

(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2017-182832  
(P2017-182832A)

(43) 公開日 平成29年10月5日(2017.10.5)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 3/0484 (2013.01)</b>	G06F 3/0484 120	3C223
<b>G05B 23/02 (2006.01)</b>	G05B 23/02 301T	5E555
<b>G06F 3/0486 (2013.01)</b>	G06F 3/0484 170	
	G06F 3/0486	

審査請求 有 請求項の数 29 O L (全 65 頁)

(21) 出願番号 特願2017-111942 (P2017-111942)  
 (22) 出願日 平成29年6月6日(2017.6.6)  
 (62) 分割の表示 特願2014-178883 (P2014-178883)の分割  
 原出願日 平成17年5月4日(2005.5.4)  
 (31) 優先権主張番号 60/567,980  
 (32) 優先日 平成16年5月4日(2004.5.4)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 512132022  
 フィッシャー-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 テキサス州 78681 ラウンドロック ウェスト ルイスヘナ ブルバード 1100 ビルディング 1  
 (74) 代理人 100079049  
 弁理士 中島 淳  
 (74) 代理人 100084995  
 弁理士 加藤 和詳  
 (72) 発明者 ルーカス、マイケル、ジェー、イギリス国 エルイー17 5イーエス レスターシェアー レール ラターワース ウェールズ オーチャード 7  
 最終頁に続く

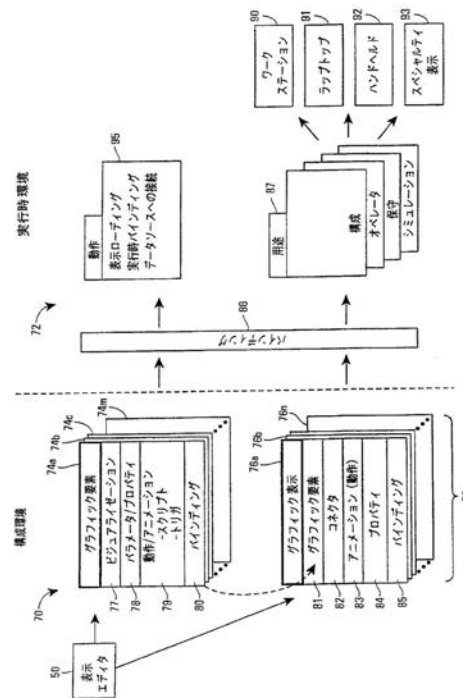
(54) 【発明の名称】 プロセス環境における関連付けられたグラフィック表示、表示エディタプログラム、表示オブジェクト、及びプロセス制御システム

(57) 【要約】 (修正有)

【課題】 プロセスプラント端末のグラフィカルユーザインタフェースを、実行時環境の下で、編集、更新が可能なエディタを提供する。

【解決手段】 グラフィック表示エディタは、実行時環境で、プロセスプラント内のオペレータビューイング機能、シミュレーション機能および保守機能のための表示を提供できる、相互関連グラフィック表示を生成するのに使用できる。グラフィック表示を生成するのに同じグラフィックエディタが使用されているため、結果として生ずるグラフィック表示は同じ様相および印象を有することができ、プラント内の実行時環境に全般的に同じ方法でバインドされることができ、プロセスプラントで使用されるプロセスグラフィック表示を構成及び生成するのに必要とされる時間を低減する。

【選択図】 図3



**【特許請求の範囲】****【請求項 1】**

プロセスプラント内の構成要素の動作を表し且つ各々他の機能及びインタフェースとは独立して実行されるように開発された、複数のインタフェースで実行される複数の機能に共通のプログラムである表示オブジェクトを生成する表示エディタプログラムであって、前記表示エディタプログラムは、

ユーザが、操作手段を表示デバイスのスクリーンに対して操作して、前記スクリーンに表示された、前記プロセスプラント内の前記構成要素の複数の視覚的特徴のライブラリから、前記視覚的特徴を編集キャンパス上に配置し、

ユーザが、前記操作手段を前記スクリーンに対して操作して、前記配置された前記視覚的特徴に対してプロパティを定義し、

ユーザが、前記操作手段を前記スクリーンに対して操作して、前記プロパティと前記構成要素とを結び付けることを可能にすることにより、

インスタンス化されて実行可能なプログラムであるクラスオブジェクトとして、表示オブジェクトを生成する、表示エディタプログラム。

**【請求項 2】**

前記表示オブジェクトは、ユーザにより指示されると、前記プロセスプラントに関する情報を表示する更なる表示オブジェクトを呼び出す呼び出し表示プログラムを備え、

前記表示エディタプログラムは、前記呼び出し表示プログラムがユーザにより指示されると、前記更なる表示オブジェクトを呼び出し且つユーザが前記更なる表示オブジェクトへ情報を入力することを可能にする、請求項 1 に記載の表示エディタプログラム。

**【請求項 3】**

前記更なる表示オブジェクトは、前記構成要素についての情報を提供する、請求項 2 に記載の表示エディタプログラム。

**【請求項 4】**

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の情報を表示するプログラムを含み、

前記更なる表示オブジェクトは、前記プロセスプラントの部分の第 2 の情報を表示するプログラムを含み、

前記第 1 の情報は、前記プロセスプラントの部分について、前記第 2 の情報のデータソースとは異なるデータソースから得られる、請求項 1 に記載の表示エディタプログラム。

**【請求項 5】**

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、

前記更なる表示オブジェクトは、前記プロセスプラントの、前記第 1 の構成要素とは異なる第 2 の構成要素を表示する、請求項 1 に記載の表示エディタプログラム。

**【請求項 6】**

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、

前記更なる表示オブジェクトは、前記プロセスプラントの第 1 の構成要素の詳細な情報を表示する、

請求項 1 に記載の表示エディタプログラム。

**【請求項 7】**

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の機能的目的のための第 1 の表示を提供するプログラムを含み、

前記更なる表示オブジェクトは、前記プロセスプラントの部分の、前記第 1 の機能的目的とは異なる第 2 の機能的目的のための第 2 の表示を提供するプログラムを含み、

請求項 1 に記載の表示エディタプログラム。

10

20

30

40

50

**【請求項 8】**

前記第 1 の機能的目的のための前記第 1 の表示は、制御オペレータ表示、保守表示、ビジネス表示、シミュレーション表示、及びエンジニアリング表示のうちの何れか一つの表示であり、

前記第 2 の機能的目的のための前記第 2 の表示は、前記制御オペレータ表示、前記保守表示、前記ビジネス表示、前記シミュレーション表示、及び前記エンジニアリング表示のうちの、前記一つの表示以外の何れか一つの表示である、

請求項 7 に記載の表示エディタプログラム。

**【請求項 9】**

前記制御オペレータ表示は、前記プロセスプラントの運転中に一つ以上の構成要素と関連するプロセス値を示す、請求項 8 に記載の表示エディタプログラム。

10

**【請求項 10】**

前記保守表示は、前記プロセスプラント内の構成要素の健全性またはステータスの表示を示す、請求項 8 又は請求項 9 に記載の表示エディタプログラム。

**【請求項 11】**

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記更なる表示オブジェクトは、前記プロセスプラント内で実行される制御ルーチンを示す表示にユーザがアクセスすることを可能にする、請求項 1 に記載の表示エディタプログラム。

**【請求項 12】**

20

前記表示オブジェクトには、ユーザにより定義可能な、前記表示オブジェクトの実行中における前記視覚的特徴の動作のプログラムが含まれる、請求項 1 に記載の表示エディタプログラム。

**【請求項 13】**

前記動作は、アニメーションによって定義される、請求項 1 に記載の表示エディタプログラム。

**【請求項 14】**

プロセスプラント内の構成要素の動作を表し且つ各々他の機能及びインタフェースとは独立して実行されるように開発された、複数のインタフェースで実行される複数の機能に共通のプログラムである表示オブジェクトであって、

30

前記表示オブジェクトは、インスタンス化されて実行可能なクラスオブジェクトであり、

前記表示オブジェクトは、

前記プロセスプラント内の前記構成要素の複数の視覚的特徴のライブラリから視覚的特徴を、ユーザが操作手段を表示デバイスのスクリーンに対して操作することにより、選択することを可能にするプログラムと、

前記ユーザにより選択された前記視覚的特徴に対してプロパティを、ユーザが、前記操作手段を前記スクリーンに対して操作することにより、定義することを可能にするプログラムと、

を含み、

40

前記プロパティと前記構成要素とが、ユーザによって対応付けられている、表示オブジェクト。

**【請求項 15】**

前記表示オブジェクトは、ユーザにより指示されると、前記プロセスプラントに関する情報を表示する更なる表示オブジェクトを呼び出す呼び出し表示プログラムを備える、請求項 14 に記載の表示オブジェクト。

**【請求項 16】**

前記更なる表示オブジェクトは、前記構成要素についての情報を提供する、請求項 15 に記載の表示オブジェクト。

**【請求項 17】**

50

請求項 1 4 に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の情報を表示し、  
前記更なる表示オブジェクトは、前記プロセスプラントの部分の第 2 の情報を表示し、  
前記第 1 の情報は、前記プロセスプラントの部分について、前記第 2 の情報のデータソースとは異なるデータソースから得られる、一組の表示オブジェクト。

【請求項 1 8】

請求項 1 4 に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、  
前記更なる表示オブジェクトは、前記プロセスプラントの、前記第 1 の構成要素とは異なる第 2 の構成要素を表示する、一組の表示オブジェクト。

10

【請求項 1 9】

請求項 1 4 に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、  
前記更なる表示オブジェクトは、前記プロセスプラントの第 1 の構成要素の詳細な情報を表示する、  
一組の表示オブジェクト。

20

【請求項 2 0】

請求項 1 4 に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の機能的目的のための第 1 の表示を提供し、  
前記更なる表示オブジェクトは、前記プロセスプラントの部分の、前記第 1 の機能的目的とは異なる第 2 の機能的目的のための第 2 の表示を提供する、  
一組の表示オブジェクト。

30

【請求項 2 1】

前記第 1 の機能的目的のための前記第 1 の表示は、制御オペレータ表示、保守表示、ビジネス表示、シミュレーション表示、及びエンジニアリング表示のうちの何れか一つの表示であり、  
前記第 2 の機能的目的のための前記第 2 の表示は、前記制御オペレータ表示、前記保守表示、前記ビジネス表示、前記シミュレーション表示、及び前記エンジニアリング表示のうちの、前記一つの表示以外の何れか一つの表示である、  
請求項 2 0 に記載の一組の表示オブジェクト。

【請求項 2 2】

前記制御オペレータ表示は、前記プロセスプラントの運転中に一つ以上の構成要素と関連するプロセス値を示す、請求項 2 1 に記載の一組の表示オブジェクト。

40

【請求項 2 3】

前記保守表示は、前記プロセスプラント内の物理的構成要素の健全性またはステータスの表示を示す、請求項 2 1 又は請求項 2 2 に記載の一組の表示オブジェクト。

【請求項 2 4】

請求項 1 4 に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記更なる表示オブジェクトは、前記プロセスプラント内で実行される制御ルーチンを

50

示す表示にユーザがアクセスすることを可能にする、一組の表示オブジェクト。

【請求項 25】

前記表示オブジェクトには、ユーザにより定義可能な、前記表示オブジェクトの実行中における前記視覚的特徴の動作が含まれる、請求項 14 に記載の表示オブジェクト。

【請求項 26】

前記動作は、アニメーションによって定義される、請求項 25 に記載の表示オブジェクト。

【請求項 27】

プロセスプラントで使用するためのプロセス制御システムであって、  
 プロセスを実行するために共に動作する複数の構成要素と、  
 前記複数の構成要素に通信により接続されたプロセスコントローラと、  
 前記プロセスコントローラに実装され、前記複数の構成要素の動作を制御する一つ以上の制御ルーチンと、  
 プロセッサおよび表示エリアを含む表示デバイスと、  
 請求項 14 ~ 請求項 16、請求項 25、及び請求項 26 の何れか 1 項に記載の表示オブジェクトと、  
 を含むプロセス制御システム。

10

【請求項 28】

プロセスプラントで使用するためのプロセス制御システムであって、  
 プロセスを実行するために共に動作する複数の構成要素と、  
 前記複数の構成要素に通信により接続されたプロセスコントローラと、  
 前記プロセスコントローラに実装され、前記複数の構成要素の動作を制御する一つ以上の制御ルーチンと、  
 プロセッサおよび表示エリアを含む表示デバイスと、  
 請求項 17 ~ 請求項 24 の何れか 1 項に記載の一組の表示オブジェクトと、  
 を含むプロセス制御システム。

20

【請求項 29】

プロセスプラントで使用するためのプロセス制御システムであって、  
 プロセスを実行するために共に動作する複数の構成要素と、  
 前記複数の構成要素に通信により接続されたプロセスコントローラと、  
 前記プロセスコントローラに実装され、前記複数の構成要素の動作を制御する一つ以上の制御ルーチンと、  
 プロセッサおよび表示エリアを含む表示デバイスと、  
 請求項 1 ~ 請求項 13 の何れか 1 項に記載の表示エディタプログラムと、  
 を含むプロセス制御システム。

30

【発明の詳細な説明】

【技術分野】

【0001】

関連出願

本出願は、優先権の目的のために、2004年5月4日に提出された、「プロセス制御システムを表わし、モニターし、それと対話するためのグラフィカルユーザインタフェース」と題する米国仮出願番号第60/567,980号に係る特典の正式に提出された出願であり、前記特典を請求するものであり、これを全体として参照により明示的に援用する。本出願は、「プロセスプラントにおけるグラフィック表示要素、プロセスモジュールおよび制御モジュールの統合」と題して、2003年7月21日に提出され、2004年8月5日に米国特許出願第2004/0153804号公報として公告された米国特許出願第10/625,481号にも関連しており、これは「プロセスプラントにおけるスマートプロセスモジュールおよびオブジェクト」と題して、2002年10月22日に提出され、2004年4月22日に米国特許出願第2004/0075689号公報として公告された、米国特許出願第10/278,469号の一部継続出願であり、これらの出願の全体の開示を、ここに明示的に参照により援用する。本出願は、「プロセスプラン

40

50

ト構成システムにおけるモジュールクラスオブジェクト」と題して、2003年2月18日に提出され、2004年10月7日に米国特許出願2004/0199925号公報として公告された、米国特許出願第10/368,151号にも関連しており、この開示の全体を、ここで参照により明示的に援用する。本出願は、本出願と同じ日付で、国際（PCT）出願として提出された下記の特許出願にも関連しており、ここで参照によりそれら全体を本出願に援用する。すなわち、「プロセス制御システム用のユーザにより構成可能なアラームおよびアラームトレンドング」（弁理士整理番号06005/41112）、「プロセスプラントにおけるプロセスモジュールおよびエキスパートシステムの統合」（弁理士整理番号06005/41113）、「統合環境における、カスタマイズされたプロセスグラフィック表示層を有するプロセスプラントユーザインタフェースシステム」（06005/41114）、「プロセス環境におけるスクリプトされたグラフィックス」（弁理士整理番号06005/41115）、「プロセス構成および制御環境へのグラフィックス統合」（弁理士整理番号06005/41116）、「プロセス環境における複数のビジュアライゼーションを有するグラフィック要素」（弁理士整理番号06005/41117）、「プロセスプラントにおいてグラフィック表示要素およびプロセスモジュールを構成するためのシステム（弁理士整理番号06005/41118）、「一元化されたプロセス制御システムインタフェース用グラフィック表示構成フレームワーク」（弁理士整理番号06005/41124）、「プロセスプラントユーザインタフェースにおけるマークアップ言語に基づく動的プロセスグラフィックス」（弁理士整理番号06005/41127）、「プロセス制御データを変更するための方法および装置」（弁理士整理番号06005/591622及び20040/59-11622）、「プロセス制御データにアクセスするための方法および装置」（弁理士整理番号06005/591623及び20040/59-11623）、「プロセス制御システム用の統合グラフィカル実行時インタフェース」（弁理士整理番号06005/591628及び20040/59-11628）、「プロセス制御システム用のサービス指向アーキテクチャ」（弁理士整理番号06005/591629及び20040/59-11629）。

10

20

#### 【0002】

本発明は、全体としてプロセスプラントに関し、より詳しくは、プラント構成、制御、保守、およびシミュレーションと関連する各種の活動における、共通のグラフィック表示要素の生成および使用を可能にするための、プロセス制御およびシミュレーションシステムのシステムレベルにおける、グラフィック表示エディタおよびグラフィック表示オブジェクトの統合および利用に関する。

30

#### 【背景技術】

#### 【0003】

化学、石油または他のプロセスで使用されるような分散プロセス制御システムは、一般に、アナログ、デジタル、または、アナログとデジタルの組合せのバスを介して、一つ以上の現場装置（フィールドデバイス）に通信により結合される一つ以上のプロセスコントローラを含んでいる。現場装置（これは、例えば、バルブ、バルブポジショナー、スイッチおよびトランスミッタ（例えば、温度、圧力、レベルおよび流量センサー）であってよい）は、プロセス環境内に設置され、例えば、バルブの開閉、プロセスパラメータの測定等のプロセス機能を行なう。スマート現場装置（例えば、FOUNDATION（登録商標）Fieldbusプロトコルのような周知のFieldbusプロトコルに準拠した現場装置）は、一般にコントローラ内で実行される制御計算、警報機能、および他の制御機能も行なうことができる。プロセスコントローラ（これも一般に、プラント環境内に設置される）は、現場装置により測定されたプロセス測定値および/または現場装置に関する他の情報を表わす信号を受け取り、例えば、プロセス制御決定を行ない、受け取った情報に基づいて制御信号を生成し、HART（登録商標）およびFieldbus現場装置等の現場装置で実行されている制御モジュールまたはブロックと調和する異なる制御モジュールを実行させるコントローラアプリケーションを実行する。コントローラにおける制御モジュールは、通信ライン上で現場装置に制御信号を送り、それによりプロセスの動作を制御する。

40

#### 【0004】

現場装置およびコントローラからの情報は通常、データハイウェイを介して、一般に制

50

御室または他のより過酷なプラント環境から離れた場所に配置される、例えば、オペレータワークステーション、パソコン、データヒストリアン、レポートジェネレータ、セントライズドデータベース等の、一つ以上の他のハードウェアデバイスで利用できるようにされる。これらハードウェアデバイスは、例えば、オペレータがプロセスに対して、例えば、プロセス制御ルーチンの設定を変更すること、コントローラまたは現場装置内で制御モジュールの動作を変更すること、プロセスの現在の状態を確認すること、現場装置およびコントローラによって生成されるアラームを確認すること、要員をトレーニングするまたはプロセス制御ソフトウェアを試験する目的でプロセスの動作をシミュレートすること、構成データベースを維持し更新すること等の機能を行なうことを可能にするアプリケーションを実行する。

10

**【0005】**

例として、DeltaV（登録商標）制御システム（Emerson Process Management社により発売）は、プロセスプラント内のさまざまな場所に設置された異なるデバイス内に記憶され、それらによって実行される複数のアプリケーションを含んでいる。一つ以上のオペレータワークステーションに存在する構成アプリケーションは、ユーザがプロセス制御モジュールを生成あるいは変更し、これらプロセス制御モジュールを分散された専用のコントローラに、データハイウェイを介してダウンロードすることを可能にする。一般にこれら制御モジュールは、通信により相互接続された機能ブロックからなり、これらは制御スキーム内でそれらへの入力に基づいて機能を行ない、制御スキーム内の他の機能ブロックに出力を与える、オブジェクト指向プログラミングプロトコルにおけるオブジェクトである。構成アプリケーションはまた、ビューイング（表示）アプリケーションによって使用されて、オペレータにデータを表示し、オペレータがプロセス制御ルーチン内での設定（例えば設定点）を変えることを可能にするオペレータインタフェースを、設計者が生成または変更することを可能にしてもよい。各専用コントローラおよび（場合によっては）現場装置は、実際のプロセス制御機能を実行するために、割り当てられダウンロードされた制御モジュールを実行させるコントローラアプリケーションを記憶し、実行する。ビューイングアプリケーション（一つ以上のオペレータワークステーション上で実行されることができ）は、コントローラアプリケーションからデータハイウェイを介してデータを受け取り、このデータをプロセス制御システム設計者、オペレータ、またはユーザに、ユーザインタフェースを用いて表示し、例えば、オペレータ用のビュー、エンジニア用のビュー、技術者用のビュー等の、任意の多数の異なるビューを提供してよい。データヒストリアンアプリケーションは一般にデータヒストリアンデバイスに記憶され、これにより実行され、データハイウェイ上で与えられたデータのうちの全部または全てを収集し記憶する一方、構成データベースアプリケーションはデータハイウェイに付加されたまた別のコンピュータで実行され、現在のプロセス制御ルーチン構成およびそれと関連したデータを記憶することができる。或いは、構成データベースは構成アプリケーションと同じワークステーション内に設置されてもよい。

20

30

**【発明の開示】****【発明が解決しようとする課題】****【0006】**

プロセス制御環境で使用される制御およびサポートアプリケーションの数およびタイプが増大するにつれて、ユーザがこれらアプリケーションを効果的に構成し使用できるようにするため、さまざまなグラフィカル表示アプリケーションが提供されてきた。例えば、構成エンジニアが、プロセスプラント内の制御デバイスにダウンロードすべき制御プログラムをグラフィカル（図式的）に生成できるようにするため、グラフィカル表示アプリケーションを使用して、制御構成アプリケーションをサポートすることが行なわれてきた。加えて、制御オペレータがプロセスプラントの現在の機能状態またはプロセスプラントのエリアを確認することを可能にするため、保守要員がプロセスプラント内のハードウェアデバイスの状態を確認することを可能にするため、プロセスプラントのシミュレーションを可能にするため等に、グラフィカル表示アプリケーションが使用されてきた。しかしなが

40

50

ら、これらグラフィカル表示アプリケーションは、過去において、それらが関連している特定のアプリケーションの一部として、またはそれをサポートするために生成され、したがって一般にそれらの有用性は、それらが生成された目的である特定のプロセス機能に限定される。例えば、制御または他のオペレータをサポートするよう生成されたグラフィカルプログラムを、保守、構成またはシミュレーション機能で用いることは、不可能とは言わないまでも困難である。

#### 【0007】

具体的な例として、いくつかのプロセス制御構成アプリケーションは、現在、プロセスプラントに対する制御計画（戦略）を生成するのに使用されるテンプレートオブジェクト、例えば、機能ブロックテンプレートオブジェクトおよび場合によっては制御モジュールテンプレートオブジェクトのライブラリを含んでいる。テンプレートオブジェクトは、デフォルトプロパティ、設定およびそれらと関連したメソッドを有しており、グラフィカル構成アプリケーションを用いるエンジニアは、これらテンプレートオブジェクトを選択し、選択されたテンプレートオブジェクトのコピーを実質的に構成スクリーンに配置して、制御モジュールを開発することができる。テンプレートオブジェクトを選択し、構成スクリーンに配置する過程の中で、エンジニアはこれらオブジェクトの入力および出力を相互接続し、それらのパラメータ、名称、タグおよび他のプロパティを変更して、プロセスプラントにおける特定の使用に対して特定の制御モジュールを生成する。一つ以上のこのような制御モジュールを生成後、エンジニアは次いで、制御モジュールをインスタンス化（実体化）し、それを適切な一つ以上のコントローラまたは現場装置にダウンロードして、プロセスプラントの運転中に実行することができる。

10

20

#### 【0008】

その後、エンジニアは、さまざまなグラフィカル表示生成アプリケーションを用いて、表示生成アプリケーションで表示オブジェクトを選択し、構築することにより、プロセスプラント内のオペレータ、保守要員等に対して、一つ以上の表示を生成することができる。これらの表示は一般に、ワークステーションの一つ以上においてシステムワイドに（システム全体で）実行され、プラント内の制御システムまたはデバイスの動作状態について予め構成された表示をオペレータまたは保守要員に与える。これら表示は、一般に、コントローラまたはプロセスプラント内のデバイスによって生成されるアラームを受け取り表示する警報表示、プロセスプラント内のコントローラおよび他のデバイスの動作状態を示す制御表示、プロセスプラント内のデバイスの機能状態を示す保守表示等の形態を取る。しかしながら、これら表示は、一般に予め構成され、プロセスプラント内のプロセス制御モジュールまたはデバイスから受け取った情報またはデータを既知の方法で表示する。あるシステムではこのような表示は、物理的または論理的要素を表わし、物理的または論理的要素に通信により結合されて物理的または論理的要素についてのデータを受け取るグラフィック描写によって生成される。表示スクリーン上のグラフィックは、あるイベント、例えば、受信データ（例えば、タンクの充てん量が半分であることを示すデータ、流量センサーによって測定された流量を示すデータ等）に応じて変わってよい。しかしながら、構成、オペレータ制御、保守およびシミュレーション活動に使用されるグラフィカル表示は、一般に異なるグラフィカルエディタを用いて、互いに別々に生成される。更にまた、これら表示の限定されたグラフィカル機能は実行するのが難しく、いかなるグラフィカルオブジェクトの一部としても実行されない。

30

40

#### 【0009】

したがって、制御構成アプリケーションと同様に、表示生成アプリケーションは、スクリーン上に任意の所望の構成で配置してオペレータ表示、保守表示等を生成できるテンプレートグラフィカル表示アイテム（例えば、タンク、バルブ、センサー、スライドバーのようなオペレータ制御ボタン、オン/オフスイッチ等）を有してよい。個々のグラフィックアイテムは、スクリーン上に配置されるとき、ユーザに対して、プロセスプラントの内部動作についての何らかの情報またはその表示を与える方法で、スクリーン上で相互接続されることができる。しかしながら、グラフィック表示をアニメート（アニメーショ

50



ン化)するためには、表示作成者は、グラフィカルアイテムのそれぞれを、プロセスプラント内で生成されるデータ(例えば、センサーによって測定されるデータあるいはバルブの位置を示すデータ等)に、グラフィックアイテムとプロセスプラント内の当該データソースとの間の通信リンクを指定することによって、手動で結合しなければならない。この処理は単調で時間が掛かり、かつエラーを孕む恐れがある。

#### 【0010】

制御構成アプリケーション内の制御テンプレートオブジェクトおよび表示生成アプリケーション内の表示アイテムは、それらをコピー及び使用して、多くの異なる制御モジュールおよびグラフィカル表示が生成できるので便利であるが、プロセスプラント内の異なる機器に対して、多数の同じ制御モジュールおよびグラフィカル表示を生成する必要が起ることが多い。例えば、多くの中型から大型プロセスプラントは、同じ基本的な汎用制御モジュールおよび表示を用いて制御し確認することができる、同じまたは同様の機器の実体を多数有している。しかしながら、これら多数の制御モジュールおよび表示を生成するためには、汎用制御モジュールまたは表示モジュールを生成し、次いで、適用可能なさまざまな機器のそれぞれに対して、この汎用制御または表示モジュールをコピーする。もちろん、コピーした後で、各新しい制御または表示モジュールのそれぞれを構成アプリケーション内で手動で変更して、それが付加される特定の機器を指定しなければならず、次いでこれらの制御および表示モジュールの全てをインスタンス化し、プロセス制御システムにダウンロードしなければならない。

10

#### 【0011】

残念なことに、上述の制御モジュールおよび表示アイテムは、いかなる点でもモジュール式ではない。したがって、コピーした後で、制御モジュールおよび表示のそれぞれは、適切な構成アプリケーションを用いて手動でかつ個別に変更されることにより、それらが関連すべきプラント内の機器を指定しなければならない。同じタイプの機器のコピー(すなわち、複製(レプリケート)された機器)を多数有しているプラントでは、この処理は単調で時間が掛かり、オペレータがもたらすエラーを孕む恐れがある。更にまた、ひとたびプログラムされると、これらの異なる制御モジュールおよび表示は互いに他を感知しない。したがって、ひとたび生成された制御モジュールに変更を施すには、エンジニアまたはオペレータは、異なる複製された機器に対する異なる制御モジュールのそれぞれに対して同じ変更を手動で行なわなければならない、これまた時間が掛かり単調である。同じ問題は、プラント内の複製された機器の異なるセットに対して生成されるグラフィカルビューにも当てはまる。言い換えれば、ひとたび特定の制御モジュールまたは特定のグラフィカルビューが(個別にまたはテンプレートオブジェクトからのコピーによって)生成され、次いでプラント内の機器の特定のセットに結合されると、この制御モジュールまたはグラフィカルビューは、それと同じまたは同様の他の制御モジュールまたはグラフィカル表示に対していかなる自動的な感知もせず、システム内で別々のエンティティまたはオブジェクトとして存在する。結果として、特定のタイプの制御モジュールおよびグラフィカル表示のうちのいずれにも適用可能な変更を、それらのモジュールおよび表示に対して個別に施さなければならない。この問題は、グラフィカルビューが、同じ機器に対して生成されるがプラント内の異なる機能的コンテキスト(状況)で生成される場合、例えば、制御ビューイング、保守ビューイングおよびシミュレーション機能に対して生成される場合、より顕著になる。この場合、グラフィカルビューは、互いに他に対するいかなる認識または感知もなしに別々に生成される。

20

30

40

#### 【0012】

したがって、グラフィック表示は、プロセスプラント内で行なわれる一般的な異なる活動に対して使用される異なるアプリケーションで提供され、それらと関連付けられてきたが、これらグラフィック表示および関連グラフィック表示エディタは、一般にそれらがサポートするよう生成されたアプリケーションの機能レベルで追加されてきた。結果として、グラフィカルエディタはそれらが存在する範囲で、ユーザが、特定のアプリケーションによって必要とされる特定の機能性をサポートするグラフィックスを生成することを可能

