

(19) 日本国特許庁(JP)

## (12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-113564

(P2011-113564A)

(43) 公開日 平成23年6月9日(2011.6.9)

(51) Int.Cl.

**G05B 23/02** (2006.01)  
**G06F 3/048** (2006.01)

F 1

GO5B 23/02 301P  
 GO5B 23/02 V  
 GO6F 3/048 652A

テーマコード (参考)

5E501  
 5H223

審査請求 未請求 請求項の数 48 O L (全 39 頁)

(21) 出願番号 特願2010-259071 (P2010-259071)  
 (22) 出願日 平成22年11月19日 (2010.11.19)  
 (31) 優先権主張番号 12/623,502  
 (32) 優先日 平成21年11月23日 (2009.11.23)  
 (33) 優先権主張国 米国(US)

(71) 出願人 594120847  
 フィッシュヤーローズマウント システムズ、 インコーポレイテッド  
 アメリカ合衆国 78759 テキサス  
 オースティン リサーチ パーク ブラザ  
 ビルディング 111 リサーチ プル  
 バード 12301  
 (74) 代理人 100079049  
 弁理士 中島 淳  
 (74) 代理人 100084995  
 弁理士 加藤 和詳  
 (74) 代理人 100085279  
 弁理士 西元 勝一

最終頁に続く

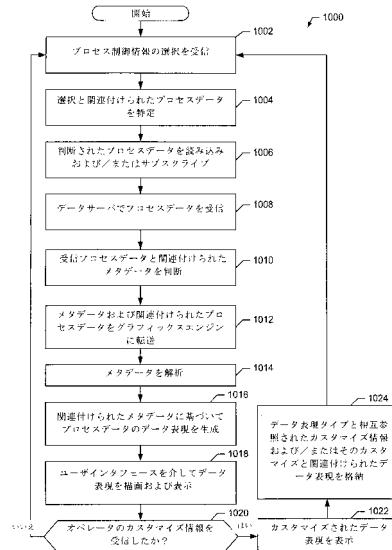
(54) 【発明の名称】プロセス制御システムと関連付けられたデータを動的に表示する方法および装置

## (57) 【要約】

【課題】本発明は、プロセス制御システムと関連付けられたデータを動的に表示する方法および装置を提供する。

【解決手段】本発明の例示的方法は、ユーザによって選択されたプロセス制御情報をグラフィックスエンジンで受信することと、選択されたプロセス制御情報と関連付けられたプロセスデータを特定することと、特定されたプロセスデータをプロセスコントローラから取得することと、取得されたプロセスデータと関連付けられたメタデータを判断することと、メタデータに基づいてグラフィックスエンジンでデータ表現を生成することと、ユーザインターフェースを介してデータ表現を表示することとを含む。

【選択図】図10



**【特許請求の範囲】****【請求項 1】**

プロセス制御システムと関連付けられたデータを動的に表示する方法であって、  
ユーザによって選択されたプロセス制御情報をグラフィックスエンジンで受信すること  
と、

前記選択されたプロセス制御情報と関連付けられたプロセスデータを特定することと、  
前記特定されたプロセスデータをプロセスコントローラから取得することと、  
前記取得されたプロセスデータと関連付けられたメタデータを判断することと、  
前記メタデータに基づいて前記グラフィックスエンジンでデータ表現を生成することと  
、

10

ユーザインターフェースを介して前記データ表現を表示することと、  
を含む方法。

**【請求項 2】**

前記プロセスデータまたは前記メタデータの少なくとも 1 つに対する変更を受信  
することと、

前記変更を受信したことに応じて、前記変更されたプロセスデータまたは前記変更され  
たメタデータの少なくとも 1 つに基づいて前記データ表現を修正することと、

前記ユーザインターフェースを介して前記修正されたデータ表現を表示することと、  
をさらに含む、請求項 1 に記載の方法。

**【請求項 3】**

前記データ表現を修正することが、前記データ表現に第 2 のデータ表現を追加すること  
、前記データ表現にパラメータを追加すること、前記データ表現からパラメータを削除す  
ること、前記データ表現のパラメータの種類を修正すること、または前記データ表現の視  
覚化を修正することのうちの少なくとも 1 つを含む、請求項 2 に記載の方法。

20

**【請求項 4】**

前記データ表現を生成することが、  
前記グラフィックスエンジンで前記メタデータのタイプを特定することと、  
前記特定されたメタデータのタイプと、データベースにある少なくとも 1 つのデータ表  
現タイプとを相互参照することと、

前記特定されたメタデータのタイプと関連付けられた前記少なくとも 1 つのデータ表現  
タイプのインスタンスを生成することと、

前記メタデータを前記少なくとも 1 つのデータ表現タイプの前記インスタンスにあるデ  
ータフィールドに関連付けることと、

前記メタデータと関連付けられた前記プロセスデータを前記少なくとも 1 つのデータ表  
現タイプのインスタンスにある前記データフィールドと関連付けて、前記データ表現を生  
成することと、

を含む、請求項 1 に記載の方法。

30

**【請求項 5】**

データ表現タイプを選択した後に、前記特定されたメタデータのタイプと前記少なく  
とも 1 つのデータ表現タイプとを相互参照するよう前記ユーザに促すことをさらに含む、請  
求項 4 に記載の方法。

40

**【請求項 6】**

前記データ表現を描画した後に前記少なくとも 1 つのデータ表現を表示することをさら  
に含む、請求項 1 に記載の方法。

**【請求項 7】**

前記メタデータを判断することが、前記プロセスデータとメタデータタイプとをデータ  
ベースで相互参照することを含む、請求項 1 に記載の方法。

**【請求項 8】**

前記メタデータを判断することが、前記プロセスデータと関連付けられたメタデータを  
特定することを含む、請求項 1 に記載の方法。

50

**【請求項 9】**

前記選択されたプロセス制御情報が、バッチ識別名、プロセス識別名、コントローラ識別名、現場装置識別名、前記プロセスデータと関連付けられたファイル、前記プロセスデータへのリンク、1つ以上の機能ロック、1つ以上のパラメータ、1つ以上のプロセス制御エンドポイント、またはプロセス制御ルーチンのうちの少なくとも1つによって特定される、請求項1に記載の方法。

**【請求項 10】**

前記プロセスデータを取得することが、前記プロセスデータをサブスクライブすることまたは読み込むことのうちの少なくとも1つを含む、請求項1に記載の方法。

**【請求項 11】**

データ識別名によって前記メタデータを解析した後に前記データ表現を生成することをさらに含む、請求項1に記載の方法。

**【請求項 12】**

前記データ表現と関連付けられたカスタマイズ情報を受信することと、  
前記データ表現と前記カスタマイズ情報を参照することと、  
前記カスタマイズ情報に基づいて前記データ表現を修正することと、  
前記ユーザインタフェースを介して前記修正されたデータ表現を表示することと、  
前記データ表現に対する前記カスタマイズ情報の参照をデータベースに格納することと  
、  
をさらに含む、請求項1に記載の方法。

**【請求項 13】**

前記カスタマイズ情報が、1つ以上の前記データ表現の表示の修正、1つ以上の前記データ表現の色の変更、1つ以上の前記データ表現のレイアウトの変更、1つ以上の前記データ表現と関連付けられたグラフィックスの変更、1つ以上の前記データ表現の表示形式の変更、または1つ以上の前記データ表現と関連付けられた識別情報の変更のうちの少なくとも1つを含む、請求項12に記載の方法。

**【請求項 14】**

前記メタデータに基づいて前記グラフィックスエンジンで第2のデータ表現を生成することと、

前記ユーザインタフェースを介して前記第2のデータ表現を表示することと、  
をさらに含む、請求項1に記載の方法。

**【請求項 15】**

プロセス制御システムと関連付けられたデータを動的に表示する装置であって、  
選択されたプロセス制御情報と関連付けられたプロセスデータを特定するためのデータマネージャと、  
前記特定されたプロセスデータと関連付けられたメタデータを判断するためのメタデータジェネレータと、  
前記メタデータに基づいてデータ表現を生成するためのデータ表現ジェネレータと、  
ユーザインタフェースを介して前記データ表現を表示するためのデータ表現表示マネージャと、

を含む装置。

**【請求項 16】**

ユーザによって選択された前記プロセス制御情報を受信するための選択レシーバと、  
前記特定されたプロセスデータをプロセスコントローラから取得し、前記データマネージャに前記特定されたプロセスデータを転送するためのデータリトリーバと、  
をさらに含む、請求項15に記載の装置。

**【請求項 17】**

前記データマネージャ、前記選択レシーバ、前記データ表現表示マネージャ、または前記データ表現ジェネレータのうちの少なくとも1つがグラフィックスエンジン内に含まれる、請求項16に記載の装置。

10

20

30

40

50

**【請求項 18】**

前記データリトリーバまたは前記メタデータジェネレータのうちの少なくとも1つがデータサーバ内に含まれる、請求項16に記載の装置。

**【請求項 19】**

前記データサーバが前記グラフィックスエンジン内に含まれる、請求項18に記載の装置。

**【請求項 20】**

前記データリトリーバが前記プロセスデータをサブスクライブするか読み込むかのうちの少なくとも1つによって前記プロセスデータを取得することになっている、請求項16に記載の装置。

**【請求項 21】**

前記選択レシーバが、前記プロセスデータまたは前記メタデータのうちの少なくとも1つに対する変更を受信することになっている。

前記データ表現ジェネレータが、前記変更されたプロセスデータまたは前記変更されたメタデータのうちの前記少なくとも1つに基づいて前記データ表現を修正することになっている。

前記データ表現表示マネージャが、前記ユーザインターフェースを介して前記修正されたデータ表現を表示することになっている。

請求項16に記載の装置。

**【請求項 22】**

前記データ表現を変更することが、前記データ表現に第2のデータ表現を追加すること、前記データ表現にパラメータを追加すること、前記データ表現からパラメータを削除すること、前記データ表現のパラメータの種類を修正すること、または前記データ表現の視覚化を修正することのうちの少なくとも1つ含む、請求項21に記載の装置。

**【請求項 23】**

前記データ表現ジェネレータが、  
前記メタデータのタイプを特定することと、  
前記特定されたメタデータのタイプと少なくとも1つのデータ表現タイプとをメタデータルールデータベースで相互参照することと、

前記特定されたメタデータのタイプと関連付けられた前記少なくとも1つのデータ表現タイプのインスタンスを生成することと、

前記メタデータを前記少なくとも1つのデータ表現タイプの前記インスタンスにあるデータフィールドと関連付けることと、

前記メタデータと関連付けられた前記プロセスデータを前記少なくとも1つのデータ表現タイプのインスタンスにある前記データフィールドと関連付けて前記データ表現を作成することと、

によって前記データ表現を生成することになっている、請求項15に記載の装置。

**【請求項 24】**

前記データ表現ジェネレータが、データ表現タイプを選択した後に前記特定されたメタデータのタイプと前記少なくとも1つのデータ表現タイプとを相互参照するようユーザに促すプロンプトを送信することになっている、請求項23に記載の装置。

**【請求項 25】**

前記データ表現表示マネージャが、前記データ表現を描画した後に前記少なくとも1つのデータ表現を表示するようになっている、請求項15に記載の装置。

**【請求項 26】**

前記メタデータジェネレータが、前記プロセスデータとメタデータタイプとをメタデータのタイプデータベースで相互参照することによって前記メタデータを判断することになっている、請求項15に記載の装置。

**【請求項 27】**

前記メタデータジェネレータが、前記プロセスデータと関連付けられたメタデータを特

10

20

30

40

50

定することによって前記メタデータを判断することになっている、請求項 15 に記載の装置。

**【請求項 28】**

前記データマネージャが、バッチ識別名、プロセス識別名、コントローラ識別名、現場装置識別名、前記プロセスデータと関連付けられたファイル、前記プロセスデータへのリンク、1つ以上の機能ロック、1つ以上のパラメータ、1つ以上のプロセス制御エンドポイント、またはプロセス制御ルーチンのうちの少なくとも1つによって前記選択されたプロセス制御情報を特定することになっている、請求項 15 に記載の装置。

**【請求項 29】**

前記データ表現ジェネレータが前記データ表現を生成する前にデータ識別名によって前記メタデータを解析するメタデータパーサをさらに含む、請求項 15 に記載の装置。 10

**【請求項 30】**

前記データ表現表示マネージャが、前記データ表現と関連付けられたカスタマイズ情報を受信することになっており、前記データ表現ジェネレータが、

前記データ表現に対して前記カスタマイズ情報を照会することになっており、

前記カスタマイズ情報に基づいて前記データ表現を修正することになっており、

前記データに対する前記カスタマイズ情報の照会をデータ表現データベースに格納することになっている、

請求項 15 に記載の装置。

**【請求項 31】**

前記データ表現表示マネージャが、前記ユーザインターフェースを介して前記修正されたデータ表現を表示することになっている、請求項 30 に記載の装置。 20

**【請求項 32】**

前記データ表現表示マネージャが、前記ユーザインターフェースを介して前記データ表現を表示することになっている提示プロセッサに前記データ表現を転送することによって前記データ表現を表示することになっている、請求項 15 に記載の装置。

**【請求項 33】**

命令を格納している機械アクセス可能なメディアであって、実行されると、マシンが、ユーザによって選択されたプロセス制御情報を受信し、

前記選択されたプロセス制御情報と関連付けられたプロセスデータを特定し、 30

前記特定されたプロセスデータをプロセスコントローラから取得し、

前記取得されたプロセスデータと関連付けられたメタデータを判断し、

前記メタデータに基づいてデータ表現を生成し、

ユーザインターフェースを介して前記データ表現を表示する、

機械アクセス可能なメディア。

**【請求項 34】**

前記命令が実行されると、前記マシンが、

前記プロセスデータまたは前記メタデータのうちの少なくとも1つに対する変更を受信し、

前記変更を受信したことに対応して、前記変更されたプロセスデータまたは前記変更されたメタデータの少なくとも1つに基づいて前記データ表現を修正し、 40

前記ユーザインターフェースを介して前記修正されたデータ表現を表示する、

請求項 33 に記載の機械アクセス可能なメディア。

**【請求項 35】**

前記命令が実行されると、前記マシンが、前記データ表現に第2のデータ表現を追加すること、前記データ表現にパラメータを追加すること、前記データ表現からパラメータを削除すること、前記データ表現のパラメータの種類を修正すること、または前記データ表現の視覚化を修正することのうちの少なくとも1つによって前記データ表現を修正する、請求項 34 に記載の機械アクセス可能なメディア。

**【請求項 36】**

10

20

30

40

50

前記命令が実行されると、前記マシンが、  
前記メタデータのタイプを特定し、  
前記特定されたメタデータのタイプと少なくとも1つのデータ表現タイプとをデータベースで相互参照し、  
前記特定されたメタデータのタイプと関連付けられた前記少なくとも1つのデータ表現タイプのインスタンスを生成し、  
前記メタデータを前記少なくとも1つのデータ表現タイプの前記インスタンスにあるデータフィールドに関連付け、  
前記メタデータと関連付けられた前記プロセスデータを前記少なくとも1つのデータ表現タイプのインスタンスにある前記データフィールドと関連付けて、前記データ表現を生成する、  
請求項33に記載の機械アクセス可能なメディア。

【請求項37】

前記命令が実行されると、前記マシンが、データ表現タイプを選択した後に前記特定されたメタデータのタイプと前記少なくとも1つのデータ表現タイプとを相互参照するよう前記ユーザに促す、請求項36に記載の機械アクセス可能なメディア。

【請求項38】

前記命令が実行されると、前記マシンが、前記データ表現を描画した後に前記少なくとも1つのデータ表現を表示する、請求項33に記載の機械アクセス可能なメディア。

【請求項39】

前記命令が実行されると、前記マシンが、前記プロセスデータとメタデータタイプとを相互参照することによって前記メタデータを判断する、請求項33に記載の機械アクセス可能なメディア。

【請求項40】

前記命令が実行されると、前記マシンが、前記プロセスデータと関連付けられたメタデータを特定することによって前記メタデータを判断する、請求項33に記載の機械アクセス可能なメディア。

【請求項41】

前記命令が実行されると、前記マシンが、前記プロセスデータをサブスクライブするか読み込むかの少なくとも1つによって前記プロセスデータを取得する、請求項33に記載の機械アクセス可能なメディア。

【請求項42】

前記命令が実行されると、前記マシンが、データ識別名によって前記メタデータを解析した後に前記データ表現を生成する、請求項33に記載の機械アクセス可能なメディア。

【請求項43】

前記命令が実行されると、前記マシンが、  
前記データ表現と関連付けられたカスタマイズ情報を受信し、  
前記データ表現に対して前記カスタマイズ情報を照会し、  
前記カスタマイズ情報を基づいて前記データ表現を修正し、  
前記ユーザインターフェースを介して前記修正されたデータ表現を表示し、  
前記データ表現に対する前記カスタマイズ情報の照会をデータベースに格納する、  
請求項33に記載の機械アクセス可能なメディア。

