

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7239248号

(P7239248)

(45)発行日 令和5年3月14日(2023.3.14)

(24)登録日 令和5年3月6日(2023.3.6)

(51)国際特許分類

F I

G 0 5 B 23/02 (2006.01)

G 0 5 B 23/02 3 0 1 Z

請求項の数 16 外国語出願 (全35頁)

(21)出願番号	特願2017-204058(P2017-204058)	(73)特許権者	512132022
(22)出願日	平成29年10月20日(2017.10.20)		フィッシャー - ローズマウント システ
(65)公開番号	特開2018-106687(P2018-106687		ムズ, インコーポレイテッド
	A)		アメリカ合衆国 テキサス 7 8 6 8 1 -
(43)公開日	平成30年7月5日(2018.7.5)		7 4 3 0 ラウンド ロック ウェスト ル
審査請求日	令和2年10月19日(2020.10.19)		イス ヘナ ブルバード 1 1 0 0 ビルデ
(31)優先権主張番号	15/299,679		イング 1 エマーソン プロセス マネー
(32)優先日	平成28年10月21日(2016.10.21)		ジメント
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	100113608
			弁理士 平川 明
		(74)代理人	100138357
			弁理士 矢澤 広伸
		(72)発明者	クリスチャン ジェームス クルス
			アメリカ合衆国 ミネソタ 5 5 1 1 7
			セントポール ウェスタン アベニュー
			最終頁に続く

(54)【発明の名称】 動的装置記述言語メニュー用装置及び方法

(57)【特許請求の範囲】

【請求項 1】

プロセスプラント内の装置記述言語 (D D L) に基づくホストシステム上で D D L インターフェースを構成するための動的 D D L メニュー用方法であって、前記ホストシステムが、前記プロセスプラント内で使用される複数のプロセス制御装置に接続され、前記方法が、

前記ホストシステムで、前記複数のプロセス制御装置のうちの選択された装置から装置記述識別を受信することであって、前記装置記述識別が、前記複数のプロセス制御装置のうちの前記選択された装置に対する装置記述を識別し、前記装置記述が、前記複数のプロセス制御装置のうちの前記選択された装置の 1 つ以上の特徴と関連する、変数、方法、命令、メニュー、または表示形式を含む、前記複数のプロセス制御装置のうちの前記選択された装置に関するデータ及び動作工程を含む、受信することと、

前記装置記述識別によって識別された前記装置記述内に記載された前記複数のプロセス制御装置のうちの前記選択された装置に関するデータ及び動作工程を含むように、前記受信された装置記述識別によって識別された前記装置記述により前記ホストシステムを更新することと、

前記装置記述から前記ホストシステムに D D L メニュー構造体をエクスポートすることであって、前記 D D L メニュー構造体が、構成インターフェースを介してユーザ選択可能な要素として前記ホストシステムによって提供される、エクスポートすることと、

前記ホストシステムによって、D D L グラフィカルユーザインターフェースに前記エク

10

20

スポートされた DDL メニュー構造体のうちの前記選択された 1 つ以上を追加する第 1 の入力に回答して、前記エクスポートされた DDL メニュー構造体のうちの前記 1 つ以上の選択された構造体を前記 DDL グラフィカルユーザインターフェースに追加することと、を含む、方法。

【請求項 2】

前記エクスポートされた DDL メニュー構造体のうちの選択された構造体の前記 DDL グラフィカルユーザインターフェースへの追加が、

前記エクスポートされた DDL メニュー構造体のうちの選択された構造体を、前記 DDL グラフィカルユーザインターフェース内の前記エクスポートされた DDL メニュー構造体のうちの前記選択された構造体のグラフィカルな表現にマッピングすることを含む、請求項 1 に記載の方法。

10

【請求項 3】

前記エクスポートされた DDL メニュー構造体のうちの選択された構造体の DDL メニュー への追加が、

前記エクスポートされた DDL メニュー構造体のうちの選択された構造体を、前記 DDL グラフィカルユーザインターフェース内の前記エクスポートされた DDL メニュー構造体のうちの前記選択された構造体の値にマッピングすることを含む、請求項 1 または 2 に記載の方法。

【請求項 4】

前記エクスポートされた DDL メニュー構造体のうちの 1 つ以上の第 1 の構造体が、前記エクスポートされた DDL メニュー構造体のうちの第 2 の構造体を条件とし、前記方法が、

20

表示装置を介して、前記エクスポートされた DDL メニュー構造体を提示することをさらに含み、

前記エクスポートされた DDL メニュー構造体のうちの第 2 の選択された構造体の前記 DDL グラフィカルユーザインターフェースへの追加が、前記表示装置にその後の選択に対して前記エクスポートされた DDL メニュー構造体のうちの前記 1 つ以上の第 1 の構造体のみを提示させる、請求項 1 から 3 のいずれか 1 項に記載の方法。

【請求項 5】

前記 DDL グラフィカルユーザインターフェースに追加された、前記エクスポートされた DDL メニュー構造体のうちの前記選択された構造体を、前記装置記述とは別個の前記ホストシステム上に DDL ファイルデータ構造として格納することをさらに含む、請求項 1 から 4 のいずれか 1 項に記載の方法。

30

【請求項 6】

プロセスプラント内のホストシステム上で実行するように適合された装置記述言語 (DDL) に基づくホストアプリケーションを有する動的 DDL メニュー及び DDL グラフィカルユーザインターフェース構成システムであって、前記ホストシステムが、前記プロセスプラント内で使用される複数のプロセス制御装置に接続され、前記 DDL グラフィカルユーザインターフェース構成システムが、

プロセッサと、

40

表示ユニットと、

前記プロセッサに動作可能に連結され、DDL メニュー構造体を格納するように適合されたデータベースであって、前記 DDL メニュー構造体が、装置記述からエクスポートされ、前記装置記述が、プロセス制御装置の 1 つ以上の特徴と関連する、変数、方法、命令、メニュー、または表示形式を含む、プロセス制御装置に関するデータ及び動作工程を含む、データベースと、

前記データベース内に格納された前記 DDL メニュー構造体に対する前記表示ユニット上に表示を作成するために、コンピュータ読み取り可能な装置上に格納され、前記プロセッサ上で実行するように適合された表示アプリケーションであって、前記表示が、前記 DDL メニュー構造体を提示するメニュー構造体テンプレート、及び DDL グラフィカルユ

50

ーザインターフェースを提示するインターフェース構成テンプレートを含み、前記表示アプリケーションが、前記メニュー構造体テンプレート内の前記DDLメニュー構造体の異なる構造体を選択する第1の入力が、前記DDLグラフィカルユーザインターフェースに追加するためのDDLメニュー構造体を特定することを可能にし、かつ前記DDLグラフィカルユーザインターフェースを構成するための前記インターフェース構成テンプレート内の前記選択されたDDLメニュー構造体と関連するグラフィカルな要素を提示することを可能にするように、前記プロセッサ上で実行されるように適合され、前記DDLグラフィカルユーザインターフェースが、前記ホストシステムによって維持される、表示アプリケーションと、を含む、装置記述言語（DDL）グラフィカルユーザインターフェース構成システム。

10

【請求項7】

前記メニュー構造体テンプレートが、前記DDLメニュー構造体の異なるグループを特定する複数のフォルダを有するナビゲーションツリーを含み、前記表示アプリケーションが、前記ナビゲーションツリー内の前記フォルダの異なるフォルダを選択する第2の入力が、表示されるDDLメニュー構造体のグループを特定することを可能にし、かつ前記メニュー構造体テンプレート内の選択されたフォルダと関連する前記DDLメニュー構造体を提示することを可能にするように、前記プロセッサ上で実行するように適合される、請求項6に記載のDDLグラフィカルユーザインターフェース構成システム。

【請求項8】

1つ以上の第1のDDLメニュー構造体が、第2のDDLメニュー構造体に依存し、前記表示アプリケーションが、前記第2のDDLメニュー構造体を選択する入力に応答して、選択のために前記メニュー構造体テンプレート内の前記1つ以上の第1のDDLメニュー構造体のみを提示するように前記プロセッサ上で実行するように適合され、かつ前記第1のDDLメニュー構造体のみを選択する第2の入力が、前記DDLグラフィカルユーザインターフェースに追加することを可能にするように適合される、請求項6または7に記載のDDLグラフィカルユーザインターフェース構成システム。

20

【請求項9】

前記DDLグラフィカルユーザインターフェースが、DDLファイルデータ構造として前記ホストシステムによって維持される、請求項6から8のいずれか1項に記載のDDLグラフィカルユーザインターフェース構成システム。

30

【請求項10】

前記表示アプリケーションが、第2の入力が前記プロセス制御装置の異なる装置を選択することを可能にし、かつ前記メニュー構造体テンプレート内の前記プロセス制御装置の前記選択された装置と関連する前記DDLメニュー構造体を提示することを可能にするように、前記プロセッサ上で実行するように適合される、請求項6から9のいずれか1項に記載のDDLグラフィカルユーザインターフェース構成システム。

【請求項11】

複数のプロセス制御装置の選択された装置から装置記述識別を受信するために、コンピュータ読み取り可能な装置上に格納され、前記プロセッサ上で実行するように適合されたDDLグラフィカルユーザインターフェース構成ルーチンをさらに含み、前記装置記述識別が、前記複数のプロセス制御装置の前記選択された装置に対する前記装置記述を識別し、前記DDLグラフィカルユーザインターフェース構成ルーチンが、前記受信された装置記述識別によって識別された前記装置記述により前記ホストシステムを更新し、かつ前記装置記述から前記ホストシステムにDDLメニュー構造体をエクスポートするように、前記プロセッサ上で実行するようにさらに適合される、請求項6から10のいずれか1項に記載のDDLグラフィカルユーザインターフェース構成システム。

40

【請求項12】

プロセスプラント内の装置記述言語（DDL）に基づくホストシステム上のDDLメニューを構成するための動的DDLメニュー用の方法であって、前記ホストシステムが、前記プロセスプラント内で使用される複数のプロセス制御装置に接続され、前記方法が、

50

前記複数のプロセス制御装置の選択された装置に対する複数のＤＤＬメニュー構造体のグラフィカルな表現を含み、前記ＤＤＬメニュー構造体の選択された構造体を要求する第１の入力を可能にする、メニュー構造体テンプレートを提示することであって、前記ＤＤＬメニュー構造体が、前記複数のプロセス制御装置の前記選択された装置に対する装置記述から前記ホストシステムにエクスポートされ、前記装置記述が、前記複数のプロセス制御装置の前記選択された装置の１つ以上の特徴と関連する、変数、方法、命令、メニュー、または表示形式を含む、前記複数のプロセス制御装置の前記選択された装置に関するデータ及び動作工程を含む、提示することと、

前記ＤＤＬメニュー構造体のうちの１つを選択する第１の入力に応答して、ＤＤＬグラフィカルユーザインターフェースのグラフィカルな表現及び前記ＤＤＬメニュー構造体の前記選択された構造体の前記グラフィカルな表現を含み、かつ前記ＤＤＬグラフィカルユーザインターフェース内の前記ＤＤＬメニュー構造体の前記選択された構造体を構成する第２の入力を可能にする、インターフェース構成テンプレートを提示することと、

前記ＤＤＬグラフィカルユーザインターフェース内の前記ＤＤＬメニュー構造体の前記選択された構造体を構成する第２の入力に応答して、前記ＤＤＬメニュー構造体の選択された構造体を要求する前記第１の入力を可能にし、かつ前記装置記述とは別個の前記ホストシステム上のＤＤＬファイルデータ構造として前記構成されたＤＤＬメニュー構造体を含む、前記ＤＤＬグラフィカルユーザインターフェースを格納する第３の入力を可能にすることと、を含む、方法。

【請求項１３】

前記複数のプロセス制御装置の前記選択された装置に対するＤＤＬメニュースタイルの表示を含み、かつ前記ＤＤＬメニュースタイルの選択されたスタイルを要求する第４の入力を可能にする、メニュースタイルテンプレートを提示することと、

ＤＤＬメニュースタイルを選択する第４の入力に応答して、前記ＤＤＬグラフィカルユーザインターフェースのグラフィカルな表現及び前記ＤＤＬメニュースタイルの前記選択されたスタイルの前記グラフィカルな表現を含む、前記インターフェース構成テンプレートを提示することと、をさらに含む、請求項１２に記載の方法。

【請求項１４】

前記複数のプロセス制御装置の選択された装置を要求する第４の入力を可能にすることと、

前記複数のプロセス制御装置のうちの１つを選択する第４の入力に応答して、前記メニュー構造体テンプレートを提示することと、をさらに含む、請求項１２または１３に記載の方法。

【請求項１５】

前記インターフェース構成テンプレート内に提示されたＤＤＬメニュー構造体のグラフィカルな表現を選択する第４の入力に応答して、前記ＤＤＬグラフィカルユーザインターフェースから前記選択されたＤＤＬメニュー構造体を可能にし、削除することと、前記削除されたＤＤＬメニュー構造体の前記グラフィカルな表現なしに、前記ＤＤＬグラフィカルユーザインターフェースの前記グラフィカルな表現を含む、前記インターフェース構成テンプレートを提示することと、をさらに含む、請求項１２から１４のいずれか１項に記載の方法。

【請求項１６】

前記インターフェース構成テンプレート内に提示されたＤＤＬメニュー構造体のグラフィカルな表現を選択する第４の入力に応答して、前記ＤＤＬメニュー構造体の前記選択されたグラフィカルな表現の前記値を変更する第５の入力を可能にすることと、

前記ＤＤＬグラフィカルユーザインターフェース内の前記ＤＤＬメニュー構造体の前記値を変更する第５の入力に応答して、前記ＤＤＬグラフィカルユーザインターフェース内の前記ＤＤＬメニュー構造体の前記値を変更することと、をさらに含む、請求項１２または１５に記載の方法。

【発明の詳細な説明】

【技術分野】

【 0 0 0 1 】

本開示は、一般的に、プロセスプラント内のプロセス制御システムに関し、より具体的には、装置記述言語メニューの動的な作成及び修正に関する。

【 背景技術 】

【 0 0 0 2 】

プロセス制御システムは、製品が製造されるか、またはプロセスが制御される工場及び／またはプラント（例えば、薬品の製造、電力プラント制御等）で広く使用されている。プロセス制御システムはまた、例えば、油及びガス掘削ならびに処理プロセス等の天然資源の回収にも使用されている。実際に、1つ以上のプロセス制御システムの適用を通して、事実上任意の製造プロセス、資源回収プロセス等を自動化することができる。プロセス制御システムは、最終的に、農業においても同様に、より広く使用されるだろう。

10

【 0 0 0 3 】

薬品、石油、または他のプロセスで使用されるものと同様のプロセス制御システムは、典型的に、アナログ、デジタル、またはアナログ／デジタルバスの組み合わせを介して、少なくとも1つのホストまたはオペレータワークステーションに、かつフィールド装置等の1つ以上のプロセス制御及び計測装置に通信可能に連結された、1つ以上の集中または分散プロセス制御装置を含む。例えば、バルブ、バルブポジショナ、スイッチ、送信器、及びセンサ（例えば、温度、圧力、及び流量センサ）であり得る、フィールド装置は、開口及び閉鎖バルブ及び測定プロセスパラメータ等のプロセス内で機能を実行する。プロセス制御装置は、フィールド装置及び／またはフィールド装置に関する他の情報によって作成されるか、またはこれと関連するプロセス測定値またはプロセス変数を示す信号を受信し、この情報を使用して制御ルーチンを実行し、次に、バスのうちの1つ以上にわたってフィールド装置に伝送される制御信号を作成してプロセスの動作を制御する。フィールド装置及び制御装置からの情報は、典型的に、オペレータワークステーションによって実行された1つ以上のアプリケーションに利用可能とされ、プロセスの現在の状態を視認すること、プロセスの動作を修正すること等、プロセスに関する所望の機能をオペレータが実行することを可能にする。