50

にただけであった。以前のプロセスプラントは、プラント構成およびサポートのコンテキスト（状況）で行なわれる各種の活動または複数の活動のグラフィカルニーズによって使用され得る、またはそれをサポートし得るグラフィカル表示エディタを提供してこなかった。したがって、例えば制御構成活動をサポートし、あるいは可能にするのに使用されるグラフィカル表示エディタは、ユーザが制御プログラムを生成することを可能にするだけであり、オペレータまたは保守表示のニーズまたは機能をサポートしなかった。同様に、プラントの運転中に制御オペレータまたは保守技術者に提供されるオペレータビュー、保守ビュー等を生成するのに使用されるグラフィック表示エディタは、構成活動、シミュレーション活動等と関連した機能をサポートしなかった。グラフィック表示のニーズが、プロセスプラントの個々の機能レベルで（例えば、制御構成、保守サポート、制御オペレータサポートおよびシミュレーションサポート機能レベルで）サポートされる結果として、これら各種のエディタによって生成される様々な表示は、プラント内の同じ構成要素のモデリングおよび表現に終わり、結果としてプロセスプラントでの各種の様々な要員によるグラフィカル表示の労力が重複する。この労力の重複は、異なる用途に対して同じプロセス要素を表現する異なるグラフィカル表示を生成するのに必要な労力ばかりでなく、異なる表示アプリケーションで使用されるグラフィック要素を、それらが関連しているプロセスプラント内の実際のハードウェアまたはソフトウェア要素に結合するのに必要な労力に現れる。

10

#### 【0013】

各種のプロセスプラント活動のためのグラフィカルサポートは、実際に活動が行なわれているという事実の後に、かつ行なわれている実際の活動の一部として提供されてきたので、グラフィカルサポートは、プラント環境において共通のグラフィックスがプラント内で各種の異なる機能レベルで生成され、使用されることを可能にする方法で統合されていない。このグラフィックスの非統合は、異なる機能に対して実際に生成されるグラフィックスが、機能間で、またはアプリケーション間で異なることになり、これは一つの特定のタイプのグラフィカル表示には慣れているがプラント内の異なる動作または機能と関連した異なる表示を時々見なければならぬ、一部のユーザを混乱させ得るものがある。同様に、上記のように、プラントの各種の異なる機能レベルでグラフィカル表示サポートを与えることは、表示を生成すること、及び表示内の要素をプラント内の実際のハードウェアまたはソフトウェア要素に適切に接続する際の両方において、グラフィックサポートの重複となる。

20

30

#### 【0014】

加えて、個々のデバイス内の異なるコントローラおよび問題に対して実行される制御ループと関連する状態、エラー、アラーム等を検出するのに、エラー検出およびその他のプログラミングが有用である。このようなエラー検出は、従来、プロセスプラントの異なる機能レベルで行なわれ、それらの異なる機能性活動に対して生成されたグラフィック表示上で表示されてきた。したがって、プロセス制御システムをプログラミングして、プロセスプラント内で、異なる、多種多様に設置されている可能性のあるデバイスからのデータを分析することによって検出しなければならないシステムのレベル状態またはエラーを認識することは困難であったし、またこれらのタイプのエラーを、このようなシステムのレベル状態の情報を、オペレータまたは保守要員に示す又は呈示するようには生成されなかったオペレータ表示上に示すことは、更により困難であった。また、表示内の異なる要素に対するこれらの代替的な情報またはデータのソースにより、オペレータ表示内のオブジェクトをアニメートすることは困難である。

40

#### 【課題を解決するための手段】

#### 【0015】

実行時環境で、グラフィカル表示が互いに他と関連付けられる方法で、それらを生成することを可能にするための、グラフィカル表示サポートが、プロセスプラント構成、モニタリングおよびシミュレーションシステム内に提供される。具体的には、単一のグラフィック表示エディタが、例えば実行時環境で互いに他からアクセスされることにより、グラ

50

フィック表示のうちの一つの中のプロセスエンティティについて更なる情報を与え、プロセスプラントの隣接する部分を通してスクロールし、または、プロセスプラント内の異なる機能、例えば、オペレータビューイング機能、シミュレーション機能および保守機能に対して、異なる表示を与えることができる、各種の相互関連グラフィック表示を生成するのに使用されることができる。グラフィック表示を生成するのに、同じグラフィックエディタが使用されているため、結果として生ずるグラフィック表示は、同じ様相および印象を有することができ、プラント内の実行時環境に全般的に同じ方法でバインド（結合）されてよく、これによりプロセスプラントで使用されるプロセスグラフィック表示を構成及び生成するのに必要とされる時間を低減する。

【図面の簡単な説明】

【0016】

【図1】プロセスプラント内の各種の機能と関連した表示ルーチンおよび他のアプリケーションを実行するオペレータワークステーション、並びに、プラントの各種の機能エリアのそれぞれに対して、グラフィック表示要素およびグラフィック表示を生成するのに使用できるシステムレベルグラフィカルサポートを提供するワークステーションを含む、プロセスプラント内に設置された分散プロセス制御ネットワークのブロック図である。

【図2】プロセスプラント制御、ビューイングおよびシミュレーションシステムにおけるシステムレベルグラフィカルサポートの統合を示す論理的ブロック図である。

【図3】グラフィック要素および表示が生成される構成環境、および、グラフィック要素および表示を実行できる実行時環境を示す論理図である。

【図4】グラフィカルエディタによって作成され、ユーザが、ポンプ要素の形態のグラフィック要素を生成または編集することを可能にする、簡略化された表示スクリーン（画面）である。

【図5】反応器の形態のグラフィック要素を生成する、グラフィカルエディタによって作成された更なる表示スクリーンである。

【図6】各種のビジュアライゼーションをグラフィック要素に結合またはそれと関連付けることができる一つの方法を示すブロック図である。

【図7】グラフィック要素と関連する第1のプロパティビューを示すスクリーン表示の一部を示す。

【図8】グラフィック要素と関連する第2のプロパティビューを示すスクリーン表示の一部を示す。

【図9】グラフィック要素のプロパティと関連する動作を示す表示スクリーンの動作/アニメーション部を示す。

【図10】ビジュアルトリガを、グラフィック要素のプロパティおよびビジュアライゼーションと統合することができる方法を示すブロック図である。

【図11】グラフィック要素のビジュアライゼーションに対して、変換アニメーションを提供、あるいは、定義するのに使用できるダイアログボックスを示す。

【図12】グラフィック要素のビジュアライゼーションに対して、色アニメーションを含むプロパティアニメーションを提供、あるいは、定義するのに使用できるダイアログボックスを示す。

【図13】解決テーブルを介して、グラフィック要素を実行時環境に結合することができる方法を全体として示す図である。

【図14】プロセス環境の複数の異なるデータソースにバインド（結合）されている表示要素を示すブロック図である。

【図15】複数の表示要素およびコネクタからグラフィック表示を生成する、グラフィカルエディタによって作成された表示スクリーンである。

【図16】グラフィック表示内に配置されるグラフィック要素と関連する各種のビジュアライゼーションを図示するグラフィック表示を生成するための表示スクリーンである。

【図17】各種の相互接続されたグラフィック要素およびコネクタからなるグラフィック表示と関連する表示スクリーンである。

10

20

30

40

50

【図18】図17の表示内の要素に対して、制御パネルおよびフェースプレートを示すために、図17のグラフィック表示を介してアクセスされることができると一組のグラフィック表示スクリーンである。

【図19】各種の表示要素からなる石灰キルンを示し、プロセスプラント内の各種の他のアプリケーションおよびデータソースからデータが提供されることを示す、グラフィック表示の表示スクリーンである。

【図20A】同じ様相および印象を有する石灰キルンの、オペレータビュー、エンジニアリングビュー、ビジネスビュー、シミュレーションビューおよび保守ビューを含む異なるビューを示すグラフィック表示スクリーンである。

【図20B】同じ様相および印象を有する石灰キルンの、オペレータビュー、エンジニアリングビュー、ビジネスビュー、シミュレーションビューおよび保守ビューを含む異なるビューを示すグラフィック表示スクリーンである。

【図20C】同じ様相および印象を有する石灰キルンの、オペレータビュー、エンジニアリングビュー、ビジネスビュー、シミュレーションビューおよび保守ビューを含む異なるビューを示すグラフィック表示スクリーンである。

【図20D】同じ様相および印象を有する石灰キルンの、オペレータビュー、エンジニアリングビュー、ビジネスビュー、シミュレーションビューおよび保守ビューを含む異なるビューを示すグラフィック表示スクリーンである。

【図20E】同じ様相および印象を有する石灰キルンの、オペレータビュー、エンジニアリングビュー、ビジネスビュー、シミュレーションビューおよび保守ビューを含む異なるビューを示すグラフィック表示スクリーンである。

【図21A】図20A~20Eに示されている石灰キルンを運転するのに使用される制御ルーチンに関連する表示スクリーンである。

【図21B】図20A~20Eに示されている石灰キルンを運転するのに使用される制御ルーチンに関連する表示スクリーンである。

【図22】物理的レベルおよび機能レベルを含む各種のレベルにおける各種の異なるグラフィック表示間の関連を示すブロック図である。

【図23】構成システムと関連するグラフィカル構成階層を示す第1の構成スクリーンを示す。

【図24】グラフィック要素およびグラフィック表示を、構成システム内の他の要素に対して指定し、それと統合することができる一つの方法を示すグラフィカル構成階層を示す第2の構成スクリーンを示す。

【図25】一つ以上のグラフィック表示を、プロセスプラント構成システム内で指定する方法を簡単に示すためにユーザに提供することができるダイアログボックスを示す。

【発明を実施するための形態】

【0017】

図1は、システムレベルのグラフィカルサポートが、プラント10の各種の機能エリアに設けられているプロセスプラント10の例を示す。典型的に、プロセスプラント10は、入出力(I/O)デバイスまたはカード18(これは例えば、Fieldbusインタフェース、Profibus(登録商標)インタフェース、HART(登録商標)インタフェース、標準の4-20maインタフェース等であってよい)を介して一つ以上の現場装置14および16にそれぞれ接続された、一つ以上のコントローラ12を有する分散プロセス制御システムを含んでいる。コントローラ12は、例えばEthernet(登録商標)リンクであってよいデータハイウェイ24を介して、一つ以上のホストまたはオペレータワークステーション20~23にも結合されている。データベース28は、データハイウェイ24に接続されていてよく、プラント10内のコントローラおよび現場装置と関連するパラメータ、ステータスおよび他のデータを収集し、記憶するデータヒストリアンとして、および/または、コントローラ12および現場装置14および16にダウンロードされてそれらの内に記憶されているプラント10内のプロセス制御システムの現在の構成を記憶する構成データベースとして動作する。データベース28は、ここで述べる方法で生成されるグラフィカルオブジェクトを追加的に記憶して、プロセスプラント

10

20

30

40

50

10内でグラフィカルサポートを提供することもできる。コントローラ12、I/Oカード18および現場装置14および16は、一般に、時に過酷なプラント環境内に設置され、分散されるが、オペレータワークステーション20~23およびデータベース28は、一般にコントローラまたは保守要員が容易にアクセス可能な制御室、またはその他のより過酷でない環境内に設置される。しかしながら、場合によっては、これらの機能を実行するのにハンドヘルド（携帯）デバイスを使用してもよく、これらハンドヘルドデバイスは一般にプラント内の様々な場所に運ばれる。

#### 【0018】

周知のように、コントローラ12（例えば、Emerson Process Management社から発売されているDeltaV（登録商標）コントローラであってよい）のそれぞれは、任意の数の、異なる、独立して実行される、制御モジュールまたはブロック29を用いて制御計画を実行するコントローラアプリケーションを記憶し、実行する。制御モジュール29のそれぞれは一般に機能ブロックと呼ばれるもので構成することができ、これは各機能ブロックが全体的な制御ルーチンの一部またはサブルーチンであり、（リンクと呼ばれる通信を介して）他の機能ブロックと連動して動作して、プロセスプラント10内のプロセス制御ループを実行する。周知のように、オブジェクト指向プログラミングプロトコルでのオブジェクトであってよい機能ブロックは、一般に例えばトランスミッタ、センサーまたは他のプロセスパラメータ測定デバイスと関連する入力機能、例えばPID、ファジイ論理等の制御を行なう制御ルーチンと関連する制御機能制御機能、または、バルブ等のある種のデバイスの動作を制御する出力機能のうちの一つを実行して、プロセスプラント10内でいくつかの物理的機能を行なう。もちろん、例えば、モデル予測コントローラ（MPC）、オブティマイザー等の、ハイブリッドおよび他の種類の複雑な機能ブロックも存在する。FieldbusプロトコルおよびDeltaV（登録商標）システムプロトコルは、オブジェクト指向プログラミングプロトコルで設計され、実行される制御モジュールおよび機能ブロックを使用するが、制御モジュールは、例えば、順次機能ブロック、ラダーロジック等を含む任意の所望の制御プログラミング方式を用いて設計することができ、また、機能ブロックまたは何か他の特定のプログラミング技法を用いて設計及び実行されることには限定されない。

#### 【0019】

図1に示すプラント10では、コントローラ12に接続された現場装置14および16は、標準の4-20 maデバイスであってよく、プロセッサおよびメモリを含むスマート現場装置（例えば、HART（登録商標）、Profibus（登録商標）、またはFOUNDATION（登録商標）Fieldbus現場装置）であってよく、または任意の他の所望のタイプのデバイスであってよい。例えばFieldbus現場装置のようなこれらデバイス（図1において参照番号16で示す）のうちいくつかは、例えばコントローラ12で実行される制御計画と関連する機能ブロックのようなモジュールまたはサブモジュールを記憶し、実行できる。Fieldbus現場装置16のうち2つの異なるものの内に配置されているとして図1に示す機能ブロック30は、周知のように、コントローラ12内で制御モジュール29の実行と関連して実行されて、プロセス制御を実行してもよい。もちろん、現場装置14および16は、例えば、センサー、バルブ、トランスミッタ、ポジショナー等の任意のタイプのデバイスであってよく、また、I/Oデバイス18は、例えば、HART（登録商標）、Fieldbus、Profibus（登録商標）等の任意の所望の通信またはコントローラプロトコルに準拠した任意のタイプのI/Oデバイスであってよい。

#### 【0020】

図1のプロセスプラント10では、ワークステーション20~23は、プラント10内の同じまたは異なる要員によって行なわれる各種の異なる機能に対して使用される各種のアプリケーションを含むことができる。ワークステーション20~23のそれぞれは、各種のアプリケーション、プログラム、データ構造等を記憶しているメモリ31、およびメモリ31に記憶されているアプリケーションのうち任意のものを実行するのに使用できるプロセッサ32を含んでいる。図1に示す例では、ワークステーション20は、構成ワークステーションに指定され、権限を与えられた構成エンジニアがアクセスして、制御ルーチンまたはモジュール

ル（例えば、モジュール29および30）を生成し、プラント10の各種のコントローラ12およびデバイス16にダウンロードすることができる、例えば、制御モジュール生成アプリケーション、オペレータインタフェースアプリケーションおよび他のデータ構造を含むことができる、一つ以上の構成アプリケーション33を含んでいる。ワークステーション21は、全体として制御オペレータビューイング（閲覧用）ワークステーションとして図1に示されており、プロセスプラント10の運転中に制御オペレータに各種の表示を与えて、オペレータがプロセスプラント10内で、またはプラントのさまざまな部分で起こることを確認するとともに、制御できるようにすることができる、多数の表示アプリケーション34を含んでいる。アプリケーション34は、例えば、制御診断アプリケーション、チューニングアプリケーション、レポート生成アプリケーション、または制御オペレータが制御機能を行なうのを支援するのに使用できる任意の他の制御サポートアプリケーションのような、サポートアプリケーション34aを含んでいてよい。同様に、ワークステーション22は、保守ビューイングワークステーションとして図示されており、各種の保守要員が、プラント10の保守ニーズを確認し、各種のデバイス12、14、16等の動作または稼動状況を確認するのに使用できる、多数の保守アプリケーション35を含んでいる。もちろん、アプリケーション35は、例えば、保守診断アプリケーション、較正アプリケーション、振動分析アプリケーション、レポート生成アプリケーション、または保守要員がプラント10内で保守機能を行なうのを支援するのに使用できる任意の他の保守サポートアプリケーションのような、サポートアプリケーション35aを含んでいてよい。加えてワークステーション23はシミュレーションワークステーションとして示されており、トレーニング目的、プラントの保守および制御を支援するプラントモデリング目的等を含む任意の多数の目的のために、プラント10またはプラント10の各種の部分の運転をシミュレートするのに使用できる多数のシミュレーションアプリケーション36を含んでいる。典型的には、ワークステーション20～23のそれぞれは、表示スクリーン37を、キーボード、マウス等のような他の標準周辺装置と共に含んでいる。

10

20

#### 【0021】

各種の構成、制御、保守およびシミュレーションアプリケーション33～36は、それらの機能の一つに特化された異なるワークステーションに設置されていることとして図1に示されているが、これらの、または他のプラント機能と関連する各種のアプリケーション33～36は、プラント10のニーズおよび設定により、プラント10内の同じまたは異なるワークステーションまたはコンピュータに設置され、実行されてよいことが当然ながら理解されよう。したがって、例えば、一つ以上のシミュレーションアプリケーション36および制御アプリケーション33は、同じワークステーション20～23で実行されてよく、一方、異なる個々のシミュレーションアプリケーション36または異なる個々の制御アプリケーション33は、ワークステーション20～23のうちの異なるもので実行されてよい。

30

#### 【0022】

過去において、プラント10の異なる機能エリアで使用される異なるアプリケーションの開発は、全く独立的に行なわれた。したがって、構成アプリケーション33の開発は、シミュレーションアプリケーション36、保守アプリケーション35またはオペレータ制御アプリケーション34と統合されていなかった。事実上、多くの場合にプラントは、異なる会社またはソフトウェア供給者によって開発され、また事実上プラント10内の他のソフトウェアから独立して実行されるよう開発された、異なる機能エリアのためのアプリケーションを含んでいる場合があった。このようなプラント10の各種の機能エリアと関連した異なるアプリケーションの独立した開発および運転の結果として、プラント要員は一般に、構成、オペレータ制御、保守およびシミュレーション機能レベルのそれぞれにおいて、プラントを別々に構成またはセットアップする必要があった。具体的には、同じまたは異なるプラント要員は一般的に、異なるプログラムを用いて、機能レベルのそれぞれにおいて、新しいデータ構造およびグラフィック表示を設定しなければならなかった。したがって、図1では、各種のアプリケーション33～36のそれぞれについて、行われた構成、制御、保守およびシミュレーション機能は、一般に、プラント要員がこれらの構成、オペレータ制御、

40

50

保守およびシミュレーション機能を行なうのを支援するための、異なるグラフィック表示エディタおよびデータベース構造を含んでいたか、あるいは使用した。多くの場合、これらの異なるグラフィック表示エディタおよびデータベースは、異なるグラフィック表示を生成して、プラント10の同じ部分またはエリア、またはプラント10内の同じハードウェアを表現あるいはモデル化するために、また、異なるプラント要員が、構成、オペレータ制御、保守またはシミュレーション活動の状況において、プロセスプラントに何が起こりつつあるかを視覚化し理解するのを支援するのに使用された。

#### 【0023】

アプリケーション33~36およびプラント10内の各種の機能のそれぞれに対する関連する表示は、一般に互いに無関係に、時には異なる人々によって、また異なる会社によってすら開発され、実行されたため、プロセスプラントの異なる機能エリアにおいて生成されまたは使用されるグラフィック表示は、グラフィカルなビジュアライゼーション（視覚性）を提供する点からは、首尾一貫したまたは容易に理解される方法では全く統合されていなかった。結果として、プラントの各種の異なる機能レベルにおけるグラフィック表示の独立した生成および実行により、グラフィック表示は機能間で異なる様相を呈し、グラフィック表示には、機能エリアにわたって首尾一貫した様相および印象がなかった。さらに、このような独立した生成は、プラントの同じ部分またはエリアだが異なる機能的用途のためにグラフィック表示を生成する際に労力の重複となり、そのように生成されたグラフィック表示は、例えばコントローラ12および現場装置14、16等のプラント10内の各種のデバイスに、プラント10の機能レベルごとに別々に結合され、それらからデータを受け取ることが必要とされた。この事実はまた、異なる表示のために同じハードウェア要素を追跡するのに、データベース構造の重複を必要とした。したがってこれまでは、例えば第1のアプリケーション（例えば、アプリケーション35のうちの一つ）は、保守目的のためにプラント10の一部を示す保守表示を生成するのに使用され、一方、第2のアプリケーション（例えば、アプリケーション34のうちの一つ）は、制御目的のために、プラント10の同じ部分を示す制御オペレータ表示を生成するのに使用された。これらの異なる表示エディタによって別々に生成された表示は、様相および印象において全く異なる場合があり、ユーザが保守表示とオペレータ表示との間で、混乱することなしに容易に、あるいは各タイプの表示についてトレーニングの必要なしに、行ったり来たりするよう切り換えるのは困難であった。同様に、異なるアプリケーション34および35において、両方の表示を独立して生成する際に労力が重複し、また保守表示および制御オペレータ表示を、プラント10内で同じハードウェア要素に別々に結合あるいは接続して、時には同じまたは同様のデータをこれらのハードウェア要素から受け取るためのデータベース構造を生成するには、さらなる追加の労力が必要であった。

#### 【0024】

これらの非能率を軽減し、プラント10内でより広く有用で理解しやすいグラフィックを提供するためには、プロセスプラント10のシステムレベルにグラフィカルサポート層が設けられて、構成、オペレータビューイング、保守ビューイング、シミュレーションおよびプラント10の他の機能エリアを含むプラント10の各種の機能エリアのそれぞれのグラフィック表示およびデータ構造ニーズをサポートする。このシステムレベルのサポートは、プラント動作レベル40、プラント機能性レベル42およびシステムレベル44を示す図2に図式的に表現されている。図2から分かるように、プラント動作レベル40は、コントローラ12、現場装置14、16等を含んでおり、これらは、制御ルーチンまたはモジュール29および30、並びにプラント10内で実行される他のソフトウェアを実行して、プラントの実行時中にプラント運転を実施する。プラント機能性レベル42は、構成機能ブロック46、制御機能ブロック47、保守機能ブロック48およびシミュレーションブロック49を含むこととして表現されているが、他のあるいは異なる機能（例えば、エンジニアリングおよびビジネス機能）も同様に設けられてよい。構成機能ブロック46は、プラント動作レベル40内の構成要素とインタフェース又は通信する構成ルーチン33を実行して、プラント動作レベル40に制御計画または制御モジュールを提供する。制御機能ブロック47は、制御ビューイングおよび

10

20

30

40

50

その他のアプリケーション34および34aを含んでおり、これらも、プラント動作レベル40内の各種の物理的および論理的構成要素と一般的には直接にインタフェース又は通信することにより、プラント10内で、オペレータにより起動された変更を実行する、制御表示34を介してオペレータに情報を提供する、制御アプリケーション34aのためのデータを獲得する等を行う。保守機能ブロック48は保守ルーチンおよびアプリケーション35および35aを含んでおり、これは、プラント動作レベル40内の各種の物理的および論理的構成要素とインタフェース又は通信することにより、保守手続きを実行する、保守データを収集する、保守表示35を介して保守要員に保守データまたは情報を提供する、診断アプリケーション35aを実行する等を行なう。同様に、シミュレーション機能ブロック49はシミュレーションルーチン36を含んでおり、これは、プラント10のシミュレーションを実行し、プラント動作レベル40内の構成要素に通信により結合してプラント10についてのデータを取得することができる。

10

#### 【0025】

図2に示すように、システムレベルサポート層44は、プラント機能性層42内で機能ブロック46～49のそれぞれに結合し、それらをサポートすることにより、例えば、各種の機能エリア46～49で使用するための共通のデータベースおよび表示構造（例えば、ソフトウェアオブジェクト、グラフィック要素およびグラフィック表示等）の生成および維持を可能にする。より詳しくは、システムレベルサポート層44は、機能ブロック46～49のそれぞれで行なわれるグラフィカル活動を共に統合する、あるいはシステムサポート層44において生成される共通のデータベース構造およびグラフィック要素を用いて開発することを可能にする、アプリケーション、データベースおよびグラフィカルサポート要素を含んでいる。このシステムレベルのサポートを提供するために、システムサポート層44は、グラフィカルエディタ50およびグラフィカルオブジェクトデータベース52を含んでいてよい。グラフィカルエディタ50は、グラフィック要素54およびグラフィック表示56を生成するのに使用でき、グラフィックオブジェクトデータベース52は、要素54および表示56をエディタ52およびブロック46～49における各種のアプリケーションがアクセス可能なメモリに記憶する。データベース52は、他のオブジェクト58、および、グラフィック要素54をプラント動作レベル40内の個々のハードウェアおよびソフトウェア要素に接続するデータ構造をも記憶していてよい。加えてデータベース52は、グラフィック要素又は表示テンプレート、あるいはさらなるグラフィック要素または表示を生成するのに使用できるプリミティブ（原型）を記憶していてよい。図2から分かるように、グラフィック表示要素54、表示56および他のデータベース構造58は、機能ブロック46～49のうちの任意の、および全てにおいて使用されることにより、それらの機能ブロックと関連したグラフィックスを生成し、使用することができる。

20

30

#### 【0026】

一般的に、システムレベルサポートブロック44は、図1のプロセスプラント10で使用されるグラフィックスを機能エリア46～49の全てにおいて統合する方法を提供し、これにより、異なる機能的コンテキストにおいて同じプラント機器に対して異なるグラフィック要素を繰り返し生成する必要性を低減する又は排除し、ユーザが機能エリア46～49のそれぞれにおいて、それらの機能エリアと関連したグラフィカルビューで表示されている機器と関連したデータを結合するのを容易にする。システムレベルサポート層44は、機能エリア46～49のそれぞれにおける複数のアプリケーション、また、異なる機能エリア46～49における異なるアプリケーション等に対して、グラフィックスおよびデータベースサポートを提供するのに使用できることが理解されよう。

40

#### 【0027】

再び図1を参照すると、システムレベルサポートブロック44は、他のワークステーション20～23のそれぞれに接続できる追加のワークステーションまたはユーザインタフェース60を用いて実装できる。ワークステーション60は、一般にグラフィックスエディタ50およびデータベース52を記憶することができ、所望であれば他の要素54、56および58を記憶してよい。加えて、ワークステーション60は、データバス24を介して、別々の有線または無

50



線通信接続（図1に点線で示す）を介して、または、任意の他の所望の方法で、通信によりワークステーション20～23に接続することができる。図1に示す構成では、ワークステーション60は、表示エディタ50を記憶及び実行して、ユーザが、グラフィック要素を生成すること、及びそのような要素を一つ以上のグラフィック表示にグループ化することを可能にし、ここではこれらの両方を表示モジュールと呼ぶ。これら表示モジュールは次いでデータベース52に記憶され、図2に示す各種の機能ブロック46～49によってアクセス及び使用され、各種のワークステーション20～23に実装されることができる。図示のため、システムレベルブロック44および機能レベルブロック46～49の機能は、図1における異なるまたは別々のワークステーション20～23および60で実装されることとして示してあるが、これら各種のブロックのうちの任意のものと関連したアプリケーションのうちの任意のものまたは全てが、同じまたは異なるワークステーション、または、プロセスプラント10内の又はこれと関連する他のコンピュータで実行され得ることが理解されよう。したがって、グラフィックスエディタ50は、他のワークステーション20～23のうちの任意のもの、または、プラント10と関連する任意の他のコンピュータに記憶され、その上で実行されてよく、スタンドアロンまたは別のコンピュータに記憶され、その上で実行される必要はない。

10

20

30

40

50

#### 【0028】

上述したように、図2のシステムレベル層44は、システムレベルの表示およびデータベースオブジェクトを実装し、これらは、各種の機能性環境で使用できると共に、より上位レベルの表示機能を提供することができる。一般的に、図2のシステムレベル44で生成される表示オブジェクトは、グラフィック要素およびグラフィック表示として分類することができる。グラフィック要素は、一般に、プラント内の特定の物理的エンティティ（例えば、バルブ、センサー、ポンプ、コントローラ等のようなハードウェアデバイス）と結びつけられたプリミティブまたは下位レベルの表示オブジェクトである。グラフィック表示は一般に、一組（セット）の相互接続されたグラフィック要素からなり、プラント内のより複雑なハードウェアのセット（例えば、ユニット、エリア等）を表現しモデル化するのに使用され、異なるハードウェアユニット間の相互接続を含んでいる。追加的に、グラフィック表示は、プラントや、他のアプリケーション（例えば、ワークステーション20～23および60内で実行される診断およびビジネスアプリケーション）等から提供されるグラフ、チャートおよびその他のデータを含んでよい。