【請求項44】

前記命令が実行されると、前記マシンが、  
前記メタデータに基づいて第2のデータ表現を生成し、  
前記ユーザインターフェースを介して前記第2のデータ表現を表示する、  
請求項33に記載の機械アクセス可能なメディア。

【請求項45】

プロセス制御システムと関連付けられたデータを動的に表示する装置であって、  
稼働すると、前記装置が、

10

20

30

40

50

ユーザによって選択されたプロセス制御情報を受信し、  
 前記選択されたプロセス制御情報と関連付けられたプロセスデータを特定し、  
 前記特定されたプロセスデータをプロセスコントローラから取得し、  
 前記取得されたプロセスデータと関連付けられたメタデータを判断し、  
 前記メタデータに基づいてデータ表現を生成し、  
 ユーザインターフェースを介して前記データ表現を表示する、  
 1つ以上のプロセッサを含む装置。

**【請求項 4 6】**

前記1つ以上の前記プロセッサが稼働すると、前記装置が、  
 前記プロセスデータまたは前記メタデータの少なくとも1つに対する変更を受信し

10

、  
 前記変更を受信したことに応じて、前記変更されたプロセスデータまたは前記変更されたメタデータの少なくとも1つに基づいて前記データ表現を修正し、  
 前記ユーザインターフェースを介して前記修正されたデータ表現を表示する、  
 請求項45に記載の装置。

**【請求項 4 7】**

前記1つ以上の前記プロセッサが稼働すると、前記装置が、  
 前記メタデータのタイプを特定し、  
 前記特定されたメタデータのタイプと少なくとも1つのデータ表現タイプとをデータベース内で相互参照し、  
 前記特定されたメタデータのタイプと関連付けられた前記少なくとも1つのデータ表現タイプのインスタンスを生成し、  
 前記メタデータを前記少なくとも1つのデータ表現タイプの前記インスタンスにあるデータフィールドに関連付け、  
 前記メタデータと関連付けられた前記プロセスデータを前記少なくとも1つのデータ表現タイプのインスタンスにある前記データフィールドと関連付けて、前記データ表現を生成する、  
 請求項45に記載の装置。

20

**【請求項 4 8】**

前記1つ以上のプロセッサが稼働すると、前記装置が、  
 前記データ表現と関連付けられたカスタマイズ情報を受信し、  
 前記データ表現と前記カスタマイズ情報を参照し、  
 前記カスタマイズ情報に基づいて前記データ表現を修正し、  
 前記ユーザインターフェースを介して前記修正されたデータ表現を表示し、  
 前記データ表現に対する前記カスタマイズ情報の照会をデータベースに格納する、  
 請求項45に記載の装置。

30

**【発明の詳細な説明】**

**【技術分野】**

**【0001】**

本開示は、一般にプロセス制御システムに関し、特に、プロセス制御システムと関連付けられたデータを動的に表示するための方法および装置に関する。

40

**【背景技術】**

**【0002】**

化学、石油、または他のプロセスなどで使用されるようなプロセス制御システムは、アナログ、デジタル、またはアナログ/デジタル複合バス経由で少なくとも1つのホストまたはオペレーターワークステーションと1つ以上の現場装置とに通信可能に連結された1つ以上のプロセスコントローラおよび入出力(I/O)装置を概して含む。例えば、バルブ、バルブポジショナ、スイッチ、および送信器(温度、圧力、および流量センサなど)といった現場装置は、バルブの開閉やプロセス制御パラメータの測定など、プロセス内のプロセス制御機能を実行する。プロセスコントローラは、現場装置によって計測されたプロ

50

セス測定値を表す信号を受信し、この情報を処理して制御ルーチンを実装した後、バスまたは他の通信回線を通じて現場装置に送られる制御信号を生成して、このプロセスの動作を制御する。このようにして、プロセスコントローラは、現場装置を使用して、現場装置を通信可能に連結しているバスおよび／または他の通信リンク経由で制御ストラテジまたはルーチンを実行および調整することができる。

#### 【0003】

現場装置およびコントローラからの情報は、プロセスの現状の参照（グラフィカルユーザインターフェース経由などで）、プロセスの評価、あるいはプロセスの動作の修正（視覚的なオブジェクトダイアグラム経由などで）など、プロセスに関する所望の機能をオペレータが実行できるように、オペレータワークステーション（プロセッサベースのシステムなど）によって実行される1つ以上のアプリケーション（すなわち、ソフトウェアルーチン、プログラムなど）で利用できるようにしてもよい。また、多数のプロセス制御システムは、1つ以上のアプリケーションステーションを含む。一般に、これらのアプリケーションステーションは、ローカルエリアネットワーク（LAN）経由でプロセス制御システム内のコントローラ、オペレータワークステーション、および他のシステムと通信可能に連結されているパーソナルコンピュータやワークステーションなどを使用して実装される。各アプリケーションステーションは、プロセス制御システム内でキャンペーン管理機能、保守管理機能、仮想制御機能、診断機能、リアルタイム監視機能、安全関連機能、構成機能などを実行する1つ以上のストラテジ、ルーチン、またはアプリケーションを実行し得る。

10

20

30

#### 【0004】

加えて、プロセス制御システムは、プロセス制御ルーチン、現場装置、コントローラ、および／または通信を監視するための監視情報、診断情報、および／またはアラームを表すデータ表現を概して含む。データ表現は、図、グラフ、データテーブル、リストボックス、グラフィカル記号、テキストなどの形態でプロセスデータをグラフィカルに表示することにより、オペレータを支援する。現在は、処理領域、一群の現場装置、プロセスの一部分、データのタイプ、および／またはプロセス制御コンポーネントに固有である可能性のあるユーザインターフェースでのデータ表現の表示を、プロセスのオペレータが手動で構成しなくてはならない場合がある。しかし、ユーザインターフェースがオペレータ、プロセス制御システム、プロセス制御領域、および／または一群の現場装置ごとに構成されなくてはならない場合があるため、このような手動構成は、プロセス制御オペレータにとって大きな負担となり得る。また、手動構成は、データ表現が重複する複数のユーザインターフェース画面を作成することに至る可能性がある。さらには、プロセス制御システムが修正されたときに、場合によっては、データ表現に対応するユーザインターフェースを変更して、その修正を反映する必要がある。

40

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0005】

本発明は、プロセス制御システムと関連付けられたデータを動的に表示する方法および装置を提供しようとするものである。

40

#### 【課題を解決するための手段】

#### 【0006】

プロセス制御システムと関連付けられたデータを動的に表示する例示的方法および装置が記載される。一実施例において、方法は、ユーザによって選択されたプロセス制御情報グラフィックスエンジンで受信することと、選択されたプロセス制御情報と関連付けられたプロセスデータを特定することと、特定されたプロセスデータをプロセスコントローラから取得することとを含む。前記例示的方法は、取得されたプロセスデータと関連付けられたメタデータを判断することと、そのメタデータに基づいてグラフィックスエンジンでデータ表現を生成することと、ユーザインターフェースを介してデータ表現を表示することとをさらに含む。

50

## 【0007】

例示的装置は、選択されたプロセス制御情報と関連付けられたプロセスデータを特定するためのデータマネージャと、取得されたプロセスデータと関連付けられたメタデータを判断するためのメタデータジェネレータとを含む。前記例示的装置は、メタデータに基づいてデータ表現を生成するためのデータ表現ジェネレータと、ユーザインターフェースを通してデータ表現を表示するためのデータ表現表示マネージャも含む。

## 【図面の簡単な説明】

## 【0008】

【図1】グラフィックスエンジン例およびデータサーバ例を含むプロセス制御システム例を図示するブロック図を示す。  
10

【図2】図1のグラフィックスエンジン例およびデータサーバ例の機能ブロック図を示す。

【図3】バッチベースのプロセス制御システムのデータ表現例を示す。

【図4】追加プロセスデータに関する修正を含む図3のデータ表現例を示す。

【図5】プロセス制御オペレータによって指定されたカスタマイズを含む図4のデータ表現例を示す。  
20

【図6】図3から図5のバッチと関連付けられた説明変数の寄与度のデータ表現例を示す。

【図7】図3から図5のバッチと関連付けられた被説明変数の寄与度が追加された図6のデータ表現例を示す。

【図8】プロセス制御システムの一部分のステータスのデータ表現例を示す。

【図9】ステータスチャートデータ表現が追加された図8のデータ表現例を示す。

【図10】図1および/または2のグラフィックスエンジン例、データサーバ例、選択レシーバ例、データマネージャ例、メタデータパーサ例、データ表現ジェネレータ例、データ表現表示マネージャ例、データリトリーバ例、および/またはメタデータジェネレータ例を実装するために使用され得る方法例のフローチャートである。  
30

【図11A】図1および/または2のグラフィックスエンジン例、データサーバ例、選択レシーバ例、データマネージャ例、メタデータパーサ例、データ表現ジェネレータ例、データ表現表示マネージャ例、データリトリーバ例、および/またはメタデータジェネレータ例を実装するために使用され得る方法例のフローチャートである。

【図11B】図1および/または2のグラフィックスエンジン例、データサーバ例、選択レシーバ例、データマネージャ例、メタデータパーサ例、データ表現ジェネレータ例、データ表現表示マネージャ例、データリトリーバ例、および/またはメタデータジェネレータ例を実装するために使用され得る方法例のフローチャートである。  
40

【図12】図1および/または2のグラフィックスエンジン例、データサーバ例、選択レシーバ例、データマネージャ例、メタデータパーサ例、データ表現ジェネレータ例、データ表現表示マネージャ例、データリトリーバ例、および/またはメタデータジェネレータ例を実装するために使用され得る方法例のフローチャートである。

【図13】本明細書に記載の方法例および装置例を実装する目的で使用され得るプロセッサシステム例のブロック図である。

## 【発明を実施するための形態】

## 【0009】

以下、コンポーネントの中でも特に、ハードウェア上で実行されるソフトウェアおよび/またはファームウェアを含む方法例および装置例について説明しているが、これらの例は単なる例に過ぎず、制限的なものとみなさるべきでないことに留意されたい。例えば、ハードウェア、ソフトウェア、およびファームウェアコンポーネントの一部または全部が、ハードウェアでのみ、ソフトウェアでのみ、あるいはハードウェアとソフトウェアとの任意の組み合わせで具現化され得ると考えられる。そのため、以下に方法例および装置例を記載するものの、当業者であれば、提供された実施例がかかる方法と装置を実装する唯一の方法でないことは容易に理解できるであろう。例えば、方法例および装置例は、  
50

口セス制御システムと関連付けられたデータを動的に表示することと関連付けて記載されているものの、この方法例および装置例は、より一般的に適用可能であり、任意の自動化システム、バッチ処理システム、製造システム、産業制御システム、安全装備されたシステムなどと関連付けられたデータを動的に表示するように実装され得る。

#### 【0010】

概して、プロセス制御システムは、プロセスの状況をオペレータに知らせるためのプロセスデータ、アラーム、監視情報、および／または診断情報などの情報の管理および表示を含む。プロセス制御システムは、プロセス制御ルーチン、現場装置、コントローラ、および／または通信を監視するためのセンサ、モニタ、および／または診断装置を含み得る。さらに、プロセス制御システムと関連付けられた情報が、オペレータがユーザインタフェースを介してワークステーションで参照できるグラフィカルなデータ表現として表示されてもよい。データ表現は、現場装置と関連付けられたプロセスデータ、現場装置間の相互接続、現場装置間の通信、現場装置の出力経路、現場装置の機能状態、および／またはプロセス制御システムと関連付けられている可能性のあるその他任意の情報も含む場合がある。加えて、データ表現は、プロセス制御システム全体と、あるいはプロセス制御システムの1つ以上の部分と関連付けられていてもよい。

#### 【0011】

現在、データ表現は、プロセス制御システム内の既知のコンポーネントおよび／または接続、プロセス制御システムと関連付けられた機能ブロック図、および／またはプロセス制御システムのその他任意の表現に基づき、オペレータおよび／またはプロセス制御エンジニアによって手動で構成される。データ表現は、図、グラフ、データテーブル、リストボックス、グラフィカル記号、テキストなどの形態でプロセスデータをグラフィカルに表示することにより、オペレータを支援する。ただし、一般にデータ表現は、エンジニアおよび／またはオペレータによって指定された表示画面に対して固定または静的である。例えば、グラフデータ表現は、プロセスにおける現場装置のステータスを示す目的で使用され得る。グラフデータ表現は、オペレータ、デザイナー、および／またはエンジニアによって現場装置からのプロセスデータと手動でリンクされる1つ以上のデータフィールドを含み得る。グラフデータ表現は、データフィールドで参照されるプロセスデータを読み込み、プロセスデータを表示し得る。ただし、プロセスデータをデータフィールドとこのように手動でリンクすると、静的または固定データ表現が作成される。第2の現場装置がプロセスに追加されると、オペレータおよび／またはエンジニアが第2の現場装置からのプロセスデータをグラフデータ表現と手動でリンクしなくてはならない場合がある。

#### 【0012】

場合によっては、1つ以上のデータ表現を表示しているユーザインタフェースが複数のプロセスデータソースに対して構成される必要があることから、このような手動構成は、プロセス制御オペレータおよび／またはエンジニアにとって大きな負担となる場合がある。また、このような手動構成により、ユーザインタフェース画面および／またはデータ表現で情報が重複する場合もある。さらに、プロセス制御システムが修正されたときに、対応するユーザインタフェースおよび／またはデータ表現を修正して変更を反映しなくてはならない場合もある。

#### 【0013】

本明細書に記載の方法例および装置例では、オペレータおよび／またはエンジニアによって選択されたプロセスデータの種類（プロセス制御データなど）に基づいてユーザインタフェースを介して表示され得るデータ表現が動的に作成される。本明細書に記載の方法例および装置例では、オペレータおよび／またはエンジニア（ユーザなど）からプロセス制御情報の選択を受信し、選択されたプロセス制御情報と関連付けられているプロセスデータを特定することにより、データ表現が動的に作成される。プロセス制御情報は、プロセスデータを生成、処理、および／または管理するプロセス制御コンポーネントのいかなる識別名も含み得る。例えば、プロセス制御情報は、バッチ識別名、プロセス識別名、コントローラ識別名、現場装置識別名、プロセスデータと関連付けられたファイル、プロセ

10

20

30

40

50

データへのリンク、1つ以上の機能ブロック、1つ以上のパラメータ、1つ以上のプロセス制御エンドポイント、および／またはプロセス制御ルーチンを含み得る。プロセス制御コンポーネントは、現場装置、プロセスコントローラ、ルーチン、アルゴリズム、I／Oカード、および／またはデータを生成および／または処理できるその他任意のプロセスコントローラを含み得る。

【0014】

選択されたプロセスデータを特定すると、本明細書に記載の方法例および装置例は、そのプロセスデータと関連付けられているメタデータを特定および／または作成する。メタデータは、プロセスデータと関連付けられた提示情報を指定することによってそのプロセスデータを説明する。一般に、提示情報は、例えば、プロセスデータのタイプ、プロセスデータの表示設定、プロセスデータの形式、プロセスデータの長さおよび／またはサイズ、プロセスデータの位置、および／またはプロセスデータに関するその他任意の定義または説明を含み得る。本明細書に記載の方法例および装置例は、メタデータを使用し、メタデータルールデータベースで指定された対応するデータ表現タイプとそのメタデータとを相互参照することにより、適切なデータ表現タイプを判断する。この方法例および装置例は、判断されたデータ表現タイプのインスタンスを作成し、新たに作成されたデータ表現インスタンスとプロセスデータを関連付ける。この方法例および装置例は、プロセスデータの位置をデータ表現インスタンス内に含まれる対応データフィールドにリンクおよび／または挿入することによってプロセスデータをデータ表現インスタンスと関連付ける場合もある。このデータ表現は、プロセスデータをデータ表現タイプのインスタンスと関連付けることによってインスタンスから生成される。その後、本明細書に記載の方法例および装置例は、ユーザインターフェースを介してデータ表現を描写および表示し得る。

10

20

30

40

【0015】

いくつかの実施例においては、プロセスデータがメタデータを含み、かつ／またはメタデータと関連付けられている場合がある。この含まれているメタデータは、プロセスデータを生成する現場装置、プロセスデータをコンパイルおよび管理するコントローラ、および／またはプロセスデータを処理し、保存するルーチンおよび／またはアルゴリズムによってプロセスデータに追加されてもよい。メタデータがプロセスデータと一緒に含まれていない他の実施例においては、本明細書に記載の方法例および装置例が、プロセスデータの種類を基に、選択されたプロセスデータと、メタデータ定義を含むメタデータのタイプデータベースとを相互参照することによってメタデータを生成する。生成されたメタデータは次に、プロセスデータと関連付けられ、本明細書に記載の方法例および装置例によって使用されて適切なデータ表現を生成する。