20

【 0 0 0 4 】

プロセスプラント内の様々な装置は、制御ループ等の論理プロセスを作成するように、物理及び／または論理グループ内で相互接続される場合がある。同様に、制御ループは、サブユニットを作成するように、他の制御ループ及び／または装置と相互接続される場合がある。サブユニットは、ユニットを作成するように、他のサブユニットと相互接続される場合があり、次に、領域を作成するように、他のユニットと相互接続され得る。プロセスプラントは、一般的に、相互接続された領域を含み、事業者は、一般的に、相互接続され得るプロセスプラントを含む。結果として、プロセスプラントは、相互接続されたアセットを有する多数のレベルの階層を含み、事業者は、相互接続されたプロセスプラントを含み得る。言い換えれば、プロセスプラントに関連するアセットまたはプロセスプラント自体が、より高いレベルでアセットを形成するように、共にグループ分けされ得る。

30

【 0 0 0 5 】

プロセス制御システムが実行される様式は、長年にわたって発展してきた。より古い世代のプロセス制御システムは、典型的に、専用の集中ハードウェア及び配線接続を用いて実行されていた。

40

【 0 0 0 6 】

しかしながら、現代のプロセス制御システムは、典型的に、ワークステーションの高度に分散されたネットワーク、インテリジェント制御装置、スマートフィールド装置等を用いて実行され、そのうちのいくつか、またはすべてが、全体的なプロセス制御戦略または方式の一部を実行し得る。具体的には、大部分の現代のプロセス制御システムは、スマートフィールド装置、及び互いに、かつ／または1つ以上のデジタルデータバスを介した1つ以上のプロセス制御装置に通信可能に連結される、他のプロセス制御構成要素を含む。スマートフィールド装置に加えて、現代のプロセス制御システムはまた、共有されたデ

50

デジタルデータバス等に対向する制御装置に典型的に直接連結される、例えば、4～20ミリアンペア（mA）の装置、0～10ボルトの直流（VDC）装置等のアナログフィールド装置を含み得る。

【0007】

典型的な産業またはプロセスプラントでは、分散制御システム（DCS）を用いて、プラントで実行される産業プロセスの多くを制御している。プラントは、ユーザ入力／出力（I/O）、ディスクI/O、ならびに集中制御室と通信可能に接続された、1つ以上のプロセス制御装置及びプロセスI/Oサブシステムを有するコンピュータ技術において知られる他の周辺機器を含むコンピュータシステムを有する集中制御室を有し得る。さらに、1つ以上のフィールド装置は、典型的に、プラント内で制御及び測定を実行するために、I/Oサブシステム及びプロセス制御装置に接続される。プロセスI/Oサブシステムは、プラント全体にわたって様々なフィールド装置に接続された複数のI/Oポートを含み得るが、フィールド装置は、様々な種類の分析機器、シリコン圧力センサ、容量圧力センサ、抵抗温度検出器、熱電対、歪みゲージ、リミットスイッチ、オン／オフスイッチ、流れ送信器、圧力送信器、静電容量レベルスイッチ、重量計、トランスデューサ、バルブポジショナ、バルブ制御装置、アクチュエータ、ソレノイド、表示灯、またはプロセスプラントにおいて典型的に使用される任意の他の装置を含んでもよい。

10

【0008】

本明細書で使用される場合、「フィールド装置」という用語は、これらの装置、ならびに制御システムにおける機能を実行する任意の他の装置を包含する。任意の事例において、フィールド装置は、例えば、入力装置（例えば、温度、圧力、流量等のプロセス制御パラメータを示す状態信号を提供するセンサ等の装置）、ならびに制御装置及び／または他のフィールド装置から受信された命令に応答して動作を実行する制御オペレータまたはアクチュエータを含んでもよい。

20

【0009】

従来より、アナログフィールド装置は、2線ツイストペア電流ループによって制御装置に接続され、各装置は、単一の2線ツイストペアによって制御装置に接続される。アナログフィールド装置は、特定の範囲内で電気信号にตอบสนองするか、または電気信号を伝送することができる。典型的な構成では、2線の対間のおよそ20～25ボルトの電圧差、及びループを通して走行する4～20mAの電流を有するのが一般的である。信号を制御室に伝送するアナログフィールド装置は、電流ループを通して走行する電流を変調し、電流は、検知されたプロセス変数に比例する。

30

【0010】

制御室の制御下で動作を実行するアナログフィールド装置は、電流がプロセスI/OシステムのI/Oポートによって変調され、次に、制御装置によって制御される、ループを通る電流の大きさによって制御される。アクティブ電子機器を有する従来型の2線アナログ装置はまた、ループから最大40ミリワットの電力を受信することができる。より多くの電力を必要とするアナログフィールド装置は、典型的に、4線を用いて制御装置に接続され、その線のうちの2つが装置に電力を送達する。かかる装置は、4線装置として当該技術分野で知られており、典型的に2線装置のように電力制限されない。

40

【0011】

離散フィールド装置は、二値信号を伝送するか、または二値信号にตอบสนองすることができる。典型的に、離散フィールド装置は、24ボルトの信号（ACまたはDCのいずれか）、110または240ボルトのAC信号、または5ボルトのDC信号により動作する。当然ながら、離散装置は、特定の制御環境によって要求される任意の電氣的仕様に従って動作するように設計され得る。離散入力フィールド装置は、単に、制御装置への接続を作成するか、または解除するかのいずれかのスイッチであり、離散出力フィールド装置は、制御装置からの信号の存在または不在に基づいて動作を行うだろう。

【0012】

歴史的に見て、大部分の従来型のフィールド装置は、フィールド装置によって実行され

50

る主な機能に直接関連する単一入力または単一出力のいずれかを有していた。例えば、従来型のアナログ抵抗温度センサによって実行される唯一の機能は、2線ツイストペアを通して流れる電流を変調させることによって温度を伝送することであり、従来型のアナログバルブポジションによって実行される唯一の機能は、2線ツイストペアを通して流れる電流の大きさに基づいて、完全な開口位置と完全な閉鎖位置との間のどこかにバルブを位置付けることである。

【0013】

最近では、アナログ信号を伝送するために使用される電流ループにデジタルデータを重ね合わせる、ハイブリッドシステムの一部であるフィールド装置が利用可能になっている。かかる1つのハイブリッドシステムは、制御技術において、ハイウェイアドレス可能遠隔トランスデューサ(HART)プロトコルとして知られている。HARTシステムは、アナログ制御信号を伝送するか、または検知されたプロセス変数を受信するための電流ループにおける(従来型のシステムにおけるような)電流の大きさを使用するが、また、電流ループ信号にデジタルキャリア信号を重ね合わせる。HARTプロトコルは、ベル202周波数偏移キーイング(FSK)基準を使用して、4~20mAのアナログ信号の上部に低レベルのデジタル信号を重ね合わせる。これは、双方向フィールド通信が起こり、単なる通常のプロセス変数を超えた追加の情報がスマートフィールド機器にノードから通信されることを可能にする。HARTプロトコルは、4~20mAの信号を遮断することなく1200bpsで通信し、ホストアプリケーション(マスター)がフィールド装置から毎秒2つ以上のデジタル更新を得ることを可能にする。デジタルFSK信号は、位相連続であり、4~20mAの信号への介入がない。

【0014】

FSK信号は、比較的遅いため、毎秒およそ2~3つの更新速度で二次的なプロセス変数または他のパラメータの更新を提供することができる。一般的に、デジタルキャリア信号は、二次的な診断情報を伝送するために使用され、フィールド装置の主な制御機能を実現するためには使用されない。デジタルキャリア信号を介して提供される情報の例には、二次的なプロセス変数、診断情報(センサ診断、装置診断、配線診断、及びプロセス診断を含む)、動作温度、センサ温度、較正情報、装置IDナンバー、構成材料、構成、またはプログラミング情報等が含まれる。したがって、単一ハイブリッドフィールド装置は、様々な入力及び出力変数を有してもよく、様々な機能を実行し得る。

【0015】

最近では、より新しい制御プロトコルが、アメリカ計測学会(ISA)によって定義されている。新しいプロトコルは、一般的に、フィールドバスと称され、Standards and Practice Subcommittee 50に対する頭字語である、SP50と特に称されている。フィールドバスプロトコルは、2つのサブプロトコルを定義する。H1フィールドバスネットワークは、最大毎秒31.25キロバイトの速度でデータを伝送し、ネットワークに連結されたフィールド装置に電力を提供する。H2フィールドバスネットワークは、最大毎秒2.5メガバイトの速度でデータを伝送し、ネットワークに接続されたフィールド装置には電力を提供せず、冗長伝送メディアが提供される。フィールドバスは、非専売のオープン規格であり、今や産業界に流布しており、このように、多くの種類のフィールドバス装置が開発され、プロセスプラントにおいて使用されている。フィールド装置は、HART及び4~20mAの装置等、他の種類のフィールド装置に加えて使用されるため、別個の支持及びI/O通信構造は、これらの異なる種類の装置の各々に関連している。

【0016】

典型的に、すべてデジタルな性質である、より新しいスマートフィールド装置は、メンテナンスモードを有し、より古い制御システムからアクセス不能であるか、または互換性がない機能を改善してきた。分散制御システムのすべての構成要素が同じ規格(フィールドバス規格等)に従っている場合であっても、ある製造業者の制御機器が、別の製造業者のフィールド装置によって提供された二次的な機能または二次的な情報にアクセスできな

10

20

30

40

50

い場合がある。

【 0 0 1 7 】

典型的に、これらの基盤によって定義された通信プロトコルは、各装置が自体を識別する方法を特定する基準を含み、装置記述（DD）として知られているものの使用を通してプロセス制御システムと通信し、DDが、プロトコルのアプリケーション層及び装置と通信する必要がある様々なユーザインターフェース定義を定義する。DDは、国際電気標準会議の規格（例えば、IEC 61804）として構築された、よく知られ、よく支持されている装置記述言語（DDL）（電子装置記述言語（EDDL）としても知られている）で記載されている。各装置の種類は、典型的に、変数データ（パラメータ）、通信（アドレス指定情報）、方法、命令／動作（例えば、較正）、及び装置の様々な特徴と関連するグラフィカルユーザインターフェース（例えば、メニュー及び表示形式）を含む、フィールド装置に関するデータ及び動作工程の形式的記述である、それ自体のDDを有する。装置のすべてのアクセス可能な変数についての情報が、一般的に、装置記述に含まれ、それによって装置の互換性及び装置との可能な通信を定義する。かかる変数は、例えば、プロセス測定値、任意の導出値、ならびに範囲、センサの種類、線形化の選択、構成材料、製造業者、リビジョン番号等の装置のすべての内部パラメータを含む。DDLは、ホストシステムまたはホストアプリケーションに装置に関するDD内の情報を通信、解読、及び表示する方法を指示する。

10

【 0 0 1 8 】

様々な装置に対するDDは、典型的に、多数の異なる様式で使用されている。例えば、プロセスアプリケーションまたはホストアプリケーションがプロセスプラント内で実行されたとき、プロセスアプリケーションを維持する責任のあるメンテナンス要員は、様々な装置の様々なパラメータについての支援情報を得る必要があり得る。同様に、プロセスアプリケーションを記載するシステム設計者は、装置についてのさらなる情報を得るためにDDを使用し得る。装置製造業者は、一般的に、これらのDDを様々なプロセス制御システムコンピュータまたは様々なプロセスプラント関連のアプリケーションへと容易に複製することができるように、コンピュータ読み取り可能な媒体上にDDを提供する。具体的には、装置製造業者は、典型的に、DDLにおいて、装置に関連するパラメータ、装置と通信する方法、装置に対する制限等を定義する、製造業者が製造する各装置に対するDDを提供する。

20

【 0 0 1 9 】

DDLに基づくホストは、DD開発者（例えば、装置製造業者、プロトコルの基盤）がそれを見るユーザにとって重要であると考える装置と関連する、装置の種類、及び重要なパラメータ、制限等を判定するために、DDL内の装置に対するDDを読み取り及び解釈することができる。このように、DDLの構造（変数、方法、命令、メニュー、ならびに画像、グラフ、グリッド、波形、及びチャート等の表示形式）は、DDLに基づくホストまたはホストアプリケーションによって解釈されて、ユーザに表示される。DDLに基づくホストは、装置に対するグラフィック要素の固有の特性またはパラメータとして装置と関連するパラメータ、制限等を定義することができる。DDLに基づくホストはまた、装置に対する視覚化を選択するか、または定義するためのパラメータを有してもよく、DDからの情報に基づくか、または装置に対するDDによって定義された装置の種類に対して格納されたテンプレートに基づいて、装置に対する基本的な動作及び動画を提供するために使用される1つ以上の一般的なスクリプトを選択してもよい。リソースファイルはまた、DDL情報を表示に変換するために使用され得る。しかしながら、リソースファイルは、典型的に、DD開発者によって設計されており、その事例では、表示は、装置に対する視覚化及び動画を定義するようにいくらか構成可能ではあるが、一般的に、情報の表示に関していえば静的なものであり、ユーザは、開発者のリソースファイルによって特定された情報を視認することを強いられる。このように、ユーザは、ユーザが視認する情報及び／または情報が視認される様式の制御下にはない。DDによって特定される装置に対するグラフィカルユーザインターフェースの観点から、ユーザは、ユーザにとって重要だと思

30

40

50

われる情報を検索するために、複数のメニューを通して繰り返すことが必要とされ得る。

【発明の概要】

【0020】

装置記述言語（DDL）メニュー構成システム及び方法は、装置記述の開発者によって提供されたリソースファイルに基づく情報を視認するのではなく、ユーザにとって重要であると思われる情報を視認するために、ユーザがDDLメニューを構成及び維持することを可能にする。ユーザは、プロセス装置を選択し、メニュー要素に関連するこれらのプロセス装置を選択して、装置記述が異なる開発者（例えば、異なるプロセス装置製造業者）からのものである場合であっても、異なるプロセス装置に対する異なる装置記述からのメニュー要素を含むDDLメニューに追加することができる。メニューは、DDLホスト上で保存され、メニューについての情報を視認するための要求に応じてユーザが作動し得る。ユーザはまた、新しいメニュー要素によりメニューを再構成することもできる。

10

【0021】

本明細書に開示されているシステム及び方法は、ユーザによって選択されるプロセス制御装置に対する装置記述によりDDLホストを更新する。装置記述は、他の品目の中で特に、プロセス制御装置についての情報を視認するためのメニュー及び/または表示形式を含む。装置記述からのDDLメニュー構造体が、装置記述からエクスポートされ、ユーザがユーザにとって重要なDDLメニュー構造体を選択することができるように、構成インターフェースに表示される。次に、ユーザは、ユーザの選択に従ってDDLメニューを構成するために、DDLメニュー構造体を構成し、DDLメニューに追加し得る。DDLメニューは、ユーザが観察する関心がある様々なDDLメニュー構造体にマッピングされる。DDLメニューは、DDLホストによって維持され、その結果、ユーザは、要求に応じてDDLメニューを作動させるか、または停止させ、新しいDDLメニュー構造体を追加するか、DDLメニュー内のDDLメニュー構造体を修正するか、またはDDLメニューからDDLメニュー構造体を削除することによって、DDLメニューを再構成し得る。

20

【図面の簡単な説明】

【0022】

【図1】本開示に従う、分散プロセス制御システムの組み合わされたブロック及び概略図である。

【図2】本開示に従う、インターネットを介した様々なデータベースによりネットワーク化された分散プロセス制御システムのDDLに基づくホストシステムのブロック図である。

30

【図3】本開示に従う、DDLに基づくホストによって維持された新しいDDLグラフィカルユーザインターフェースを構成するための例示的なグラフィカルユーザインターフェース構成ルーチンのフローチャートである。

【図4】本開示に従う、DDLグラフィカルユーザインターフェースを作成するための例示的なグラフィカルユーザインターフェースである。

【図5】本開示に従う、DDLグラフィカルユーザインターフェースを作成するための装置を選択するための例示的なグラフィカルユーザインターフェースである。

【図6】本開示に従う、選択された装置の装置記述を検索するための例示的な装置記述検索ルーチンのフローチャートである。

40

【図7】本開示に従う、DDLグラフィカルユーザインターフェースを作成及び構成する例示的なグラフィカルユーザインターフェースである。

【図8】本開示に従う、DDLに基づくホストシステムによって維持されたユーザ定義された装置記述メニューを作成する際のプロセス中の相互作用を示す例示的なシーケンス図である。

