。オーディオ関連デバイスの個々の管理は、G U I のコントロール・パネル・ビューである複数のコントロール・パネル・ビューから行われ得る。

10

【 0 0 4 9 】

オーディオ関連デバイスのそれぞれは、それぞれのオーディオ関連デバイスに対応する 1 つ以上の G U I のコントロール・パネル・ビューを用いて管理され得る。1 つ以上の G U I のコントロール・パネル・ビューは、P E M 1 5 0 によって生成され、ディスプレイに表示され得る。コントロール・パネル・ビューはまた、P E M 1 5 0 に含まれるコントロール面であり得る。ここで用いられるものとして、用語「コントロール面」は、ボタン、ノブ、計測器 ( m e t e r )、L E D、スライダ、インジケータ、数値ディスプレイ、または、オーディオ関連デバイスに関連した他の任意のハードウェアベースのユーザ・インターフェースといったハードウェアを指す。

20

【 0 0 5 0 】

P E M 1 5 0 は、オーディオ・ネットワーク 1 2 0 上でオーディオ関連デバイスによって提供される情報に基づいて、対応するオーディオ関連デバイスのための格納されている所定のコントロール・パネル・ビューを識別し、かつ、用い得る。このような情報は、オーディオ関連デバイスのオーディオ・ノード構造を含み得る。あるいは、P E M 1 5 0 は、格納されている所定のコントロール・パネル・ビュー、コントロール・パネル・ビューの格納されている所定のアスペクト、および/またはコントロール・パネル・ビューを生成するために P E 1 5 0 によって生成されるコントロール・パネル・ビューのアスペクトを含む生成されたコントロール・パネル・ビューを用い得る。生成されたコントロール・パネル・ビューがオーディオ・ネットワーク 1 2 0 上で対応するオーディオ関連デバイスによって提供されるオーディオ・ノード構造といった情報に基づいて形成され得る。

30

【 0 0 5 1 】

オーディオ関連デバイスのオーディオ・ノード構造に対応する格納されている所定の G U I のコントロール・パネル・ビューが同様に識別され、オーディオ関連デバイスの機能性を表示するために用いられ得る。加えて、格納されている所定の G U I のコントロール・パネル・ビュー、格納されている所定の G U I のアスペクト、および/または P E M 1 5 0 によって生成される G U I のアスペクトを含む生成された G U I のコントロール・パネル・ビューは、G U I のコントロール・パネル・ビューを形成するために用いられ得る。生成された G U I のコントロール・パネル・ビューは、オーディオ・ネットワーク 1 2 0 上で対応するオーディオ関連デバイスによって提供されるオーディオ・ノード構造といった情報に基づいて同様に形成され得る。ここで用いられるものとして、用語「G U I のアスペクト」は、オーディオ関連デバイスのモニタリング、構成、および/または制御を提供するそれぞれの特徴であって、G U I に表示され得るそれぞれの特徴として定義され

40

50

る。加えて、用語「コンポーネント」は、コントロール・パネル・ビューに含まれているさまざまな関連したユーザ・インターフェースを指し、一方で用語「アспект」は、コントロール・パネル・ビューに統合され得るスタンド・アロン・ユーザ・インターフェースを指す。

【 0 0 5 2 】

従って、格納されている所定の GUI のコントロール・パネル・ビューは、GUI のコントロール・パネル・ビューにおいて前もって開発され、視覚的に配置され、かつ相互関連する複数の GUI コンポーネントを含み得る。GUI コンポーネントは、関連したオーディオ関連デバイスの機能性の少なくとも一部分を表し得る。格納されている所定の GUI のコントロール・パネル・ビューのコンポーネントの例は、ミキサーにおける多数のオーディオ・チャンネルのそれぞれを示している。表されているオーディオ・チャンネルのそれぞれは、ボリュームを制御するスライダ、スライダの位置を（デシベルで）示すボリューム・インジケータ、および入力信号を（デシベルで）示すオーディオ信号インジケータを含み得る。この例において、スライダおよびボリューム・インジケータは、スライダが動くとき、ボリューム・インジケータがボリュームの変化する大きさを反映するように相互に関連している。

10

【 0 0 5 3 】

格納されている所定の GUI のコントロール・パネル・ビューに含まれていない格納されている所定の GUI のアспектは、PEM 150 によって、現存の格納されている所定の GUI のコントロール・パネル・ビューに追加され得る視覚的 / 制御的特徴を含む。これらの視覚的 / 制御的特徴は、対応するオーディオ関連デバイスの機能性を表す GUI のコントロール・パネル・ビューを形成するために追加され得る。このような追加された所定の GUI のアспектはまた、格納されている所定の GUI のコントロール・パネル・ビューに含まれる GUI コンポーネントに相互に関連し得る。PEM 150 はまた、オーディオ関連デバイスによって提供されるオーディオ・ノード構造といった情報に基づいて、格納されている所定の GUI のアспектの 1 つ以上から GUI のコントロール・パネル・ビューを形成し得る。オーディオ関連デバイスが、関連した格納されている所定の GUI のコントロール・パネル・ビューを有していないとき、GUI のコントロール・パネル・ビューが形成され得る。この場合には、所定の GUI のアспектはまた、形成される GUI のコントロール・パネル・ビュー内で相互に関連し得る。

20

30

【 0 0 5 4 】

格納されている所定の GUI のコントロール・パネル・ビュー、または特定のオーディオ関連デバイスに対する GUI のコントロール・パネル・ビューに格納されている所定の GUI のアспектがないとき、PEM 150 は、対応するオーディオ関連デバイスによってオーディオ・ネットワーク上で提供される情報に基づいて、GUI のアспектおよび GUI 全体のコントロール・パネル・ビューを生成し得る。GUI のアспектは、同様に相互に関連し得る。

【 0 0 5 5 】

従って、ネットワーク構成のために、PEM 150 は、オーディオ・システム 100 全体および個々のオーディオ関連デバイスのエンド・ツー・エンド動作を管理するために構成され得る。トラブルシューティング、オーディオ・システムならびにオーディオ関連デバイス設定の格納、ネットワーク管理、およびネットワーク通信管理もまた、第 1 の PEM を用いて行われ得る。

40

【 0 0 5 6 】

ミキサー 124 は、第 2 の PEM 152 を含み得る。第 2 の PEM 152 の動作は、モニタリング、制御および格納機能性を含む第 1 の PEM 150 と同様に構成され得る。加えて、また同様の機能性を有する第 3 の PEM 154 は、オフィスビルといったオフサイト・ロケーション 160 に位置し得る。

【 0 0 5 7 】

PEM 150、152、154 の機能性は、その PEM 150、152、154 が動作

50

しているそれぞれのデバイスの処理能力に基づいて構成され得る。加えて、PEM150、152、154は、所望のレベルの機能性および特徴を提供するように構成され得る。例えば、オフサイト・ロケーション160に位置する第3のPEM154は、監視の役割を行うGUIを有するコンピュータといったコンピュータデバイスにおいて動作し得る。従って、エラー・ロギング、アラームおよびシステム動作は、第3のPEM154を用いて管理され得る。監視の役割のパフォーマンスをサポートする第3のPEM154を用いて表示されるGUIのコントロール・パネル・ビューが、ライブラリからアクセスされ得、および/または第1のPEM150に対して上述されるようなオーディオ関連デバイスとの通信に基づいて生成され得る。第3のPEM154はまた、オーディオ・システム100内で、遠隔なテクニカルおよびトラブルシューティング・アシスタンスを提供するために、個々のオーディオ関連デバイスセットをビューし、かつ、調整する能力を有して構成され得る。

10

#### 【0058】

第2のPEM152は、ミキサー124のユーザ・インターフェースおよび処理能力をとミキサー124の機能性とシェアし得る。従って、処理といった演算能力、データ格納および/または表示能力がミキサー124において、より低いとき、第2のPEM152は縮退した機能性を有し得る。第2のPEM152は、その第2のPEM152が動作しているオーディオ関連デバイスの動作能力を決定するように構成され得る。

#### 【0059】

第2のPEM152の機能性は、ミキサー124の決定された動作能力に一致し得る。従って、ミキサー124が、通信ポート、および/またはそれ自身の動作パラメータならびに設定の表示を通信および/または制限するメモリ能力、またはオーディオ・システム100における他のオーディオ関連デバイスを有するとき、第2のPEM152は、比較的、より小さく、より複雑でないコントロール・パネル・ビューを用い得る。加えて、第2のPEM152は、オーディオ関連デバイスの動作パラメータおよび設定の、より小さい部分を要求し得る。例えば、オーディオ関連デバイスが2つのチャンネルオーディオ増幅器であるとき、第2のPEM152は、1つのチャンネルだけを表示し、表示されたチャンネルに関連した動作パラメータおよび設定を取得し得る。第2のチャンネルが要求されるとき、第2のPEM152は、第2のチャンネルおよび関連した動作パラメータならびに設定だけを表示し得る。

20

30

#### 【0060】

##### II. オーディオ関連デバイス

図2は、オーディオ・ネットワーク120に結合されているオーディオ関連デバイス200の例のブロック図である。オーディオ関連デバイス200は、ミキサー、増幅器、マイクロフォン、照明パネル、ウォール・コントローラ、またはオーディオ・サウンド・システムおよび/またはオーディオ性能に関連した他の任意のデバイスであり得る。オーディオ関連デバイス200は、通信ポート202、プロセッサ204、メモリ206、オーディオI/O208、およびユーザ・インターフェース210を含む。オプション的に、オーディオ関連デバイス200は、外部ユーザ・インターフェース・ポート212を含み得る。他の例において、より多くの機能性が、オーディオ関連デバイス200に示され得る。加えて、より少ない、またはより多くのブロックがオーディオ関連デバイス200の機能性を記述するために用いられ得る。

40

#### 【0061】

通信ポート202は、オーディオ・ネットワーク120上で通信を可能にするハードウェアおよび/またはソフトウェアの任意の構成であり得る。プロセッサ204は、オーディオ関連デバイス200内で、全体的な機能性を管理し、計算を行い、かつ、入力/出力を制御する等、命令、論理タスクおよび/または動作コードを実行することが可能な任意のデバイスであり得る。プロセッサ204はまた、通信ポート202の動作を制御し得る。

#### 【0062】

50

メモリ206は、プロセッサ204によって実行可能な命令、オーディオ関連デバイス200の設定、(後述される)PEMによって使用可能なオーディオ・ノード構造、動作パラメータ、セットポイント、およびオーディオ関連デバイス200ならびにオーディオ・システム100(図1)における動作に関する他のデータまたは情報を格納するように構成されている揮発性および/または不揮発性メモリデバイスであり得る。メモリ206は、実行のために取り出される命令といった取り出されるデータを保存する媒体である。それは、ハードディスクドライブ、CDドライブ、DVDドライブ、ランダムアクセスメモリ(「RAM」)、読み出し専用メモリ(「ROM」)、プログラマブルROM(「PROM」)、電氣的消去可能ROM(「EEPROM」)、フラッシュメモリ、または他の任意のデジタル格納デバイスを含み得る。

10

#### 【0063】

オーディオI/Oは、オーディオ関連デバイス200に直接的に提供される任意の入力および出力であり得る。例えば、オーディオ増幅器は、処理され、増幅され、かつ、スピーカを駆動するために用いられる出力に送達されるオーディオ入力を有し得る。ユーザ・インターフェース210は、ボタン、スライダ、ノブ、タッチ・スクリーン、またはユーザ・コマンドが、オーディオ関連デバイス200に提供されることを可能にする他の任意の形式のインターフェースを含み得る。加えて、ユーザ・インターフェース210は、オーディオ関連デバイス200のユーザによって受け取られるための任意の形式のオーディオおよび/または視覚的出力を含み得る。オーディオおよび/または視覚的な出力の例は、発光ダイオード(「LED」)、計測器、液晶ディスプレイ(「LCD」)といったディスプレイ、もしくはオーディオ関連デバイス210の状態および/または動作のインディケーションをユーザに提供する他の任意のメカニズムを含む。

20

#### 【0064】

外部ユーザ・インターフェース・ポート212は、オーディオ関連デバイス200が、1つ以上の他のオーディオ関連デバイスと通信することを可能にするオプション的な通信ポートであり得る。外部ユーザ・インターフェース・ポート212は、オーディオ・ネットワーク120によって提供される通信パスの外側にある他のオーディオ関連デバイスに通信リンクを提供し得る。例えば、オーディオ・ネットワーク120上で通信する増幅器は、外部ユーザ・インターフェース・ポート212を介して、ウォール・コントローラから入力信号を受け取り得る。ウォール・コントローラの例は、ロータリー・スイッチに

30

関連したデジタル信号である出力、およびウォール・コントローラに含まれているディスプレイ上に表示されるフィードバック・インディケーションである入力を有し得る。この例において、ウォール・コントローラは、増幅器の出力利得を調整するために入力信号を提供し得る。加えて、ウォール・コントローラは、外部ユーザ・インターフェース・ポート212を介して増幅器から出力信号を受け取り得る。出力信号は、出力利得を表し得、かつ、ウォール・コントローラのディスプレイ上に表示され得る。

#### 【0065】

図3は、図2に図解されるオーディオ関連デバイス200の一部、および図1に図解されるオーディオ・システムの一部のより詳細なブロック図である。図3において、通信ポート202、プロセッサ204、オーディオI/O208、およびユーザ・インターフェース210が示されている。通信ポート202は、ネットワーク・インターフェース302を有する通信モジュール300を含む。通信モジュール302は、メモリ206に格納され得、かつ、オーディオ関連デバイス200の任意のへ通信のために用いられ得る。

40

#### 【0066】

ネットワーク・インターフェース・モジュール302は、メッセージ層306およびルーティング層308を含む。メッセージ層306は、オーディオ関連デバイス200とオーディオ・システムにおける他の任意のオーディオ関連デバイスおよび/または機器の間でメッセージを通信するよう構成されている。オーディオ・システム内のメッセージのさまざまなアスペクトがより詳細に後述される。ルーティング層308は、異なるプロトコルおよび/または物理媒体の組み合わせとの通信が可能である複数のサブシステムを

50

含む。

【 0 0 6 7 】

E t h e r n e t ( 登 録 商 標 ) 通 信 リンク 3 1 0、 F i r e W i r e 通 信 リンク 3 1 2、 U S B 通 信 リンク 3 1 4、 R S 2 3 2 通 信 リンク 3 1 6、 R S 4 8 5 通 信 リンク 3 1 8 が オーディオ関連デバイス 2 0 0 内で用いられ得る。他の例において、他の任意の形式の通信リンクが、オーディオ信号、オーディオ関連デバイスの設定およびオーディオ・システムに関連したデータを転送するために用いられ得る。例えば、オーディオ関連デバイスは、オーディオ・ネットワーク 1 2 0 上で、E t h e r n e t ( 登 録 商 標 ) 通 信 リンク 3 1 0 を用いて、または図 3 に示される U S B 通 信 リンク 3 1 4 を用いて、別のオーディオ関連デバイス 3 2 0 と、直接的に通信し得る。ルーティング層 3 0 8 とインターフェースするさまざまな通信リンクのそれぞれは、パケット・サービス層、信頼層、ネットワーク層、データリンク層および物理層を含み得る。それぞれの通信リンク内において、ルーティング層は、オーディオ・ネットワーク上で、または他のオーディオ関連デバイスから受け取られるメッセージを提供する。

10

【 0 0 6 8 】

オーディオ・ストリーム・モジュール 3 2 2 およびオーディオ・サービス・モジュール 3 2 4 はまた、ネットワーク・インターフェース 3 0 2 に含まれ得る。オーディオ・ストリーム・モジュール 3 2 2 は、オーディオ・ネットワーク 1 2 0 上でストリーミングオーディオとしてオーディオ信号 3 1 6 を受け取り、かつ、オーディオ信号 3 1 6 を送達し得る。加えて、オーディオ・ストリーム・モジュール 3 2 2 は、オーディオ・ネットワーク 1 2 0 からストリーム・オーディオを受け取り得る。オーディオ・ストリーム・モジュールの例は、P e a k A u d i o ( C o l o r a d o 州、B o u l d e r ) によって製造されている C o b r a n e t システム ( 登 録 商 標 ) である。ストリーミングオーディオは、オーディオ・ストリーム・モジュール 3 2 2 と別のオーディオ関連デバイスにおける別のオーディオ・ストリーム・モジュールとの間で送達され得る。異なるオーディオ関連デバイスのオーディオ・ストリーム・モジュールは、通信するために互換性があるべきである。

20

【 0 0 6 9 】

オーディオ・サービス・モジュール 3 2 4 は、異なるオーディオ・ストリーム・モジュール間で互換性のある通信を作成するためにオーディオ・ストリーム・モジュール 3 2 2 を自動的に構成するように動作可能である。オーディオ・サービス・モジュール 3 2 4 は、対応するオーディオ関連デバイスのネットワーク・インターフェース・モジュールの設定に基づいて、オーディオ・ストリーム・モジュール 3 2 2 のそれぞれの能力を比較し得る。他のオーディオ・サービス・モジュールとのネゴシエーションに基づいて、オーディオ・サービス・モジュール 3 2 4 は、すべてのオーディオ・ストリーム・モジュールに互換性があり、かつ、共通する設定を選択し得る。従って、オーディオ・ストリーム・モジュールのすべては、もっとも制限された通信能力を有するオーディオ関連デバイスに基づく設定を有するように構成され得る。自動的に設定をネゴシエーションすることによって、オーディオ・サービス・モジュール 3 2 4 は、オーディオ・システムにおいて、お互いに互換性があり、かつ、通信が可能であるすべてのオーディオ・ストリーム・モジュールを構成し得る。オーディオ・サービス・モジュール 3 2 4 によって自動的にネゴシエーションされ得る設定の例は、チャンネル・カウント、ビット深度、サンプル周波数および待ち時間を含む。

30

40

【 0 0 7 0 】

図 3 において、オーディオ関連デバイス 2 0 0 はまた、オーディオ・ネットワーク 1 2 0 上で、P E M 3 2 8 と通信し得る。先述されたように、P E M 3 2 8 は、オーディオ・ネットワーク 1 2 0 上で、オーディオ関連デバイス 2 0 0 を管理、制御、モニタし得る。第 1 のスイッチ 3 3 0 および第 2 のスイッチ 3 3 2 はまた、オーディオ・ネットワーク 1 0 と結合され得る。第 1 のスイッチ 3 3 0 および第 2 のスイッチ 3 3 2 は、オーディオ・ネットワーク 1 2 0 と第 1 のオーディオ・ネットワーク 3 3 4 および第 2 のオーディオ・

50

ネットワーク 336 のそれぞれの間で通信パスを提供し得る。他の複数のオーディオ関連デバイス 334 は、第 1 のオーディオ・ネットワーク 334 および第 2 のオーディオ・ネットワーク 336 に結合され得る。他のオーディオ関連デバイス 338 は、オーディオ・ネットワーク 120 上で、第 1 のスイッチ 330 および第 2 のスイッチ 332 を用いて通信が可能である。

#### 【0071】

##### III. オーディオ・ノード構造

各オーディオ関連デバイス内の機能性がオーディオ・ノード構造を用いて表され得る。オーディオ・ノード構造はまた、GUI 内に 1 つ以上の GUI のコントロール・パネル・ビューを生成するために用いられ得る。GUI のコントロール・パネル・ビューは、対応するオーディオ関連デバイスをモニタ、制御、および構成するために用いられ得る。加えて、オーディオ・ノード構造は、1 つ以上のコントロール面を構成するために用いられ得る。

10

#### 【0072】

図 4 は、オーディオ関連デバイスを表すオーディオ・ノード構造 400 の例である。オーディオ・ノード構造 400 は、ノード 402 に含まれている複数のコンポーネントを含む。ノード 402 は、オーディオ関連デバイスを表し得、かつ、ノード ID を含む。あるいは、ノードは、機能的なオーディオ関連デバイスを形成する多くのノードが存在している、より大きいシステムのサブコンポーネントを形成し得る。各ノードは、1 つ以上の仮想デバイスを含み得る。各仮想デバイスは、1 つ以上のオブジェクト、および状態変数を含み得る。ネットワーク能力を有して構成されているノードは、1 つ以上のネットワーク・インターフェース上で通信し得る。

20

#### 【0073】

図 4 において、ノード 402 の例は、第 1 の仮想デバイス 404 および第 2 の仮想デバイス 406 であるコンポーネントを含む。他の構成の例において、追加的な、またはより少ない仮想デバイスが、特定のオーディオ関連デバイスの機能性に基いて含まれ得る。第 1 の仮想デバイス 404 は、ノード・レベル動作を担うように構成されているノード・マネージャ仮想デバイスである。例えば、ノード・マネージャは、ノード 402 における仮想デバイスのリストを維持し得る。仮想デバイスのリストを用いて、ノード・マネージャ仮想デバイスは、そのノードにおいて、どの仮想デバイスが存在しているかを他のオーディオ関連デバイスに通信し得る。加えて、ノード・マネージャ仮想デバイスは、任意のノード・レベル情報を維持することを担い得る。

30

#### 【0074】

第 1 の仮想デバイス 404 および第 2 の仮想デバイス 406 は、このシステムにおけるデバイスのそれぞれを表す 3 レベル階層のトップレベルである。一般的に、オーディオ・ノード構造における仮想デバイスのそれぞれは、デバイス内の機能ユニットを表す。第 1 の仮想デバイス 404 は、第 2 の階層レベルを表すオブジェクト 408 を含む。後述されるように、オブジェクトは、処理ブロック、制御エレメント、ステータス・エレメント等を表すために用いられ得る。オブジェクト 408 は、第 1 のオブジェクト 410 および第 2 のオブジェクト 412 である追加的なコンポーネントを有し得る。第 1 のオブジェクト 410 および第 2 のオブジェクト 412 は、オブジェクト 408 内の追加的な機能エレメントを表す。

40

#### 【0075】

第 1 のオブジェクト 410 および第 2 のオブジェクト 412、第 1 の仮想デバイス 404 および第 2 の仮想デバイス 406 のそれぞれは、状態変数 414 であるコンポーネントであって、オーディオ・ノード構造のコンポーネントを含む。後述されるように、状態変数は、周波数、フェーダ位置、遅延時間、またはデバイス内の他の任意の変化可能なパラメータといった変化され得るデータ変数ある。従って、このような変数データポイントが、オーディオ・ノード構造を用いて表される特定のオーディオ関連デバイスの機能性に依存して仮想デバイスおよび/またはオブジェクトに存在し得る。状態変数は、第 3 の階層

50

レベルを表している。

【 0 0 7 6 】

ノード 4 0 2、仮想デバイス 4 0 4、4 0 6、オブジェクト 4 0 8、4 1 0、4 1 2 および状態変数 4 1 4 は、各オーディオ関連デバイスのオーディオ・ノード構造を作り上げるコンポーネントである。ノード 4 0 2、仮想デバイス 4 0 4、4 0 6、オブジェクト 4 0 8、4 1 0、4 1 2 および状態変数 4 1 4 は、特定のオーディオ関連デバイスの機能性を反映する任意の構成であり得る。従って、異なるオーディオ関連デバイスのオーディオ・ノード構造は、同一であり得、または、著しく異なり得る。

【 0 0 7 7 】

ノード 4 0 2、仮想デバイス 4 0 4、4 0 6、オブジェクト 4 0 8、4 1 0、4 1 2 および状態変数 4 1 4 のそれぞれ（つまり、コンポーネント）は、属性を含み得る。属性は、どのように、関連したノード、仮想デバイス、オブジェクト、または状態変数が用いられるか記述し、また、名前、測定単位および/または他のデバイス、オブジェクトおよび/または状態変数との関連性を記述し得る。加えて、属性は、対応する仮想デバイス、オブジェクトおよび状態変数のコントロール・パネル・ビューにおける配置といった記述的情報、配線の詳細、または対応するコンポーネントの他の任意の記述もしくは特性を提供し得る。

【 0 0 7 8 】

オーディオ関連デバイスのオーディオ・ノード構造内において、パラメトリック EQ といった信号処理のために用いられるコンポーネント、メカニカル・フェーダといった制御コンポーネント、または出力計といったセンサ・コンポーネント間で差異がないかもしれない。加えて、パスワードおよびミディ（「MIDI」）チャンネルといったグローバルアイテムが、オーディオ・ノード構造に含まれ得る。状態変数、オブジェクトおよび/または仮想デバイスを有するオーディオ・システムに含まれている異なるオーディオ関連デバイスの機能性のすべてを表すことによって、サブスクリプションおよび制御といった通信のための同一のメカニズムが、幅広くさまざまな異なるオーディオ関連デバイスを含むオーディオ・システムにわたってあまねく適用され得る。

【 0 0 7 9 】

ノードは、そこに包含される仮想デバイスを動的に作成し、破壊し得る。オーディオ関連デバイスにおいて動作可能なプログラムは、動的に作成された仮想デバイスの例である。動的に作成された仮想デバイスの別の例は、オーディオ関連デバイスにおけるコントロール・パネル・ビューのためのオートメーション仮想デバイスである。オブジェクトはまた、動的に作成され、破壊され得る。動的に作成されたオブジェクトの例は、オーディオ関連デバイスのコントロール・パネル・ビュー上の挿入物（insert）であり得る。この例において、エフェクト・オブジェクトは、インスタンス化され得、それから、オブジェクトは、このような関連性を記述するように構成可能なルーティング属性を有する状態変数の変化を介してエフェクト・オブジェクトにリンクされ得る。

【 0 0 8 0 】

状態変数

オーディオ・ノード構造内において、オーディオ関連デバイスの機能性において修正可能なパラメータは、状態変数（「SV」）コンポーネント内で保持され得る。このようにして、SV は、測定された値、セットポイント値、チューニング値、設定値、またはオーディオ関連デバイス内の変化可能なパラメータである他の任意のパラメータであり得る実際の値を含む。実際の値は、他の実際の値に基づいて、自動的にアップデート、手動でアップデートおよび/または論理的にアップデートされ得る。SV の例は、周波数のようなオーディオ・オブジェクトのパラメータおよびコントロール面上のフェーダの位置を含むウォール・コントローラのようなシンプルなオーディオ関連デバイスは、制限された数の SV だけを包含し得る。ミキシング・コンソールといった他のオーディオ関連デバイスは、何十万もの SV を包含し得る。SV 上の一般的な動作は、変数を「セットし」、かつ、変数を「獲得する」ことを含む。例えば、このような動作は、EQ の周波数をセットす

10

20

30

40

50

ること、およびオーディオ関連デバイスのコントロール面上の表示のための遅延時間を獲得することに変換され得る。このような基本動作は、オートメーションといった、より複雑な形式の使用の基本を形成し得る。

#### 【 0 0 8 1 】

##### 属性

オーディオパラメータの値を保持することに加えて、S Vはまた、どのように対応するS Vが用いられるかを記述する多くの属性を所有し得る。2つの属性の例は、Gain S Vがセットされ得るデシベル(d B)値のレンジを表す「最小」および「最大」である。別の属性は、d B値を保持するために用いられるデータ・タイプであり得る。例えば、データ・タイプは、IEEEフロート、バイト、ワードとしてS Vを示し得る。従って、属性情報を含むことによって、他のオーディオ関連デバイスが、S Vに対して特にプログラムされておらず、それ故に、S Vの特性を知らないときでさえ、S Vはオーディオ・システムにおける他のオーディオ関連デバイスにアクセス可能である。このようにして、属性は、その属性を発見するために予備知識を有しておらず、従って、S Vを用いるために必要とされる知識を取得する他のオーディオ関連デバイスによって用いられ得る。

10

#### 【 0 0 8 2 】

属性はまた、オーディオ・ノード構造を記述する情報を提供し得る。従って、オーディオ・ノード構造のいくらかの部分が未知であるとき、未知のコンポーネントに関連した属性は、コンポーネントの機能性を決定するために用いられ得る。加えて、属性は、オーディオ・ノード構造とコントロール・パネル・ビューとの間の対応を提供し得る。例えば、属性は、オーディオ関連デバイスに対して格納されている所定のコントロール・パネル・ビューを識別するために用いられ得る。加えて、格納されている所定のコントロール・パネル・ビューがオーディオ・ノード構造に完全に対応しないとき、または、格納されている所定のコントロール・パネル・ビューが全く存在しないとき、オーディオ・ノード構造に含まれる属性はコントロール・パネル・ビューを形成する情報を提供し得る。このようにして、オーディオ・デバイスのオーディオ・ノード構造が、格納されている所定のコントロール・パネル・ビュー、または格納されている所定のアスペクトを有しないとき、オーディオ・ノード構造の属性は、コントロール・パネル・ビューおよび/またはアスペクトを生成する情報を提供し得る。

20

#### 【 0 0 8 3 】

属性はまた、生成されたアスペクトの格納されている所定のコントロール・パネル・ビューへの統合または取り込みが可能である情報、またはコントロール・パネル・ビューに多くの生成されたアスペクトを形成する情報を含み得る。例えば、オーディオ・ノード構造は、ミキサーにおけるチャンネルを表すオブジェクトおよびS Vを含み得る。オブジェクトおよびS Vに関連した属性は、チャンネルが、制御能力を提供するスライダとの割合として表示されるべきであることを示し得る。加えて、属性は、それぞれに対して生成されるさまざまなアスペクトの配置を記述し得る。さらに、属性は、アスペクト間の配線がどのように表示されるべきか、アスペクト、カラー、サイズ、シェイプ、またはコントロール・パネル・ビューにおけるオーディオ関連デバイス機能性のディスプレイに関連する他の任意のパラメータ間の相互間の関連がどのように表示されるべきかを記述し得る。属性の概念は、オーディオ・ノード構造全体にわたり適用され、S V、オブジェクト、仮想デバイス、およびノード・マネージャ仮想デバイスを含む。

30

40

#### 【 0 0 8 4 】

属性は、その属性が用いられ得る方法を表す3つのカテゴリの1つに分類され得る。属性のカテゴリは、静的、インスタンス、および動的を含む。静的属性は、固定されており、変化しない。従って、静的属性のデータベースは、オーディオ・ノード構造における、S Vといったコンポーネントを制御するために必要とされる発見の量を最小にするものとして定義され得る。インスタンス属性は、S Vのインスタンスごとにセットされ得る。動的属性は、クライアントセットの、またはランタイム時に修正され得る静的属性、またはインスタンス属性である。静的およびインスタンス属性は、クエリされ得る。動的属性は

50

、一般的に、静的属性またはインスタンス属性のいずれかと類似している。動的属性は、しかしながら、その値がランタイム時にオーバーライドされ得るので静的属性またはインスタンス属性とは異なる。

【0085】

属性は、属性識別（「属性ID」）およびクラス識別（「クラスID」）を含み得る。属性IDは、各属性を一意的に識別する方法である。属性IDは、符号なしのバイト（「UBYTE」）を用いてゼロから数えられ得、所定の名前を用いて表され得る。オーディオ・ノード構造の階層の各レベルでの属性は、それ自身の列挙を有し得る。従って、例えば、SVクラスに対する3の属性IDは、オブジェクト・クラスにおいて異なる属性を指し得る。クラスIDは、属性値の一意のグループを識別する符号なしのワード（「WORD」）といった所定の数を用いて表され得る（クラスID）。例えば、SVが、特定の値にセットされた11個すべての属性を有するとき、属性のこのコレクションおよびそれらの値は、SVクラスとして識別され、代表クラスIDを割り当てられ得る。このクラスIDは、長文式で属性を記述することを省くために省略表現の形式として用いられ得る。クラスIDは、属性値のセット（SVクラス、オブジェクト・クラス等）を表す階層のすべてのレベルにおいて用いられ得る。SVおよびオブジェクト・クラスの再使用を促すために、クラスIDのマスター・リストが維持され得、クラスが登録され得る。

10

【0086】

クラス

SVクラス、オブジェクト・クラス、仮想デバイス・クラス、またはノード・マネージャ仮想デバイス・クラスといったクラスは、数的に、かつ、それぞれクラスID（「WORD」）およびクラス名（「STRING」）といったストリング名によって表され得る。各クラスIDは、ID、属性名およびカテゴリといった属性のセットを含み得る。クラス名は、オーディオ・ノード構造における階層レベルを識別するためにプレフィックスされ得る。例えば、SVのクラス名は、「SVClass」であり得る。SVクラス名の例は、SVClassPeqFreq、SVClassPeqQおよびSVClassPeqGainである。クラスIDはまた、SVクラス、オブジェクト・クラス、仮想デバイス・クラスおよびノード・マネージャ仮想デバイスのいずれかに対して用いられ得る。SVクラスIDにおける属性のセットの例は以下のとおりである。