【0016】

本明細書に記載の方法例および装置例は、データ表現と関連付けられたプロセスデータおよび／またはメタデータに変更が生じると、表示されているデータ表現を自動的に更新および／または修正し得る。例えば、プロセス制御システムに対する変更により、そのプロセスデータと関連付けられた生成済みプロセスデータおよび／またはメタデータが追加、削除、および／または修正され得る。他の実施例においては、オペレータおよび／またはエンジニアが、データ表現内に表示されているプロセスデータを追加、削除、および／または修正してもよい。これらの両実施例においては、本明細書に記載の方法および装置が、変更されたシステムと関連付けられたメタデータを生成および／または特定し、影響を受けたデータ表現を自動的に更新し、修正されたデータ表現を再描画し、変更されたデータ表現を表示する。このように、プロセスデータ、メタデータ、および／またはプロセス制御システムに対して施された変更は、影響を受けたデータ表現で自動的に更新されるため、変更されたプロセスデータをオペレータおよび／またはエンジニアが手動で構成し直したり、データ表現内の適切なデータフィールドに改めてリンクしたりする必要がない。

【0017】

さらに、またはあるいは、本明細書に記載の方法例および装置例により、プロセスオペ

50

レータ、エンジニア、および／またはデザイナーが、設定および／または要件に基づいてデータ表現の一部分をカスタマイズすることができる。オペレータがデータ表現をカスタマイズすると、その方法例および／または装置例は、データ表現と共にカスタマイズ情報を記憶する。また、この方法例および装置例が、他の同様のタイプのデータ表現を更新したり、カスタマイズ情報と共にデータ表現タイプを更新したりしてもよい。カスタマイズ情報は、例えば、1つ以上のデータ表現の表示の修正、1つ以上のデータ表現の色の変更、1つ以上のデータ表現のレイアウトの変更、1つ以上のデータ表現と関連付けられたグラフィックスの変更、1つ以上のデータ表現の表示形式の変更、および／または1つ以上のデータ表現と関連付けられた識別情報の変更などを含み得る。

## 【0018】

10

メタデータに基づいてデータ表現を自動的に生成することにより、本明細書に記載の方法例および装置例は、ユーザインタフェースの開発中ではなく、ユーザによるリクエスト時にデータ表現を作成する動的なユーザインタフェース提示システムを提供する。さらに、プロセスデータの種類に基づいてデータ表現を自動的に作成することにより、本明細書に記載の方法例および装置例は、制御システムおよび対応するグラフィカルユーザインタフェースの同時開発を可能にし、エンジニアおよび／またはオペレータの生産性を改善する。例えばエンジニアは、プロセス制御システムの制御ストラテジの開発に注力できるようになり、プロセスデータがどのように示されるかを判断する必要がなくなる。同時にユーザインタフェース開発者は、ユーザインタフェースコントロールの開発に注力できるようになり、プロセスデータのタイプおよび／または位置を判断する必要がなくなる。さらに、本明細書に記載の方法例および装置例は、選択されたプロセス制御情報のみに基づいてデータ表現を動的に作成することにより、プロセス制御システムの柔軟性を改善する。データ表現タイプとメタデータとを相互参照する目的で使用されるデータベースは、エンジニアおよび／またはオペレータの要件に基づいて更新および／または改訂される場合があるため、プロセス制御システムのスケーラビリティも改善され得る。

20

## 【0019】

図1は、グラフィックスエンジン例102とデータサーバ例104とを含むプロセス制御環境例100を示すブロック図である。グラフィックスエンジン例102およびデータサーバ例104は、プロセス制御システム106内に位置する。ただし、他の実施例においては、グラフィックスエンジン102および／またはデータサーバ104が、サーバ内のプロセス制御システム106、分散コンピューティングネットワーク、および／またはプロセス制御システム106と通信可能に連結され得るその他任意の計算装置の外に位置する場合がある。さらにプロセス制御システム例106は、追加データサーバ（図示せず）および／または追加グラフィックスエンジン（図示せず）を含む場合がある。

30

## 【0020】

プロセス制御システム例106は、任意のタイプの製造施設、処理施設、自動化施設、安全装備された施設、および／またはその他任意のプロセス制御またはシステムを含み得る。いくつかの実施例において、プロセス制御システム106は、異なる場所に位置する複数の施設を含み得る。加えてプロセス制御環境例100は、同じ施設内に含まれていたり、異なる施設に位置したりする他のプロセス制御システム（図示せず）も含み得る。

40

## 【0021】

プロセス制御環境例100は、以下でさらに詳述されている方法例および装置例が好都合に使用され得るタイプのシステムを例示する目的で提供されている。ただし、本明細書に記載の方法例および装置例は、所望される場合には、図1に示されるプロセス制御環境例100および／またはプロセス制御システム106よりも複雑または単純な他のシステム、および／またはプロセス制御アクティビティ、事業管理アクティビティ、通信アクティビティなどと関連して使用されるシステムで好都合に使用され得る。

## 【0022】

図1のプロセス制御システム例106は、データサーバ104と通信可能に連結され得るコントローラ108を含む。プロセス制御システム106はまた、現場装置110（入

50

力および／または出力装置など)も含む。現場装置 110 は、入力情報を受信し、出力情報を生成し、かつ／またはプロセスを制御することのできる任意のタイプのプロセス制御コンポーネントを含み得る。現場装置 110 は、例えば、プロセスを制御するためのバルブ、ポンプ、ファン、ヒーター、クーラー、および／またはミキサーなどの制御装置を含み得る。加えて、現場装置 110 は、例えば、プロセスの部分を測定するための温度センサ、圧力計、濃度計、液面計、流量計、および／または蒸気センサなどの測定または監視装置も含み得る。制御装置は、入力経路 112 を介してコントローラ 108 から命令を受信し、指定されたコマンドを実行し得るとともに、現場装置 110 によって実装および／または制御されるプロセスに変更を施し得る。さらに、測定装置は、プロセスデータ、環境データ、および／または入力装置データを測定し、測定されたデータをプロセスデータとして出力経路 114 を介してコントローラ 108 に送信する。このプロセスデータは、現場装置 110 の各自から受信した測定出力値に対応する変数の値を含み得る。

10

#### 【0023】

図 1 の図示実施例において、コントローラ例 108 は、入力経路 112 および／または出力経路 114 を介してプロセス制御システム 106 内の現場装置 110 と通信し得る。入力経路 112 および出力経路 114 は、データバスによって実装されてもよい。このデータバスは、プロセス制御システム 106 内の中間通信コンポーネントに連結されてもよい。これらの通信コンポーネントは、コマンド領域にある現場装置 110 をデータバスと通信可能に連結するためのフィールド結合ボックスを含み得る。加えて、通信コンポーネントは、現場装置 110 および／またはフィールド結合ボックスへの通信経路を組成するための整理キャビネットを含み得る。さらに通信コンポーネントは、現場装置 110 からデータを受信し、そのデータをコントローラ例 108 が受信可能な通信に変換するための I/O カード 116 (I/O 装置など) を含み得る。加えて、これらの I/O カード 116 は、コントローラ 108 からのデータまたは通信を、対応する現場装置 110 が処理可能なデータ形式に変換し得る。一実施例において、データバスは、Fieldbus プロトコルまたは他のタイプの有線のおよび／または無線通信プロトコル (Profibus プロトコルや HART プロトコルなど) を使用して実装されてもよい。

20

#### 【0024】

図 1 のコントローラ例 108 は、プロセス制御システム 106 内で現場装置 110 を制御するために、プロセス制御アルゴリズム、機能、および／または命令など、1つ以上の制御ルーチンを管理する。制御ルーチンは、プロセス監視アプリケーション、アラーム管理アプリケーション、プロセストレンド分析および／または履歴管理アプリケーション、診断アプリケーション、バッチ処理および／またはキャンペーン管理アプリケーション、統計アプリケーション、ストリーミングビデオアプリケーション、高度な制御アプリケーション、安全装備されたアプリケーションなどを含み得る。制御ルーチンによって、プロセス制御システム 106 が特定の品質しきい値内で所望の製品の指定量を確実に生産し得る。例えば、プロセス制御システム 106 は、最終段階および／またはバッチ処理時に製品を生産するバッヂシステムとして構成される場合がある。他の実施例において、プロセス制御システム 106 は、製品を絶え間なく生産する連続プロセス製造システムを含み得る。さらにコントローラ 108 は、制御ルーチン内で利用されるプロセスデータをデータサーバ例 104 に転送する場合がある。他の実施例において、データサーバ 104 は、プロセスデータの有無を確認するため、および／またはグラフィックスエンジン 102 からリクエストを受けたときにコントローラ 108 を周期的にポーリングする場合がある。

30

#### 【0025】

図 1 のプロセス制御環境例 100 は、ローカルエリアネットワーク (LAN) 122 を介してグラフィックスエンジン 102 と通信可能に連結されたワークステーション 120 を含む。ワークステーション例 120 は、パーソナルコンピュータ、ラップトップ、サーバ、コントローラ、スマートフォン、携帯情報端末 (PDA)、マイクロコンピュータなどを含む任意のコンピューティング装置を含み得る。加えて、ワークステーション 120 は、任意の適切なコンピュータシステムまたは処理システム (図 13 のプロセッサシステ

40

50

ム P 10 など)を使用して実装され得る。例えば、ワークステーション 106 は、シングルプロセッサパーソナルコンピュータ、シングルまたはマルチプロセッサワークステーションなどを使用して実装できる可能性がある。

#### 【0026】

図 1 の実施例は、プロセス制御システム 106 にとって外部のワークステーション例 120 を示す。他の実施例において、ワークステーション 120 は、プロセス制御システム 106 内に含まれる場合があり、かつ / またはコントローラ 108 と通信可能に直接連結される場合がある。加えて、プロセス制御環境 100 は、他のワークステーション(図示せず)をコントローラ 108、データサーバ 104、および / またはグラフィックスエンジン 102 と通信可能に連結し、かつ / またはワークステーション 120 を他のプロセス制御システム内の他のコントローラ(図示せず)および / またはグラフィックスエンジン(図示せず)と通信可能に連結するためのルータ(図示せず)を含み得る。さらにプロセス制御環境 100 は、プロセス制御環境 100 内のリソースへのアクセスをリモートワークステーション(プロセス制御環境 100 外のワークステーションなど)に提供するためのファイアウォール(図示せず)を含み得る。

10

#### 【0027】

LAN 例 122 は、任意の所望の通信メディアおよびプロトコルを使用して実装され得る。例えば、LAN 122 は、有線または無線のイーサネット(登録商標)通信方式に基づいてもよい。ただし、その他任意の適切な通信メディアおよびプロトコルも使用できる可能性がある。さらに、LAN 122 は 1 つだけ示されているが、ワークステーション 120 内の複数の LAN および適切な通信ハードウェアを使用して、ワークステーション 120 とそれぞれの類似ワークステーション(図示せず)との間で冗長通信経路を設けてよい。

20

#### 【0028】

プロセス制御システム 106 にアクセス可能なワークステーション例 120 および / または他のワークステーション(図示せず)は、プロセス制御システム 106 内の 1 つ以上のプロセスを参照、修正、および / または矯正するように構成されてもよい。例えば、ワークステーション 120 は、グラフィックスエンジン 102 によって生成されるデータ表現をフォーマット、管理、および / または表示するユーザインタフェース(UI)提示プロセッサ 124 を含んでもよい。UI 提示プロセッサ 124 は、ユーザインタフェース 126 を介してデータ表現を表示する。ユーザインタフェース例 126 は、ワークステーション 120 の一部であるグラフフィカルウインドウに表示されることで 1 つ以上のデータ表現を示してもよい。ワークステーション 120 は、追加 UI 提示プロセッサ(図示せず)を介して複数のユーザインタフェース 126 を表示できてもよい。あるいは、実施例 UI 提示プロセッサ 124 が複数のユーザインタフェース 126 を管理できてもよい。

30

#### 【0029】

図 1 のユーザインタフェース例 126 は、現場装置 110 と関連付けられたデータ表現を示す。オペレータは、オペレータがデータ表現として表示したいプロセス制御情報を選択することにより、データ表現の生成を開始する。プロセス制御情報は、ユーザインタフェース 126 でオペレータがアイコン、ファイル名、1 つ以上の機能ブロック、プロセス制御システムの概略図の一部分、および / またはそのプロセス制御情報と関連付けられた識別名を選択することによって選択され得る。グラフィックスエンジン 102 は、選択を受信し、プロセス制御変数、現場装置 110、入力経路 112、出力経路 114、機能ブロック、パラメータ、アプリケーション、ルーチン、および / または選択されたプロセス制御情報と関連付けられた他のプロセス制御コンポーネントを判断する。次にグラフィックスエンジン 102 は、選択されたプロセス制御情報と関連付けられたプロセス制御コンポーネントの識別名をデータサーバ 104 に送信することにより、プロセスデータを取得する。データサーバ例 104 は、プロセスデータおよび / またはプロセスデータの格納位置を特定し、この情報をグラフィックスエンジン 102 に転送する。いくつかの実施例においては、プロセスデータがすでにデータサーバ 104 に格納されている場合がある。他

40

50

の実施例においては、プロセスデータが、コントローラ 108 に格納されているか、位置している場合がある。後者の実施例においては、データサーバ 104 がコントローラ 108 にリクエストされた処理データを転送するようにリクエストしてもよい。このリクエストは、読み込まれたリクエストおよび / またはサブスクライブリクエストを含み得る。

#### 【0030】

プロセスデータをグラフィックスエンジン 102 に送信する前に、図 1 のデータサーバ例 104 が、プロセスデータと関連付けられたメタデータを判断する。場合によっては、メタデータがすでにプロセスデータと関連付けられており、かつ / または組み込まれている場合がある。プロセスデータがメタデータを含まない他の実施例においては、データサーバ 104 が、プロセスデータとメタデータタイプとをデータベースで相互参照することによってメタデータを生成する。選択されたプロセスデータと関連付けられたメタデータを判断すると、データサーバ 104 は、メタデータおよびプロセスデータをグラフィックスエンジン 102 に転送する。次にグラフィックスエンジン 102 が、適用可能なメタデータ識別名によってメタデータを解析し得る。メタデータ内のメタデータ識別名は、関連付けられたプロセスデータがデータ表現内でどのように表示されるかを説明している。例えば、メタデータ識別名は、データ表現のタイプ（グラフ、チャート、テーブルなど）、プロセスデータ（フォントサイズ、フォントタイプ、フォントカラー、テキスト位置合わせなど）を表示するための形式、プロセスデータ（小数、16進数、ストリング、テキストなど）を表示するためのデータ型、および / またはプロセスデータ（列、行、グラフデータ、ステータスデータなど）の表示特性を表すタグを含み得る。

10

20

30

#### 【0031】

次にグラフィックスエンジン例 102 は、解析されたメタデータのタイプを特定し、そのメタデータタイプとデータ表現タイプとを相互参照する場合がある。例えば、グラフィックスエンジン 102 は、データ表現タイプのインスタンスを生成し、データ表現のインスタンスにメタデータを関連付ける場合がある。具体的には、グラフィックスエンジン 102 が、データ表現タイプテンプレートをコピーし、そのデータ表現コピー内のデータフィールドにメタデータを一致させる場合がある。例えば、テーブルデータ表現がデータフィールドを行または列に含む場合がある。グラフィックスエンジン例 102 は、列メタデータ識別名によってメタデータの位置を特定し、一致したメタデータと関連付けられたプロセスデータを列データフィールドにリンクする場合がある。

#### 【0032】

いくつかの実施例においては、グラフィックスエンジン 102 が、メタデータタイプに対応する複数のデータ表現タイプを判断する場合がある。これらの実施例においては、グラフィックスエンジン 102 がワークステーション 120 で、オペレータにデータ表現タイプを選択するよう促す場合がある。あるいは、同じタイプの他のデータ表現がメタデータタイプと関連付けられるよう指定されたかどうかをグラフィックスエンジン 102 が判断する場合がある。さらには、グラフィックスエンジン 102 が、一致するデータ表現タイプごとにデータ表現を作成する場合もある。

#### 【0033】

図 1 のグラフィックスエンジン 102 は、データ表現を描画し、その描画をワークステーション 120 内の UI 提示プロセッサ 124 に送信することにより、作成されたデータ表現を表示する。次に UI 提示プロセッサ 124 は、ユーザインターフェース 126 を介してその描画をデータ表現の表示版に変換する。グラフィックスエンジン 102 は、データ表現と関連付けられたデータフィールドの各々にアクセスし、それらのデータフィールドにリンクされているプロセスデータを取得することによってそのデータ表現を表示装置に描画した後、そのデータ表現の機能（グラフ機能、テーブル機能、チャート機能など）に従ってプロセスデータの表示描画を作成する。

40

#### 【0034】

図 1 のグラフィックスエンジン例 102 はまた、プロセスデータおよび / またはメタデータに対する変更に基づいてデータ表現を修正する。いくつかの実施例においては、現場