【図9】ユーザ定義された装置記述メニューを示す例示的なグラフィカルユーザインターフェースである。

【図10】本開示に従う、DDLグラフィカルユーザインターフェースを作動させるための例示的なメニュー作動ルーチンのフローチャートである。

【図11】本開示に従う、既存のDDLグラフィカルユーザインターフェースを編集する

50

ための例示的な装置記述メニュー編集ルーチンのフローチャートである。

【図 1 2】本開示に従う、新しい装置に対する D D L メニュー構造体を既存の D D L グラフィカルユーザインターフェースに追加するための例示的な装置追加ルーチンのフローチャートである。

【図 1 3】本開示に従う、D D L グラフィカルユーザインターフェース内のメニュー構造体を修正するための例示的な D D L メニュー修正ルーチンのフローチャートである。

【図 1 4】本開示に従う、D D L グラフィカルユーザインターフェースから D D L メニュー構造体を削除するための例示的な D D L メニュー構造体削除ルーチンのフローチャートである。

【図 1 5】本開示に従う、D D L メニュー構造体を D D L グラフィカルユーザインターフェースに追加するための例示的な D D L メニュー構造体追加ルーチンのフローチャートである。

10

【図 1 6】本開示に従う、D D L グラフィカルユーザインターフェース内の D D L メニュー構造体を修正するための例示的な D D L メニュー構造体修正ルーチンのフローチャートである。

【発明を実施するための形態】

【0 0 2 3】

次に、図 1 を参照すると、配線分散プロセス制御システム 1 0 は、1 つ以上のホストワークステーションまたはコンピュータ 1 4 (任意の種類のパersonalコンピュータまたはワークステーションであり得る) に接続された 1 つ以上のプロセス制御装置 1 2 を含む。プロセス制御装置 1 2 はまた、入力 / 出力 (I / O) 装置 2 0、2 2 のバンクに接続され、その各々が、次に、1 つ以上のフィールド装置 2 5 ~ 3 9 に接続される。例示のみの方法によって、F i s h e r - R o s e m o u n t S y s t e m s , I n c によって販売されている D e l t a V (商標) 制御装置であり得る、制御装置 1 2 は、例えば、イーサネット接続 4 0 または他の通信リンクを介してホストコンピュータ 1 4 に通信可能に接続される。同様に、制御装置 1 2 は、例えば、標準的な 4 ~ 2 0 m A の装置及び / またはフィールドバスまたは H A R T プロトコル等の任意のスマート通信プロトコルと関連する任意の所望のハードウェア及びソフトウェアを用いて、フィールド装置 2 5 ~ 3 9 に通信可能に接続される。一般的に知られているように、制御装置 1 2 は、その中に格納されるか、または別様に、それと関連するプロセス制御ルーチンを実行するか、または監視し、装置 2 5 ~ 3 9 と通信して、任意の所望の様式でプロセスを制御する。

20

30

【0 0 2 4】

フィールド装置 2 5 ~ 3 9 は、センサ、バルブ、送信器、ポジショナ等の任意の種類の装置であってもよく、バンク 2 0 及び 2 2 内の I / O カードは、H A R T、フィールドバス、プロフィバス等の任意の所望の通信または制御装置プロトコルに適合する任意の種類の I / O 装置であってもよい。図 1 に図示されている実施形態では、フィールド装置 2 5 ~ 2 7 は、アナログラインを介して I / O カード 2 2 A と通信する、標準的な 4 ~ 2 0 m A の装置である。フィールド装置 2 8 ~ 3 1 は、H A R T 互換性 I / O 装置 2 0 A に接続された H A R T 装置として図示されている。同様に、フィールド装置 3 2 ~ 3 9 は、例えば、フィールドバスプロトコル通信を用いて、デジタルバス 4 2 または 4 4 を介して I / O カード 2 0 B または 2 2 B と通信する、フィールドバスフィールド装置等のスマート装置である。当然ながら、フィールド装置 2 5 ~ 3 9 及び I / O カード 2 0 及び 2 2 のバンクは、将来開発される任意の規格またはプロトコルを含む、4 ~ 2 0 m A、H A R T、またはフィールドバスプロトコル以外の任意の他の所望の規格に適合することができるだろう。

40

【0 0 2 5】

制御装置 1 2 の各々は、機能ブロックと一般に称されるものを用いて制御戦略を実行するように構成され、各機能ブロックは、全体的な制御ルーチンの一部 (例えば、サブルーチン) であり、プロセス制御システム 1 0 内のプロセス制御ループを実行するための他の機能ブロック (リンクと呼ばれる通信を介した) と連動して動作する。機能ブロックは、

50

典型的に、プロセス制御システム 10 内の一部の物理機能を実行するために、送信器、センサ、または他のプロセスパラメータ測定装置と関連するもの等の入力機能、PID、ファジー論理等の制御を実行する制御ルーチンと関連するもの等の制御機能、またはバルブ等のある装置の動作を制御する出力機能のうちの 1 つを実行する。当然ながら、ハイブリッド及び他の種類の機能ブロックが存在する。これらの機能ブロックのグループは、モジュールと呼ばれる。機能ブロック及びモジュールは、これらの機能ブロックが標準的な 4 ~ 20 mA の装置及びいくつかの種類のスマートフィールド装置に対して使用されるか、または関連するときに典型的に当てはまる、制御装置 12 内に格納され、かつこれによって実行される場合があり、またはフィールドバス装置に当てはまる場合がある、フィールド装置自体内に格納され、かつこれによって実行される場合がある。図 1 に図示されている制御システム 10 は、機能ブロック制御戦略を用いるものとして記載されているが、制御戦略はまた、ラダー論理、一連のフローチャート等の他の慣習を用いて、かつ任意の所望の特性または非専売プログラミング言語を用いて、実行または設計されることができるだろう。

10

【0026】

またさらに、知られている様式において、ワークステーション 14 のうちの 1 つ以上が、オペレータ、構成エンジニア、メンテナンス人員等のユーザが、プラント内のプロセス制御ネットワーク 10 とインターフェースすることを可能にするためのユーザインターフェースアプリケーションを含んでもよい。具体的には、ワークステーション 14 は、プロセス制御システム 10 の進行状態に関連する情報等のプラントからの情報を得るために、フィールド装置 25 ~ 39 及びこれらのフィールド装置内のモジュール等により、制御装置 12 または I/O バンク 20、22 内のデータベース、制御モジュール、または他のルーチンと通信するように、ワークステーション 14 内のプロセッサ上で実行され得る、1 つ以上のユーザインターフェースアプリケーションを含み得る。ユーザインターフェースアプリケーションは、ワークステーション 14 のうちの 1 つ以上と関連する表示装置上に、この収集された情報を処理及び/または表示し得る。収集、処理、及び/または表示された情報は、例えば、プロセス状態情報、プラント内に作成されたアラーム及びアラート、メンテナンスデータ等であり得る。同様に、1 つ以上のアプリケーションは、プラント内の設定値または他の制御変数の変更等の制御オペレータの活動を実行するために、プラント内で実行されるモジュールの作成または構成等の構成活動を実行するために、ワークステーション 14 内に格納及び実行され得る。当然ながら、ルーチンの数及び種類は、本明細書で提供されている説明によって限定されるものではなく、他の数及び種類のプロセス制御関連のルーチンが、所望の場合、ワークステーション 14 内に格納及び実行され得る。ワークステーション 14 はまた、例えば、インターネット、エクストラネット、バス、イーサネット 40 等を介して、企業 WAN ならびに遠隔位置からプラント 10 の遠隔監視または通信を可能にするコンピュータシステムと接続され得る。

20

30

【0027】

図 1 の考察から明らかであるように、ホストワークステーション 14 と制御装置 12 との間及び制御装置 12 とフィールド装置 25 ~ 39 との間の通信は、HART、フィールドバス、及び 4 ~ 20 mA の有線通信接続のうちの 1 つ以上を含む、有線通信接続により実行される。しかしながら、上で述べられているように、配線通信接続は、信頼可能であり、設置及び構成が容易で、無線ネットワーク等の機能能力を分析するか、または視認するための能力をオペレータまたは他のユーザに提供する様式の無線通信により、図 1 のプロセス環境内で置換されるか、または増大され得る。

40

【0028】

例えば、無線ネットワークは、プロセス制御システム全体にわたって使用され得る。結果として、センサ及びアクチュエータ等のプロセス制御システム内の I/O 装置のいくつかまたはすべてが使用されてもよく、有線技術、無線技術、またはそれらの組み合わせを用いて、プロセス制御システムに通信可能に連結され得る。例えば、有線通信は、制御装置 12、ワークステーション 14、及びフィールド装置 25 ~ 31 のいくつかの間で維持

50

されてもよく、無線通信は、制御装置 12、ワークステーション 14、及びフィールド装置 32～39の他方の間で構築され得る。無線技術は、限定されないが、ZigBee、WiFi、Bluetooth、超広帯域(UWB)等、または任意の他の近距離無線技術、ならびに衛星、Wi-Max、及び他の遠距離無線通信を含んでもよい。具体的には、無線技術は、プロセス制御データを伝送するための任意の民生無線製品を含んでもよい。ネットワークプロトコルは、無線技術の上部で実行され得るか、または新規の制御規格が、無線通信のために展開され得る。

【0029】

図2を参照すると、プロセス制御システム10の一部であり得る、DDLに基づくホストシステム50、及びより具体的には、ワークステーション14は、プロセス制御システム10を監視及び動作させるための多数のホストアプリケーション、及びより具体的には、フィールド装置25～39を含み得る。例えば、ホストシステム50は、プロセス制御、シミュレーション、メンテナンス、診断、構成等に関するホストアプリケーションを含んでもよい。ホストシステム50(または、各ホストアプリケーション)はまた、フィールド装置25～39のうちの1つ以上の装置記述(DD)を格納するローカルライブラリまたはデータベース52を有してもよい。図2に示されているように、ホストシステム50は、直接接続または間接接続(例えば、イーサネット40を介した)のいずれかであり得る、インターネット54に接続される。代替的な実施形態では、インターネット54は、すべてまたは部分的に、企業WANと置換され得る。

【0030】

図2に示されているように、ホストシステム50は、インターネット54を介して多数のデータベースまたはシステムに通信可能に連結され、その各々が、プロセス制御システム10と通信し得る。例えば、管理情報システム56は、限定されないが、入力、作成等を含む、プロセス制御システム10からの様々な情報を収集し得る。HART通信基盤データベース58、基盤フィールドバスデータベース60、プロフィバス(プロセスフィールドバス)データベース62、エマソンプロセス管理データベース64等の装置製造業者のデータベース等の様々なDDデータベースが、同様に、インターネット54を介してホストシステム50に通信可能に連結され得る。データベース56、58、60、62、64は、装置に対する装置記述を含む、プロセス制御システム10で使用される様々な装置についての情報を含有し得る。例えば、HART通信基盤データベース58は、プロセス制御システム10で使用される様々なHART装置に対する装置記述を含有してもよく、基盤フィールドバスデータベース60は、プロセス制御システム10で使用される様々なフィールドバス装置に対する装置記述を含有してもよく、プロフィバスデータベース62は、プロセス制御システム10で使用される様々なプロフィバス装置に対する装置記述を含有してもよい。装置記述はまた、装置製造業者からのプロセス制御システム10で使用される様々な装置に対する、エマソンプロセス管理データベース64等の異なる製造業者によって提供された様々なデータベース内に格納され得る。

【0031】

上で述べられているように、装置記述は、変数、データ(パラメータ)、通信(アドレス指定情報)、方法、命令/動作(例えば、較正)、及び装置の様々な特徴と関連するグラフィカルユーザインターフェース(例えば、メニュー及び表示形式)を含む、ある種類のフィールド装置に関するデータ及び動作工程の形式的記述であり、装置製造業者またはDD開発者によってよく知られ、よく支持されている装置記述言語(DDL)で記載されている。例えば、国際電気標準会議(IEC)規格IEC61804-3:2010(E)は、装置パラメータ及び依存性、装置機能(例えば、シミュレーションモード、較正等)、グラフィカルな表現(例えば、メニュー、改善されたユーザインターフェース等)、制御装置との相互作用、グラフ化システム、及び永続データストア等のフィールド装置の特性を記載するための一般的な言語としての電子装置記述言語(EDDL)を特定する。典型的に、装置記述は、対応するフィールド装置内のすべてのパラメータ及びデータにアクセスするために、DDLに基づくホストシステム50によって使用されるメニューの詳細

細及びグラフィカル表示特徴を含む、装置の特定の特徴及び機能を記載する装置記述言語仕様に従って調製された「DDL」拡張子によるテキストファイル等の電子データファイルとして提供される。概して、装置記述は、識別語及び名前の形式を用いた一連の複合文であり、プロセス測定値、導出値等の装置、及び範囲、センサの種類、構成材料等の内部パラメータに対するすべてのアクセス可能な変数を含む。例えば、装置記述に関する文は、一般的に、限定されないが、VARIABLE、MENU、COMMAND、及びMETHODを含み、その各々が、それ自体の構造化情報を有する。

【0032】

VARIABLEは、フィールド装置に含有されるか、またはフィールド装置（例えば、圧力送信器内の圧力、上及び下範囲制限、装置タグ等）と相互作用するホストシステム50によって使用される、任意の値またはデータの種類（例えば、整数、浮動小数点、英数字等）であり得る。VARIABLEに対する構造化情報は、VARIABLEが表示される方法（例えば、変数の名前）、関連する装置、ヘルプファイル等をさらに特定し得る。各COMMANDに対して、装置記述は、命令（例えば、要求、応答、状態、応答目的等）に関連するほとんどすべてに関するデータ構造を特定する。COMMANDは、装置によって認識されたすべての命令に対して提供される。METHODは、ホストが装置に対して実行する動作のセット（例えば、インストール、校正、命令等）である。オペレータは、例えば、ホストを介して提示されたMENUオプションを通してMETHODを起動してもよく、動作のセットは、動作が記載されている順序で実行される。MENUは、エンドユーザに対する提示である。VARIABLE、情報、または他のMENUをオペレータに提示するために使用することができる。

【0033】

DDLに基づくホスト50は、オペレータ等のプロセスプラントのエンドユーザが、ユーザによって選択された各装置に対するDDからの情報を用いてDDLグラフィカルユーザインターフェースを構成することを可能にする構成インターフェースを含み、その結果、ホスト50は、DD内のMENU構造体に基づいて、カスタマイズされたメニューを動的に作成及び維持し得る。具体的には、ユーザインターフェースは、ユーザが、DDからホスト50にDDL構造をエクスポートさせることによってカスタマイズされたDDLグラフィカルユーザインターフェースに対するDDL構造を選択すること、及びDDL構造ならびに直面され得る任意の依存性を明確に説明／提示することを可能にする。DDLグラフィカルユーザインターフェースが選択されたDDL構造にマッピングされるように、選択されたDDL構造及び特定のメニュー構造体を、DDLグラフィカルユーザインターフェースに追加してもよく、構成されたDDLグラフィカルユーザインターフェース（カスタマイズされたメニュー）は、任意の時間に格納及び検索される。構成されたDDLグラフィカルユーザインターフェースは、新規または追加のDDL構造により再構成され、新規または追加の装置（すなわち、追加のDD）に対するDDL構造により再構成され、DDLグラフィカルユーザインターフェース内の既存のDDL構造を修正することによって再構成される等、視認できないDDLグラフィカルユーザインターフェースを介して提示されたデータを視認するために、ユーザによって作動され得る。したがって、ユーザは、ユーザに強制された、製造者が設計するか、またはDD開発者が設計したメニューを有することとは対照的に、ユーザにとって最も重要であると思われる情報に基づいて、メニューをカスタマイズし得る。DDLに基づくホスト50は、DDL構造体情報の一体性を維持し、DDL構造体データ値を取得するために必要とされる命令を出す。