20

【0087】

【表1】

30

ID	属性名	データ・タイプ	カテゴリ
0	データ・タイプ		静的
1	名前ストリング	ストリング	インスタンス+動的
2	最小値	データ・タイプ	静的+動的
3	最大値	データ・タイプ	静的+動的
4	制御法則		静的
5	フラグ	UBYTE	静的
6	デフォルト値	データ・タイプ	インスタンス
7	コントロール・プリミティブ		インスタンス
8	コントロール座標		インスタンス
9	エンコーダ・モード	UBYTE	

40

表1

オブジェクト・クラスおよびその関連したクラスIDは、オブジェクトの特定の先験的な知識を定義し得る。オブジェクト・クラスは、オブジェクトが、オブジェクト・クラス

50

およびSVの値によって全体的に定義されることを可能にする。オブジェクト・クラスは、静的およびインスタンス属性から形成され得る。オブジェクト・クラスを定義する静的属性は、オブジェクトによって包含される任意のサブオブジェクト、およびオブジェクトによって包含される任意のSVを含み得る。例えば、3つのバンドEqualizerおよび5つのバンドEqualizerは、異なるオブジェクト・クラスである。なぜなら、それらは、異なる数のSVを包含するからである。オブジェクト・クラスIDは、オブジェクト・クラスを特定するために用いられるUWORDであり得る。オブジェクト・クラスIDの1つのマスター・レジストリがあり得る。デベロッパーは、それらのオブジェクトIDをマスター・レジストリに登録し得る。加えて、同一のオブジェクトが、オーディオ関連デバイスのファミリに渡り用いられ得る。オブジェクト・クラスIDは、先述されるような属性のセットを含み得る。オブジェクト・クラスは、フレーズ「ObjClass」で始まる命名規則を有し得る。オブジェクト・クラスIDにおける属性のセットの例は以下のとおりである。

【0088】

【表2】

ID	属性名	データ・タイプ	カテゴリ
0	クラス名	STRING	静的
1	名前ストリング	STRING	インスタンス
2	フラグ	UBYTE	インスタンス

表2

仮想デバイス、またはノード・マネージャ仮想デバイスはまた、クラスを含み得る。仮想デバイスのクラスおよび関連したクラスIDは、仮想デバイスの一意的特性を表し得、容易な識別および異なる種類の仮想デバイスの区別のために用いられ得る。2つの異なる種類の仮想デバイスの例は、ミキシング・コンソールといったオーディオ関連デバイスのノード内のオートメーション仮想デバイスおよび入力面仮想デバイスである。一意のクラスIDの割り当ての例は、1つ以上のオーディオ関連デバイスに含まれている異なるソフトウェアプログラムを表す仮想デバイスのグループへのクラスIDの割り当てであり得る。仮想デバイスのクラスIDは、UWORDとして表され得る。仮想デバイスはまた、クラス名を有し得る。仮想デバイス・クラス名は、ユーザに対して人間読み取り可能省略表現を提供するために用いられ得る。このクラス名は、一意である必要はない。仮想デバイス・クラスIDにおける属性のセットの例は以下のとおりである。

【0089】

【表3】

ID	属性名	データ・タイプ	カテゴリ
0	クラス名	STRING	静的
1	名前ストリング	STRING	インスタンス

表3

ノード・マネージャ仮想デバイス・クラスIDにおける属性のセットの例は、以下を含む。

【0090】

【表 4】

ID	属性名	データ・タイプ	カテゴリ	
0	クラス名	STRING	静的	
1	名前ストリング	STRING	インスタンス	
2	フラグ	UWORD	インスタンス	
3	シリアル・ナンバー	BLOCK	インスタンス	
4	ソフトウェア・バージョン	STRING	インスタンス	10
5	リザーブ	UWORD (0にセット)		
6	ホップ・カウント	UBYTE	インスタンス+動的	
7	ベニュー・テーブル・サイズ	UWORD	静的	
8	ユーザ名 A	STRING	インスタンス+動的	
9	ユーザ名 B	STRING	インスタンス+動的	
10	ユーザ名 C	STRING	インスタンス+動的	
11	ユーザ名 D	STRING	インスタンス+動的	
13	ユーザ・パスワード A	STRING	インスタンス+動的	20
13	ユーザ・パスワード B	STRING	インスタンス+動的	
14	ユーザ・パスワード C	STRING	インスタンス+動的	
15	ユーザ・パスワード D	STRING	インスタンス+動的	
16	アドレス・モード	UBYTE	インスタンス+動的	

表 4

ここで、「BLOCK」データ・タイプは、 $N \times N$ のバイトのサイズのデータを含むデータ構造を表す。

## 【0091】

静的属性

静的属性は、クラスのそれぞれに対してあらかじめ定義され得る。例えば、SVクラスは、異なるSV値、最小SV値、最大SV値、制御法則およびフラグを表すデータのセットの静的属性を含み得る。

## 【0092】

制御法則は、SVがどのように制御されるべきかを記述し得る。例えば、周波数に対するSVは、対数であり得、利得SVは、およそ0dBで付加的な分解能を有する対数であり得る。SVが、そのクラスのMinとMaxとの間の浮動小数点値を取るとき、ユーザに、あるルックアンドフィールを与えるために制御法則を特定することが所望され得る。例えば、周波数変数の場合には、ユーザがエンコーダをターンするとき、またはスピナー上の「アップ」コントロールボタンを押すとき、次の値が前の値から間隔が対数的に置かれることが所望され得る。制御法則はまた、SVが受容し得る粒度を特定するために用いられ得る。例えば、利得SVは、0.1dBの最大分解能有し得る。すべてのステップがあらかじめ定義されているので、列挙されたSVの場合には、制御法則は省略され得る。

## 【0093】

フラグは、属性に関連したSVをさらに定義するためにあらかじめ定義され得る。フラグの例は、周期的に、または変化毎に、サブスクリプションのタイプを自動的にセットするためにサブスクリプションにおいて用いられ得るセンサ属性を含む。センサSVの例は、出力計、しきい値計測器、または電力増幅器温度を含む。非センサSVの例は、周波数またはMIDIチャンネルを含む。フラグの他の例は、SVがイベント・サブスクリプシ

30

40

50

ョンにサブスクライブされ得ることを示すイベント・タイプ、または最大、最小、またはSV名ストリングが、ドキュメントされた静的クラス値から変化されたときといった変化を示す修正されるフラグを含む。SV名ストリングは、SVクラス内の「インスタンス属性」であるが、それは、普通、オブジェクト内で静的であり、従って、それになされた変化は、修正されたフラグを用いて追跡され得る。

【0094】

オブジェクト・クラス、仮想デバイス・クラス、またはノード・マネージャ仮想デバイス・クラスはまた、クラス名といった静的属性を含み得る。オブジェクト・クラス名は、ユーザに対して人間読み取り可能省略表現を提供するために用いられ得る。オブジェクト・クラス名は、オーディオ関連デバイスのファミリの間で一意である必要はない。例えば、ミキサーである第1のオーディオ関連デバイスは、オブジェクト・クラス名「ObjClassSC\_PEQ\_5」を有し得、dbx DriveRackといったオーディオ関連デバイスの同一のファミリにおける第2のオーディオ関連デバイスは、クラス名「ObjClassDR\_PEQ\_5」を有し得る。双方のオーディオ関連デバイスは、クラス名「5Band PEQ」を有し得る。クラス名は、GUIのコントロール・パネル・ビュー上のドロップダウンから選択され得、ストリングとして格納され得る。

10

【0095】

仮想デバイスにおいて、静的特性は、クラスIDに関連した値を有する仮想デバイスに属する属性であり得る。1つの例において、静的属性は、インスタンスごとにセットされず、かつ、ランタイム時に動的に変化されないかもしれない。仮想デバイスに対する静的属性はまた、名前ストリングを含み得る。名前ストリングは、仮想デバイスのインスタンス化を命名するために用いられ得る。名前ストリングは、クラス名と同一であり得る。同一名の複数のインスタンスがある際には、複数のインスタンスが、スペースおよび小数を加えることによって区別され得る。例えば、ミキサーであるオーディオ関連デバイスにおいて、「MHx Input Bank」の仮想デバイス・クラス名を有する入力フェーダ・バンクの2つのインスタンス化があるとき、各インスタンスに対する仮想デバイスの名前ストリングは、「MHx Input Bank 1」および「MHx Input Bank 2」であり得る。

20

【0096】

ノード・マネージャ仮想デバイスの静的属性はまた、クラス・タイプを含み得る。ノード・マネージャのクラス・タイプは、ノード・マネージャの静的属性の値を決定し得る。属性は、インスタンスごとに基づいて変化しないかもしれないし、ランタイム時に変化されえないかもしれない。例えば、オーディオ関連デバイスの特定のタイプに対するノード・マネージャ・クラスにおける静的属性は、ノード・マネージャのすべてのインスタンス化において同一の値を有する。ノード・マネージャはまた、ベニュー・テーブル・サイズを示す静的属性を含み得る。ベニュー・テーブル・サイズの静的属性は、オーディオ関連デバイスによってサポートされるベニュー・テーブルのサイズを発見するためにクエリされ得る。ベニュー・テーブルの使用および動作は、後述される。

30

【0097】

インスタンス属性

インスタンス属性は、SV、オブジェクト、仮想デバイス、またはノード・マネージャ仮想デバイス・クラスの各インスタンスに対して一意の値にセットされ得る静的属性である。

40

【0098】

SVに関しては、オブジェクトまたは仮想デバイスは、値を定義するSVをインスタンス化し得る。従って、これらの属性は、オブジェクトまたは仮想デバイスに対して静的であり得る。SVクラスは、SV ID、名前ストリング、デフォルト値、コントロール・プリミティブ、コントロール座標 (coordinate)、およびエンコーダ・モードといったインスタンス属性を含む。SV IDは、オブジェクトまたは仮想デバイス内の各SVの識別子として用いられるUWORDであり得る。SV IDは、ゼロベースであ

50

る必要はなく、またオブジェクトまたは仮想デバイスを完全に占める必要はない。SVの名前ストリングは、SVに対して人間読み取り可能プロンプトを提供する。このストリングは、オブジェクト内のSVに対して一意である必要はない。例えば、パラメトリックEQの各バンドに対して、利得を表す各SVは、SVの名前ストリング「Level」を有する。デフォルト値は、オブジェクトまたは仮想デバイスを初期化するために用いられ得る。デフォルト値は、SVのデータ・タイプにおいて表され得る。デフォルト値は、プログラマが、オブジェクトの必要性に従って、これを一意に特定し得るように動的であり得る。例えば、2つのクロスオーバー周波数のSVを有するクロスオーバー・オブジェクトは、それぞれ100Hzおよび1.0KHzでそれらが初期化されることを所望する。コントロール・プリミティブは、オブジェクト、または仮想デバイスに対するジェネリックGUIのコントロール・パネル・ビュー、またはジェネリックGUIの aspekt 作るために用いられ得る。オブジェクト、または仮想デバイスは、コントロール・プリミティブが、各SVに対して用いられること示唆し得る。コントロール座標は、ジェネリックGUIのコントロール・パネル・ビュー、またはジェネリックGUIの aspect に対するコントロール・プリミティブの配置に対して用いられ得る。エンコーダ・モードは、どのようにエンコードといったGUIの aspect 上のSV値を表示するかを示唆し得る。Encoder Mode 属性の Data Type は、UBYTEであり得る。

10

#### 【0099】

オブジェクトはまた、名前ストリングといったインスタンス属性を含み得る。名前ストリングは、インスタンス化されたオブジェクトの名前であり得、STRINGデータ・タイプに格納され得る。例えば、プロダクトは、インスタンス化「Input 1」、「Input 2」等を有するオブジェクト・クラス「Input\_Channel」を有し得る。オブジェクトはまた、フラグであるインスタンス属性を含み得る。フラグは、対応するオブジェクトに関しての情報を提供し得る。例えば、フラグの1つは、修正されるフラグあり得る。修正されるフラグは、オブジェクト内の1つ以上のSVが、それらの静的クラス定義から修正されたときを示すビットであり得る。可能な修正は、「SV Dynamic Attributes」のもとでリストアップされる。修正されたオブジェクトを認識した際に、コントロール面を有する別のオーディオ関連デバイスは、変化に対処するためにさらにクエリを行い得る。

20

#### 【0100】

ノード・マネージャ仮想デバイスはまた、ノード・マネージャが作成される時、カスタム値にセットされ得るインスタンス属性を含み得る。このようなインスタンス属性は、ランタイム中に動的に変化可能ではないかもしれない。ノード・マネージャのインスタンス属性の例は、先述されたノード・マネージャ仮想デバイス・クラスの例に含まれる(表4)。

30

#### 【0101】

表4の例において、ノード・マネージャ仮想デバイスの仮想デバイス名前ストリングは、ノードおよびノードが属するワークグループの双方の名前を指定し得る。名前ストリング属性は、ユーザセットでき(user configurable)、不揮発性のメモリに格納され得る。シリアル・ナンバー属性は、各ノードを識別する一意のナンバーであり得る。例えば、一意のナンバーは、アドレス衝突を解決するために用いられ得る。シリアル・ナンバーの1つのタイプの例は、16進法のナンバーストリングである統計的に一意の128ビットIDであり得る。例えば、統計的に一意の128ビットIDの生成は、Microsoft(登録商標)のアプリケーション「uidgen」を起動するPCを有して行われ得る。生成されたシリアル・ナンバーは、製造中にノードの不揮発性メモリに書き込まれ得、決して変化され得ない。シリアル・ナンバーの別のタイプの例は、先頭から80ビットがゼロにセットされた48ビットEthernet(登録商標)MACアドレスである。オーディオ関連デバイスが2つ以上のネットワーク・インターフェースを有する際には、通常、有効である、または、より頻繁に有効であるインターフェースからのMACアドレスが用いられ得る。

40

50

【 0 1 0 2 】

表 4 におけるソフトウェア・バージョン属性は、オーディオ関連デバイスにおいて動作可能な内蔵のファームウェアまたは PC アプリケーション・ソフトウェアのリビジョンであり得る。ソフトウェア・バージョンは、任意の形式のストリングによって表され得る。ホップ・カウント属性は、出力メッセージ ( o u t g o i n g m e s s a g e ) が初期化されるホップの数を示すために用いられ得る。ホップ・カウント属性のデータ・タイプは、 U B Y T E であり得、デフォルトは、所定のホップの数であり得る。ユーザ名 A D 属性は、ユーザの各カテゴリに対するユーザ名を包含し得る。ユーザ・パスワード A D 属性は、ユーザの各カテゴリに対するパスワードを包含し得る。

10

【 0 1 0 3 】

表 4 におけるアドレス・モード属性は、ノードが、アドレス・ネゴシエーション中にランダムな、またはシーケンシャルのノード・アドレスを選ぶか否かを制御し得る。アドレス・ネゴシエーションは後述される。オーディオ・システムの例において、すべてのオーディオ関連デバイスは、アドレス衝突の可能性を最小にするためにランダムアドレッシングを用いるためにセットされ得る。あるいは、ツアー中のサウンド・オーディオシステムのような他のオーディオ・システムにおいて、シーケンシャルにナンバリングされたノードを有することが所望され得る。従って、アドレス・モード属性は、シーケンシャル・ノード・アドレッシングにセットされ得、シーケンシャル・ナンバリングは、1 度に 1 つずつノード ( オディオ関連デバイス ) をオンし、各々の次のノードが、最後のアドレスより 1 つ大きいアドレスを選択することにより達成され得る。アドレス・モード属性は、 0

20

R a n d o m および 1 S e q u e n t i a l のフォーマットの例を有する U B Y T E に含まれ得る。

【 0 1 0 4 】

動的属性

動的属性は、クライアントセットであり得、またはランタイム時に修正され得る属性のサブセットである。このことは、 S V に対してユーザ定義のレンジをおよび名前をセットするとき有用である。この例において、動的に修正され得る S V 属性は、最小値、最大値、および名前ストリングである。修正可能な属性が、公表値から逸脱し、「動的」になるとき、発見プロセスは、実際の値を決定するために付加的なステップを必要とするかもしれない。この理由のために、動的属性のリストは、発見中に付加的なオーバーヘッドを制限するために特定の必要性がある状況に制限され得る。

30

【 0 1 0 5 】

仮想デバイスはまた、動的属性を利用し得る。例えば、動的属性は、仮想デバイスのクラス ID の一部として含まれ得る。この例において、動的属性は、仮想デバイスが、ランタイム時に異なるオブジェクトを包含するために再構成され得ることを示すために用いられるビットフラグであり得る。ビットフラグがクリアされるとき、動的属性は、仮想デバイスがオブジェクトの静的な構成を包含し、そのクラス ID によって全体的に知り得ることを示し得る。オーディオ関連デバイスが再構成可能なとき、1 つより多い動的仮想デバイス・クラスの仕様がオーディオ関連デバイスに起こり得る。このシナリオにおいて、再構成可能なオーディオ関連デバイスは、オーディオ関連デバイスにおいて実行されるオーディオ関連機能または動作のセットを表すために動的仮想デバイスを用い得る。従って、クラス・タイプは、オーディオ関連機能または動作のセットのバージョンを識別するために用いられ得る。加えて、クラス・タイプは、オーディオ関連機能または動作のセットが、ソフトウェア・パッチ、ソフトウェア・アップグレード、別の仮想デバイスにおけるオーディオ関連機能または動作セット等と互換性があるかどうかを決定するために用いられ得る。例えば、ユーザが、動的仮想デバイスによって表されるオーディオ関連デバイスのオーディオ関連機能または動作のセットの値を、クリップボードに複製するとき、仮想デバイスタイプのシンプルなチェックは、ユーザがそのオーディオ関連機能または動作のセットの値を、それ自身の一意的動的仮想デバイス・クラスを有する別の仮想デバイスにおける互換性がないオーディオ関連機能または動作セットにペーストし得ることをストップ

40

50

する。

【0106】

通信サポート

S V、オブジェクトおよび仮想デバイスは、オーディオ関連デバイス間の通信をサポートする。S V、オブジェクトおよび仮想デバイスのクエリおよび修正は、メッセージベースの通信インフラを有し得る。加えて、ポインタ等を介してクエリおよび修正を行うダイレクトコーリングがまた可能である。通信インフラは、S V、オブジェクトおよび仮想デバイスを用いる通信インフラのサポートが異なるオーディオ・ノード構造を有する異なるオーディオ関連デバイスにわたり一様であり得るようにはあらかじめ決定され得る。

【0107】

所定のメッセージのサポートは、クエリおよび修正されているオーディオ・ノード構造コンポーネントにいくらか依存し得る。S Vは、実際の値を含み得、それゆえに、関連した実際の値の修正に関連したメッセージをサポートする。符号ありのフォーマット済の置換用の実際の値、または割合の変化値を有するセットポイントといったS Vの実際の値を変化させるメッセージがサポートされ得る。

【0108】

さまざまな形式の「獲得 ( g e t )」メッセージがサポートされ得る。獲得メッセージは、1つ以上のS V、オブジェクト、または仮想デバイスに関連した属性を取得する獲得属性メッセージを含み得る。加えて、獲得メッセージは、獲得値ストリング・メッセージを有するといったS Vから実際の値を要求し得る。オブジェクトおよび仮想デバイスの場合には、リストまたはテーブルを提供する獲得メッセージがサポートされ得る。後述されるように、ノード内の仮想デバイスのリストは、仮想デバイス・ノード・マネージャから要求され得る。同様に、仮想デバイス内のオブジェクトまたはS Vのリストおよび/またはオブジェクト内のオブジェクトのリストといった他のコンポーネント内部のコンポーネントのリストは、獲得メッセージを用いて要求され得る。獲得メッセージはまた、それぞれの仮想デバイスまたはオブジェクトによって表される機能性に含まれている入力および出力に関連した情報を提供する仮想デバイスおよび/またはオブジェクトによってサポートされ得る。例えば、メッセージは、仮想デバイスI/Oのリストおよび/またはオブジェクトI/Oのリストに対してクエリし得る。さらに、S V、オブジェクトおよび/または仮想デバイスに関連した他の任意の情報を提供する獲得メッセージがまたサポートされ得る。

【0109】

「セット ( s e t )」メッセージがまたサポートされ得る。セットメッセージは、S Vにおいて実際の値をセットし、属性をセットし、リストにおいて1つ以上の値をセットし、テーブルを修正等し得る。加えて、「サブスクライブ ( s u b s c r i b e )」および「アンサブスクライブ ( u n s u b s c r i b e )」は、S V、オブジェクトおよび仮想デバイスによってサポートされ得る。情報に対するオーディオ関連デバイスをポーリングすることを伴う「獲得」メッセージとは対照的に、「サブスクライブ」メッセージは、情報を提供するイベント・ドリブン・メカニズムを作成し得る。例えば、サブスクライブ・メッセージは、変化が行われたいかなる時にも、コンポーネント変化内のS Vの実際の値を要求するために用いられ得る。このようなサブスクライブ・メッセージは、同一または異なるオーディオ・ノード構造を有する1つ以上の個々のオーディオ関連デバイス間のサブスクリプション可能にするためにサポートされ得る。例えば、第1のオーディオ関連デバイスにおける第1のオブジェクトまたは仮想デバイスは、第2のオーディオ関連デバイスにおける1つ以上の他のオブジェクトおよび/または仮想デバイスにおけるすべてのS V、選択されたS V、または1つのS Vにサブスクライブし得る。サブスクライブ・メッセージは、実際の値、エラー、構成および/またはステータスといったコンポーネントについて利用可能な任意の情報に関連し得る。

【0110】

ルーティング

メッセージのルーティングは、オーディオ関連デバイスのノード構造におけるオブジェクトといったコンポーネント間であり得る。加えて、メッセージは、異なるオーディオ関連デバイスのノード構造におけるコンポーネント間でルーティングされ得る。ルーティングは、仮想デバイスおよび/またはオブジェクト間で構成され得る。異なるオーディオ関連デバイスにおけるコンポーネント間でのルーティングに対して、コンポーネントは、1つ以上の物理的インターフェースに結び付けられており、物理的インターフェースは、オーディオネットワークへ/からオーディオをストリームし得る。

【0111】

ルーティングは、各コンポーネントを構成することによって構築され得る。このような構成は、各ルーティングのソースおよび目的地を一意に識別することを伴い得る。例えば、第1のオブジェクトといった第1のコンポーネントは、入力が、第2のオブジェクトといった第2のコンポーネントの出力からであること、および第1のコンポーネントの出力が、第2のコンポーネントの入力に提供されることを識別するために構成され得る。各コンポーネントは、多数のチャンネルを有し得る。チャンネルは、ゼロベースの整数値といった所定の識別子によって識別され得る。チャンネルは、入力または出力のいずれかであるように構成され得る。チャンネル識別子は、一意のオーディオ・アドレスを形成するためにオーディオ関連デバイスのオーディオシステムアドレスと組み合わせられ得る。例えば、0xABC00000001のオーディオ・アドレスを有するオーディオ関連デバイスは、I/O ModeのOutputであるチャンネル12を有し得る。

10

【0112】

オーディオ・アドレスは、出力オーディオ・アドレス、または入力オーディオ・アドレスであり得る。従って、オーディオ関連デバイスは、入力チャンネルがどこでそのオーディオソースを取得すべきかを識別するために出力オーディオ・アドレスを受け取り得る入力チャンネルを有し得る。加えて、オーディオ関連デバイスは、出力チャンネルがどこにその出力をルーティングすべきかを識別するために入力オーディオ・アドレスを受け取り得る入力チャンネルを有し得る。例は以下のとおりである。

20

入力アドレスを特定する出力接続(第1のオブジェクト)

オーディオ・アドレス: 0xABCD00000002, チャンネル: 2, I/O Mode: 入力

出力アドレスを特定する入力接続(第2のオブジェクト)

オーディオ・アドレス: 0xABCD00000001, チャンネル: 12, I/O Mode: 出力

30

サブスクリプション

サブスクリプションは、何か変化があった際にコントロール・パネル・ビューおよび/またはオーディオ関連デバイスに通知するために用いられ得るメカニズムである。サブスクリプションは、「変化毎」に、または「周期的」といった所定の条件に基づき得る。すべての測定は、サブスクリプションを介して処理され得る。例えば、第1のコントロール・パネル・ビューは、SVにサブスクライブする。そのSVが、第2のコントロール・パネル・ビューによって変化されるとき、第1のコントロール・パネル・ビューには、SVの新しい実際の値が伝えられる。通知は、SVSetメッセージといったセットメッセージを用いて処理され得る。すべてのサブスクリプション・メッセージは、SVの実際の値といった即時の応答を自動的に生成し得る。その応答は、同一の値を有する異なるコントロール・パネル・ビューを同期させるために即時であり得る。

40

【0113】

サブスクリプション・メッセージにおいて、サブスクライブされるものは、変化を報告するために特定された仮想デバイス、オブジェクトおよびSD IDを用いるように指示され得る。サブスクライブされるものの観点からして、変化がサブスクライブするものに報告されるとき、サブスクライブするもの上のSVは、サブスクライブされるものに、サブスクライブされるもの上の特定のSVが変化されたことをサブスクライブするものに通知することとは対照的に、新しい値にセットされる。例えば、このメカニズムは、サブス

50

クライブするものがサブスクライブされるもののフルミラーされたデータのセットを保持していないとき、有用である。ウォール・コントローラまたはカスタム・コントロール・パネルといったシンプルなコントロール面であるサブスクライブするものは、SV自身のオーディオ・アドレスを有するそれ自身のSVのセットを有し得る。サブスクライブされるものは、サブスクライブするものの側上で必要とされる変換なしに、これらのSVを直接的に単純にセットし得る。従って、各SVは、マッピングされる0からNのオーディオ・アドレスのセットおよびSV IDを有し得る。SVが変化するとき、所定のサブスクリプションのリストは、サブスクライブされるものによってアクセスされ得、SV\_Setといったセットメッセージが、変化されたSVにマッピングされるサブスクリプションのリスト上のサブスクライブされるもののそれぞれに送達され得る。ネットワーク・トラフィックを最小にするために、通知は、バンドルされ、通常のバンドリング技術のいずれか1つを用いて任意の1つのノードに送達さえ得る。

10

各SVクラスは、それぞれのSVがセンサSVまたは非センサSVとみなされるべきかを定義し得る。すべてのセンサSVは、周期速度でといった所定の条件に基づいて通知し得る。周期速度は、サブスクライブするものによって特定され得る。オーバーヘッドであるために、センサが、サブスクライブするものによって示唆された速度でサポートし得ないとき、センサは、より遅い速度で通知し得る。センサは、サブスクライブするものによって特定された速度より早い速度で通知しないように構成され得る。サブスクリプション・メッセージは、サブスクリプションがセンサSVに対してであるとき、用いられる速度を含み得る。非センサSVは、変化毎に通知するように自動的に割り与えられ得る。

20

#### 【0114】

サブスクリプションはまた、イベント・サブスクリプション、またはバリュー・サブスクリプションといった所定の条件に基づき得る。バリュー・サブスクリプションは、SVの実際の値が第3者によって変化されたとき、SVの新しい実際の値が通知される要求である。例えば、コヒーレントな状態でとどまる必要がある複数のコントロール・パネル・ビューが、ミキサー・コアであるオーディオ関連デバイスに接続しているとき、バリュー・サブスクリプションが用いられ得る。第1のコントロール・パネル・ビューが、ミキサー・コアにおいてミュートであるとき、第2のコントロール・パネル・ビューは、バリュー・サブスクリプションを有して、第2のコントロール・パネル・ビューにおいてミュートインジケータを制御するSVに通知され得る。

30

#### 【0115】

イベント・サブスクリプションは、スイッチ・コンタクト・クロージングといったハードウェア・トリガに応答してSVの現在の値が通知される要求である。SVは、それに関連した入力ハードウェアを有するとき、イベント・サブスクリプションをサポートし得る。イベント・サブスクリプションの能力は、SVクラス宣言の一部として示され得る。イベント・サブスクリプションをサポートしていないSVからイベント・サブスクリプションを取得する試みは、エラーをもって拒絶される。

#### 【0116】

アプリケーションの例において、イベント・サブスクリプションは、割り当て可能なコントロール面上の「バンク」ボタンをインプリメントするために用いられ得る。バンクスイッチは、出力オーディオ・アドレスとして「マクロ(macro)」という名前を保持するSVを維持する。バンク・ボタンが押下されるとき、イベント・サブスクリプションは、SVの実際の値をサブスクライブしているオブジェクトに提供し得、サブスクライブしているオブジェクトは、「マクロ」を実行し得、これによって、フェーダを再マッピングする。

40

#### 【0117】

イベントおよびバリュー・サブスクリプションは、相互に排他的でない。イベントおよびバリュー・サブスクリプションの双方は、同時に有効であり得る。この例は、オートメーション・オブジェクトである。先述された例において、オートメーション・オブジェクトは、バンクスイッチが押下されたときに、記録をするためにイベント・サブスクリプ

50

ション、および、異なる値にリプログラムされているスイッチをキャプチャするためにバリュー・サブスクリプションを用い得る。先述の例はまた、2つの別々のSVとしてコード化され得、1つは、実際のスイッチの押下を表し、別の1つは、スイッチの押下で送達され得る値を表す。この仕組みでは、イベント・サブスクリプションは、オーディオ関連デバイス上でサポートされ得る。

**【0118】**

サブスクリプションはまた、マルチレベルであり得る。SVサブスクリプションでは、サブスクリプションは1つサブスクリプション要求メッセージ内の単一のSV、またはSVのリストに対してであり得る。オブジェクトおよび仮想デバイスサブのサブスクリプションはまた、単一のSVまたは複数のSVに対してであり得る。オブジェクトにおけるSVのすべては、SubscribeAllメッセージといった単一のメッセージを有してサブスクライブされ得る。仮想デバイスまたはオブジェクトが複数のオブジェクトを包含するとき、サブスクライブされるものは、サブスクリプションにおける仮想デバイスまたはオブジェクトの階層オーディオ・ノード構造を形成するSVのすべてを含み得る。1つの例において、サブスクリプション・データ・メッセージで送達されるべきデータの量を最小にするために、サブスクライブされるものは、サブスクライブするものの仮想デバイスおよび/またはオブジェクトが、サブスクライブされるものとの構造を有すると仮定し得る。サブスクライブされるものが、サブスクライブされるもののオーディオ・ノード構造の一部をミラーするオーディオ・ノード構造の一部を有するとき、アドレッシング情報は、サブスクリプション・データ・メッセージペイロードに含まれるSVの実際の値から省かれ得る。ノード全体へのサブスクリプションがまた要求され得る。ノードによって生成されるエラーのすべてにサブスクライブするために、SubscribeErrorsメッセージといったノード構造メッセージが、サブスクライブされるものの仮想デバイス・ノード・マネージャに向けられ得る。

**【0119】**

サブスクライブするものは、代替目的地の仮想デバイスまたはオブジェクトに供給されるべきサブスクリプション・データ・メッセージを要求し得る。代替目的地はまた、「ミラーされた」オーディオ・ノード構造を含み得る。ミラーされたオーディオ・ノード構造は、GUI、コントロール面、またはオートメーションデバイスといったオーディオ関連デバイスが、サブスクライブするものとサブスクライブされるものとの間にコンポーネントとの間でアドレス衝突を起さずに他の任意のオーディオ関連デバイスにサブスクライブすることを可能にする。

**【0120】**

1つの例において、サブスクライブするもののオブジェクトのSVのリストは、個々のSV IDのそれぞれがサブスクライブデータメッセージで送達される必要がないようにサブスクライブされるもののオブジェクト上のSVのリストをミラーする。オブジェクト全体へのサブスクリプションはまた、サブスクリプションがセンサデータおよび非センサデータのすべてに対してであるか、すべてのセンサまたはすべての非センサに対してであるか否かを特定し得る。仮想デバイスへのサブスクリプションは、サブスクリプション・メッセージを仮想デバイスにアドレッシングすることによって行われ得る。仮想デバイスに包含されているSVのすべて、および仮想デバイスに包含されているすべてのオブジェクトがサブスクライブされ得る。サブスクライブするものの仮想デバイス構造は、サブスクライブされるものの仮想デバイスの構造と同一であり得る。