#### 【0029】

図3は全体として、グラフィック要素およびグラフィック表示の開発、およびこれら要素および表示が存在しうる2つの環境、具体的には構成環境70および実行時環境72におけるグラフィック表示を示す。一般的に、グラフィック要素74（別々の要素オブジェクト74a、74b等として表現されている）およびグラフィック表示76（別々の表示オブジェクト76a、76b等として表現されている）の形態での表示オブジェクトは、例えば表示エディタ50を用いて構成環境70で生成される。オブジェクト74および76は、生成された後、データベース52に記憶されることができる。オブジェクト74および76は、プロセスプラント10内の特定のハードウェアまたは論理的要素にバインドもしくは結合されないジェネリック（汎用）オブジェクトを定義するクラスオブジェクト（ここでは、表示モジュールクラスオブジェクトと呼ぶ）として生成されてよい。しかしながら、クラスオブジェクトは、クラスオブジェクトと同じ基本的なプロパティを有して、プロセスプラント10内の特定のハードウェアに結合もしくはバインドされる実行時グラフィカルオブジェクトを生成するのに使用することができる。しかしながら、一般的にクラスオブジェクトは、それからインスタンス化された子オブジェクトに依然として結びつけられており、したがってクラスオブジェクトに対する変更は、子オブジェクトに自動的に伝播されることができ、これはこれら子オブジェクトが実行時環境内でインスタンス化される場合でも同様である。

#### 【0030】

図3に示すように、グラフィック要素オブジェクト74のそれぞれは、多くの異なる状況で、グラフィック要素を有用にする多数の構成要素を含んでいる。具体的には、各グラフ

ィック要素74は、一つ以上のビジュアライゼーション（視覚的特徴）77、任意の数のパラメータまたはプロパティ78、スクリプトまたはトリガを用いて実行できる任意の数の動作またはアニメーション79、及びバインディング80を含んでいる。一般的に、各ビジュアライゼーション77は、グラフィック要素74が実行時環境72で実行される際、表示スクリーンに実際に表示されるべき視覚的プロパティまたは要素を定義する。典型的にはビジュアライゼーションは、物理的または論理的デバイスまたはデバイスのグループの表現を定義するが、ビジュアライゼーションはその他のエンティティを表現することもできる。ビジュアライゼーション77は、エンティティのグラフィカル描写の仕様を定義する任意の所望の記述またはプログラミングパラダイムを用いて実行時環境72で実行されることができる。一実施形態では、ビジュアライゼーション77は、PGXMLまたはAvalonコントロール（これらはMicrosoft（登録商標）によって供給される周知のコントロールであり、オブジェクトベースであるため、ビジュアライゼーションを標準のWindows（登録商標）タイプ表示で容易に実行可能にし、表示環境間でポータブルにする方法を提供する）を用いて実行してよい。この機能は、以下でより詳細に論じると共に"Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface"（プロセスプラントユーザインタフェースにおけるマークアップ言語ベースの動的プロセスグラフィックス）と題する同時係属出願（弁理士整理番号06005/41127）説明されており、これを参照により本明細書に援用する。

10

#### 【0031】

一般的、パラメータおよびプロパティ78は、ビジュアライゼーションによって表現されるエンティティと関連する変数またはその他のプロパティ（例えば、静的または可変の固有（intrinsic）プロパティ）を定義し、これらプロパティは要素74の作成者によって定義できる。同様に、動作およびアニメーション79はルーチンまたはプログラム（プロパティに対して変換を行ない、プロパティ値等に基づいてプロセスエンティティの状態を検出するスクリプトとして実行できるルーチンまたはプログラム）を定義し、アニメーションルーチンは、ビジュアライゼーション77が表示スクリーン上に表現される時に、ビジュアライゼーション77に基づいて又はそれを用いて行なわれるべきグラフィックビジュアライゼーションまたは挙動を任意のルーチン、または、ユーザがビジュアライゼーション77を使用又はそれと対話して、プロセスの変化（例えば、プロセスへの入力の変化）を発生させることを可能にするルーチンを含むことができる。これらの動作およびアニメーションは、ビジュアライゼーション77に、より興味を持たせる、理解可しやすい、または、有用なグラフィカルプロパティを提供し、ユーザが、ビジュアライゼーション77と対話することを可能にする。場合によっては、これらの動作またはアニメーションは、ビジュアライゼーションの各種の構成要素の色、サイズ（例えば、高さおよび幅、線サイズ、フォント等）の変化、塗りつぶしの色、およびアニメーション（例えば、色の変化、回転、サイズおよび縮尺の変化、傾斜角の変化等）の形態を取ることができる。これらの動作およびアニメーションは、グラフィカルプロパティ並びにユーザ対話プロパティをグラフィック要素74に提供するものである。静的または固定バインド、またはエイリアス（別名）を用いるバインドであってよいバインディング80は、グラフィック要素74が、実行時環境72で表示の一部として実行される際に、パラメータまたはプロパティ78を実行時環境72内のデータ、タグまたはその他のエンティティにバインドする方法を定義する。基本的には、各グラフィック要素74のためのバインディング80は、グラフィック要素74を、プラント環境内のどこかで定義される一つ以上のエンティティまたはデータ要素に結合する方法を確定し、したがって、実際の実行時環境72とグラフィック要素74との間のインタフェースを定義する。

20

30

40

#### 【0032】

図3に示すように、グラフィック表示オブジェクト76のそれぞれは、例えば一つ以上のグラフィック要素81、コネクタ要素82、動作およびアニメーション83、プロパティ84およびバインディング85への参照またはそのコピーのような、多数の構成要素を含んでいる。一般的に、グラフィック表示76は、パイプ、ライン、コンベヤベルト等を表わすコネクタ

50

要素82と共に視覚的に接続されることができ各種のグラフィック要素81の相互作用を表現する表示であってよい。このようなコネクタオブジェクトは、米国特許出願第2004/0153804号公報に記述されている。図3における点線は、グラフィック表示オブジェクト76aによるグラフィック要素74のうちの一つへの参照を示す。グラフィック要素74を参照するグラフィック表示76は、そのグラフィック要素74のプロパティ、パラメータ、動作およびアニメーション等の全てを含んでいることが理解されよう。グラフィック要素74と同様に、各グラフィック表示76は、それに関連する一つ以上の追加の動作またはアニメーションを含んでよく、これらは例えば、表示上のアニメーション、ユーザインタフェース対話、データ操作等を行うことができる。同様に、各グラフィック表示76は、表示と関連する任意の数のプロパティを含んでいてよく、一般にこれらプロパティは、表示内で表現されるユニット、エリア、または要素の他のグループのプロパティを定義する。もちろん、バインディング85は、グラフィック表示76を、プラント環境内のどこかで定義される一つ以上のエンティティまたはデータ要素に結合する方法を定義し、したがって、実際の実行時環境72とグラフィック表示76との間のインタフェースを定義する。

10

20

30

40

50

#### 【0033】

グラフィック要素74およびグラフィック表示76は、ひとたび生成されると、例えば図1のワークステーション20~23のうち任意のものにおいて、実行時環境72にバインドされ、そこで実行されてよい。具体的には、グラフィック要素74またはグラフィック表示76がクラスオブジェクトとして生成され、データベース52に記憶された後、その要素または表示は、実際の実行時オブジェクトとしてインスタンス化され、実行時環境72で実行されることができる。ブロック86で示すように、インスタンス化プロセスは、オブジェクト74および76で定義されたバインドをフィルイン（記入）する。これは、プロセスプラントまたはプロセス制御システム内の適切な変数名、タグ、エイリアス等をロードして、プロセスプラント内の実際エンティティと、プラント10内の表示デバイス上で実行されるグラフィックオブジェクトとの間に特定の接続を提供することができる、一つ以上の解決テーブルを用いて達成できる。バインドプロセスの一部として、オブジェクト74および76は、解決テーブルによって定義されるプロセスプラント内のデータソースに接続し、それによりプラントにアクセスして、プロセスプラント10に論理的かつ通信により接続される。

#### 【0034】

ブロック87で示すように、表示要素74またはグラフィック表示76は、実行時環境72内の、いくつか挙げるならば構成表示、制御オペレータ表示、保守表示およびシミュレーション表示を含む多数の異なる機能内で、または、それらの一部として実行されることができる。加えて、表示オブジェクト74および76は、例えば図2で表現されている例えば予測制御または予測保守機能、システムレベルエラー検出、診断等を含む機能レベルのうち各種のものから得たデータを用いるような、システムレベル機能を行なう際に使用することもできる。事実上、表示76は、ひとたび構成環境70で生成され、データベース52に記憶されたら、多数の異なる活動において使用されてよい。更にまた、表示オブジェクト74および76は、任意の所望の表示またはコンピュータハードウェア（例えば、ワークステーション90、ラップトップコンピュータ91、携帯情報端末（PDA）や電話機等のようなハンドヘルドデバイス92）、または、任意の他の特殊ディスプレイ93（例えば、複数のモニターを有する大形スクリーンディスプレイ等）上で実行されてよい。所望なら、単一のグラフィック表示76は、一つ以上のビュー（例えば、構成ビュー、オペレータビュー、保守ビューおよびシミュレーションビュー）を含むよう、層（レイヤ）状にされてよい。或いは、別々のグラフィック表示76が、これら別々のビューを提供し、これらの各種の機能に対して生成される表示にわたって首尾一貫した様相および印象を提供するように、同じまたは同様のグラフィック要素81を用いて構成されてよい。

#### 【0035】

ブロック95で示すように、実行時環境72に移植されるため、グラフィック要素74またはグラフィック表示76はコピーまたはインスタンス化され、実行時マシンにロードされてよい。一般的に、表示オブジェクト74または76は呼び出されるか、あるいは、実行時マシン

上で実際に実行される時にのみ実行時環境72にバインドされることが望ましく、これをここでは実行時バインドと呼ぶ。すなわち、インスタンス化されたオブジェクトのそれぞれに対する解決テーブルは、表示オブジェクトが、実行時コンピュータ内で実際に動作または実行される時にのみ、実行時環境にフィルインまたはバインドされる。この手続きは、ビジュアライゼーション、制御、スクリプト等を含む表示オブジェクトが、オブジェクトの一つ以上のビジュアライゼーションが表示スクリーン上に実際にレンダリングされる時にのみ実行され、従ってその時のみ処理パワーを使用することを保証する。したがって、表示オブジェクトは好ましくはオブジェクトが実行時コンピュータ上で実際に実行される時にのみ実行時環境72にバインドされ、これは表示オブジェクト74および76を、これらオブジェクトによって生成されるビジュアライゼーションを見るユーザの活動によって規定されるように、実行時環境72に断続的に接続できることを意味する。具体的には、これらオブジェクトはそれらを見るのが要求された時点で実行時環境72にバインドされてよく、またユーザから見られない場合（例えばユーザが、これらオブジェクトがビジュアライゼーションを提供しているスクリーンを、最小化あるいは閉じる場合）には、アンバインドまたは解放してよい。

10

**【0036】**

表示オブジェクト74および76は、したがってスタンドアロン環境、すなわち構成環境70で生成されてよいオブジェクトであるが、プロセスプラント環境内で定義される他のオブジェクトまたはデータ構造、または、例えば、任意の制御、シミュレーション、保守、または構成環境内で定義されるオブジェクト、データ構造、アプリケーション等を含む、プロセスプラント環境内で実行される任意のアプリケーションと結合または接続されてよい。更にまた、表示オブジェクト74および76は、ひとたび生成されたら、直接参照、変数または解決テーブルによって定義されたタグを介して直接的に物理的または論理的プロセスエンティティにバインドされてよく、或いは、表示オブジェクト74または76が実行時環境72内でダウンロードまたはインスタンス化される時、あるいは場合によっては、表示オブジェクト74または76が実行時環境72内で実際に実行されている時のいずれかに解決されることができる、エイリアス名称、変数およびパラメータの使用を通じて間接的に、物理的または論理的プロセスエンティティにバインドされてよい。

20

**【0037】**

図3の表示エディタ50は、表示オブジェクト74および76の使用の容易さおよび汎用性を高めることを可能にするために、表示オブジェクト74および76を各種のレベルの詳細さで生成することができる。例えば、より基本的な物理的および論理的エンティティのプロパティおよび動作を定義するためにグラフィック要素74をまず生成し、次いで、より複雑な物理的または論理的エンティティ、または物理的または論理的エンティティのグループを表現する、より上位レベル又はより複雑な表示を生成するために、一つ以上のグラフィック要素74を相互接続することによってグラフィック表示76を生成してもよい。もちろん、グラフィカル要素74およびグラフィカル表示76の両方は格納および各種の異なるカテゴリからアクセスされることにより、より上位レベルの表示オブジェクトの生成をユーザにとってより簡単にすることができる。

30

**【0038】**

図4は、表示エディタ50によって生成されることができ、例としてのスクリーン表示100を示す。スクリーン100（ポンプのグラフィック要素を生成するプロセスを表現している）は、メイン編集部102、要素階層部104、プロパティ定義部106およびビジュアライゼーション部108を含んでいる。メイン編集部102はユーザまたは設計者に作業スペースを提供し、グラフィック要素に対するビジュアライゼーションを定義または生成し、即ちグラフィック要素（この場合はビジュアライゼーション109で示すポンプのプロパティ）を定義する。一般的、要素階層部104は、階層的なビューまたはツリー構造を用いて、メイン編集部102内のビジュアライゼーション109と関連する構成要素を提供する。図4の例では階層部104は、メイン編集部102で定義されているビジュアライゼーション109が、円（Circle1と命名）および2つの長方形（Rect1およびRect2と命名）のプリミティブ

40

50

またはサブ要素を含んでいることを示している。図4には示していないが、階層部104は、ビジュアライゼーション109に対して定義されるアニメーション、動作および他の表示特徴（例えば、スクリプト、ビジュアルトリガ等）の表示を含んでいてよい。

#### 【0039】

プロパティ定義部106は、編集部102に示したビジュアライゼーション109と関連したグラフィック要素に対して現在定義されている固有プロパティを含むプロパティの全てを示す。図4の例としてのスクリーン100は、ビジュアライゼーション109と関連するポンプがオンであるか、オフであるかを定義するIsOnプロパティ、およびビジュアライゼーション109と関連するポンプの速度を定義するSpeedプロパティを含む2つのプロパティを示す。ユーザまたは設計者は、プロパティ定義部106内で他の変数やプロパティ等の名前、タイプおよびバインドを定義することによって他のプロパティおよびパラメータをグラフィック要素に追加し、それによりグラフィック要素の他の様相を定義することができる。図4の部106に示した2つのプロパティは、プールおよび浮動小数点変数であるとしてリスト（一覧）されている。しかしながら、他の種類の変数も、代わりにまたは加えて使用できる。したがって、例えば部分106で定義されるプロパティは、アレイ（配列）、テーブル（表）、列挙リストまたは任意の他の種類の変数またはデータ構造であることができる。

#### 【0040】

所望なら、スクリーン100内で定義されているグラフィック要素は、それと関連した複数のビジュアライゼーションを有していてよい。これらビジュアライゼーションのうちの様々なものは、ビジュアライゼーション部108で描写されてよく、また個別に選択されてメイン編集部102に配置されてよい。例えば、2つのビジュアライゼーション110Aおよび110Bが図4のビジュアライゼーション部108に示してあるが、任意の他の数およびタイプのビジュアライゼーションが生成されているグラフィック要素に対して定義できる。グラフィック要素生成プロセス中、各ビジュアライゼーションは、例えば、部108のビジュアライゼーションの表示上での右クリックまたはダブルクリック、ビジュアライゼーションの表示の編集部102へのドラッグおよびドロップ等によって、メイン編集部102に配置されてよい。ひとたびメイン編集部102に配置されると、ビジュアライゼーションは編集され、その表示プロパティを定義または再定義されてよい。一般的に、ビジュアライゼーションのうちの 하나가生成されているグラフィック要素に対するデフォルト（初期設定）ビジュアライゼーションとして設定または定義さうる。このデフォルト設定は、例えばビジュアライゼーション部108でデフォルトビジュアライゼーションをハイライトしたり、図4のビジュアライゼーション110Aについて図示されるように点線で囲んだり、または任意の他の方法等の何らかの方法で示されてよい。

#### 【0041】

一般的に、要素エディタ50によって生成されるビジュアライゼーションは、定義された方法で配置またはグループ化される各種の形状の一つ以上のコンポジット（合成物）からなっていてよい。これらビジュアライゼーションは、したがって形状コンポジットと呼ぶことができる。例えば、形状コンポジットは、円、線、点、多角形、四角形、長方形、三角形、または他の単純なグラフィック形状を含んでいてよい。この方法で定義する場合、個々の動作またはアニメーションは、形状コンポジットを構成する異なる形状のそれぞれに適用されるか、あるいは、それと関連付けられてよい。もちろん形状コンポジットは、要素のより複雑な芸術的表現を含んでいてよい。形状コンポジットを定義または構築するため、ユーザまたは設計者は、メイン編集部102に任意の数のプリミティブグラフィック要素を加え、これらプリミティブを任意の所望の方法でグループ化することができる。ひとたび生成されると、形状コンポジットは実際のオブジェクト（これは実行時にXAMLオブジェクトとして実行される場合がある）に対するビジュアライゼーションを定義し、これはグラフィック要素を実行時環境で使用される時、スクリーンまたは表示上に、ビジュアライゼーションとして表示されることができる。形状コンポジットを構成する形状またはプリミティブは、階層部104のコンポジット階層に示されてよい。

#### 【0042】

単一のグラフィック要素に対する複数のビジュアライゼーションの生成は、異なるビジュアライゼーションが異なる状況で、または異なる用途のために使用されることを可能にし、したがって共通の要素の異なるビジュアライゼーションを表示しながらも、プラント内で異なる目的のために生成される表示は、同じグラフィック要素から生成（すなわち、同じグラフィカルクラスオブジェクトから生成）されることができる。例えば、ビジュアライゼーション110A、110B等のそれぞれを異なる機能的コンテキストで使用してよく、グラフィック要素が制御オペレータの表示の一部として使用される場合は第1のビジュアライゼーション110Aを使用してよく、グラフィック要素が保守表示の一部として使用される場合は第2のビジュアライゼーション110Bを使用してよく、また、ポンプ要素がシミュレーション表示で使用される場合は第3のビジュアライゼーション（図示せず）を使用してよい。一方で、異なるビジュアライゼーションを異なるタイプの表示デバイス上での表示目的のために使用してもよい。例えば、グラフィック要素の第1のビジュアライゼーションは典型的なコンピュータまたはワークステーション上での使用に適しているよく、一方異なるビジュアライゼーションをPDAまたは電話機のようなハンドヘルドデバイス上で有利に使用されるように生成してよく、また一方、さらに別のビジュアライゼーションを、大形スクリーン表示または複数のスクリーンモニター用に生成してもよい。ここで分かるように、グラフィック要素用の異なるビジュアライゼーションは、ビジュアライゼーションを表示しようとする表示のサイズに基づいて調整されてよく、したがって、より大形のスクリーン（例えば典型的なコンピュータスクリーン）に適した第1のビジュアライゼーションを生成してよく、また、相対的に小さなスクリーン（例えば、PDAまたは無線電話機のようなハンドヘルドデバイスの表示スクリーン）により適した第2のビジュアライゼーションを生成してもよい。このように、同じグラフィック要素に対して異なるビジュアライゼーションを異なる実行時デバイスで使用することができる。

#### 【0043】

一方、グラフィック要素の開発サイクルにおける異なる時点で、異なるビジュアライゼーションをグラフィック要素に加えることができる。例えば、快適な図面またはビジュアライゼーションの作成に適応していない構成エンジニアは、グラフィック要素を初めて生成する場合、予備的かつ基本的な性質の第1の基本ビジュアライゼーション（例えば、棒線画タイプの図面）を作成し、このビジュアライゼーションをその時点でのグラフィック要素として記憶させる可能性がある。より後の日付または時点で、熟練した製図技術者または技能者が、より複雑かつ芸術（技巧）的に快適なビジュアライゼーションを生成してもよく、その第2のビジュアライゼーションを代替的ビジュアライゼーションとしてグラフィック要素に加えることができる。所望なら、その時点で第2のビジュアライゼーションを、デフォルトビジュアライゼーションとして設定することができ、この第2のビジュアライゼーションをマスターまたはクラスグラフィック要素からインスタンス化された実行時グラフィック要素の全てに亘って伝播させることにより、この第2のビジュアライゼーションは、実行時表示で現われるか、あるいは、実行時表示での使用に利用可能にできる。

#### 【0044】

別の例では、異なるビジュアライゼーションを、同じグラフィック要素に提供して、表示の異なるテーマ、グラフィカル標準、基準（norm）またはスタイルをサポートしてよい。既知のように、異なる業界では一般に、ポンプ、バルブ、センサー、および他のプロセスプラントエンティティを表現するのに異なるグラフィック基準またはグラフィカル標準を使用する。したがって、オイルおよびガス業界は、製薬業界とは異なるグラフィカル標準を使用する。要素エディタ50を用いることにより、各グラフィック要素に異なるビジュアライゼーションを与えて、例えばオイルおよびガス基準および製薬基準のような、複数のグラフィカル標準または基準をサポートすることができる。この方法で、同じグラフィック要素を生成されるグラフィック表示で使用して、異なる標準または基準またはテーマの表示、例えば異なる業界でアクセス可能あるいは有用な表示をサポートできる。更にまた、異なるグラフィカルスタイル（例えば芸術的なスタイル）を、異なるビジュアライゼ

ーションで使用して、ユーザが各種の芸術的なタイプの表示を生成することを可能にしてよい。もちろん、これらは、同じグラフィック要素に複数のビジュアライゼーションを提供することの利点のほんの2、3の例にすぎず、これら複数のビジュアライゼーションの他の用途も存在する。

#### 【0045】

所望なら、グラフィック要素は、一つ以上の定義された、またはそれと関連するグラフィカル挙動を有してよい。具体的には、設計者または作成者はグラフィック要素のビジュアライゼーションがスクリーン上で表示される時、各ビジュアライゼーションに対して、アニメーション（例えば、回転、線形移動、背景変化、色変化、リサイジング、色グラデーションアニメーション、不透明アニメーション、フォント特性アニメーション、映像および映像特徴（例えば、起動/停止特徴）、2次元または3次元変化）を定義してよい。この動的挙動を加えるため、ユーザは、グラフィック要素を選択し、アニメーション（アニメーションルーチンとも呼ばれる）を選択してビジュアライゼーションに加えてよい。この点で、ユーザは、選択されたアニメーションに係る構成情報を入力してよく、あるいは、スクリプトを用いて所望の挙動を定義し、これを例えばダイアログボックス、式エディタ等を用いて入力してよい。これらスクリプトは定義された場合、階層部104の階層に現われてよい。基本的にスクリプトは、表示要素に対するビジュアライゼーションが実行時活動中にスクリーン上で観られたり提示される時、表示要素の一部として動作あるいは実行されるプログラムまたはルーチンである。なお、そうである必要はないが、これらの挙動またはスクリプトはグラフィック要素に対して定義されるプロパティまたはパラメータのうちの一つ以上に結合され、それに対して動作してよい。例えば、スクリプトはグラフィック要素に関連付けられて、グラフィック要素のプロパティ部106で定義されるプロパティのうちの一つの値に基づいて、グラフィック要素のビジュアライゼーション内で色を変えることができる。例えば、ポンプビジュアライゼーション109の色は、IsOnプロパティ（これは実行時環境にバインドされる）が、Trueである、すなわち、ポンプがオンである時、黄色から緑色に変化することができる。別の例としては、ポンプ要素に対して、ポンプのSpeedプロパティの値を設定点と比較し、Speedプロパティの値があるレベルを超える場合、ビジュアライゼーション109内である種のグラフィックアニメーションを生ずるスクリプトを定義してもよい。このアニメーションは、例えば、ポンプを赤に変えること、ポンプを回転させること、ポンプ内でのモータの回転を示すこと、ポンプビジュアライゼーションを脈動または振動させること等を含んでいてよい。

#### 【0046】

例えばアニメーションまたはその他の挙動をビジュアライゼーションに提供するために、ビジュアライゼーションに対して定義されるスクリプトは、ビジュアライゼーションを構成する形状コンポジット内の形状またはプリミティブのうちの一つのものに対し、または、形状コンポジット内の形状のうちの一つのものに対して動作するよう設計することができる。スクリプトは、実行時環境で実際の物理的要素にバインドされる固有プロパティに対して動作する、あるいはそれを使用するため、これらスクリプトはビジュアライゼーションが実際のプラント運転に基づいて、あるいはグラフィック要素と関連する実際のエンティティのプロパティを反映する、プロセスプラントの他のエリアからのデータに基づいて、変化することを可能にする。

#### 【0047】

異なるタイプのグラフィック要素は、それに適した異なるグラフィックスを有してよいことが理解されよう。したがって、グラフィック要素に与えられることができる挙動は、本明細書で与えられた例によって限定されない。例えば、回転機器に対するグラフィック要素は、振動グラフィックス、運動、色変化等を提供するスクリプトを含んでいてよく、センサーのような機器に対するグラフィック要素は、限界超えまたは限界未満条件、較正のニーズ条件等を表現するスクリプトを含んでいてよい。もちろん、グラフィック要素に対して任意のプロパティを使用または定義してよく、これらプロパティは一般に表わされているエンティティのタイプに基づいている。異なるビジュアライゼーションに対し

10

20

30

40

50

て異なる挙動を提供するために、異なるビジュアライゼーションのそれぞれに対して各種のスク립トを提供されてよいことも理解されよう。一方、いくつかのスク립トは、グラフィック要素と関連したビジュアライゼーションのそれぞれに対して挙動を与えるために使用されてもよい。

#### 【0048】

加えて、スク립トまたは他のルーチンはグラフィック要素と関連付けられて、グラフィック要素に対するパラメータの一つ以上に基づく関連する物理的エンティティについてのある条件を検出してよい。これらの条件は、エンティティと関連する物理的状態、通信状態、デバイス状態、値状態等を含むエンティティの状態を検出することを含んでよい。これらの検出された条件または状態は全て、前記状態または値に結合され、グラフィックスク립トによって生成されたアニメーションまたは他の動作または挙動を用いてグラフィックに反映されてよい。例えば、ポンプ要素には、その要素の一つ以上のパラメータ、即ち図4のプロパティ部106で定義または呈示された一つ以上のパラメータに基づいて、関連するポンプの過熱状態を検出するスク립トが提供されてよい。別の例としては、ポンプの過度な振動または任意の他の条件を検出するスク립トをグラフィック要素に対して提供してよい。例えば過熱または過度な振動のような特定の条件が検出された場合、アニメーションまたはグラフィック要素と関連した他の動作が、その条件に対して動作して、グラフィック要素のビジュアライゼーション内で、またはそれに対して、この状態を示すグラフィカル表示を提供してよい。

#### 【0049】

図5は、表示エディタ50で作成されることができるとスクリーン表示112の別の例を示す。図4のスクリーン100と同様に、スクリーン表示112は、メイン編集部114、パレットビュー116、階層ビュー118およびプロパティビュー120を含んでいる。反応器要素122の第1のビジュアライゼーションが、メイン編集部114で表現されている。階層ビュー118に示すように、要素のタイトルは、Reactor1であり、この要素は、Visual1（デフォルトビジュアライゼーション）、Visual2およびVisual3と称する3つのビジュアライゼーションを含んでいる。階層ビュー118において、Visual1の見出しの下に示したように、第1のビジュアライゼーションは、長方形要素および楕円形要素を含むキャンバス背景から成っている。現在定義されているプロパティ（この場合、ビジュアライゼーションの名前、高さおよび幅）が、プロパティビュー120にリスト化されている。階層ビュー118でビジュアライゼーションが選択されると、ビジュアライゼーションと関連した子プリミティブまたは要素の全てが、編集部114に示され、現在選択されている要素のプロパティがプロパティビュー120に表示される。

#### 【0050】

スクリーン112では、パレットビュー116はビジュアライゼーションを生成するのに使用できる多数の基本的な要素を含んでいる。例えば、パレットビュー116は、一組の基本的なUI（ユーザインタフェース）要素（例えば、ボタン、テキストボックススライダ、ノブ等）、一組の基本的なパネル、および、一組の基本的な形状を含んでいる。定義されたパネルは、ユーザが要素をキャンバスエリアに対する座標によって明示的に配置できるエリアを定義するキャンバスパネル、ユーザが要素を互いに他に対して水平または垂直のいずれかに配置できるエリアを定義するドックパネル、および、フロー方向を指示すると共に、フローパネルエリア内でその内容をブレイクし、ラップみ、整列させるのに使用してよいフローパネルを含んでよい。フロー方向は、上、右、左および下の任意の組合せであってよく、例えば、左から右へおよび上から下へ、または、右から左へ、トップからボトムへ等であってよい。更にまた、パレットビュー116内の基本的な形状は、ISA（Instrument Society of America）シンボル、トランスミッタシンボル、バルブシンボル、PI&Dダイアグラムシンボルまたは他の制御シンボル等、または任意の他の所望の形状を含んでよく、その全てがグラフィック要素を構築するのに使用できる。

#### 【0051】

もちろん、他の基本的な要素（例えば、基本的な制御要素、デバイス等）を、プリミテ



ィブとしてパレットビュー116に提供され、定義されているグラフィック要素に対してビジュアルライゼーションを生成するのに使用できる。パレットビュー116はまた、ユーザ定義されたカテゴリまたは要素のリストを有してよく、これはユーザがパレットビュー116から編集ビュー114に任意の他の有用な形状をドラッグできるようにするために生成されてよい。所望なら、そのようなユーザ定義されたカテゴリへのアクセスはそれらを生成したユーザに限定されてよく、またこれらのカテゴリおよび要素は、特定のユーザに関連するユーザ選択ファイルに記憶させてもよい。しかしながら、組み込みカテゴリおよびパレットアイテムは、データベースに記憶されていてよく、全てのユーザにグローバルに利用可能であってよい。いずれにしても、パレットビュー116はグラフィック要素を構成するのに使用できるプリミティブ要素のライブラリを表示し、または、それにアクセスするのに使用されてよく、これらのライブラリは、ロック、バージョン管理、特定のユーザに限定すること等が、必要に応じて可能である。