50

装置 110 および / またはプロセス制御システム 106 に対する変更により、プロセスデータが追加、削除、および / または修正される場合がある。他の実施例においては、プロセス制御エンジニアおよび / またはオペレータが、プロセスデータおよび / またはプロセスデータと関連付けられたメタデータを追加、削除、および / または修正する場合もある。グラフィックスエンジン 102 は、プロセスデータおよび / またはメタデータの変更を検出し、影響を受けたデータ表現を後で修正する場合がある。グラフィックスエンジン 102 は、データサーバ 104 から変更の合図を受信することにより、プロセスデータおよび / またはメタデータに対する変更を検出しえる。他の実施例においては、グラフィックスエンジン 102 が、データサーバ 104 をポーリングして、メタデータおよび / またはプロセスデータに対する変更の有無を確認し得る。さらに / あるいは、グラフィックスエンジン 102 が、UI 提示プロセッサ 124 を介して、オペレータによって開始された変更をワークステーション 120 から受信する場合もある。

10

#### 【0035】

例えば、データ表現が第 1 の温度センサ現場装置 110 のステータスを示す場合がある。第 2 の温度センサがプロセス制御システム 106 に追加されると、グラフィックスエンジン 102 は、新しい温度センサからの出力経路、および / または新しい温度センサの出力経路と関連付けられたメタデータを検出する場合がある。次にグラフィックスエンジン 102 は、第 1 の温度センサのステータスと関連付けられたデータ表現を判断し、そのデータ表現内の適切なデータフィールドに第 2 の温度センサのステータスを参照するリンクを追加する。別の実施例においては、オペレータが、データ型表現をテキストフィールドからグラフに変更することによってセンサからの出力経路と関連付けられたメタデータを修正する場合がある。メタデータの変更を検出すると、グラフィックスエンジン例 102 は、表現タイプをグラフに変更し、センサからのプロセスデータをグラフ内に含まれるデータフィールドにリンクすることによってデータ表現を修正する。次にグラフィックスエンジン 102 は、そのグラフデータ表現をユーザインターフェース 124 で表示できるように描画し得る。

20

#### 【0036】

さらにグラフィックスエンジン例 102 は、オペレータおよび / またはエンジニアによるデータ表現のカスタマイズを管理する場合がある。オペレータがデータ表現をカスタマイズしたときに、グラフィックスエンジン 102 は、関連付けられたプロセスデータおよび / またはメタデータを判断し得る。グラフィックスエンジン 102 は、カスタマイズされた情報に基づいてデータ表現を修正し、表示する場合もある。さらにグラフィックスエンジン 102 は、同じデータ表現に後でアクセスした場合に、そのデータ表現を表示する前に同じカスタマイズをそのデータ表現に適用できるように、データ表現のカスタマイズ情報を記憶する場合がある。

30

#### 【0037】

図 2 は、図 1 のグラフィックスエンジン例 102 およびデータサーバ例 104 の機能ブロック図 200 である。グラフィックスエンジン例 102 およびデータサーバ例 104 は別々の機能ブロックとして示されているが、他の実施例において、データサーバ 104 がグラフィックスエンジン 102 内に含まれていてもよい。あるいは、グラフィックスエンジン 102 およびデータサーバ 104 が 1 つの機能ブロック内で結合していてもよい。加えて図 2 は、データサーバ 104 および UI 提示プロセッサ 124 と通信可能に連結されたグラフィックスエンジン 102 を示しているが、グラフィックスエンジン 102 が他のデータサーバおよび / または UI 提示プロセッサと通信可能に連結されていてもよい。さらにデータサーバ例 104 がコントローラ 108 と通信可能に連結されているが、データサーバ 104 は他のコントローラと通信可能に連結されていてもよい。

40

#### 【0038】

オペレータによって選択されたプロセス制御情報を UI 提示プロセッサ 124 を介して受信するために、グラフィックスエンジン例 102 は選択レシーバ例 202 を含む。選択されたプロセス制御情報は、バッチ識別名、プロセス識別名、コントローラ識別名、現場

50

装置識別名、プロセスデータと関連付けられたファイル、プロセスデータへのリンク、1つ以上の機能ブロック、1つ以上のパラメータ、1つ以上のプロセス制御エンドポイント、および／またはプロセス制御ルーチンを含み得るが、これらに限定されない。U I 提示プロセッサ124は、選択されたプロセス制御情報をコンパイルし、この情報を選択レシーバ202に転送する。プロセス制御情報を受信すると、選択レシーバ202はこのプロセス制御情報を、データマネージャ204が処理できるようになるまでキューに入れておく。

#### 【0039】

選択されたプロセス制御情報と関連付けられたプロセスデータを特定するために、図2のグラフィックスエンジン例102は、データマネージャ204を含む。データマネージャ例204は、選択されたプロセス制御情報と関連付けられたプロセスデータを生成、処理、および／または管理し得るプロセス制御コンポーネントを特定することによってプロセスデータを特定する。このプロセス制御コンポーネントは、選択されたプロセス制御情報に対応しており、現場装置、プロセスコントローラ、ルーチン、アルゴリズム、I/Oカード、および／またはデータを生成および／または処理できるその他任意のプロセスコントローラも含み得る。

10

#### 【0040】

プロセス制御コンポーネントを特定すると、データマネージャ例204は、生成、処理、および／または特定されたプロセス制御コンポーネントと関連付けられている可能性があるプロセスデータを求めるリクエストをデータサーバ104に送信する。例えば、オペレータが、U I 提示プロセッサ124によって表示される概略図上で現場装置を選択して現場装置によって生成されたプロセスデータを参照してもよい。U I 提示プロセッサ124は、現場装置の識別値（プロセス制御情報など）を選択レシーバ202に送信し、次にそれらの識別値をデータマネージャ204に転送する。データマネージャ204は、現場装置識別値を現場装置（プロセス制御コンポーネントなど）として識別し、その現場装置と関連付けられているプロセスデータを求めるリクエストをデータサーバ104に送信する。

20

#### 【0041】

他の実施例においては、オペレータが概略図の複数部分または複数の機能ブロックを選択してもよい。これらの実施例において、プロセス制御情報は、複数の現場装置識別名、アラーム識別名、現場装置からの出力経路に基づいてルーチンによって計算されるパラメータ、および／または現場装置によって生成される出力経路に対応する出力変数名を含み得る。これらの実施例においては、データマネージャ204が、関連付けられたプロセスデータを求めるリクエストをデータサーバ104に送信する前に、タイプ、位置、および／またはその他任意のフィルタリング可能な特性別にプロセス制御情報を整理する。このようにして、データマネージャ例204は、プロセス制御情報の各部分がデータサーバ104によって取得されるプロセスデータに対応するようにプロセスデータを整理する。

30

#### 【0042】

プロセスデータを取得するために、図2のデータサーバ例104はデータリトリーバ206を含む。データリトリーバ例206は、データマネージャ204からリクエストを受信し、リクエストされた処理制御コンポーネントおよび／または情報と関連付けられたプロセスデータを特定して、プロセスデータの位置を判断し、そのプロセスデータをデータマネージャ204に転送する。いくつかの実施例においては、プロセスデータが、データリトリーバ206によってアクセス可能なデータサーバ104のデータベース内に位置する場合がある。プロセスデータがデータサーバ104に位置しないとデータリトリーバ206が判断する他の実施例においては、データリトリーバ206が、プロセスデータを求めるリクエストをコントローラ108に送信する場合がある。どちらの実施例においても、データリトリーバ206は、読み込まれたリクエストおよび／またはサブスクライブリクエストを送信することによってプロセスデータを取得し得る。サブスクライブリクエストが送信される実施例においては、データリトリーバ206が、後にデータマネージャ2

40

50

0 4 に転送するプロセスデータを周期的に受信する場合がある。これらの実施例においては、データリトリーバ 2 0 6 がプロセスデータを利用できるようになると、U I 提示プロセッサ 1 2 4 によって表示されるデータ表現が現在のプロセスデータと共に自動的に更新され得る。

#### 【 0 0 4 3 】

データリトリーバ 2 0 6 は、リクエストされた処理データを受信すると、データマネージャ 2 0 4 によって送信されたプロセス制御コンポーネントおよび / または情報にそのプロセスデータをリンクし、リンクされた情報をデータマネージャ 2 0 4 に送信する。このようにして、データマネージャ 2 0 4 は、受信したプロセスデータが選択されたプロセス制御情報および / またはコンポーネントに対応すると判断することができる。他の実装例においては、データリトリーバ 2 0 6 が、対応するプロセス制御情報および / またはコンポーネントをリンクしないでデータマネージャ 2 0 4 にプロセスデータを送信する場合がある。

10

#### 【 0 0 4 4 】

プロセス制御システムで、および / またはデータ表現内で利用されているプロセスデータに対して変更が行われる実施例においては、データリトリーバ例 2 0 6 が、変更の合図を受信し、その変更と関連付けられたプロセスデータを取得し、データマネージャ 2 0 4 にそのプロセスデータを転送する。あるいは、データリトリーバ 2 0 6 が周期的にコントローラ 1 0 8 をポーリングして、プロセスデータおよび / またはデータ表現に影響を及ぼす可能性のあるプロセス制御システムに何らかの変更があるかどうかを判断する場合もある。例えば、データリトリーバ 2 0 6 が、特定のプロセス制御領域にある現場装置からのプロセスデータを取得した場合、データリトリーバ 2 0 6 はその特定のプロセス制御領域に新たに追加された現場装置からデータマネージャ 2 0 4 にプロセスデータを転送する場合がある。

20

#### 【 0 0 4 5 】

プロセスデータを取得することに加え、データサーバ例 1 0 4 は、取得されたプロセスデータと関連付けられたメタデータを判断するためのメタデータジェネレータ 2 0 8 を含む。メタデータジェネレータ例 2 0 8 は、取得されたプロセスデータのコピーをデータリトリーバ 2 0 6 から受信し、そのプロセスデータ内に組み込まれてあり、かつ / またはそのプロセスデータと関連付けられたメタデータが存在するかどうかを判断する。いくつかの実施例においては、組み込まれた、かつ / または関連付けられたそのメタデータが、プロセスデータのヘッダ内に含まれる場合がある。他の実施例においては、メタデータが、プロセスデータのストリング、データ語、および / または行内に含まれる場合がある。プロセスデータと関連付けられたメタデータが存在する場合には、メタデータジェネレータ 2 0 8 が、そのメタデータがデータ表現を生成するのに十分かどうかを判断する。データ表現を生成するのに十分なメタデータは、プロセスデータの種類を特定するメタデータ、そのプロセスデータの形式、および / またはプロセスデータの表示設定を含み得る。

30

#### 【 0 0 4 6 】

例えば、メッセージおよび / またはデータ語を介して送信されるプロセスデータは、バッチ番号識別名について説明し、「2 0 0 8 0 2 2 9 . 1 6 3 4 5 0」というプロセスデータ値を有する場合がある。このメッセージおよび / またはデータ語は、「C o l u m n V a l u e」、「< N a m e B a t c h I D >」、「< L e n g t h 1 4 >」、および「< T y p e D e c i m a l >」など、データ値と関連付けられたメタデータを表すバイトおよび / または語も含み得る。メタデータジェネレータ 2 0 8 は、「<」、「>」、「C o l u m n V a l u e」、「N a m e」、「L e n g t h #」、および「T y p e」などのコード名、識別名、および / または記号によってメタデータを識別する。十分なメタデータを特定すると、メタデータジェネレータ 2 0 8 は、プロセスデータおよびメタデータをグラフィックスエンジン 1 0 2 に転送する。

40

#### 【 0 0 4 7 】

別の実施例において、メッセージまたはデータ語がプロセスデータだけを含む場合、お

50

および／または不十分なメタデータだけを含む場合、メタデータジェネレータ例208は、プロセスデータを特定し、そのプロセスデータとメタデータタイプとを相互参照する場合がある。メタデータのタイプデータベース210は、メタデータジェネレータ208がプロセスデータとメタデータタイプとを相互参照するのに利用する可能性のあるリストを含み得る。メタデータのタイプデータベース210は、電子的消去が可能なプログラマブル読み取り専用メモリ(EEPROM)、ランダムアクセスメモリ(RAM)、読み取り専用メモリ(ROM)、および／またはその他任意のタイプのメモリによって実装され得る。

#### 【0048】

一実施例において、メタデータジェネレータ208は「20080229.163450」というプロセスデータ値を受信し得る。次にメタデータジェネレータ例208は、このプロセスデータ値が「#####.#####」という形式の14桁10進値であると判断し得る。メタデータジェネレータ208は、メタデータのタイプデータベース210にアクセスして、「#####.#####」という形式とメタデータの識別名またはタイプとを相互参照し得る。本実施例においては、「#####.#####」という形式が、例えば「Name Batch ID」、「Length 14」、「Type Decimal」、および「Column Value」といったメタデータ識別名など、バッチ識別名のメタデータ識別名と相互参照され得る。メタデータジェネレータ208は、ヘッダまたは他のデータバイト内のメタデータ、および／またはプロセスデータの語を組み込むことにより、かつ／またはこのメタデータをこのプロセスデータにリンクすることにより、判断されたメタデータのタイプおよび／または識別名をプロセスデータ値と関連付ける。次にメタデータジェネレータ208は、このメタデータおよびプロセスデータをグラフィックスエンジン102に送信する。

#### 【0049】

プロセスデータがデータストリングおよび／またはプロセスデータの各種部分を含み得る実施例において、メタデータジェネレータ例208は、各部分および／またはそのストリング内の各データ値のメタデータの識別名および／またはタイプを判断する。例えば、バッチタイプと関連付けられたプロセスデータを求めるリクエストにより、データリトリーバ206はそのバッチタイプと関連付けられたバッチプロセスデータストリングを取得し得る(「20080229.163450；チョコレートチップクッキー；500ポンドのダークチョコレートチップクッキーを作る；2/29/2008；0:23:31 AM；0:53:45など）。メタデータジェネレータ例208は、そのストリング内のセミコロンの位置に基づいて各種プロセスデータ値を区切る。他の実施例においては、セパレータが、カンマ、ダッシュ、および／またはその他任意の記号またはコードとなり得る。メタデータジェネレータ208は、対応するデータ形式に基づいて各プロセスデータ値のメタデータ識別名を判断し、そのメタデータをプロセスデータと関連付ける場合がある。その後メタデータジェネレータ208は、プロセスデータストリングおよび判断されたメタデータをグラフィックスエンジン102に転送する。

#### 【0050】

メタデータをデータ表現の生成に使用できるようにメタデータを解析するために、図2のグラフィックスエンジン例102は、メタデータパーサ212を含む。メタデータパーサ例212は、プロセスデータと関連付けられたどのメタデータがデータ表現の生成に関連するか、または役立つかを判断する。余分または無関係なメタデータはメタデータパーサ212によって除外され、関連するメタデータが、データ表現を作成するためにデータ表現ジェネレータ214に転送される。メタデータパーサ212は、現場装置、ルーチン、コントローラ、および／または他のプロセス制御コンポーネントによってプロセスデータと関連付けられているメタデータを解析し得る。例えば、メタデータパーサ212は、I/Oカードを特定する一部のメタデータがデータ表現の生成とは無関係であると判断する場合がある。あるいは、メタデータジェネレータ208によって生成されたメタデータは、通常はメタデータパーサ212によって解析されない。他の実施例において、メタデ

10

20

30

40

50

ータパーサ 212 は、プロセスデータがメタデータの対応部分と関連付けられるように、メタデータのストリング内でメタデータを区切る場合がある。メタデータを解析すると、メタデータパーサ 212 は、解析されたメタデータをデータ表現ジェネレータ 214 に転送する。

#### 【0051】

メタデータに基づいてデータ表現を生成または作成するために、図 2 のグラフィックスエンジン例 102 はデータ表現ジェネレータ 214 を含む。データ表現ジェネレータ例 214 は、プロセスデータおよび / またはメタデータと関連付けられたタグに基づいて、データマネージャ 204 から受信した対応するプロセスデータを、メタデータパーサ 212 から受信したメタデータと結合する。他の実施例においては、データ表現ジェネレータ 214 が、メタデータおよび関連付けられたプロセスデータをメタデータパーサ 212 から受信する場合がある。これらの他の実施例において、データ表現ジェネレータ 214 は、メタデータパーサ 212 から受信したプロセスデータの値を解決する場合があり、かつ / またはデータマネージャ 204 から受信したプロセスデータと比較して、このプロセスデータの値がメタデータジェネレータ 208 によって変更、不整合化、あるいは再フォーマットされていないことを確認する場合がある。

#### 【0052】

メタデータおよび関連付けられたプロセスデータを受信すると、データ表現ジェネレータ例 214 は、メタデータのタイプを特定し、特定されたメタデータタイプと少なくとも 1 つのデータ表現タイプとをメタデータルールデータベース 216 で相互参照し、特定されたメタデータのタイプと関連付けられた少なくとも 1 つのデータ表現タイプのインスタンスを生成することにより、1 つ以上のデータ表現を自動的に生成する。次にデータ表現ジェネレータ例 214 は、メタデータを少なくとも 1 つのデータ表現タイプのインスタンスの 1 つ以上のデータフィールドと関連付けることと、そのメタデータと関連付けられたプロセスデータを少なくとも 1 つのデータ表現タイプのインスタンスのデータフィールドと関連付けることとによってプロセスデータを少なくとも 1 つのデータ表現タイプの新たに作成されたインスタンスと関連付け、データ表現を作成する。