**【0121】**

サブスクリプションは、追加およびリムーブメッセージを用いて追加、または取り除かれ得る。例えば、サブスクライブ・メッセージは、サブスクリプションを追加するために用いられ得、アンサブスクライブ・メッセージは、サブスクリプションを取り除くために用いられ得る。サブスクライブ・メッセージは、サブスクライブされるものが、1つ以上の識別されたSVに対するサブスクライブされるもののサブスクライブのリストに、サブスクライブするものを追加するように命令する。これに反して、アンサブスクライブ・メ

10

20

30

40

50

ッセージは、サブスクライブされるものが、1つ以上の識別されたSVに対するサブスクライブのリストから、サブスクライブするものを取り除くように命令する。例えば、同一のオブジェクト上の1つのSVから別のSVにサブスクリプションを変化することが所望されるとき、第1のサブスクリプションのUnSubscribeおよび新しいサブスクリプションに対するSubscribeの双方が送達され得る。

#### 【0122】

複数のSVはまた、SVUnSubscribeAllメッセージといった単一のアンサブスクライブ・セット・メッセージにおいてアンサブスクライブされ得る。このようなアンサブスクライブ・セット・メッセージは、オブジェクトまたは仮想デバイス下のすべてのSVからアンサブスクライブするために用いられ得る。仮想デバイスまたはオブジェクトは、複数のオブジェクトを包含するとき、アンサブスクライブ・セット・メッセージは、アンサブスクライブ・メッセージで識別されるオーディオ・ノード構造の対応する部分において、すべてのSVを含み得る。サブスクライブ・メッセージと同様に、送達されるべきデータの量を最小にするために、アンサブスクライブするものは、サブスクライブするものおよびサブスクライブされるもののオブジェクトアドレスおよびSV識別子が相互交換して、用いられるようにミラーされたオーディオ・ノード構造を有し得る。

10

#### 【0123】

##### エラー処理

エラーメッセージは、プロトコルエラーおよびデバイスエラーとして一般的にカテゴリ化される。オーディオ関連デバイスが奇形、さもなければ解読不能のメッセージを受け取るとき、プロトコルエラーが起こる。このタイプのエラーの例は、誤った長さフィールドを有するメッセージである。最良の形態として、デバイスは、すべてのこのようなエラーの検出を試み、報告すべきである。デバイスが、デバイス特定の理由のために要求された動作を行うことをできないとき、デバイスエラーが起こる。このタイプのエラーの例は、一時的にビジー、さもなければ利用不可能であるサブシステムと通信しようとする試みであり得る。別のエラーの形としては、アプリケーションエラーである。アプリケーションの例は、オーディオ関連デバイスの単一のパスにおけるどこかでクリップすることである。

20

#### 【0124】

オーディオ関連デバイスは、メッセージを受け取ると、エラーを報告し得る。エラーを通信するメカニズムは、エラー認識メッセージおよびエラーサブスクリプションを含む。エラー認識メッセージは、エラーを送達デバイスに報告するために用いられ得る。エラー認識メッセージは、メッセージ・ヘッダー・フラグにおいてビットの形式を取る。デバイスが、エラーである、または結果的にエラーになるメッセージを受け取るとき、エラー情報メッセージは、送達するものに戻され得る。

30

#### 【0125】

エラーサブスクリプションは、当事者が、オーディオ関連デバイスによって生成されるエラー認識メッセージにサブスクライブすることを可能にする。エラーサブスクリプションは、動作中に、当事者がすべてのエラーをモニタ、またはログを取り得るように用いられ得る。エラーサブスクリプションでは、サブスクライブされるものによって送達されるすべてのエラーメッセージがサブスクライブするものに複製され得る。エラーサブスクリプションデータは、エラーメッセージおよびエラーを起したオリジナルメッセージを含み得る。そのメッセージのサイズを縮小するために、オリジナルのメッセージが長すぎるとき、オリジナルのサブスクリプション・メッセージが最大のサイズに制限され得、それによってエラーを起したオリジナルのメッセージが結果的にトランケーションされる。

40

#### 【0126】

##### イベント・ロギング

イベント・ロギングは、ノード上の重大な出来事を格納し、報告する方法である。イベントは、ベニュー・プリセットのリコールといった情報的な性質なものであり得、または、電源異常のようなより深刻なエラーであり得る。イベントは、ノードによって記録され

50

、コントロール・パネル・ビューといったイベント・ログ・クライアント・イベントを読み出すときまで、不揮発性ストレージに格納され得る。

【0127】

イベント・ロギングは、多くの点においてエラー処理とは異なり得る。

a) イベント・ロギングは、良性イベントが可能である。 ; エラー処理は、故障モードだけを報告する。

b) イベント・ロギングは、クライアントがオンラインでないとき、イベントを保存し得る。 ; エラー処理は、クライアントがオンラインでないとき、イベントを保存し得えない。

c) イベント・ログは、「オン・デマンド」を供給される。エラーは、サブスクリプションを介してプッシュされる。

d) イベントは、カテゴリを割り当てられ、優先権を与えられ得る。エラーは、カテゴリおよび優先権を与えられ得ない。

【0128】

イベント・ログは、メッセージを用いてXMLファイルといった所定のフォーマット、またはバイナリフォーマットでエクスポートされ得る。XMLは、オーディオ関連デバイスのメモリ・リソースが、結果生じるファイルを生成するために十分である所定のフォーマットである。制限されたリソースを有するオーディオ関連デバイスは、バイナリフォーマット、または独自に圧縮されたフォーマットさえ用い得る。

【0129】

XMLイベント・ログは、FTPプロトコルといったファイル転送プロトコルを介して転送され得る。IP対応でないネットワーク上で、ログをエクスポートすることを必要とするオーディオ関連デバイスは、メッセージを介してイベント・ログエントリを転送するバイナリ・イベント・ログ・フォーマットを用い得る。あるいは、イベント・ログは、Web Browser (登録商標) を介して直接的にアクセスされ得る。

【0130】

ワークグループ

ノードは、ワークグループにおいて構成され得る。一般的に、ノードのワークグループは、大きな共鳴板といった比較的複雑な機能を有するオーディオ関連デバイスに配置される。オーディオ関連デバイス内において、ノードは、協調動作機能に統合される。ワークグループは、階層的であるが、オーバーラップし得ない。各ワークグループは、ストリング名によって識別され得、階層のレベルは、「\」といった所定の文字によって分けられ得る。ワークグループの例は、コントロール面の機能性を表すノードおよびオーディオトラックの機能性を表すノードといった2つのノードを含むコンソールといったオーディオ関連デバイスであり得る。双方のノードがアクティブであるとき、コンソールは、「全体 (whole)」であると考えられ得る。このオーディオ関連デバイスのワークグループは、「FOH Mixer」として指定され得、ノードは、「FOH Mixer \ Surface」および「FOH Mixer \ Audio」と呼ばれ得る。

【0131】

ワークグループはまた、オーディオ・システムをセグメント化するために用いられ得る。オーディオ・システムのセグメント化は、オーディオ・システムにおけるサブシステムへのアクセスを制御するために、それ以外の場合には、オーディオ・システムの機能性をサブシステムに分けるために所望され得る。例えば、スタジアムのオーディオ・システムのデザインでは、トップレベルのワークグループは、「Stadium」として識別され得る。中間のレベルのワークグループは、「Sky Boxes」として識別され得、個々のスカイボックスのそれぞれは、それ自身のワークグループとして識別され得る。従って、この例において、スカイボックスナンバー10におけるオーディオ関連デバイスのワークグループは、「Stadium \ Sky Boxes \ Sky box 10」であり得る。発見は、スカイボックスの内側のコントロール面が、そのスカイボックス内のノードだけを発見するように選択し得るよう分離され得るが、ワークグループ「Stadium \

10

20

30

40

50

「SkyBoxes」に割り当てられるコントロール面は、スカイボックスのいずれかにおける任意のオーディオ関連デバイスを発見し得る。発見は、より詳細に後述される。

【0132】

ワークグループによるセグメント化は、アプリケーション層内の発見プロセスに限定され得る。他のメッセージはなお、異なるワークグループ間でパスされ得る。ワークグループの割り当ては、属性を有し得る。「名前ストリング」といったワークグループストリングは、ノード・マネージャ仮想デバイスの属性であり得る。このストリングは、セットされ、他のオーディオ関連デバイスによってクエリ得る。

【0133】

パフォーマンス機器マネージャ(「PEM」)

図5は、PEMおよびオーディオ・システムの一部のブロック図である。PEM500は、1つ以上のメモリデバイスに格納されている複数の命令およびデータで形成され得る。メモリデバイスは、取り出されるためのデータを保存する任意の媒体であり得、ハードディスクドライブ、コンパクトディスクドライブ、デジタル多用途ディスクドライブ、RAM、ROM、PROM、EEPROM、フラッシュメモリ、または他の任意のデジタル格納装置を含み得る。PEM500は、オーディオ・システムにおける任意のオーディオ関連デバイス上に常駐する。

【0134】

PEM500は、メモリに格納されているアプリケーション層502およびネットワーク・インターフェース504を含み得る。ネットワーク・インターフェース504は、PEM500とオーディオ関連デバイス506、または複数のオーディオ関連デバイスとの間で通信が可能であるように構成され得る。PEMのネットワーク・インターフェース504は、先述されたオーディオ関連デバイス506のネットワーク・インターフェースと同様な態様で動作し得る。例えば、PEM500は、デバイス506に直接的に接続され得るか、オーディオ・ネットワーク120を用いてデバイス506と通信し得る。オーディオ・ネットワーク120はまた、先述されたような第1のスイッチ330を介して第1のオーディオ・ネットワーク334に結合され得る。PEM500は、先述されたようなEthernet(登録商標)508、FireWire510、USB512、RS232 514、またはRS485 516といった任意の通信リンクを用いてデバイス506と通信し得る。加えて、PEM500は、先述されたオーディオ・ストリーム・モジュール322およびオーディオ・サービス・モジュール324を含み得る。

【0135】

ネットワーク・インターフェース504はまた、ルーティング・テーブル518を含み得る。ルーティング・テーブル518は、ノードとPEMとの間でデータをルーティングするために用いられ得る。ルーティング・テーブル518は、オーディオ・ネットワーク120上のノード間でルートの一覧、およびノード間で送達されるネットワーク情報を格納し得る。ルーティング・テーブル518は、重複ネットワークを可能とするノードのセット間の複数のルートを含み得る。ルートは、最短トランジット時間に基づいて選択され得る。いくつかの要因が、最短トランジット時間を決定することにおいて考慮され得る。例えば、ノード間の帯域幅は、特定のルートの「コスト」における要因であり得る。加えて、ホップ・カウントおよびメッセージ・サイズも考慮され得る。これらの要因に基づいて、「コスト」は、各ルートに関連付けられ得、例えば、異なるルートが、コストを最小、または所定のしきい値未満にコストを維持するように選択され得る。

【0136】

アプリケーション層が動作しているデバイスの能力に依存して、PEMのアプリケーション層502は、「厚い」アプリケーション層502a、または「薄い」アプリケーション層502bのいずれかであり得る。「厚い」PEMでは、デバイスのオーディオ・ノード構造全体が取得され、オーディオ・ノード構造全体を表す適切なコントロール・パネル・ビューのセットを決定するために分析され得る。しかしながら、「薄い」PEMは、表示されるべきコントロール・パネル・ビューに対して必要とされるオーディオ・ノード構

10

20

30

40

50

造の一部だけを取得し、分析し得る。従って、PEMによって取得され、かつ、分析されたコントロール・パネル・ビューは、PEM 500が動作しているオーディオ関連デバイス506の演算能力に依存する複雑さのスライド・スケールに基づき得る。いくつかの状況において、「厚い」PEMは、処理能力を保存するために、またはネットワーク・トラフィックを制限するために「薄い」PEMと同様な態様で動作し得る。例えば、デバイスにクエリする前に、「厚い」PEMは、オーディオ・ネットワーク120上のトラフィックの量が多いことを認識し得、それ故に、ネットワーク・トラフィックを減らすためにオーディオ関連デバイス506のオーディオ・ノード構造の一部だけを取得し得る。

#### 【0137】

PEMのアプリケーション層502は、GUIのコア520、プラグイン・インターフェース522、および複数のプラグイン523を含み得る。プラグイン523はジェネリック・プラグイン524および1つ以上のプロダクト・プラグイン526を含み得る。GUIのコア520は、デバイス506のオーディオ・ノード構造にクエリするように構成され得る。このオーディオ・ノード構造データは、デバイス506に対する適切なコントロール・パネル検索するためにプロダクト・プラグイン526および/またはジェネリック・プラグイン524にブロードキャストされ得る。デバイスにおけるオーディオ・ノード構造の各部分が、既知で、かつ、所定のコントロール・パネル・ビューを有するか否かに関わらず、(プロダクト・プラグイン526および/またはジェネリック・プラグイン524とともに)GUIのコア520は、ディスプレイドライバ528に、ディスプレイがオーディオ・ノード構造の各レベルに対してコントロール・パネル・ビューを表示するために駆動するように指示し得る。コントロール・パネル・ビューが、オーディオ・ノード構造全体に対して既知であるとき、所定のコントロール・パネル・ビューは、ディスプレイドライバ525を用いて表示され得、デバイスをモニタし、かつ、制御するために用いられ得る。しかしながら、オーディオ・ノード構造のすべて、または一部が未知であるとき、「ジェネリック」コントロール・パネル・ビューまたはオーディオ・ノード構造の未知の部分に対する「ジェネリック」GUIのアスペクトが動的に生成され、ディスプレイドライバ528を用いて表示され得る。「ジェネリック」コントロール・パネル・ビュー、または所定のコントロール・パネル・ビューに含まれるGUIのアスペクトは、デバイスを制御するために用いられ得る。

#### 【0138】

GUIのコア520は、ネットワーク・インターフェース504を介してデバイス506と通信し得る。GUI520、ジェネリック・プラグイン524およびプロダクト・プラグイン526間の通信は、プラグイン・インターフェース522を通過し得る。ジェネリック・プラグイン524およびプロダクト・プラグイン526はまた、プラグイン・インターフェース522を用いてお互いに通信し得る。

#### 【0139】

図6は、GUIのコア520の例のブロック図である。GUIのコア520は、ノード分解能モジュール600を含み得る。(直接的に、またはネットワークを介して)PEMに接続された各オーディオ関連デバイスは、一意のノード識別(「ID」)にも用いて識別され得るノードである。ノード分解能モジュール600は、PEM500に接続されている各デバイスにノードIDを、またはノード・アドレスを割り当てるノードIDエンジン602を含み得る。新しいデバイスが、PEM500に接続されているとき、ノードIDエンジン602は、ノードIDを新しいノードに割り当て得る。このようなノードIDを割り当てるために用いられ得る方法が後述される。

#### 【0140】

GUIのコア520はまた、システム・プラグイン・ノード・マネージャ604を含み得る。システム・プラグイン・ノード・マネージャ604は、プラグイン・インターフェース522を介してプロダクト・プラグイン526およびジェネリック・プラグイン524と通信し得る。システム・プラグイン・ノード・マネージャ604は、GUIマネージャ606、システムワイド設定モジュール608および制御ライブラリ610を含み得る

10

20

30

40

50

## 【0141】

GUIマネージャ606は、コントロール・パネル・ビューのGUIフォーマットのモードを管理するように構成され得る。例えば、GUIマネージャ606は、ベニュー・ビューとプロダクト・ビュー間で切り替え得る。ベニュー・ビューでは、コントロール・パネル・ビューのGUIは、同時に、ユーザがオーディオ・システムにおける複数のデバイスをビューすることを可能にする。例えば、ベニュー・ビューは、オーディオ関連デバイス間の相互関連と同様にシステムにおけるすべてのオーディオ関連デバイスを表示し得る。このモードは、フローティング・ウィンドウを用いてセットアップされ得る。各フローティング・ウィンドウは、デバイスに関しての情報を表示する。プロダクト・ビューでは、単一のオーディオ関連デバイスに対するコントロール・パネル・ビューのGUIが表示され得る。いくつかの例において、プロダクト・ビューは、ユーザが、オーディオ関連デバイスに対するさまざまな仮想デバイスまたはオブジェクトの間を切り替えることを可能にするメニューまたはタブ階層を有し得る。ユーザは、必要に応じプロダクト・ビューとベニュー・ビュー間で切り替え得る。例えば、タブレット・コンピュータ上で見られるといったタッチ・スクリーン・インターフェースを用いるとき、ユーザは、プロダクト・ビューを用い得、また、マウスおよびキーボードを有するコンピュータを用いるとき、ユーザは、ベニュー・ビューを用い得る。図27は、システムのベニュー・ビューの例を示し、一方、図28、29は、プロダクト・ビューの例を示す。

10

## 【0142】

システムワイド設定モジュール608は、後に使用されるためにオーディオ・システムのすべて、または一部の設定を格納するように構成され得る。これらの保存された設定は、後にロードされ、セーブされた構成を再格納するために各オーディオ関連デバイスに送達され得る。GUIマネージャ606はまた、それぞれのプロダクト・プラグイン526およびジェネリック・プラグイン524の各々によってすべての設定およびコントロール・パネル・ビューが保存されることを指示し得る。

20

## 【0143】

GUIマネージャ606はまた、カスタム・コントロール・パネル・ビューを生成する能力を提供するように構成され得る。カスタム・コントロール・パネル・ビューは、プロダクト・プラグイン526およびジェネリック・プラグイン524からの所定のGUIのAspectまたはジェネリックGUIのAspectを用いて異なる複数のデバイスからデータ変数を有するように構成され得る。GUIマネージャ606はまた、複数のホットボタンを有するように構成され得る。ホットボタンは、プロダクト・プラグイン526およびジェネリック・プラグイン524における特定のデバイスの特定のコントロール・パネル・ビューにマッピングされ得るか、または、ユーザによって作成されたカスタム・コントロール・パネル・ビューにマッピングされ得る。

30

## 【0144】

カスタム・コントロール・パネル・ビューを作成するとき、異なるプロダクト・プラグインからの特定の格納されている所定のGUIのAspectが、格納されている所定のコントロール・パネル・ビューから作成中のカスタマイズされたコントロール・パネル・ビューにドラッグされ得る。例えば、格納されている所定のコントロール・パネル・ビュー上のスライダは、マウスでクリックされ、カスタマイズされたコントロール・パネル・ビューに移動され得る。加えて、カスタマイズされたコントロール・パネル・ビュー内において、ドラッグされ、ドロップされたGUIのAspectは、異なるGUIのAspectに変化、または「変形」され得る。例えば、ドラッグされ、ドロップされたスライダは、ノブ、ボタン、ロータリ・エンコーダ、リスト・ボックス等に変化され得る。変化されたGUIのAspectはまた、異なる値を有するように構成され得る。例えば、0~100%のレンジを有し、ドラッグされドロップされるスライダであって、ボタンに変化されたスライダの場合には、このボタンは、20%は押下した状態で、60%は押下していない状態を有するように構成され得る。

40

50

## 【 0 1 4 5 】

G U I マネージャ 6 0 6 はまた、ベニュー機能コントロールを含み得る。ベニュー機能は、デバイスに含まれているベニュー設定の所定の複数のグループを参照し得る。ベニュー設定は、オーディオ関連デバイスのメモリにベニュー・テーブルとして格納され得る。ベニュー・テーブルに含まれているベニュー設定のそれぞれは、オーディオ関連デバイスのオーディオ・ノード構造の対応するコンポーネントによって識別され得る。従って、ベニュー設定のグループが選択されたとき、ベニュー設定のそれぞれは対応するコンポーネントに存在する現存の設定を置換し得る。

## 【 0 1 4 6 】

例えば、デバイスのグループは、演奏に対する E Q 設定のグループおよび演劇に対する異なる E Q 設定のグループを有し得る。別の例において、いくつかのオーディオ関連デバイスは、あるベニューにおいて無効にされ、別のベニューにおいて有効にされ得る。G U I マネージャ 6 0 6 は、ベニュー・テーブルからのベニュー設定のグループの 1 つの選択することを提供する機能を含み得る。ユーザによってベニュー設定のグループの 1 つが選択されると、ベニュー設定のグループ識別子を含むセット・ベニュー・メッセージが P E M によって、オーディオ関連デバイスのすべてに送達され得る。ベニュー設定グループに対応するベニュー設定のグループを有するこれらのオーディオ関連デバイスは、セット・ベニュー・メッセージに示されているベニュー設定のグループを有する動作に切り替え得る。このようなベニュー設定のグループを有しないこれらのオーディオ・デバイスは、セット・ベニュー・メッセージを無視し得、および/または無効にされる。

## 【 0 1 4 7 】

制御ライブラリ 6 1 0 がまた、G U I のコア 5 2 0 に提供され得る。制御ライブラリ 6 1 0 は、ユーザ定義の S V に関連付けられ得る G U I のコントロール・パネル・ビューおよび/または G U I のアスペクトのセットを含み得る。例えば、ユーザは、制御ライブラリ 6 1 0 からスライダバーを表す G U I のアスペクトを引き、そのスライダバーをオーディオ・システムにおける異なるオーディオ関連デバイスのいくつかの S V と関連付け得る。従って、ユーザは、G U I コントロールパネルビュー上のいくつかのスライダ(または他のコントロール)を同時に制御するために「マスター・コントローラ」としてカスタムスライダバーを用い得る。加えて、ユーザは、G U I のコントロール・パネル・ビューを用いてマスター・コントローラに掘り下げ得、マスタースライダを用いて調整されているデータ変数のオフセットを調整する。

## 【 0 1 4 8 】

G U I のコア 5 2 0 はまた、サブスクリプション・マネージャ 6 1 2 を含み得る。サブスクリプション・マネージャ 6 1 2 は、コントロール・パネル・ビューおよび/またはデバイスにおいて変化されるデータの更新および通知を管理し得る。例えば、G U I のコントロール・パネル・ビューは、関連した S V 値を有するスライダを包含し得る。ユーザが、G U I のコントロール・パネル・ビューにおけるスライダを移動(従って、スライダに関連した S V の実際の値を変化)するとき、サブスクリプション・マネージャ 6 1 2 は、サブスクリプションを管理し得る。例えば、サブスクリプション・マネージャ 6 1 2 は、データが適切なフォーマットで、対応する S V に以前にサブスクライブした対応するオーディオ関連デバイスに送達されたことを確認し得る。

## 【 0 1 4 9 】

G U I のコア 5 2 0 はまた、特定のコントロール・パネル・ビュー上でユーザに可視的である G U I コントロール、および G U I コントロールに関連し S V 値を追跡するリンク・マネージャ 6 1 4 を含み得る。リンク・マネージャ 6 1 4 は、リンク・エンジン 6 1 6 およびイベント・エンジン 6 1 8 を含み得る。イベント・エンジン 6 1 8 は、コントロール・パネル・ビューまたは対応するオーディオ関連デバイスにおける任意の S V の実際の値への変化を通知され得る。コントロール・パネル・ビューにおける S V への変化がなされるとき、リンク・エンジン 6 1 6 は、変化された S V に関連づけられた任意のコントロール・パネル・ビューおよび/またはオーディオ関連デバイスに通知し得る。通知に回答

10

20

30

40

50

して、コントロール・パネル・ビューは、オーディオ関連デバイスからではなく、対応するプロダクト・プラグイン526またはジェネリック・プラグイン524におけるコントロール・パネル・ビューからSVのアップデートされた実際の値を要求し得る。従って、さまざまなコントロール・パネル・ビューがPEM500におけるデータでアップデートされるので、オーディオ関連デバイスへのネットワーク・トラフィックおよびオーディオ関連デバイスからのネットワーク・トラフィックが最小にされ得る。

#### 【0150】

同様に、SVが、オーディオ関連デバイスで変化するとき、デバイスは、1つのコントロール・パネル・ビューだけにサブスクライブされ得る。従って、1つのコントロール・パネル・ビューが、デバイスからのメッセージによってアップデートされるとき、イベント・エンジン618は、変化されたSVの実際の値を表示する対応するすべてのコントロール・パネル・ビューに通知し得る。これに回答して、対応するコントロール・パネル・ビューは、デバイスからではなく、アップデートされたコントロール・パネル・ビューからアップデートされたSVの実際の値を要求する。例えば、各オーディオ関連デバイスは、構成状態を含み得る。構成状態は、オーディオ関連デバイスにおけるSVが変化されたことを表す単一の数であり得る。構成状態は、他の診断およびメンテナンスに関連したアクティビティと同様に、重大な変化が別のGUIのコントロール・パネル・ビューからのデバイスになされたか否かを決定するために有用であり得る。

10

#### 【0151】

##### 発見エンジン

GUIのコア520はまた、プロダクト・プラグイン526またはジェネリック・プラグイン524と通信し得る発見エンジン620を含み得る。オーディオ関連デバイスのオーディオ・ノード構造における1つ以上のコンポーネントが未知であるとき、発見エンジン620は、プロダクト・プラグイン526またはジェネリック・プラグイン524によって開始され得る。オーディオ関連デバイスのオーディオ・ノード構造における未知のコンポーネントは、対応する格納されている所定のGUIのコントロール・パネル・ビューおよび/またはGUIのアスペクトを含まない仮想デバイス、オブジェクト、またはSVであり得る。言い換えれば、コンポーネントは、オーディオ・システム内においてあらかじめ決定されない。

20

#### 【0152】

有効であるときには、発見エンジン620は、オーディオ・ノード構造に関してオーディオ関連デバイスにクエリし得る。発見エンジン620は、SVおよび/またはそれに関連した属性、オブジェクトおよび/またはそれに関連した属性、ならびに、仮想デバイスおよび/またはそれに関連した属性のそれぞれを決定するSV発見モジュール622、オブジェクト発見モジュール624および仮想デバイス発見モジュール626を含み得る。

30

#### 【0153】

静的オーディオ・ノード構造構成を有するオーディオ関連デバイスにとって、このようなデバイスのオーディオ・ノード構造はプロダクト・プラグインによって既に既知であり得るので、発見エンジン620での発見は必要でないかもしれない。このような状況では、現在のSV実際の値のクエリが要求されるすべてである。しかしながら、オーディオ関連デバイスのオーディオ・ノード構造のいずれかの部分が既知でないとき、発見エンジン620での発見がインプリメントされ得る。加えて、いくつかのオーディオ関連デバイスは、構成可能な仮想デバイスを含む。構成可能な仮想デバイスは、オブジェクトおよび/またはSVの異なる構成を有するように、ユーザによって構成され得る。従って、オーディオ・ノード構造は、構成可能であり、このような構成可能な仮想デバイス内のオブジェクトおよびSVが、発見エンジン620を用いて発見されることが必要とされ得る。

40

#### 【0154】

未知である静的オーディオ・ノード構造の例が互換性の問題として上げられる。例えば、ツアーの際にツアーのオーディオ・システムのための新しいオーディオ関連デバイスを受け取るツアーのサウンドプロバイダを考慮する。ツアーのサウンドプロバイダによって

50

受け取られる新しいオーディオ関連デバイスは、新しいオーディオ関連デバイスによって置換される現存のデバイスに存在しないさらなる機能および/または能力を提供する新しいファームウェアのリビジョンを有する。ツアーのサウンドオペレータが、オーディオ・ノード構造、ならび新しいオーディオ関連デバイスに備えられている対応するGUIのコントロール・パネル・ビューおよび/またはGUIの aspek tの新しいコンポーネントをロードしてプロダクト・プラグインをアップデートしないとき、新しい特徴へのアクセスは可能ではないかもしれない。また、一層悪いことに、ツアーのサウンドオペレータは、現存の特徴を制御し得ないかもしれない。発見エンジン620は、新しいデバイスのオーディオ・ノード構造を発見するメカニズムと、および新しいデバイスの特徴のすべてをビュー、制御およびエディットするデフォルト・エディタが作成されることを可能にする情報を提供するメカニズムとを提供することによってこのような問題を克服し得る。

10

## 【0155】

発見エンジン620は、オーディオ関連デバイスのオーディオ・ノード構造内のコンポーネントを発見するために「獲得」方法を用い得る。オーディオ関連デバイスは、オーディオ・ノード構造に関連した情報を提供するために「Info return」メッセージで応答し得る。未知の仮想デバイス、オブジェクト、またはSVがオーディオ関連デバイスにおいて存在とき、発見エンジン620は、詳細な属性のためにオーディオ関連デバイスにクエリするように配置され得る。このタイプの発見は、未知の発見(unknown discovery)と呼ばれる。有効なときには、発見エンジン620は、オーディオ関連デバイスのオーディオノード構造全体、またはオーディオ関連デバイスのオーディオノード構造の一部を発見し得る。加えて、発見エンジン620は、オーディオ・ノード構造において特定されたコンポーネント、つまり、1つ以上の特定された仮想デバイス、オブジェクトまたはSVを発見するように有効にされ得る。

20

## 【0156】

一度オーディオ関連デバイスがアドレス指定され、かつ、オーディオ・ネットワーク上で利用可能になると、オーディオ・ノード構造のクエリが用いられ得る。オーディオ・ノード構造のクエリは、任意のオーディオ関連デバイスのオーディオ・ノード構造を決定するために用いられ得る。オーディオ・ノード構造のクエリは、呼び出しおよび応答を有効にする所定の要求メッセージを用い得る。オーディオ関連デバイスのクエリの結果として、オーディオ関連デバイスの機能を表す各仮想デバイス、オブジェクトおよびSVのそれぞれが取得され得る。

30

## 【0157】

オーディオ・ノード構造の不明のコンポーネントが識別されるときはいつでも、発見エンジン620は、未知のコンポーネントのより詳細な属性情報を発見するように、かつ、オーディオ・ノード構造全体がなんであるのかと、または、未知のコンポーネントが何を表すのかを確実にするように有効にされ得る。従って、オーディオ・ノード構造のクエリによって提供される呼び出しおよび応答機能および/または発見エンジン620によって提供される発見機能は、オーディオ・ノード構造のコンポーネントのすべてと、および/またはオーディオ・ノード構造内のコンポーネントの詳細な属性とを確実にするために用いられ得る。

40

## 【0158】

発見エンジン620が、オーディオ・ノード構造全体を決定するように有効されるとき、オーディオ・ノード構造全体のクエリ要求は、単一メッセージで発見エンジン620に通信され得る。発見エンジン620は、それから、一連のオーディオ・ノード構造のクエリを用いて当該のオーディオ関連デバイスと通信し得る。呼び出しへの応答は収集され、一時的に格納され得るか、その応答は単にプラグイン523に転送され得る。応答が収集されるとき、発見エンジン620は、その応答に提供される仮想デバイス、オブジェクトおよび/またはSVに含まれている属性に基づいて当該のオーディオ関連デバイスのオーディオ・ノード構造全体を構築し得る。オーディオ・ノード構造全体は、それから、単一のメッセージでプラグイン523に提供される。

50

## 【 0 1 5 9 】

あるいは、発見エンジン 6 2 0 は、オーディオ・ノード構造に存在する、未知のコンポーネントに関連したより詳細な情報を取得するために有効にされ得る。このシナリオでは、仮想デバイス、オブジェクトまたは S V は、発見エンジン 6 2 0 には未知であると識別され得る。発見エンジン 6 2 0 は、それから、1つ以上のコンポーネント特定のオーディオ・ノード構造のクエリを用いて当該のオーディオ関連デバイスと通信し得る。クエリに対する応答は、未知のコンポーネントの詳細な属性を含み得る。詳細な属性は、当該のコンポーネントを構築、または定義するために用いられ得、かつ、たった今定義された未知のコンポーネントを単一のメッセージで提供するために用いられ得る。あるいは、詳細な属性が取得され、発見エンジン 6 2 0 によるさらに処理されることなしに、プラグイン 5 2 3 に提供され得る。