10

20

30

40

50

#### 【0052】

上記のように、任意のグラフィック要素に対するビジュアルライゼーションは、それと関連したアニメーションおよび/または動作を有することができ、そのようなアニメーションまたは動作はスクリーン112の動作/アニメーションビュー123に示されてよい。ビジュアルライゼーションがアニメーションまたは動作を含む場合は、これらのアニメーションまたは動作も、例えば、星印等の特殊なシンボルにより階層に示すことができる。階層ビュー118で選択された場合、ビジュアルライゼーションまたはビジュアルライゼーションのサブ要素に対して定義された任意の動作またはアニメーションは、動作/アニメーションビュー123に示される。動作またはアニメーションは、ビュー123でそのような動作またはアニメーションを定義することによって、あるいは階層ビュー118にそのような動作またはアニメーションを加えることによって、生成及び指定されてよい。ユーザが、動作またはアニメーションを生成または編集することを望む場合は、エディタ50はダイアログまたは編集ボックスを提供して、この機能を十分に指定または定義できるようにしてもよい。もちろん、動作またはアニメーションは、スクリプト、ビジュアルトリガまたは他のプログラムを用いて定義することができる。

#### 【0053】

スクリーン112の使用中に、ユーザまたは設計者はパレットビュー116で異なるアイテムをドラッグおよびドロップし、そうでなければ選択し、所望のビジュアルライゼーションを生成するためにこれらのアイテムを編集ビュー114で配置することによって、要素のビジュアルライゼーションを生成することができる。一つ以上のツールバー124は、任意の標準の編集機能、例えば、新しいビジュアルライゼーションまたはアニメーションの追加、削除、移動、編集、プリミティブまたは要素を例えば背後から前方に配置すること、接続要素を提供して生成されているビジュアルライゼーションへの接続を示す又はそれを可能にすること、異なるプリミティブをグループ化してそれらが互いに他に対する位置を保持するようにすること、線およびテキスト等のような静的要素の追加等を提供するのに使用できる。

#### 【0054】

図6に示すように、グラフィック要素130は、複数のビジュアルライゼーション132またはそれと関連したビジュアル表現を有してよく、異なるビジュアルライゼーション132は異なる時点で、または、グラフィック要素130が使用される異なる表示で、使用されてよいことと理解される。上述のように、任意のビジュアルライゼーション132は、任意の数のプリミティブ134並びにアニメーションおよび動作136からなっていてよい。更にまた、グラフィック要素130は任意の数のプロパティまたはパラメータ138を含んでいてよく、これらは、動作およびアニメーション136に結合されるかまたはそれによって使用されて、ビジュアルライゼーション132への変更を行ない、ビジュアルライゼーション132と関連したビジュアルトリガを動作させることができる。更にまた、ビジュアルライゼーション132またはビジュアルライゼーション132を構成する個々のプリミティブは、例えば、マウスオーバーイベント、マウスクリックイベント等の予め定義されたイベントに対して定義された動作

を有してよい。これらの動作（これらをモルーションと呼ぶ）は、イベントハンドラーを、設定または定義可能にして更にグラフィック要素130の挙動をカスタマイズできるようにし、ユーザがビジュアライゼーション132と対話して例えば実行時環境内で変化を生じさせることができるようにする。具体的には、ユーザは、値または他の数または情報をビジュアライゼーション132に入力する、ビジュアライゼーション上のスライダバー等の要素を動かす、あるいは、何らか他の動作を行って例えばビジュアライゼーション内のプロパティを変えることによって、ビジュアライゼーション132と対話することができる。このビジュアライゼーションプロパティの変化は、スクリプトを介して、あるいは直接に、プロセス入力（例えば、プロセス実行時変数）に結合されて、その変数を変えてよい。具体的には、ビジュアライゼーションプロパティは、グラフィック要素に対して定義されたプロパティに接続されてよく、これはさらにプロセス入力にバインドされてよい。このように、ユーザは動作またはアニメーションルーチンを介してビジュアライゼーションと対話し、プロセスまたは他の実行時環境（例えば、シミュレーション環境）への変化を生じさせるか、あるいは入力を与える。

#### 【0055】

所望なら、ビジュアライゼーション132はユーザによって指定されるか、あるいはプログラムの指定されることによって、グラフィック要素130に接続されてよい。具体的には、グラフィック要素130は、例えば、グラフィック要素により表現されている要素のメーカー、または、グラフィック要素130と関連する他のパラメータ（例えば、グラフィック要素と関連したデバイスの状態）に基づいて、ビジュアライゼーションがプログラムの

#### 【0056】

上記のように、図5のプロパティビュー120は、階層的なビュー118で選択されたアイテムに対して、即ちメイン編集ビュー114で表現されたアイテムに対して、定義されたプロパティおよびイベントを提供、あるいは示す。ユーザはプロパティビュー120内で、例えばツールバーボタンを用いて、プロパティ、固有プロパティおよびイベント間をトグルする（切り替える）ことができる。図7は、この場合IsOnおよびSpeedプロパティを含んでいる固有プロパティが示されているプロパティビュー120Aを示す。この場合、プロパティビュー120Aは、これらの変数のデータタイプおよびこれらの変数に対する何らかのデフォルト設定を示す。上記したように、ユーザはこのプロパティリストに追加、削除、または編集して、エニユメレーション、テーブル構造等を含む任意の所望のデータタイプでプロパティを定義することができる。所望なら、プロパティビュー120は、これらプロパティに対して定義された任意の実行時バインドも示してよく、これらの実行時バインドは、固定された変数またはタグであってよく、あるいは、グラフィック要素の実行時マシンでの実行時またはダウンロード時、フィルアウト（記入）されるエイリアスを用いるタグであってもよい。

#### 【0057】

グラフィック要素に対して定義されたプロパティの値は、アニメーション、動作等に対するトリガ条件として使用でき、これらの挙動はグラフィック要素と関連した一つ以上のスクリプトによって定義されてよい。例えば、モータが、True（真）であるOnプロパティを有している場合は、モータ動作のアニメーションをグラフィック表示でトリガできる。更にまた、プリミティブプロパティ、例えば、長方形の形態のプリミティブである塗りつぶされた長方形プロパティは、グラフィック要素プロパティにバインドでき、これによりグラフィック要素プロパティの変化はプリミティブプロパティに作用する。同様に、ビジュアライゼーションに対する変化はプロパティに結合されてよく、これによりビジュアライゼーションに対する変化によりプロパティ値の変化を生じうる。

#### 【0058】

所望なら、エディタ50はユーザが一つ以上の変換機能を指定することにより、グラフィック要素プロパティとアニメーションまたは動作との間のより望ましい結合を提供可能にしてよい。例えば、ユーザは、TankLevel（タンク内の流体の液面レベルを定義する）の

10

20

30

40

50

グラフィック要素プロパティをプリミティブプロパティである塗りつぶされた長方形にバインドして、それによりビジュアライゼーションの一部として定義されたプリミティブの塗りつぶし色で、タンクの水位をグラフィカルに示すことを望む可能性がある。しかしながら、この場合ユーザは、プロパティ (TankLevel) をエニューメレートされたセットまたは条件に変換する変換機能を定義して、TankLevelが第1および第2の液面の間にある場合、塗りつぶされた長方形緑色に設定され、TankLevelが第2および第3の液面の間にある場合、塗りつぶされた長方形が黄色に設定され、TankLevelが第3の液面より高い場合、塗りつぶされた長方形が赤色に設定されるようにしてよい。この変換機能は、スクリプトあるいはグラフィック要素で実行される任意の他のプログラムとして定義されてよく、任意の所望のプロパティ変換 (例えば、プロパティ値を長さ、フォント名称、現地語化された文字列、期間、回転、色グラデーション、不透明度、ブラシパターン等に変えること) を生じさせるのに使用されてよい。同様に、変換ルーチンは、ビジュアライゼーションを介したユーザからの入力を任意の所望のプロパティ値に変換してもよい。

10

20

30

40

50

**【0059】**

別の例としては、変換機能を用いて、グラフィック要素プロパティまたはパラメータを外部参照にバインドしてよい。この場合、エディタ50はユーザが、ソース値 (実行時変数) をデスティネーション値 (グラフィカル変数) に変換する、またはその逆を行なうための、グラフィック要素が実行時環境で使用されると自動的に適用される、一つ以上の変換機能を指定することを可能にしてよい。例えば、速度デスティネーション変数 (これは、速度を数値として与えるデータソースにバインドされてよい) は、ソース変数の値に基づいて、文字列タイプ変数 (例えば、「遅い」、「並」、および「急速」のうちの一つを示す文字列) に変換することができる。このような変換機能は、ユニットを変換する、あるいは、他の変換を提供するのに使用してもよい。いずれにせよ、これらの変換機能は、スクリプトまたは他のプログラムとして実行されてよく、任意の目的に対して使用して、動作またはアニメーションを提供し、あるいは、グラフィック要素に対する表示プロパティに何らかの影響を与えることができる。

**【0060】**

もちろん、プロセス制御内で異なる物理的要素に対して生成された異なるタイプのグラフィック要素に対して任意の所望のプロパティを定義することができる。例えば、ポンプ要素は、ポンプ状態 (例えば、オンまたはオフ)、加圧プロパティ、減圧プロパティ、流入プロパティおよび流出プロパティを含んでいてよい。同様に、アクチュエータと共に使用される調整バルブは、例えば、名称プロパティ、密度増加プロパティ、密度減少プロパティ、流入プロパティ、流出プロパティ、加圧プロパティ、減圧プロパティ、温度上昇プロパティ、温度低下プロパティ、バルブ位置プロパティ、バルブ開およびバルブ閉プロパティ (これは、バルブがいったいに開であるか閉であるかを定義してよい)、設定点、プロセス値、バルブタイプ (例えば、リニア、クイックオープニング、イコールパーセンテージ) 等を含んでいてよい。このリストはもちろん、網羅的であることを意図していない。更にまた、これらプロパティのいずれもが、グラフィック要素のアニメーションまたは動作に接続できる。

**【0061】**

図8は、階層的なビュー118 (図5) 内で、選択された要素 (この場合、全体としてReactor1要素) に対して、グラフィック要素プロパティと関連したアニメーションおよびバインドを確認するのに使用できる第2のプロパティビュー120Bを描いている。図8のプロパティビュー120Bは、IsOnプロパティが、グラフィック要素のビジュアルの各種の構成要素に関係付けられる方法を示す。具体的には、テーブルエントリ140で示すように、ビジュアライゼーションの背景に関係付けられ、IsOnプロパティの値に基づいて動作するアニメーションが一つ有る。テーブルエントリ142は、IsOnプロパティとのバインド関係を示し、ビジュアライゼーション内で、IsOnプロパティを背景要素またはプリミティブに関係付けるアニメーション、スクリプトおよび変換にアクセスするのに使用できる。この場合、ユーザは、テーブルエントリボックス140および142のボタンを選択することによって、追

加的な情報を得ることができる。例えば、テーブルエントリ140でボタンを選択することによって、ユーザはIsOnパラメータの値に基づいて、ビジュアライゼーションの背景色を変えるアニメーションにアクセスできる。このようなアニメーションボタンをクリックすると、エディタ50が、アニメーションの管理および構成を可能にするプロパティアニメーションダイアログボックスを開くようにしてもよい。更にまた、ボックス142でバインドボタンをクリックすることによってバインドを実行及び管理することができ、クリックによりエディタ50はダイアログボックスを開いて、バインドの生成、ビューイングおよび編集を可能にする。もちろん、グラフィック要素の他のプロパティに対して、同様のスクリーンを提供することができ、また、プロパティスクリーン120Bに示されているプロパティのリストは、階層ビュー118で現在選択されているアイテムに依存しうる。

10

**【0062】**

再び図5を参照すると、階層ビュー118でビジュアライゼーションが選択された場合、プロパティビュー120は、ビジュアライゼーションプロパティ（例えば、名前、ビジュアライゼーションに一意の識別子を与えるID、記述、および、このビジュアライゼーションが、グラフィック要素のデフォルトビジュアライゼーションとして設定されているかどうか）を表示する。階層ビュー118またはプロパティビュー120は、ビジュアライゼーションがロックされているかどうか、すなわち、それが修正できるかどうかをも示してよい。

**【0063】**

追加的に、階層部118からコネクタ要素が選択された場合、プロパティビュー120はコネクタ要素のプロパティを示し、これは、コネクタタイプ（例えば、流体ライン、電気ライン、ガスライン、円形または正方形ダクト、コンベヤベルト等）、接続がデバイスに入るものか、デバイスから出るものか（すなわち、グラフィック要素によって表わされているエンティティに対する材料の流れの方向）、必要または許容される接続の最小および最大数、接続の幅および高さ、グラフィック要素に対する接続の位置（例えば、上、左）等の表示を含んでいてよい。

20

**【0064】**

階層ビュー118でプリミティブが選択された場合、プロパティビュー120は、プリミティブのプロパティを示し、これは例えば、ID、カテゴリ記述、プリミティブが選択可能及び/又は可視であるかどうか、サイズ及び長さ或いはその他のプリミティブの形状又は性質の定義、背景色およびプリミティブの塗りつぶしスペース等を含んでいてよい。もちろん、階層ビュー118内で選択された任意の他のアイテムは、そのアイテムのプロパティをプロパティビュー120に表示させ、プロパティのタイプおよび性質は選択されたアイテムの特質に左右される。

30

**【0065】**

所望なら、イベントシートを提供して、階層ビュー118内で選択された階層アイテムのそれぞれ、または、全てと関連したイベントを示してもよい。このようなイベントシートには、ユーザが、グラフィック要素に対して動作を行なった場合に生ずるイベント（例えば、ユーザがグラフィック要素のビジュアライゼーション又はその一部分上でクリックした場合に起こること、またはユーザがグラフィック要素のビジュアライゼーションまたはその一部分上にマウスカーソルを配置した場合に起こること、を定義する「オンクリック」イベントおよび「オンマウスオーバーイベント」）を含めることができよう。このイベントシートの一部として、ユーザにはイベントの発生時に動作するスクリプト（これは、ここでは、ルーチンとして定義される）を定義あるいはそれにアクセスするために、スクリプトエディタへのアクセスを提供されてよい。

40

**【0066】**

追加的に、所望なら、グラフィック要素は、それと関連したカスタムイベントを有することができる。カスタムイベントは、一般にいくつかの外側または外部イベントの結果として定義されるイベントであり、あるいは、グラフィック要素のユーザが取った動作の結果として外部アプリケーションまたはデータソースとの通信を必要とするイベントである。実質的に、グラフィックイベントは、グラフィック要素によって送信されてグラフィッ

50

ク要素のビジュアライゼーションに対する動作の発生を知らせるメッセージである。グラフィック要素のコンテining（グラフィック要素を収容する）アプリケーション又はグラフィック要素を実行する実行時アプリケーションは、これらイベントを、例えば、C#シタックスを用いて、または任意の既知の方法でイベント通知に登録することによって、捕捉できる。一例では、コンテiningアプリケーションは、ポンプ過熱カスタムイベントに登録され、スクリプトまたは他のルーチンを実行することにより、イベントがトリガされた時に、ユーザがイベントを取り扱うことを可能にするマイハンドラー機能を提供してよい。これらカスタムイベントは、グラフィック要素が制御オペレータスクリーンの一部として実行される場合、特に有用である。この場合、グラフィック要素それ自体が、例えば、ポンプ熱プロパティ（これは、外部測定値にバインドされる）を設定点または他の限界と比較することによってイベント条件を判定してよく、イベントをユーザに知らせることによってイベント応答をトリガし、アプリケーション、スクリプト、ダイアログ等を実行して、ユーザがイベント（この場合、過熱ポンプ）を取り扱うまたは処理することを可能にしてよい。このイベントハンドリングプロセスを可能にするため、要素エディタ50はユーザが、イベント引数、すなわちイベントとして認識されるもの、および、イベントの発生から生ずるもの、を定義することを可能にする。

10

20

30

40

50

**【0067】**

もちろん、所望なら、グラフィック要素は、Avalonコントロールクラスから引き継いだ標準のイベントを提供することができる。既知のように、Avalonオブジェクトまたは制御は、Microsoft Longhornオペレーティングシステム用の、Avalonユーザインタフェースアーキテクチャ用Microsoft定義グラフィックスであり、これはユーザインタフェース上でベクトルグラフィックスの描画をサポートする。これら標準のイベントは、例えばフェースプレート表示またはデバイス詳細表示の表示をクリックまたはダブルクリック（マウス）イベントに結合することを含んでいてよい。例えば、クリックまたはダブルクリックイベント（これは、グラフィック要素のビジュアライゼーションの境界線内でのマウスクリックまたはダブルクリックを報告するイベントである）は、ビジュアライゼーションによって表わされているデバイスまたはエンティティのフェースプレートの表示または詳細表示の表示をトリガし、表わされているエンティティの状態、設計、製造等についてのより完全なまたは徹底的な情報をユーザに提供することができる。もちろん、標準のイベントを含む他のイベントを、キーダウン/アップイベント（これは、キーが押されたことを知らせるイベントである）、マウスイベント（マウスエンター、マウスホバー、マウスムーブ、マウスホイール等の動作を含む）、または任意の他のユーザ開始イベントに対して定義してよい。

**【0068】**

したがって、所望なら、グラフィック要素イベントは、プリミティブイベントのイベントハンドラーまたは変換機能等におけるように、グラフィック要素のプロパティ内での変化により、イベントハンドラーのスクリプトでトリガされてよい。更にまた、グラフィック要素のプリミティブは、例えば、マウスおよびキーボードを介するそれらとのユーザ対話によってトリガされることができイベントを呈示してよい。これらイベントは、ユーザが、形状およびプリミティブレベルで、グラフィック要素と対話する選択肢を提供する。開発者は、例えばC#メソッドを用いて、イベントハンドラーを指定することによって、グラフィック要素内で、内部的にイベントを取り扱うことができる。

**【0069】**

再び図5を参照すると、動作/アニメーションビュー123は、編集ビュー114で現在表示されているビジュアライゼーションに対して定義された変換アニメーションおよびプロパティアニメーションのリストを提供または示してよい。ユーザは、例えば、ビュー123内の行上でダブルクリックして、エディタ50にダイアログボックス（これは、変換アニメーションダイアログボックスまたはプロパティアニメーションダイアログボックスであってよい）を呈示させ、それにより、選択したアニメーションが編集できるようにしてよい。一般に、現在選択されているビジュアライゼーションのアニメーションのみが示されること

になるが、Show-all (全てを表示) ボックス144を選択すれば、アニメーションおよびビジュアライゼーションの全てに対する動作を示すことができる。アニメーションに意味のある名称または記述、例えば「animate lever (レバーを動かす)」、「rotate motor (モータを回す)」等が与えられていると仮定すると、動作/アニメーションビュー123は、ビジュアライゼーションに対して定義されたアニメーション挙動の容易な確認、およびそれへのアクセスを提供する。図9は、IsOnプロパティに対して定義された動作を示す例としての動作/アニメーションビュー123Aを示す。この場合、IsOnプロパティが、「True」であれば、Visual1と呼ばれるビジュアライゼーションのRectangle1と呼ばれるプリミティブは、「red (赤)」の記入 (fill) 値で記入動作を実行する。同様に、IsOnプロパティが、「True」であれば、Visual1と呼ばれるビジュアライゼーションのEllipse1と呼ばれるプリミティブは、False値で可視化動作を実行する (例えば、見えなくなる)。もちろん、動作およびアニメーションは、ビジュアライゼーションの別々のプリミティブに対して別々に定義されてよいが、これらの動作およびアニメーションは、同じイベント、トリガまたはプロパティ変化により同時に動作して、より複雑であるが快適なアニメーションを呈示することができる。更にまた、動作およびアニメーションは、単一の動作 (例えば、サイズや塗りつぶし色の増大等)、またはオフにされるまで連続的に起こる反復動作を行なうことを含んでよい。

10

20

30

40

50

#### 【0070】

図10は、ビジュアルトリガ148を、グラフィック要素152のプロパティ (例えば、固有プロパティ等) に基づいて、グラフィック要素152のビジュアライゼーション150上で実装する方法を示す。具体的には、グラフィック要素152の固有プロパティのうちの一つ以上は、ビジュアルトリガ148に対して定義された一つ以上のプロパティトリガ154 (それらの中の3つを図10に示す) によってモニターされる。各プロパティトリガ (これは、スクリプトを用いて実装されてよい) は、点線155で示したようないくつかのグラフィック要素プロパティ値をモニターできる。その後、各プロパティトリガは、モニターされているグラフィック要素プロパティ値が満足されるか、あるいは指定された条件に一致すると、点線156で説明したように、ビジュアライゼーション150の一つ以上のプリミティブプロパティ値を設定することができる。このように、例えば、プロパティトリガ154のうちの一つは、一つ以上のグラフィック要素プロパティをモニターして、これらプロパティの一つ以上の値が、特定の範囲内となる時を判定できる。この条件が満足されると、プロパティトリガ154のうちの一つは、アニメーションまたは他のビジュアルスクリプトを、例えばビジュアライゼーション150のプリミティブまたは他の要素に対して実行することにより、ビジュアルトリガ148を提供することができる。もちろん、プロパティトリガ154のうちの複数のものが共に動作して、ビジュアルトリガ148の一部として、複数の同時の変化またはアニメーションを提供してよく、あるいは、プロパティトリガ154のうちの異なるものが、例えば異なる固有プロパティまたは同じ固有プロパティの異なる値に基づいて独立的に動作することにより、異なる時点で、または、異なるプロセス状態に応じて、ビジュアルトリガ148の異なる動作を提供してよい。この方法で、色変化、アニメーション等を、ビジュアライゼーション150で、グラフィック要素152の固有プロパティの値に基づいて提供することができる。

#### 【0071】

所望なら、エディタ50は、現在選択されているビジュアライゼーションに対して、全てのプロパティトリガをリストするビジュアルトリガパネルを提供あるいは表示してよい。図9には、このようなパネルを示してあり、このパネルでは、要素プロパティコラムはグラフィック要素プロパティ名称をリストしており、第1の値 (Value) カラム (列) は、観察されるグラフィック要素プロパティ値を示しており、ターゲット (Target) カラムはビジュアライゼーションまたは変更されるであろうプリミティブの識別を提供しており、パス (Path) カラムは変更されるプリミティブプロパティであり、また、第2の値 (Value) カラムは、観察されているグラフィック要素プロパティ値が満足されると適用されるプリミティブプロパティである。もちろん、アニメーションおよび他のビジュアライゼー

ションの変更が、この構造を用いて提供されてよい。

#### 【0072】

図11は、ユーザが変換アニメーションを指定するのを支援する、あるいは、可能にするのに使用できる変換アニメーションダイアログボックス160を示す。図示のように、変換アニメーションダイアログボックス160は、ユーザがピクセルにより方向および距離を指定することによりアニメーション内での移動を定義可能にする移動 (Move) 部162、ユーザがアニメーションの回転方向および角度を指定可能にする回転 (Rotate) 部164、および、ユーザが水平および垂直方向の両方においてアニメーションの拡大縮小の度合いを指定すること、および、アスペクト比をロックすべきかどうかを指定すること、を可能にする縮尺 (Scale) 部166を含んでいる。ダイアログボックス160は、ユーザが水平方向および垂直方向の両方において、アニメーションに傾斜 (スキュー) が加えられることを指定可能にする傾斜 (Skew) 部168も含んでいる。設定 (Settings) 部170は、ユーザがアニメーションを連続的にするか否かを定義すること、および、その他の移動、回転、拡大縮小および傾斜動作のそれぞれを可能すること、を可能にする。更にまた、プレビュー (Preview) ボックス172は、アニメーションのプレビューを示すことができる。

10

#### 【0073】

ここで理解されるように、アニメーションは、ある期間に亘って値が変化するオブジェクトである。プロパティアニメーションは、アニメーションをプリミティブプロパティと関連付けることによって達成されることができる。プロパティアニメーションは、テキストの色を変化させる、あるいは、線のような要素を点滅させる程度の僅かさであることができる。一方、プロパティアニメーションは、例えばポリラインの点を動かす等、より複雑であってもよい。もちろん、これらはアニメーションのほんの数個の例にすぎず、その他のアニメーションは、色を加えるまたは変えること、プリミティブのサイズ (例えば、幅、長さまたは点サイズ) を変えること、プリミティブを移動させること、回転させること、傾斜させること、拡大縮小すること等を含んでよい。更にまた、その他のアニメーションを、ビジュアライゼーションの任意のレベルで、例えばプリミティブレベルまたはビジュアライゼーション全体の一部として、提供することができよう。もちろん、所望なら任意の特定のビジュアライゼーションまたはビジュアライゼーションの任意の特定のプリミティブに対して、複数のアニメーションを提供あるいは適用することができる。更にまた、特定のビジュアライゼーションまたはビジュアライゼーションのプリミティブと関連する複数のアニメーションは、プロパティ値の変化に基づいて、または、ユーザ開始トリガリングイベント (例えば、マウスイベント) に基づいて、同時にまたは異なる時点で動作してもよい。

20

30

#### 【0074】

図12は、ユーザが、アニメーションのプロパティを定義あるいは変更することにより、アニメーションを定義可能にするための、エディタ50によって作成されることができるプロパティアニメーションダイアログボックス180を示す。ダイアログボックス186は、境界 (Bounds) 定義部182および時間 (Timeline) 定義部184を含んでいる。境界定義部182は、アニメーションの境界を提供あるいは定義する。具体的には、「from」プロパティは、アニメーションの開始値を定義し、「to」プロパティは、アニメーションの終了値を定義する。時間定義部184は、アニメーションの持続時間、アニメーションの開始時間および終了時間を定義する。持続時間プロパティはアニメーションが完了するのに要する時間の長さを定義し、開始時間プロパティはアニメーションのスタート時間に対する時間オフセットを定義し、また終了時間プロパティは開始時間に対するアニメーションの終了時間を定義する。速度 (Speed) 定義部186は、ユーザが、例えばスライダーを用いて、速度、加速度、および減速度を指定することを可能にする。ここで明らかなように、速度プロパティはアニメーションの速度を定義し、加速度プロパティは、時間が経過するにつれアニメーションをより速く動作させ、一方、減速度プロパティは時間が経過するにつれアニメーションをより遅く動作させる。反復 (Repeat) 定義部188は、ユーザが、例えば指定された回数に亘って、指定されたカウントまたは持続時間に亘って、または、オフになるま

40

50

で連続的に、等のアニメーションが反復される方法を定義することを可能にする。

【0075】

また別のアニメーションダイアログボックス190（これは、ダイアログボックス180を介してアクセスされてよい）は、境界定義部182を用いる時に、アニメーションで境界として使用すべき異なる色を指定するのに使用できる。同様のダイアログボックスが、ユーザがその他の数値でない値をアニメーションのプロパティとして選択可能にするために使用されてよい。図示していないが、他のアニメーションプロパティも、この、または、その他のダイアログボックスを用いて、同様に選択または指定されてよい。例えば、自動反転プロパティは、アニメーションが順方向に完了した後、逆に再生するかどうかを示すブール値を定義するのに使用してよく、また「by」プロパティは、アニメーションがその開始値を変える総量を定義するのに使用してよい。相対速度プロパティは、親アニメーション、すなわち、ビジュアライゼーションの親要素と関連したアニメーションに比較して、アニメーションの時間が経過する相対速度を定義するのに使用してよい。1の値は例えばアニメーションが親アニメーションと同じ速度で進行することを示してよく、2の値はアニメーションがその親アニメーションの2倍の速度で進行することを示してよい、等である。

10

【0076】

図示していないが、ダイアログボックスを用いて、アニメーション内での、アニメーションのフォントプロパティ（例えば、テキストのスタイル、サイズ、フォント、および色）またはその変化を設定することもできる。追加的に、線のエッジまたはラインスタイル、厚さおよび色を設定するのに、また、それらのプロパティを変えるのに、ストロークプロパティダイアログボックスを使用することができる。

20

【0077】

グラフィックダイアログボックスを用いて、グラフィック要素の他の機能を編集してもよい。例えば、グラフィックダイアログボックスを用いて新しいプロセスグラフィックスを生成してよく、これは、グラフィック要素プロパティをグラフィック要素に追加すること、および、グラフィック要素プロパティを実行時環境にバインドすることを含んでいる。このようなインデントダイアログボックスは、ユーザが、制御システムまたは他の実行時環境内で定義された異なるタグまたは変数をブラウズ（閲覧）して、所望のタグ、変数、名前等を突き止めてバインドを行なうことを可能にするブラウザを提供することができる。同様に、他のダイアログボックスを用いて、ビジュアライゼーションを追加し、ビジュアルトリガを追加し、グラフィック要素または表示をブラウズイベントを追加してもよい。