#### 【0053】

データ表現ジェネレータ例 214 は、コード名、識別名、および / または記号によってメタデータタイプを特定する。次にデータ表現ジェネレータ 214 は、メタデータルールデータベース 216 にアクセスして、特定されたメタデータのタイプと 1 つ以上のデータ表現タイプとを相互参照する。メタデータルールデータベース 216 は、EEPROM、RAM、ROM、および / またはその他任意のタイプのメモリによって実装されてよく、およびデータ表現タイプの識別名と相互参照されるメタデータ識別名または記号によって構成されるメタデータタイプ 1 つ以上のリストを含み得る。例えば、「Column」というメタデータ識別名は、テーブルデータ表現タイプと相互参照され得る。他の実施例においては、「Time Axis Data」というメタデータ識別名が折れ線グラフデータ表現タイプと相互参照され得る。さらに別の例においては、「Expected Contribution」というメタデータ識別名が棒グラフデータ表現タイプと相互参照され得る。メタデータルールデータベース 216 に格納されるリストは、グラフィックスエンジン 102 のデザイナーおよび / またはグラフィックスデザイナーによって最初に作成されてもよい。さらにこのリストは、プロセス制御エンジニアおよび / またはオペレータによって更新または修正されてもよい。

#### 【0054】

いくつかの実施例において、データ表現ジェネレータ 214 は、プロセスデータの一部分と関連付けられた複数のメタデータの識別名および / またはタイプを判断する場合があり、その複数のメタデータの識別名および / またはタイプとデータ表現タイプとを相互参照する場合がある。加えて、いくつかの実施例においては、データ表現ジェネレータ 214 が、そのメタデータのタイプおよび / または識別名に基づいて 2 つ以上の可能なデータ表現タイプを特定する場合がある。これらの実施例において、データ表現ジェネレータ 214

10

20

30

40

50

14は、U I提示プロセッサ124を介してユーザにデータ表現タイプを選択するよう促す場合がある。あるいは、データ表現ジェネレータ214は、2つ以上のデータ表現タイプを生成する場合があり、またはデータ表現データベース218にアクセスして、ユーザがメタデータのタイプおよび／または識別名に基づいてデータ表現タイプの設定を指定したかどうかを判断する場合もある。データ表現データベース218は、EEPROM、RAM、ROM、および／またはその他任意のタイプのメモリによって実装され得る。

#### 【0055】

データ表現タイプを判断すると、データ表現ジェネレータ例214は、データ表現タイプテンプレートのコピーを作成することによってデータ表現タイプのインスタンスを生成する。インスタンスとは、プロセスデータとのリンクを含まないデータ表現のことである。インスタンスからデータ表現を作成するために、データ表現ジェネレータ214は、メタデータの識別名をインスタンス内のデータフィールドに一致させることによってメタデータをデータ表現インスタンスと関連付ける。例えば、テーブルデータ表現タイプは、列および行のデータフィールドを持ち得る。データ表現ジェネレータ例214は、「Column」または「Row」という識別名内のメタデータを特定し、関連付けられたプロセスデータを対応データフィールドとリンクする。データ表現ジェネレータ214はまた、データ表現が表示されたときにU I提示プロセッサ124がプロセスデータをフォーマットできるように、データフィールド内にメタデータをリンクしているか、含んでいる場合がある。例えば、「Type Decimal」というメタデータ識別名は、プロセスデータが十進数形式で表示されるように、データフィールドでプロセスデータと関連付けられている場合がある。

10

20

30

#### 【0056】

データ表現ジェネレータ例214は、データフィールドに一致するメタデータと関連付けられたプロセスデータを特定し、そのプロセスデータを一致するデータフィールドに挿入することによってプロセスデータをデータフィールドにリンクする。あるいは、データ表現ジェネレータ214は、プロセスデータにアクセスするためにデータ表現表示マネージャ220によって使用され得るプロセスデータの位置（ファイルディレクトリの位置など）を挿入してもよい。データ表現ジェネレータ例214は、データ表現タイプと関連付けられたプロセスデータのすべてがリンクされるまで、インスタンス内のデータフィールドをリンクする。そのデータ表現タイプのインスタンスのデータフィールドがプロセスデータにリンクされると、データ表現が作成される。

#### 【0057】

データ表現を作成すると、図2のデータ表現ジェネレータ例214は、連結されたプロセスデータおよび関連付けられたメタデータを含むデータ表現のコピーをデータ表現データベース218に格納する。ユーザ（オペレータおよび／またはエンジニアなど）が、同じプロセス制御情報および／またはリクエストを選択してデータ表現を参照したときに同じデータ表現が速やかにアクセスされ、表示されるように、コピーが格納される。データ表現データベース218は、データ表現用のユーザカスタマイズ情報を格納してもよい。

#### 【0058】

データ表現ジェネレータ例214は、データ表現をデータ表現表示マネージャ220に送信することによってデータ表現の表示を開始する。データ表現表示マネージャ例220は、データ表現を表示用に描画し、U I提示プロセッサ124にその描画を転送することにより、受信したデータ表現を表示する。データ表現表示マネージャ220は、データフィールド内のプロセスデータをグラフィカルデータ表現に適用することに基づいてデータ表現のイメージを生成することによってデータ表現を描画し得る。データフィールドがプロセスデータの位置を含む実施例においては、データ表現表示マネージャ220が、プロセスデータ値のデータ位置にアクセスする。

40

#### 【0059】

データ表現の描画を受信すると、U I提示プロセッサ124は、ユーザが参照可能なユーザインターフェース（ユーザインターフェース126など）にデータ表現を表示する。U I

50

提示プロセッサ例 124 は、ユーザによって指定された 1 つ以上のユーザインタフェースにデータ表現を表示する場合がある。いくつかの実施例において、ユーザインタフェースは、例えば、制御インターフェース、グラフィカルインターフェース、ウェブブラウザ、アプリケーション、および / またはデータ表現を表示できるその他任意の表示プログラムを含み得る。

#### 【0060】

ユーザがデータ表現をカスタマイズしてもよい実施例においては、データ表現表示マネージャ例 220 が、UI 提示プロセッサ 124 からカスタマイズ情報を受信する。このカスタマイズ情報は、データ表現、データ表現と関連付けられたメタデータ、および / またはデータ表現と関連付けられたプロセスデータを含み得る。データ表現は、ユーザがデータ表現の表示を修正すること、データ表現の色を変更すること、データ表現のレイアウトを変更すること、データ表現と関連付けられたグラフィックスを変更すること、データ表現の表示形式を変更すること、データ表現と関連付けられた識別情報を変更することなどによってカスタマイズされ得る。カスタマイズ情報を受信すると、データ表現表示マネージャ 220 は、データ表現ジェネレータ 214 にそのカスタマイズ情報を転送する。

10

#### 【0061】

データ表現ジェネレータ例 214 は、関連付けられたデータ表現とカスタマイズ情報を参照し、データ表現データベース 218 にアクセスしてそのカスタマイズをデータ表現と共に格納し得る。加えてデータ表現ジェネレータ 214 は、カスタマイズ情報に基づいてデータ表現を修正する。例えば、データ表現内に表示されているデータのフォントカラーを変更することをユーザが選択すると、データ表現ジェネレータ 214 は、フォントカラーの変更をカスタマイズ情報として受信し、ユーザがフォントカラーの変更を実行するために指定したメタデータおよび / またはプロセスデータを判断し、フォントカラーと関連付けられたメタデータをユーザが選択したフォントカラーに変更することによってフォントカラーの変更をそのプロセスデータに適用する。データ表現がフォントカラーと関連付けられたメタデータおよび / またはメタデータ識別名を含まない実施例においては、データ表現ジェネレータ例 214 が、フォントカラーメタデータを作成し、そのフォントカラーメタデータを対応するプロセスデータと関連付ける。

20

#### 【0062】

データ表現を更新してユーザからのカスタマイズ情報を反映すると、データ表現ジェネレータ 214 は、そのデータ表現をデータ表現表示マネージャ 220 に送信する。次にデータ表現表示マネージャ 220 は、そのカスタマイズ情報を用いてデータ表現を再描画し、描画されたデータ表現をユーザインタフェース内に表示するために UI 提示プロセッサ 124 に送信する。

30

#### 【0063】

加えて、ユーザがプロセスデータおよび / またはデータ表現の定義を変更することによってデータ表現を修正する実施例においては、UI 提示プロセッサ 124 が、修正されたデータ表現をデータ表現ジェネレータ 214 に送信する。プロセスデータを変更することは、例えば、第 2 のデータ表現をデータ表現に追加すること、パラメータ（プロセスデータなど）をデータ表現に追加すること、データ表現からパラメータを削除すること、データ表現のパラメータの種類（プロセスデータの種類および / またはメタデータタイプなど）を修正すること、またはデータ表現の視覚化を修正することなども含み得る。他の実施例においては、データ表現に対する変更が、現場装置（図 1 の現場装置 110 など）、プロセス制御システム（プロセス制御システム 106 など）、および / またはコントローラ（コントローラ 108 など）内のプロセスデータ、メタデータ、および / またはコンポーネントに対する変更に起因する場合がある。

40

#### 【0064】

変更情報を受信すると、データ表現ジェネレータ 214 は、プロセスデータおよび / またはメタデータに対する変更に基づいて、メタデータ、プロセスデータ、および / またはデータ表現の定義を修正する。次にデータ表現ジェネレータ 214 は、データ表現データ

50

ベース 218 に修正されたデータ表現を格納し、修正されたデータ表現をユーザインタフェース内に表示するためにデータ表現表示マネージャ 220 に送信する。このようにして、データ表現ジェネレータ 214 は、ユーザおよび／またはシステムによって指定されたプロセスデータおよび／またはメタデータに対するいかなる変更も、影響を受けたデータ表現で自動的に反映されるようとする。

#### 【0065】

グラフィックスエンジン 102 およびデータサーバ 104 を実装する方法例が図 2 に描かれているが、図 2 に示すインタフェース、データ構造、要素、プロセス、および／または装置の 1 つ以上が、その他任意の方法で結合、分割、再編成、省略、除去、および／または実装され得る。例えば、図 2 に示す選択レシーバ例 202、データマネージャ例 204、データリトリーバ例 206、メタデータジェネレータ例 208、メタデータのタイプデータベース例 210、メタデータパーサ例 212、データ表現ジェネレータ例 214、メタデータルールデータベース例 216、データ表現データベース例 218、および／またはデータ表現表示マネージャ例 220 は、例えば、1 つ以上のコンピューティング装置および／またはコンピューティングプラットフォーム（図 13 の処理プラットフォーム例 P10 など）によって実行される機械アクセスまたは読み取り可能な命令を使用して、別々に、および／または任意の組み合わせで実装され得る。

#### 【0066】

さらに、選択レシーバ例 202、データマネージャ例 204、データリトリーバ例 206、メタデータジェネレータ例 208、メタデータのタイプデータベース例 210、メタデータパーサ例 212、データ表現ジェネレータ例 214、メタデータルールデータベース例 216、データ表現データベース例 218、データ表現表示マネージャ例 220、および／またはさらに一般的にはグラフィックスエンジン 102 および／またはデータサーバ 104 は、ハードウェア、ソフトウェア、ファームウェア、および／またはハードウェア、ソフトウェア、および／またはファームウェアの任意の組み合わせによって実装され得る。そのため、例えば、選択レシーバ例 202、データマネージャ例 204、データリトリーバ例 206、メタデータジェネレータ例 208、メタデータのタイプデータベース例 210、メタデータパーサ例 212、データ表現ジェネレータ例 214、メタデータルールデータベース例 216、データ表現データベース例 218、データ表現表示マネージャ例 220、および／またはさらに一般的にはグラフィックスエンジン 102 および／またはデータサーバ 104 のいずれも、1 つ以上の回路、プログラマブルプロセッサ、特定用途向け集積回路（ASIC）、プログラマブル論理デバイス（PLD）、および／またはフィールドプログラマブル論理デバイス（FPLD）などによって実装できる。

#### 【0067】

図 3 は、ユーザインタフェース 302 を介して表示されるバッチベースのプロセス制御システムのデータ表現例 300 を示す。バッチベースのプロセス制御システムは、図 1 のプロセス制御システム 106 を含み得る。本実施例においては、プロセス制御システムがクッキーを製造する場合があり、ここでは各タイプのクッキーが、異なるバッチ（チョコレートチップ、ピーナッツバター、オートミールレーズン、ジンジャースナップなど）と関連付けられている。ユーザインタフェース例 302 は、データ表現 300 を含む表示領域 304 を含む。加えて、ユーザインタフェース 302 はメニューバー 306 も含む。本実施例においては、メニューバー 306 のステータス項目 308 が選択され、データ表現 300 用の表示領域 304 を開く。他の実施例においては、メニューバー 306 に含まれるメニュー選択項目が少ない場合または多い場合がある。加えて、メニューバー 306 は、プロセスデータを表示、管理、修正、カスタマイズ、および／または監視するためのアイコンおよび他の選択可能な機能を含み得る。

#### 【0068】

ユーザインタフェース例 302 は、情報選択バー 310 も含む。情報選択バー 310 から、オペレータおよび／またはエンジニアは表示領域 304 内に表示するプロセス制御情報を指定することができる。本実施例においては、オペレータが、プロセス制御情報（バ

10

20

30

40

50

ツチ情報など)の名前を入力することによってプロセス制御情報を指定し得る。他の実施例においては、オペレータおよび/またはエンジニアが、ファイルディレクトリを閲覧すること、ファイル名を選択すること、概略図の一部分を強調表示すること、機能ダイアグラム内の機能ボックスを強調表示することなどによってプロセス制御情報を選択し得る。

#### 【0069】

オペレータおよび/またはエンジニアが情報選択バー310内でプロセス制御情報を入力すると、図1および/または図2のグラフィックスエンジン102が、「バッチ情報」というプロセス制御情報と関連付けられたプロセスコンポーネントを特定する。グラフィックスエンジン102は、データサーバ104を使用して、関連付けられたプロセスデータを取得し、メタデータを生成する。図3の実施例においては、プロセスデータが、バッチごとにデータストリングまたはデータ語の形態であり得る。例えば、データ表現300の第1の行(チョコレートチップクッキー・バッチなど)のプロセスデータは、「20080229.163450;チョコレートチップクッキー;500ポンドのダークチョコレートチップクッキーを作る;2/29/2008;10:23:31 AM;0:53:45」というデータストリングとして構成され得る。他の実施例においては、プロセスデータがプロセスデータのタイプによって組成され得る。

#### 【0070】

さらに、プロセスデータは、バッチID、レシピ、説明、開始時刻、経過時間の各ヘッダと関連付けられたメタデータを含むメタデータと関連付けられている場合もある。バッチIDヘッダ下のプロセスデータは、例えば、「<Name Batch ID>」、「<Length 14>」、「<Type String>」、「<column>」などを含むメタデータの識別名および/またはタイプと関連付けられている場合がある。同様に、レシピヘッダ下のプロセスデータは、例えば、「<Name Recipe>」、「<Length 25>」、「<Type String>」、「<column>」などを含むメタデータの識別名および/またはタイプと関連付けられている場合がある。同様に、他のヘッダの下のプロセスデータは、メタデータの識別名および/またはタイプと関連付けられている場合がある。

#### 【0071】

図1および/または図2のグラフィックスエンジン102は、プロセスデータと関連付けられたメタデータがテーブルデータ表現タイプと相互参照されていると判断することによって図2のデータ表現300を生成する。次にグラフィックスエンジン102は、どのプロセスデータがテーブルデータ表現タイプのインスタンス内のどのデータフィールドに対応するかを判断する。本実施例において、グラフィックスエンジン102(データ表現ジェネレータ214など)は、5つの異なる列に整理できる5つの異なるタイプのプロセスデータが存在するとメタデータから判断し得る。グラフィックスエンジン102は、各列の表題として、「<Name / Value / >」(「<Name Batch ID>」など)というメタデータ識別名を使用して、各プロセスデータの種類を列に割り当てる。グラフィックスエンジン102はまた、プロセスデータと関連付けられたメタデータを列の識別名に一致させることによって各データフィールド内のプロセスデータをリンクする。さらにグラフィックスエンジン102は、各行のプロセスデータが同じデータストリングおよび/またはデータ語と関連付けられるようにプロセスデータを配列してもよい。次にグラフィックスエンジン102は、プロセスデータを対応データフィールドにリンクし、それによって生成されるテーブルデータ表現300を描画し、その描画をUI提示プロセッサ124に送信して、ユーザインターフェース302経由でデータ表現300を表示する。

#### 【0072】

さらに経過時間と関連付けられたプロセスデータが時間の経過に伴って周期的に変更される場合がある。このプロセスデータに関しては、データサーバ104が、経過時間処理データを取得するためにコントローラ108をサブスクライブしてもよい。その後、経過時間処理データが変更されると、データ表現300は、経過時間処理データの最新の値を