10

## 【 0 1 6 0 】

オーディオ・ノード構造を決定するためのオーディオ関連デバイスとの初期の通信は、プラグイン 5 2 3 の 1 つを用いて行われ得る。オーディオ関連デバイスのオーディオ・ノード構造を取得するために、オーディオ関連デバイスは、先ず、プラグイン 5 2 3 の 1 つによって、そのノード上の仮想デバイスのリストに対してクエリされ得る。メッセージ `GetVdList` といった所定のメッセージがオーディオ関連デバイスのノード・マネージャ仮想デバイスに送達され得る。ノード・マネージャは、それ自身を含むそのノード上の仮想デバイスのリストで返答し得る。一度仮想デバイスが識別されると、プラグイン 5 2 3 によって、仮想デバイスのすべてが既知であるかどうかが決まる。いずれかの仮想デバイスが未知であるとき、プラグイン 5 2 3 は、オーディオ・ノード構造全体または未知の仮想デバイスを決定するために発見エンジン 6 2 0 を有効にし得る。あるいは、プラグイン 5 2 3 は、オーディオ・ノード構造を決定し得る。別の代替において、発見エンジンは、オーディオ関連デバイスとすべての通信を行い、受け取った情報をプラグイン 5 2 3 にパスし、プラグイン 5 2 3 の指示でオーディオ関連デバイスとさらなる通信を行い得る。

20

## 【 0 1 6 1 】

ノード上の 1 つ以上の仮想デバイスが未知であるとき、リスト内の未知の仮想デバイスのそれぞれは、未知の仮想デバイスに含まれている S V およびオブジェクトを決定するためにオーディオ・ノード構造のクエリを用いてクエリされ得る。仮想デバイスレベル S V を取得するために、`GetSVList` メッセージといった仮想デバイスにアドレス指定される所定の獲得 S V メッセージは、仮想デバイスで用いられる S V クラスのリストにクエリするために用いられ得る。獲得 S V メッセージは、仮想デバイス上の各 S V に対するクラス ID を含むリストを戻し得る。仮想デバイスにおけるオブジェクトを決定するために、`GetObjectList` メッセージといった仮想デバイスにアドレス指定される所定の獲得 S V メッセージが用いられ得る。仮想デバイスに含まれるオブジェクトのリストは、獲得オブジェクト・メッセージに回答して返送され得る。オーディオ・ノード構造のクエリは、発見エンジンまたはプラグインによって開始され得る。あるいは、リストではなく、仮想デバイス、オブジェクトおよび S V が個々にクエリされ得る。

30

## 【 0 1 6 2 】

最終的に、仮想デバイスに含まれている S V の実際の値が取得され得る。実際の値は、`MultisVGet` メッセージといった所定の獲得実際の値メッセージのクエリを用いて取得され得る。獲得実際の値メッセージに回答して、仮想デバイスに含まれている S V の実際の値が返送され得る。あるいは、S V がサブスクライブされ得る。実際の値がアップデートされるときはいつでも、サブスクライブされた S V が自動的に提供される。

40

## 【 0 1 6 3 】

先述は、未知である構成可能（または動的）な仮想デバイスまたは静的な仮想デバイスの存在および/または未知のコンポーネントを含む構成可能（または動的）な仮想デバイスまたは静的な仮想デバイスの存在を仮定した。オーディオ・ノード構造が未知である静的な仮想デバイスの場合において、オブジェクトおよび S V が既知であるので、仮想デバ

50

イスへのクエリ以上にさらなるクエリは必要とされないかもしれない。従って、プラグインまたは発見エンジン 6 2 0 は、獲得 S V メッセージ ( G e t S V L i s t )、または獲得プロジェクトメッセージ ( G e t O b j e c t L i s t ) を送達する必要はないかもしれない。

**【 0 1 6 4 】**

オブジェクトの受け取られたリストにおいて識別されるオブジェクト・クラスが既知であるとき、S V の実際の値へのクエリだけが、上述の 2 つの方法のうちの 1 つを用いて必要とされ得る。仮想デバイスにおける、未知である各オブジェクトに対して、サブオブジェクトおよび S V のリストが、獲得 S V メッセージおよび / または獲得プロジェクトメッセージを用いて要求され得る。未知のオブジェクト上で行われる必要があるかもしれない他の動作は、オブジェクトにおけるすべての S V の動的属性を発見することである。動的属性は、オブジェクト自身によってセットされ得る。このようなオブジェクトの動的属性を取得するメカニズムは、S V G e t D y n A t t r i b u t e s とした所定の獲得動的属性メッセージを有し得る。

10

**【 0 1 6 5 】**

未知の S V が識別される際には、S V G e t S t a t i c A t t r i b u t e s とした所定の獲得動的属性メッセージが未知の S V を包含するオブジェクトまたは仮想デバイスに送達され得る。これに回答して提供される静的属性は、未知の S V を定義するために用いられ得る。

**【 0 1 6 6 】**

あるいは、仮想デバイスに対するより効果的な発見メカニズムがまた用いられ得る。所定のオーディオ関連デバイスノード構造のクエリ・メッセージのセットがオーディオ関連デバイスに送達され得る。所定のクエリ・メッセージのセットは、凝縮フォーマットのオーディオ・ノード構造を要求し得る。これに回答して、各仮想デバイスは、オーディオ関連デバイスの機能性を十分に記載した返答を提供し得る。機能性は、例えば、仮想デバイス内のあらかじめ決定されるオブジェクトおよび S V メンバーの登録したクラス ID によって十分に記載され得る。これらのオーディオ関連デバイスノード構造のクエリ・メッセージのいずれかが、未知であるオブジェクトまたは S V クラスを返送するとき、先述されるようなそのコンポーネントのより詳細な発見が行われ得る。

20

**【 0 1 6 7 】**

ゾーン・コントローラであるオーディオ関連デバイスの発見の例がこれから記載される。ゾーン・コントローラは、備え付けられたオーディオ内の単一なゾーンに対してオーディオ・ソース・セレクトおよび利得制御を提供し得るシンプルなウォール・コントローラである。ゾーン・コントローラの機能性は、オーディオ能力を持たないユーザ・インターフェースオブジェクトとしてモデル化され得る。ゾーン・コントローラは、利得およびソース選択機能を行うルータ・オブジェクトに対して 2 つのサブスクリプションを有するように構成され得る。プロセッシングが異なるノード上の別々の処理オブジェクト用いて行われ得る。

30

**【 0 1 6 8 】**

構成不可能な仮想デバイスを有するスタンドアロンノードとしてモデル化されるとき、ゾーン・コントローラは、以下のとおりに見なされ得る。S V C l a s s G a i n S V とした利得 S V は、オーディオ処理オブジェクトにおける利得 S V にリンクされ得る。S V C l a s s C M I n p u t としたソース選択 S V は、オーディオ処理オブジェクトにおけるルーティング S V にリンクされ得る。G e t V D L i s t ( i n f o ) とした獲得メッセージが、ゾーン・コントローラクラスにおける仮想デバイスを記載するために用いられ得る。この情報にて、このノードが構成可能なメンバーを有していないことが分かる。仮想デバイスは、選択を制御するメカニカルポットおよびボリューム ( コントロール面 ) を制御する 2 つの瞬間スイッチ以外のユーザ・インターフェースを有しない。ゾーン・コントローラ・オブジェクトにアドレス指定される D e s c r i b e V D とした記述メッセージがインスタンス属性、S V の実際の値、およびのゾーン・コントローラ仮想

40

50

デバイスのサブスクリプションを返し得る。

【 0 1 6 9 】

デバイスからクエリさせたデータを受け取ると、発見エンジン 6 2 0 は、プラグイン・インターフェース 5 2 2 を介して情報をプロダクト・プラグイン 5 2 6、またはジェネリック・プラグイン 5 2 4 に通信し得る。対応するデバイスに対して選択されるべきまたは作成されるべきコントロール・パネル・ビューおよび/または GUI のアスペクトを決定するために、情報がプロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 によって用いられ得る。

【 0 1 7 0 】

プロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 はまた、未知のコントロール・パネル・ビューおよび/または GUI のアスペクトの決定を補助するように、発見エンジン 6 2 0 を有効にする。未知のコントロール・パネル・ビューまたは GUI のアスペクトは、オーディオ関連デバイスのオーディオ・ノード構造のいくつか、またはすべてを表すプロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 のいずれかから利用可能でない 1 つである。例えば、プロダクト・プラグイン 5 2 6、またはジェネリック・プラグイン 5 2 4 が、オーディオ・ノード構造の 1 つ以上のコンポーネントに対して GUI のアスペクトを作成するためにオーディオ関連デバイスから取得されるオーディオ・ノード構造に含まれている十分な情報を有しないとき、発見エンジン 6 2 0 がさらなる詳細を取得するために有効にされ得る。

【 0 1 7 1 】

ノード発見マネージャ

GUI のコア 5 2 0 はまた、ノード発見マネージャ 6 2 8 を含み得る。ノード発見マネージャ 6 2 8 は、ノード属性要求エンジン 6 3 0 およびプラグイン・サーチ・エンジン 6 3 2 を含む。ノード属性要求エンジン 6 3 0 は、ノードクラス属性に関してデバイスにクエリし得る。クラス属性は、オーディオ関連デバイスの機能性を示し得る。加えて、クラス属性は、オーディオ関連デバイスに対応するプロダクト・プラグイン 5 2 6 に一致させるために情報を提供し得る。例えば、クラス属性は、増幅器といったオーディオ関連デバイスのタイプ、モデルといったオーディオ関連プロダクトファミリ、および/またはモデルナンバーまたはプロダクト名といったプロダクト識別子を記述し得る。

【 0 1 7 2 】

ノード発見マネージャ 6 2 8 は、プロダクト・プラグイン 5 2 6 のいずれかがデバイスを含むクラスを有するかどうかを決定するためにブロードキャストされたメッセージを有するプロダクト・プラグイン 5 2 6 にクエリするためにクラス属性を用い得る。言い換えれば、プロダクト・プラグイン 5 2 6 がデバイスに関連しているかどうか、従って、デバイスに対して所定のコントロール・パネル・ビューを有するかどうかを決定する。クエリでブロードキャストされるクラス属性は、プロダクト・プラグイン 5 2 6 に格納されているクラス属性と比較され得る。プロダクト・プラグイン 5 2 6 の 1 つがクラス属性認識し、オーディオ関連デバイスとの関連性を示すとき、ノード発見マネージャ 6 2 8 は、プロダクト・プラグイン 5 2 6 の起動およびプラグイン・インターフェース 5 2 2 を有する通信ノードの作成を可能にし得る。

【 0 1 7 3 】

プラグイン 5 2 3 が、オーディオ関連デバイスのオーディオ・ノード構造のコンポーネントを含まないとき、プラグイン・サーチ・エンジン 6 3 2 は有効にされ得る。プラグイン・サーチ・エンジン 6 3 2 は、オーディオ・ノード構造に含まれている未知のコンポーネントに対して他のプラグイン 5 2 3 にクエリするために有効にされ得る。加えて、プラグイン・サーチ・エンジン 6 3 2 は、GUI のアスペクトに対して他のプラグイン 5 2 3 にクエリするために有効にされ得る。プロダクト・プラグイン 5 2 6 が、オーディオ関連デバイスのオーディオ・ノード構造をサポートするために必要とされる GUI コンポーネントのすべてであって、所定のコントロール・パネル・ビューにおいてその GUI コンポーネントのすべてを含まないとき、GUI のアスペクトのクエリは有効にされ得る。他の

10

20

30

40

50

プラグイン 5 2 3 が、それぞれのクエリに含まれる未知のコンポーネントまたは G U I のアスペクトに関する情報を含まないとき、プラグイン・サーチ・エンジン 6 3 2 がまた発見エンジン 6 2 0 を有効にする。

#### 【 0 1 7 4 】

G U I のコア 5 2 0 はまた、プラグイン・ローダ 6 3 4 を含み得る。プラグイン・ローダ 6 3 4 は、G U I のコア 5 2 0 にとって現在利用可能なすべてのプロダクト・プラグイン 5 2 6 の動作中のリストを維持し得る。従って、G U I のコア 5 2 0 は、プラグイン・ローダ 6 3 4 によって維持されるリストに基づいて、プラグイン・インターフェース 5 2 2 との通信を介してプロダクト・プラグイン 5 2 6 にクエリおよび / または起動を開始し得る。

10

#### 【 0 1 7 5 】

##### プラグイン・インターフェース

図 7 は、このシステムにおいて用いられ得るプラグイン・インターフェース 5 2 2 のブロック図である。プラグイン・インターフェース 5 2 2 は、格納モジュール 7 0 0 およびネットワーク・モジュール 7 0 1 を含む。ネットワーク・モジュール 7 0 1 は、G U I のコア 5 2 0 とプロダクト・プラグイン 5 2 6 との、および / またはジェネリック・プラグイン 5 2 4 との間の通信インターフェースの役目を果たす通信コンジット 7 0 2 を提供し得る。格納モジュール 7 0 0 は、プロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 を起動するプラグイン・ランチャ 7 0 2 を含む。追加的に、ネットワーク・モジュール 7 0 1 は、G U I のコア 5 2 0 を用いて、プロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 と関連しているものとして識別されるオーディオ関連デバイス間の通信をセットアップするノード通信インスタンスエータ 7 0 4 を含み得る。オーディオ関連デバイスおよびプラグイン 5 2 3 とのそれぞれの通信は別々で、独立したものであり得るので、プラグイン 5 2 1 が、オーディオ関連デバイスへの通信パスの知識 ( k n o w l e d g e ) を含むことを必要とし得ない。このような知識は、オーディオ関連デバイス、オーディオ関連デバイスの I D 等とともに用いられ得る通信プロトコルを含み得る。このような態様で、プラグイン・インターフェース 5 2 2 は、( G U I のコア 5 2 0 を介して ) デバイスへの通信コンジットを開くことによってオーディオ・ネットワーク 1 2 0 からプロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 をシールドし得る。従って、プロダクト・プラグイン 5 2 6 またはジェネリック・プラグイン 5 2 4 は、オーディオ・ネットワーク 1 2 0 の知識なしにオーディオ関連デバイスとオーディオ・ネットワーク 1 2 0 上で通信し得る。

20

30

#### 【 0 1 7 6 】

##### プロダクト・プラグイン

図 8 は、このシステムにおいて用いられ得るプロダクト・プラグイン 5 2 6 のブロック図である。このシステムは、オーディオ・システムに含まれているオーディオ関連デバイスに関連付けられ得る複数のプロダクト・プラグインを含み得る。プロダクト・プラグイン 5 2 6 は、1 つ以上の関連したオーディオ関連デバイスをサポートし得る。例えば、同様なモデルナンバー、機能性および / またはメーカーを有するオーディオ関連デバイスは、同一のプロダクト・プラグイン 5 2 6 と関連付けられ得る。一般的に、プロダクト・プラグイン 5 2 6 は、関連したオーディオ関連デバイスの機能性および制御能力を表すために用いられ得る格納されている所定のオーディオ関連デバイス特定の情報を含み得る。このような情報は、格納されている所定の G U I のコントロール・パネル・ビュー、格納されている所定の G U I のアスペクト、関連したオーディオ関連デバイスのクラス識別情報、オーディオ関連デバイスの格納されている所定のノード構造、格納されている所定のノード構造の部分を含み得る。

40

#### 【 0 1 7 7 】

プロダクト・プラグイン 5 2 6 は、タイプ発見モジュール 8 0 0 およびコントロール・パネル・ビュー・ビルダー 8 0 1 を含み得る。コントロール・パネル・ビュー・ビルダー 8 0 1 は、ノード・インスタンス化エンジン 8 0 2、コントロール・パネル・ライブラリ

50

804、コントロール・パネル・データ・ストレージ806およびコントロール・パネル・ビュー・バッファ808を含み得る。タイプ発見モジュール800は、プラグイン・インターフェース522を介してGUIのコア520と通信するノードクラス分析810を含み得る。ノードクラス分析810は、プラグイン・インターフェース522を介してノード属性要求エンジン630からブロードキャストされたクラス情報を受け取る。クラス情報は、オーディオ関連デバイスに前もって割り与えられるオブジェクト・クラスIDおよびプロダクト名といったクラス属性を含む。クラス情報は、クラス属性クエリに回答してオーディオ関連デバイスによって提供され得る。クラス属性は、オーディオ関連デバイスの性質およびタイプを示す任意の記号、ワード、数字、またはそれらのいくつかの組み合わせであり得る。

10

**【0178】**

クラス属性は、ノード・クエリで、ノード発見マネージャ628によってプロダクト・プラグイン526にブロードキャストされ得る。あるいは、プロダクト・プラグイン526のそれぞれは、個々にノード・クエリに送達され得る。クラス属性を含むノード・クエリは、プロダクト・プラグイン526の1つがデバイスのクラス属性に対応するか否かを決定するために用いられ得る。ライブラリ・ストレージ・モジュール816に格納されているクラス属性がクエリに含まれているクラス属性と比較され得る。例えば、ライブラリ・ストレージ・モジュール816は、プロダクト動的リンク・ライブラリ(「DLL」)を含み得る。DLLは、プロダクト・プラグイン526に関連した各オーディオ関連デバイスのオブジェクト・クラスIDおよびプロダクト名を含む静的メンバーを有するパグリック・プロダクトDLL記述(「PDD」)クラスを含み得る。一度クラス属性情報が比較されると、ノードクラス分析810は、プロダクト・プラグイン526が、オーディオ関連デバイスの対応するクラス属性に関連しているか否かを示すメッセージをGUIのコア520に送達し得る。

20

**【0179】**

プロダクト・プラグイン526がデバイスのクラス属性に対応するとき、通信コンジットがプラグイン・インターフェース522を介してセットアップされ得る。プラグイン・インターフェース522を用いて、取り出し(retrieval)エンジン812は、オーディオ関連デバイスからオーディオ・ノード構造を取得するために用いられ得る。オーディオ・ノード構造は、オーディオ関連デバイスの機能性を表し得、SV、オブジェクトおよび/または仮想デバイスを含み得る。クラス分析814は、それから、ライブラリ・ストレージ816との比較を行うためにオーディオ・ノード構造をトラバースし得る。ライブラリ・ストレージ816は、プロダクト・プラグイン526に関連したオーディオ関連デバイスのそれぞれに対する1つ以上の所定のオーディオ・ノード構造および/またはオーディオ・ノード構造コンポーネントを含み得る。オーディオ・ノード構造、および/またはオーディオ・ノード構造コンポーネントは、プロダクト・プラグイン526によってサポートされるオーディオ関連デバイスのそれぞれの機能性を示す構成において仮想デバイス、オブジェクトおよび/またはSVを含み得る。

30

**【0180】**

1つの例において、プロダクト・プラグイン526は、オーディオ関連デバイスに含まれている仮想デバイスに対してオーディオ関連デバイスにクエリし得る。あるいは、ノード発見マネージャ628は、クラス属性の一部として仮想デバイスのリストを受け取り得る。別の例において、プロダクト・プラグイン526は、オーディオ関連デバイスに含まれている仮想デバイス、オブジェクトおよび/またはSVのそれぞれに対してクエリし得る。

40

**【0181】**

仮想デバイス、オブジェクトおよび/またはSVが既知であるとき、オーディオ・ノード構造のその部分の表現およびオーディオ・ノード構造のその部分が既知であるという表示が、ノード構造ストレージ818に格納され得る。例えば、表現は、既知である仮想デバイスのリストおよび既知である仮想デバイスの表示であり得る。別の例において、オー

50

ディオ・ノード構造の既知の部分およびオーディオ・ノード構造の既知の部分に含まれている属性が、ノード構造ストレージ 8 1 8 に格納され得る。オーディオ関連デバイスから受け取られたオーディオ・ノード構造のいずれかの部分が、ライブラリ・ストレージ 8 1 8 に含まれているオーディオ・ノード構造と、またはノード構造コンポーネントとに含まれている所定の仮想デバイス、オブジェクト、SVと対応していることが見出されないとき、プラグイン・サーチ・エンジン 6 3 2 は、取り出しエンジン 8 1 2 によって、オーディオ・ノード構造の未知のコンポーネントに対して他のプロダクト・プラグイン 5 2 6 およびジェネリック・プラグイン 5 2 4 にクエリするために有効にされ得る。あるいは、プロダクト・プラグイン 4 2 4 が、このようなクエリを行い得る。

**【 0 1 8 2 】**

未知のコンポーネントが別のプラグイン 5 2 3 に存在するとき、未知のコンポーネントが取り出され、ライブラリ・ストレージ 8 1 6 に格納され、かつ、オーディオ・ノード構造を十分に定義するために用いられ得る。オーディオ・ノード構造の対応するコンポーネントが、それから、ノード構造ストレージに格納され得る。未知のコンポーネントが、プロダクト・プラグイン 5 2 6 およびジェネリック・プラグイン 5 2 4 のいずれかから利用可能でないとき、発見エンジン 6 2 0 は、オーディオ関連デバイスのオーディオ・ノード構造全体、または先述されたようなオーディオ関連デバイスの未知のコンポーネントのいずれかを決定するために開始され得る。発見エンジン 6 2 0 によって決定されたオーディオ・ノード構造および/またはコンポーネントがオーディオ関連デバイスのオーディオ・ノード構造に関連付けられ、ライブラリ・ストレージ 8 1 6 に格納され得る。発見エンジン 6 2 0 によって取得された、オーディオ・ノード構造の未知の部分に対する詳細な属性がまた、ノード構造ストレージ 8 1 8 に格納され得る。

**【 0 1 8 3 】**

発見エンジン 6 2 0 は、複数のプロダクト・プラグイン 5 2 6 および/またはジェネリック・プラグイン 5 2 4 のいずれかの 1 つによって有効にされ得るので、これは好都合である。つまり、プロダクト・プラグイン 5 2 6 およびジェネリック・プラグイン 5 2 4 のそれぞれは、発見エンジン 6 2 0 を動作させる命令を格納および実行をする必要がない。加えて、発見エンジン 6 2 0 内の命令に対する変化およびアップデートが一度なされ得る。

**【 0 1 8 4 】**

一度オーディオ関連デバイスのオーディオ・ノード構造全体が決定され、表示がノード構造ストレージ 8 1 8 に格納されると、ノード・インスタンス化エンジン 8 0 2 は、1 つ以上のコントロール・パネル・ビューを識別するために、ノード構造ストレージ 8 1 8 に格納されているオーディオ・ノード構造を利用し得る。コントロール・パネル・ビューは、オーディオ・ノード構造のコンポーネントおよび/またはオーディオ・ノード構造のコンポーネントに含まれている属性に基づいて識別され得る。例えば、格納されている所定のコントロール・パネル・ビューが 1 つ以上の特別に識別された仮想デバイスに関連付けら得る。

**【 0 1 8 5 】**

ノード・インスタンス化エンジン 8 0 2 は、コントロール・パネル・ライブラリ 8 0 4 にアクセスし、オーディオ・ノード構造のコンポーネントに対応する 1 つ以上の GUI のコントロール・パネル・ビューを取得し得る。コントロール・パネル・ライブラリ 8 0 4 は、ディスプレイに表示され得る所定の GUI のコントロール・パネル・ビューおよび/または所定 GUI のアスペクトを格納し得る。対応する GUI のコントロール・パネル・ビューは、ノード・インスタンス化エンジン 8 0 2 によって構成され、インスタンス化のために格納され得る。ユーザによって、オーディオ関連デバイスの機能のモニタ、制御および動作を可能にすることを要求されるとき、GUI のコントロール・パネル・ビューは、ノード・インスタンス化エンジン 8 0 2 によってインスタンス化され、かつ、表示され得る。

**【 0 1 8 6 】**

ノード・インスタンス化エンジン 802 はまた、コントロール・パネル・ビューにおいて S V に対してサブスクライブ・メッセージングをセットアップし得る。加えて、ノード・インスタンス化エンジン 802 は、コントロール・パネル・ビューおよび関連した S V に関してリンク・マネージャ 614 と通信し得る。コントロール・パネル・データ・ストレージ 806 は、ノード・インスタンス化エンジン 802 によってインスタンス化される G U I のコントロール・パネル・ビューに含まれている S V の実際の値を格納するために用いられ得る。S V の実際の値は、取り出しエンジン 812 を用いてオーディオ関連デバイスから取り出され、コントロール・パネル・データ・ストレージ 806 で格納されるためにノード・インスタンス化エンジン 802 に提供され得る。コントロール・パネル・データ・ストレージ 806 はまた、先述されるように G U I マネージャ 606 によって指示 10  
されるとき、すべての設定およびコントロール・パネル・ビューを格納し得る。

**【 0 1 8 7 】**

コントロール・パネル・ビュー・バッファ 808 はまた、G U I のコントロール・パネル・ビューに関連したデータをバッファするために提供され得る。コントロール・パネル・ビュー・バッファ 808 は、コントロール・パネル・ビューおよびその関連データをバッファし得、一方、ノード・インスタンス化エンジン 802 は、インスタンス化および表示のために G U I のコントロール・パネル・ビューを用意する。加えて、複数の G U I のコントロール・パネル・ビューが用意され、コントロール・パネル・ビュー・バッファ 808 に格納され得る。従って、ユーザが、さまざまな G U I のコントロール・パネル・ビューを要求するとき、そのビューは、コントロール・パネル・ビュー・バッファ 808 においてバッファされているコントロール・パネル・ビューのセットから抽出され得る。抽出された G U I のコントロール・パネル・ビューは、ディスプレイ上に、コントロール・パネル・データ・ストレージ 806 に格納されている S V の実際の値を有してインスタンス化され得る。実際の値がコントロール・パネル・データ・ストレージ 808 に格納され、および/または G U I のコントロール・パネル・ビューにおいてインスタンス化される一方で、実際の値がアップデートされ得る。このことは、獲得コマンドを用いて、またはサブスクリプションによって起きる。 20

**【 0 1 8 8 】**

コントロール・パネル・ビュー・バッファ 808 において形成され、格納されるコントロール・パネル・ビューの数は、P E M の能力に基づき得る。P E M が「厚い」P E M であるときは、オーディオ・ノード構造全体に対する G U I のコントロール・パネル・ビューが構築され、格納され得る。あるいは、P E M が「薄い」P E M であるときは、わずかな、または 1 つだけのコントロール・パネル・ビューが、いつでも構築され、格納され得る。 30

**【 0 1 8 9 】**

ノード・インスタンス化エンジン 802 によって選択される 1 つ以上のコントロール・パネル・ビューが、オーディオ関連デバイスのオーディオ・ノード構造全体をサポートしていないとき、ノード・インスタンス化エンジン 802、または発見エンジンは、オーディオ・ノード構造のサポートされていないコンポーネントをサポートし得る 1 つ以上の格納されている所定の G U I のアスペクトに対してコントロール・パネル・ライブラリ 804 をチェックし得る。加えて、別のプロダクト・プラグイン 526 がオーディオ・ノード構造のサポートされていない部分を表し得る 1 つ以上の格納されている所定の G U I のアスペクトに対してチェックされ得る。例えば、オーディオ・ノード構造のサポートされていない部分は、それ以外の場合にはサポートされている仮想デバイス内に包含されるオブジェクトおよび/または S V のいくつかであり得る。 40

**【 0 1 9 0 】**

オーディオ・ノード構造を表し得る G U I のアスペクトを決定することは、オーディオ・ノード構造に含まれている属性に基づき得る。例えば、オーディオ・ノード構造のコンポーネントの属性がゼロから 100 % までの可変コントロールを必要とするコントロール G U I のアスペクトを記述し得るか、またはスライダがコントロール G U I のアスペクト 50

の機能性を表すために必要であることを特別に示し得る。別の例において、ノード構造コンポーネントの属性がゼロから100%までの可変インジケータを必要とするインジケータGUIの aspekto を記述し得るか、または数値ディスプレイがインジケータGUIの aspekto の機能性を表すために必要であることを特別に示し得る。

#### 【0191】

オーディオ・ノード構造の属性に記述される適正を満たすために欠けているGUIの aspekto があらかじめ決定され、かつ、コントロール・パネル・ライブラリ804、または別のプロダクト・プラグイン526に格納されているとき、格納されている所定のGUIの aspekto は、取り出され、所定のGUIのコントロール・パネル・ビューに統合され得る。このGUI aspekto の統合は、配置、サイズ調整、カラー調整、格納されている所定のGUIのコントロール・パネル・ビューにおける現存するGUIコンポーネントとのGUIの aspekto の相互関連、および/または対応するオーディオ・ノード・構造の属性によって識別される他の任意の動作および/または構成を含み得る。GUIの aspekto の取り込みに続いて、ノード・インスタンス化エンジン802は、オーディオ関連デバイスのオーディオ・ノード構造に関連して新しい所定のGUIのコントロール・パネル・ビューとして統合されたGUIの aspekto を含むGUIのコントロール・パネル・ビューを格納し得る。新しい所定のGUIのコントロール・パネル・ビューは、コントロール・パネル・ライブラリ804に格納され得る。統合されたGUIの aspekto を含むGUIのコントロール・パネル・ビューは、格納され、ともにディスプレイ上でインスタンス化され得る。

#### 【0192】

欠けているGUIの aspekto が、プロダクト・プラグイン526のいずれかから利用可能でないとき、ジェネリック・プラグイン524が有効にされ得る。ジェネリック・プラグイン524は、オーディオ・ノード構造をサポートしているデフォルト・パネル(エディタ)といった1つ以上のジェネリックGUIコントロール・パネルを構築するために有効にされ得る。ジェネリック・プラグイン524によって構築されたジェネリックGUIコントロール・パネルがノード・インスタンス化エンジン802に提供され得る。ノード・インスタンス化エンジン802は、オーディオ関連デバイスの機能性を表示するためにジェネリックGUIコントロール・パネルを用い得る。あるいは、格納されている所定のGUIの aspekto に関連していないオーディオ・ノード構造のコンポーネントの機能性を表すジェネリックGUIの aspekto を生成するために、ジェネリック・プラグイン524がプロダクト・プラグイン526によって有効にされ得る。発見エンジン620を用いて生成されたおよび/または識別された任意のGUIの aspekto はまた、所定のコントロール・パネル・ビューの欠けている部分を充填するためにノード・インスタンス化エンジン802によって用いられ得る。ノード・インスタンス化エンジン802はまた、オーディオ・ノード構造の属性に基づいて、生成されたGUIの aspekto を所定のコントロール・パネル・ビューの現存するGUIコンポーネントとを相互に関連付ける。

#### 【0193】

ノード・インスタンス化エンジン802はまた、インスタンス化されたGUIのコントロール・パネル・ビューを表示することを提供するデバイスの演算能力およびコントロール面の使用可能性を考慮し得る。デバイスの能力に基づいて、ノード・インスタンス化エンジン802は、格納されている所定のGUIのコントロール・パネル・ビュー、または格納されている所定のGUIの aspekto を簡素化、エンハンス、削除、または装飾(embellish)し得る。例えば、ディスプレイが小さく、表示のために低い解像度、または制限された「実際の面積(real estate)」といった制限されたディスプレイ能力しか有さないときには、格納されている所定のGUIのコントロール・パネル・ビューに含まれる選択されたGUIコンポーネントは、より簡素なGUIコンポーネントに変化され得るか、別のコンポーネントに組み合わされ得るか、または完全に削除され得る。別の例において、ディスプレイが比較的大きく、高い解像度のディスプレイ能力を含むとき、格納されている所定のGUIのコントロール・パネル・ビューは、グラフィカ

ル的により詳細な、または自動化等を有するといったより複雑なGUIコンポーネントを含むように変化され得る。GUIのコントロール・パネル・ビューを表示することを提供するデバイスの演算能力を考慮することはまた、格納されている所定のGUIのAspectおよび/またはGUIのAspectの統合に影響し得る。

#### 【0194】

コントロール面を有効にするように構成されている格納されている所定のコントロール・パネル・ビューに対する変化がまた、ハードウェアのコントロール面と一致するようになされ得る。例えば、オーディオ・ノード構造における属性によって記述されるようにコントロール・パネル・ビューのハードウェアが指示するとき、コントロール面におけるアナログ計測器を駆動する機能性を含む格納されている所定のコントロール・パネル・ビューが、デジタル数値ディスプレイを駆動する機能性に变化され得る。

10

#### 【0195】

##### ジェネリック・プラグイン

図9は、ジェネリック・プラグイン524のブロック図である。ジェネリック・プラグイン524は、デバイスのクラス属性が、プラグイン・ローダ634において現在リストアップされているプロダクト・プラグイン526のいずれかによって認識されていない際に有効にされ得る。有効にされるとき、ジェネリック・プラグイン524は、オーディオ関連デバイスのオーディオ・ノード構造を決定し得る。オーディオ・ノード構造は、先述されるように発見エンジン620によって決定され得る。加えて、ジェネリック・プラグイン524が、プロダクト・プラグイン526に格納されている、格納されている所定のGUIのコントロール・パネル・ビューによって完全にサポートされていないとき、ジェネリック・プラグイン524は1つ以上のジェネリックGUIのコントロール・パネル・ビューを作成するために有効にされ得る。