30

【0078】

したがって、ここで理解されるように、グラフィカルエディタ50は、グラフィック要素の固有プロパティを定義するためのサポート、及びグラフィック要素のビジュアライゼーションのためのサポートを提供し、これらは、グラフィックプリミティブまたは形状を生成および操作すること、動的挙動、具体的には、変換アニメーション（回転、移動、拡大縮小、および傾斜）およびプロパティアニメーション（長さアニメーション、色アニメーション等）を加えること、および、動的挙動の実行をトリガする条件を定義することを含む。更にまた、グラフィカルエディタ50はグラフィック要素をデータベースに記憶させる、および、その中で検索するためのサポートを提供し、これは、xmlプロブ(blob)へのグラフィック要素のシリアルライズ処理に対するサポート、および、グラフィック要素ライブラリ内のグラフィック要素を分類することに対するサポートを含む。更にまた、システムは、ユーザ定義されたカテゴリまたはセットにグラフィック要素を格納すること、このようなグラフィック要素をロックすること、または、それらにセキュリティを提供すること、ユーザ定義された、または、他の有用なカテゴリにグラフィック要素を格納すること、グラフィカル要素をバージョン作成すること等を提供する。

40

【0079】

いずれにせよ、グラフィック要素は、ひとたび生成されると、プロセスプラントの実行

50



時環境内のプロセス変数または他のデータにバインドされずに、グラフィック要素データベースに記憶される。この記憶されたグラフィック要素は、任意の表示において必ずしも使用されないが、この要素は、今や、実行時環境にダウンロードすること、および、プロセスプラントまたはプロセス制御システム内の特定の実際またはシミュレートされた物理的要素にバインドすることが可能なので、このような用途に利用できる。グラフィック要素の固有プロパティ（例えば、プロセス変数、設定点、現在の速度等）は、バインドされると、実行時環境内のデータ参照に結合及びバインドされる。

#### 【0080】

図13は、グラフィック要素を、プロセスプラントまたはプロセスプラントで使用されるプロセス制御システム内の実際の物理的構成要素または要素に結合またはバインドすることができる方法を示す。具体的には、各グラフィック要素192は、要素に対するビジュアライゼーションを実質的に定義するXAML194、および関連するスクリプト196、アニメーション198、トリガ200、動作202およびイベント204を含んでいる。グラフィック要素192は、参照テーブル206も含んでおり、これは、XAML194において利用可能あるいは公開（オープン）され呈示されている、XAML194と関連する全ての参照をリストまたは含んでいる。参照テーブル206は本質的に、グラフィック要素のプロパティおよびパラメータ、並びに、スクリプト196、アニメーション198、トリガ200、動作202、およびイベント204によって使用される任意の他の変数または参照からなっているか、あるいはそれらを含んでいる。参照テーブル206内の変数またはエンティティは、他のプログラム内の変数、テーブル、テーブルエントリを、または、プロセス制御システム内のどこかで定義された任意の他のタイプまたは種類のデータを、参照するかまたはそれらにバインドされてよい。

10

20

#### 【0081】

図13に示すように、参照テーブル206内の参照または変数を実際のプロセス制御環境または他の実行時環境に結合するのに、解決テーブル208が使用される。一般的に、解決テーブル208（これは、直接あるいはエイリアスを介して参照を定義することができる）は、グラフィック要素192が、実行時環境内の特定の表示で使用されるよう実際に構成されて実行時マシンにダウンロードされる時、グラフィック要素192に対して提供または生成される。表示内のグラフィック要素192の動作前または動作中に、解決テーブル208は、エイリアスおよび他のパラメータを解決し、これらの解決されたデータ接続を参照テーブル206に結合して、参照テーブル206内の変数とプロセス制御システムまたは他の実行時環境内の実際のデータソースとの間のバインドを提供する。

30

#### 【0082】

実行時中のプロセス制御システム内での処理パワーの不必要な使用を防止するため、グラフィック要素、およびそれらが使用されているグラフィック表示は、いずれの表示スクリーン上でも実際に表示または使用されていない時は、実行時環境にバインドされたままでなくてよい。代わりに、解決テーブル208は、グラフィック要素192が実行時環境内で動作またはスクリーン上に表示されている時にのみ、グラフィック要素192の参照テーブル206にバインドされてよい。

#### 【0083】

各グラフィック要素は、参照テーブル206および解決テーブル208の使用を介してプロセス制御システムまたは実行時環境に結合されるため、また、バインドはグラフィック要素がシステムレベルにおいて生成され実行時環境またはマシンにコピーおよびロードされた後に生ずるため、グラフィック要素192は、別々の時に異なる使用のために、異なるデータソースに別々にバインドされてよい。更にまた、グラフィック要素192は、図2の異なる機能エリア42内のデータソース（例えば、制御活動、保守活動、モデリング活動、シミュレーション活動、構成活動等と関連したデータソース）を含む、任意の数の異なるソースによって生成され、あるいはそれを介して利用可能なデータにバインドされることができる。

40

#### 【0084】

例えば、図14に示すように、表示要素192は、非常に異なるタイプのデータソースで生

50

成されたデータ、あるいは、それから利用可能にされたデータに結合することができ、表示要素192を使用して該データを表示または処理することができる。これらデータソースは、例えば、周知のDeltaV（登録商標）制御システム等のコントローラプログラムのような制御環境データソース210、周知のOPC接続インタフェースを介して他のシステムへの接続を提供するOPCデータソース212、周知のAMSシステムのような保守データソース214、周知のOvationシステムのような、より上位レベルのあるいはビジネス（業務）用システム216、および、競合的な制御アプリケーションのような競合的なシステム218を使用するデータソースすら、含むことができる。この方法で、グラフィック要素192はシステム内の任意のアプリケーションからのデータを用いて、システムの任意のレベルにおける物理的要素のデータおよび/またはビジュアライゼーションを表示するのに使用されることができ、これは、たとえ該データが、競合的なシステムと関連する、共に動作するよう意図されなかったアプリケーションを含む非常に異なるタイプのアプリケーションに由来し、あるいはそこで発生したとしても同様である。したがって、グラフィック要素およびグラフィック要素から生成されるグラフィック表示はシステムレベルにおいて生成されるため、それらを使用して任意の目的のためにプラント内で起こることのビジュアライゼーションを提供することができる。これは、たとえ異なるタイプのソフトウェアが、プラント内の異なるハードウェアおよびソフトウェア構成要素にアクセスし、それらを実行させていても、同様である。

10

**【0085】**

グラフィック要素のモジュール性質の御蔭で、周知の十分にサポートされたデバイス記述言語（DDL）で書かれたデバイス記述（DD）から、グラフィック要素を自動的にまたは半自動的に生成することができる。具体的には、デバイスメーカーは一般に、デバイスと関連したパラメータ、デバイスと通信する方法、デバイスに対する制限等をDDLで定義するDDを、彼らが作る各デバイスに対して提供する。結果として、グラフィック要素生成アプリケーションは、デバイスに対するDDLによるDDを読み取って、デバイスのタイプおよびそのデバイスと関連した重要なパラメータ、制限等を判断することができ、次いで、デバイスに対してグラフィック要素の固有プロパティまたはパラメータとしてこれらパラメータを定義することができる。プログラムはまた、デバイスに対してビジュアライゼーションとして基本的な形状コンポジットを選択または定義してよく、またDDからの情報に基づいて、あるいはデバイスのDDによって定義されたデバイスタイプのデバイスに対して記憶されているテンプレートに基づいて、デバイスに対して基本的な動作およびアニメーションを提供するために使用すべき一つ以上の汎用（ジェネリック）スクリプトを選択してよい。所望なら、このプロセス中に、プログラムは、デバイスについての情報を提供するよう、あるいは、グラフィック要素に対してどんなアニメーション、ビジュアライゼーション、プリミティブ等を使用すべきかについての選択を行うよう、ユーザに問合せてよい。

20

30

**【0086】**

より完全な、または、十分に定義されたグラフィック要素のために、プログラムは、異なるタイプのデバイス（例えば、センサー、バルブ、モータ、タンク等）に対して、各種の汎用グラフィカルテンプレートを記憶してよい。プログラムは、次いで、デバイスに対するDDによって定義されたデバイスタイプに基づいて、使用すべきテンプレートグラフィック要素を決定してよい。所望なら、テンプレートは、グラフィック要素で使用すべき、利用できる各種の選択肢またはオプションを提供するかあるいは有してよく、これらの選択肢はデバイスに対するDD内の情報またはユーザ入力に基づいて決定されてよい。したがって、例えばテンプレートは各種のデバイスのサブタイプ（例えば、センサー）と関連する各種の固有パラメータを提供してよく、プログラムはDD内の情報に基づいてグラフィック要素に対して定義すべき固有パラメータを決定してよい。

40

**【0087】**

各種の基本的スクリプトを、テンプレートの一部として提供してよく、ビジュアライゼーションに挙動を与えるのに使用してよい。ここでも、グラフィック要素で使用すべきスクリプトは、DD内の情報（例えば、デバイスタイプ等）に基づいて、あるいは所望ならユ

50

ーザに対する質問に基づいて、自動的に選ぶことができる。追加的に、スクリプトで使用される各種の制限または変数を、DD内の情報から決定してよい。したがって、例えばDDが、デバイスが1個の回転機器であることを示している場合は、例えば回転グラフィックを提供するスクリプト等のスクリプトがグラフィック要素内で使用あるいは提供されるよう選択されてよく、いくつかの様相（例えば、グラフィックが色等を変える場合の回転グラフィックの速度）はDD内で定義されたデバイスと関連する制限に基づいてよい。これらのリミットは、例えば通常または定格運転速度、定義された速度過剰または速度不足条件または制限等であってよい。別の例としては、デバイスがセンサーである場合は、センサーのハイおよびロー値を、センサーの現在の読み取り値と関連するグラフィックス、センサーが壊れているかどうかを表現するグラフィックス等を提供するのに使用してよい。

10

**【0088】**

この方法では、デバイスがプロセスプラントに取り付けられた時あるいはその中で認識された時に、デバイスのタイプおよびデバイスに対してDD内で定義されたデバイスの既知の特性に基づいて、グラフィック要素に対するいくつかの基本的なスクリプト、グラフィックアニメーション、ビジュアライゼーションおよび固有パラメータを定義することによって、基本的なグラフィック要素がデバイスに対するDDから自動的に生成されてよい。グラフィック要素のこの自動生成は、たとえデバイスに対するグラフィックのエディティングが何か有ったとしても、多くのこと或いは全く何も行なう必要なしに、新しく加えられたデバイスを自動的にグラフィックスに統合する相当な能力をユーザに与え、したがって、デバイスは、それらデバイスに対するDDがシステムにロードされた場合、少なくとも基本的レベルにおいて、グラフィック表示で自動的にサポートされてよい。言い換えれば、ユーザはデバイスに対するDDをシステムに提供し、次いで、DDからデバイスに対するグラフィック要素を生成するプログラムを実行させることによって、デバイスに対するDDからグラフィック要素を自動的に生成してよい。その後、このグラフィック要素は一つ以上のグラフィック表示で使用されることにより、デバイスに対してビジュアライゼーションをモデル化あるいは提供してよく、これは、デバイスに対するグラフィック要素を手動で生成する必要なしに、デバイスをモデル化あるいはグラフィック表示内でデバイスを図示する能力をユーザに提供する。

20

**【0089】**

上記のように、図3の多数のグラフィック要素74が、ひとたび生成されたら、グラフィカルエディタ50を使用して一つ以上のグラフィック表示（例えば、図3の表示76）を生成することができる。実際には所望なら、表示エディタ50は、多数の予め構成されたグラフィック要素（これらは、異なる業界、異なる機能用途のビジュアライゼーション等のそれらと関連各種のビジュアライゼーションを有してよい）と共に、ユーザまたは購入者に提供してよい。グラフィカルエディタ50は、次いで、ユーザが、カスタム要素を生成すること、並びに、グラフィック表示76を生成または構築することを可能にする。

30

**【0090】**

一般的に、表示76を生成するためには、ユーザはグラフィック要素74および他のビジュアル要素のライブラリから選択し、それらを配置して表示を構築する。グラフィック表示が完成状態になった後、生じたデータ構造またはオブジェクトは、単一の定義エンティティとして定義された各種の要素、固有プロパティおよびビジュアライゼーションの全てを有する表示クラスオブジェクトとして、データベース（例えば、構成データベース）に記憶されてよい。しかしながら、この表示クラスオブジェクトは、プロセス変数にはバインドされず、全ての実行時表示で必ずしも使用されるわけではない。所望なら、このクラスオブジェクトは、全てがXMLエンティティとして記憶され共に連結されたビジュアライゼーション、スクリプト等を有するxmlプロップとして、データベースに記憶されてよい。その後、個々のグラフィック表示をこのクラスオブジェクトから生成してよく、これらの個々のグラフィカル表示を指定しオペレータワークステーションまたは他の実行時環境にダウンロードしてもよい。

40

**【0091】**

50

表示がダウンロードされると、その中のグラフィック要素定義はAvalonコントロールに変換され、アセンブリにコンパイルされ、ターゲットマシンすなわち実行時マシン上に展開される。それ自体がAvalonエンティティである表示は、コンパイルされた制御アセンブリを参照することになり、さらに、制御または保守実行時アプリケーションのようなバックエンドデータソースにAvalonコントロールを接続するデータアダプタとして働くカスタムデータソースが生成されてよい。この方法により、グラフィック表示（および、所望なら個々のグラフィック要素）を一つの言語で生成及び編集し、別の言語または形態（例えば、XMLプロパティ）で記憶し、第3の言語または形態（例えば、各種の実行可能言語のうちの任意のもので書かれたスクリプトに結合されたAvalonコントロール）で実行させることができる。

10

#### 【0092】

ここで図15を参照すると、スクリーン220をグラフィカルエディタ50で作成して、ユーザが一つ以上のグラフィック表示を生成することを可能にできる。一般的に、グラフィック表示はプラント内の物理的デバイスを表現する相互接続された要素から構成されており、それらのデバイスと関連した追加の情報を含むかあるいは表示してよい。グラフィック表示内のグラフィック要素のそれぞれは、プロセス変数、定数、または他の外部値に等式化される一組の固有プロパティを含んでおり、上述のように、各グラフィック要素はいくつかのビジュアル表現（これは動的挙動、例えば色変化またはアニメーションを含んでいてよい）を有してよい。追加的に、グラフィック表示は、静的要素（例えば、テキスト、ボックス等）、ユーザがいくつかの方法でグラフィック表示と対話することを可能にするユーザダイナモ、ユーザにプロセスまたは他の情報を表示できる変数ボックス等を含んでいてよい。

20

#### 【0093】

図5のグラフィック要素に対する編集スクリーン112と同様に、図15の例としてのグラフィック表示エディタスクリーン220は、グラフィック表示が構築されているメイン編集キャンパス224、および、パレット部226（ここでは、テンプレートグラフィック要素、プリミティブまたは他のライブラリ要素が表示でき、これら各種の要素が選択してメイン編集キャンパス224へのドラッグおよびドロップができる）を含んでいる。メイン編集キャンパス224は、その中でユーザが対話してグラフィックス（プリミティブまたはコンポジットのいずれか）を生成および編集するダイアグラムビューを提供し、ベクトルグラフィックス編集およびビューイング機能を提供する。エディタ50は、ユーザが編集キャンパス224に現在表示されている表示フレームの内側または外側であってよい限界のないスペース内のどこかに、グラフィックスオブジェクト（例えばグラフィック要素）を配置することを可能にする。したがって、メイン編集キャンパス224は現在生成されている表示のほんの一部を表現することができ、ユーザはビューをパン（pan）して表示の他の部分を表示し、ズームインおよびズームアウトしてビューの倍率レベルを変え、ビューを回転させて任意の時点で表示に対して作業するのに最も便利な角度を得ることができる。しかしながら、ここで理解されるように、グラフィックオブジェクトの全ての描画および操作は、メイン編集キャンパス224において完成される。描画を補助するためビューのX-Y位置はルーラ227から読み取ることができ、倍率レベルはツールバー228上のドロップダウンリストに表示されてよく、ズームポップアップ上のエントリを選択すること等によって変更されてよい。

30

40

#### 【0094】

スクリーン220は、メイン編集キャンパス224内の要素の階層的な表示およびリストを表現する階層部230、キャンパス224に生成されている表示と関連する、あるいはキャンパス224内のハイライトされている要素と関連するプロパティまたはパラメータをリストするプロパティ部232、および異なるパラメータまたは機能をプロセス制御システムまたは任意の他の実行時環境内の要素にバインドする方法を示す、あるいはリストするバインディング部234をも含んでいる。追加的に、スクリーン220は、メイン編集キャンパス224に生成されているグラフィック表示と関連する各種のビューまたは層をリストするツールバー

50

236を含んでいる。より具体的には、いずれの特定の表示も、異なるコンテキストで異なる人々が使用することができる各種の層またはビュー（例えば、オペレータ表示、保守表示、シミュレーション表示、モデリング表示、エンジニアリング表示、ビジネス表示等）を有することができる。図15の例としてのスクリーン220では、ツールバー236は典型的な制御オペレータ用ビューを提供するOperationビュー（現在、メイン編集キャンパス102に示されている）、Eng/Maintenanceビュー（これは、典型的な保守またはエンジニア用ビューを提供する）およびTrainingビュー（これは、生成されている表示に表現されたプロセスプラントまたはプロセスプラントの一部をシミュレートする際に使用されるシミュレーションビューを提供する）として示されている3つのビューまたはレベルを含んでいる。図15のメイン編集キャンパス224内には選択された要素が無いため、プロパティ部232は生成されている表示の現在の名称を、表示の特徴（例えば、幅および高さ等）、作成者によって与えられた記述、名称、背景記述、および作成者が表示に対して格納しておきたい任意の他の情報と共に、表示する。同様に、図15のメイン編集キャンパス224には配置された実際の要素が無いため、階層部230は表示の名称のみを示しており、バインディング部234にはいかなるバインディング（バインド関係）も示されていない。

10

20

30

40

50

#### 【0095】

表示の生成中に、ユーザは要素（例えば、グラフィック要素）を、例えばパレット部226内で定義された要素の一組のカテゴリのうちの一つを介してその要素にアクセスすることによって、メイン編集キャンパス224に配置することができる。図15の例ではパレット部226は、メイン編集キャンパス224に配置され、共に接続されて完全な表示を生成できる、異なるカテゴリの要素を示し、このような表示はアクチュエータ要素、計算および制御要素、処理要素、プロパティおよび測定値、形状、ユーザインタフェース制御およびユーザ定義された要素を含む。もちろん、パレットビュー226において、任意の他の種類およびカテゴリ、またはサブカテゴリの予め定義された要素を提供またはそれらにアクセスしてもよい。この例では、アクチュエータはバルブおよび他のアクチュエータ要素を含んでいてよく、一方、計算および制御要素は任意の制御関連要素（例えば、コントローラの表示）、制御ループ（例えば、PID制御ループまたは他の種類の制御ループ）、機能ブロック、制御モジュール等を含んでいてよい。図15に示すように、処理要素は、タンク、反応器、攪拌器、または何らかの方法で材料を処理する他の要素、並びに任意の他の種類のデバイス、ユニット等を含んでいてよい。プロパティおよび測定値は、プロパティまたは測定値または実行時環境内の他のデータ（例えば、プロセス変数、アラーム等）を示すよう設計されたボックスまたは表示要素を含んでいてよい。形状はプリミティブまたは他の予め定義された形状であってよく、一方、UI制御はユーザが表示スクリーン上で操作して表示への入力を行なうことができるボタン、スライダ、ノブ、ツールボックス等のような各種のユーザインタフェース制御要素を含んでいてよい。もちろん、ユーザ定義された要素は任意の予め定義された要素（例えば、任意の他のグラフィック要素またはグラフィック要素から作られたより上位レベルの要素）を含んでいてよい。場合によっては、ユーザ定義された要素は、プロセスユニット、プロセスプラントエリア、または他のより上位レベルのプロセスエンティティを含んでいてよい。上術のように、ユーザ定義された要素へのアクセスはエディタ50を使用するユーザの個々人により限定されてよく、あるいはグローバルにアクセス可能であってよい。任意の他の要素、形状等を、任意の適当な見出しの下でパレット部226内に配置して、これら要素を組織化された状態かつユーザが容易にアクセス可能な状態に保持できることが、もちろん実現される。

#### 【0096】

ユーザが、要素（例えば、混合タンク）をメイン編集キャンパス224に配置すると、この要素はその要素に対するデフォルトビジュアライゼーションを用いてキャンパス224に描画されることができる。この場合、グラフィック階層部230は要素を示し、図5に表現されたのと同様の方法で、そのグラフィック要素と関連するビジュアライゼーション、アニメーション、プリミティブ等の、その要素に対するサブ要素の階層を提供する。更にまた、キャンパス224に、実行時環境にバインドできるビジュアライゼーションまたはアニメ

ーションを含んでいるグラフィック要素が示されている場合、バインディングサブスクリプション部234は、現在定義されているバインドを示す。

【0097】

所望なら、ユーザは任意の所望の方法で異なるビジュアライゼーションを選択することによって、メイン編集キャンバス224内またはパレット部226内のグラフィック要素のビジュアライゼーションを選択または変更できる。ユーザは、ドロップダウンリストまたはボックス（これは、例えば要素上でマウスポインタを右クリックすることによってアクセスできる）を用いて、または階層部230内の異なるビジュアライゼーションを選択することによって、または任意の他の所望の方法で、このビジュアライゼーション選択機能を行なってよい。図16はスクリーン220のメイン編集キャンバス224およびパレット部226の一部を示し、この図では垂直タンク要素240が、パレット部226からメイン編集キャンバス224内に配置されているとして示されている。垂直タンク要素240は、パレット部226に表現されたテンプレートまたはクラス垂直タンク要素241のコピーまたはインスタンス化である。しかしながら、マウスポインタがビジュアライゼーション240上に在る時、垂直タンク240に対する追加のビジュアライゼーション（マウス上での右クリックによってアクセスされるサイドバー242に示されているビジュアライゼーション1~8を含む、）が、表示が編集キャンバス224に生成されている時、垂直タンク要素240に対して使用すべきビジュアライゼーションとして、選択され得る。パレット部226に示すように、テンプレートタンク要素241の可能なビジュアライゼーションを取得もしくは見るために、パレット部226のテンプレート垂直タンク要素241上で右クリックすることによって、同様タイプのサイドバー表示243を得てもよい。サイドバー242の異なるビジュアライゼーションを選択すると、キャンバス224で使用するタンク要素240のビジュアライゼーションが変わり、一方、サイドバー243の異なるビジュアライゼーションを選択すると、パレット部226で記憶されたテンプレートタンク要素241に対するデフォルト設定またはビジュアライゼーションが変わる。

10

20

【0098】

パレット部226からグラフィック要素および他の要素を選択して、表示を生成することに加えて、ユーザは、図15に表現されたツールバー228から、アイテムを使用または選択して、基本的な図面ツール（例えば、ライン、正方形、長方形、円、五角形等のような形状、テキスト等）を取得してもよく、これらの簡単な図面ツールまたは要素を使用して、表示内でラインを描き、またはテキストを加えることができる。追加的に、ユーザはツールバー228に示されているツールバーコネクタ要素245を用いて、キャンバス224内で表示にコネクタ要素を加えてよい。コネクタ要素245は、選択された時、例えばドロップダウンメニュー、ダイアログボックス等を用いて、表示で使用するためのコネクタのリストをユーザに提供してよい。コネクタ要素は、1個のハードウェア要素（例えば、タンク、攪拌器、ポンプ等）を、別のハードウェア要素（例えば、バルブ、センサー等）に実際に接続するパイプ、コンベヤベルト、電気ライン、流体フローライン、または、他の種類のコネクタを含むことが可能である。このようなコネクタ要素、及び物理的デバイス（例えば、バルブタンク、ポンプ等）の異なる表現間にこのようなコネクタ要素を提供することは、米国特許出願第2004/0153804号公報に、より詳細に説明されており、これは本開示に参照により明示的に援用される。コネクタ要素は、ユーザが生成されている表示内の異なる要素同士を相互接続し、グラフィック要素について上述したように、異なる要素に対して定義されたコネクタ点とそれらを繋ぐことを可能にしてよい。例えば、パイプコネクタはパイプコネクタ点を有する異なる要素を取り付けるのに使用されてよく、一方ダクトコネクタはダクトコネクタ点が定義された要素を取り付けるのに使用されてよい。所望ならエディタ50は、ユーザが異なるグラフィック要素（例えば、バルブおよびタンク）を、バルブおよびタンクの接続点で定義された適切なタイプのコネクタで接続することだけを可能にする接続ルールを実装してよい。もちろん、接続要素の外観は、接続要素で生成されている表示によりよい様相および印象を提供するよう、接続タイプに基づいて異なっている。

30

40

50

## 【0099】

もちろん、ツールバー228は他のベクトル図面ツールおよび標準コマンドまたは機能（例えば、典型的なファイルオプション（新規、開く、保存、名前をつけて保存、閉じる、終了等）、編集オプション（元に戻す、繰り返す、カット、コピー、ペースト、削除等）、検索オプション等）を含んでいてよい。追加的に、ツールバー228はグラフィック関連コマンドを提供するメニューアイテム（例えば、プロパティ追加機能（これは、グラフィック要素の固有プロパティを追加するのに使用できる）、イベント追加機能（これは、グラフィカルイベントを追加するのに使用できる）、ビジュアライゼーション追加機能（これは、ユーザがビジュアライゼーションを追加することを可能にできる）、ビジュアルトリガ追加機能（これは、ダイアログボックスを開いてユーザが生成すべきトリガに関する情報を入力することを可能にできる）等）を含んでいてよい。更にまた、メニューは、書式機能（例えば、テキスト、ラインおよび2次元および3次元の画像のText、Line およびFillプロパティに関連する設定等）、および、図形機能（これは、オブジェクトのグループ化またはグループ解除、前景から背景へのオブジェクトの順序の選択、オブジェクトの垂直および水平両方の整列等を含んでいてよい）を提供してよい。ツールバーメニュー228は、ユーザがオブジェクトにアニメーションを適用することを可能にする機能も提供してよく、このようなアニメーションは、要素の幅、色、形状等を動かして、形状にゆがみ、回転または拡大縮小を与えること、要素を回転または反転すること等によって、要素のプリミティブまたはオブジェクトの一つ以上の要素に適用されてよい。もちろん、ユーザは、表示内で各要素と関連したアニメーションおよび動作にアクセスしてよく、適当なダイアログボックスまたは他のプログラミングツールを用いてこれらのアニメーションおよび動作を変更してよい。

10

20

## 【0100】

ここで理解されるように、ユーザまたは設計者は表示エディタ50を使用して、具体的には図15のスクリーン220とインタフェースして、測定値、アクチュエータおよびプロセス機器を動的に示す、標準の2次元またはさらに進んで3次元のグラフィック要素からなるグラフィック表示を、迅速かつ容易に生成することができる。また、制御および計算と関連した静的な要素およびユーザダイナモがサポートされており、表示内に提供されてよい。追加的に、ユーザは例えばわずかに異なる外観または周囲の情報を有する同じ要素を用いて、オペレータ、保守およびシミュレーション表示を生成することによって、グラフィック表示の異なるユーザのインタフェース要件に対応する層を定義することができる。

30

## 【0101】

図17は、グラフィカルエディタ50を用いて生成できる制御オペレータ表示の形態の、例としてのグラフィック表示の表示スクリーン300を示す。表示スクリーン300は、コネクタ要素316（この場合流体フローラインである）を介して、バルブ要素312及び一つ以上のポンプ要素314に接続された反応器要素310を含んでいる。流体フローまたはストリーム要素318は、スクリーン300によって表現されたプラントの部分に入り、それから出る材料の流れへの参照を提供する。理解されるように、表示スクリーン300内のグラフィックスは、各種のグラフィック要素を選択し、相互接続することによって生成されてよく、したがって表示スクリーン300は、個々のグラフィック要素に対して上述した方法で開発された高解像度グラフィックス、アニメーション、動作、ビジュアルトリガ等を含んでいてよい。もちろん、アニメーション、動作、ビジュアルトリガ、流体フロー要素等を含む追加のグラフィックスを、グラフィック表示レベルにおいて追加してよい。この方法で、アニメーションおよびユーザ動作を、グラフィック表示に組み込むことができる。例えば、この機能を用いてプロセス機器に対する静的なグラフィック構成要素を修正して、機器のステータス（例えば、モータがオンであるか、または始動されているかどうか、バルブの動作コンディションまたは位置等）を表示してよい。追加的に、アニメーションは、機器と関連する動的データを表現するのに使用されてよく、例えば充てんされているタンクの水位を（例えば、塗りつぶし技法を用いて）示す、または、運動を示す表示の変化（アニメーション）により攪拌機のステータス（状態）を示してよい。同様に、データ表示要素319を