【0034】

概して、DDLグラフィカルユーザインターフェースの構成、格納、及び検索は、DDL条件、FILEデータ、及びLOCAL変数を用いることによって達成され得る。DDL条件は、変数に依存する子オブジェクトを伴う産業においてよく理解されている。変数またはパラメータ間の依存性等の内部依存性が複雑であり得ると仮定すると、DDL条件は、これらの依存性を扱う論理である。具体的には、DDL条件の使用は、構成インターフェースが選択または入力のために利用可能なDDL構造のみならず、構造体を選択する

10

20

30

40

50

か、または入力を提供する際に直面され得る任意の依存性を表示することを可能にする。同様の様式において、DDL条件は、構成インターフェースに、選択された構造体または入力に関連しないDDL構造を「隠さ」せ得る。「隠す」という用語の使用は、DDL構造を視野から隠すこと、DDL構造が選択されることを防ぐこと、または入力がエンターされることを防ぐこと（例えば、メニュー構造体をグレイにする）のいずれかを意味するものとよく理解されている。例えば、ユーザが装置パラメータ（例えば、温度）を選択した場合、構成インターフェースは、そのパラメータに関係しないすべての情報、アイコン、変数等を「隠し」得、選択に関連するDDL構造（例えば、温度設定、温度ユニットオプション等）のみを残す。このように、DDL条件は、構成インターフェースに、先に選択されたDDL構造体に対して有効であるか、または関係するDDL構造のみを提供させ、DDLグラフィカルユーザインターフェース内で先に選択されたDDL構造体の使用は、これらのDDL構造のうちの1つ以上を選択する際に条件付けられ得る。これらのDDL条件はまた、時に膨大な数のDDL構造及び存在する依存性を与えられたメニューを構成する際にユーザを支援するために利用され得る。

10

【0035】

ユーザが構成したDDLグラフィカルユーザインターフェースに対するユーザの選択（例えば、選択されたDDLメニュー構造体、値等）は、FILEデータと称されるDDLファイルデータ構造として格納される場合があり、一定の変数の値が、ユーザのデータベース内に格納され得る。これらの変数は、LOCAL変数と称され、その値は、装置ファームウェア内に格納される他のDDL変数（すなわち、装置変数値）と違って、装置ファームウェア内に格納される必要がない。FILEデータとしての選択の格納は、FILEデータが装置自体に対する任意の追加の作業を必要としないため、既存のファームウェアに対する変更のないDDLグラフィカルユーザインターフェースを可能にする。すなわち、DDLグラフィカルユーザインターフェースの構成が、DDL内のDDLに基づくホスト50の装置外部で行われる。この目的のために、LOCAL変数は、装置内に格納されないが、DDL FILEデータとして格納される。したがって、DDL FILEデータは、DDLグラフィカルユーザインターフェースを、DDLグラフィカルユーザインターフェースに追加されたDDLメニュー構造体にマッピングする、ローカルリソースファイルとして機能する。DDLグラフィカルユーザインターフェースが作動されたとき、DDLに基づくホスト（または、DDLに基づくホストアプリケーション）は、DDL情報（例えば、DDLメニュー構造体、変数値等）を表示に変換するために、FILEデータを利用する。

20

30

【0036】

Variable及びVariableを視認するためのMENUを含む、FILEデータを利用するDDLグラフィカルユーザインターフェースに対するDDLの部分的な疑似コードの例を、下記のように提供する。

40

50

【表 2】

```

        case 0 :
            gauge__showing__pressure
            break ;
        case 1 :
            sweep__chart__plotting__pressure
            break ;
        default :
            pressure__value__as__text、
            COLUMNBREAK、
            pressure__upper__range__value、
            COLUMNBREAK、
            pressure__lower__range__value、
            break ;
    }
}
}
VARIABLE configure__param1
{
    LABEL 「プロセス変数の第1の項目」
    CLASS LOCAL ;
    DEFAULT 0 ;
    TYPE ENUMERATION ;
    ITEMS
    {
        0、「PVの計器」、
        1、「PVの掃引チャート」；
        2、「テキストとしてのPV」
    }
}

```

【0037】

上の例から、ユーザが構成したメニューの選択 (configure__param1、configure__param2、及びconfigure__param3) は、DDLに基づくホスト50等、ユーザのデータベース内のFILEデータ (user__configuration) として定義され、ローカルに格納される。DDLグラフィカルユーザインターフェースを表示するために、DDLが、例えば、ワークステーション14によって利用されたとき、DDL内のFILE文の使用が、メニューの選択を要求する。

【0038】

MENU文は、Variableを得るためのグラフィカルユーザインターフェース (process__variables__root__menu) を記載し、インターフェースのレイアウトを判定するために、上に示されているユーザの選択 (configure

__param1)を実行する。次に、グラフィカルユーザインターフェースの選択の詳細(例えば、圧力計、圧力チャート、値)が、製造業者またはDD開発者からDD内に提供されたデフォルトメニューからの変化を信号伝達する「スイッチ」により定義される。上の例では、画面のタイトル及び表示スタイルが、それぞれ、LABEL(「プロセス変数」)及びSTYLE WINDOWにより特定される。次に、ユーザの選択(configure__param1)を用いて画面レイアウトが判定される。この例におけるユーザ選択(configure__param1)は、同じ情報(gauge__showing__pressure、sweep__chart__plotting__pressure、pressure__value__as__text)に対する1つ以上の形式(例えば、計器、チャート、テキスト)にあり得る、情報(圧力)が表示される方法を定義する。これらの形式は、DDのDDL MENU構造体の一部である。

10

【0039】

上で述べられているように、条件とは、変数に依存する子オブジェクトの使用を指す。例えば、ユーザが圧力変数を監視するための表示を構成した場合、圧力は、上及び下制限間で維持されると予測され、これらの制限の範囲外にある圧力読み取りが、アラートまたはアラームをトリガし得る。したがって、変数圧力を示すためのグラフィカルユーザインターフェースの構成は、圧力に対する上及び下範囲値に対して条件または子オブジェクトを起動し得る。ユーザは、変数(圧力)及び変数に関連する他のメニュー構造体(例えば、計器、チャート、テキスト)を選択することができ、グラフィカルユーザインターフェースに追加される変数(例えば、pressure__upper__range__value、pressure__lower__range__value)に依存する子オブジェクト(例えば、他の変数)により自動的に提示される。DDL条件はまた、かかる選択を義務にしてもよく、その結果、DDLグラフィカルユーザインターフェース内のメニュー構造体(例えば、圧力)の算入が、さらなるメニュー構造体(例えば、pressure__upper__range__value、pressure__lower__range__value)の選択時に条件付けられる。

20

【0040】

VARIABLE文は、MENU文に記載されているグラフィカルユーザインターフェース(process__variables__root__menu)に従って表示される変数データを記載する。Variableは、フィールド装置内に含有されるか、またはフィールド装置と相互作用するホストシステム50によって使用される、データ(例えば、圧力送信器内の圧力、上及び下範囲制限、装置タグ等)の任意の値または種類(例えば、列挙、整数、浮動小数点、英数字等)である。Variableに対する構造化情報は、Variableが表示される方法、表示の名前、関連する装置、ヘルプファイル等をさらに特定し得る。上の例では、Variable圧力は、「プロセス変数の第1の項目」という表示の名前により列挙された値(TYPE ENUMERATION)である。ITEMの属性は、ユーザに表示されるDDの選択された要素(「PVの計器」、「PVの掃引チャート」、「テキストとしてのPV」)を特定する。

30

【0041】

ループ、ユニット、領域等の装置の系列または装置の論理グループ等、特定の装置または装置のグループに対するDDLグラフィカルユーザインターフェースを作成するために、装置に対するDDが検索され、DD内のすべてのDDLメニュー構造体がエクスポートされる。図3は、DDLに基づくホスト50によって維持された新規のDDLグラフィカルユーザインターフェースを構成するためのDDLグラフィカルユーザインターフェース構成ルーチン100のフローチャートである。図3を参照すると、構成ルーチン100は、ブロック102で装置(または、装置のグループ)に対するDDLグラフィカルユーザインターフェースを構成するための命令が受信されたかどうかを判定する。図4は、ユーザが装置または装置のグループ(「ユーザ定義されたメニューを作成」)に対するDDLグラフィカルユーザインターフェースを作成するためのオプションを選択し得る、DDLグラフィカルユーザインターフェース(GUI)の一例を提供している。図4に示されて

40

50

いるように、ユーザ定義されたメニューを作成するためのオプションは、エマソンプロセス管理（商標）によって販売されるAMS（登録商標）Suiteの一部として提供されるAMS（登録商標）装置管理者等の装置管理ツールを介して提供されてもよい。図4に示されているように、オプションは、GUIツールバー内のドロップダウンメニューから、または装置を選択することによって、例えば、装置に対するDDLグラフィカルユーザインターフェースを作成するためのオプションを有するメニューを表示する、装置の名前またはアイコンを右クリックすることによって選択され得る。いったんユーザがDDLグラフィカルユーザインターフェースを作成するためのオプションを選択すると、GUIは、DDLグラフィカルユーザインターフェースを構成するために、図5に示されているウィンドウまたはフレームを表示し得る。図5のウィンドウは、リストから装置の特定の装置を選択し、DDL構造体をエクスポートさせ、選択された装置に関するメニューを構成するために、ユーザに提示され得るDDLグラフィカルユーザインターフェースを作成及び構成するための構成インターフェースに対する表示の一例である。

10

【0042】

ブロック102で、DDLグラフィカルユーザインターフェースを構成するための命令を受信すると、構成ルーチン100は、装置または装置のグループがブロック104で選択されたかどうかを判定する。ユーザは、図4に示されているDevice1~Device6等、GUI内に提示された装置のリストから1つ以上の装置を選択し得る。ユーザによって選択された装置は、ユーザの制御/アクセス可能性内及び/または装置の特定の物理または論理グループ内のこうした装置であり得る。例えば、ユーザがオペレータである場合、ユーザは、ユーザが監視及び/または制御する許可を有する、これらの装置のリストと共に提示され得る。代替的に、または加えて、ユーザに提示された装置のリストは、ループ、ユニット、領域等の物理及び/または論理グループ内のリストであり得る。例えば、図4に示されているように、Device1~Device6は、特定の機器片に対するすべての装置である。

20

【0043】

ユーザは、例えば、リスト内に提示された装置を表すアイコンを、図5に示されている構成ウィンドウにドラッグする（すなわち、ドラッグ&ドロップ）ことによって装置を選択し得る。代替的に、ユーザが装置を選択することによって（例えば、装置のアイコンを右クリックすることによって）DDLグラフィカルユーザインターフェースを作成するためのオプションを選択した場合、選択された装置が、構成ウィンドウに自動的に追加され得る。構成ルーチン100は、特定の期間経過後にタイムアウト（ブロック106）になり得る。

30

【0044】

いったん1つ以上の装置が選択されると、ブロック108で、構成ルーチン100が、選択された装置に対するDDを検索する。装置に対するDDは、装置自体から、管理情報システム56から、様々なDDデータベース58、60、62のうちの1つから、または装置製造業者のデータベース64から予め提供されていた場合、DDLに基づくホスト50のDDライブラリ52から検索され得る。図6は、図3のブロック108で選択された装置の装置記述を検索するためのDD検索ルーチン200の一例である。ブロック202で始まり、DDLに基づくホスト50は、装置に接続し、既知の命令を用いて、装置に対するDD識別を要求する。DD識別は、装置と通信するために使用されたプロトコルにおいて特定され得る。例えば、HARTプロトコルが使用される場合、その装置に対するDD識別を要求するための命令#0が装置に伝送され得る。装置に対する要求を、DDLに基づくホスト50から通信リンク40を介して、またはDDLに基づくホスト50と装置との間の任意の他の通信リンクを介して伝送することができる。

40

【0045】

ブロック204で、DD識別は、ブロック202によって伝送された要求に応答して装置から受信され、装置に対する受信されたDD識別が、メモリ内に格納される。知られているように、装置によって提供されたDD識別は、装置に対する製造者ID、装置の識別

50

子、装置のリビジョン等の情報を含有し得る。ブロック 206 で、DD 検索ルーチン 200 は、DDL に基づくホスト 50（または、DDL に基づくホストアプリケーション）が受信された DD 識別内の装置情報を用いて装置に対する DD を有するかどうかを判定する。例えば、ブロック 206 は、装置に対する DD 識別によって識別された DD に対する DDL に基づくホスト 50 についてのローカル DD ライブラリ 52 またはホストアプリケーションの調査を含有し得る。

【0046】

DDL に基づくホスト 50 が装置に対する DD を有していないと判定された場合、ブロック 208 で、DD 検索ルーチン 200 が、装置に対する DD を有するインターネット 54 に接続された、HART データベース 58 等の DD データベースを識別して、装置に対する DD を得るために、データベースに対する要求を送送する。DD データベースは、インターネット 54 を介して要求を送送し、かかる要求に対する応答を分析することによって識別され得る。当然ながら、HART データベース 58、基盤フィールドバスデータベース 60、プロフィブスデータベース 62、または 1 つ以上の製造業者のデータベース 64 等の、可能なまたは既知の DD データベースのインターネットアドレスが、所望の DD に対する調査を行うためにこれらのデータベースに接続し得る、DDL に基づくホスト 50 によって格納され得る。DD 検索ルーチン 200 はまた、所望の DD に対する調査を行うために、任意の所望の調査エンジン、ブラウザ等を使用してもよい。所望された場合、DD 検索ルーチン 200 は、オペレータがインターネット 54 に対する適切な DD を発見することを支援することを可能にするように、対話型の画面を用いてオペレータと相互作用し得る。データベースが装置に対する DD を含有していると発見されると、DD 検索ルーチン 200 は、装置に対する DD を得るために、データベースに要求を送送する。かかるデータベースに対する要求は、ブロック 204 で得られる装置に対する DD 識別内に含有された情報のいくつかまたはすべてを含有し得る。

【0047】

ブロック 208 で、装置に対する DD が DDL に基づくホスト 50 にダウンロードされた後か、またはブロック 206 で、DDL に基づくホスト 50 が装置に対する DD を有すると判定されたときに、ブロック 210 で、DD 検索ルーチン 200 が、DDL に基づくホスト 50 を更新する。ユーザは、DDL に基づくホスト 50 が必要に応じて DD に対して自動的に更新されるべきであることを特定し得る。代替的に、ブロック 210 で、所望の DD により DDL に基づくホスト 50 を更新するために、命令を送送することができる。装置に対する DD による DDL に基づくホスト 50 の更新は、特定の位置で装置に対する DD をメモリに保存すること、及び必要に応じて、特定の位置に対する要求を DDL に基づくホスト 50 に挿入することを伴い得る。装置に対する DD による DDL に基づくホスト 50 の更新はまた、DD ライブラリ 52 内の装置に対する DD を挿入することを伴い得る。

【0048】

図 3 を参照すると、いったん選択された装置が検索されると、構成ルーチン 100 が、装置に対する DD を読み取り、ブロック 110 で、すべての DDL メニュー及び DDL メニュー構造体（例えば、変数、グラフ、画像、グリッド、チャート等のメニュー内に表示されたメニュー項目またはパラメータ）を DDL に基づくホスト 50（または、ホストアプリケーション）に内にエクスポートさせる。例えば、構成ルーチン 100 は、構成インターフェースを通してすべての DDL メニュー構造体をユーザに利用可能にするために、DD をスキャン及び分析し得、その結果、ユーザが所望するメニューオプションを有する DDL グラフィカルユーザインターフェース及び DDL グラフィカルユーザインターフェースに適用されるパラメータを設計するように、DDL メニュー構造体及び値が、構成ルーチン 100、ならびに下にさらに記載されている追加のルーチンに提供され得る。製造業者は、ユーザがアクセスすることができるものを制限することを未だ望んでいる場合があることを理解すべきである。このように、DDL メニュー及び DDL メニュー構造体のエクスポートは、ユーザが情報を視認する方法の観点から、ユーザが DDL グラフィカ