20

#### 【0196】

プロダクト・プラグイン526のいずれもオーディオ関連デバイスのクラス属性をサポートしないとき、ジェネリック・プラグイン524は、プラグイン・インターフェース522を介して、デバイスによって提供されるオーディオ・ノード構造に基づいて「ジェネリック」コントロール・パネル・ビューの1つ、またはそのセットを動的に作成するためにGUIのコア520によって有効にされ得る。あるいは、オーディオ関連デバイスのクラス属性をサポートしているプロダクト・プラグイン526の1つが、デバイスのオーディオ・ノード構造における1つ以上のコンポーネントに対応する所定のコントロール・パネルを有しないとき、ジェネリック・プラグイン524は、デバイスによって提供されるオーディオ・ノード構造に基づいて、1つ以上の「ジェネリック」コントロール・パネル・ビューを動的に作成するためにプロダクト・プラグイン526によって有効にされ得る。加えて、格納されている所定のGUIのAspectがオーディオ・ノード構造における1つ以上のコンポーネントに対して存在しないとき、ジェネリック・プラグイン524が同様に、ジェネリックGUIのAspectを生成するために有効にされ得る。ジェネリック・プラグイン524は、コントロール・パネル・ファクトリ900、属性分析902、ノード・インスタンス化エンジン904、配置エンジン906、取り出しエンジン908、コントロール・パネル・ビュー・バッファ910、ノード構造ストレージ912、およびコントロール・パネル・データ・ストレージ914を含み得る。

30

40

#### 【0197】

GUIのコア520(図5および図6)は、オーディオ関連デバイスのオーディオ・ノード構造を取得するためにオーディオ関連デバイスおよび/またはプロダクト・プラグイン526と通信するように有効にされ得る。あるいは、取り出しエンジン808は、オーディオ関連デバイスのオーディオ・ノード構造および属性を取得し得る。一度取得されると、オーディオ関連デバイスのオーディオ・ノード構造および/または属性は、先述されるように、ノード構造ストレージ912に格納され得る。加えて、取得されたSVの実際の値が先述されるように、コントロール・パネル・データ・ストレージ914に格納され得る。

50

## 【 0 1 9 8 】

属性分析 9 0 2 は、オーディオ関連デバイスの格納されているオーディオ・ノード構造に関連した属性をレビューし得る。オーディオ・ノード構造の属性の分析に基づいて、ノード・インスタンス化エンジン 9 0 4 は、ジェネリック G U I のコントロール・パネル・ビューおよび/またはジェネリック G U I のアスペクトの 1 つ、またはそのセットを生成するためにコントロール・パネル・ファクトリ 9 0 0 を開始し得る。ジェネリック G U I のコントロール・パネル・ビューまたはジェネリック G U I のアスペクトにおける G U I コンポーネントは、格納されているオーディオ・ノード構造に含まれている属性に基づいて決定され得る。

## 【 0 1 9 9 】

ノード・インスタンス化エンジン 9 0 4 は、ジェネリック G U I のコントロール・パネル・ビューに統合するために属性によって記述される格納されている所定の G U I のアスペクトに対してプロダクト・プラグイン 5 2 6 を検索するようにプラグイン・サーチ・エンジン 6 3 2 ( 図 6 ) を有効にし得る。あるいは、ノード・インスタンス化エンジン 9 0 4 は、プロダクト・プラグイン 5 2 6 から格納されている所定の G U I のアスペクトを検索し、取得し得る。このようにして、1 つ以上のジェネリック・コントロール・パネル・ビューは、所定の G U I のアスペクトおよび生成された「ジェネリック」G U I のアスペクトを含み得る。

## 【 0 2 0 0 】

ジェネリック G U I のコントロール・パネル・ビューのすべて、または一部を生成するためにオーディオ関連デバイスから受け取られるオーディオ・ノード構造において十分な属性が利用可能でないとき、発見エンジン 6 2 0 は、デバイスから追加的な情報を取得し、提供するために有効にされ得る。このような追加的な情報は、オーディオ・ノード構造に関して追加的な情報を含み得る。例えば、オブジェクトおよび/または S V の情報の取り出しは、オーディオ・ノード構造に含まれる仮想デバイスだけを以前供給していたオーディオ関連デバイスから取得され得る。加えて、取得される情報は、サイズ、カラー、形状、配線、オーディオ・ノード構造における別のコンポーネントとの相互関連、機能性の詳細、関連配置等といったオーディオ・ノード構造のコンポーネントの詳細な属性を含み得る。

## 【 0 2 0 1 】

ノード・インスタンス化・エンジン 9 0 4 は、表示のために完了したジェネリック G U I のコントロール・パネル・ビューまたはジェネリック G U I のアスペクトをインスタンス化し得る。加えて、S V の実際の値がコントロール・パネル・データ・ストレージ 9 1 4 から取り出され、ジェネリック G U I のコントロール・パネル・ビューまたはジェネリック G U I のアスペクトに含まれ得る。あるいは、ジェネリック G U I のアスペクトは、格納されている所定の G U I のコントロール・パネル・ビューを完了するようにプロダクト・プラグインに提供され得る。ジェネリック G U I のコントロール・パネル・ビューまたはジェネリック G U I のアスペクトは、後ほど取り出されるためにオーディオ関連デバイスに関連して格納され得る。ジェネリック・コントロール・パネル・ビューまたはジェネリック G U I のアスペクトの格納がコントロール・パネル・データ・ストレージ 9 1 4 にされ得る。

## 【 0 2 0 2 】

ノード・インスタンス化エンジン 9 0 4 はまた、ジェネリック G U I のコントロール・パネル・ビューにおける各 S V に対してサブスクライブ・メッセージングをセットアップし、ジェネリック・コントロール・パネル・ビューおよびそれに関連した S V に関してリンク・マネージャ 6 1 4 と通信し得る。ジェネリック・プラグイン 5 2 4 はまた、配置エンジン 9 0 6 をオプション的に含み得る。配置エンジン 9 0 6 は、オーディオ・ノード構造に含まれている属性に基づいて「厚い」P E M 内のジェネリック・コントロール・パネル・ビューにおける G U I のアスペクトの配置を管理し得る。加えて、「厚い」P E M 内で、コントロール・パネル・ビュー・バッファ 9 1 0 はまた、デバイスに対するジェネリ

10

20

30

40

50

ック・コントロール・パネル・ビューの追加的な層に関してバッファ・データにオプション的に提供され得る。ノード・インスタンス化エンジン 904 はまた、オーディオ・ノード構造を取得し、ジェネリック・コントロール・パネル・ビューおよび/またはジェネリック GUI のアスペクトを構築するとき、先述されるようなディスプレイを含むデバイスの能力を考慮し得る。

#### 【0203】

V . オーディオ関連デバイス ( 続き )

図 10 は、図 2 および図 3 に図解されるオーディオ関連デバイス 200 の一部のブロック図である。オーディオ関連デバイス 200 の例の一部は、ネットワーク・インターフェース 302、オーディオ・ストリーム 322、仮想デバイス・エンジン 1002、オブジェクト・エンジン 1004、SV エンジン 1006、ノード識別モジュール 1008、サブスクリプション・マネージャ・モジュール 1010、デバイス・ツー・デバイス通信マネージャ・モジュール 1012、およびリンク・マネージャ・モジュール 1014 を含む。図 10 に示されるさまざまなモジュールは、メモリ 206 に格納され得、プロセッサ 204 ( 図 2 ) によって選択的に実行可能である。他の例において、追加的なモジュールが、オーディオ関連デバイス 200 の記述された機能性を表すために含まれ得る。

オーディオ関連デバイス 200 のそれぞれは、オーディオ・ネットワーク 120 ( 図 1 ) 上のノードとして表され得る。各オーディオ・ノード構造は、仮想デバイス、オブジェクト、およびノードを含む 3 レベル階層を含み得る。各仮想デバイスは、1 つ以上のオブジェクトおよび 1 つ以上の SV をグループ化し得る。仮想デバイス、オブジェクト、および SV は、対応するオーディオ関連デバイス 200 の機能性、設定および動作的なアスペクトを表す。SV は、通常、可変データの形式である動作パラメータである。

#### 【0204】

仮想デバイス・エンジン 1002 は、ノードに含まれている 1 つ以上の仮想デバイスの複数の属性を追跡し、維持するように構成される。属性は、静的、または動的であり得、対応する仮想デバイスの能力、機能性および動作性の記述を提供し得る。加えて、ネスト化されたノードの構成のために、仮想デバイス・エンジン 1002 はまた、仮想デバイスに含まれるオブジェクトおよび SV 追跡し、維持し得る。オブジェクト・エンジン 1004 は、ノードにおけるオブジェクトに関連した属性を追跡し、維持するように構成されている。加えて、オブジェクト・エンジン 1004 は、オブジェクトにネスト化された SV を追跡し、維持するように構成されている。SV エンジン 1006 は、オーディオ関連デバイス ( ノード ) に存在する実際のデータの値を追跡、維持するように構成されている。

#### 【0205】

オーディオ関連デバイス 200 が当初ネットワークに結合されているとき、ノード識別モジュール 1008 はノード ID 生成するように構成されている。ノード ID は、ネットワーク 120 上のオーディオ関連デバイス 200 を一意に識別し得る。ノード ID は、手動による割り当てによって、またはアドレス・ネゴシエーションを介してといった多くの異なる態様で割り与えられ得る。先述されるように、ノード ID はまた、PEM によって割り与えられ得る。ノード ID が手動で割り与えられ得る例において、ノード ID は、オーディオ関連デバイス 200 に関連した物理的な ( 有形の ) スイッチを用いてセットされ得る。例えば、ノード ID は、オーディオ関連デバイス 200 上の DIP スイッチ、またはパネル設定を介して割り与えられ得る。

#### 【0206】

ノード ID は、オーディオ・ネットワーク上のオーディオ関連デバイス間のアドレス・ネゴシエーションを介して割り与えられ得る。いくつかの例において、オーディオ関連デバイスは、オーディオ・ネットワークからのメッセージで受け取られるノード ID に基づいて、ノード ID を割り与えられ得る。別の例において、オーディオ関連デバイス 200 は、例えば、ランダムに、またはシーケンシャルにノード ID を選択することによって ID ノード ID を生成し得る。ランダム・アドレス・ネゴシエーションは、アドレス衝突の可能性を最小にし得る。シーケンシャル・アドレス・ネゴシエーションは、オーディオ関連

10

20

30

40

50

デバイスが、名前より、むしろ数値アドレスによって識別され得る例において特に有用であり得る。シーケンシャル・アドレス・ネゴシエーションが用いられる例において、第1のオーディオ関連デバイスがアドレス1、アドレス2等を取るごとにオーディオ関連デバイスがパワー・アップし得る。いくつかの例において、パワー・シーケンサは、オーディオ関連デバイスをシーケンシャルにパワー・アップするために用いられ得る。

**【0207】**

ランダム、またはシーケンシャルのいずれかの態様でノードIDを選択する方法の例が図11に示されている。パワー・アップすると、オーディオ関連デバイス200は、ステップ1101に示されるように有効なノードIDが割り与えられたかどうかを決定し得る。オーディオ関連デバイス200が有効なノードIDを有するとき、そのノードIDが用いられ得る。パワー・アップした際に、オーディオ関連デバイス200が有効なノードIDを有しないとき、オーディオ関連デバイス200は、アドレス・ネゴシエーション・プロセスをスタートし得る。オーディオ関連デバイス200は、パワー・アップした際に、「0」といった所定の無効なノードIDを用いて、またはメモリ206に格納されているノードIDが無効であるとカテゴリ化する属性によってノードIDが無効であることを決定し得る。いくつかの例において、ノードIDは、オーディオ関連デバイス200をパワー・アップしたとき、その同一のノードは、パワー・アップされた前回の時から再格納されるように、同一のノードIDが不揮発性ストレージに格納され得る。他の例において、ノードIDは、パワー・ダウンした際に無効にされ得る。

**【0208】**

ステップ1102では、オーディオ関連デバイス200が、シーケンシャル・アドレス・ネゴシエーションに対して構成されているか否かの決定がなされる。オーディオ関連デバイス200が、シーケンシャル・アドレス・ネゴシエーションに対して構成されているとき、オーディオ関連デバイス200は、ステップ1004に示されるようにテスト・ノードIDを所定のアドレスにセットし得る。例えば、所定のアドレスは、「1」であり得る。オーディオ関連デバイス200は、ステップ1006に示されるようにネットワークの混雑を最小にするためにある時間（例えば、ランダムな時間）を待機し得る。テスト・ノードIDは、ステップ1008に示されるようにネットワーク上の他のオーディオ関連デバイスに、テスト・ノードIDが既に用いられているか否を決定する要求アドレス・メッセージとしてブロードキャストされ得る。要求アドレス・メッセージを受け取ると、ネットワーク上の他のオーディオ関連デバイスは、ステップ1010に示されるように、テスト・ノードIDが既に用いられているか否を決定し得る。オーディオ関連デバイス200が、要求アドレス・メッセージに対する応答を受け取らないとき、ノードIDは、ステップ1002に示されるようにテスト・ノードIDにセットされ得る。テスト・ノードIDが用いられているとき、一意のノードIDが選択されるまで、ステップ1014に示されるように、オーディオ関連デバイス200はテスト・ノードIDを引き続きしてインクリメントし得る。

**【0209】**

オーディオ関連デバイス200がシーケンシャル・アドレス・ネゴシエーションに対して構成されていないとき、ステップ1016に示されるように、オーディオ関連デバイス200がランダム・アドレス・ネゴシエーションに対して構成されているか否の決定がされ得る。ランダム・アドレス・ネゴシエーション構成でないときには、オーディオ関連デバイスはノードIDを生成するために別の方法を用い得る。しかしながら、オーディオ関連デバイス200がランダム・アドレス・ネゴシエーションに対して構成されているとき、オーディオ関連デバイス200はステップ1018に示されるように、ランダム・ノードIDを生成し得る。いくつかの例において、オーディオ関連デバイスのシリアル・ナンバーは、ノードIDをランダムに生成するためのシード（seed）であり得る。ネットワーク上の他のオーディオ関連デバイスのノードIDとの衝突を避けるために、オーディオ関連デバイス200は、ステップ1020に示されるように、生成されたノードIDを有する要求アドレス・メッセージをブロードキャストし得る。ステップ1022に示される

ように、ネットワークの混雑を最小にするために、ブロードキャストはある時間（例えば、ランダム時間）待機後に起こり得る。要求アドレス・メッセージを受け取ると、ネットワーク上の他のオーディオ関連デバイスは、ステップ1024に示されるように、生成されたノードIDが既に用いられているか否かを決定し得る。生成されたノードIDが用いられているとき、オーディオ関連デバイス200は通知され得る。オーディオ関連デバイス200は、一意のノードIDが選択されるまでランダム・ノードIDを引き続き生成し得る。オーディオ関連デバイス200が要求アドレス・メッセージに対する応答を受け取らないとき、ノードIDは、ステップ1026に示されるように生成されたノードIDにセットされ得る。

#### 【0210】

あるいは、オーディオ関連デバイスは、メモリ206に格納されているノードIDを用い得る。例えば、ノードIDを割り与えられる前に、オーディオ関連デバイス200は当初は「0」といった所定のノードIDを有し得る。いくつかの例において、ノードIDは、オーディオ関連デバイス200がパワー・アップしたとき、同一のノードIDは、パワー・アップされた前回の時から再格納されるように、不揮発性ストレージに格納され得る。例えば、メモリに格納されているノードIDが、「0」といった所定のノードIDでないために、オーディオ関連デバイス200はノードIDが以前割り当てられたと決定し得る。他の例において、ノードIDはパワー・ダウンした際に無効にされ得る。

サブスクリプション・マネージャ・モジュール1010は、オーディオ・ネットワークにおける他のオーディオ関連デバイスによってオーディオ関連デバイスのサブスクリプションを管理するように構成されている。加えて、サブスクリプション・マネージャ・モジュール1010は、オーディオ関連デバイスによって、オーディオ・ネットワーク120上の他のオーディオ関連デバイスへのサブスクリプションを管理するように構成されている。データ変数が変化するとき、サブスクリプションは別のオーディオ関連デバイスからの1つ以上のデータ変数の受け取りを要求するように用いられる。例えば、第1のオーディオ関連デバイスが、第2のオーディオ関連デバイスのオーディオ信号レベルにサブスクライブし得る。この例において、オーディオ信号レベルが変化するとき、第2のオーディオ関連デバイスは、オーディオ信号レベルを表すアップデートされたデータ変数を、オーディオ・ネットワーク120上の第1のオーディオ関連デバイスに提供し得る。同様に、第2のオーディオ関連デバイスは、第1のオーディオ関連デバイスの1つ以上のデータ変数にサブスクライブし得る。このようなサブスクリプションは、サブスクリプション・マネージャ・モジュール1010によって管理され、かつ、統治される。

#### 【0211】

デバイス・ツー・デバイス通信モジュール1012は、オーディオ・システムにおける他のオーディオ関連デバイスとの非ネットワーク通信を確立し、維持するように構成されている。先述されたように、オーディオ関連デバイスは、ネットワーク通信の外側のオーディオ関連デバイスと直接的に通信し得る。従って、デバイス・ツー・デバイス通信モジュール1012は、ノードIDおよび別のオーディオ関連デバイスと確立された通信に関連した他の関連する情報ならびに設定を格納するように構成されている。加えて、オーディオ関連デバイス間でパスされたメッセージは、デバイス・ツー・デバイス通信モジュール1012によって送達され、かつ、受け取られ得る。さらに、デバイス・ツー・デバイス通信モジュール1012は、送達されるメッセージを生成し、受け取られたメッセージを処理し得る。

#### 【0212】

リンク・マネージャ・モジュール1014は、他の複数のオーディオ関連デバイスへのサブスクリプションを管理するように構成されている。リンク・マネージャ1014は、オーディオ関連デバイス200内のデータ変数を管理するように構成され得る。管理されているデータ変数が変化するとき、リンク・マネージャ・モジュール1014は、その変化を他のオーディオ関連デバイスに通知し得る。その通知に回答して、他のオーディオ関連デバイスは、変化されたデータ変数を有するオーディオ関連デバイスにサブスクライブ

10

20

30

40

50

し、アップデートされたデータの変数の受け取りを要求し得る。リンク・マネージャの動作はまた、図5のPEMに関して記載される。

【0213】

リンク・マネージャ・モジュール1014はまた、特定のオーディオ関連デバイスに対してサブスクリプションの負荷を減らすように構成されている。例えば、第1のオーディオ関連デバイスがあるデータ変数に対する多くのサブスクリプションの負荷を負うとき、より低いサブスクリプション要求を有する第2のオーディオ関連デバイスのリンク・マネージャ・モジュールが、その負荷を軽減するために用いられ得る。このシナリオでは、第2のオーディオ関連デバイスは、特定のデータ変数を受け取るために、第1のオーディオ関連デバイスにサブスクライブし得る。加えて、第2のオーディオ関連デバイスのリンク・マネージャ・モジュールは、特定のデータ変数が受け取られたとき、他のオーディオ関連デバイスに注意を促すように構成され得る。従って、他のオーディオ関連デバイスは、アップデートされた特定のデータ変数を取得するために、第1のオーディオ関連デバイスではなく、第2のオーディオ関連デバイスにサブスクライブし得、それにより、第1のオーディオ関連デバイスの負荷が減少される。

10

【0214】

VI. ネットワークノード発見

図12は、オーディオ・ネットワーク120および複数「N」のオーディオ関連デバイス1204に結合されているスイッチ1202の構成の例である。この構成の例において、スイッチ1202は、ネットワーク接続1206および複数「N」のポート1208を有し得る。ネットワーク接続1206は、ネットワーク・アドレスを有し得る。ポート1208のそれぞれは、所定のノードIDを有し得る。

20

【0215】

複数のオーディオ関連デバイス1204は、関連したスピーカを有する増幅器のアレイであり得る。アレイにおけるスピーカのそれぞれの位置は、アレイから所望の音響効果を生成するために関連した増幅器の設定を指示し得る。

【0216】

(図示されていないが) PEMは、アレイにおける増幅器のそれぞれに対する格納されている所定の設定を含み得る。PEMはまた、スイッチ1202の格納されている所定の設定であって、ポート1208とノードIDとの間の対応を示す設定を含み得る。オーディオ関連デバイス1204およびスイッチ1202が起動されているとき、PEMは、スイッチ1202のために、オーディオ・ネットワーク120上のオーディオ関連デバイス1204を見ることができない。しかしながら、ポート1208への所定のノードIDの割り当てのために、PEMは、ノードIDに対応するポートの数1208に基づいて設定をオーディオ関連デバイス1204にダウンロードし得る。従って、デバイス1204のそれぞれが、アレイにおけるスピーカの位置に対応するスイッチ1202上のポート1208に結合されている。

30

【0217】

一度デバイスがノード・アドレスを割り当てられると、PEMおよび他のデバイスは、そのデバイスを発見し得る。以下の例は、複数のオーディオ関連デバイス間の発見に関連する。しかしながら、PEM、または他のデバイスは、オーディオ・ネットワーク上のオーディオ関連デバイスを発見し得る。例えば、PEMのノード分解能モジュール600(図6)は、後述されるように用いられ得る。このことをする際に、ルーティング層308(図3)は、所定のネットワーク・インターフェース上に常駐するオーディオ関連デバイスの存在を決定するために、またオーディオ関連デバイスをアドレス指定するために必要であるネットワーク情報を発見するために、Ethernet(登録商標)ネットワーク上のTCP/IP、485パスの上のトークン・ネットワーク・サービス、またはUSB上のUSBサービスといったアンダーライする層によって供給されるサービスを用い得る。いくつかの例において、オーディオ関連デバイス200(図2)は、ノードIDおよびネットワーク・アドレスを特定し得る。

40

50

## 【0218】

図13は、オーディオ・ネットワーク上のノードであって、システムとともに用いられ得るノードを発見する方法を示すフローチャートである。ルーティング・テーブルが、ブロック1302に示されるように、要求されたノードに対して検索される。ノードIDがルーティング・テーブルにおいて見つけれるとき、そのルートが用いられ得る(ブロック1303)。ノードIDに対するノードがルーティング・テーブルに見つけれないとき(1304)、デバイスと通信するルートが決定され得る。ルートを決定するために、オーディオ関連デバイスは、ステップ1306に示されるように、要求されたルートを特定するメッセージをブロードキャストし得る。オーディオ関連デバイスは、それから、ブロック1308に示されるように、ブロードキャストに対する応答に対して待機し得る。オーディオ・ネットワーク上の、ブロードキャストを受け取るデバイスは、要求されたノードへのルートを検索し得る。例えば、その要求を受け取るデバイスが、デバイスのルーティング・テーブルを検索し得る。別の例として、その要求を受け取るデバイスは、要求されたノードがデバイスのノードIDと一致するか否かを決定し得る。デバイスがブロードキャストに対する応答を受け取るとき(ステップ1310)、応答に基づいて、ルーティング・テーブルに新しいエントリが作成される(ステップ1312)。ブロードキャストから応答が受け取られないとき、ステップ1314に示されるように、ルートを要求するさらなるブロードキャストがされ得る。ブロードキャストに対して応答が受け取られないとき、ステップ1316に示されるように、「not found」エラーが返送され得る。いくつかの例において、デバイスは、ノードとの通信の必要性が生じるときだけ、オーディオ・ネットワーク上のこのような他のノードを見つけ得る。要求ドリブン・アプローチは、接続される必要があるノードに対するルートを要求するだけで、ルーティング・テーブルのサイズを最小する。しかしながら、他の例において、デバイスは、あるアプリケーション層レベルですべてのノードと通信し得ないが、デバイスはオーディオ・ネットワーク全体のモニタを所望し得る。

10

20

## 【0219】

デバイスは、ノード発見に対してDiscoinfoメッセージといった発見メッセージを用い得る。いくつかの例において、2つの異なる形式のDiscoinfoメッセージであり得る。クエリおよび応答である。Discoinfoメッセージの形式は、メッセージにおけるフラグフィールドの状態によって区別され得る。クエリ形式のDiscoinfoメッセージは、オーディオ・ネットワーク上のノードを見つけ出し、メッセージを送達するものに関する情報を、受け取るオーディオ関連デバイスにパスするために用いられ得る。クエリ形式のDiscoinfoメッセージを用いるために、送達するものは、送達するもののネットワークの詳細とともに位置付けられようとするオーディオ関連デバイスのノード・アドレスを特定し得る。送達するものは、それから、メッセージをオーディオ・ネットワーク上にブロードキャストし得る。以下の表は、Discoinfoメッセージのフォーマットおよび内容の例である。この表では、符号なしのロング・コンピュータ・データタイプが「ULONG」と表されている。

30

## 【0220】

【表 5】

メッセージ領域	データ・タイプ	データ例	
バージョン	UBYTE	2	
ヘッダーの長さ	UBYTE	25	
メッセージの長さ	ULONG	0xNNNNNNNN	
ソース・アドレス	NODE ID	0xNODE00000000	
目的アドレス	NODE ID	0xNODE00000000	10
メッセージID	UWORD		
フラグ	UWORD		
ホップ・カウント	UBYTE	0xNN	
シーケンス・ナンバー	UWORD	0xNNNN	
ペイロード...			
ノード	UWORD	送達するもののノード・アドレス	
コスト	UBYTE	srcへ戻る総コスト	
シリアル・ナンバー	BLOCK	送達するもののシリアル・ナンバー	
最大メッセージ・サイズ	ULONG	送達するものが処理し得る最大メッセージ・サイズ	20
キープ・アライブ期間	UWORD	キープ・アライブ率/毎秒	
ネットワークID	UBYTE	0xNN	
ネットワーク情報	ネットワーク特定	送達するもののネットワーク特定の情報	

表 5

あるいは、ノードの1つがプロキシの役割を果たし得る。この例において、オーディオ関連デバイス1004は、スイッチ1002に結合されている他のオーディオ関連デバイス1004にクエリし得る(図10)。プロキシ・オーディオ関連デバイス1004は、それから、設定およびノードIDといったオーディオ関連デバイス情報をPEM、または別のオーディオ関連デバイスに送達し得る。PEM、または別のオーディオ関連デバイスは、それから、適宜に、スイッチ1002に結合されているオーディオ関連デバイス1004の存在、およびスイッチ1002を介するオーディオ関連デバイス1004との通信を認識させられ得る。

【0221】

例えば、第1のノードおよび第2のノードは、ルータによって第3のノードに接続されていると仮定する。第3のノードは、第1のノードに接続し得る。第3のノードは、SetProxyState(On)メッセージといったプロキシ・イネイブル・メッセージを第1のノードに送達することによって、第1のノードをプロキシとしてセットし得る。第3のノードは、それから、(クエリ形式の)DiscInfoメッセージを第1のノードに送達し得る。これに回答して、第1のノードは、その第1のノードのルーティング・テーブルにおいて第3のノードのノードIDおよびネットワーク・アドレスを保存し得る。第1のノードは、それから、オーディオ・ネットワーク上でクエリ形式のDiscInfoメッセージをブロードキャストし得る。第1のノードによるこのブロードキャストは、第2のノードによって受け取られ得る。これに回答して、第2のノードは、その第2のノードのルーティング・テーブルにおいて第3のノードのノードIDおよびネットワーク・アドレスを保存し得る。

【0222】

第2のノードは、ここで、第1ノードを介して第3のノードと通信し得る。第2のノードは、第1のノードのノードIDをネットワーク・アドレスとして用いて、応答形式のDiscInfoメッセージを第3のノードに送達し得る。このメッセージは、第1のノードによって受け取られ得、第1のノードは、そのノードIDが第3のノードのノードIDである認識し得、それから、そのメッセージを第3のノードに転送する。第1のノードからメッセージを受け取ると、第3のノードは、そのメッセージが、プロキシを介してからルーティングされたことを認識し得る。例えば、第3のノードは、一致があるか否かを決定するためにメッセージを送達したノードのネットワーク・アドレスを、プロキシ・アドレスとアドレスと比較し得る。第3のノードは、それから、第1のノードのネットワーク・アドレスを有する第2のノードのノードIDを有してルーティング・テーブルにエン

10

#### 【0223】

いくつかの例において、このシステムは、ファイアウォールの後ろにあるプライベート・オーディオ・ネットワークとして構成され得、または、システムは、Internetといった共有の公共のインフラを介してアクセス可能である。このような例において、仮想プライベート・ネットワーク(「VPN」)が用いられ得る。VPNは、2つのネットワークの間、またはホストとネットワークの間にトンネルを作成し得る。VPNはまた、認証および/または暗号化サービスを提供し得る。例えば、ノードは、VPN用いて、Internet上のどこからでもアクセスされ得る。

20

#### 【0224】

いくつかの例は、オーディオ・ネットワークに結合しているノードが当該の他のデバイスによって見逃されている可能性を減らすためにフェールセーフ・ノード発見を含み得る。デバイスは、DiscFailSafeメッセージといった発見メッセージを、ある間隔で(例えば、1分毎に)オーディオ・ネットワークにブロードキャストし得る。ルーティング・テーブルに関して後述されるように、シリアル・ナンバーは、ノード・アドレスによって示されるノードを一意に識別し得る。受け取られる各DiscInfo応答メッセージに対して、デバイスは、DiscFailSafeメッセージのソースによって示されるノードIDへのルートに対してルーティング・テーブルを検索し得る。各識別

30

#### 【0225】

オーディオ関連デバイス200は、オーディオ・ネットワーク上で引き続くデバイスの参加を追跡し得る。ルートが「タイム・アウト」し、無効と判断される前に、ルート上で許される最大の無活動期間を決定するためにキープ・アライブ期間(「KAP」)が提供

40

#### 【0226】

オーディオ・ネットワーク上の各デバイスは、ルーティング・テーブルをインプリメントし、維持し得る。ルーティング・テーブルは、ノード間のデータ・パスを選択するため

50

に用いられ得る。ルーティング・テーブルは、オーディオ・ネットワーク上の他のデバイスへのルートの一覧、および他のデバイスへメッセージを送達するために必要とされるネットワーク情報を格納し得る。いくつかの例において、そのテーブルは、少なくとも20のルートおよび最低10の他のデバイスを収容し得る。

#### 【0227】

いくつかの例において、ルーティング・テーブルは、以下のコンポーネントを包含し得る。以下のコンポーネントはノード・アドレス、シリアル・ナンバー、最大メッセージ・サイズ、ネットワーク・インターフェース、ネットワーク・アドレス、およびコストである。ノード・アドレス・コンポーネントは、ルーティング・テーブルにおいてエントリがされるリモート・デバイスを識別し得る。リモート・ノードが、(マルチ・ホームの)1つより多いネットワーク・インターフェースを有するとき、デバイスに対するそのテーブルにおいていくつかのエントリがあり得、1つのエントリは各ネットワーク・インターフェースに対してである。このような場合には、各ルートは、同一のノード・アドレスを有し得る。ルーティング・テーブルのシリアル・ナンバー・コンポーネントは、ノード・アドレスによって示されるデバイスを一意に識別し得る。最大メッセージ・サイズ・コンポーネントは、ルーティング・テーブルにおけるエントリに関連したデバイスを通し得るメッセージの最大サイズを記述し得る。ネットワーク・インターフェース・コンポーネントは、デバイスによって用いられるネットワーク・インターフェースのタイプを記述し得る。先述されるように、デバイスが、複数のネットワーク・インターフェースを有するとき、そのデバイスは、ルーティング・テーブルにおいて1つより多いエントリを有し得る。ネットワーク・アドレス・コンポーネントがデバイスに関連したネットワーク・アドレスを記述するために用いられ得る。コスト・コンポーネントは、ルートに対するトランジット時間を表し得る。ノードが一般的に、最小費用のデータ・パスを介してデータをルーティングする一方で、他の要因が、選択されたルートに影響を及ぼし得る。例えば、ノード間の通信帯域幅は、ルートを決定する際の要因であり得る。例えば、Ethernet(登録商標)に関連したコストは、「1」であり得、一方、USBは、「10」コストを有し得、RS232は、「100」コストを有し得る。