40

50

スクリーン300内に配置して、表示300の外部で開発されたが表示300の構成要素の動作と関連しているプロセスデータまたは他のデータを示してもよい。

#### 【0102】

所望なら、グラフィック表示300はダイナモまたはユーザインタフェースボタン320Aおよび320Bも含んでいてよく、これらはユーザが表示スクリーン300との対話を介して、追加の情報を確認すること、または実行時環境に対して処置を取ることを可能にする方法で、ユーザが表示300とインタフェースすることを可能にする。場合によってはこれらの対話は上述したグラフィック要素に対するビジュアルトリガまたは動作で実施されてよい。例えば、ボタン320Aは、冷却塔構成についての情報の更なるビューをユーザに提供してよく、一方、ボタン320Bは反応器310に対してフェースプレート表示を提供してよい。したがって、表示300は、3次元の構成要素（従来のように、例えば、In-Tool、Auto-Cad、Visio（登録商標）のようなWindows（登録商標）メタファイル、ベクトルドローイング、JPEGおよびビットマップイメージフォーマットを含む多数の異なるソースからインポートされたグラフィックスを基にしてよい）を用いて、プロセスまたはプロセスの一部を構成する機器を示す一組の要素を含んでいるが、スクリーン300は今や、例えば、要素レベルにおいて回転、サイズ変更、拡大縮小、スキューイング、色変化等を行なって、より興味を持てるリアリスティックなアニメーション、即ちより理解しやすい表示を提供するためのアニメーションを含むことができる。

10

#### 【0103】

更にまた、基本的なユーザインタフェース構成要素（例えば、ノブ、ダイアル、スライドバーおよびボタン）を表示300に示してよく、制御システムまたは他の実行時環境内で情報または制御に動的にリンクしてもよい。データビュー要素またはダイナモもまた、制御機能における測定値、アラーム、イベント等と関連したキーパラメータを提供あるいはそれらを示してよい。例えば、ダイナモを制御ループパラメータに関連する情報を有する更なるスクリーンを表示するのに使用してよく、エンジニアリングユニットをダイナモの次に示して、ダイナモによって表示された値にコンテキストを提供してよい。制御ループと関連するプロセスアラームのステータスを、色変化、例えば制御パラメータ値の背景色の変化により、ダイナモに反映してよい。また、表示の乱雑さを無くするため、設計された通常の動作モードにはループがないという事実を、色変化を介して示してよい。もちろん、このようなダイナモはあらゆる標準に準拠するよう作成されてよい。

20

30

#### 【0104】

一例では、オペレータが、ユーザ対話可能な構成要素またはダイナモにアクセスした場合、スクリプトまたは他のプログラムが更なるスクリーンまたは表示（例えば図18にその例を示すようなフェースプレート表示または制御パネル表示）を引き出してもよい。例えば、ユーザインタフェースボタン（例えば、スクリーン300に示されているボタン320B）にアクセスした場合、反応器310に対するフェースプレートがオペレータに呈示されてよく、オペレータは次いでこのフェースプレートを使用して反応器310についての詳細を修正あるいは確認することができる。図18の例では、フェースプレート情報350は、ユーザがスクリーン300でボタン320Aを介してアクセスできる反応器310に対する制御ループ（FIG2\_28/TC2-1と命名）と関連している。ユーザは、フェースプレート350の一部分350Aを用いて、ボタン352（その動作は動作ルーチンによって定義される）によりモードを変える（カスケードから、例えば手動または自動に）、スライダ表示354上で動作パラメータの現在の値を確認する、矢印355を用いて制御ループと関連する設定点を変える等のことを行なうことができる。追加的に、ユーザは表示部350Bで制御ループの制限およびチューニングパラメータについての情報が与えられてよく、部分356でシミュレーション機能を可能にすることができる。所望なら、部分350Bは、部分350A上でボタン358のうちの一つを選択することによって、部分350Aのビューから得ることができる。同様に、ユーザは、制御ループについての更にまた別の情報（例えば、トレンドデータ、診断データ等）にアクセスでき、あるいは他のボタン358を介して制御および診断プログラム（例えば、ループチューニングプログラム）にアクセスしそれを実行させてよい。したがって、スクリーン

40

50



300上で行われたあるいは可能にされたユーザ動作に応じて、表示300を介して、任意の他の活動、スクリーン、および動作にアクセスすることができる。

【0105】

制御システムが、同様の機器の定義においてエイリアスの使用をサポートしている場合は、動的表示構成要素は表示スクリーン300で選択された機器に基づいて、動的参照をサポートするよう設計されてもよい。このような場合は、予め構成されたエイリアスおよび属性を、表示オブジェクトの一部として通常に定義されるオブジェクトタグまたはグラフィカル属性の代わりに使用できる。このエイリアス機能は高度な柔軟性および再利用性をサポートしているが、それは同様の表示オブジェクトが異なるI/O点に接続し、異なるグラフィカル属性、外観、およびセキュリティを表現できるためである。このような機能は、プラント内の異なる複製機器に対して同様の表示オブジェクトまたは同様の表示を再構築する必要性を無くすことができる。この方法では、同じグラフィック表示を、プラント内の構成および用法が同じで異なるハードウェアユニットを見るのに使用できる。

10

【0106】

もちろん、図17の表示スクリーン300は、スクリーン300の最上部に示されているツールバーに加えて、またはその代わりに、水平（表示の下）および垂直（表示の右）両方のツールバーを含むツールバーをサポートするよう設計されてもよい。所望なら、デフォルトツールバーを提供して、時刻および日付表示、アラームに肯定応答する、またはアラームを止めるのに必要なアラーム表示への直接のアクセスによるアラームリストの確認、アラーム要約表示またはメニュー、メインメニュー、または他の標準のメニューまたは表示、システムステータス表示等へのナビゲーションをサポートしてよい。

20

【0107】

図19は、より高度なグラフィック表示、すなわち、より多くの要素、接続、ユーザインタフェース動作、ダイナモ、および他のデータ参照を有するものと関連する表示スクリーン400を示す。具体的には、スクリーン400はエアおよび他の燃焼燃料製品がポンピングされ、あるいはポンプ412、バルブ404および関連コネクタ要素を介して、キルンバレルドライヤ408の入力406に送られる石灰キルンユニットの運転を示す。同様に、タンク409内のプロセス材料は、バレルドライヤ408を通して石灰を運ぶシートコンベヤデバイス410から供給される。もちろん、これら要素およびスクリーン400に示す他のデバイスのそれぞれは個別に生成されてよく、上述したスクリーン400内に配置できるグラフィック要素であってよい。バレルドライヤ408の入力406は、ドライヤ408が動作している時の火または炎415をアニメーションとして示すことにより、スクリーン400の観察者にキルンユニットの運転を明確に示すアニメーション化されたグラフィック要素を含んでいてよい。追加的に、ドライヤ408のさまざまな点における温度を、ダイナモまたは温度表示ボックス416で示してよく、また所望なら、これらのさまざまな点における温度を、炎アニメーション、ドライヤ408の色を介して、あるいは任意の他の方法で示してよい。同様に、他のプロセスパラメータ（例えば、フード圧力、キルン速度、トータルエア、メタノール、入力、1次および2次エア入力、オイルおよびガス入力等）が、プロセス制御システム内の特定の参照に結合されてそこから得られるボックス内の変数値と共に、図19に示すようなパラメータボックスで示されてよい。もちろん、スクリーン400では、石灰キルンユニットの他の物理的構成要素も示される。

30

40

【0108】

しかしながら、実行時環境内の他のデータソース（例えば、制御システムから、あるいは制御システム内のデバイスからのデータを処理する他のアプリケーション）から得ることができる、各種の他の情報を、スクリーン400上に提供してよい。このようなアプリケーションは、例えば、制御アプリケーション、保守アプリケーション、診断アプリケーション、ビジネスアプリケーション等を含むことができる。例えば、システムの温度（度単位）対時間を示すトレンド（動向）グラフ420が表示スクリーン400の上部中央に表示され、またデータヒストリアンまたはトレンドアプリケーションに結合され、このデータを自動的にプロットする。同様に、キルン412の運転の要約を提供するチャート422（これは、

50

例えばキルンの利益、エネルギー利用、生産速度等の各種のビジネス的側面を追跡するビジネスアプリケーションによって与えられてよい)が、スクリーン400の上部左側に表示される。更にまた、表示スクリーン400の上部部分のユーザインタフェースボタン424を選択することによって、ユーザにとって有用な場合がある他の情報(例えば、ヒストリカルトレンド、ヘルプ情報、キルン情報、制御制約条件、キルンエネルギーおよび他の情報)にアクセスしてよい。更にまた、アラームバナー430または他のバナーをスクリーン400の下部に提供してよく、また、アラミングアプリケーションに結合させてもよい。もちろん、これら全ての表示スクリーン400の要素および機能を提供し、上述した表示エディタ50の機能を用いて表示にプログラムしてもよい。追加的に、これらおよび他の機能は、任意の所望の方法によりグラフィック表示で組み合わせ、任意の所望のタイプの表示を生成してよい。更にまた、任意の所望のアニメーションおよびグラフィカル動作をスクリーン400で提供して、ユーザがスクリーン400等からより直接的に他の関連情報を見ることができるよう、ユーザにとってより有用または理解しやすいビジュアル情報を提供することができる。

10

20

30

40

50

#### 【0109】

したがって、図19に示すように、コントローラサブシステムによってアクセスされるプロセス要素、保守、制御、診断、チューニング、およびビジネスアプリケーションの形態のアプリケーション、データベース(例えばデータヒストリアン)からの履歴データまたはトレンドデータ、または、任意の他のデータソースを含む、各種の異なるデータソースからの情報を、表示スクリーン400に表示することができる。更にまた、このような表示へのユーザリンクを介して、表示スクリーン400から直接、他の情報または表示にアクセスしてよい。この方法で、スクリーン400によって示されているグラフィック表示はユーザに、より有用でありうる方法で、より多くの情報を提供する。

#### 【0110】

エディタ50は共通のグラフィック要素、接続要素等のセットを用いて任意のグラフィック表示を生成するのに使用できるため、表示エディタ50は同様の、または関連する表示を容易に生成するのに使用できる。このような表示は、同じプラントハードウェアのセットに関連し、それを示しながら、そのハードウェアに対する異なる目的の(例えば、制御オペレータ目的、シミュレーション目的、保守目的等の)異なる情報を示すことができる。この場合、プラントまたはプラントの一部と関連したハードウェア要素を示す共通の、または基本の表示を作成することができ、その基本の表示を用いながら、異なるユーザまたはユーザのタイプに対して異なる情報を提供し、異なる表示を生成することができる。

#### 【0111】

一例として、図20A~20Eは、石灰キルンユニットに対して生成された、オペレータのビュー、エンジニアのビュー、管理者のビュー、シミュレーションビューおよび保守ビューを含む、例としての表示を示し、これは同じ構成の同じ基本的なハードウェア要素を示し、同じビジュアライゼーションを用いてはいるが、異なる機能性目的に対して加えられた異なる情報を有している。このように、図20A~20Eの表示のそれぞれは同じ基本的なグラフィック要素を用いて同じ表示エディタで作られているため、同じ様相および印象を有しており、したがって呈示されている情報、および、それがプラント内のハードウェアにいかに関係しているかについての理解を保ちながらナビゲートするのが容易である。

#### 【0112】

具体的には、図20Aは、ポンプ504を介してキルンドライヤ508のコールドエンド(低温側)506に原材料を供給する石灰およびマッド供給源502(これは、ストリーム要素であってよい)を有するキルンユニットのオペレータ用ビュー500を示す。コンベヤ510はキルンドライヤ508のホットエンド(高温側)512から処理された石灰を除去し、石灰をタンク514に堆積させる。製品ラインまたはストリーム要素516は、タンク514を出る製品(石灰)の量を示す。同様に、燃料源ストリームインジケータ520からの燃料は、バルブ522(その色はバルブの動作をグラフィカルに示してよい)を介して、キルンドライヤ508のホットエンド512に供給される。ファン524は空気が燃料と混合されるキルンドライヤ508の

ホットエンド512に、ダクトコネクタ要素526を介して空気をポンプ送りする。アニメーション（例えば、火または炎528等）をキルンドライヤ508のホットエンドに示して、キルンドライヤ508の運転状態を示してもよい。同様に、誘引ファン530は、追加の配管532を介してキルンドライヤ508のコールドエンド506から空気を引き込み、この空気をスタックストリーム要素534によって示されているスタックに送る。更にまた、各種のプロセスパラメータ（例えば、システムの各種の部分における温度、キルンドライヤ508のバレルの速度、空気および燃料流量等）の値を示す、各種の変数またはパラメータボックスが示されている。ここで分かるように、このハードウェアおよびこれらの変数ボックスは、図20A~20Eのスクリーン全体を通して共通であり、これら表示に同じ様相および印象を持たせる表示の基本的な要素を提供する。

10

#### 【0113】

しかしながら、図20A~20Eのスクリーンのそれぞれは、プラント内で異なる機能に合わせて作られた追加の情報、ユーザインタフェースボタンおよび動作を含んでいる。例えば、図20Aのスクリーン500は、オペレータ用ビューであり、ボックス540に、ラボテストによって測定され、評価され、与えられた残留炭酸塩を、ボックス542にキルンの総エネルギーおよび特定のエネルギーを、並びに、トレンドグラフ544に最適動作の達成に重要なモデル予測制御ルーチンによって与えられる温度の過去および未来のトレンドを示す。追加的に、スクリーン500はユーザが関連したアイテムについての追加の情報（例えば、制御ループ情報、要素の更なる運転情報等）を見ることを可能にする、各種のユーザインタフェースボタンまたはダイナモ548を提供する。

20

#### 【0114】

図20Bのスクリーン550はエンジニアリングビューを示し、これは図20Aのそれと非常に似ているが、点線552を介して制御にとって重要な測定位置および信号経路を示すことにより、基本的な制御システムおよびモデル予測制御システムがいかに協働して運転を最適化するかを示している。図20Cのスクリーン560は管理ビューを提供し、キルンユニットの基本的な運転を示すが、システムの個々の構成要素または制御ループについて追加の情報を得ることを可能にするユーザインタフェースボタンは提供しない。更にまた、管理ビュー560は、過去の日および月にわたるラボ測定値による総エネルギー、特定のエネルギーおよび残留炭酸塩（変換されていない供給材）を示すチャート562を含んでいる。この情報は、例えばビジネスコンピュータで実行されているビジネスアプリケーションから与えられてよい。

30

#### 【0115】

更にまた、図20Dは、シミュレータがシミュレートされたシステム内のパラメータを変え、シミュレートされた結果を見ることを可能にするシミュレーションビュー570を提供する。ビュー570は、例えばオペレータを訓練すること、さまざまな未来の動作モードをテストすること等に使用されてよい。図20Dに示すように、シミュレータは、スクリーン570上の一つ以上のユーザインタフェースボタンを介してシミュレーションオペレータに与えられる一つ以上のダイアログボックス570を介して、パラメータを変更してよい。プロセスシミュレーションビュー570は、トレーニング用にはオフライン状況で、あるいは、未来の問題を検出する際に有用であり得る追加的情報を提供するためにはオンライン状況で、使用できる。プロセスシミュレーションは、機器のタイプおよびそれらの接続が、オペレータグラフィックスの構築においてシミュレーション機能を有するスマートオブジェクトまたはグラフィック要素を用いた結果として既知であるため、これらのグラフィックスから生成することができる。これらのスマートオブジェクトは、米国特許出願第2004/0153804号公報に詳細に説明されている。

40

#### 【0116】

同様に、図20Eは、キルンユニット内の機器の状態についての、またはその詳細の情報を提供する保守ビュー580を示す。ビュー580では、機器の故障は、機器の健全性の表示582および584を用いて示されてよい。この場合、表示582および584（図20Eではこれら全てを示していない）は半円であり、塗りつぶし色が機器の現在のモニターされた健全性を示

50

すか、あるいは指定している。ビュー580では、表示584は健全性が最適な状態に達していない機器を示している。もちろん、これらの表示582および584は保守または診断アプリケーションから与えられてよい。更にまた、ここで理解されるように、プロセス問題が検出された場合、保守技能者はスクリーン580を使用して詳細を掘り下げ、問題を解決するための推奨された手続きを見つけることができる。一例として、目詰まりしたマッドフィルタは洗浄が必要となり、そのため供給材を遮断すること、およびキルンをアイドルモードにすることが必要となり、これは、図20Aのオペレータのビュー500から行うことができる。作業者は、上流のプロセス問題によって生ずる場合がある過度の塵について、供給材の品質を検査する必要も感ずるであろう。このように、この簡単な例で示すように、異なる機能間の各種のインタラクション（例えば、問題を検出し、それを解決すること）が、同じ様相および印象を有し、したがってナビゲートするのが容易な異なるビュー間を切り換えることによって、簡単かつ容易に行なえる。もちろん、図20A~20Eの各種のスクリーン内の情報は、表示上で層状にしてよく、また所望ならスクリーンのユーザに基づいてのみ示してよい。また、各要素またはスマートオブジェクトに対して示される情報は、ユーザまたはユーザ個人により変化してよい。

10

20

30

40

50

#### 【0117】

更にまた、図20A~20Eの表示に対して追加の表示を提供し、それらと関連付けられてよい。これら表示は、制御構成表示（例えば、図21Aおよび21Bに示す）を含んでいてよい。図21Aの制御表示585は、図20A~20Eに表現されたキルンユニットの各種のハードウェア要素に、及びそれらからコントローラに対して各種の制御信号が送られる方法、並びに、これら制御要素と関連するタグまたは変数名称を示す。図21Bの制御表示590は、図20A~20Eのキルンユニットの制御を実行する制御ルーチンを示す制御モジュールを示す。そのようなものとして具体的には示していないが、図21Aの制御表示585は、（これら要素と関連した同じまたは異なるビジュアライゼーションを有する）スクリーン20A~20Eの同じ要素を用いて作られてよく、あるいは異なる要素を用いてもよい。同様に、図21Aおよび21Bの表示585および590は、表示エディタ50を用いて作成してよい。これら表示には上述したグラフィックおよびアニメーション機能のうちの任意のものが提供されてよい。更にまた、制御表示585および590には、オペレータが例えばそれらの上に与えられたユーザインタフェースボタン548のうちの一つにより、図20Aのオペレータ用ビュー500を介してアクセスして、オペレータ用ビュー500から現在の制御セットアップに容易にアクセスすることを可能にしてよい。

#### 【0118】

したがって、上に与えられた議論から分かるように、各種の関連表示をエディタ50で生成でき、またこれら表示は多数の方法で層状にすることができる。具体的には、表示を上述したように層にして、関連するまたは同様のオペレータ、ビジネス、シミュレーション、保守およびエンジニアリングビューを提供してよく、またこれらビューは互いに他から容易にアクセス可能であってよい。更にまた、さまざまな表示を、プラントの特定の階層的、論理的または物理的構造を反映する方法で層にし、あるいは共に接続してよい。したがって、例えばプラントのさまざまな物理的または地理的エリアに対して表示が作られてよい。したがって、プラントの主要なエリアを示す単一の表示を生成してよく、また、プラントの各エリア内で例えばユニット毎ベースで基本的な構造を示す追加の表示を生成してよく、一方、更に別の表示を各ユニットに対して生成してもよい。この方法では、ユーザは表示を詳細に展開して、プラントのますます小さな部分と関連するますます詳しい詳細を得ることができる。別の例として、プラントの異なる区分または部分に対する表示は、ユーザインタフェースボタンを介して共に連結され、または付加されてよく、したがってオペレータは異なる表示を、前後方向に容易にスクロールして、論理的な方法で、プラントまたはプラントの構成要素の異なっているが物理的に接続された区分を見ることができる。

#### 【0119】

これらタイプの表示層化は、図22の図でより詳細に示されている。具体的には、一般的

または全体的なプラントオーバービュー表示600は、プラント全体の基本的または上位レベルの構造を示す表示を提供できるが、それはプラントの任意の特定の区分または部分について、多くの詳細を提供しない場合がある。ユーザは、表示600から、プラントの任意の特定の区分またはエリアを（インタフェースボタンを用いて）選択し、あるいはそのまま詳細に展開して、プラントのエリアAの一つ以上の表示602またはエリアBの一つ以上の表示604を引き出すことができよう。図22に示すように、エリアAはそれと関連したn個の別々の表示を有してよく、それらの全てがプラントのそのエリアを通るプロセスフロー、またはプラントのエリアAと関連した何らかの他の論理的構造を反映するよう、互いに他と直列に論理的に接続されている。602a、602b、...602nで示されたエリアAのnの表示は、ユーザが、一つの表示から次の表示にスクロールすることができるページフォワードまたはページバックタイプの動作を用いて、アクセスされてよい。この方法では、ユーザはオペレータにとって意味をなす方法で、エリアAの一つの区分からエリアAの別の区分に、表示を容易にスクロールすることができる。更にまた、図21に示すように、オペレータは表示602a～602nのうちの別々のものから、予め定義された情報または他の表示情報を得ることができる。したがって、オペレータはエリアAの表示602aを見る場合、表示602a内のキーパラメータの、予め定義されたトレンドを表現する更なる表示610を得ることができる。同様に、表示602bを見る場合、オペレータは、スタートアップおよびシャットダウン手続き情報を有するリストまたはドキュメント612にアクセスすることができる。もちろん、ユーザは、表示602a、602b等の間を、フォワード（前）およびリバース（後ろ）方向にスクロールすることができる。

10

20

**【0120】**

同様に、エリアB（これは、m個の表示604a～604mを含んでいるとして示されている）を見る場合、ユーザは、表示内のフォワードおよびリバース（ネクストおよびバック）ボタンを用いて、同じ詳細のレベルにおける表示間をスクロールしてよく、したがって、オペレータまたは他のユーザは、プロセスの表示された区分の上流または下流の情報を含む表示にアクセスすることができる。また、ダイナモまたは他のユーザインタフェースボタンを提供して、別の表示にアクセスし、現在の表示内の要素と関連した他の情報を得ることを可能にしてよい。これらツールの使用により、表示階層（オーバービュー表示を用いて、それから、各プロセスエリア内のキー表示にアクセスしてよい）が、生成できる。

**【0121】**

加えて、図20A～20Eに示すように表示を層状にしてよく、したがって多数の表示が、プラントの同じ区分または部分に対して存在するものの、それらは異なる機能的目的、例えばオペレータ動作、保守動作、シミュレーション動作、ビジネス動作、エンジニアリング動作等に使用される。これら別々の機能的表示は、図22で表示602a、602b等の下に層化されているものとして示してあり、また所望なら、互いに他からアクセスされてよい。したがって、オペレータまたはユーザが表示602aを見ている時、ユーザは例えば保守ビュー、シミュレーションビュー、ビジネスビュー等、プラントのその区分に対する他の機能的表示間で切り換え、またはそれらに（ただしそのようなアクセスが許される場合に）アクセスすることができる。もちろん、同じ基本的な表示要素を用いた、この方法での保守ビュー、オペレータビュー、シミュレーションビュー等の層化も、これらビュー間の容易な切り換えおよびプラント内の異なる機能についてプラント内で起こりつつあることのよりよい理解を提供する。

30

40

**【0122】**

この機能性及びその他の機能性を実装するためには、グラフィック表示は、そのグラフィック表示に対して意図された役割（または機能）の表示を含むよう生成されてよい。このような役割は、例えばフェイスプレート表示、詳細表示、1次制御表示、概念表示、保守表示、ビジネス表示、シミュレーション表示、または任意のその他のユーザ定義された役割を含むことができる。役割はグラフィック表示の一部であり、グラフィック表示が制御モジュールまたはハードウェアデバイスに指定されている場合、実行時中のその表示の用途およびそれへのアクセスを定義するのに使用されてよい。グラフィック表示に指定さ

50

れた役割の表示は、特定の表示をどこに、また構成システムまたはプラントのどの区分に、指定できるかを示すのに使用できる。追加的に、この役割情報は、ある特定のユーザが、該ユーザによって行なわれているジョブに基づいて、どの表示にアクセスできるかを判定するのに使用されてよい。例えば、制御オペレータは、シミュレーションまたはビジネス表示として定義されたグラフィック表示を見たり、またはそれにアクセスすることはできない場合がある。

#### 【0123】

上述したグラフィック表示の機能は、特殊な表示を生成して重要な機器のステータスを容易にモニターすることを可能にするのに使用してもよい。これらタイプの表示のいくつかの例は、プロセスシャットダウン、振動モニタリング、バーナー管理、煤煙換気扇運転、および安全システムステータスに関する一次出力 (first out) 表示を含んでいる。もちろん関連表示は、情報を要約するよう構成することができ、また動作機器がアニメーション、例えば煤煙換気扇を含む場合は、アニメーションを効果的に使用して、オペレータが迅速にシステムの運転にアクセスまたはそれを理解することを可能にしてよい。追加的に、大部分の制御システムの計算機能は、運転コスト、能率等のオンライン計算を実行するのに使用してよく、またこのタイプの情報はオペレータのグラフィック表示に容易に組み込まれてよく、したがってオペレータはこの情報を使用してプロセス動作を改善することができる。また、各種の技法を用いて、制御システム内のサブシステム情報をまとめてよく、標準グラフィックスおよびダイナモを用いてオペレータ表示を生成し、より上位レベルの表示からサブシステム情報へのアクセスを可能にしてよい。場合によっては、マトリックス値の3D描画を行なって情報 (例えば、シートゲージング情報) を示してよい。

10

20

#### 【0124】

加えて、グラフィック要素および表示は、例えばMicrosoft (登録商標) Avalonコントロールによって提供されるようなベクトルグラフィックスを用いて、柔軟性および速度を提供し、有利に実装してよいことに気づくであろう。ベクトルグラフィックスの使用は、拡大縮小可能なグラフィックスを用いて、生成されている表示の拡大縮小およびサイズ変更を行ない、特定の表示マシンに適合させ、並びに拡大縮小を使用するアニメーションを提供可能にする能力を提供する。

#### 【0125】

一般的に、また題する同時係属出願でより詳細に説明されているように、各プロセスグラフィック表示の生成、およびその中に含まれているグラフィック表示要素は、第1の宣言フォーマットに記載された、それぞれのテキスト形式の記述に記録される。各記述内のスクリプトコマンドは、描画すべきグラフィックスの複雑さとは無関係に、表示を定義するための能率的な非メモリ集約的メカニズムを提供する。宣言フォーマットおよび、即ちスクリプトコマンドは、多数の異なるマークアップ言語のうちの任意のものに基づいてよい。より詳しくは、XMLに基づくマークアップ言語に依存して各表示および表示要素に対するレンダリング (描画) 定義 (XMLプロップとも呼ばれる) を記載してよく、またこのXMLプロップは、これら表示および要素をプロセス実行時環境にダウンロードする前に構成ライブラリまたはデータベースに記憶するのに使用することができる。高度なグラフィックス (例えば、アニメーション) をサポートするために、マークアップ言語はベクトルグラフィックススキームに従ってこのようなグラフィックスを定義してもよい。

30

40

#### 【0126】

上記のように、また "Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface" (プロセスプラントユーザインタフェースにおけるマークアップ言語に基づく動的プロセスグラフィックス) と題する同時係属出願 (これは、参照により援用されている) でより詳しく説明したように、プロセスグラフィックスの動的特徴は、プロセスプラントパラメータの現在の値を、オンラインまたはシミュレーション条件が変化するにつれて反映するよう設計されている。この目的のため、プロセスグラフィックスは、このような変化を反映するデータに対するソースにリンクしてよい。各XMLベースの記述はしたがって、一般にデータに従って修正されるべきそれぞれの動的グラフィック

50

パラメータ（例えば、タンク内部の色変化）についてデータソース場所を識別する、一つ以上のデータソース参照を含んでよい。データソース場所は、エディタを介した構成において後に指定するため、スクリプトがエイリアスまたはプレースホルダー（代替物）を識別して、後で指定されるべきデータソースまたは経路情報を参照するように、オープン（公開）にしておいてもよい。プロセスグラフィック表示のデータソース情報および他の特性（例えば、イベントハンドリング等の挙動）がXMLベースの記述を介して指定されるため、このXMLベースの言語はPGXMLまたはプロセスグラフィックスXMLと呼ばれてもよい。

#### 【0127】

プロセスグラフィック表示およびその構成要素を定義することを目指した構成および設計作業が完了次第、構成エンジニアまたはその他のユーザは、オペレータワークステーションまたは他のユーザ表示デバイスにプロセスグラフィックスをダウンロードするための準備として、PGXML記述を処理することを選択することができる。一般的に、グラフィック表示および表示要素の各PGXML記述は、(i) 使用すべきグラフィックス描画エンジンと互換性のあるベクトルグラフィックスフォーマットでのスクリプト、および(ii) 表示のデータソース参照および任意の他の非グラフィックス機能性（例えば挙動）を指定する命令を有するコードを生成するよう処理される。スクリプトのベクトルグラフィックスフォーマットは、宣言的、またはXMLベースの言語であってもよい。Microsoft（登録商標） Avalonユーザインタフェースアーキテクチャを利用する実施形態では、ベクトルグラフィックススクリプトは、Microsoft XAMLで記載されてよい。他の実施形態では、オープンソースフォーマット、SVG（Scalable Vector Graphics）を利用してよい。コードは、C#または任意の他の適切なプログラミング言語で記載してよい。

10

20

#### 【0128】

いくつかの実施形態では、ベクトルグラフィックススクリプトおよび関連コードは、次いで組み合わされ、オペレータワークステーションまたは他のユーザ表示デバイスに対する実行可能コマンドが記載されたファイルにコンパイルされる。この目的のため、各プロセスグラフィック表示およびグラフィック表示要素に対してそれぞれ動的リンクライブラリ（DLL）ファイルを生成してよい。いずれにせよ、ベクトルグラフィックススクリプトおよび関連コードのこのようなコンパイルは、ダウンロードの前に行なうことによりネットワークデータの送信条件を最小限化できる。