10

20

30

30

40

50

表示することができる。

#### 【0073】

図4は、データ表現400を作成するためにプロセスデータ（装置および段階と関連付けられたプロセスデータなど）が追加された図3のデータ表現例300を示す。図4の実施例においては、図3のデータ表現300が装置および段階のプロセスデータを含むようにオペレータおよび／またはエンジニアが指定してもよい。他の実施例においては、プロセス制御システムに対する変更により、装置および段階と関連付けられたプロセスデータおよび／またはメタデータが「バッチ情報」というプロセス制御情報に追加される場合がある。これらの他の実施例において、グラフィックスエンジン102は、コントローラ108から変更の通知を受信し得る。

10

#### 【0074】

変更を受信すると、データサーバ104は、装置および段階プロセスデータのメタデータを判断し、この情報をグラフィックスエンジン102に転送する。次にグラフィックスエンジン102は、図3のデータ表現300にアクセスし、この装置および段階プロセスデータと関連付けられたメタデータが列データフィールドに対応すると判断し、装置列および段階列を作成する。次にグラフィックスエンジン102は、適切なバッチIDを取得するために装置および段階プロセスデータをデータ表現400にリンクし、ユーザインターフェース302内に表示するためにデータ表現400を再描画する。本実施例においては、装置および段階プロセスデータが、バッチデータのデータ語および／またはデータストリング内に含まれている可能性がある。あるいは、装置および段階プロセスデータが、装置および段階プロセスデータと関連付けられたバッチIDを特定するメタデータを含み得る。

20

#### 【0075】

図5は、データ表現500を作成するためにプロセス制御オペレータによってカスタマイズ情報が実装された図4のデータ表現例400を示す。本実施例においては、オペレータが、メニューバー306のツールタブ502を選択し、テーブル編集ボックス506を表示するためのカスタマイズアイコン504を選択することによってカスタマイズ情報を指定した。テーブル編集ボックス506を使用して、オペレータは、データ表現500のテキストのフォントタイプをTimes New Romanに変更し、バッチIDヘッダと関連付けられたプロセスデータのフォントサイズを上げ、開始時刻ヘッダおよび経過時間ヘッダと関連付けられたプロセスデータのフォントサイズを下げ、レシピヘッダおよび装置ヘッダと関連付けられたプロセスデータのフォントを太字化することができる。加えて、オペレータは、開始時列および経過時間列をバッチID列の隣に移動することができる。他の実施例においては、各種編集ボックスが、あるタイプのデータ表現と関連付けられている編集および／またはカスタマイズ機能を含み得る。例えば、棒グラフデータ表現は、グラフの棒を対象とするカスタマイズ機能を含み得る。

30

#### 【0076】

オペレータによって選択されたカスタマイズ情報を受信すると、図1および／または図2のグラフィックスエンジン102は、図4のデータ表現400を更新し、カスタマイズ情報で指定された変更を反映して、データ表現500を作成する。加えて、グラフィックスエンジン102は、カスタマイズ情報およびデータ表現500をデータ表現データベース218に格納する。いくつかの実施例においては、オペレータがカスタマイズを選択し、実行しているときにグラフィックスエンジン102がデータ表現500を更新する。他の実施例においては、グラフィックスエンジン102が、選択されたカスタマイズ情報を受信し、カスタマイズ情報に基づいてデータ表現500を一度に修正する場合がある。

40

#### 【0077】

図6は、図3から図5のバッチ20080229.163450と関連する変量に対する変数の説明可能な寄与度(explained contribution)のデータ表現例600を示す。本実施例においては、オペレータが、情報選択バー310に情報を入力することによって「バッチ20080229.163450 变数の寄与度」というプロセス制御情報を選択

50

する。変数の説明可能な寄与度に関するデータ表現 600 は、変数の寄与度量および／または平均的なバッチデータからバッチの変量に対するバッチ処理のコンポーネントを示す場合がある。さらに、変数の説明可能な寄与度は、バッチ処理内のプロセス変量から判断され得る変数の寄与度を定義するものであってよい。

#### 【0078】

このプロセス制御情報を受信すると、図 1 および／または図 2 のグラフィックスエンジン 102 は、プロセス制御情報と関連付けられたプロセス制御コンポーネントを判断し、対応するプロセスデータをデータサーバ 104 から取得する。さらにデータサーバ例 104 は、取得されたプロセスデータと関連付けられたメタデータを判断する。本実施例においては、データサーバ 104 が、各変数と関連付けられたメタデータ（「<Variable Name Media Flow>」、「<Variable Name Mixer Temp>」、「<Variable Name Water Temp>」など）を判断する場合がある。メタデータは、プロセスデータの形式（「<Type Decimal>」および／または「<Length 4>」など）も含み得る。

10

#### 【0079】

グラフィックスエンジン例 102 は、メタデータが棒グラフデータ表現タイプと相互参照されていると判断し、棒グラフデータ表現タイプのインスタンスを作成する場合がある。グラフィックスエンジン 102 は、データ表現 600 を作成するためにインスタンス内のプロセスデータをデータフィールドにリンクする目的でこのメタデータを使用する場合がある。次にグラフィックスエンジン 102 は、データ表現 600 を図 6 のユーザインターフェース 302 内に表示するために描画する。

20

#### 【0080】

図 7 は、データ表現 700 を作成するために図 3 から図 5 のバッチと関連付けられた変数の説明不能な寄与度（unexplained contribution）を含む図 6 のデータ表現例 600 を示す。本実施例においては、図 1 および／または図 2 のコントローラ 108 におけるアルゴリズムおよび／またはルーチンに対する更新により、「バッチ 20080229.163450 変数の寄与度」というプロセス制御情報と関連付けられた変量に対するバッチ変数の説明不能な寄与度の計算が再定義および／または追加される場合がある。変数の説明不能な寄与度は、バッチ処理内でプロセスバリエーション変量から判断できない変数の寄与度を定義していくよい。

30

#### 【0081】

図 1 および／または図 2 のグラフィックスエンジン 102 および／またはデータサーバ 104 が変数の寄与度の計算に対する更新または修正の合図を受信すると、データサーバ 104 が、「20080229.163450 変数の寄与度」というプロセス制御情報と関連付けられたプロセスデータおよび／またはメタデータを取得する。次にデータサーバ 104 は、変数の説明不能な寄与度プロセスデータを特定するメタデータを含むプロセスデータと関連付けられたメタデータを判断し得る。次にデータサーバ 104 は、そのメタデータおよびプロセスデータをグラフィックスエンジン 102 に転送する。グラフィックスエンジン 102 は、メタデータおよびプロセスデータと図 6 のデータ表現 600 のデータフィールドとを相互参照して、変数の説明不能な寄与度プロセスデータがどのように表示されるかを判断する。本実施例においては、グラフィックスエンジン 102 は、説明不能および説明可能な変数の寄与度データを変数ごとに 1 本の複合バー内に表示すると判断し得る。次にグラフィックスエンジン 102 は、プロセスデータを適切なデータフィールドにリンクしてデータ表現 700 を作成し、データ表現 700 をユーザインターフェース 302 内で表示するために描画する。このようにして、コントローラ 108 におけるアルゴリズムまたはルーチンに対する更新または変更が、対応するデータ表現 700 で自動的に反映されるため、オペレータは新しいデータ表現を作成する必要がない。

40

#### 【0082】

図 8 は、プロセス制御システムの一部分（図 1 のプロセス制御システム 106 の一部分など）のステータスのデータ表現例 800 を示す。データ表現例 800 は、ミキサー 80

50

6に連結されているポンプ804に連結されたタンク802を示す。データ表現800は、タンクのステータス(例:23.4%充填)、タンク802とポンプ804との間の接続のステータス(例:開)、ポンプ804のステータス(例:速度2.5グラム/秒)、ポンプ804とミキサー806との間の接続のステータス(例:開)、ミキサー806のステータス(例:400回転/秒)、およびミキサー806からの出力のステータス(例:速度2.5グラム/秒)を含む。

#### 【0083】

オペレータおよび/またはエンジニアは、情報選択バー310に「ミキサー概要」プロセス制御情報を入力することによってデータ表現800を作成してもよい。次に図1および/または図2のグラフィックスエンジン102が、選択されたプロセス制御情報と関連付けられたプロセス制御コンポーネント(タンク802、ポンプ804、ミキサー806など)を特定する。次にグラフィックスエンジン102は、これらのコンポーネントを使用して、これらのコンポーネントと関連付けられたプロセスデータを、図1および/または図2のデータサーバ104および/またはコントローラから取得する。さらにデータサーバ104は、取得されたプロセスデータと関連付けられたメタデータを判断する。

10

#### 【0084】

図8の実施例において、タンク802のメタデータは、「<Device Tank>」、「<Connection Pump>」、および/または「<Status Percentage>」を含み得る。同様に、ポンプ804のメタデータは、「<Device Pump>」、「<Connection Tank>」、「<Connection Mixer>」、および/または「<Status Rate>」を含み得る。同様に、ミキサー806のメタデータは、「<Device Mixer>」、「<Connection Pump>」、「<Connection Out>」、および/または「<Status Revolutions>」を含み得る。グラフィックスエンジン例102は、このメタデータを使用してデータ表現タイプを判断する。例えば、グラフィックスエンジン102は、タンク802のメタデータに基づいてタンクデータ表現タイプを、ポンプ804のメタデータに基づいてポンプデータ表現タイプを、そしてミキサー806のメタデータに基づいてミキサーデータ表現タイプを選択し得る。

20

#### 【0085】

次にグラフィックスエンジン102は、タンク802、ポンプ804、およびミキサー806のデータ表現タイプの各インスタンスのデータフィールドに適切なプロセスデータをリンクし得る。グラフィックスエンジン102はまた、「<Connection>」メタデータが、タンク802、ポンプ804、およびミキサー806のデータ表現が結合されることを表すとも判断し得る。次にグラフィックスエンジン102は、データ表現を結合して、データ表現800を作成する。加えて、グラフィックスエンジン102は、データ表現800を、ユーザインターフェース302を介して表示するために描画する。

30

#### 【0086】

図9は、図8のデータ表現例800にステータスチャートデータ表現900が追加され、ユーザインターフェース302内に表示された状態を示す。本実施例においては、オペレータおよび/またはエンジニアが、データ表現800によって通常提供される場合よりも多くのプロセスデータを提供するために、データ表現900の追加を指定してもよい。オペレータは、ツールタブ502を選択し、データ表現アイコン904を選択することによってステータスチャートデータ表現900を表示領域304に加えればよい。データ表現アイコン904を選択することにより、オペレータは、情報選択バー310に「+ステータスチャート」と入力してデータ表現900を表示領域に追加することができる。

40

#### 【0087】

「ステータスチャート」プロセス制御情報を受信すると、図1および/または図2のグラフィックスエンジン例102が、「Status Chart」プロセス制御情報と関連付けられたプロセス制御コンポーネントを判断する。次にデータサーバ104は、判断されたプロセス制御コンポーネントと関連付けられたプロセスデータを取得し、取得され

50

たプロセスデータのメタデータを判断する。本実施例においては、「ステータスチャート」のメタデータが、「<Title CTRL LP01 Overview>」、「<Name Status>」、「<Name Time>」、「<Name Flow>」、および「<Name Faults>」を含み得る。プロセスデータは、メタデータを指定しているデータ型および／またはデータ長と関連付けられている場合もある。

#### 【0088】

次にグラフィックスエンジン102は、判断されたメタデータとチャートデータ表現タイプとを相互参照し、メタデータに基づいてどのプロセスデータがチャートデータ表現タイプのインスタンス内にあるどのデータフィールドと関連付けられているかを特定し、プロセスデータをデータフィールドにリンクしてデータ表現900を作成する。さらにグラフィックスエンジン102は、データ表現900を、ユーザインタフェース302を介して表示するために描画する。加えて、グラフィックスエンジン102は、ステータスチャートデータ表現900をデータ表現800と関連付け、データ表現データベース218内にこの関連付けを格納することによってデータ表現800を修正し得る。このようにして、オペレータが後でデータ表現800を参照することを選択した場合に、ステータスチャートデータ表現900もユーザインタフェース302を介して表示されるため、オペレータがステータスチャートデータ表現900を表示することを明示的に指定する必要がない。

10

#### 【0089】

図10、図11A、および図12は、図1および／または図2の選択レシーバ例202、データマネージャ例204、データリトリーバ例206、メタデータジェネレータ例208、メタデータのタイプデータベース例210、メタデータパーサ例212、データ表現ジェネレータ例214、メタデータルールデータベース例216、データ表現データベース例218、データ表現表示マネージャ例220、および／またはさらに一般的にはグラフィックスエンジン102および／またはデータサーバ104を実装するために行なわれ得る方法例のフローチャートである。図10、図11A、図11B、および／または図12の方法例は、プロセッサ、コントローラ、および／またはその他任意の適切な処理装置によっても実行され得る。例えば、図10、図11A、図11B、および／または図12の方法例は、フラッシュメモリ、CD、DVD、フロッピー（登録商標）ディスク、ROM、RAM、プログラマブルROM（PROM）、電子的にプログラム可能なROM（EPROM）、電子的に消去可能なPROM（EEPROM）、光記憶ディスク、光記憶装置、磁気記憶装置ディスク、磁気記憶装置、および／または方法またはデータ構造の形態でプログラムコードおよび／または命令を担持または格納する目的で使用でき、プロセッサ、汎用または特殊用途のコンピュータ、またはプロセッサを有する他のマシン（図13に関連して後述するプロセッサプラットフォーム例P10など）によってアクセスできるその他任意のメディアなど、任意の有形コンピュータ可読メディアに格納されている符号化された命令で具現化され得る。上記の組み合わせも、コンピュータ可読メディアの範囲内に含まれる。

20

30

#### 【0090】

方法は、例えば、プロセッサ、汎用コンピュータ、特殊用途コンピュータ、または特殊用途処理機に1つ以上の特定の方法を実行させる命令および／またはデータを含む。あるいは、図10、図11A、図11B、および図12の方法例の一部分もしくは全部は、ASIC、PLD、FPLD、個別論理、ハードウェア、ファームウェアなどの任意の組み合わせを用いて実装され得る。

40

#### 【0091】

また、図10、図11A、図11B、および図12の方法例の一部または全部が代わりに手動操作によって実装されたり、ファームウェア、ソフトウェア、個別論理、および／またはハードウェアなどの任意の組み合わせなど、前述の技術の任意の組み合わせとして実装されたりする場合もある。さらに、図10、図11A、図11B、および図12の動作例を実装するその他多数の方法が用いられる場合もある。例えば、ブロックの実行順序

50

が変更される場合があり、かつ／または記載されているブロックの1つ以上が変更、除去、再分割、または結合される場合がある。加えて、図10、図11A、図11B、および／または図12の方法例の一部または全部が、例えば、個別の処理スレッド、プロセッサ、デバイス、個別論理、回路などによって順次および／または並行して実行される場合もある。

#### 【0092】

図10の方法例1000は、選択されたプロセス制御情報に基づいて動的なデータ表現を生成する。複数の方法例1000が、複数のデータ表現を生成するために、並行または連続して実行され得る。加えて、複数の用途がプロセス制御情報を選択する実施例においては、方法例1000がユーザごとに実行されたり、単一の方法例1000がユーザ全員を対象に実行されたりする場合もある。10

#### 【0093】

図10の方法例1000は、ユーザからプロセス制御情報の選択を（図2の選択レシーバ202などを介して）受信することから始まる（ブロック1002）。方法例1000は、選択されたプロセス制御情報と関連付けられたプロセスデータを（データマネージャ204などを介して）特定する（ブロック1004）。このプロセスデータは、プロセス制御情報と関連付けられている可能性があるプロセス制御コンポーネントから判断および／または特定されてもよい。次に方法例1000は、判断されたプロセスデータを（データリトリーバ206などを介して）読み込みおよび／またはサブスクライブする（ブロック1006）。次に方法例1000は、データサーバ104でこのプロセスデータを受信する（ブロック1008）。20

#### 【0094】

図10の方法例1000は、受信プロセスデータと関連付けられたメタデータを（メタデータジェネレータ208などを介して）判断することによって処理を続行する（ブロック1010）。方法例1000は、プロセスデータと関連付けられており、かつ／またはプロセスデータ内に組み込まれている可能性のあるメタデータを判断する場合があり、あるいはプロセスデータのタイプおよび／または形式に基づいてメタデータを作成および／または生成する場合がある。次に方法例1000は、メタデータおよび関連付けられたプロセスデータをグラフィックスエンジン102に転送する（ブロック1012）。次に方法例1000は、（メタデータパーサ212などを介して）メタデータを解析する（ブロック1014）。グラフィックスエンジン102がデータサーバ104内に含まれる他の実施例においては、方法例1000が、メタデータを判断すると同時にメタデータを解析する場合がある。30