#### 【0228】

##### VII. プロトコル

図14は、このシステムのネットワーク・プロトコルにおいて用いられ得る構造を示すブロック図である。ヘッダー情報は、バージョン、長さ、ソース・アドレス、目的アドレス、メッセージID、フラグ、カウント、および、シーケンス・ナンバーを含み得る。バージョンは、プロトコルのバージョンを示し得る。長さは、ヘッダーおよびメッセージの長さを示すために用いられ得る。ソース・アドレスおよび目的アドレスは、メッセージの送達元および目的地を示すために用いられ得る。

#### 【0229】

これらのアドレスは、送達元および目的デバイスのノード、仮想デバイスおよびオブジェクトに関する情報を包含し得る。例えば、ソースおよび目的アドレスが48ビットであるとき、0~23番目のビットは、オブジェクトのタイプを示すために用いられ得、23~31番目のビットは、仮想デバイスのタイプを示すために用いられ得、31~47番目のビットは、ノード・アドレスを示し得る。メッセージIDは、存在するメッセージのタイプを示すために用いられ得る。

#### 【0230】

フラグは、メッセージが要求、返答、迷惑情報、または別の動作か否かを示すために用いられ得る。例えば、フラグは、要求認識、認識、エラー、イベント、保障された、またはマルチ・パート・メッセージを示し得る。例えば、フラグが要求認識であるとき、動作が完了されるまでは、返答は送達されない。例えば、要求認識フラグが、SVの値を変化させる動作とともに用いられる。SV値が変化されると、認識フラグを有するメッセージが、それに応答して用いられ得る。連続する複数の動作が行われるべきいくつかの例において、認識フラグが受け取られるまで、各動作メッセージは送達されるために待機し得る。

要求認識および認識フラグのこのような使用は、ネットワーク・トラフィックを減らすために用いられ得る。

【0231】

ホップ・カウントは、送達メッセージに対するホップの数を示すために用いられ得る。シーケンス・ナンバーは、ネットワーク問題を解決するための診断補助として用いられ得る。例えば、シーケンス・ナンバーは、メッセージが送達される毎にインクリメントされ得る。

【0232】

図14は、カウント、SV\_ID、データ・タイプおよび値を含み得るSVメッセージの例を示す。SV\_IDは、目的SVを示すために用いられ得る。データ・タイプは、SVデータのサイズおよびフォーマットを示し得る。例えば、データ・タイプは、値がバイト、ワード、フロート、ストリング、またはブロックか否かを示し得る。その値は、実際に転送されているSVデータを示すために用いられ得る。図15は、このシステムに用いられ得るアプリケーション層のメッセージの例を包含するテーブルである。図16は、このシステムに用いられ得るルーティング層のメッセージの例を包含するテーブルである。

【0233】

VIII. オーディオ関連デバイスの発見

図17および図18は、オーディオ・ノード構造、従って、このシステムに用いられ得るオーディオ関連デバイスの機能性を発見するための方法の例を示すフローチャートである。ブロック1702では、GUIのコアは、オーディオ関連デバイスに送達された属性クエリに回答して、オーディオ関連デバイスであるノードからクラス属性を受け取り得る。GUIのコアは、ブロック1704で、クラス属性を含むオーディオ・ノード・クエリを用いてプロダクト・プラグインにクエリし得る。ブロック1706では、プロダクト・プラグインが、ノードのクラス属性を分析し、それを認識しないとき、ブロック1708で、システムにおいてさらなるプロダクト・プラグインがあるかどうか決定される。さらなるプロダクト・プラグインがあるとき、動作は、ブロック1704に戻り、GUIのコアは、クラス属性を分析するために別のプロダクト・プラグインにクエリする。オーディオ・システムにおけるプロダクト・プラグインのいずれもが、ノードを認識しないとき、ノードは、ブロック1710で、ジェネリック・プラグインにパスされる(図19を参照して後述される)。

【0234】

ブロック1706で、クエリされているプロダクト・プラグインが、ノードの属性を認識するとき、ブロック1716において、仮想デバイスのリストがオーディオ関連デバイスにおけるノード・マネージャ仮想デバイスから要求される。ブロック1718において、仮想デバイスが仮想デバイスのリストから選択される。ブロック1720において、仮想デバイスが既知であるか否かを決定するために、仮想デバイスが、プロダクト・プラグインによって、プロダクト・プラグインのライブラリ・ストレージに含まれている所定の仮想デバイスと比較される。仮想デバイスが、ライブラリ・ストレージに存在するとき、ブロック1722で、仮想デバイスが構成可能であるかどうか決定される。仮想デバイスに含まれているオブジェクトおよびSVがすべて既知であるので、仮想デバイスが構成可能でないとき、ブロック1724で、プロダクト・プラグインは、オーディオ関連デバイスからSVの実際の値を取得し得る。ブロック1726において、仮想デバイスが既知である表示およびオーディオ関連デバイスのオーディオ・ノード構造のその部分がノード構造ストレージに格納され得る。対応するSVの実際の値は、ブロック1728で、コントロール・パネル・データ・ストレージに格納され得る。ブロック1730で、すべての仮想デバイスが既知であるかどうか確かめるために、仮想デバイスのリストに含まれているすべての仮想デバイスがチェックされたかどうか決定される。チェックされていないとき、動作はブロック1718に戻り、別の仮想デバイスを分析し、決定するためにその別の仮想デバイスを選択する。仮想デバイスのリストにおけるすべての仮想デバイスがチェックされているとき、動作はブロック1732で終了する。

## 【 0 2 3 5 】

再びブロック 1720 を参照して、プロダクト・プラグインが、仮想デバイスを認識しないとき、他のプロダクト・プラグインは、ブロック 1736 で、仮想デバイスの知識に対してクエリされ得る。他のプロダクト・プラグインのクエリが、プラグイン・サーチ・エンジン、またはそのプロダクト・プラグインを用いて行われ得る。仮想デバイスが、別のプロダクト・プラグインによって既知であるとき、ブロック 1738 で、仮想デバイスが取り出され、ライブラリ・ストレージに格納され得、動作はその仮想デバイスが構成可能であるかどうかを決定するために、ブロック 1722 に続く。仮想デバイスが、別のプロダクト・プラグインによって既知でないとき、仮想デバイスを記述する詳細なオーディオ・ノード構造属性が、ブロック 1740 で、オーディオ関連デバイスから取り出され得る。仮想デバイスの詳細なオーディオ・ノード構造属性が、発見エンジンによってオーディオ関連デバイスから取り出され得る。ブロック 1742 において、仮想デバイスの詳細なオーディオ・ノード構造属性が、ノード構造ストレージに格納され得る。

10

## 【 0 2 3 6 】

動作は、図 18 に続く。そこで、未知の仮想デバイスが含まれているオブジェクトのリストが、ブロック 1744 において、オーディオ関連デバイスから取得され得る。同様に、ブロック 1722 (図 17) において、仮想デバイスが構成可能であるとき、構成可能な仮想デバイスに含まれているオブジェクトのリストが、ブロック 1744 において、オーディオ関連デバイスから取得され得る。オブジェクトのリストが、プロダクト・プラグイン、または発見エンジンによって取得され得る。

20

## 【 0 2 3 7 】

ブロック 1746 において、オブジェクトがオブジェクトのリストから選択される。選択されたオブジェクトが、プロダクト・プラグインのライブラリ・ストレージに存在しているかどうか決定され、ブロック 1747 で、既知であるかどうか決定される。オブジェクトが既知であるとき、オブジェクトに含まれている SV の実際の値がブロック 1748 で、取得され得る。SV の実際の値が取り出しエンジンによって取得され得る。ブロック 1750 において、オブジェクトが既知である表示およびその対応するオーディオ・ノード構造がノード構造ストレージに格納され得る。ブロック 1752 において、SV の実際の値が、コントロール・パネル・データ・ストレージに格納され得る。オブジェクトが、ブロック 1754 において、既知であるとき、仮想デバイスにおけるすべてのオブジェクトがチェックされたかどうか決定される。すべてのオブジェクトがチェックされていたとき、ブロック 1756 において、仮想デバイスにおけるオブジェクトに含まれていない SV があるかどうか決定される。オブジェクトに含まれていない SV があるとき、ブロック 1758 において、オブジェクトに含まれていない SV のリストがオーディオ関連デバイスから取得され、動作が、後述されるように、SV を選択し、SV が既知であるかどうかをチェックするためにブロック 1772 に進む。オブジェクトに含まれていない SV であって、仮想デバイスに含まれているその SV が無いとき、動作が、すべての仮想デバイスがチェックされたかどうかを決定するために図 17 のブロック 1730 に戻る。ブロック 1754 で、仮想デバイスにおけるすべてのオブジェクトがチェックされていないとき、動作は、オブジェクトのリストから別のオブジェクトを選択するためにブロック 1746 に戻る。

30

40

## 【 0 2 3 8 】

ブロック 1747 で、オブジェクトが未知であるとき、他のプロダクト・プラグインが、ブロック 1760 で、オブジェクトの知識に対してクエリされ得る。他のプロダクト・プラグインのクエリは、プラグイン・サーチ・エンジン、またはプロダクト・プラグインを用いて行われ得る。オブジェクトが、別のプロダクト・プラグインによって既知であるとき、オブジェクトは、ブロック 1762 で、取り出され、ライブラリ・ストレージに格納され得、動作が、SV に対する実際の値を取得するためにブロック 1748 に続く。ブロック 1760 において、オブジェクトが、別のプロダクト・プラグインによって既知でないとき、オブジェクトを記述する詳細なオーディオ・ノード構造属性が、ブロック 17

50

64で、オーディオ関連デバイスから取り出される。オブジェクトの詳細なオーディオ・ノード構造属性が、発見エンジンによってオーディオ関連デバイスから取り出され得る。ブロック1766において、オブジェクトの詳細なオーディオ・ノード構造属性は、ノード構造ストレージに格納され得る。

#### 【0239】

ブロック1768において、未知のオブジェクトおよび/または仮想デバイスに含まれているSVのリストが、オーディオ関連デバイスから取得され得る。ブロック1770において、SVの1つが、SVのリストから選択される。ブロック1722において、SVがライブラリ・ストレージに格納されているかどうか、従って、既知であるかどうかは決定される。SVが既知であるとき、SVの実際の値が、ブロック1744で、オーディオ関連デバイスから取得される。ブロック1776において、SVが既知である表示およびその対応するオーディオ・ノード構造がノード構造ストレージに格納され得る。加えて、ブロック1778で、SVの実際の値がコントロール・パネル・データ・ストレージに格納される。ブロック1780において、すべてのSVが既知であるかどうかを決定するために、SVのリストにおけるそれらのすべてのSVがチェックされたかどうかは決定される。すべてのSVがチェックされていないとき、動作は、オーディオ関連デバイスから取得されたSVのリストから別のSVを選択するためにブロック1770に戻る。オブジェクトにおけるSVのすべてがチェックされているとき、動作は、仮想デバイスにおけるオブジェクトがチェックされたかどうかをチェックするためにブロック1754に戻る。

#### 【0240】

ブロック1772において、SVが、プロダクト・プラグインによって既知でないとき、SVが、ブロック1782において、別のプロダクト・プラグインによって既知であるかどうかは決定される。他のプロダクト・プラグインのクエリは、プラグイン・サーチ・エンジン、またはプロダクト・プラグインを用いて行われ得る。別のプロダクト・プラグインによって既知であるとき、SVは、ブロック1784において、別のプロダクト・プラグインから取り出され、ライブラリ・ストレージに格納され、動作は、オーディオ関連デバイスからSVの実際の値を取り出すためにブロック1744に続く。SVが、別のプロダクト・プラグインによって既知でないとき、SVを記述する詳細なオーディオ・ノード構造属性が、ブロック1786において、オーディオ関連デバイスから取り出され、動作が、ノード構造ストレージにSVを格納するためにブロック1776に続く。SVの詳細なオーディオ・ノード構造属性が、発見エンジンによって、オーディオ関連デバイスから取り出され得る。

#### 【0241】

オーディオ関連デバイスのオーディオ・ノード構造における各SV、オブジェクトおよび仮想デバイス、または未知のSV、オブジェクトおよび仮想デバイスの詳細なオーディオ・ノード構造属性が、ノード構造ストレージに格納されるまで、処理は継続し得る。あるいは、制限された能力を有する「薄い」PEMの場合には、オーディオ・ノード構造または詳細な属性のより小さい部分が、いつでもノード構造ストレージに格納され得、追加的なオーディオ・ノード構造または詳細な属性が、必要に応じて格納され得る。

#### 【0242】

再度、図17のブロック1710を参照して、ノードのクラス属性が、プロダクト・プラグインのいずれによっても認識されないとき、ノードはジェネリック・プラグインにパスされる。図19では、ブロック1902において、ジェネリック・プラグインの取り出しエンジンは、ノード・クエリに回答して仮想デバイスのリストを受け取る。ブロック1904では、仮想デバイスが、仮想デバイスのリストから選択される。プラグイン・サーチ・エンジンが、ブロック1906において、選択された仮想デバイスの知識に対してプロダクト・プラグインにクエリするために有効にされ得る。あるいは、取り出しエンジンは、他のプロダクト・プラグインにクエリし得る。仮想デバイスがプロダクト・プラグインによって既知であり、構成可能でないとき(ブロック1908)、仮想デバイスのSVの実際の値がブロック1910で取得される。仮想デバイスが既知である表示およびその

10

20

30

40

50

対応するオーディオ・ノード構造が、ブロック1912において、ノード格納ストレージに格納され得る。仮想デバイス内のSVの実際の値が、ブロック1914において、コントロール・パネル・データ・ストレージに格納され得る。ブロック1916では、すべての仮想デバイスがチェックされたかどうか決定され、チェックされているとき、動作がブロック1918で終了する。すべての仮想デバイスがチェックされていないとき、動作が1904に戻り、リストから別の仮想デバイスを選択する。ブロック1906において、仮想デバイスが未知であるとき、詳細な属性がブロック1928で要求され、ブロック1930で、ノード構造ストレージに格納され得る。

#### 【0243】

ブロック1932において、未知の仮想デバイスにおけるオブジェクトのリストが取得される。同様に、ブロック1908において、仮想デバイスが構成可能であるとき、動作は、構成可能であるデバイスにおけるオブジェクトのリストを要求するためにブロック1932に進む。ブロック1934において、オブジェクトがリストから選択され、ブロック1936において、選択されたオブジェクトが、プロダクト・プラグインのいずれかによって既知であるかどうか決定される。オブジェクトが既知であるとき、オブジェクトに含まれているSVの実際の値が、ブロック1938で取得される。オブジェクトが既知である表示およびその対応するオーディオ・ノード構造が、ブロック1940において、ノード構造ストレージに格納される。ブロック1942では、オブジェクトに関連するSVの実際の値が、オーディオ関連デバイスから取得され、コントロール・パネル・データ・ストレージに格納され得る。ブロック1944で、オブジェクトのすべてがチェックされたかどうか決定される。すべてのオブジェクトがチェックされていないとき、動作はブロック1934に戻り、すべてのオブジェクトがチェックされているとき、ブロック1946でオブジェクトに含まれていない仮想デバイスにSVがあるかどうか決定される。SVがないとき、動作は、リストにおけるすべての仮想デバイスがチェックされたかどうかを決定するためにブロック1918に戻る。このようなSVがあるとき、動作は、後述されるように、SVを取得するためにブロック1966に進む。

#### 【0244】

ブロック1936において、オブジェクトが既知でないとき、オブジェクトの詳細な属性が、ブロック1960で取得され、ブロック1962でノード構造ストレージに格納され得る。未知のオブジェクトのSVのリストがブロック1964で取得され、リストにおけるSVの1つが、ブロック1966で選択される。ブロック1968において、SVが、プロダクト・プラグインの1つによって認識されているかどうか決定される。SVが既知であるとき、SVの実際の値が、ブロック1970でプロダクト・プラグインから取り出される。ブロック1972において、既知である表示およびオーディオ・ノード構造のその部分がノード構造ストレージに格納される。ブロック1974において、SVの実際の値が受け取られ、ブロック1976において、SVの実際の値が、コントロール・パネル・データ・ストレージに格納される。

#### 【0245】

ブロック1978において、すべてのSVがチェックされたかどうか決定される。すべてのSVがチェックされていないとき、動作は、別のSVを取得するためにブロック1966に戻る。SVのすべてがチェックされているとき、動作は、すべてのオブジェクトがチェックされたかどうかを決定するためにブロック1944に戻る。ブロック1968において、SVが既知でないとき、SVの詳細な属性がブロック1980で取得され、動作は、ブロック1972に続く。そこでは、SVの詳細な属性がノード構造ストレージに格納される。動作は、オーディオ・ノード構造が、既知であると識別されるか、またはさらなる発見が、オーディオ・ノード構造における未知のコンポーネントを決定するために行われるまで継続する。

#### 【0246】

IX. コントロール・パネル・ディスプレイ

図20は、PEMの能力が変化し得るとき、プロダクト属性を発見するためにシステム

10

20

30

40

50

において用いられ得る方法を示すフローチャートである。例えば、PEMは、そのディスプレイ上で、格納されている所定のGUIのコントロール・パネル・ビュー全体を表示することができないデバイス上に常駐し得る。別の例として、デバイスは、制限された処理能力を有し得、従って、同時に、格納されている所定のGUIのコントロール・パネル・ビュー全体の表示することを用意することは所望されないかもしれない。その代わりに、GUIのコントロール・パネル・ビューは、例えば、ユーザのコマンドに基づいて、必要に応じて生成され得る。従って、デバイス処理能力およびディスプレイのサイズのさまざまなレベルのスケラビリティがサポートされ得る。これはまた、図4を参照して記述されるように、「薄いアプリケーション層」と呼ばれる。

【0247】

図20では、ブロック2002において、特定のオーディオ関連デバイスのオーディオ・ノード構造のすべてがチェックされたかどうか、およびオーディオ・ノード構造の表示が、ノード構造ストレージに格納されているか否かが決定される。オーディオ・ノード構造のすべてがチェックされていないとき、動作はこのことが起きたかどうかを決定するために継続する。ノード構造ストレージが、オーディオ・ノード構造全体に対する表示を包含するとき、PEMが常駐するデバイスが、ブロック2004で、デバイス・ディスプレイ能力を決定するためにチェックされる。ブロック2006において、1つ以上の格納されている所定のコントロール・パネル・ビューが、オーディオ関連デバイスのオーディオ・ノード構造に対して存在するかが決定される。

【0248】

コントロール・パネル・ビューが存在するとき、格納されているコントロール・パネル・ビューが、ブロック2008で、表示デバイスの能力と互換性があるかが決定される。ブロック2010において、コントロール・パネル・ビューに含まれているオーディオ関連デバイスのコンポーネントの機能性のすべてが、所定のコントロール・パネル・ビューによってサポートされているかが決定される。それがサポートされているとき、ディスプレイがブロック2012でインスタンス化される。ブロック2014において、コントロール・パネル・ビューに含まれているSVの実際の値がコントロール・パネル・データ・ストレージから加えられる。インスタンス化されたコントロール・パネル・ビューが、ブロック2016において、表示のために格納され得る。表示デバイスの能力に基づいて、オーディオ・ノード構造のさらなるコントロール・パネル・ビューが、ブロック2018でインスタンス化されるべきか否かが決定される。これ以上のコントロール・パネル・ビューがインスタンス化されないとき、動作はブロック2020で終了する。さらなるコントロール・パネル・ビューがインスタンス化されるとき、動作は、ブロック2006に戻り、オーディオ関連デバイスのオーディオ・ノード構造に関連した別のコントロール・パネル・ビューをチェックする。

【0249】

ブロック2006に戻り、コントロール・パネル・ビューがオーディオ関連デバイスのオーディオ・ノード構造に対して存在しないとき、ジェネリック・プラグインは、ブロック2030で有効にされ得る。ブロック2032において、ジェネリック・プラグインは、ノード構造ストレージに格納されているオーディオ・ノード構造に基づいてジェネリック・コントロール・パネル・ビューを作成し得、動作が、ジェネリック・コントロール・パネル・ビューのディスプレイをインスタンス化するためにブロック2012に進む。ブロック2010に戻って、オーディオ・ノード構造の一部を表しているすべてのコンポーネントが、所定のコントロール・パネル・ビューにおいてサポートされていないとき、サポートされていないコンポーネントに対応する所定のGUIのアスペクトが、ブロック2036で、プロダクト・プラグインに格納されているかが決定される。格納されている所定のGUIのアスペクトが、プロダクト・プラグインに存在するとき、GUIのアスペクトが、ブロック2038で取り出される。ブロック2040において、取り出されたGUIのアスペクトは、格納されている所定のコントロール・パネル・ビューに統合される。統合されたGUIのアスペクトを含む所定のコントロール・パネル・ビューは、ブロック2

10

20

30

40

50

042において、オーディオ・デバイスのオーディオ・ノード構造に対応する新しい所定のコントロール・パネル・ビューとして格納され得る。

【0250】

ブロック2044において、格納されている所定のコントロール・パネル・ビューによってサポートされていないオーディオ・ノード構造のさらなるコンポーネントあるかどうか決定される。サポートされていないオーディオ・ノード構造のさらなるコンポーネントがあるとき、動作は、格納されている所定のGUIのアスペクトをチェックするためにブロック2036に戻る。オーディオ・ノード構造のさらなるコンポーネントがないとき、動作は、ブロック2012に進み、ディスプレイをインスタンス化する。

【0251】

再度、ブロック2008を参照して、格納されている所定のコントロール・パネル・ビューが、ディスプレイを駆動するデバイスの能力と互換性がないとき、ブロック2050において、コンポーネントが、互換性があるオーディオ・ノード構造から選択され得る。それから、動作は、GUIのアスペクトを取り出すためにブロック2036に続く。再度、ブロック2036を参照して、GUIのアスペクトがあらかじめ決定されておらず、プロダクト・プラグインに格納されていないとき、他のプロダクト・プラグインが、ブロック2052で、コントロール・パネル・ビューから欠けている機能性を表している格納されているGUIのアスペクトの存在に対してチェックされ得る。GUIのアスペクトがあらかじめ決定されており、別のプロダクト・プラグインに格納されているとき、動作は、格納されている所定のGUIのアスペクトを取り出し、処理するためにブロック2038  
20  
に戻る。GUIのアスペクトが他のいずれのプロダクト・プラグインに格納されていないとき、ジェネリック・プロダクトインが、ブロック2054で有効にされ得る。ブロック2056において、ジェネリック・プラグインは、ジェネリックGUIのアスペクトを生成し得、動作は、GUIのアスペクトを格納されている所定のコントロール・パネル・ビューに加えるためにブロック2040において継続する。デバイスの能力に左右されて、動作はオーディオ関連デバイス機能性が1つ以上のコントロール・パネル・ビューにインスタンス化されるまで継続する。

【0252】

図21は、PEMの能力が変化し得るとき、プロダクト属性を発見するために、このシステムに用いられ得る方法の別の例のフローチャートである。この例において、PEMは  
30  
、オーディオ関連デバイスの完全なる制御のためのすべてデータを包含する。仮想デバイスのリストがオーディオ関連デバイスから受け取られるとき、動作はブロック2102から始まる。ブロック2010において、PEMが動作しているデバイスは、デバイスが、仮想デバイス全体のリストを表示することが可能であるか否かがチェックされる。デバイスが、すべての仮想デバイスのリストを表示することが可能であるとき、仮想デバイス全体のリストが、ブロック2106で取得される。ブロック2108において、リストにおける仮想デバイスのすべてが表示される。ブロック2104において、デバイスが、すべての仮想デバイスを表示することが可能でないとき、表示され得る仮想デバイスの一部が、ブロック2110で取得され、ブロック2112で表示される。ブロック2114において、ユーザが他の仮想デバイスを見ることを所望するとき、現在見られている仮想デ  
40  
バイスが、ブロック2116で処分され得、動作は、さらなる仮想デバイスを取得し、表示するためにブロック2110に戻る。例えば、ユーザは、すべての仮想デバイスをレビューするためにデバイス上の「ページ・ダウン」ボタンを選択することによって他のデバイスを見得る。

【0253】

さらなる仮想デバイスが、ブロック2114において、見られないとき、ブロック2120で、ユーザが、表示されている特定の仮想デバイスに関してさらなる情報を取得することを所望するかどうか決定される。例えば、ユーザは、オーディオ・ノード構造おけるにオブジェクトであるコントロール・パネル・ビューの一部をクリックし得る。ユーザがクリックしないとき、動作は、ブロック2122で終了する。さらなる情報が所望され  
50

るとき、オーディオ関連デバイスに対するオブジェクトのリストが、ブロック 2 1 2 4 で要求され得る。

【 0 2 5 4 】

ブロック 2 1 2 6 において、PEM が常駐しているデバイスが、オブジェクト全体のリストを表示することが可能であるとき、オブジェクトのリスト全体が、ブロック 2 1 2 8 で、取得され得る。取得されたリスト上のオブジェクトが、ブロック 2 1 3 0 で表示され得る。ブロック 2 1 2 6 において、デバイスがオブジェクトのリスト全体を表示することが可能でないとき、オブジェクトの一部だけが、ブロック 2 1 3 2 で取得され、ブロック 2 1 3 4 で表示される。仮想デバイスに関して先述されるように、ユーザが、ブロック 2 1 3 6 でオブジェクトのリストをスクロールすることを所望するかどうかは決定される。ユーザがスクロールすることを所望するとき、ブロック 2 1 3 8 で、現在見られているオブジェクトが処分され、動作がブロック 2 1 3 2 に戻る。そこでは、さらなるオブジェクトが取得され、表示される。

10

【 0 2 5 5 】

図 2 2 を参照して、ブロック 2 1 4 0 では、ユーザがオブジェクトに対する SV の実際の値をレビューすることを所望するかどうかは決定さえる（図 2 2 は、図 2 1 のパス C および D から継続しているフローチャート図 2 1 の続きである）。ユーザがレビューすることを所望しないとき、動作は、ブロック 2 1 4 2 で終了する。ユーザが SV をレビューすることを所望するとき、SV のリストが、ブロック 2 1 4 4 で要求され、取得される。ブロック 2 1 4 6 では、取得されたリスト上のすべての SV が表示され得るかどうかは決定される。デバイスが、SV のリスト全体を表示することが可能であるとき、リスト全体が、ブロック 2 1 4 8 で取得され、ブロック 2 1 5 0 で表示される。それ以外の場合には、SV の一部がブロック 2 1 5 2 で取得され、ブロック 2 1 5 4 で表示される。ブロック 2 1 5 6 において、リストからさらなる SV が所望されるかどうかは決定される。所望されないとき、動作は、ブロック 2 1 5 8 で終了する。さらなる所望される SV があるとき、ユーザは、SV をスクロールし得る。現在表示されている SV が、ブロック 2 1 6 0 で処分される。動作は、それから、さらなる SV を取得し、表示するためにブロック 2 1 5 2 に戻る。

20

【 0 2 5 6 】

図 2 3 は、（図 8 および図 9 に示される）ノード・インスタンス化エンジンの動作の例を示すフローチャートである。プロダクト・プラグインが、ブロック 2 3 0 2 において、オーディオ・ノード構造に対応する格納されている所定のコントロール・パネル・ビューを有するかどうかは決定される。プロダクト・プラグインが格納されている所定のコントロール・パネル・ビューを有するとき、所定のコントロール・パネル・ビューが、ブロック 2 3 0 4 において、プロダクト・プラグインのコントロール・パネル・ライブラリから選択される。対応するコントロール・パネル・ビューがないとき、コントロール・パネル・ビューが、ブロック 2 3 0 6 において、ジェネリック・プラグインを用いて構築され得る。ノード構造のいずれかが、ブロック 2 3 0 8 において、コントロール・パネル・ビューから欠けているかどうかは決定される。ノード構造のいずれかが欠けているとき、ブロック 2 3 1 0 において、欠けている構造は、GUI のコアの発見エンジンを用いて充填される。ブロック 2 3 0 8 において、コントロール・パネル・ビューからの構造が完了するとき、ノード・インスタンス化エンジンは、ブロック 2 3 1 2 において、コントロール・パネル・ビューにおける SV の 1 つ以上へのサブスクリプションをセットアップし得る。

30

【 0 2 5 7 】

ブロック 2 3 1 4 において、ノード・インスタンス化エンジンは、コントロール・パネル・ビューにおける SV のリンク・マネージャに通知し得る。ブロック 2 3 1 6 において、表示デバイスが、さらなるオーディオ・ノード構造を格納し、かつ、コントロール・パネル・ビューをインスタンス化する能力を有するかどうかは決定される。表示デバイスが能力を有しないとき、動作はブロック 2 3 1 8 で終了する。表示デバイスが、追加的な能力を有するとき、ブロック 2 3 2 0 において、インスタンス化されていないさらなるオー

40

50

ディオ・ノード構造があるかどうか決定される。さらなるオーディオ・ノード構造がないとき、動作は、ブロック 2 3 1 8 で終了する。コントロール・パネル・ビューにおいてインスタンス化されるさらなるオーディオ・ノード構造があるとき、動作は、残りのオーディオ・ノード構造のコントロール・パネル・ビューの 1 つ以上の層がインスタンス化され、格納され得るようにブロック 2 3 0 2 に戻る。

#### 【 0 2 5 8 】

##### X . リンク・マネージャ

図 2 4 は、このシステムに用いられ得るリンク・マネージャのフローチャートである。S V が変化するとき、すべての関連したコントロール・パネル・ビューおよびデバイスが、S V の新しい値を有してアップデータされるように、この方法が用いられ得る。コントロール・パネル・ビューにおける S V の値が変化するとき ( ブロック 2 4 0 2 )、この新しい S V 値は、プロダクト・プラグインに送達され得る ( ブロック 2 4 0 4 )。プロダクト・プラグインは、ブロック 2 4 0 6 に示されるように、対応するコントロール・パネル・ビューにおいて S V の値をアップデートし得る。リンク・マネージャはまた、S V 値がブロック 2 4 0 8 に示されるように変化したことを通知され得る。リンク・マネージャは、コントロール・パネル・ビューが、変化された S V にサブスクライブされているか否かを決定し得る ( ブロック 2 4 1 0 )。S V とコントロール・パネル・ビューとの間でサブスクリプションが存在するとき、リンク・マネージャは、変化された S V に関連したコントロール・パネル・ビューのそれぞれに注意を促し得る ( ブロック 2 4 1 2 )。注意を促されたコントロール・パネル・ビューは、それから、S V の新しい値を決定するためにプロダクト・プラグインまたは別のデバイスにクエリし得る ( ブロック 2 4 1 4 )。リンク・マネージャはまた、いずれかのデバイスが、変化された S V にサブスクライブしているか否かを決定し得る ( ブロック 2 4 1 6 )。S V とデバイスとの間でサブスクリプションが存在するとき、リンク・マネージャは、変化された S V に関連したデバイスのそれぞれに注意を促し得る ( ブロック 2 4 1 8 )。注意を促されたコントロール・パネル・ビューは、それから、S V の新しい値を決定するためにプロダクト・プラグインまたは別のデバイスにクエリし得る ( ブロック 2 4 2 0 )。

#### 【 0 2 5 9 】

##### X I . サブスクリプション

図 2 5 は、デバイス間のサブスクリプションが、1 つ以上の他のオーディオ関連デバイスを用いて 1 つ以上のオーディオ関連デバイスにおけるパラメータをモニタおよび / または制御するように用いられ得る方法を示すブロック図である。サブスクリプションは、先述されるように 1 つ以上のサブスクライブするものに、S V の値における変化を通知するために用いられ得る。加えて、サブスクリプションは、1 つ以上のデバイスにおける S V の値の変化を指示するために用いられ得る。さらに、モニタおよび制御するためにサブスクリプションの組み合わせが、オーディオ関連デバイス間でセットアップされ得る。図 2 5 に示される例は、第 1 のオーディオ関連デバイス 2 5 0 2、第 2 のオーディオ関連デバイス 2 5 0 4、および第 3 のオーディオ関連デバイス 2 5 0 6 を含む。

#### 【 0 2 6 0 】

第 2 のオーディオ関連デバイス 2 5 0 4 は、第 1 のオーディオ関連デバイス 2 5 0 2 とサブスクライブ・ストリングをセットアップする。サブスクライブ・ストリングは、図 1 5 に示されるような S V S u b s c r i b e S t r i n g メッセージといったサブスクリプション・メッセージ 2 5 0 8 を用いてセットアップされ得る。人間読み取り可能ストリングが、サブスクリプション応答メッセージ 2 5 1 0 に含まれ得る。S V S u b s c r i b e S t r i n g メッセージの例は、以下を含む。