10

20

30

40

50

ルユーザインターフェースを構成することを可能にするが、ユーザが任意かつすべての情報を視認することを必ずしも可能にしない。すなわち、装置製造業者によって開発されたリソースファイルから作成されたデフォルト装置メニューに通常表示されるデータは、ユーザに利用可能なままであるが、DDLメニュー及びDDLメニュー構造体のエクスポートは、ユーザにさらなるデータをエクスポートしない。むしろ、DDLメニュー及びDDLメニュー構造体のエクスポートは、ユーザがすでにデフォルト装置のメニューで視認することができるメニュー構造体を取り、そのデータがDDLグラフィカルユーザインターフェース内に提示される方法をユーザが構成することができるように、それらをオプションにする。

【0049】

ブロック112で、DDLメニュー構造体は、構成インターフェースにおいてユーザに提示され、その一例が、図7に示される。図7に見られるように、構成インターフェースは、メニュー構造体テンプレート300内にエクスポートされたDDLメニュー構造体(「パラメータ」)を提示する。例示的な目的のためにVariable#(Variable1、Variable2、Variable3等)として示されているが、DDLメニュー構造体の名前は、ユーザに理解可能な様式で記載され、かつ明確に記載される。メニュー構造体テンプレート300に加えて、異なる種類のメニュー(例えば、グループボックス、ウィンドウ、メニュー、ページ)に対するオプションを提示するために、メニュースタイルテンプレート302がまた、提供されてもよい。構成テンプレート304は、DDLグラフィカルユーザインターフェースに対するテンプレートとして機能し、ユーザがDDLグラフィカルユーザインターフェースを設計することができる領域を提供する。

【0050】

ブロック114で、DDLメニュー構造体を選択されてもよく、ユーザの選択(例えば、配置、メニュースタイル等)に従ってブロック116でDDLグラフィカルユーザインターフェースに追加されてもよい。例えば、図7を参照すると、ユーザは、DDLメニュー構造体を表す特定のグラフィカルなアイコンをメニュー構造体テンプレート300から選択し、アイコンを構成テンプレート304にドラッグして、DDLメニュー構造体をユーザが望む場所に配置し得る。構成テンプレート304内で、ユーザは、任意の所望の形態でメニュー構造体を配置し得る。同様に、ユーザは、メニュースタイルを表すアイコンをメニュースタイルテンプレート302から選択し、アイコンを構成テンプレート304にドラッグして、メニュースタイルをユーザが望む場所に配置し得る。構成テンプレート304内で、ユーザは、任意の所望の形態でメニュースタイル及び/またはメニュー構造体を配置し得る。例えば、特定の装置を監視するためのすべての関連する情報を示すDDLグラフィカルユーザインターフェースを有するのではなく、ユーザは、複数の装置を選択して、DDLグラフィカルユーザインターフェース内に表示するための各装置に対する温度変数を選択し、それによってDDが特定の装置または装置の種類にのみ特有であるため、装置DDからデフォルトメニューにとって支持されない特徴である、温度のみの読み取り、圧力のみの読み取り、または複数の装置に対する任意の他の変数を示すようにカスタマイズされたDDLグラフィカルユーザインターフェースを作成し得る。

【0051】

ブロック116で、メニュー構造体がDDLグラフィカルユーザインターフェースに追加されたとき、構成ルーチン100はまた、追加されたメニュー構造体が1つ以上の依存性(例えば、変数に依存する子オブジェクト)を有するかどうかを判定し得る。かかる事例において、構成ルーチン100は、ブロック118での制御をブロック114に戻し、DDL条件を用いて予め選択されたメニュー構造体に依存するメニュー構造体のみを提示し得る。例えば、追加された変数が選択された装置に対する圧力である場合、圧力に対する依存性は、上及び下範囲値等の重要なパラメータであり得る。DDL条件を用いて、構成ルーチン100は、すべての他のメニュー構造体を「隠し」、圧力に対する上及び下範囲のメニュー構造体のみをユーザに提示し得る。さらに、DDL条件を用いて、構成ルーチン100は、条件が満たされるまで、ユーザがメニューを完了及び保存することを防ぐ

10

20

30

40

50

ことさえ可能であり、その結果、ＤＤＬグラフィカルユーザインターフェース内のメニュー構造体の算入が、さらなるメニュー構造体の選択時に条件付けられる。

【００５２】

装置（または、装置のグループ）はＤＤ内に多くのメニュー構造体を有し得ると仮定すると、ＤＤＬ条件はまた、メニューを構成する際にユーザをガイドするために使用されてもよい。構成ルーチン１００は、ＤＤＬグラフィカルユーザインターフェースを構成する目的に対してメニュー構造体をエクスポートさせると同時に、ＤＤＬ条件を使用して、メニュー構造体のサブセットのみを提示し、他のすべてを「隠し」、次に、予め選択されたメニュー構造体に関連するこれらのメニュー構造体のみを提示することによって、ユーザに対するメニュー構造体の提示を単純化し得る。例えば、構成ルーチン１００は、変数に対するすべての他のメニュー構造体（例えば、グラフ、チャート、依存性等）を「隠す」と同時に、選択された装置に対する変数（例えば、圧力、温度等）のみを提示することにより始まり得る。いったん変数（例えば、圧力）が選択されると、すべての他のメニュー構造体（選択されない変数を含む）が、隠され得るか、または別様に、選択された変数に関連するもの（例えば、圧力グラフ、圧力チャート、圧力依存性等）を除いて選択不能であり得る。いったん変数に対するＤＤＬ条件が満たされると、構成ルーチン１００は、選択された装置に対する変数のみを提示することに戻ってもよい。

【００５３】

一実施形態では、関連するメニュー構造体の提示であっても、ＤＤＬ条件に基づいている場合がある。例えば、ユーザが変数「圧力」を選択すると、次に、構成ルーチン１００は、圧力に対する上及び下範囲値に関するメニュー構造体を提示し得る。一度のみユーザが上及び下範囲値に関するメニュー構造体を選択すると、構成ルーチン１００は、圧力に対するグラフ、チャート等に関するメニュー構造体を提示する。したがって、グラフ、チャート等は、圧力に対する上及び下範囲値に依存して作成され、次に、変数圧力に依存する。この様式では、ＤＤＬ条件を利用して、構成を通してユーザをガイドするために、関連するメニュー構造体であっても互いに依存させ得る。いくつかの事例では、構成ルーチン１００が、メニュー構造体を選択されるまでユーザがメニューを完了及び保存することを進めないか、または可能にしないように、ＤＤＬ条件は、義務的であり得る（例えば、圧力に対する上及び下範囲値を選択する）。他の事例では、ＤＤＬ条件は、選択的であり得る（例えば、圧力に対するグラフまたはチャートを選択する）。したがって、ＤＤＬ条件の使用は、ＤＤＬグラフィカルユーザインターフェース内の算入に対するオプションとして利用可能なＤＤＬメニュー構造体の数によりユーザを圧倒させることなく、メニューの構成を通してユーザをガイドするために使用され得る。

【００５４】

ブロック１１８で判定されるように、ＤＤＬグラフィカルユーザインターフェースの構成をユーザが終えた場合、ブロック１２０で、メニューを保存し得る。終わっていない場合、制御が、次のＤＤＬメニュー構造体の選択に対するブロック１１４に戻り得る。ブロック１２０で、構成されたＤＤＬグラフィカルユーザインターフェースが保存されると、ＤＤＬグラフィカルユーザインターフェースは、ＤＤＬに基づくホスト５０により保存される。具体的には、ＤＤＬグラフィカルユーザインターフェースが、ＤＤＬに基づくホスト５０等のユーザのデータベース内に格納された、選択されるＤＤＬメニュー構造体の値によるＦＩＬＥデータとして保存される。ＤＤＬメニュー構造体は、装置により格納されるではなく、ＬＯＣＡＬ変数としてのＦＩＬＥデータとして格納され得る。ユーザの選択をＦＩＬＥデータとして格納することによって、ＤＤＬグラフィカルユーザインターフェースは、既存のファームウェアを変更する必要なく作成及び格納され得、装置自体への追加の変更を必要としない。すなわち、解法は、全体的にＤＤＬ内にある。したがって、ユーザは、ＤＤＬに基づくホスト５０によって動的に作成及び維持されるＤＤＬグラフィカルユーザインターフェースを構成することができる。図４、５及び７に示されているような構成インターフェースは、ＤＤＬグラフィカルユーザインターフェースに追加された、こうした構造を有する、ユーザにとって重要であるＤＤＬメニュー構造体をユーザが選び出

10

20

30

40

50

すことを可能にする。下にさらに考察されるように、ユーザは、任意の時間にＤＤＬグラフィカルユーザインターフェースを要求及び作動させ、ＤＤＬグラフィカルユーザインターフェースを隠して、異なるＤＤＬメニュー構造体によりＤＤＬグラフィカルユーザインターフェースを再構成し得る。

【００５５】

図８は、図３に開示されている方法に基づき、ＤＤＬに基づくホストシステム５０によって維持されたＤＤＬグラフィカルユーザインターフェースを作成及び構成する際のプロセス間の相互作用を示すシーケンス図である。ＤＤＬに基づくホスト５０に対して実行され得る、ＤＤＬホストアプリケーション３５０により始まり、新たにユーザが定義したメニュー（「作成」）を作成する非同期要求が、メニュー作成者インターフェース３５２に出される。ＤＤＬホストアプリケーション３５０は、ＡＭＳ（登録商標）Ｓｕｉｔｅによるアプリケーションであってもよく、メニュー作成者インターフェース３５２は、図５及び７に図示されているメニューを作成及び構成するための構成インターフェースであってもよく、要求は、ＡＭＳ（登録商標）アプリケーションからメニューを作成するための構成インターフェースに対してなされる。新規のＤＤＬグラフィカルユーザインターフェースを作成するための要求は、ユーザが図４の「ユーザが定義するメニューを作成」を選択したときに、図３のブロック１０２でＤＤＬグラフィカルユーザインターフェースを構成するための命令に応答してなされ得る。次に、ＤＤＬグラフィカルユーザインターフェースを作成する別の非同期要求（「作成」）が、メニュー作成者インターフェース３５２から、ＡＭＳ（登録商標）装置管理者または図４に図示されている他の装置管理ツール等の装置ファインダインターフェース３５４になされ得る。代替的に、新規のＤＤＬグラフィカルユーザインターフェースを作成するための要求は、すでに別様に起動されている場合、メニュー作成者インターフェース３５２に由来し得る。同様に、新規のＤＤＬグラフィカルユーザインターフェースを作成するための要求は、図４に関して考察されているような装置管理ツール内の装置のリストからユーザが装置を選択するとき等、装置ファインダインターフェース３５４に由来し得、その事例では、要求は、装置ファインダインターフェース３５４からメニュー作成者インターフェース３５２に出される。

【００５６】

新たにユーザが定義したメニューを作成するための要求は、図３のブロック１０４のような選択に対するユーザに提示する装置のリストを検索するために、装置ファインダインターフェース３５４が、データストアアプリケーション３５６に非同期要求（「Ｇｅｔ Ｄｅｖｉｃｅ」）を出させる。それに応じて、データストアアプリケーション３５６は、装置のリスト（「装置リスト」）を装置ファインダインターフェース３５４に戻す。装置のリストは、ユーザが監視及び／制御する許可を与えられた装置、または装置ファインダインターフェースにおいて作成された、ループ、ユニット、領域等の選択に基づく特定のループ、ユニット、領域等内の装置であり得る。例えば、ユーザがナビゲーションツリーを用いて特定のユニットを選択した場合、そのユニットに対する装置（そのユニット内のすべてのループを含む）が、データストア３５６から検索され得る。同様に、ユーザがナビゲーションツリーを用いてユニット内の特定のループを選択した場合、そのループに対する装置が、データストア３５６から検索され得る。

【００５７】

次に、装置ファインダインターフェース３５４は、これらの装置をユーザに提示してもよく、それに応じて、装置ファインダインターフェース３５４は、図３のブロック１０４に対応して、ユーザからの装置の選択（「Ｄｅｖｉｃｅ 選択」）を示すシーケンスから外部的にメッセージを受信し得る。それに応じて、非同期メッセージ（「Ｄｅｖｉｃｅ」）は、装置ファインダインターフェース３５４から、選択された装置を表すメニュー作成者インターフェース３５２に伝送される。次に、作成者のインターフェース３５２は、選択された装置に対するすべてのＤＤＬメニュー構造体を検索するために、データストアアプリケーションに非同期要求（「Ｇｅｔ Ｄｅｖｉｃｅ Ｐａｒａｍｅｔｅｒ」）を出す。ＤＤＬメニュー構造体を検索するための要求が、図３の及び図６のブロック１０８に図示され

10

20

30

40

50

ているような選択された装置に対するDDを検索するためのデータストアアプリケーション356内のサブルーチンを起動させ得る。

【0058】

データストアアプリケーション356は、選択された装置(Device Parameter)に関するメニュー構造体のリストによりメニュー作成者インターフェース352に回答し、その地点で、メニュー構造体は、ユーザに提示するために、メニュー作成者352に(または、よって)エクスポートされる。次に、メニュー作成者インターフェース352は、図3のブロック114~118及び図7に図示されているインターフェースに対応して、メニューに追加するためにユーザによって選択されたDDLメニュー構造体を示すシーケンスから外部的に非同期メッセージ(「Parameterを選択/Menuに追加」)を受信し得る。

10

【0059】

メニュー構造体のDDLグラフィカルユーザインターフェースへの追加は、DDLグラフィカルユーザインターフェースを選択されたDDLメニュー構造体にマッピングすることを伴い得る。DDLグラフィカルユーザインターフェースの選択されたDDLメニュー構造体へのマッピングは、DDLグラフィカルユーザインターフェースを作動させたときにDDLメニュー構造体を要求するための要求または命令をDDLグラフィカルユーザインターフェースのFILEデータに挿入することを伴い得る。マッピングはまた、DDLメニュー構造体にリンクされた(マッピングされた)DDLグラフィカルユーザインターフェースのFILEデータに値を追加することを伴い得る。例えば、上に提供されたFILEデータを利用するDDLグラフィカルユーザインターフェースに対するDDLの部分的な疑似コードの例を参照すると、DDLメニュー構造体「sweep__chart__plotting__pressure」、及び「pressure__value__as__text」は、DDLグラフィカルユーザインターフェースが作動されたときに、DDLメニュー構造体のグラフィカルな表現を表示装置に表示するための要求として機能し得る。Variable(圧力)は、DDLメニュー構造体「gauge__showing__pressure」、「sweep__chart__plotting__pressure」、及び「pressure__value__as__text」にそれぞれマッピングされる、「0」、「1」、及び「2」の値を有する。

20

【0060】

いったんDDLメニュー構造体を選択及び追加されると、図3のブロック120に対応して、DDLグラフィカルユーザインターフェースを保存する(メニューを保存)ためのデータストア365の非同期要求がなされる。上で考察されているように、DDLグラフィカルユーザインターフェースは、DDLに基づくホスト50にDDL FILEデータとして格納され得る。DDLに基づくホスト50は、関連する装置から検索され、構成ルーチン100中にエクスポートされたような、DDLメニュー構造体情報の一体性を維持する任務を負い、DDLメニュー構造体は、DDデータベース52内等、ローカルに保存され得る。DDLに基づくホスト50は、DDL構造体データ値を取得するために必要とされる命令をさらに出す。例えば、FILEデータ内のDDL構造が要求として機能する場合、DDLに基づくホスト50は、これらの要求を解釈し、ローカルデータベースからDDLグラフィカルユーザインターフェースに対するDDL構造を検索し、DDLグラフィカルユーザインターフェースの一部としてDDL構造を表示するための命令を出す。