#### 【0095】

メタデータを解析すると、方法例1000は、関連付けられたメタデータに基づいて（データ表現ジェネレータ214などを介して）データ表現を生成し、プロセスデータを表示する（ブロック1016）。方法例1000は、表示されるメタデータおよび／またはプロセスデータのグループおよび／または部分ごとにデータ表現を生成する場合がある。方法例1000がデータ表現を生成する方法に関するさらに詳しい説明を、図11Aおよび11B内で合わせて後述する。データ表現を生成した後、方法例1000は、ユーザインターフェースを介して、（UI提示プロセッサ124などを介して）データ表現を描画し（データ表現表示マネージャ220などを介して）表示する（ブロック1018）。40

#### 【0096】

次に方法例1000は、表示されたデータ表現のいずれかをユーザがカスタマイズしたかどうかを判断する（ブロック1020）。方法例1000は、UI提示プロセッサ124からカスタマイズ情報を受信することによって、および／またはデータインターフェースをポーリングしてデータ表現が変更および／または修正されたかどうかを判断することによって、ユーザがデータ表現をカスタマイズしたかどうかを判断し得る。方法例1000は、ユーザが少なくとも1つのデータ表現もカスタマイズしていないと判断すると、別のワークステーションで同じユーザおよび／または異なるユーザからプロセス制御情報の選択50

を受信することに戻る（ブロック 1002）。

#### 【0097】

ただし方法例 1000 は、ユーザが少なくとも 1 つのデータ表現（ブロック 1020）をカスタマイズしたと判断すると、カスタマイズ情報に基づいてデータ表現を表示する（ブロック 1022）。加えて、方法例 1000 は、関連付けられたデータ表現、データ表現タイプ、および／またはプロセスデータを有するカスタマイズ情報を（データ表現ジェネレータ 214などを介して）データ表現データベース 218 に格納する（ブロック 1024）。次に方法例 1000 は、別のワークステーションで同じユーザおよび／または異なるユーザからプロセス制御情報の選択を受信することに戻る場合がある（ブロック 1002）。

10

#### 【0098】

図 11A および 11B の方法例 1016 は、図 10 に記載のとおり、受信メタデータに基づいてデータ表現を生成する。方法例 1016 は、図 2 のデータ表現ジェネレータ 214 によって実装および／または実行され得る。複数の方法例 1016 は、複数のデータ表現を生成するために、並行または連続して実行され得る。加えて、メタデータの複数部分が受信される実施例においては、方法例 1016 が部分ごとに実装されるか、あるいは単一の方法例 1016 がすべての部分に対して実装され得る。

#### 【0099】

図 11A の方法例 1016 は、メタデータを受信することから始まる（ブロック 1102）。メタデータは、図 2 のメタデータパーサ 212 から送信され得る。次に方法例 1016 は、受信メタデータのタイプを特定する（ブロック 1104）。メタデータのタイプは、メタデータ内に含まれるメタデータ識別名、コード語、および／または記号によって特定され得る。次に方法例 1016 は、図 2 のメタデータルールデータベース 216 で、各メタデータタイプとデータ表現タイプとを相互参照する（ブロック 1106）。単一のデータ表現内に表示されるように指定されたプロセスデータの一部分と複数のメタデータタイプが関連付けられている実施例においては、方法例 1016 が、複数のメタデータタイプと一致および／または相互参照しているデータ表現タイプを選択する場合がある。

20

#### 【0100】

次に図 11A の方法例 1016 は、一致したデータ表現タイプをコンパイルし（ブロック 1108）、プロセスデータの同一部分と関連付けられた 1 つよりも多くのデータ表現タイプが存在するかどうかを判断する（ブロック 1110）。そのプロセスデータに対応する 1 つよりも多くのデータ表現タイプが存在し得る場合、図 11B の方法例 1016 はデータ表現データベース 218 にアクセスし、どのデータ表現タイプが選択されるべきかを示す事前カスタマイズ情報が存在するかどうかを判断する（ブロック 1112）。事前カスタマイズ情報がなく、かつ／または選択すべきデータ表現タイプを指定し得る他の情報もない場合、方法例 1016 は、ユーザ（オペレータまたはエンジニアなど）にいずれかのデータ表現タイプを選択するよう促す（ブロック 1114）。次に方法例 1016 は、明示されたデータ表現タイプを選択する（ブロック 1116）。ただし、事前カスタマイズ情報が存在する場合（ブロック 1112）、方法例 1016 は、明示されたデータ表現タイプを自動的に選択する（ブロック 1116）。他の実施例においては、方法例 1016 が、一致するすべてのデータ表現タイプを選択する場合がある。

30

#### 【0101】

図 11A の方法例 1016 は、選択されたデータ表現タイプごとにデータ表現のインスタンスを生成することによって処理を続行する（ブロック 1118）。加えて、方法例 1016 は、そのプロセスデータのデータ表現タイプが 1 つしかないと判断すると（ブロック 1110）、そのデータ表現タイプのデータ表現のインスタンスを生成する（ブロック 1118）。次に方法例 1016 は、生成されたデータ表現タイプの各インスタンス内にあるデータフィールドに対応するプロセスデータを特定する（ブロック 1120）。プロセスデータを特定することは、データ表現に一致するメタデータとどのプロセスデータが関連付けられているかを判断することを含み得る。加えて、プロセスデータを特定するこ

40

50

とは、各データフィールドとどのプロセスデータが関連付けられているかを判断することを含み得る。次に方法例 1016 は、プロセスデータを受信し、そのプロセスデータをデータ表現タイプのインスタンスにある対応データフィールドと関連付けてデータ表現を作成する。プロセスデータを関連付けることは、そのプロセスデータをデータフィールドに挿入すること、および / またはプロセスデータの位置を適切なデータフィールドに挿入することによってプロセスデータをリンクすることを含み得る。データ表現を生成すると、方法例 1016 は終了する。

#### 【0102】

図 12 の方法例 1200 は、メタデータおよび / またはプロセスデータに対する変更に基づいて動的にデータ表現を更新する。複数の方法例 1200 は、複数のデータ表現を更新するために、並行または連続して実行され得る。加えて、複数の用途がメタデータおよび / またはプロセスデータを選択する実施例においては、方法例 1200 がユーザごとに実装されたり、単一の方法例 1200 がユーザ全員を対象に実装されたりする場合もある。  
10

#### 【0103】

図 12 の方法例 1200 は、ユーザインターフェースを介して表示される 1 つ以上のデータ表現と関連付けられているメタデータおよび / またはプロセスデータに対する変更を(データ表現表示マネージャ 220などを介して)受信することから始まる(ブロック 1202)。変更されたメタデータおよび / またはプロセスデータは、現場装置、コントローラ、ルーチン、および / またはプロセス制御システムに対する変更に起因する場合がある。図 12 の実施例においては、変更されたメタデータおよび / またはプロセスデータの明示が図 1 および図 2 のコントローラ 108 によって送信される場合がある。あるいは、ユーザが、表示されているデータ表現を修正することによって、および / またはメタデータの定義および / またはプロセスデータを変更することによってプロセスデータおよび / またはメタデータを変える場合もある。本実施例においては、変更されたメタデータおよび / またはプロセスデータの明示が UI 提示プロセッサ 124 によって送信される場合がある。他の実施例においては、方法例 1200 が、コントローラ 108 および / または UI 提示プロセッサ 124 をポーリングして、データ表現に影響を及ぼす可能性のあるメタデータおよび / またはプロセスデータに対する変更の有無を確認する場合がある。  
20

#### 【0104】

次に図 12 の方法例 1200 は、メタデータまたはプロセスデータに変更があったかどうかを判断する(ブロック 1204)。プロセスデータに変更があった場合、方法例 1200 は、変更されたプロセスデータと関連付けられたメタデータを(メタデータジェネレータ 208 および / またはデータ表現ジェネレータ 214などを介して)生成する(ブロック 1206)。次に方法例 1200 は、新たに生成されたメタデータおよび変更されたプロセスデータをグラフィックスエンジン 102 に転送して、影響を受けたデータ表現を更新する(ブロック 1208)。方法例 1200 は、メタデータを解析する場合もある(ブロック 1210)。  
30

#### 【0105】

メタデータを解析すると、方法例 1200 は、関連付けられたメタデータに基づいて、変更されたプロセスデータと関連付けられたデータ表現を生成および / または修正する(ブロック 1214)。方法例 1200 は、変更されたプロセスデータを生成済みのデータ表現に一致させることによってデータ表現を特定する。データ表現を修正した後、方法例 1200 は、データ表現を描画および / または更新し、ユーザインターフェースを介してそのデータ表現を表示する(ブロック 1214)。  
40

#### 【0106】

次に方法例 1200 は、修正されたデータ表現のいずれかでもユーザがカスタマイズしたかどうかを判断する(ブロック 1216)。方法例 1200 は、ユーザが少なくとも 1 つの修正されたデータ表現もカスタマイズしていないと判断すると、別のワークステーションで同じユーザおよび / または異なるユーザからメタデータおよび / またはプロセスデ  
50

ータに対する変更の選択を受信することに戻る（ブロック 1202）。

#### 【0107】

ただし、方法例 1200 は、ユーザが少なくとも 1 つのデータ表現をカスタマイズしたと判断すると（ブロック 1216）、カスタマイズ情報を、関連付けられたデータ表現、データ表現タイプ、および／またはプロセスデータと共にデータ表現データベース 218 に格納する（ブロック 1218）。加えて、方法例 1200 は、カスタマイズおよび／または修正されたデータ表現を表示する（ブロック 1220）。次に方法例 1200 は、別のワークステーションで同じユーザおよび／または異なるユーザのいずれかからメタデータおよび／またはプロセスデータに対する変更の選択を受信することに戻る場合がある（ブロック 1202）。

10

#### 【0108】

方法例 1200 は、メタデータで変更があったと判断すると（ブロック 1204）、変更されたメタデータと関連付けられたプロセス制御情報および／またはコンポーネントを特定する（ブロック 1230）。方法例 1200 は、変更されたメタデータを解析する場合もある（ブロック 1232）。次に方法例 1200 は、対応する変更されたメタデータに基づいて 1 つ以上のデータ表現を特定し、修正する（ブロック 1234）。次に方法例 1200 は、1 つ以上の修正されたデータ表現の表示を描画および／または更新する（ブロック 1236）。次に方法例 1200 は、修正されたデータ表現のいずれかをユーザがカスタマイズしたかどうかを判断する（ブロック 1216）。方法例 1200 は、ユーザが少なくとも 1 つの修正されたデータ表現もカスタマイズしていないと判断すると、別のワークステーションで同じユーザおよび／または異なるユーザのいずれかからメタデータおよび／またはプロセスデータに対する変更の選択を受信することに戻る（ブロック 1202）。

20

#### 【0109】

ただし、方法例 1200 は、ユーザが少なくとも 1 つのデータ表現をカスタマイズしたと判断すると（ブロック 1216）、カスタマイズ情報を、関連付けられたデータ表現、データ表現タイプ、および／またはプロセスデータと共にデータ表現データベース 218 に格納する（ブロック 1218）。加えて、方法例 1200 は、カスタマイズおよび／または修正されたデータ表現を表示する（ブロック 1220）。次に方法例 1200 は、別のワークステーションで同じユーザおよび／または異なるユーザのいずれかからメタデータおよび／またはプロセスデータに対する変更の選択を受信することに戻る場合がある（ブロック 1202）。

30

#### 【0110】

図 13 は、本明細書に記載の方法例および装置例を実装する目的で使用され得るプロセッサシステム例 P10 のブロック図である。例えば、プロセッサシステム例 P10 と類似または同一のプロセッサシステムが、図 1 および／または図 2 の選択レシーバ例 202、データマネージャ例 204、データリトリーバ例 206、メタデータジェネレータ例 208、メタデータのタイプデータベース例 210、メタデータパーサ例 212、データ表現ジェネレータ例 214、メタデータルールデータベース例 216、データ表現データベース例 218、データ表現表示マネージャ例 220、および／またはさらに一般的にはグラフィックスエンジン 102 および／またはデータサーバ 104 を実装する目的で使用され得る。プロセッサシステム例 P10 は、複数の周辺機器、インターフェース、チップ、メモリなどを含むものとして後述されているが、これらの要素のうちの 1 つ以上が、選択レシーバ例 202、データマネージャ例 204、データリトリーバ例 206、メタデータジェネレータ例 208、メタデータのタイプデータベース例 210、メタデータパーサ例 212、データ表現ジェネレータ例 214、メタデータルールデータベース例 216、データ表現データベース例 218、データ表現表示マネージャ例 220、および／またはより一般的にはグラフィックスエンジン 102、および／またはデータサーバ 104 のうちの 1 つ以上を実装する目的で使用される他のプロセッサシステム例から省略されている場合がある。

40

50

**【 0 1 1 1 】**

図13に表すとおり、プロセッサシステムP10は、相互接続バスP14に連結されているプロセッサP12を含む。プロセッサP12は、レジスタセットまたはレジスタースペースP16を含む。これは、図13で完全にオンチップであるものとして描かれているが、完全または部分的にオフチップであり、専用電気接続および／または相互接続バスP14を介して直接プロセッサP12に連結されている可能性もある。プロセッサP12は、任意の適切なプロセッサ、処理機器、またはマイクロプロセッサであってもよい。図13に表されていないが、システムP10は、マルチプロセッサシステムであってもよく、したがって、プロセッサP12と同一または類似であり、相互接続バスP14に通信可能に連結されている1つ以上の追加プロセッサを含み得る。

10

**【 0 1 1 2 】**

図13のプロセッサP12はチップセットP18に連結されており、メモリコントローラP20および周辺入出力(I/O)コントローラP22を含む。周知のとおり、チップセットは、I/Oおよびメモリ管理機能に加え、チップセットP18に連結された1つ以上のプロセッサによってアクセス可能または使用される複数の汎用および／または特殊用途のレジスタやタイマーなども提供するのが一般的である。メモリコントローラP20は、プロセッサP12(複数のプロセッサがある場合にはプロセッサ)がシステムメモリP24および大容量記憶メモリP25にアクセスできるようにする機能を実行する。

20

**【 0 1 1 3 】**

システムメモリP24は、例えば静态RAM(SRAM)、ダイナミックランダムアクセスメモリ(DRAM)、フラッシュメモリ、読み取り専用メモリ(ROM)など、いかなる所望のタイプの揮発性および／または不揮発メモリも含み得る。大容量記憶メモリP25は、いかなる所望のタイプの大量記憶装置も含み得る。例えば、グラフィックスエンジン102および／またはデータサーバ104(図2)を実装する目的でプロセッサシステム例P10が使用される場合、大容量記憶メモリP25は、ハードディスクドライブ、光学ドライブ、テープ記憶装置などを含み得る。あるいは、メタデータのタイプデータベース例210、メタデータルールデータベース例216、および／またはデータ表現データベース例218を実装する目的でプロセッサシステム例P10が使用される場合、大容量記憶メモリP25は、ソリッドステートメモリ(フラッシュメモリやRAMメモリなど)、磁気メモリ(ハードドライブなど)、あるいはメタデータのタイプデータベース例210、メタデータルールデータベース例216、および／またはデータ表現データベース例218における大容量記憶に適したその他いかなるメモリも含み得る。

30

**【 0 1 1 4 】**

周辺I/OコントローラP22は、プロセッサP12が周辺I/OバスP32を介して周辺入出力(I/O)装置P26およびP28、およびネットワークインタフェースP30と通信できるようにする機能を実行する。入出力装置P26およびP28は、例えば、キーボード、表示装置(液晶ディスプレイLCD)、ブラウン管(CRT)ディスプレイなど)、ナビゲーション装置(マウス、トラックボール、容量タッチパッド、ジョイスティックなど)など、いかなる所望のタイプの入出力装置でもあってもよい。ネットワークインタフェースP30は、例えば、プロセッサシステムP10が他のプロセッサシステムと通信できるようにするイーサネット(登録商標)装置、非同期転送モード(ATM)装置、802.11装置、DSLモデム、ケーブルモデム、セルラー方式モデムなどであり得る。

40

**【 0 1 1 5 】**

メモリコントローラP20およびI/OコントローラP22は、チップセットP18内の別々の機能ブロックとして図13に描かれているが、これらのブロックによって実行される機能は、単一の半導体回路の範囲内で統合されている場合、または2台以上の別々の集積回路を使用して実装されている場合がある。

**【 0 1 1 6 】**

上記方法例および／または装置例の少なくともいくつかは、コンピュータプロセッサに

50

よって実行される1つ以上のソフトウェアおよび／またはファームウェアプログラムによって実装される。ただし、特定用途向け集積回路、プログラマブル論理配列、およびその他のハードウェア装置を含むがこれらに限定されない専用ハードウェア実装も同様に、本明細書に記載の方法例および／または装置例のいくつかまたはすべてを全体的または部分的に実装するように構築され得る。さらに、分散処理またはコンポーネント／オブジェクト分散処理、並列処理、または仮想マシン処理を含むがこれらに限定されない代替ソフトウェア実装も、本明細書に記載の方法例および／またはシステム例を実装するように構成され得る。