#### 【 0 2 6 1 】

【表 6】

バージョン
ヘッダーの長さ
メッセージの長さ
ソース・アドレス
目的アドレス
メッセージID
フラグ
ホップ・カウント
シーケンス・ナンバー
ペイロード...
サブスクライブするもののアドレス
サブスクリプションのNo.
第1のサブスクライブされるもののSVID
第1のサブスクライブするもののSVID
第2のサブスクライブされるもののSVID
第2のサブスクライブするもののSVID

10

20

表 6

この図解される例では、ソース・アドレスは、第1のオーディオ関連デバイス2502のノードIDであり、目的アドレスは、第2のオーディオ関連デバイス2504のノードIDであり得る。メッセージIDは、サブスクライブ・ストリング・メッセージとしてメッセージを識別し得る。フラグ、ホップ・カウントおよびシーケンス・ナンバーは、オーディオ・システム特定の値であり得、ペイロードは、サブスクライブ・ストリング要求メッセージを含み得る。サブスクライブするもののアドレスは、第1のオーディオ関連デバイス2502のオーディオ・ノード・アドレスであり得る。サブスクリプションの数は、どれだけの数の第1のオーディオ関連デバイス2502へのサブスクリプションがサブスクライブ・ストリング要求に含まれているか示し得る。第1のサブスクライブするもののSVIDは、第2のオーディオ関連デバイス2504におけるSVであって、第1のオーディオ関連デバイス2502における第1のサブスクライブされるもののSVIDに対応するSVを示し得る。

30

## 【0262】

40

サブスクライブ・ストリングが、第1のオーディオ関連デバイス2502における1つ以上のSVと、対応する1つ以上の第2のオーディオ関連デバイス2504におけるSVとの間でマッピングされ得る。サブスクライブ・ストリングは、サブスクリプションに基づいて提供されるデータが、ユーザによって容易に解読され得るフォーマット（人間読み取り可能フォーマット）であるサブスクリプションを要求し得る。従って、サブスクリプションに関連したSVの値が変化するとき、新しい値が、サブスクリプション応答メッセージ2510で、第2のオーディオ関連デバイス2504に送達され得る。新しい値が表示される前に、第2のオーディオ関連デバイス2504が、複数のデータ・タイプ間で変換することを必要されないように、その値は人間読み取り可能ストリングとして送達さえ得る。

50

## 【 0 2 6 3 】

第3のオーディオ関連デバイス2506は、サブスクリプション・メッセージ2512を用いて第1のオーディオ関連デバイス2502とサブスクリプションをセットアップし得る。サブスクリプション・メッセージは、値、または割合であるサブスクリプション応答を要求し得る。第3のオーディオ関連デバイス2506は、第1のオーディオ関連デバイス2502におけるサブスクライブされるもののSVのデータ・タイプを知っているとき、サブスクリプション・メッセージは、所定の条件に基づいて実際の値を要求し得る。第3のオーディオ関連デバイス2506のノード構造のコンポーネントが第1のオーディオ関連デバイス2502においてミラーされるとき、データ・タイプは知られていないかもしれない。第3のオーディオ関連デバイス2506は、サブスクライブされるもののSVのデータ・タイプを知らないとき、サブスクリプション・メッセージが、割合で要求され得る。

10

## 【 0 2 6 4 】

第1のオーディオ・デバイス2502がSVの値を送達するのではなく、0%から100%間で変化の割合が、サブスクリプション応答メッセージ2514として送達され得る。従って、ユニット特定の変数ではなくユニット特定の変数を表している割合が送達され、かつ、受け取られ得る。その割合に基づいて、第3のオーディオ関連デバイス2506が、識別された割合分、現存の値を調整し得る。あるいは、第3のオーディオ・デバイスは、その割合から実際の値を決定するために、その割合に対応する実際の値のレンジを有し得る。従って、第3のオーディオ・デバイスは、割合が用いられるとき、データ・タイプおよび/またはユニット値を知ることが必要としない。

20

## 【 0 2 6 5 】

図25において、第1のオーディオ関連デバイス2502は、サブスクライブ・メッセージ2514を用いて第3のオーディオ関連デバイス2506におけるSVの値をセットし得る。SVの値が、SVの実際の値を送達する、または割合を送達することによって変化され得る。しかしながら、オーディオ関連デバイスを割合を用いてセットする利点はサブスクライブされるもののオーディオ関連デバイスが、データを送達するサブスクライブされるもののオーディオ関連デバイスデータ・タイプを知る必要がないことである。

## 【 0 2 6 6 】

この例のために、第2のオーディオ関連デバイス2504が、第1のオーディオ関連デバイス2502のSVに対してサブスクライブ・ストリングをセットアップすると仮定する。さらに、第3のオーディオ関連デバイス2506が、SVの値をセットするように構成されていると仮定する。このセットアップでは、第3のオーディオ関連デバイス2506からのSVのセットポイントが、人間読み取り可能形式で、変化されたSV値を有する第1のオーディオ関連デバイス2502から第2のオーディオ関連デバイス2504に出力を生成する。

30

## 【 0 2 6 7 】

第3のオーディオ関連デバイス2506および第2のオーディオ関連デバイス2504の双方は、SVのデータ・フォーマットを必ずしも知る必要はない。いくつかの状況において、第3のオーディオ関連デバイス2506は、SVのデータ・フォーマットまたはデータ・タイプを知り得る。このような場合において、第3のオーディオ関連デバイス2506は、変化された値をSVのデータ・フォーマットで、そのSVに特別に割り当て得る。しかしながら、他の例において、第3のオーディオ関連デバイス2506は、SVのデータ・フォーマットを知らないかもしれない。このような例において、第3のオーディオ関連デバイス2506は、第1のオーディオ関連デバイス2502のノード構造にクエリすることによって、SVのデータ・フォーマットを発見し得る。第1のオーディオ関連デバイスが、SVのデータ・フォーマットを決定することを試みるか否かは、第3のオーディオ関連デバイス2506の処理能力または他の要因に依存し得る。SVのデータ・フォーマットが第3のオーディオ関連デバイス2506に対して未知であるとき（および第3のオーディオ関連デバイス2506がデータ・フォーマットを決定できないとき）、第3

40

50

のオーディオ関連デバイスは依然として、相対的な割合でS Vの値をセットすることによってS Vの値を変化し得る。

【0268】

第3のオーディオ関連デバイス2506は、相対的な割合で第1のオーディオ関連デバイス2502にS Vの値を変化するように命令することによってそのS Vの値をセットし得る。例えば、第3のオーディオ関連デバイス2506は、第1のオーディオ関連デバイスに、25%分、S Vの値を増加するように命令し得る。この態様では、第3のオーディオ関連デバイス2506は、S Vの値をセットするためにS Vのデータ・フォーマットを知る必要はない。S Vをセットする第3のオーディオ関連デバイス2506に回答して、第1のオーディオ関連デバイス2502は、人間読み取り可能ストリングを、フィードバック表示として第2のオーディオ関連デバイス2504に出力し得る。フィードバック表示の例は、その割合に回答して変化する出力である。人間読み取り可能ストリングは、第2のオーディオ関連デバイス2504によるさらなる処理を施すことなく、ユーザによって理解され得るので、第2のオーディオ関連デバイス2504は、S Vのデータ・フォーマットを知ることなく、S Vの変化された値をユーザに表示し、それ以外の場合には、伝え得る。あるいは、第2のオーディオ関連デバイス2504が、不フィードバック表示の所定のデータ・タイプを有するように構成されている際には、第1のオーディオ関連デバイス2502は、適切なユニットでフィードバック値を取得するために、第2のオーディオ関連デバイス2504によって処理される値を出力し得る。

【0269】

先例は、第3のオーディオ関連デバイス2506におけるセットポイント制御入力S Vであって、サブスクリプションを用いて、第1のオーディオ関連デバイス2502におけるセットポイントS Vを変化するために用いられるセットポイント制御入力S Vの点から記載されている。他の例において、サブスクリプションは、第3のオーディオ関連デバイス2506の動作パラメータS Vであるモニタ値S Vであって、第1のオーディオ関連デバイス2502を用いて表示されるように構成されているモニタ値S Vを変化するために用いられ得る。

【0270】

さらなる別の例において、サブスクリプションは、第1のオーディオ関連デバイスにおけるセットポイントS Vにマッピングされている、およびセットポイントS Vを調整するセットポイント制御入力S Vであって、第3のオーディオ関連デバイスにおけるセットポイント制御入力S Vを調整するように用いられ得る。加えて、セットポイントS Vに回答する第1のオーディオ関連デバイス2502における動作パラメータS Vは、第3のオーディオ関連デバイス2506におけるモニタ値S Vにマッピングされ得る。従って、動作中に、第3のオーディオ関連デバイス2506は、第1のオーディオ関連デバイス2502のセットポイントS Vを調整し、モニタ値S Vを有する第1のオーディオ関連デバイスの出力(動作パラメータS V)をモニタし得る。先述されるように、動作パラメータS Vはまた、第2のオーディオ関連デバイス2504にマッピングされ得る。

【0271】

XII. 薄いパフォーマンス機器マネージャ

図26は、オーディオ・ネットワーク120を有するオーディオ・システムにおいて用いられ得る能力ベースのPEM2600のブロック図である。能力ベースのPEM2602は、図5に示される「薄いアプリケーション層」のデバイスに対応し、図5に示されるPEMの例と同一のコンポーネントの多くを有する。従って、簡潔にするために、以下の記載は、図26の「薄い」PEMと図5の「厚い」PEMとの間の違いを中心とする。

【0272】

ネットワーク・トラフィックの制約またはPEMが動作しているデバイスの能力のために、能力ベースのPEM2600は、制限された態様で動作するかもしれない。例えば、PEMが動作するデバイスの処理速度、メモリ・ストレージおよび/またはディスプレイ能力が、能力ベースのPEM2600の動作を制限し得る。先述されたように、ノードに

10

20

30

40

50

対するすべての仮想デバイス、オブジェクトおよびSVに対するコントロール・パネル・ビューを、一度にインスタンス化するのではなく、オーディオ・ノード構造の異なる部分がユーザによって表示されるように要求されるので、能力ベースのPEM2600は1つずつそのようにし得る。ノード構造を発見し、かつ、対応するコントロール・パネル・ビューをインスタンス化する能力ベースのPEM2600の動作の例が、図20~23を参照して先述された。

#### 【0273】

図26において、能力ベースのPEM2600は、ネットワーク・インターフェース2602、ノード分解能モジュール2604、システム・プラグイン・ノード・マネージャ2606、リンク・マネージャ発見マネージャ2608、発見エンジン2610およびリンク・マネージャ2612を含む。この図解された例において、この例の能力ベースのPEM2602の能力がプロダクト・プラグインを提供しないので、能力ベースのPEM2602は、プラグイン・ローダを含まないかもしれず、ジェネリック・プラグイン524とだけ通信し得る。加えて、未知のコンポーネントおよび/またはコントロール・パネル・ビューもしくはアスペクトを検索する他のプロダクト・プラグインがないので、ノード発見マネージャ2608は、属性要求エンジン2620だけ含む。

#### 【0274】

オーディオ関連デバイスの属性を要求すると、属性要求エンジンは、属性をジェネリック・プラグイン524にパスし得る。ジェネリック・プラグイン524は、オーディオ関連デバイスへの通信パスをセットアップするようにAPIインターフェースを有効にし、かつ、能力ベースのPEM2600が動作しているデバイスの能力に依存しているオーディオ関連デバイスのオーディオ・ノード構造のコンポーネントのいくつかの部分を要求するようにAPIインターフェースを有効にし得る。ジェネリック・プラグインは、オーディオ・ノード構造の部分に対して対応するコントロール・パネル・ビューを決定し、ディスプレイをインスタンス化し得る。さらなるオーディオ・ノード構造が、ユーザによって見られることを要求されるとき、現在のコントロール・パネル・ビューが処分され、さらなる情報が、別のコントロール・パネル・ビューの生成をサポートするようにオーディオ関連デバイスから要求され得る。あるいは、能力ベースのPEM2600は、十分な処理能力が利用可能なときには構築され得るコントロール・パネル・ビューのセットのサブセットを構築し得る。コントロール・パネル・ビューのサブセットは、能力ベースのPEM2600が動作しているデバイスの能力に依存して1つ以上のコントロール・パネル・ビューを含み得る。

#### 【0275】

能力ベースのPEM2600は、生成されるコントロール・パネル・ビューの複雑さを考慮し得る。ディスプレイおよび/または処理能力が制限されているとき、能力ベースのPEM2600は、比較的シンプルなGUIのアスペクトを含むコントロール・パネル・ビューを生成し得る。例えば、自動的に作成された三次元GUIのアスペクトが自動的に作成されない二次元ボックスに変換され得る。加えて、能力ベースのPEM2600は、表示デバイスの能力に基づいて、コントロール・パネル・ビューに含まれているエレメントの大きさおよび/または数を調整し得る。さらにコントロール・パネル・ビューを生成するためにオーディオ関連デバイスから取得されるオーディオ・ノード構造の量は、表示デバイスの能力に基づいて、能力ベースのPEM2600によって変化され得る。

#### 【0276】

先述されるように、このシステムは、種々の複雑さを持つ複数のオーディオ・コンポーネントを有するオーディオ・システムにおいて動作可能である。このシステムは、さまざまな物理ネットワークおよび通信プラットフォームのいずれかを用いて、種々の複雑さを有する複数の異なるオーディオ関連デバイスをシームレスに統合するネットワーク独立プロトコルを提供する。加えて、このシステムは、まだ既知でない特定のオーディオ関連デバイスのアスペクトを発見するメッセージング機能を提供することによって、システム内のオーディオ関連デバイスのアップグレード、変更、追加が可能である。このメッセー

10

20

30

40

50

能はまた、選択されたオーディオ関連デバイス間でオーディオデータおよび関連情報が効率的に、および最小のネットワーク・トラフィックで送達されることを可能にする。

【0277】

P E Mは、オーディオ関連デバイスのオーディオ・ノード構造が既知であるかどうかを決定し得る。オーディオ・ノード構造のいずれかの部分が未知であるとき、P E Mは、未知の部分を実験的に定義するようにさらなる情報を取得し得る。加えて、P E Mは、オーディオ・システム全体のコントロール・パネル・ビューを取り出しおよび/または生成し得る。さらに、P E Mは、対応するオーディオ・ノード構造に基づいて、オーディオ・システム内の各特定のオーディオ関連デバイスに対してコントロール・パネル・ビューを取り出しおよび/または生成し得る。コントロール・パネル・ビューは、オーディオ・システム内のオーディオ関連デバイスの協調された管理、制御およびモニタを可能にする。このシステム内で用いられるに通信プロトコルは、デバイス・ツー・デバイス通信と同様に、パフォーマンスマネージャとオーディオ関連デバイス間の標準通信を可能にする。

10

【0278】

本発明のさまざまな実施形態が記述されてきた一方で、本発明の範囲内において、より多くの実施形態およびインプリメンテーションが可能であることが当業者には明らかである。従って、本発明は、添付の請求項およびその均等物によってのみ制限される。

【0279】

このシステムには、リンク・マネージャが含まれている。このリンク・マネージャは、他の複数のオーディオ関連システムまたはコントロール・パネル・ビューへのサブスクリプションを管理するように構成され得る。管理されているデータ変数が変化するとき、リンク・マネージャ・モジュールは、この変化を他のオーディオ関連デバイスおよび/またはコントロール・パネル・ビューに通知し得る。この通知に回答して、他のオーディオ関連デバイスおよびコントロール・パネル・ビューは、変化したデータ変数を用いて、オーディオ関連デバイスにサブスクライブし、アップデートされたデータ変数の受け取りを要求し得る。いくつかの例において、このリンク・マネージャはまた、特定のオーディオ関連デバイスに対してサブスクリプションの負荷を減らすように構成されている。

20

【産業上の利用可能性】

【0280】

本発明によって、ユーザーは、システムの操作、メンテナンス、およびトラブルシューティングが、より容易になる。このことは、特に、ツアー中の音楽グループの場合といった、さまざまな構成で繰り返し分解され、再構築されるオーディオ・システムにおいて有用である。また、いくつかのオーディオ関連デバイスを包含するオーディオ・システムをセットアップし、かつ、制御するために用いられるグラフィカル・ユーザ・インターフェース(G U I)が、すぐに用いるために単純で、簡単になる。

30

【図面の簡単な説明】

【0281】

本発明は、図面および詳細を参照することにより、より理解される。図面におけるコンポートは、必ずしも一定に縮小拡大するのではなく、その代わりに、強調は、本発明の原理を示すことに置かれる。さらに、図面において、同様の参照番号は、異なる図面を通して対応するパーツを示す。

40

【図1】オーディオ・システムの一例のブロック図である。

【図2】図1のオーディオ・システムに図解されるオーディオ関連デバイスの1つのブロック図の例である。

【図3】図2オーディオ関連デバイスおよび図1に示されるオーディオ関連デバイスの一部のより詳細なブロック図である。

【図4】オーディオ関連デバイスを表すオーディオ・ノード構造の例のブロック図である。

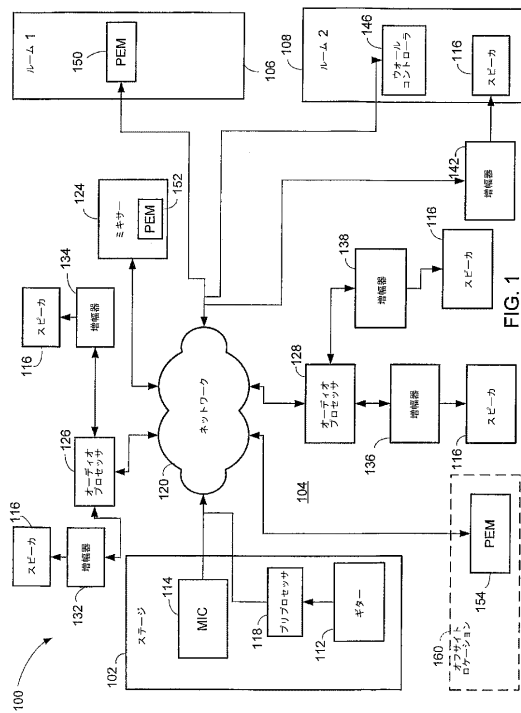
【図5】P E Mの例および図1に示されるオーディオ・システムの一部のブロック図である。

50

- 【図6】図5のPEMのGUIのコアの一部の詳細なブロック図である。
- 【図7】図5のPEMのインターフェース・モジュールの一部のブロック図である。
- 【図8】図5のPEMのプロダクト・プラグインの一部のブロック図である。
- 【図9】図5のPEMのジェネリック・プラグインの一部のブロック図である。
- 【図10】図2のオーディオ関連デバイスの一部のより詳細なブロック図である。
- 【図11】図1のオーディオ・システムにおけるオーディオ関連デバイスを発見するためのフローチャートの例である。
- 【図12】図3のスイッチおよび複数のオーディオ関連デバイスの構成の例のより詳細なブロック図である。
- 【図13】オーディオ関連デバイスとの通信パスを決定するためのフローチャートの例である。 10
- 【図14】オーディオ・デバイス間の通信のためのプロトコル構造の例である。
- 【図15】図14に図解されるプロトコルにおけるオーディオ関連デバイス間で通信され得るアプリケーション層のメッセージおよびアドレス層のメッセージIDの例のテーブルである。
- 【図16】図14に図解されるプロトコルにおけるオーディオ関連デバイス間で通信され得るルーティング層のメッセージおよびルーティング層のメッセージIDの例のテーブルである。
- 【図17】プロダクト・プラグインと関連したオーディオ関連デバイスのオーディオ・ノード構造を決定するためのフローチャートの例である。 20
- 【図18】プロダクト・プラグインと関連したオーディオ関連デバイスのオーディオ・ノード構造を決定するための図17のフローチャートの例の第2の部分である。
- 【図19】プロダクト・プラグインと関連していないオーディオ関連デバイスのオーディオ・ノード構造を決定するためのフローチャートの例である。
- 【図20】表示デバイスの能力に基づいて、どのように対応するオーディオ関連デバイスのGUIのコントロール・パネル・ビューが表示され得るかを決定するためのフローチャートの例である。
- 【図21】表示デバイスの能力に基づいて、どのように対応するオーディオ関連デバイスのGUIのコントロール・パネル・ビューが表示され得るかを決定するためのフローチャートの別の例である。 30
- 【図22】表示デバイスの能力に基づいて、どのように対応するオーディオ関連デバイスのGUIのコントロール・パネル・ビューが表示され得るかを決定するための図20のフローチャートの例の第2の部分である。
- 【図23】対応するオーディオ関連デバイスのコントロール・パネル・ビューを構築するためのフローチャートの例である。
- 【図24】異なる複数のコントロール・パネル・ビューに表示されるデータ変数をアップデートするためのフローチャートの例である。
- 【図25】1つのオーディオ関連デバイスから別のオーディオ関連デバイスへのデータ変数の通信のさまざまな例を示すフローチャートである。
- 【図26】図4のPEMのGUIのコアの部分の別の例の詳細なブロック図である。 40
- 【図27】例示的なシステムのGUIベニュー・ビューを示すコントロール・パネル・ビューの例である。
- 【図28】図27に示される製品のプロダクト・ビューを示すコントロール・パネル・ビューの例である。
- 【図29】例示的なシステムのGUIプロダクト・ビューを示すコントロール・パネル・ビューの例である。
- 【符号の説明】
- 【0282】
- 100 オーディオ・システム
- 114 マイクロフォン 50

1 1 6	スピーカ	
1 2 6、1 2 8	オーディオ・プロセッサ	
2 0 2	通信ポート	
2 0 6	メモリ	
3 0 2、5 0 4	ネットワーク・インターフェース	
3 3 0、3 3 2	スイッチ	
4 0 0	オーディオ・ノード構造	
4 0 4、4 0 6	仮想デバイス	
5 0 8	E t h e r n e t (登録商標)	
5 1 0	F i r e W i r e	10
5 1 2	U S B 5 1 2	
5 1 4	R S 2 3 2	
5 1 6	R S 4 8 5	
5 1 8	ルーティング・テーブル	
5 2 4	ジェネリック・プラグイン	
6 0 0	ノード分解能モジュール	
6 0 2	ノードIDエンジン	
6 1 2	サブスクリプション・マネージャ	
6 1 4	リンク・マネージャ	
6 1 6	リンク・エンジン	20
6 1 8	イベント・エンジン	
6 2 0	発見エンジン	
6 2 2	S V 発見モジュール	
6 2 4	オブジェクト発見モジュール	
6 2 6	仮想デバイス発見モジュール	
6 2 8	ノード発見マネージャ	
6 3 0	ノード属性要求エンジン	
6 3 2	プラグイン・サーチ・エンジン	
8 0 2	ノード・インスタンス化エンジン	
8 0 4	コントロール・パネル・ライブラリ	30
8 0 8	コントロール・パネル・ビュー・バッファ	
8 1 8	ノード構造ストレージ	
9 0 4	ノード・インスタンス化エンジン	
9 1 0	コントロール・パネル・ビュー・バッファ	
9 1 2	ノード構造ストレージ	
1 0 0 8	ノード識別モジュール	