30

#### 【0129】

グラフィック要素およびグラフィック表示は、ひとたび生成されたら、ジェネリックまたはテンプレートオブジェクトとして構成データベースに記憶させてよく、クラスベースまたは非クラスベースのオブジェクトまたは要素として記憶させてから、実行時環境で使用できる。一般的に、ここで論じるグラフィック要素およびグラフィック表示は、米国特許出願第2004/0199925号公報（これは、参照によりここに援用されている）で説明したように、表示モジュールの形態のモジュールであってもよい。クラスモジュール（オブジェクト）は、プロセスプラントまたはプロセス制御システム内のいずれの特定のハードウェアまたはデバイスにもバインドまたは結合されていないものであるが、プロセスプラントまたはプロセス制御システムにバインドされる他のオブジェクトをそれからインスタンス化できるオブジェクトである。一般的に、プロセスプラントおよび具体的にはプロセスプラントの実行時環境を構成するためには、構成エンジン（例えば、図1の構成ワークステーション20に示した制御構成アプリケーション33のうちの一つ）を使用して、グラフィックオブジェクト（グラフィック要素およびグラフィック表示を含む）を、制御モジュール、機器モジュール（例えばユニットモジュール）、プロセスモジュール等のような論理的エンティティ、または物理的エンティティ（例えば、オペレータワークステーションまたは他の表示デバイス）を含む、プロセスプラント内の他の論理的および物理的エンティティと関連付けてよい。場合によってはグラフィックオブジェクトはクラスオブジェクトであってもよく、機器モジュールクラスオブジェクトまたは制御モジュールクラスオブジェクトのような他のクラスオブジェクトと関連付けられてよく、またグラフィックオブジェクト

40

50

は個々のオブジェクトであってもよく、インスタンス化されたオブジェクト（例えば、プロセスプラント内の機器にバインドされ、プロセスプラント内の例えばコントローラ、ワークステーションまたは他の機器にダウンロードされたオブジェクト）と関連付けられてもよい。

#### 【0130】

したがって、上に識別したグラフィック要素およびグラフィック表示は、米国特許出願第2004/0199925号公報で説明したように、他のクラスオブジェクトがライブラリ内で記憶され、プロセス制御システム内で構成され、次いでその各種の部分またはサブ部分にダウンロードされる方法と同様の方法で、プロセスプラント構成システム内で構成され、プロセスプラント実行時環境内のハードウェアにダウンロードされてよい。具体的には、構成エンジン（これは、例えば図1のデータベース28のような構成データベース内のデータに対して動作する）は、構成データベース階層（例えば、図23および24に示す）を使用して、ユーザがプロセスプラントの構成を見、変更することを可能にする。図23の構成階層スクリーン700は、制御システム702（DeltaV（登録商標）制御システムとして示されている）に対する構成データベースが、ライブラリ（Library）部704、システム構成（System Configuration）部706および検索結果（Search Results）部708を含んでいることを示す。

10

#### 【0131】

ライブラリ部704は、実行時環境に指定またはダウンロードされていないが、代わりにテンプレートまたはジェネリックおよびアンバインドされたオブジェクトとして記憶されているクラスオブジェクトおよび他のオブジェクト定義を含んでいる。図23に示すように、ライブラリ部704は、制御モジュール（Control Modules）部710、デバイス（Devices）部712、プロセスモジュール（Process Modules）部714およびオペレータインタフェース（Operator Interface）部716、並びにバッチ（Batch）部および安全計装システム（Safety Instrument Systems）部（番号で示していない）を含んでいる。

20

#### 【0132】

制御モジュール部710は展開されていないが、構成データベースのこの部分は一般に、プロセスプラント内の各種のコントローラおよび他の制御デバイスで使用すべき、プロセスプラントに対して生成されたさまざまな制御モジュール（例えば、制御モジュールテンプレートおよび制御モジュールクラスオブジェクト）を含んでいる。更にまた、デバイス部712は、プロセスプラント内で使用される実際のデバイスおよびデバイスのタイプ（例えば、バルブ、センサー、コントローラ、ワークステーション等）の表示を含みうるデバイスオブジェクトの表示、並びに、プロセスプラント内の異なるデバイスまたはハードウェアと関連する論理的要素を定義するデバイスクラスオブジェクト、機器クラスオブジェクト等の表示を含んでいる。更にまた、プロセスモジュール部714は、プロセスモジュールテンプレートおよびこれらプロセスモジュールに対して定義されたプロセスブロック定義を含む、プロセスプラントに対して生成された各種のプロセスモジュールを含んでいる。このようなプロセスモジュールは、米国特許出願第2004/0199925号公報で詳細に説明されており、したがってここでは更に説明しない。

30

#### 【0133】

しかしながら、図23に示すように、構成階層700のオペレータインタフェース部716は、上述したグラフィックオブジェクトと関連する情報を記憶している。具体的には、グラフィックス（Graphics）部720は、各種のコンポジットプロセスグラフィック定義、グラフィッククラスオブジェクトおよびグラフィックテンプレートを含んでおり、これらはシステムに対して生成された各種のグラフィック要素およびグラフィック表示を定義する。コンポジットプロセスグラフィック定義（Composite Process Graphic Definitions）部は、一般にシステムに対して生成されたグラフィック要素を含んでおり、プロセスグラフィッククラス（Process Graphic Classes）部は、上述した方法でクラスオブジェクトとして生成されたグラフィック表示を含んでおり、一方、プロセスグラフィックテンプレート（Process Graphic Templates）は、非クラスベースのグラフィック表示および他のテンプレートオブジェクトを含んでいてよいことに気づくであろう。構成データベースのこの

40

50



部分に記憶されているグラフィックオブジェクトは、一般に他のオブジェクトを生成するのに使用でき、構成プロセス中にグラフィック要素およびグラフィック表示がプロセスプラント内の各種のハードウェアおよびソフトウェアにバインドされるべき方法を定義するのに使用できる、バインドされていないオブジェクトである。更にまた、オペレータインタフェース部716の下レイアウト(Layout)部722はオペレータインタフェースに対して各種のレイアウトを定義し、一方、役割(Roles)部724はオペレータインタフェース上で実行できる各種の役割、並びにグラフィック表示および要素のうちの各種のものによって実施される役割を定義する。

#### 【0134】

したがって、構成階層700で説明された構成データベースは、ジェネリックテンプレートとして、クラスオブジェクトとして、または、特定のプロセスエンティティにはバインドされていないが特定のプロセスエンティティにバインドされることになる要素または表示を定義するのに使用できる、特定の、または個々の要素または表示として、グラフィック要素およびグラフィック表示を含むグラフィックオブジェクトを記憶する部分を含んでよい。一般的に、構成エンジニアまたは他のユーザは階層700を使用して、以下により詳細に論じる方法で、グラフィック表示機能を含むようプロセスプラントを構成してよい。

#### 【0135】

図24は、システム構成部706が展開されている階層スクリーン730を示しており、これはプロセスグラフィック要素およびグラフィック表示を、システム構成の一部として、プロセスプラントの各種の論理的および物理的エンティティといかに関連付けられるかを示すためのものである。具体的には、システム構成部706の下のプロセスグラフィックス(Process Graphics)部734は、一般に実行時環境で使用のためプロセスプラントにダウンロードされるよう定義された各種のグラフィック表示を定義する。一般的に、プロセスグラフィックス部734はシステム構成全体に適用されるので、プロセスグラフィックス部734内のグラフィック表示および要素(例えばフェースプレート表示、オペレータ表示、詳細表示、保守表示、シミュレーション表示等を含んでよい)は、プロセスプラント内の各表示ユニット(例えば、各ワークステーションまたは他の表示デバイス)にダウンロードされる。しかしながら、グラフィック要素および表示は、代わりに制御計画(Control Strategies)部736および物理ネットワーク(Physical Network)部740を含む、システム構成の特定の部分と関連付けてもよい、これは表示デバイスをこれらグラフィック要素および表示が利用できるものに限定しうる。

#### 【0136】

一般的に、制御計画部736はプロセスプラントのさまざまな物理的および論理的区分に指定された各種の制御ルーチン、あるいはそれらに対して実行される計画を定義する。グラフィック表示(これもまた、例えばフェースプレート表示、制御表示、保守表示等の任意のタイプの表示であってよい)のうちの各種のものを、例えば、エリア(例えば、Area\_A 742およびArea\_B 744と称するエリア)、エリアと関連した制御モジュール(例えば、制御モジュール部746)、およびエリアに指定されたプロセスモジュール(例えば、プロセスモジュール748に)等の、制御計画部736の特定のサブ区分またはサブカテゴリに指定(割り当て)してよい。したがって、図24に示すように表示(Display)部750は、Area\_A部742(これは、プラントのArea\_Aに対して及びその中で使用される全ての表示を定義する)と関連している。展開されてはいないが、フォルダー750の下に表示は、Area\_A内のさまざまなハードウェアのセットと関連する各種の表示(例えば図22の異なる表示602a~602n)、並びに、これら表示からアクセス可能な各種のサブ表示(例えば図22の表示610および612)を含むことができる。表示に対して定義された役割は、これら表示が互いに他から、または表示デバイスを使用するオペレータにとって、アクセス可能な方法を示してよい。したがって図22について説明したように、ユーザは部分750の表示をスクロールして、プラントのArea\_Aのさまざまな部分を見てよく、あるいは、より上位レベルの表示の特定の部分のより詳細な表示(該より上位レベルの表示では同程度に詳細には示されな

10

20

30

40

50

い)を含む、任意の特定のエリアについてのより多くの詳細を見ることができる。

【0137】

更にまた、表示を、制御モジュール部746の下のDisplaysアイコン752で示されているArea\_A部742内の特定の制御モジュールに指定することができる。この場合、表示(Displays)752は、制御モジュールループ(LOOPと命名)と関連しており、具体的には、その特定の制御ループと関連する制御オペレータ用表示であってよい。繰り返すが、表示に対して定義された役割は制御表示の一部としてオペレータステーション内でその役割を示すことができる。更にまた、Area\_B部744について説明したように、表示(Displays)754は、一般にエリアと関連付けられてよく、一方、表示(Displays)756は、エリアに指定された特定のプロセスモジュールと関連付けられてよい。この場合、Displayアイコン756は、PMOD2(これは、Area\_B内の機器と関連するプロセスモジュールである)と称するプロセスモジュールと関連している。PMOD2は、例えばユニットモジュールまたは機器モジュールまたは任意の他のタイプのプロセスモジュールであってよい。

10

【0138】

したがって、一般に制御計画部736の下に示されているように、表示は、それらの制御定義が構成システム内で生成されるにつれて、特定の制御計画または制御定義と関連付けられてよい。図24の例に示すように、表示は、(例えば、エリア部742および744の場合)エリア毎ベース、並びに(例えば、制御モジュール750の場合)、制御モジュール毎ベースで、また表示部756で示されているプロセスモジュール毎ベースで、定義された制御定義と関連付けられてよい。追加的に、所望なら、機器アイテムは役割と関連付けられてよく、それら役割と関連した表示に機器要素から動的にアクセスすることを可能にする。このような機器の役割および表示は、プロセスグラフィックス部734に記憶させてよい。

20

【0139】

追加的に、表示を、オペレータワークステーションまたは他の表示デバイスを含む、プロセスプラント内の実際のハードウェア要素に割り当て、あるいはそれと関連付けてもよい。したがって、物理ネットワーク部740の下に示したように、特定の表示を特定のオペレータワークステーションまたは表示を有する他の実行時マシンに割り当ててもよい。図24の具体的な例では、物理ネットワーク部740は関連するハードウェア(例えば、CTRL1と称するコントローラ、構成ステーションであってよいProPlusステーション、およびオペレータステーション(OperatorStation)762と称するオペレータワークステーションを有する制御ネットワーク(Control Network)760を含んでいる。オペレータステーション762(これは、プラント内の特定のワークステーションまたはユーザインタフェースを定義またはそれと関連している)は、アラームおよびイベント機能、オペレータ活動、継続的ヒストリアン機能およびそれと関連したシミュレーション活動を含んでいるが、他の活動または機能(例えば、保守活動、ビジネス活動等)を任意の特定のユーザインタフェースと関連付け、その上で行なうことができる。この方法では、特定のユーザインタフェースがプロセスプラント内の役割に割り当てられ、表示デバイスに記憶あるいはそれにダウンロードされたそれらの役割に対する表示を有してよい。

30

【0140】

図24に示すように、ディスプレイ764および766は、オペレータ機能または活動に、並びに、ワークステーション762によって行なわれるシミュレーション機能または活動に指定(割り当て)される。図24には示していないが、表示は、他の表示デバイス(例えば、プロセスプラント内の他のノードと関連したもの)に、これら表示デバイス上で実行されるために割り当てられてよい。更にまた、オペレータワークステーションまたは他の実行時表示マシン内の特定の機能への表示の割り当ては、これら表示が実行時環境内で行なうことができる役割または機能(例えば、オペレータ機能、シミュレーション機能、保守機能等)を定めてよい。もちろん、これら各種の機能は構成システムによって設定された方法で、同じ表示デバイス上あるいは異なる表示デバイス上で実行されてよい。また、構成階層730はオペレータワークステーションに指定されている表示を示しているが、表示はPDAのようなハンドヘルドコンピュータ表示デバイス、電話機、ビジネスワークステー

40

50

ションの表示デバイス、または任意の他の所望のタイプの表示デバイスを含む、他の種類の表示デバイスに指定されることができる。

【0141】

構成エンジンは、例えば構成エンジニアのようなユーザが、構成階層を使用して、プロセスグラフィック表示がプロセスプラントにバインドされ、その中でダウンロードされる方法を、定義および管理することを可能にする。一般的に、例えば構成エンジニアのようなユーザは、ユーザが他のオブジェクトに対して行うことができるのと同じ方法で、構成システム内で表示オブジェクトを生成、削除、名称変更、指定およびダウンロードすることができる。より詳しくは、構成エンジンはこれら表示を、実行のためどこに（どのデバイスに）割り当てすべきか、また、実行時中にこれら表示がバインドされるべきプロセスエンティティ（論理的および物理的）を定義することによって、グラフィック表示がプロセスプラントの構成に統合されることを可能にする。所望なら、例えば、図23および24等の構成階層スクリーンを用いて、必要なダウンロードを示し、表示上でバージョン制御を実行することもできる。

10

【0142】

構成は、構成システム内の複数のレベルで生じうる。はじめに、構成エンジニアは、まず表示クラスオブジェクトをライブラリ部704内のモジュールクラスオブジェクトと関連付けてよく、これは特定の表示クラスオブジェクトを、特定のモジュールクラスオブジェクト（例えば、機器モジュールクラスオブジェクト、エリアモジュールクラスオブジェクト等）にバインドする。その後、モジュールクラスオブジェクトがインスタンス化され、プロセスプラントの特定の部分に割り当てら指定される（または何らかの構成がされる）と、インスタンス化されたモジュールオブジェクトと関連する表示オブジェクト（例えば、グラフィック表示）がモジュールオブジェクトと共に同じハードウェアにバインドされる。このように、グラフィック表示をプラント内の適当なハードウェアに割り当てる、あるいはこのグラフィック表示オブジェクトを適当なハードウェアにダウンロードするのに、追加の労力は必要ない。代わりに、この場合、グラフィック表示オブジェクトはそれがバインドされているモジュールオブジェクトの構成に従う。更にまた、上述したように、表示クラスオブジェクトがモジュールクラスオブジェクトと関連している場合は、モジュールクラスオブジェクトに対して行われた変更は、自動的にこれらの変更を関連する表示クラスオブジェクト内のそれらと同じ要素に伝播させることができ、これらの変更は次いでグラフィック表示の実際のインスタンス化されたバージョンまで、下方に伝播されることができる。一例では、グラフィック表示が、Loopと呼ばれる制御モジュールを含む撹拌器（機器クラスオブジェクト）を含んでおり、この制御モジュールが選択され、名称変更された場合、構成システムは、グラフィック表示内のLoopへの参照を、新しい名称に自動的に変えて、それによりグラフィック表示のバインドが新しい名称で更新されることを保証してよい。しかしながら、変更が実行時システムで実施されるためには、依然として実際の新しく変更された表示を実行時マシンにダウンロードする必要があるという表示の隣に配置してよい。

20

30

40

【0143】

一方、グラフィック表示をプロセスプラントまたは構成システム内の異なる構成要素に、別々にまたは直接、指定してもよい。具体的には、図24に示すように、個々のグラフィック表示を、システム構成部706の下のプロセスグラフィックス部734に配置してよい。これら表示は、例えばライブラリ部704で選択すること、および部分734にドラッグおよびドロップすることによって、この部分に配置されてよい。もちろん、グラフィック表示を階層730の特定の部分に移動させる他の方法も同様に使用してもよい。いずれにせよ、部分734のグラフィック表示は、例えばフェースプレート表示、または他のクラスベースの、または非クラスベースの表示を含む場合があり、また一般にプラント内の全てのワークステーション、またはシステム構成部706によってカバーされたプラントの区分に

50

、ダウンロードされる資格がある。

【0144】

しかしながら、ユーザは表示をプロセスグラフィックス部734の外に及び特定のサブ部分の中に移動して、これら表示にアクセスすることになるプロセス制御システム内の個々のワークステーション（または他の表示デバイス）を管理し、したがってこれら表示を使用することができる個々のデバイスを管理することができる。特定の表示を論理的または物理的プラント部に割り当てるためには、ユーザは例えばグラフィック表示を選択し、グラフィック表示を構成階層730の特定の部分にドラッグおよびドロップして、それによりそのグラフィック表示とその表示を関連付けられる論理的または物理的エンティティとの間の関連を定義することができる。この構成ステップが生じた時点で、表示がその中にドロップされたまたはそれと関連付けられた構成の部分に基づいて、表示に対する解決テーブルが自動的にフィルアウト（記入）されてよい。

10

【0145】

例えば、グラフィック表示をライブラリ部704からプロセスグラフィックス部734にドラッグして、表示を実行時環境と関連付けてよい。また、例えばプロセスグラフィックス部734のグラフィック表示を、他の部分の中、例えばArea\_Aの表示部750（この表示をArea\_Aのエンティティおよび機能に対して特化するが、いずれのArea\_A表示デバイス上でも全一般的に利用可能にする）、Area\_Bの表示部754等に、移動することができる。同様に、グラフィック表示を論理的エンティティのサブ部分と関連付ける、例えばLOOPと称する制御モジュール（表示フォルダ752で示されている）と関連付けてよく、これは、これら表示をこの論理的エンティティに特化し、この制御ループ機能を行なうマシン、あるいはこの制御ループ機能と関連付けられていると定義されているマシン上でのみ、利用できるようにする。加えて、表示を特定の表示デバイスまたは表示デバイスの機能サブ部分（例えば、オペレータステーション762内のオペレータ（Operator）機能またはオペレータステーション762内のシミュレーション（Simulation）機能770で定義されたもの）に指定してよい。具体的に詳細には示されていないが、ユーザは、エリア、セル、ユニット、機器モジュールに、並びにさまざまな制御部に、グラフィック表示を指定して、グラフィック表示がプロセスプラントと関連付けられ、その内で使用される方法を定義することもできる。

20

【0146】

グラフィック表示を、論理的エンティティ、例えばArea\_AまたはArea\_Bのようなエリア呼称で定義されるような制御計画に割り当てた場合、グラフィック表示はこれら論理的エンティティが割り当てられている物理的インタフェース（例えば、ワークステーション）に指定される。同様に、表示が、ユニット、セルまたは機器モジュールに指定された場合、表示は、そのユニット、セルまたは機器が指定されているワークステーションまたは他のインタフェースデバイスに指定される。

30

【0147】

モジュールが複数のグラフィック表示を含んでいる場合、モジュールはそれら表示に対して役割を定義してよく、あるいは表示が生成された時にグラフィック表示に指定された役割表示によって定義された、各表示が履行する役割を定義してよい。次いで、任意の表示デバイス内のそれらグラフィック表示へのアクセスを、表示に対して定義された役割と関連付け、あるいはそれによって限定してよい。更にまた、グラフィック表示を、エリアまたはモジュールに指定する場合、このグラフィック表示への参照が、それが属するエリアまたはモジュールの表示フォルダ内に、グラフィック表示として出現してよい。グラフィック表示をエリアまたはモジュールにひとたび指定すると、このエリアまたはモジュールによって所有された表示参照、即ちこの表示は、エリアまたはモジュールのあらゆる指定または移動の間、エリアまたはモジュールに従うことになる。一方、エリアまたはモジュールがワークステーションに指定された場合、そのエリアまたはモジュールに属する表示は、表示の役割によって定義されたそのワークステーションのオペレータサブシステムまたは機能に、自動的に指定されることになる。もちろん、個々のグラフィック表示は

40

50

、構成階層の物理ネットワーク部740内で、特定のインタフェースデバイスまたはインタフェースデバイスのサブシステム内に、あるいはその下に、直接配置されることによって、表示デバイスのオペレータサブシステムに直接指定されてよい。

【0148】

したがって、ここで理解されるように、グラフィック表示は、構成スクリーン730内のこれらエンティティのそれぞれに、ドラッグ及びその中にドロップされることによって、個々のノード、制御エリア、制御モジュール、プロセスモジュール、または他の論理的制御エンティティ、並びに個々の表示デバイスおよびそのサブシステムに指定されてよい。もちろん所望なら、同じグラフィック表示を複数のステーションに指定してもよく、また各表示（ダウンロードされたまたは単に指定された）のステータスを、階層730内の各種のフォルダーにおいて各表示名称に対して与えられたアイコンで示してもよい。所望なら、階層730において他のアイコンを使用して、グラフィック表示が何時メモリにロックされるか、あるいはそれらをディスク上の仮想メモリに維持することができるかどうかを示すことができる。

10

【0149】

一般的に、グラフィック表示は、表示内容（例えば表示を表示スクリーンに視覚的に現わす表わし方を定義するXAMLスクリプト）、ローカルテーブル及び参照であってよく、この参照は、他の表示への参照、並びに実行時パラメータ（例えば、制御パラメータ等）への参照、表示制御への参照、実行時エイリアス（それらのうちのいくつかは実行時にバインドされてよい）、ローカルテーブル参照、およびグローバルテーブル参照であってよい。一般的に、構成エンジンは、グラフィック要素（XAML）のグラフィック部分を参照とは別に扱うことにより、グラフィック表示を取り扱いおよび使用を、より容易にする。具体的には、構成システムは構成階層で行なわれた指定に基づく動的エイリアスを使用しない参照をフィルアウトすることによって、構成プロセス中にグラフィック表示をバインドしてよい。したがって、例えばユーザが構成階層730のプロセスグラフィックス部734またはライブラリ部704から、グラフィック表示を特定のエリアまたは制御モジュールにドラッグすると、グラフィック内でバインドが自動的に行なわれて、グラフィック表示内の参照がそのエリアまたはモジュールと関連する特定の要素にバインドされる。この方法では、ダウンロードバインドが、ユーザによって行なわれている構成に基づいて自動的に行なわれてよく、これはユーザがこれらバインドを指定するために手動で行なわなければならない作業量を低減する。

20

30

【0150】

更にまた、構成ステップで解決されていないバインドがある場合、構成エンジンは、バインドを解決する（例えばバインドまたは参照を変える）ための情報を提供するために、ユーザに問合せてよい。更にまた、所望なら、バルク編集ツール（例えば、Microsoft Excel（登録商標）のようなスプレッドシートプログラム）を使用して、構成エンジニアが構成プロセス中に、解決またはバインドパラメータをフィルアウトまたは定義するのを助けてよい。いずれにせよ、これら構成ステップは図13に対して述べた解決テーブル208（これは、グラフィック表示を実行時環境で使用することを可能にする）をフィルインする。したがって、ユーザは一般に、米国特許出願第2004/0199925号公報で記述されたプロセスモジュールに対してユーザが行うのと同じ方法で、グラフィック表示（およびグラフィック要素）のバインドおよびダウンロードを行なうことができる。

40

【0151】

もちろん、ユーザまたは構成エンジニアは、グラフィック表示上でクリックし、それを階層730から除去することによって、あるいは、さもなければダイアログボックスを起動してこのグラフィック表示を指定解除させることによって、グラフィック表示を指定解除してよい。更にまた、ユーザは特定のグラフィック表示またはグラフィック表示のセットが指定される方法を確認でき、またこのような情報は、例えば図25に示されているダイアログボックス800によって与えられてよい。図25のダイアログボックス800は、2つの表示、Display1およびDisplay2のそれぞれが、プロセスプラント内で構成される方法を示す。

50

具体的には、ダイアログボックス800は表示のそれぞれがどこで使用されるか（例えば、Display1は、Area\_Aと呼ばれるプラントエリア及びMod1と呼ばれる制御モジュールで使用される）、それが指定される物理的ステーション（例えば、Display1はOper1と呼ばれるオペレータステーションに指定される）、および、表示が実行する役割（例えば、Display1はその各用において1次表示である）を示す。

#### 【0152】

表示の役割は、任意の所望の方法で変更することもでき、グラフィック表示を指定することができる構成システムの部分、または誰がグラフィック表示を、すなわちどんな機能に対して使用することができるか、を示すのに使用してよいことが理解されよう。グラフィック表示は、所望なら、複数の役割を実行するのに使用してよい。もちろん、上記のように、ユーザがエリアまたはある種のモジュールをワークステーションまたは他の表示デバイスに指定する場合、そのエリアまたはモジュールと関連する任意の表示もワークステーションに指定され、そのエリアまたは他のモジュールに対するグラフィカルサポートがワークステーション上で与えられることが保証される。また、グラフィック表示がワークステーションまたは他の表示デバイスに指定される場合、グラフィック表示は、そのグラフィック表示の役割（例えば、グラフィック表示が、シミュレーション表示、オペレータ表示等であるかどうか）に基づいて、ワークステーションのサブシステムに指定されてよい。追加的に、ユーザはグラフィック表示をワークステーションまたは他の表示デバイスの特定のサブシステムにドラッグおよびドロップしてよく、また役割が誤って指定された場合、構成システムはダイアログボックスを呈示して誤った指定が行われていることをユーザに示してよく、あるいは、ユーザがグラフィック表示の役割を変更することを可能にしてよい。

10

20

#### 【0153】

構成エンジンおよび図23および24の構成階層730を用いて、適当なまたは所望の指定を行なった後に、ユーザオペレータは、グラフィック表示の実際のダウンロードを行なってよく、これは、グラフィック表示を、構成システムで定義された目的および役割に対して構成階層730で定義された適当なオペレータワークステーションに、実際にダウンロードさせる。このダウンロード中に、表示内のバインドは、エリア、ループ、制御モジュール、プロセスモジュール等への指定、並びに、オペレータワークステーションまたは他の表示デバイスへの指定に基づいて、解消される。

30

#### 【0154】

所望なら、構成階層（図23に示す）の検索結果部708は、ユーザがプロセスプラントまたは構成システム全体を通してグラフィック表示で使用される任意の個々のグラフィック表示またはグラフィック要素の検索を行なうことを可能にしてよい。具体的には、構成エンジンはユーザによって起動された場合、特定のグラフィック表示またはグラフィック要素の検索を行なうことができる検索エンジンを含んでいてよく、またこのような検索の結果を検索結果部708に記憶して、任意の特定のグラフィック要素またはグラフィック表示の用途の完全なリストをユーザに提供してよい。このような検索は、変更を行っているユーザがそれらの変更を実行するためにどのグラフィック表示が変更される必要があるか、あるいはどんなダウンロードが生ずる必要があり得るか、を確認することを可能にするのに有用である。

40

#### 【0155】

ここで述べるソフトウェアのうちの任意のものは、実行された時、任意のコンピュータ読取可能なメモリ（例えば、磁気ディスク、レーザディスク、またはその他の記憶媒体上）、コンピュータまたはプロセッサのRAMまたはROM等に記憶されてよい。同様に、このソフトウェアは例えばコンピュータ読取可能なディスクまたは他の可搬コンピュータ記憶メカニズム上に、あるいは、電話線、インターネット、ワールドワイドウェブ（World Wide Web）、任意の他のローカルエリアネットワークまたは広域ネットワーク等の通信チャンネル上など、任意の既知のまたは所望の配信方法を用いて、ユーザ、プロセスプラントまたはオペレータワークステーションに配信されてよい（この配信は、このようなソフトウ

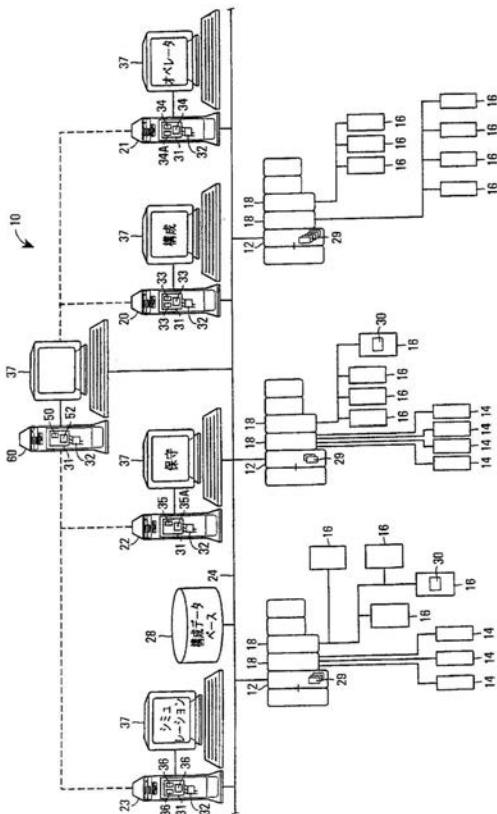
50

エアを、可搬記憶媒体を介して提供することと同じ、あるいはそれと互換できることと見なされる)。更にまた、このソフトウェアは変調または暗号化なしに直接提供されてよく、あるいは任意の適当な変調搬送波および/または暗号化技法を用いて、変調および/または暗号化されてから通信チャンネル上を伝送されてよい。