30

40

【0061】

画像(グラフィカルな表現)(選択された装置の360、変数の値を入力するためのボックス370~378と共にユーザによって選択された変数362~368のグラフィカルな表現、及びDDL METHODを起動させるためのMENUオプション380を含む、ユーザが構成したDDLグラフィカルユーザインターフェースの一例が、図9に示されている。いったんDDLグラフィカルユーザインターフェースが保存されると、DDLグラフィカルユーザインターフェースが、メニュー装置テンプレート(例えば、ウィンドウまたはフレーム)内で視認され得る。AMS(登録商標)Suiteアプリケーション

50

等の D D L ホストアプリケーションを用いて、D D L グラフィカルユーザインターフェースは、ナビゲーションメニューオプション 3 8 4 としてショートカットバー 3 8 2 を選択することによって、またはオプションとして D D L グラフィカルユーザインターフェースによりコンテキストメニューを作成するために、装置リストから装置（例えば、装置の名前またはアイコン上で右クリックする）を選択することによって、作動または要求され得る。「新規メニュー」として表示されているが、これらのオプションに対するラベルは、ユーザによって保存される D D L グラフィカルユーザインターフェースと同じであってもよい。

【 0 0 6 2 】

図 1 0 は、A M S（登録商標）装置管理者等の D D L ホストアプリケーションから D D L グラフィカルユーザインターフェースを作動させるための例示的な D D L グラフィカルユーザインターフェース作動ルーチン 4 0 0 のフローチャートである。ブロック 4 0 2 で始まり、D D L ホストアプリケーションは、例えば、D D L に基づくホスト 5 0 または他のワークステーション 1 4 を用いて起動される。D D L ホストアプリケーションは、例えば、監視及び／または制御のためのユーザの許可内の装置等、ブロック 4 0 4 で 1 つ以上の装置をユーザに列挙するか、または別様に、表示し得る。先に述べられているように、装置は、特定のループ、ユニット、領域等内にあってもよい。代替的に、または加えて、D D L ホストアプリケーションは、装置製造業者または D D 開発者によって提供されたデフォルト規格 D D メニュー、及び、選択された装置に対する任意の構成された D D L グラフィカルユーザインターフェースを含む、ブロック 4 0 6 のような装置と関連する 1 つ以上のメニューオプションを表示し得る。

【 0 0 6 3 】

ユーザは、限定されないが、表示されたメニューオプションから装置を選択することが、または表示されたメニューオプションから D D L グラフィカルユーザインターフェースを選択することを含む、様々な方法でメニューを選択し得る。例えば、ブロック 4 0 8 で、D D L グラフィカルユーザインターフェースをメニューオプションから選択してもよく、選択された D D L グラフィカルユーザインターフェースは、ブロック 4 1 0 で起動される。ブロック 4 1 2 で、例えば、装置の名前またはアイコンを右クリックすることによって装置が選択された場合、メニュー作動ルーチン 4 0 0 は、ブロック 4 1 4 で、D D L グラフィカルユーザインターフェースが存在しているか否かを判定し得る。存在している場合、4 1 2 での装置の選択が、ブロック 4 1 6 でのオプションとして D D L グラフィカルユーザインターフェースと共にコンテキストメニューを生成させ得る。D D L グラフィカルユーザインターフェースに対するオプションを選択する際、ブロック 4 1 8 で、D D L グラフィカルユーザインターフェースが起動される。D D L グラフィカルユーザインターフェースが存在しない場合、メニュー動作ルーチン 4 0 0 は、ブロック 4 2 0 で、標準的なコンテキストメニューを作成させ得る。デフォルト規格装置メニューに対するオプションを選択する際、ブロック 4 2 2 で、装置に対する D D 及びリソースファイルを用いて、デフォルト規格装置記述メニューが起動される。

【 0 0 6 4 】

いったん D D L グラフィカルユーザインターフェースが作成、構成、及び保存されると、D D L グラフィカルユーザインターフェースは、D D L グラフィカルユーザインターフェース内の 1 つ以上の D D L メニュー構造体を修正する（例えば、構造体を削除する、構造体を追加する、構造体を修正する）ように、または新規の装置に対する D D L メニュー構造体を追加するように、編集され得る。図 1 1 は、装置を修正するか、または既存の D D L グラフィカルユーザインターフェースに追加するための編集ルーチン 5 0 0 の一例である。ブロック 5 0 2 で始まり、ユーザの D D L グラフィカルユーザインターフェースが表示される。D D L グラフィカルユーザインターフェースは、図 9 に示されているものと類似した G U I に提示され得てもよく、ユーザは、ショートカットバー 3 8 2 から、またはナビゲーションメニューオプション 3 8 4 として、D D L グラフィカルユーザインターフェースを選択し得る。一例において、選択された D D L グラフィカルユーザインターフ

エースを編集するためのオプションは、ツールバーからか、またはD D Lグラフィカルユーザインターフェースを編集するためのオプションを有するコンテキストメニューを作成するために、ショートカットバー 3 8 2 またはナビゲーションメニュー 3 8 4 内に列挙されたD D Lグラフィカルユーザインターフェースを選択する（例えば、右クリックすることによって行われ得る。

【 0 0 6 5 】

いったんD D Lグラフィカルユーザインターフェースが、D D Lグラフィカルユーザインターフェースを編集するためのオプションによりブロック 5 0 4 で選択されると、ブロック 5 0 6 で、D D Lグラフィカルユーザインターフェースに対するD D Lメニュー構造体がエクスポートされる。図 3 のブロック 1 1 0 でD D Lメニュー構造体をエクスポートすることと同様に、編集ルーチン 5 0 0 は、構成インターフェースを通してすべてのD D Lメニュー構造体をユーザに利用可能にするために、D D Lグラフィカルユーザインターフェースをスキャン及び分析し、その結果、D D Lグラフィカルユーザインターフェース内のD D Lメニュー構造体及び値がユーザに表示される。加えて、ユーザがD D Lメニュー構造体をD D Lグラフィカルユーザインターフェースに追加することを望む場合、編集ルーチン 5 0 0 は、D D Lグラフィカルユーザインターフェースを元々作成しているか、または予め構成しているときに検索された装置に対するD Dを検索し得る。編集ルーチン 5 0 0 は、構成インターフェースを通してユーザに利用可能であるすべてのD D Lメニュー構造体を作成するために、同様にD Dをスキャン及び分析し得る。D DがD Dライブラリ 5 2 内に格納されなかった場合、編集ルーチン 5 0 0 は、図 6 のD D検索ルーチン 2 0 0 を利用してもよい。

【 0 0 6 6 】

ブロック 5 0 6 で、D D Lメニュー構造体をエクスポートすると、ブロック 5 0 8 で、D D Lメニュー構造体が、構成インターフェース内でユーザに提示される。構成インターフェースは、図 7 に示されているものと同じであり得、D D Lグラフィカルユーザインターフェースが、構成テンプレート 3 0 4 の編集のために表示される。いったんD D Lメニュー構造体がエクスポートされ、D D Lグラフィカルユーザインターフェースが編集のために提示されると、ユーザは、例えば、D D Lグラフィカルユーザインターフェースを修正すること、または新規装置に対するD D Lメニュー構造体をD D Lグラフィカルユーザインターフェースに追加すること等、D D Lグラフィカルユーザインターフェースを編集するためのオプションを提示され得る。ユーザの選択に基づいて、ブロック 5 1 0 で判定されるように、編集ルーチン 5 0 0 は、ブロック 5 1 2 で、新規装置に対する1つ以上のD D Lメニュー構造体をD D Lグラフィカルユーザインターフェースに追加し得るか、または例えば、D D Lメニュー構造体を削除または修正することによって、ブロック 5 1 4 で、D D Lグラフィカルユーザインターフェースを修正し得る。

【 0 0 6 7 】

ブロック 5 1 2 で、新規装置に対するD D Lメニュー構造体を追加するために、新規装置ルーチン 6 0 0 は、図 1 2 で示されているように起動され得る。新規装置ルーチンは、図 3 に示されている装置記述構成ルーチン 1 0 0 と同様であり得る。具体的には、ブロック 6 0 2 で始まり、D D Lメニュー構造体が追加され得る装置は、例えば、図 4 に示されている装置管理ツールを用いて、ユーザに検索及び提示される。装置追加ルーチン 6 0 0 は、ブロック 6 0 4 で、装置または装置のグループが選択されたかどうかを判定する。ユーザは、装置管理ツール G U I 内に提示された装置のリストから1つ以上の装置を選択し得る。D D Lグラフィカルユーザインターフェースの作成と同様に、ユーザによって選択された装置は、ユーザの制御 / アクセス可能性内の、かつ / または装置の特定の物理または論理グループ内のこうした装置であり得る。ユーザは、例えば、リスト内に提示された装置を表すアイコンを図 5 の構成ウインドウ 3 0 4 にドラッグすること（すなわち、ドラッグ & ドロップ）によって、装置を選択し得る。代替的に、ユーザが装置を選択（例えば、装置のアイコンを右クリックすることによって）した場合、選択された装置は、構成ウインドウ 3 0 4 に自動的に追加され得る。装置追加ルーチン 6 0 0 は、特定の時間経過後

10

20

30

40

50

にタイムアウト（ブロック 6 0 6）になり得る。

【 0 0 6 8 】

いったん 1 つ以上の装置が選択されると、装置追加ルーチン 6 0 0 は、ブロック 6 0 8 で、選択された装置に対する DD を検索する。装置に対する DD は、装置自体から予め提供された場合に DD L に基づくホスト 5 0 の DD ライブラリ 5 2 から、管理情報システム 5 6 から、様々な DD データベース 5 8、6 0、6 2 のうちの 1 つから、または装置管理データベース 6 4 から検索され得る。装置追加ルーチン 6 0 0 は、ブロック 6 0 8 で、選択された装置の DD を検索するための図 6 の DD 検索ルーチン 2 0 0 を利用し得る。

【 0 0 6 9 】

選択された装置に対する DD が検索されたとき、装置追加ルーチン 6 0 0 は、ブロック 6 1 0 で、新規装置に対する DD を読み取り、DD 内のすべての DD L メニュー及び DD L メニュー構造体（例えば、変数、グラフ、画像、グリッド、チャート等のメニューに表示されたメニュー項目またはパラメータ）を DD L に基づくホスト 5 0（または、ホストアプリケーション）にエクスポートする。装置追加ルーチン 6 0 0 は、構成インターフェースを通してすべての DD L メニュー構造体をユーザに利用可能にするために、DD をスキャン及び分析し得、その結果、DD L メニュー構造体を DD L グラフィカルユーザインターフェースに追加するために、DD L メニュー構造体及び値が、装置追加ルーチン 6 0 0 に提供され得る。図 3 と同様に、DD L メニュー及び DD L メニュー構造体のエクスポートは、選択された装置に対するデフォルト装置メニューにおいてユーザがすでに視認することができるメニュー構造体を取り、DD L グラフィカルユーザインターフェースに追加するための DD L メニュー構造体をユーザが選択することができるように、それらのオプションにする。

【 0 0 7 0 】

ブロック 6 1 2 で、新規装置に対する DD L メニュー構造体は、図 7 のように構成インターフェース内でユーザに提示される。一例として図 7 を用いて、編集された DD L グラフィカルユーザインターフェースが、構成テンプレート 3 0 4 内に表示され得、エクスポートされた DD L メニュー構造体は、メニュー構造体テンプレート 3 0 0 内に提供される。メニュー構造体テンプレート 3 0 0 内にエクスポートされた DD L メニュー構造体は、単に、ブロック 6 1 0 で新規装置に対してエクスポートされ得るか、または新規装置に対する DD L メニュー構造体、ならびに DD L グラフィカルユーザインターフェース内で使用される DD L メニュー構造体、及び/または DD L グラフィカルユーザインターフェースを元々作成及び構成するときに検索された装置に対する DD L メニュー構造体を含み得る。

【 0 0 7 1 】

いったん DD L メニュー構造体がエクスポート及び提示されると、DD L メニュー構造体が、ブロック 6 1 4 で選択され得、ユーザの選択（例えば、配置、メニュースタイル等）に従ってブロック 6 1 6 で DD L グラフィカルユーザインターフェースに追加される。例えば、図 7 を再び参照すると、ユーザは、DD L メニュー構造体を表す特定のグラフィカルなアイコンをメニュー構造体テンプレート 3 0 0 から選択し、アイコンを構成テンプレート 3 0 4 にドラッグして、DD L メニュー構造体をユーザが望む場所に配置し得る。構成テンプレート 3 0 4 内で、ユーザは、任意の所望の形態でメニュースタイル及び/またはメニュー構造体を配置し得る。DD L メニュー構造体を DD L グラフィカルユーザインターフェースに追加するためのプロセスは、上に記載されている図 3 のブロック 1 1 6 のものと同じであってもよい。メニュー構造体がブロック 6 1 6 でメニューに追加されると、装置追加ルーチン 6 0 0 は、依存性を有する任意の変数に対する図 3 に関して上に記載されている DD L 条件の概念を利用して、DD L グラフィカルユーザインターフェースを編集する際にユーザをガイドする。

【 0 0 7 2 】

ブロック 6 1 8 で判定されるように、新規装置に対する DD L メニュー構造体の DD L グラフィカルユーザインターフェースへの追加をユーザが終えた場合、制御が、編集ルー

10

20

30

40

50

チン 5 0 0 に戻ってもよい。終えていない場合、装置追加ルーチン 6 0 0 が D D L メニュー構造体の選択を待つため、制御は、ブロック 6 1 4 に戻ってもよい。

【 0 0 7 3 】

図 1 1 を再び参照すると、ブロック 5 1 4 で D D L グラフィカルユーザインターフェースを修正するようにユーザが選択した場合、編集ルーチン 5 0 0 は、図 1 3 に示されているように修正ルーチン 7 0 0 を起動させ得る。ブロック 7 0 2 で始まり、D D L メニュー構造体は、例えば、メニュー構造体 3 0 0 内に提示された D D L メニュー構造体を表すグラフィカルなアイコンを図 7 の構成テンプレート 3 0 4 にドラッグするか（すなわち、ドラッグ&ドロップ）、構成テンプレート 3 0 4 の外部で構成テンプレート 3 0 4 内に提示された D D L メニュー構造体を表すアイコンをドラッグするか、または D D L メニュー構造体を編集するためのオプションを有するコンテキストメニューを表示するために、D D L メニュー構造体を表すアイコンを右クリックすることによって、ユーザの入力に基づき、ブロック 5 0 8 で、D D L メニュー構造体の表示から選択される。上で述べられているように、ユーザに提示された D D L メニュー構造体は、編集される D D L グラフィカルユーザインターフェース内で使用される D D L メニュー構造体、ならびに D D L グラフィカルユーザインターフェースを元々作成及び構成するか、または予め編集するときに検索された装置に対する D D L メニュー構造体の両方を含み得る。

【 0 0 7 4 】

いったん D D L 構造体がブロック 7 0 2 で選択されると、D D L メニュー構造体を D D L グラフィカルユーザインターフェースから削除すること、D D L メニュー構造体を D D L グラフィカルユーザインターフェースに追加すること、または D D L メニュー構造体を D D L グラフィカルユーザインターフェース内で修正すること等、D D L グラフィカルユーザインターフェースを修正するためのオプションをユーザに提示するように、修正ルーチン 7 0 0 に進み得る。一実施形態では、D D L メニュー構造体を選択される様式が、D D L グラフィカルユーザインターフェースを修正するためのオプションを自動的に起動させ得る。例えば、構成テンプレート 3 0 4 の外部で構成テンプレート 3 0 4 内の D D L メニュー構造体を表すアイコンをユーザがドラッグした場合、修正ルーチン 7 0 0 は、D D L メニュー構造体を D D L グラフィカルユーザインターフェースに追加するものとしてこの動作を解釈して、ブロック 7 0 4 で、D D L メニュー構造体削除ルーチンを起動させ得る。同様に、D D L メニュー構造体を表すアイコンをメニュー構造体テンプレート 3 0 0 から構成テンプレート 3 0 4 にユーザがドラッグした場合、修正ルーチン 7 0 0 は、D D L メニュー構造体を D D L グラフィカルユーザインターフェースに追加するものとしてその動作を解釈し、ブロック 7 0 6 で、D D L メニュー構造体追加ルーチンを起動させる。ユーザがメニュー構造体テンプレート 3 0 0 内の D D L メニュー構造体を表すアイコンをクリックした場合、修正ルーチン 7 0 0 は、ブロック 7 0 8 で、D D L メニュー構造体を修正するものとしてその動作を解釈して、D D L メニュー構造体修正ルーチンを起動させる。代替的に、または加えて、ブロック 7 0 4、7 0 6、7 0 8 の修正オプションの各々は、コンテキストメニューを介して（例えば、D D L メニュー構造体を表すアイコン上で右クリックする）及び/またはツールバーメニューを介して起動され得る。