【図1】



【図2】

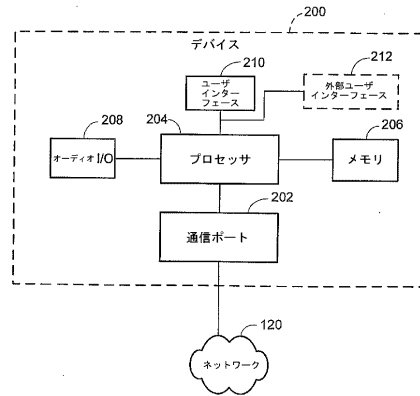


FIG. 2

【図3】

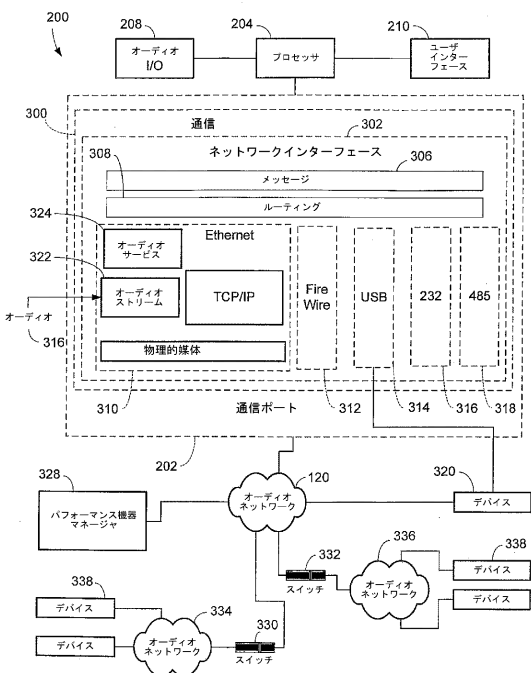


FIG. 3

【図5】

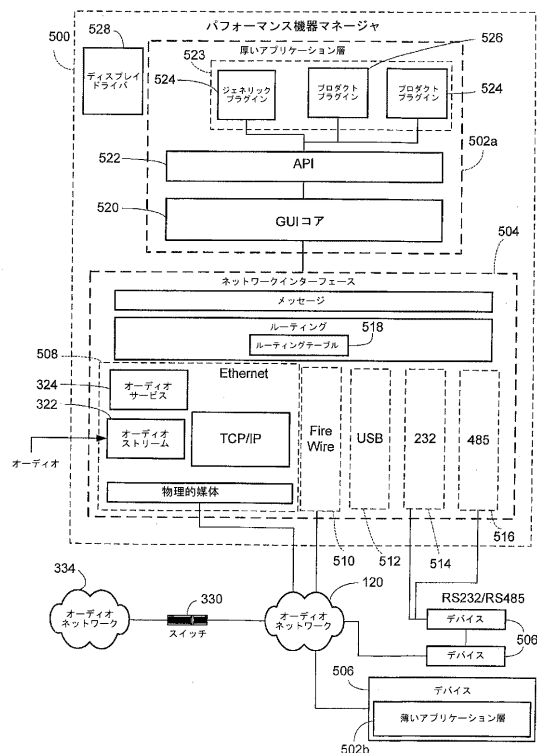


FIG. 5

【図6】

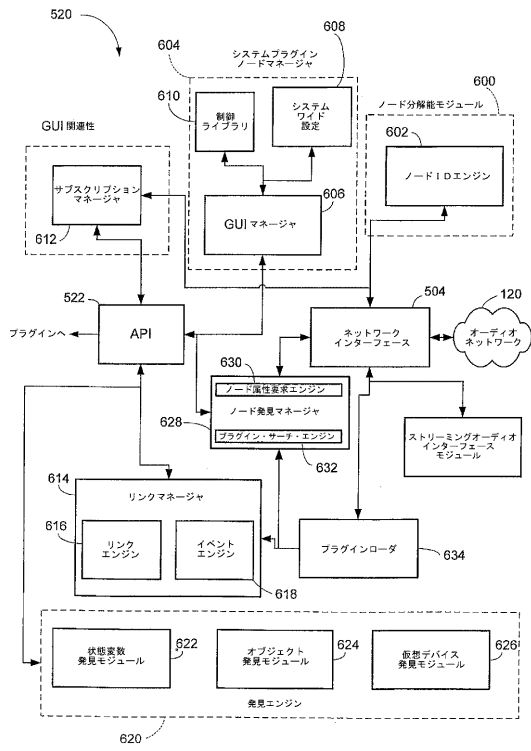


FIG. 6

【図7】

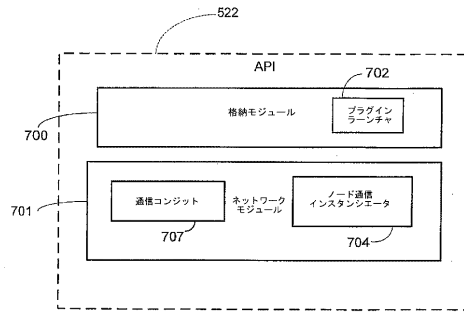


FIG. 7

【図8】

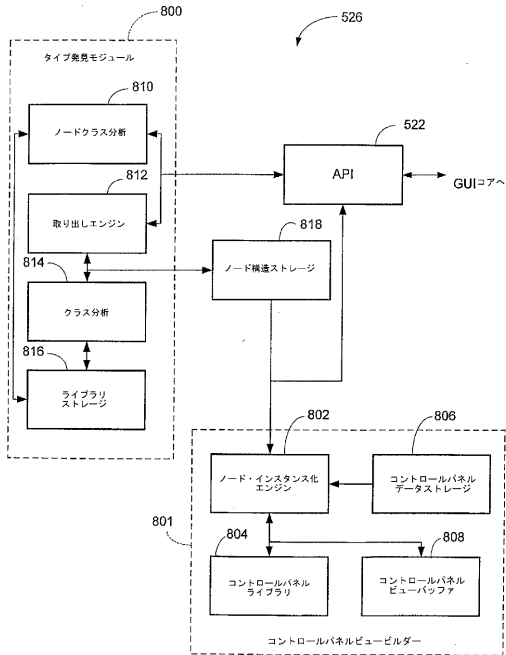


FIG. 8

【図9】

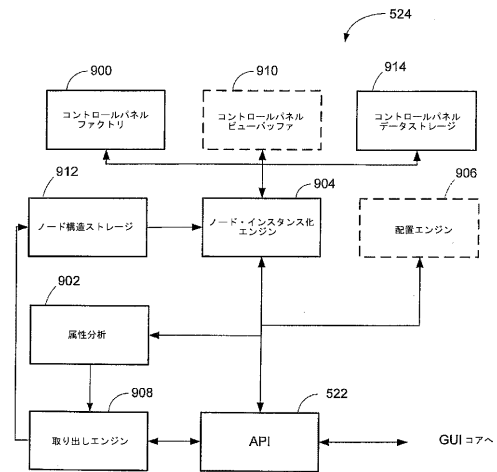


FIG. 9

【図10】

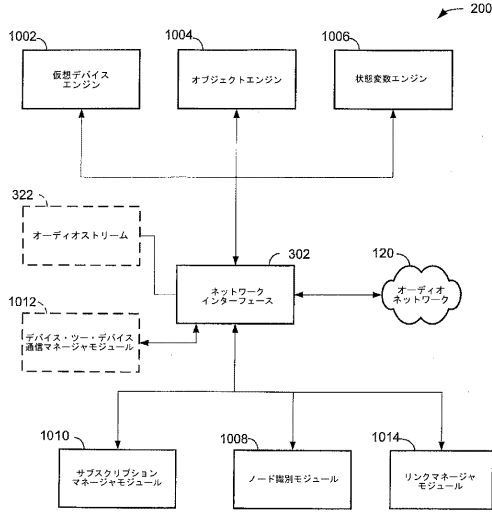


FIG. 10

【図11】

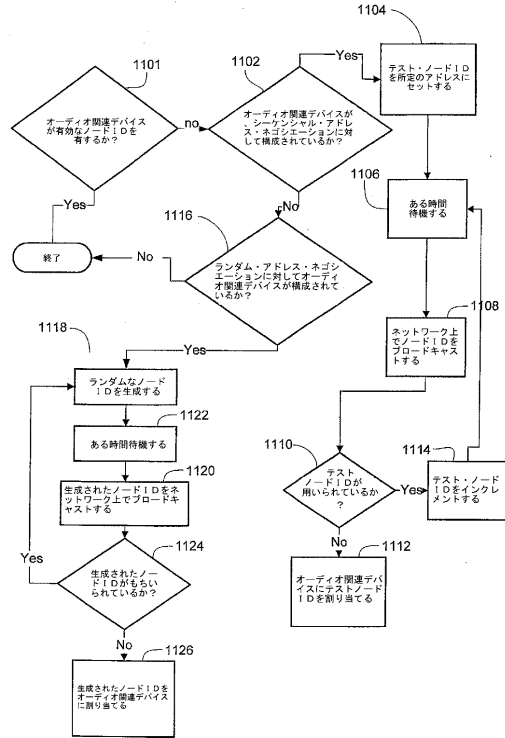


FIG. 11

【図12】

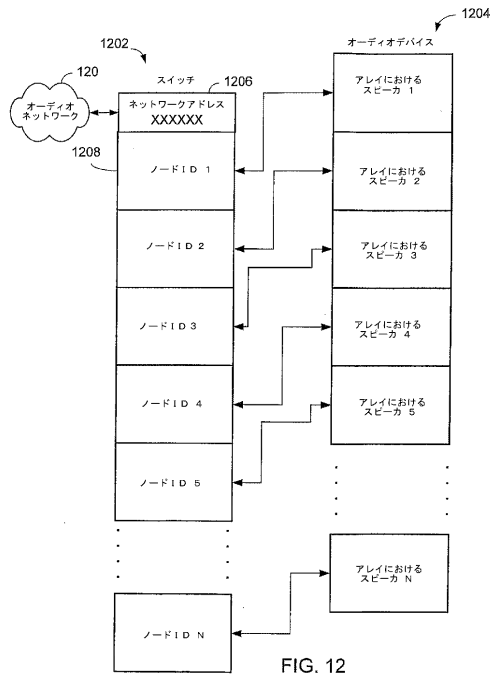


FIG. 12

【図13】

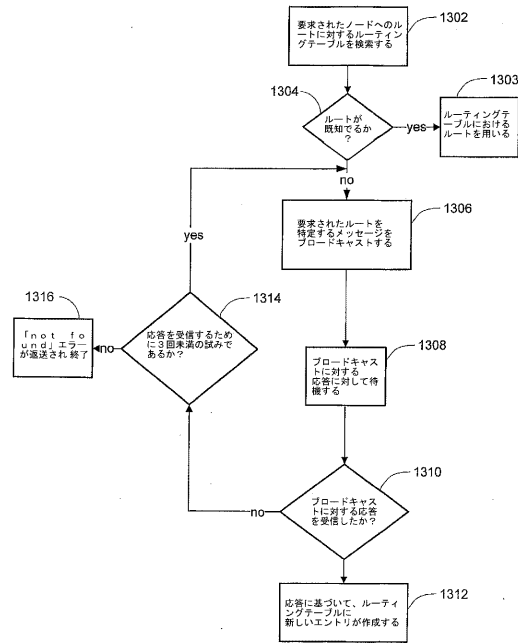


FIG. 13

【 図 14 】

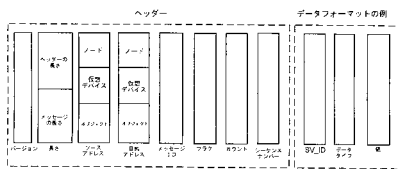


FIG. 14

【 図 15 】

MultiSVSet	0x0100	状態数をセットする
MultiObjectSVSet	0x0101	複数のオブジェクトに状態数をセットする
MultiSVSetPercent	0x0102	法則の割合として状態数をセットする
MultiSVGet	0x0103	状態数を取得する
MultiSVGetPercent	0x0104	法則の割合として状態数を取得する
MultiSVGetValueString	0x0105	法則の割合として状態数を取得する
MultiSVBump	0x0106	法則の割合として状態数を増やす
MultiSVBumpPercent	0x0107	法則の割合として状態数を増やす
RequestAddress	0x0108	ノードIDを要求する
AddressUsed	0x0109	ノードIDが利用されていることを示す
SetAddress	0x010A	ネットワークアドレスおよびノードIDをセットする
MultiSVGetAttributes	0x010B	状態数から「n」の属性値を取得する
MultiSVSetAttributes	0x010C	状態数に「n」の属性値をセットする
GetAttributes	0x010D	オブジェクトまたはIDから「n」の属性値を取得する
SetAttributes	0x010E	オブジェクトまたはIDに「n」の属性値をセットする
MultiSVSubscribe	0x010F	状態数の変化にサブスクライブする
MultiSVSubscribeString	0x0110	状態数値ストリングにサブスクライブする
MultiSVSubscribePercent	0x0111	状態数割合にサブスクライブする
MultiSVUnsubscribe	0x0112	状態数からアンサブスクライブする
SVSubscribeAll	0x0113	状態デバイスおよび状態からアンサブスクライブする
SVUnsubscribeAll	0x0114	状態デバイスおよび状態からアンサブスクライブする
SubscribeErrors	0x0115	エラーにサブスクライブする
ErrorSubscriptionData	0x0116	エラーレポートを保持するメッセージ
CreateVD	0x0117	仮想デバイスをインスタンス化するメッセージの開始
EndCreateVD	0x0118	仮想デバイスをインスタンス化するメッセージの終了
DescribeVD	0x0119	仮想デバイスの詳細な記述の要求
GetVDList	0x011A	ノードにおける仮想デバイスのリストの取得
SetSVList	0x011B	VDまたはオブジェクトにおいて状態数をセットする
GetSVList	0x011C	VDまたはオブジェクトにおいて状態数を取得する
SetObjectList	0x011D	VDまたはオブジェクトにおいてオブジェクトをセットする
GetObjectList	0x011E	VDまたはオブジェクト内のオブジェクトを取得する
GetVDIOList	0x011F	VD間のオービタルチェンジをセットする
SetVDIOList	0x0120	VD間のオービタルチェンジをセットする
GetObjectList	0x0121	オブジェクト間のオービタルチェンジを取得する
SetObjIOList	0x0122	オブジェクト間のオービタルチェンジをセットする
GetControlTable	0x0123	制御テーブルを取得する
Store	0x0124	情報を保持する
Recall	0x0125	情報をリコールする
SetVenueTable	0x0126	ベニューテーブルをセットする
GetVenueTable	0x0127	ベニューテーブルを取得する

FIG. 15

【 図 16 】

DiscoInfo	0x0000	ノードを見つけて、ルーティング情報を交換する
DiscoFailSafe	0x0001	「交付している」新しいノードを抽出する
GetNetworkInfo	0x0002	ネットワークインターフェース上で情報を取得する
SetProxy	0x0003	プロキシとして働くようにノードに知らせる
RequestAddress	0x0004	特定のノードIDの使用を要求する
AddressUsed	0x0005	ノードIDが利用されていることを通知する
SetAddress	0x0006	ネットワークアドレスおよびノードIDをセットする

FIG. 16

【 図 17 】

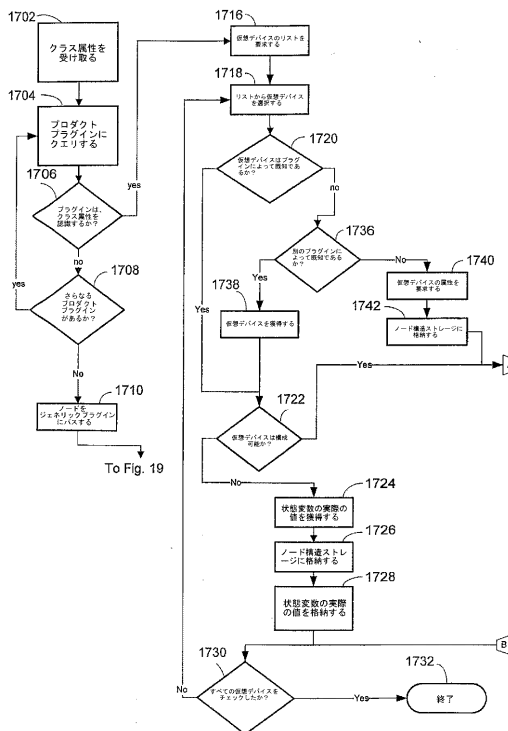


FIG. 17

【図18】

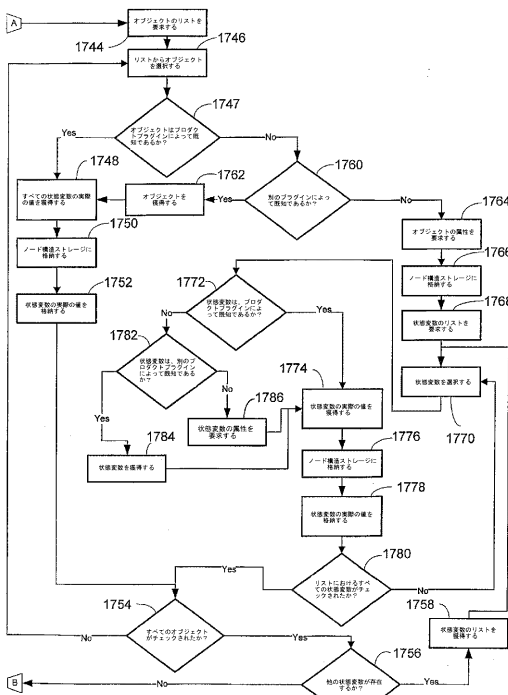


FIG. 18

【図19】

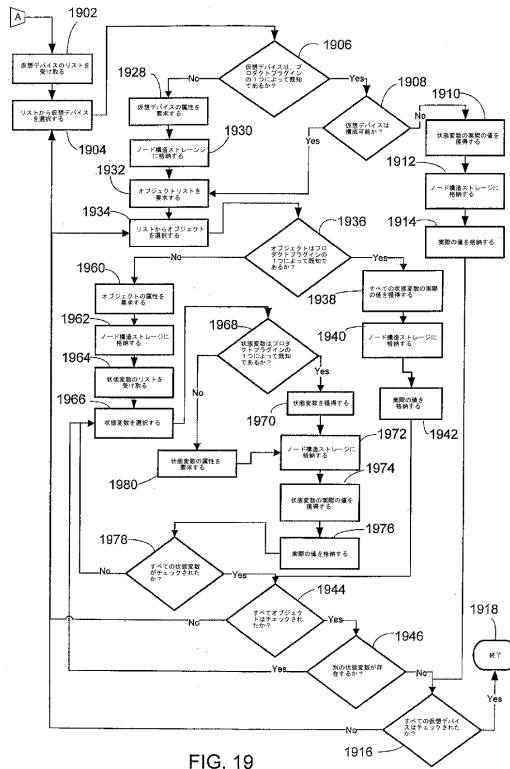


FIG. 19

【図20】

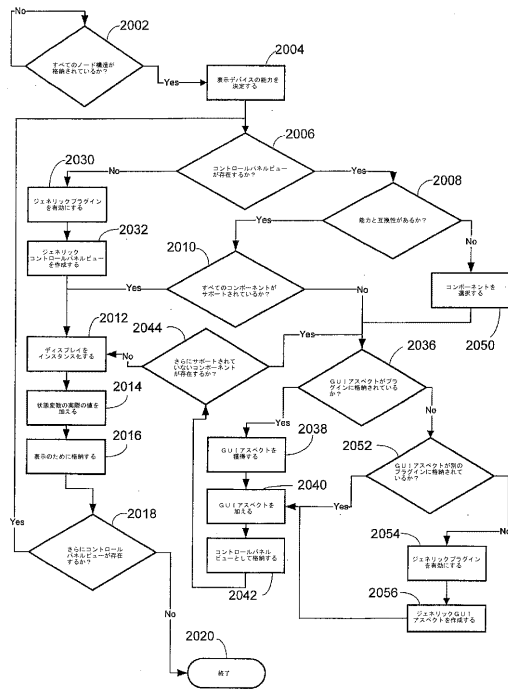


FIG. 20

【図21】

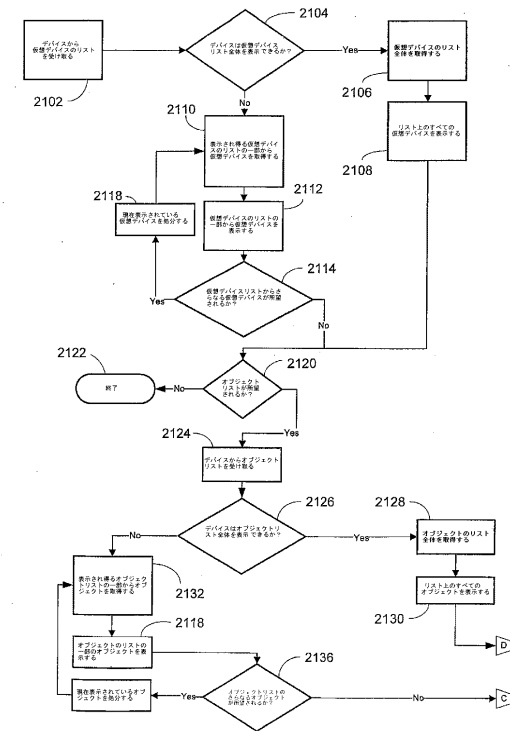


FIG. 21

【図 22】

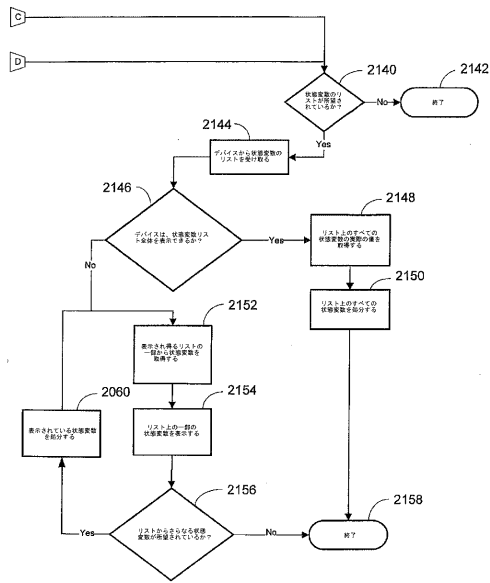


FIG. 22

【図 23】

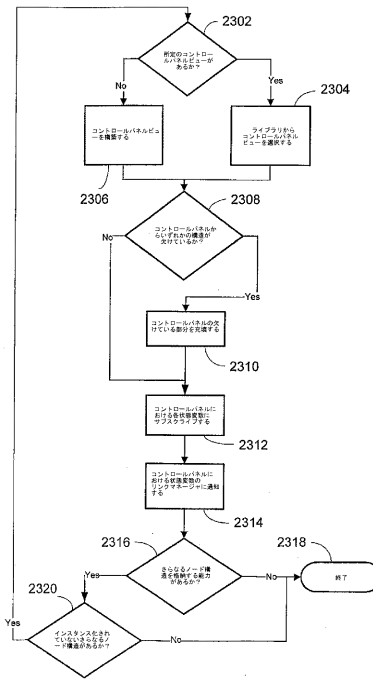


FIG. 23

【図 24】

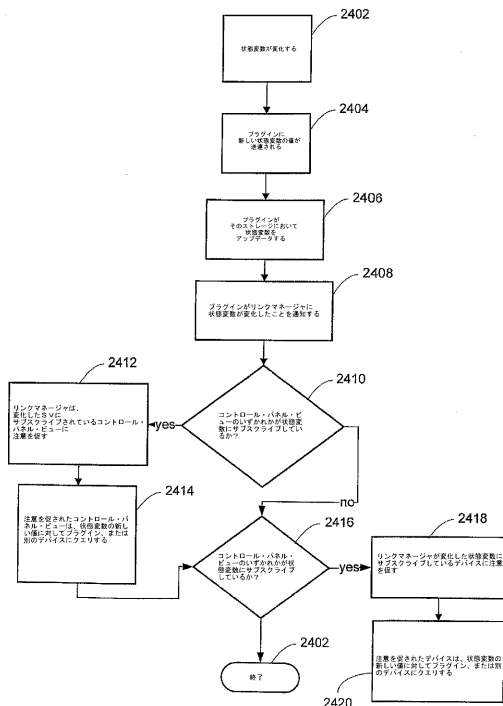


FIG. 24

【図 25】

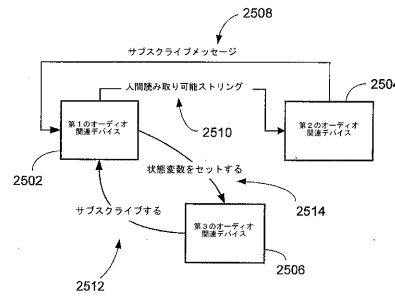


FIG. 25

【 図 26 】

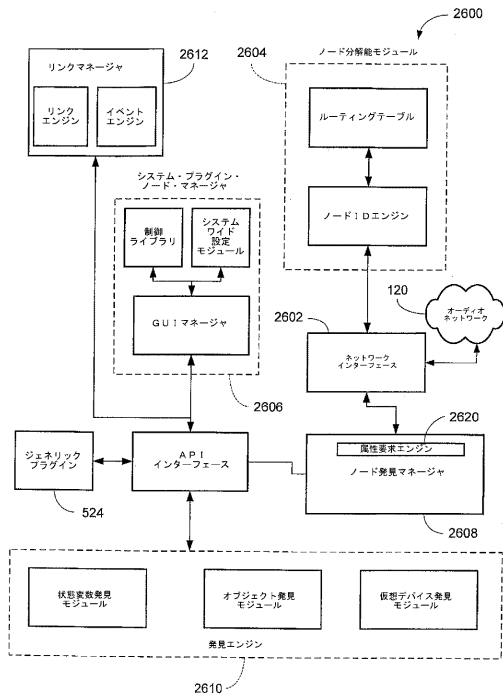


FIG. 26

【 図 4 】

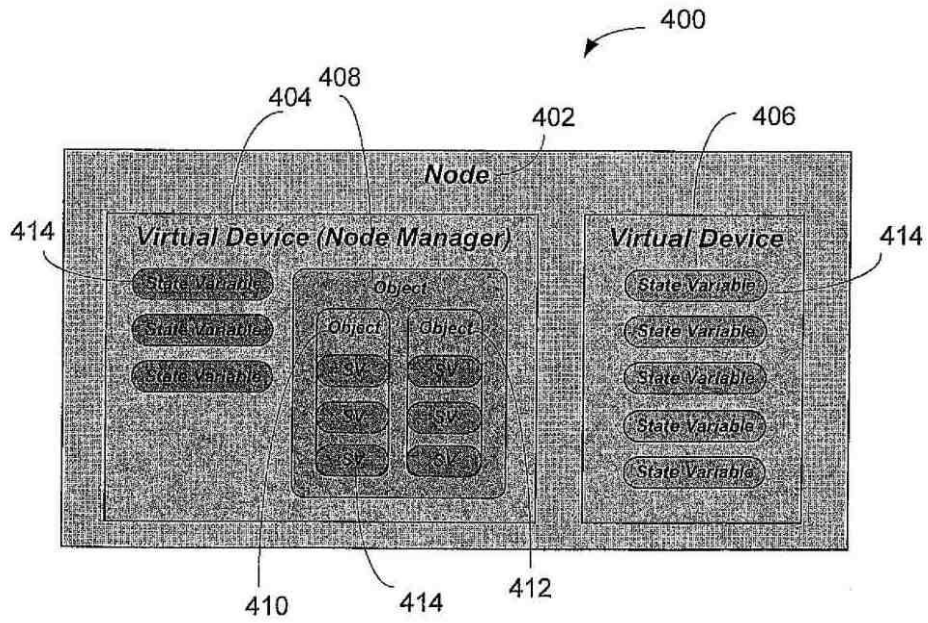


FIG. 4

【 27 】

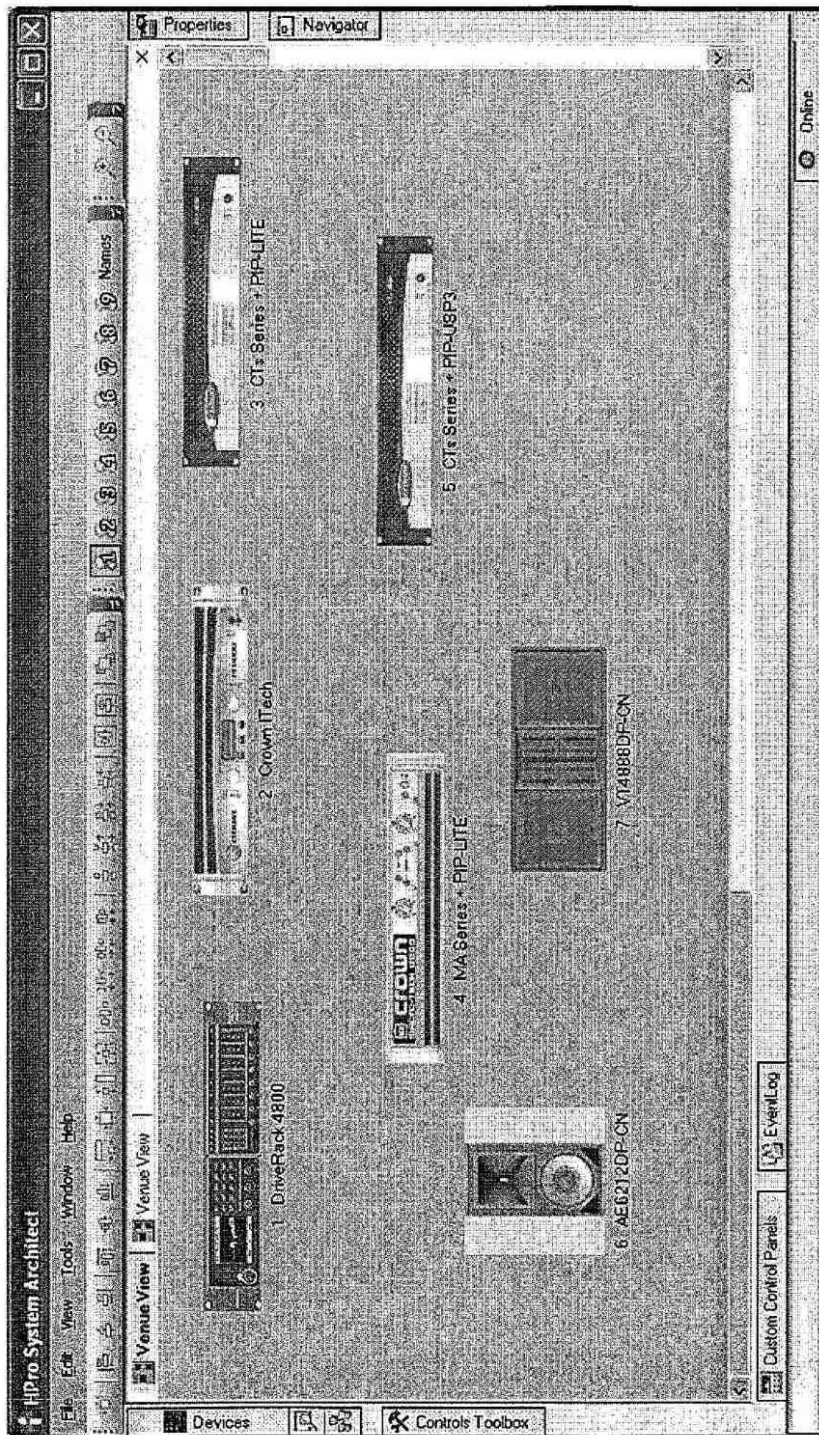


FIG. 27

【 28 】

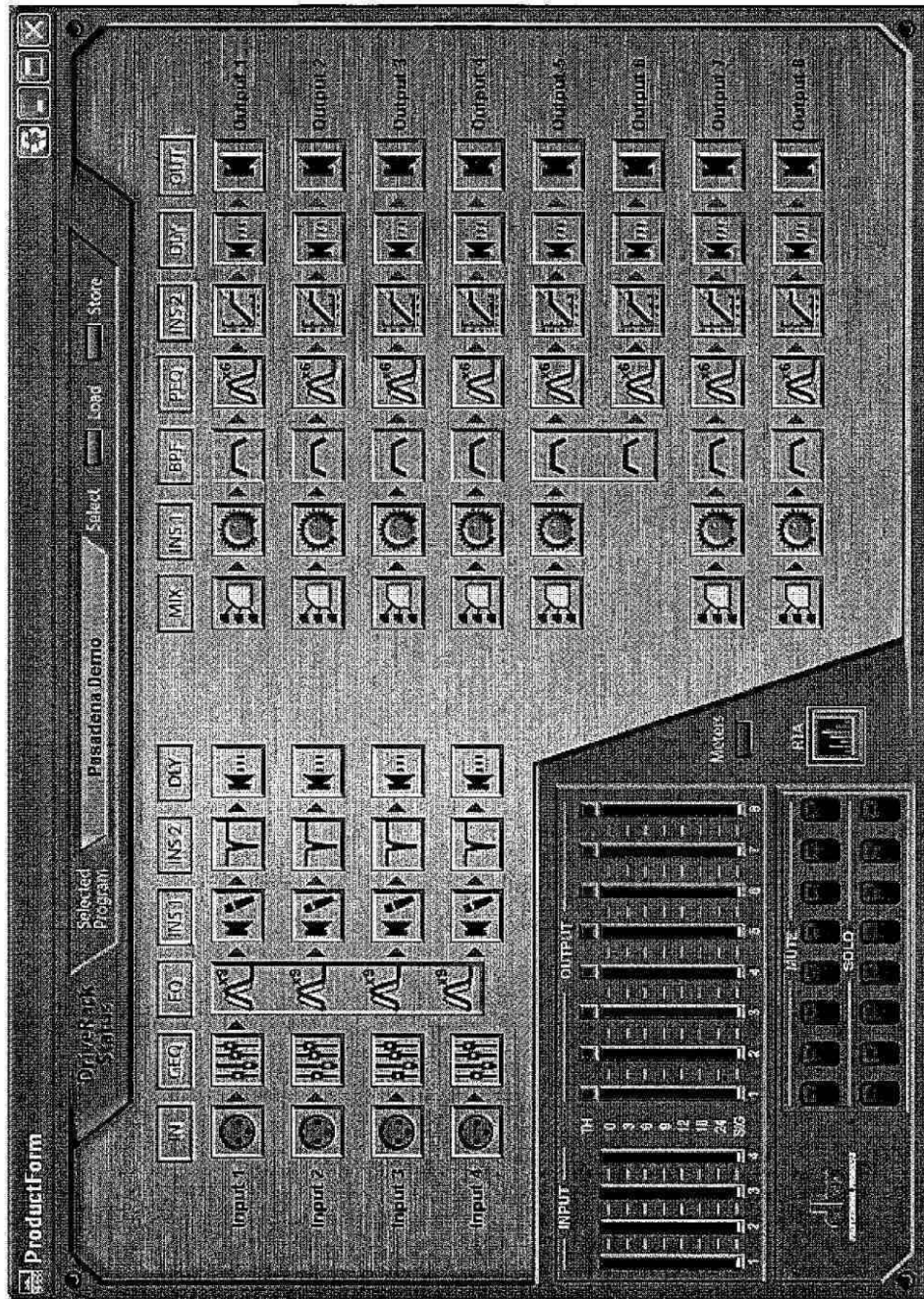


FIG. 28

【 29 】

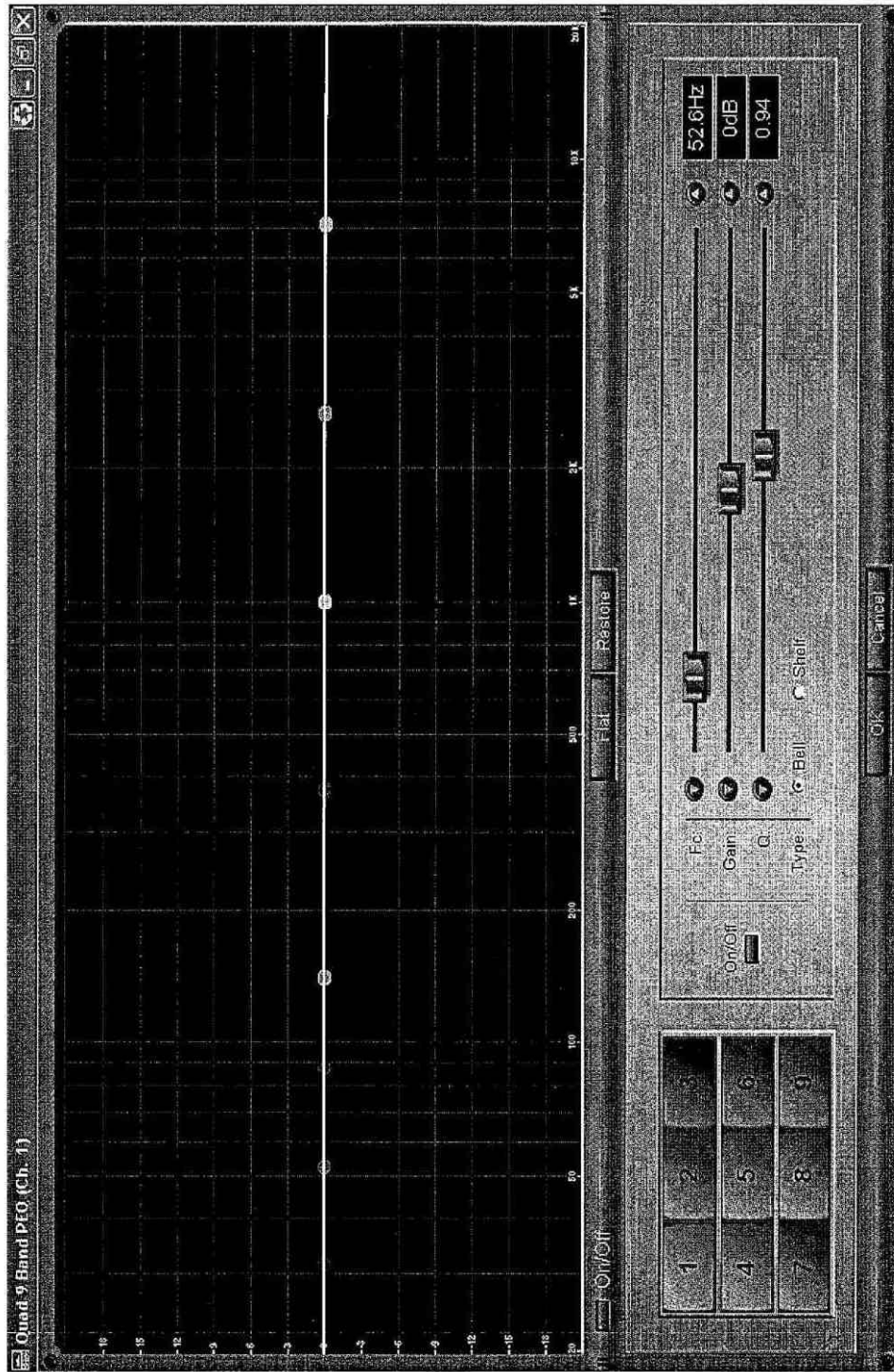


FIG. 29

## フロントページの続き

- (51)Int.Cl. F I  
**H 0 4 L 12/28 (2006.01)** H 0 4 L 12/28 2 0 0 M
- (72)発明者 リチャード アレン クレイフェルト  
 アメリカ合衆国 ユタ 84093, サンディー, イースト 8730 サウス 1530
- (72)発明者 ジョン リー  
 アメリカ合衆国 ユタ 84109, ソルト レイク シティー, サウス 3380 イースト 3127
- (72)発明者 ウィリアム エー. ブランケット  
 アメリカ合衆国 インディアナ 46530, グランジャー, ルックアウト ポインテ 52202
- (72)発明者 ティモシー ジェイ. シャトルワース  
 アメリカ合衆国 カリフォルニア 91364, ウッドランド ヒルズ, カラトラナ ドライブ 5102
- (72)発明者 アンドリュー スティーブンソン  
 イギリス国 エイチピー-2 4ジェイディー, ハートフォードシャー, ヘメル ヘンプステッド, モーベス クローズ 6
- (72)発明者 ブルース バンダー ワーフ  
 アメリカ合衆国 インディアナ 46516, エルクハート, パニオン サークル 23879
- (72)発明者 スティーブ ハッチ  
 アメリカ合衆国 ユタ 84123, ソルト レイク シティー, サドル ブラフ ドライブ 1173
- (72)発明者 マイケル ダグラス ハムリン  
 アメリカ合衆国 ユタ 84093, サンディー, レガシー パーク レーン 2859
- (72)発明者 スコット ガードナー  
 アメリカ合衆国 ユタ 84094, サンディー, モッキンバード レーン 1219

審査官 清水 正一

- (56)参考文献 特表平10-508711(JP,A)  
 特開2001-203732(JP,A)  
 特開2002-374587(JP,A)  
 特開2003-087889(JP,A)  
 特開2004-349776(JP,A)  
 特表2005-532751(JP,A)

## (58)調査した分野(Int.Cl., DB名)

H 0 4 R 3 / 0 0  
 H 0 4 R 2 7 / 0 0  
 G 0 6 F 3 / 1 4 - 3 / 1 5 3  
 G 0 6 F 1 3 / 0 0  
 G 0 6 F 1 3 / 1 4  
 H 0 4 L 1 2 / 2 8