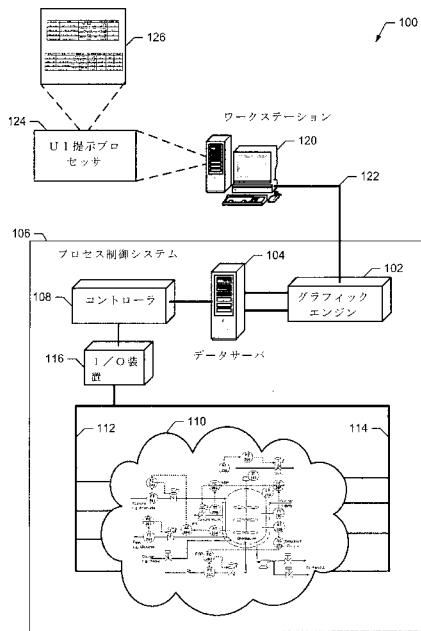
#### 【0117】

また、本明細書に記載のソフトウェア例および／またはファームウェア例の実装は、磁気メディア（磁気ディスクやテープなど）、光ディスクなどの光磁気もしくは光学メディア、1つ以上の読み取り専用（不揮発性）メモリ、ランダムアクセスメモリ、あるいはその他の書き換え可能な（揮発性）メモリを収納するメモリカードまたはその他のパッケージなどのソリッドステートメディアなどの有形記憶メディアに格納されることにも留意すべきである。したがって、本明細書に記載のソフトウェア例および／またはファームウェア例は、上記または後継の記憶メディアなどの有形記憶メディアに格納され得る。上記の仕様が特定の標準仕様およびプロトコルに関してコンポーネント例および機能について記載する範囲で、本特許の範囲はかかる標準仕様およびプロトコルに限定されないと理解される。例えば、インターネットおよび他のパケット交換ネットワーク伝送（Transmission Control Protocol（TCPなど）/Internet Protocol（IP）、User Datagram Protocol（UDP）/IP、HyperText Markup Language（HTML）、HyperText Transfer Protocol（HTTP）など）の各標準仕様は、当該技術の現況の実施例を表す。かかる標準仕様は、同じ一般機能を有しつつ、より高速または効率的な均等仕様によって周期的に取って代わられる。そのため、同じ機能を有する代替標準仕様およびプロトコルは、本特許によって想定され、添付の請求範囲の範囲内に含まれるものと意図される均等物である。

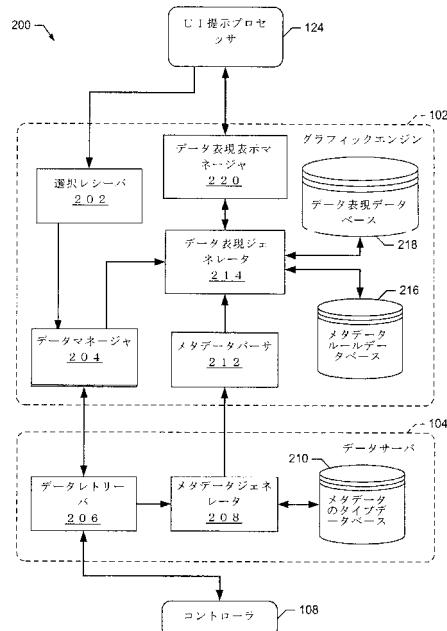
#### 【0118】

加えて、本明細書は、ハードウェアで実行されるソフトウェアまたはファームウェアを含んでいる方法例および装置例を開示しているが、かかるシステムは単なる例に過ぎず、制限的なものとみなされるべきでないことに留意する必要がある。例えば、これらのハードウェアおよびソフトウェアコンポーネントのいずれかまたは全部が、ハードウェアでのみ、ソフトウェアでのみ、ファームウェアでのみ、あるいはハードウェア、ファームウェア、および／またはソフトウェアの任意の組み合わせで具現化され得ると考えられる。したがって、方法例、システム例、機械アクセス可能なメディア例について上記仕様で説明したものの、実施例は、かかるシステム、方法、および機械アクセス可能なメディアを実装するための唯一の方法ではない。そのため、特定の方法例、システム例、および機械アクセス可能なメディア例が本明細書に記載されているものの、本特許の網羅する範囲はそれらに制限されない。むしろ、本明細書は、文字どおりまたは均等論の下で、添付の請求の範囲に正当に含まれるすべての方法、システム、および機械アクセス可能なメディアを網羅する。

〔 図 1 〕



【 図 2 】

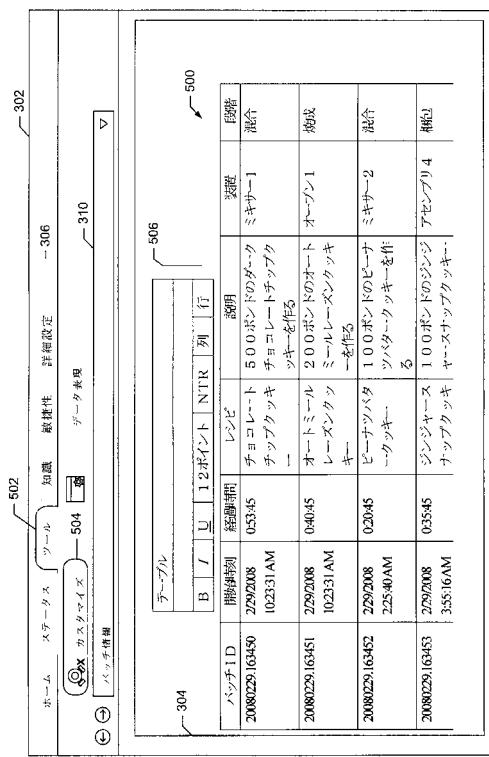


〔 図 3 〕

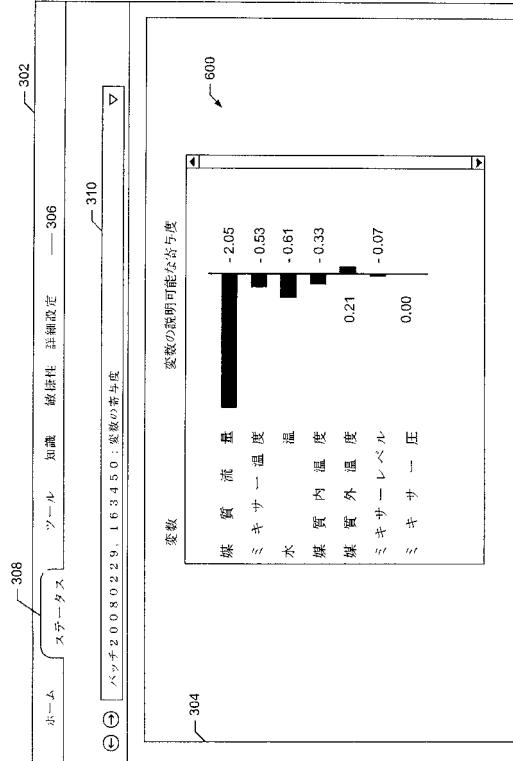
【 図 4 】

308		309		310		311		312	
ホーム	スタートス	ツール	知識	属性	詳細性	詳細設定	-306	-307	-400
▽ ベッタ作成 ▽									
バッチID	レジン	透明	樹脂	経時変動	樹脂	樹脂	ミキサー1	混合	
20080229163450	チョコレートチップ ブラック	5.00ボンドのダーツチップ レートチップクリーナーを作る	22/9/2008 10:23:31AM	0.5345	オーブン1				
20080229163451	オートミルレーレー <sup>スクリュー</sup>	2.00ボンドのオートミルレーレークリーナーを作る	22/9/2008 10:23:31AM	0.0365					
20080229163452	ビーナッツバター <sup>スクリュー</sup>	1.00ザル(1/2)ビーナッツバターキーナーを作る	22/9/2008 22:54:00AM	0.2045					
20080229163453	ショコラ <sup>スクリュー</sup>	1.00ザル(1/2)ショコラクリーナーを作る	22/9/2008 3:55:45AM	0.3545	アセンブリ4	組立			

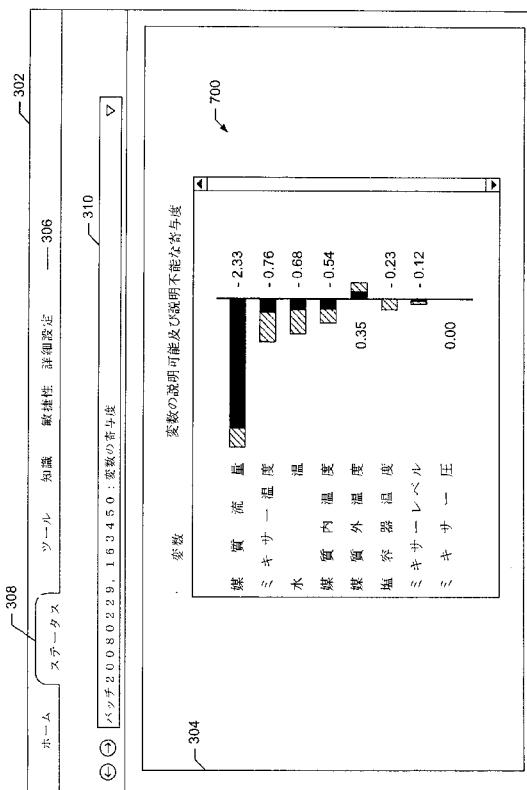
【 図 5 】



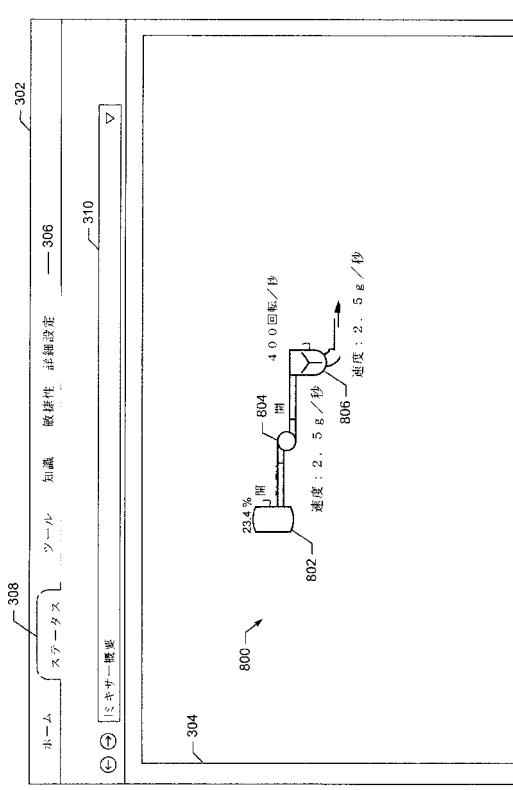
〔 四 6 〕



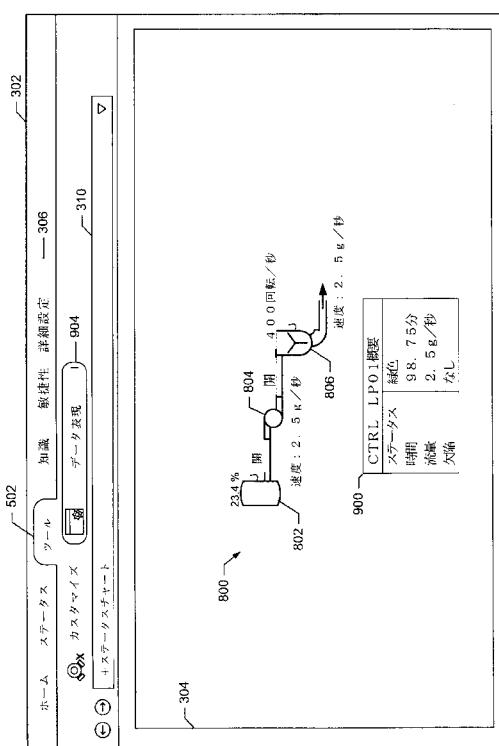
【 四 7 】



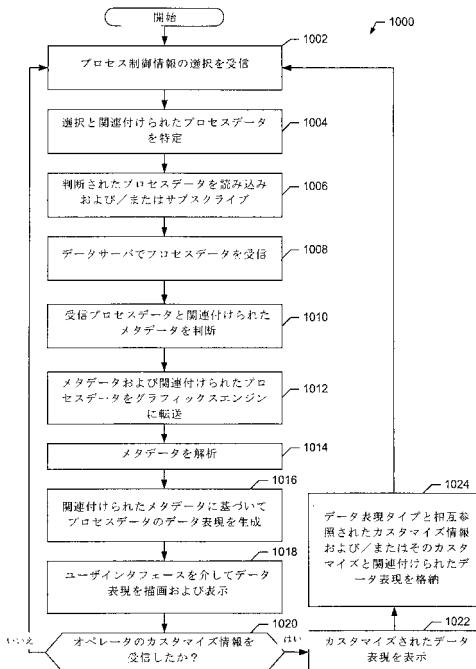
【 8 】



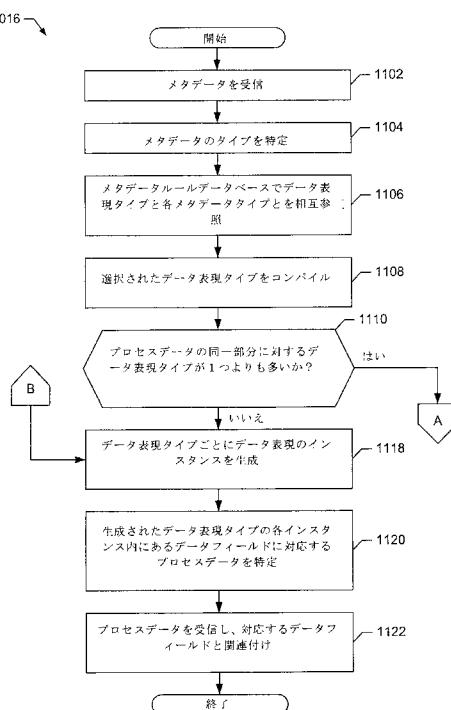
【図 9】



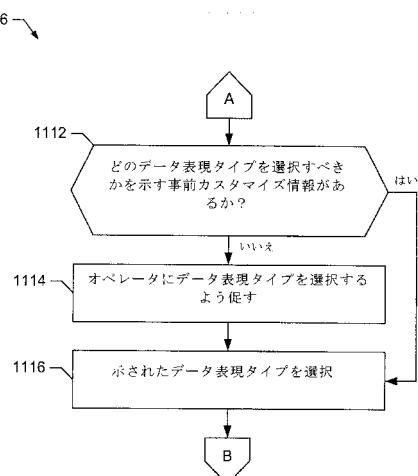
【図 10】



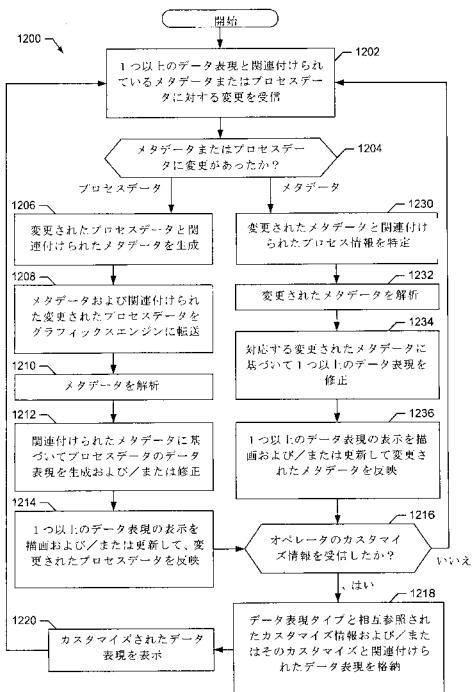
【図 11A】



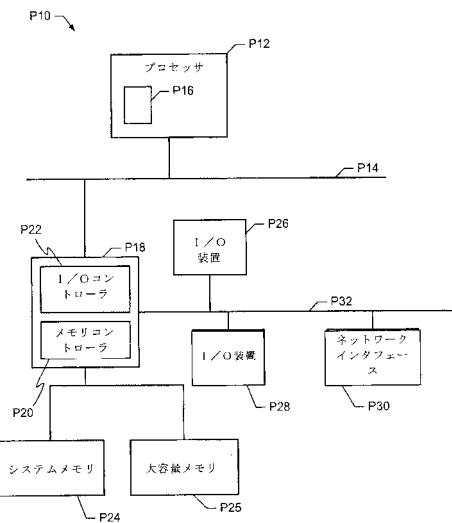
【図 11B】



【図12】



【図13】



---

フロントページの続き

(72)発明者 リン チョウ

アメリカ合衆国 78750 テキサス州 オースティン ウィンチエルシー ドライブ 106  
22

(72)発明者 ジェイムズ ヘンリー ムーア ジュニア

アメリカ合衆国 78628 テキサス州 ジョージタウン オーク プラザ ドライブ 140

F ターム(参考) 5E501 AC02 BA12 DA13 FA46

5H223 AA01 DD03 EE30