【 0 0 7 5 】

図 1 4 は、D D L メニュー構造体を D D L グラフィカルユーザインターフェースから削除するための図 1 3 のブロック 7 0 4 で示されている D D L メニュー構造体削除ルーチン 8 0 0 の一例である。ブロック 8 0 2 で始まり、D D L メニュー構造体削除ルーチン 8 0 0 は、D D L メニュー構造体を選択されたことを認証し得、ブロック 8 0 4 で、選択された D D L メニュー構造体が削除される。一例において、修正ルーチン 7 0 0 が構成テンプレート 3 0 4 の外部で構成テンプレート 3 0 4 内から D D L メニュー構造体を表すアイコンをドラッグする動作を解釈した場合、ブロック 8 0 2 を使用して、構成インターフェース内のアイコンの移動を追跡し得、一度のみアイコンが構成テンプレート 3 0 4 の外部に「ドロップ」されると、ブロック 8 0 4 で、D D L メニュー構造体の D D L グラフィカルユーザインターフェースからの削除が生じる。D D L メニュー構造体を削除するためのオ

10

20

30

40

50

クションがコンテキストメニューを介して起動された場合、ブロック 802 を使用して、ユーザが DDL メニュー構造体に対するアイコンに対して選択した（例えば、右クリックした）と認証して、ユーザがコンテキストメニューから削除オプションを選択したときに、ブロック 804 で、DDL メニュー構造体を削除し得る。同様に、DDL メニュー構造体を削除するためのオプションがツールバーメニューを介して起動された場合、ブロック 802 を使用して、ユーザが DDL メニュー構造体を選択した（例えば、ユーザが削除するために DDL メニュー構造体に対してクリックしたか、またはツールバーメニューからの削除オプションの選択が、削除のために選択する DDL メニュー構造体のリストを提示する）と認証し得る。

【0076】

例えば、複数の DDL メニュー構造体のアイコンの周りにボックスを引き寄せるためにマウスを使用すること、及び上に記載されている方法のうちの 1 つによって選択された DDL 構造を削除することによって、複数の DDL メニュー構造体を一度に削除してもよいことが理解されるだろう。DDL メニュー構造体削除ルーチン 800 は、例えば、DDL メニュー構造体が削除されることを確認するようにユーザを促すこと、及び/またはさらなる DDL メニュー構造体の削除に対してユーザを促すことによって、ブロック 806 で、ユーザが DDL メニュー構造体の削除を終えたかどうかを認証し得る。ブロック 806 で判定されるように、DDL メニュー構造体の DDL グラフィカルユーザインターフェースからの削除をユーザが終えた場合、制御は、図 11 の編集ルーチン 500 に戻り得る。終えていない場合、DDL メニュー構造体削除ルーチン 800 が DDL グラフィカルユーザインターフェースから削除するための次の DDL メニュー構造体の選択を待つため、制御は、ブロック 802 に戻り得る。

【0077】

図 15 は、DDL メニュー構造体を DDL グラフィカルユーザインターフェースに追加するための図 13 のブロック 706 に示されている DDL メニュー構造体追加ルーチン 900 の一例である。DDL グラフィカルユーザインターフェースを元々作成及び構成するか、または予め編集するときに検索された装置に対する DD からの DDL メニュー構造体が、図 11 のブロック 506 ですでにエクスポートされた場合（例えば、DDL グラフィカルユーザインターフェース内の DDL メニュー構造体のみがエクスポートされた場合）、DD からの DDL メニュー構造体が、ブロック 902 でエクスポートされ得る。図 3 のブロック 110 及び図 11 のブロック 506 で DDL メニュー構造体をエクスポートすることと同様に、DDL メニュー構造体追加ルーチン 900 は、構成インターフェースを通してすべての DDL メニュー構造体をユーザに利用可能にするために、DD をスキャン及び分析し、その結果、DD 内の DDL メニュー構造体及び値が、ユーザに表示される。

【0078】

ブロック 902 で DDL メニュー構造体をエクスポートすると、ブロック 904 で、DDL メニュー構造体が、構成インターフェース内でユーザに提示される。インターフェースは、図 7 に示されているものと同じであってもよく、DDL 構造が、DDL メニュー構造体テンプレート 302 内に表示される。いったん DDL メニュー構造体がエクスポート及び提示されると、ブロック 906 で、DDL メニュー構造体を選択され得、ブロック 908 で、DDL グラフィカルユーザインターフェースに追加され得る。DDL メニュー構造体を DDL グラフィカルユーザインターフェースに追加するためのプロセスは、上に記載されている図 3 のブロック 116 のものと同じであってもよい。例えば、ユーザは、DDL メニュー構造体に対するグラフィカルなアイコンをメニュー構造体テンプレート 302 から選択し、構成テンプレート 304 内の DDL グラフィカルユーザインターフェースにアイコンをドラッグして、構成テンプレート 304 内の任意の所望の位置にアイコンをドロップし得る。DDL 条件はまた、ブロック 906 に対して構成され、ブロック 906 で利用され得、DDL メニュー構造体追加ルーチン 900 は、これも上に記載されているような条件を満たすために、ブロック 906 に戻るブロック 910 に制御を戻し得る。

【0079】

ブロック 910 で判定されるように、DDL メニュー構造体の DDL グラフィカルユーザインターフェースへの追加をユーザが終えた場合、制御は、図 11 の編集ルーチン 500 に戻り得る。終えていない場合、DDL メニュー構造体追加ルーチン 900 が、DDL グラフィカルユーザインターフェースに追加するための次の DDL メニュー構造体の選択を待つため、制御は、ブロック 906 に戻り得る。

【0080】

図 16 は、DDL メニュー構造体を DDL グラフィカルユーザインターフェース内で修正するための図 13 のブロック 708 で示されている DDL メニュー構造体修正ルーチン 1000 の一例である。ブロック 1002 で始まり、DDL グラフィカルユーザインターフェース内の DDL メニュー構造体を選択される。例えば、ユーザは、構成テンプレート 304 内に表示された DDL グラフィカルユーザインターフェース内の DDL メニュー構造体の 1 つ以上のグラフィカルなアイコンの周りのボックス上でクリックするか、またはドラッグし得る。次に、ユーザは、コンテキストメニューまたはツールバーを介して選択された DDL メニュー構造体の値を修正するためのオプションを選択し得る。代替的に、DDL メニュー構造体の単なる選択が、DDL メニュー構造体を修正するためのオプションの選択を構成し得る。

【0081】

ブロック 1004 で、DDL メニュー構造体に対する新規値が受信される。例えば、ユーザは、限定されないが、上及び下範囲制限、値、データの種類、名前等を含む、VARIABLE に対する新規値をエンターし得る。同様に、ユーザは、構成テンプレート 304 内の選択された DDL メニュー構造体に対するアイコンを所望の位置にドラッグすることによって、DDL グラフィカルユーザインターフェース内の DDL メニュー構造体の配置に対する新規値をエンターし得る。いったんブロック 1004 で、新規値がエンター及び受信されると、ブロック 1006 で、DDL メニュー構造体が更新され得る。ブロック 1008 で判定されるように、DDL グラフィカルユーザインターフェース内の DDL メニュー構造体の修正をユーザが終えた場合、制御は、図 11 の編集ルーチン 500 に戻り得る。終えていない場合、DDL メニュー構造体修正ルーチン 1000 が、DDL グラフィカルユーザインターフェース内で修正するための次の DDL メニュー構造体の選択を待つため、制御は、ブロック 1002 に戻り得る。

【0082】

図 11 を再び参照すると、DDL メニュー構造体が削除、追加、及び/または修正されたかどうか、もしくは新規装置からの DDL メニュー構造体がユーザの構成したメニューに追加されたかどうかに関わらず、制御は、編集ルーチン 500 に戻り得る。ブロック 516 で判定されるように、DDL グラフィカルユーザインターフェースの修正をユーザが終えた場合、ブロック 518 で、更新された DDL グラフィカルユーザインターフェースが保存され得る。終えていない場合、制御は、DDL グラフィカルユーザインターフェースへの次の修正のためのブロック 510 に戻り得る。更新された DDL グラフィカルユーザインターフェースがブロック 518 で保存されると、更新された DDL グラフィカルユーザインターフェースは、図 3 のブロック 120 のように DDL に基づくホスト 50 により保存される。具体的には、更新された DDL グラフィカルユーザインターフェースは、装置に格納されるのではなく、DDL に基づくホスト 50 等のユーザデータベース内に LOCAL 変数として格納された、選択された DDL メニュー構造体の値を含む FILE データとして保存される。再び、ユーザ選択を FILE データとして格納することによって、DDL グラフィカルユーザインターフェースは、既存のファームウェアを変更する必要なく作成及び格納され得、装置自体への追加の変更を必要としない。

【0083】

前述のテキストは、本発明の多数の異なる実施形態の詳細な説明を記載しているが、本発明の範囲は、本特許の最後に記載されている特許請求の範囲の語句によって定義されることを理解するべきである。詳細な説明は、例示的なもののみとして解釈されるべきであり、すべての可能な実施形態を記載することは、不可能でないとしても、実現困難である

10

20

30

40

50

ため、本発明のすべての可能な実施形態を記載するものではない。本発明を定義する特許請求の範囲の範囲内に未だ該当するであろう、現在の技術または当該特許の出願日以降に開発された技術のいずれかを用いて、多数の代替的な実施形態を実行することができるだろう。

【 0 0 8 4 】

D D L グラフィカルユーザインターフェース構成テンプレート及びその要素は、ワークステーションまたはサーバ上で実行され得るルーチンとして記載されているが、それらはまた、ハードウェア、ファームウェア等において実行されてもよく、複数のプロセッサを含む、任意の他のプロセッサによって実行されてもよい。したがって、本明細書に記載されている要素は、標準的な多目的 C P U において、または所望される場合、特定用途向け集積回路 (A S I C) または他の配線装置等の特別に設計されたハードウェアまたはファームウェア上で実行され得る。ソフトウェア内で実行されたとき、ソフトウェアルーチンは、磁気ディスク、レーザーディスク、または他の記憶媒体上等の任意のコンピュータ読み取り可能なメモリ内に、コンピュータまたはプロセッサの R A M または R O M 内に、任意のデータベース内等に格納され得る。

【 0 0 8 5 】

したがって、本発明の趣旨及び範囲から逸脱することなく、本明細書で記載及び図示されている技術及び構造において、多くの変形及び変更が行われてもよい。したがって、本明細書に記載されている方法及び装置は、図示のみのためであり、本発明の範囲を限定するものではないことを理解するべきである。

10

20

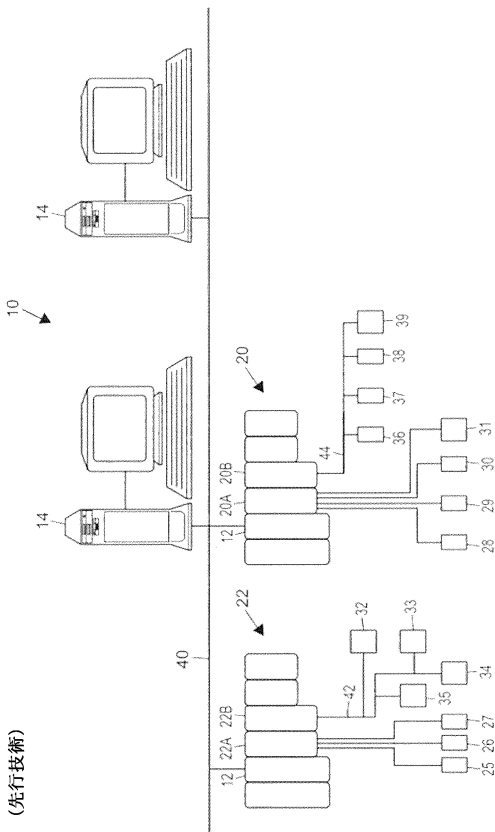
30

40

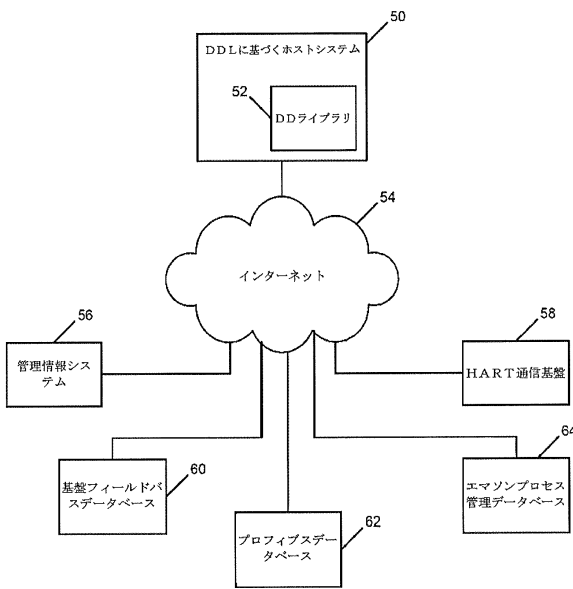
50

【図面】

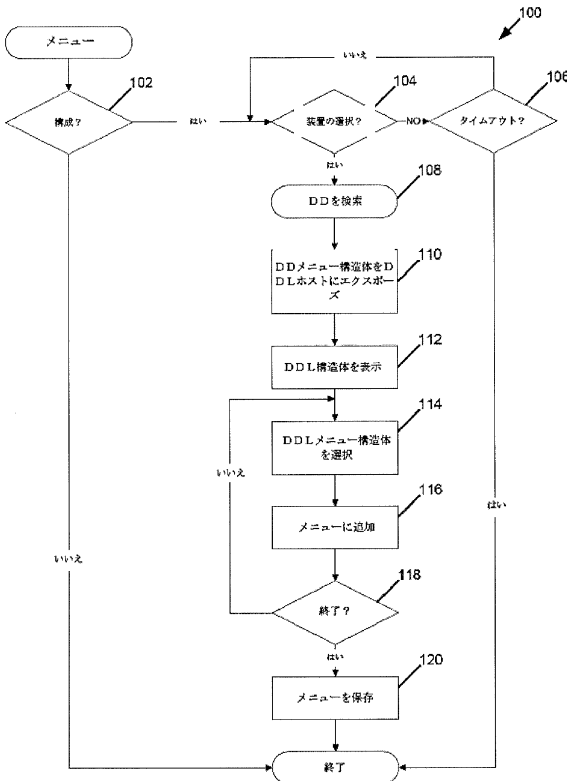
【図 1】



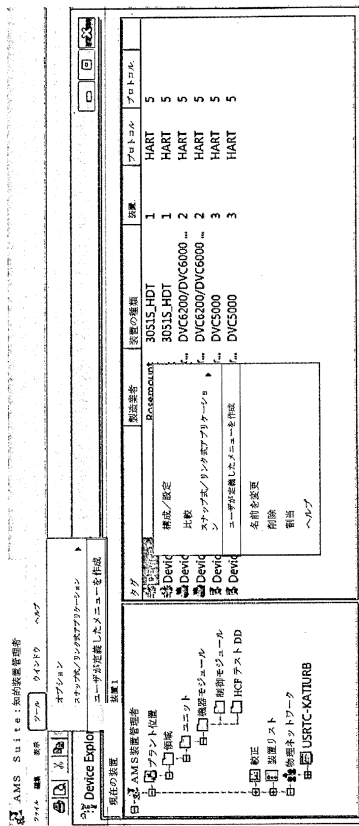
【図 2】



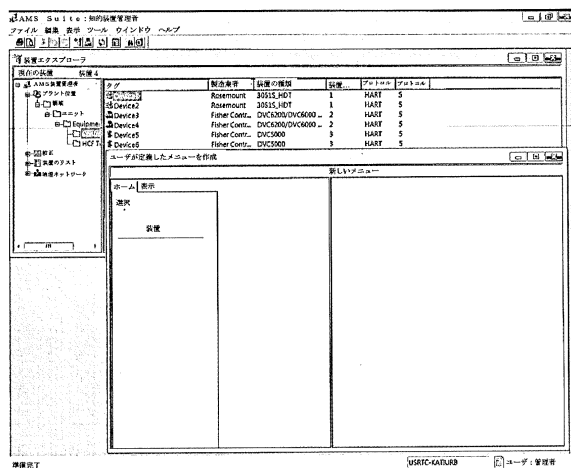
【図 3】



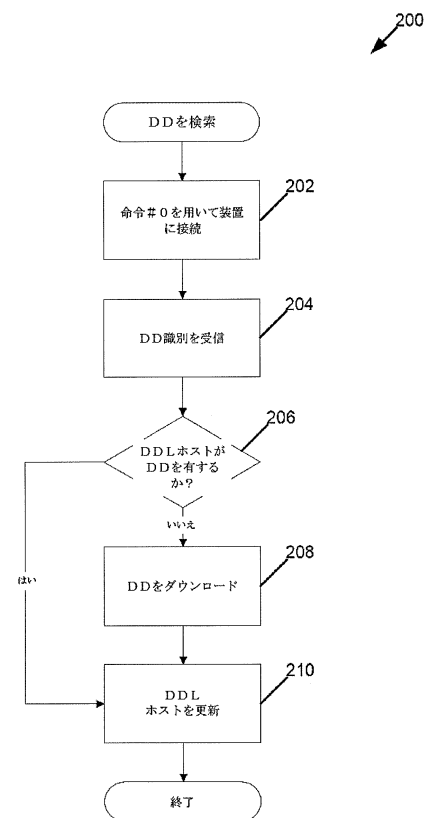
【図 4】



【 図 5 】



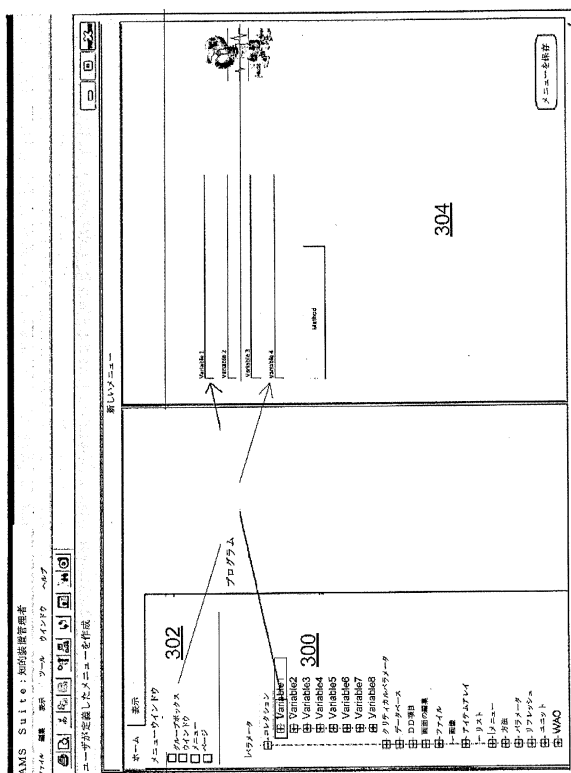
【 図 6 】



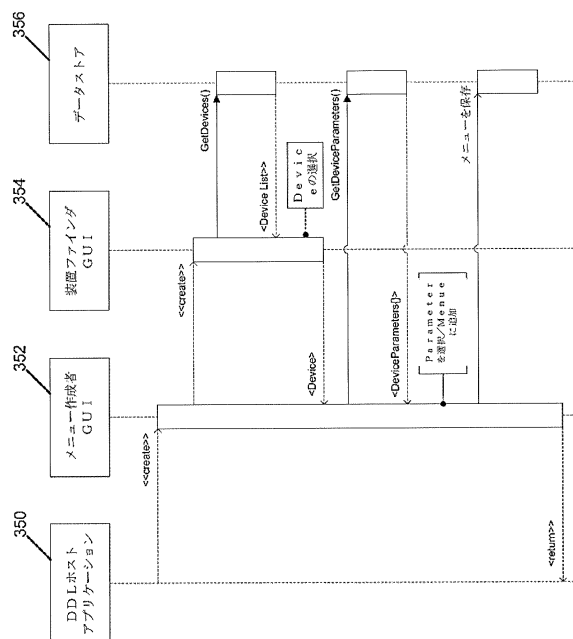
10

20

【圖 7】



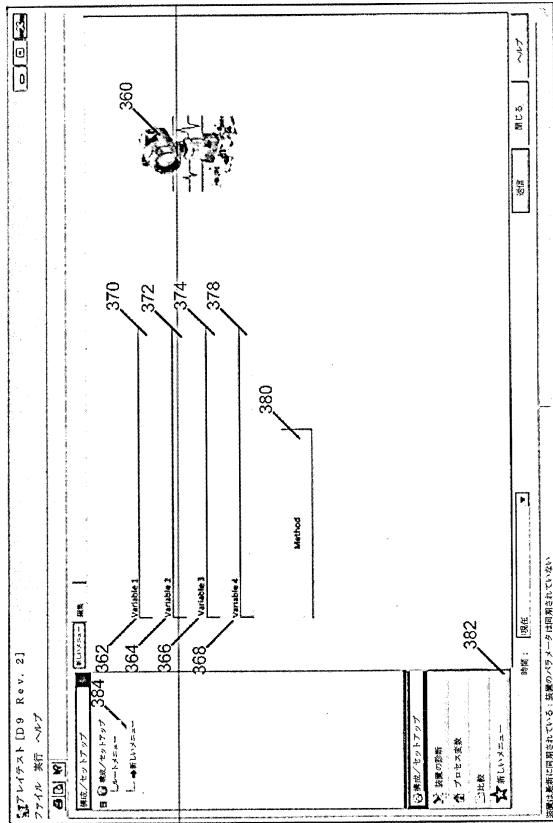
【图 8】



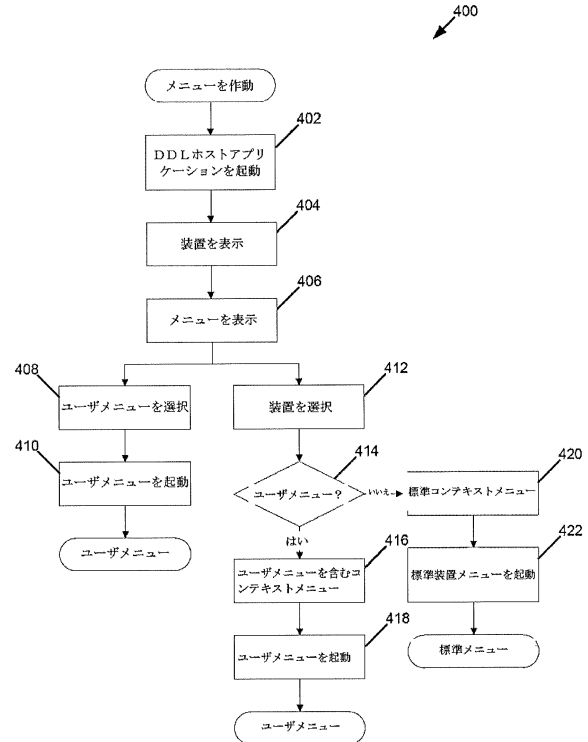
30

40

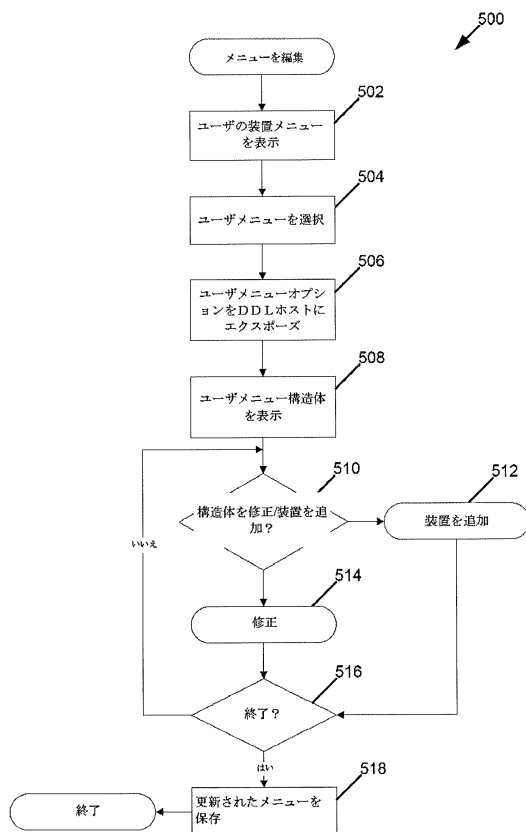
【図 9】



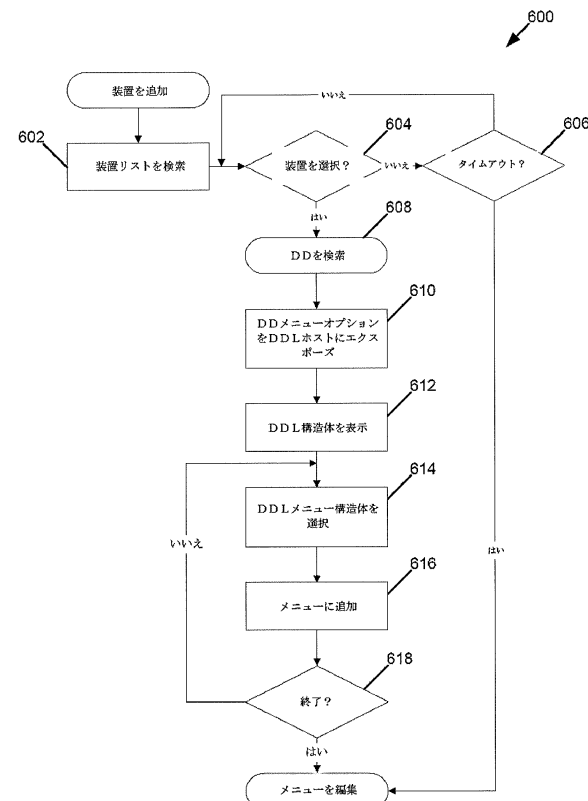
【図 10】



【図 11】



【図 12】



10

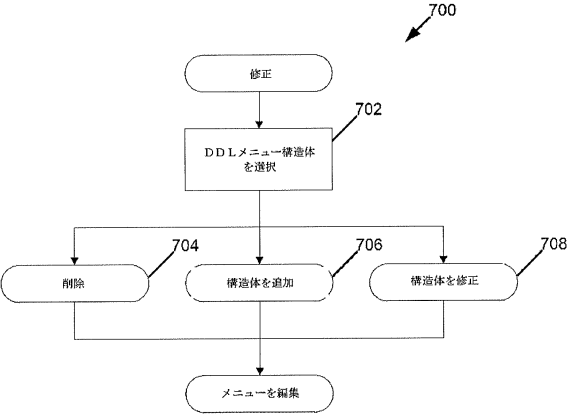
20

30

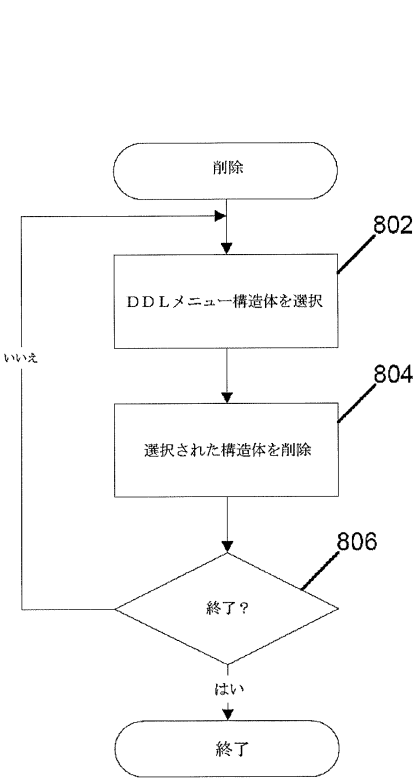
40

50

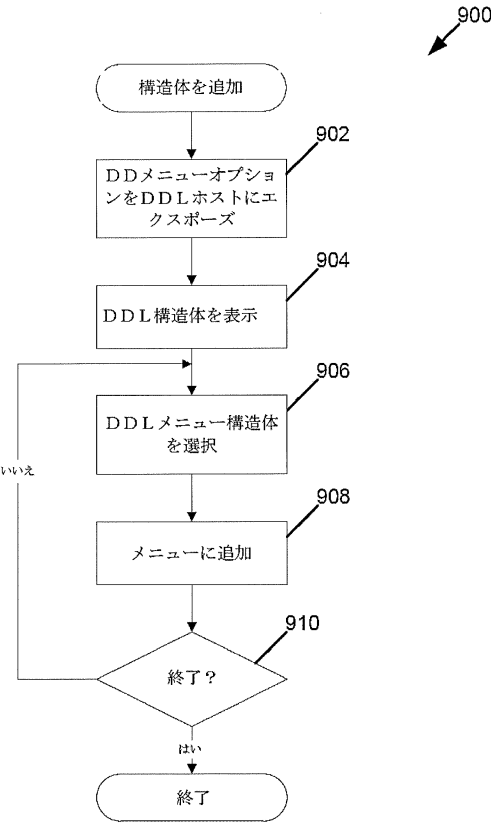
【図 13】



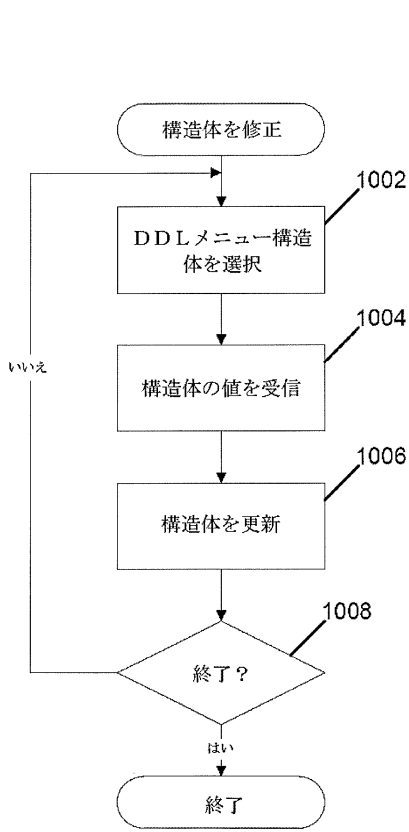
【図 14】



【図 15】



【図 16】



10

20

30

40

50

フロントページの続き

ノース 8 6 4

(72)発明者 ウォルター ヘンドリック サイターマンズ

アメリカ合衆国 ミネソタ 5 5 1 2 4 アップル バレー ガーナー レーン 1 2 9 7 5

(72)発明者 タン ガン トゥルン

アメリカ合衆国 ミネソタ 5 5 0 4 4 レイクビル インパラ パス 1 7 8 6 1

(72)発明者 ケイティ グラムズ フロスト

アメリカ合衆国 ミネソタ 5 5 4 4 6 プライマウス 5 4 ス アベニュー ノース 1 3 8 6 5

審査官 稲垣 浩司

(56)参考文献 特開 2 0 1 4 - 2 2 5 2 3 1 (J P , A)

(58)調査した分野 (Int.Cl., D B 名)

G 0 5 B 2 3 / 0 0 - 2 3 / 0 2