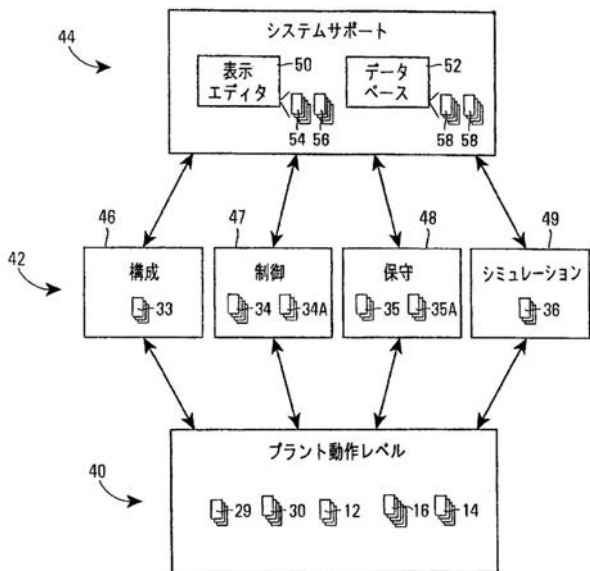
【 0 1 5 6 】

以上、本発明を例示のみを意図したものであって発明の限定を意図したものではない特定の例を参照して説明したが、発明の趣旨および範囲を逸脱することなしに、開示された実施形態に変更、追加または削除を行ってよいことは、この技術分野における当業者には明らかであろう。

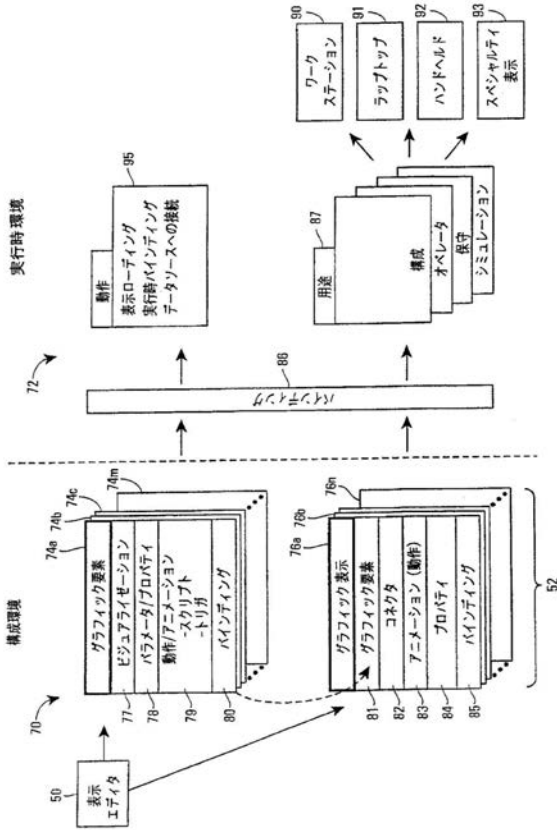
【 図 1 】



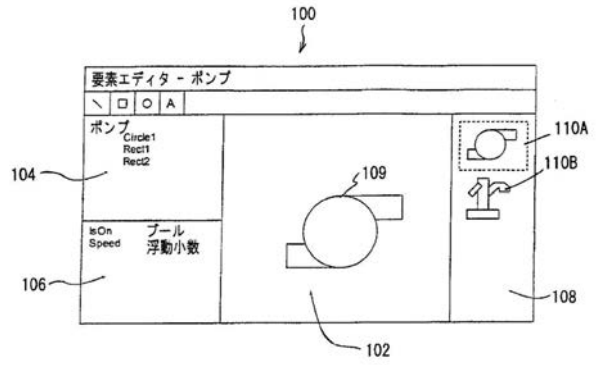
【 図 2 】



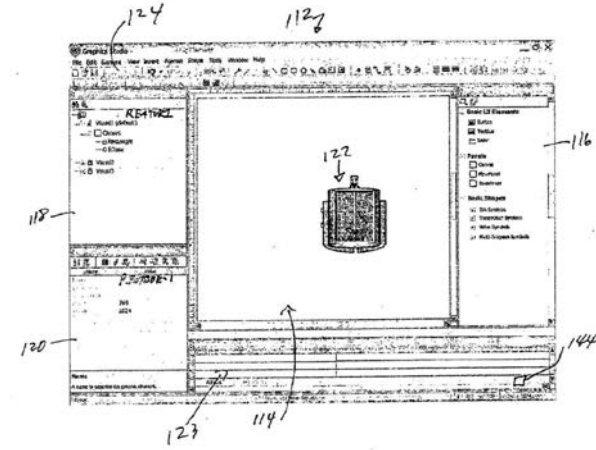
【図 3】



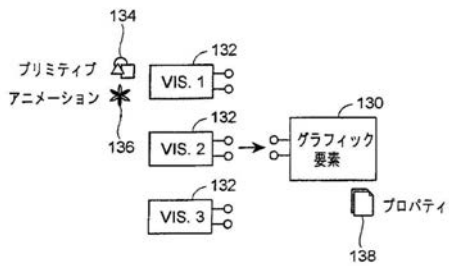
【図 4】



【図 5】



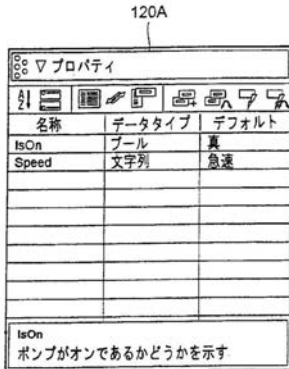
【図 6】



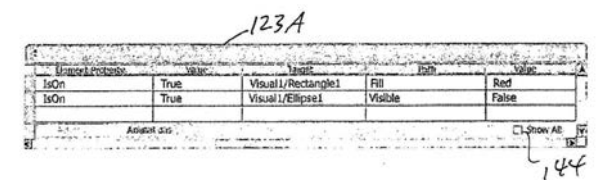
【図 8】



【図 7】

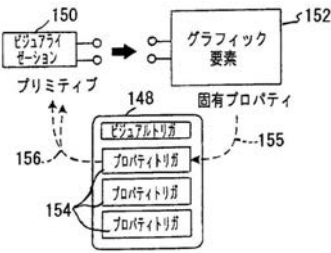


【図 9】

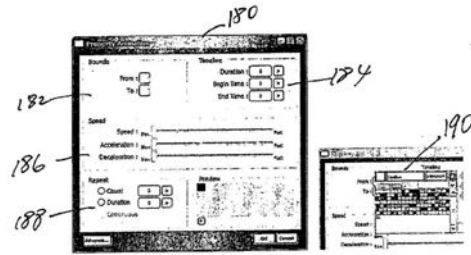




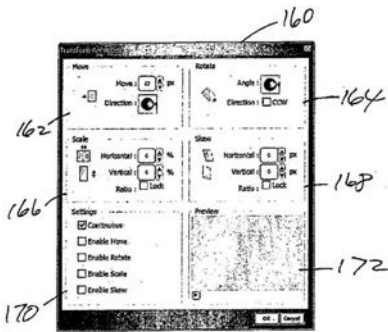
【図10】



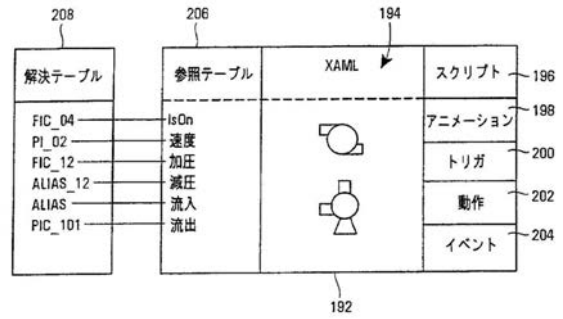
【図12】



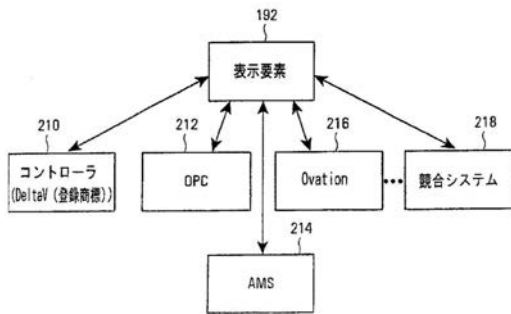
【図11】



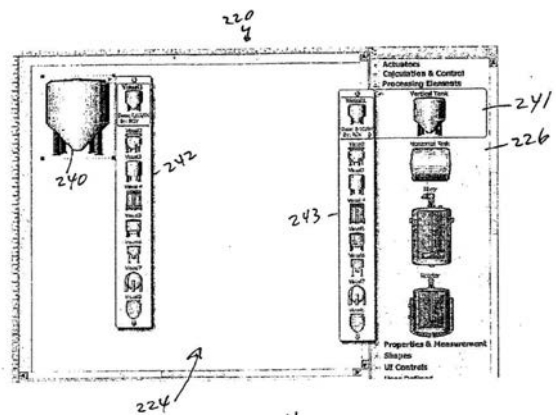
【図13】



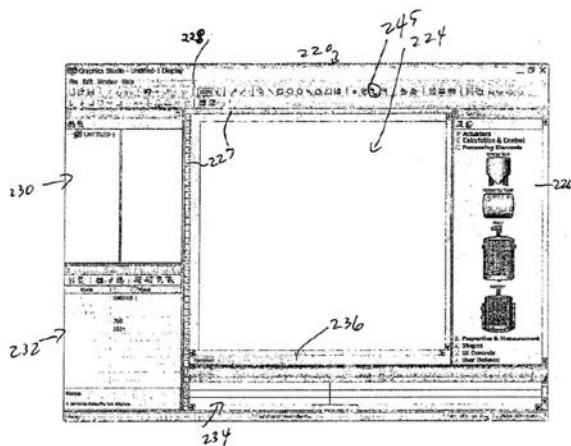
【図14】



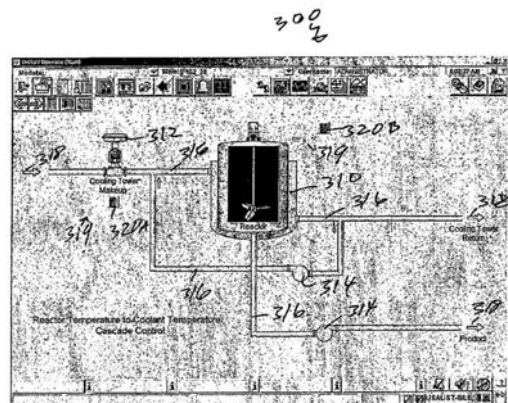
【図16】



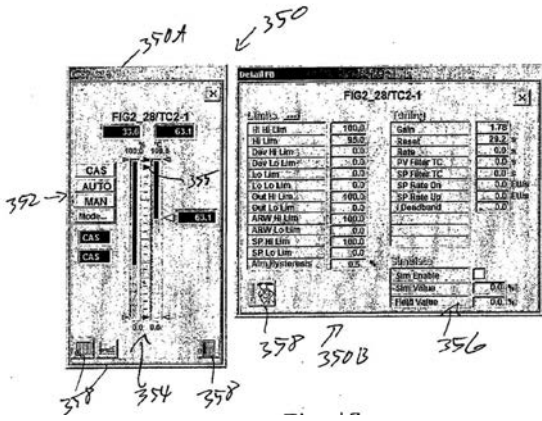
【図15】



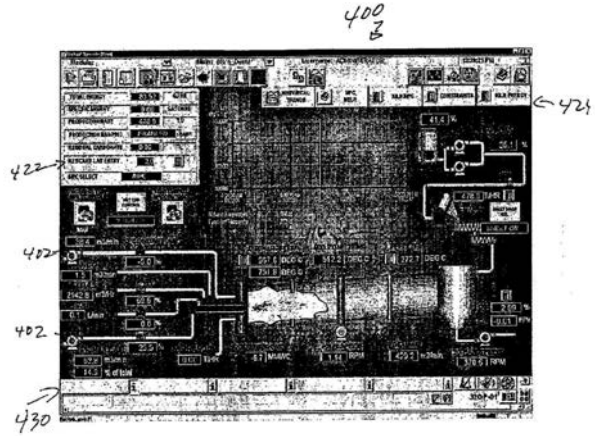
【図17】



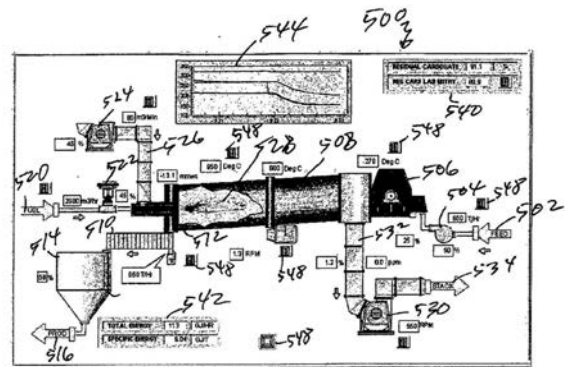
【 図 1 8 】



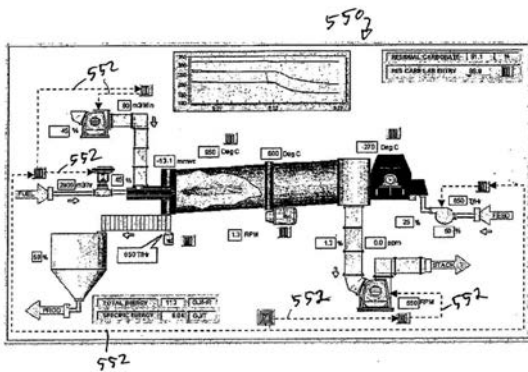
【 図 1 9 】



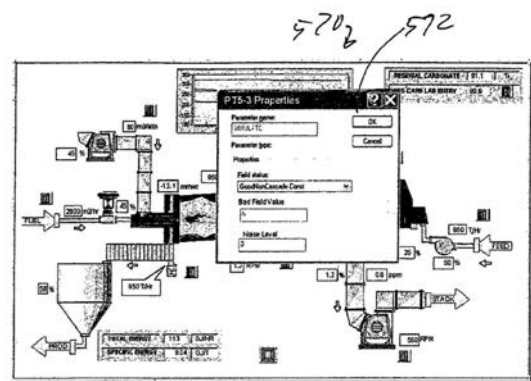
【 図 2 0 A 】



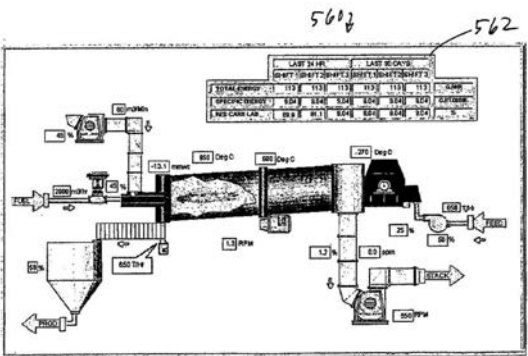
【 図 2 0 B 】



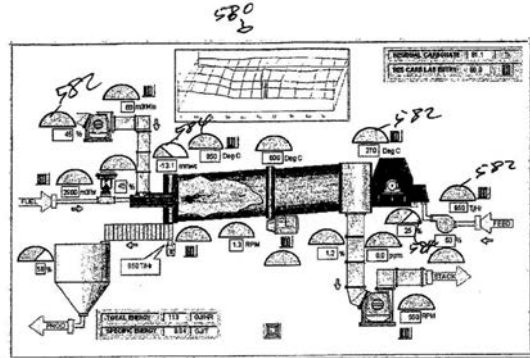
【 図 2 0 D 】



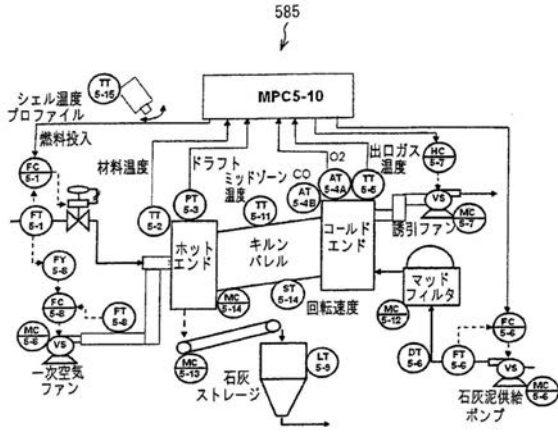
【 図 2 0 C 】



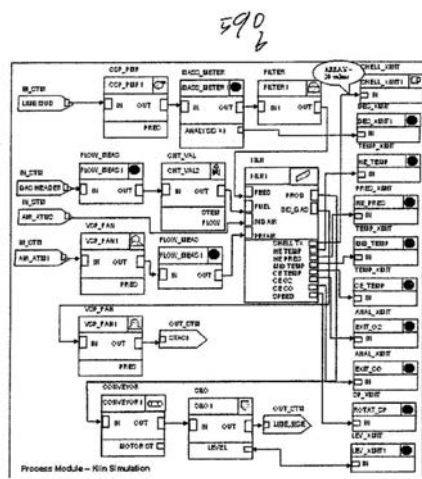
【 図 2 0 E 】



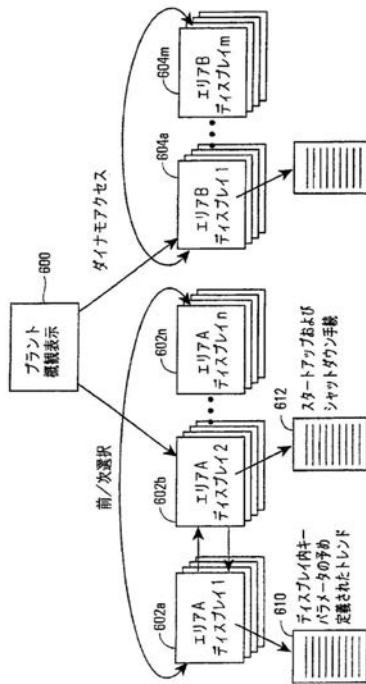
【図 2 1 A】



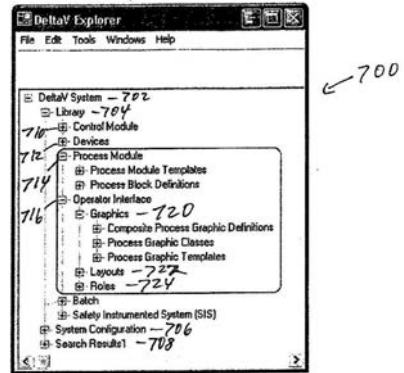
【図 2 1 B】



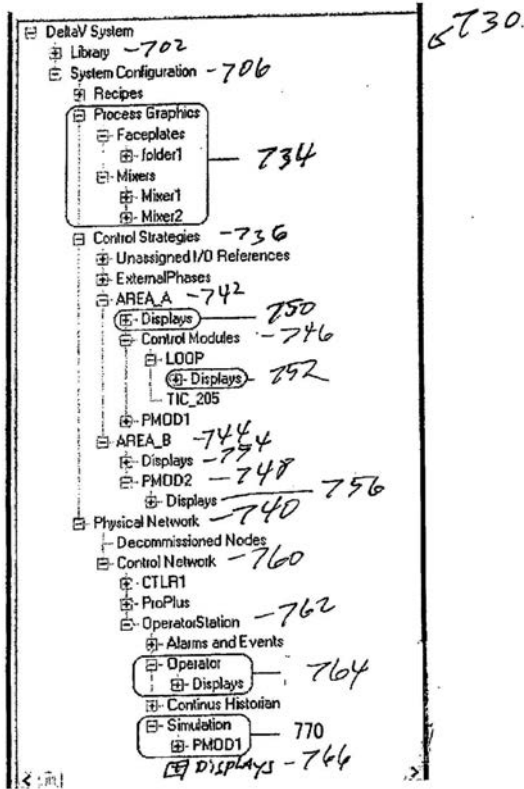
【図 2 2】



【図 2 3】



【 図 2 4 】



【 図 2 5 】

Selected Object	Direction	Object	Type
Display1	used in	Plant Area: Area_A	Primary Display
	used in	Control Module: Mod1	Primary Display
	Assigned to	Operator Station: Oper1	
Display2	used in	Plant Area: Area_B	Detail Display
	used in	Control Module: Mod2	Detail Display
	Assigned to	AppStation: Vks1	

【 手続 補正 書 】

【 提出 日 】 平成 29 年 6 月 12 日 (2017.6.12)

【 手続 補正 1 】

【 補正 対 象 書 類 名 】 特 許 請 求 の 範 囲

【 補正 対 象 項 目 名 】 全 文

【 補正 方 法 】 変 更

【 補正 の 内 容 】

【 特 許 請 求 の 範 囲 】

【 請 求 項 1 】

プロセスプラント内の構成要素の動作を表し且つ各々他の機能及びインタフェースとは独立して実行されるように開発された、複数のインタフェースで実行され且つ互いに無関係に異なる会社によって開発された複数の機能に共通のプログラムである表示オブジェクトを生成する表示エディタプログラムであって、前記表示エディタプログラムは、

ユーザが、操作手段を表示デバイスのスクリーンに対して操作して、前記スクリーンに表示された、前記プロセスプラント内の前記構成要素の複数の視覚的特徴のライブラリから、前記視覚的特徴を編集キャンバス上に配置し、

ユーザが、前記操作手段を前記スクリーンに対して操作して、前記配置された前記視覚的特徴に対してプロパティを定義し、

ユーザが、前記操作手段を前記スクリーンに対して操作して、前記プロパティと前記構成要素とを結び付けることを可能にすることにより、

インスタンス化されて実行可能なプログラムであるクラスオブジェクトとして、表示オブジェクトを生成する、表示エディタプログラム。

【 請 求 項 2 】

前記表示オブジェクトは、ユーザにより指示されると、前記プロセスプラントに関する情報を表示する更なる表示オブジェクトを呼び出す呼び出し表示プログラムを備え、

前記表示エディタプログラムは、前記呼び出し表示プログラムがユーザにより指示されると、前記更なる表示オブジェクトを呼び出し且つユーザが前記更なる表示オブジェクトへ情報を入力することを可能にする、請求項 1 に記載の表示エディタプログラム。

【請求項 3】

前記更なる表示オブジェクトは、前記構成要素についての情報を提供する、請求項 2 に記載の表示エディタプログラム。

【請求項 4】

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の情報を表示するプログラムを含み、

前記更なる表示オブジェクトは、前記プロセスプラントの部分の第 2 の情報を表示するプログラムを含み、

前記第 1 の情報は、前記プロセスプラントの部分について、前記第 2 の情報のデータソースとは異なるデータソースから得られる、請求項 1 に記載の表示エディタプログラム。

【請求項 5】

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、

前記更なる表示オブジェクトは、前記プロセスプラントの、前記第 1 の構成要素とは異なる第 2 の構成要素を表示する、請求項 1 に記載の表示エディタプログラム。

【請求項 6】

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、

前記更なる表示オブジェクトは、前記プロセスプラントの第 1 の構成要素の詳細な情報を表示する、

請求項 1 に記載の表示エディタプログラム。

【請求項 7】

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の機能的目的のための第 1 の表示を提供するプログラムを含み、

前記更なる表示オブジェクトは、前記プロセスプラントの部分の、前記第 1 の機能的目的とは異なる第 2 の機能的目的のための第 2 の表示を提供するプログラムを含む、

請求項 1 に記載の表示エディタプログラム。

【請求項 8】

前記第 1 の機能的目的のための前記第 1 の表示は、制御オペレータ表示、保守表示、ビジネス表示、シミュレーション表示、及びエンジニアリング表示のうちの何れか一つの表示であり、

前記第 2 の機能的目的のための前記第 2 の表示は、前記制御オペレータ表示、前記保守表示、前記ビジネス表示、前記シミュレーション表示、及び前記エンジニアリング表示のうちの、前記一つの表示以外の何れか一つの表示である、

請求項 7 に記載の表示エディタプログラム。

【請求項 9】

前記制御オペレータ表示は、前記プロセスプラントの運転中に一つ以上の構成要素と関連するプロセス値を示す、請求項 8 に記載の表示エディタプログラム。

【請求項 10】

前記保守表示は、前記プロセスプラント内の構成要素の健全性またはステータスの表示を示す、請求項 8 又は請求項 9 に記載の表示エディタプログラム。

【請求項 11】

前記表示エディタプログラムは、更なる表示オブジェクトを生成し、

前記更なる表示オブジェクトは、前記プロセスプラント内で実行される制御ルーチンを示す表示にユーザがアクセスすることを可能にする、請求項 1 に記載の表示エディタプロ

グラム。

【請求項 1 2】

前記表示オブジェクトには、ユーザにより定義可能な、前記表示オブジェクトの実行中における前記視覚的特徴の動作のプログラムが含まれる、請求項 1 に記載の表示エディタプログラム。

【請求項 1 3】

前記動作は、アニメーションによって定義される、請求項 1 に記載の表示エディタプログラム。

【請求項 1 4】

プロセスプラント内の構成要素の動作を表し且つ各々他の機能及びインタフェースとは独立して実行されるように開発された、複数のインタフェースで実行され且つ互いに無関係に異なる会社によって開発された複数の機能に共通のプログラムである表示オブジェクトであって、

前記表示オブジェクトは、インスタンス化されて実行可能なクラスオブジェクトであり、

前記表示オブジェクトは、

前記プロセスプラント内の前記構成要素の複数の視覚的特徴のライブラリから視覚的特徴を、ユーザが操作手段を表示デバイスのスクリーンに対して操作することにより、選択することを可能にするプログラムと、

前記ユーザにより選択された前記視覚的特徴に対してプロパティを、ユーザが、前記操作手段を前記スクリーンに対して操作することにより、定義することを可能にするプログラムと、

を含み、

前記プロパティと前記構成要素とが、ユーザによって対応付けられている、

表示オブジェクト。

【請求項 1 5】

前記表示オブジェクトは、ユーザにより指示されると、前記プロセスプラントに関する情報を表示する更なる表示オブジェクトを呼び出す呼び出し表示プログラムを備える、請求項 1 4 に記載の表示オブジェクト。

【請求項 1 6】

前記更なる表示オブジェクトは、前記構成要素についての情報を提供する、請求項 1 5 に記載の表示オブジェクト。

【請求項 1 7】

請求項 1 4 に記載の表示オブジェクトと、

更なる表示オブジェクトと、

を含む一組の表示オブジェクトであって、

前記表示オブジェクトは、前記プロセスプラントの部分の第 1 の情報を表示し、

前記更なる表示オブジェクトは、前記プロセスプラントの部分の第 2 の情報を表示し、

前記第 1 の情報は、前記プロセスプラントの部分について、前記第 2 の情報のデータソースとは異なるデータソースから得られる、一組の表示オブジェクト。

【請求項 1 8】

請求項 1 4 に記載の表示オブジェクトと、

更なる表示オブジェクトと、

を含む一組の表示オブジェクトであって、

前記表示オブジェクトは、前記プロセスプラントの第 1 の構成要素を表示し、

前記更なる表示オブジェクトは、前記プロセスプラントの、前記第 1 の構成要素とは異なる第 2 の構成要素を表示する、一組の表示オブジェクト。

【請求項 1 9】

請求項 1 4 に記載の表示オブジェクトと、

更なる表示オブジェクトと、

を含む一組の表示オブジェクトであって、  
前記表示オブジェクトは、前記プロセスプラントの第1の構成要素を表示し、  
前記更なる表示オブジェクトは、前記プロセスプラントの第1の構成要素の詳細な情報を表示する、  
一組の表示オブジェクト。

【請求項20】

請求項14に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記表示オブジェクトは、前記プロセスプラントの部分の第1の機能的目的のための第1の表示を提供し、  
前記更なる表示オブジェクトは、前記プロセスプラントの部分の、前記第1の機能的目的とは異なる第2の機能的目的のための第2の表示を提供する、  
一組の表示オブジェクト。

【請求項21】

前記第1の機能的目的のための前記第1の表示は、制御オペレータ表示、保守表示、ビジネス表示、シミュレーション表示、及びエンジニアリング表示のうちの何れか一つの表示であり、  
前記第2の機能的目的のための前記第2の表示は、前記制御オペレータ表示、前記保守表示、前記ビジネス表示、前記シミュレーション表示、及び前記エンジニアリング表示のうちの、前記一つの表示以外の何れか一つの表示である、  
請求項20に記載の一組の表示オブジェクト。

【請求項22】

前記制御オペレータ表示は、前記プロセスプラントの運転中に一つ以上の構成要素と関連するプロセス値を示す、請求項21に記載の一組の表示オブジェクト。

【請求項23】

前記保守表示は、前記プロセスプラント内の物理的構成要素の健全性またはステータスの表示を示す、請求項21又は請求項22に記載の一組の表示オブジェクト。

【請求項24】

請求項14に記載の表示オブジェクトと、  
更なる表示オブジェクトと、  
を含む一組の表示オブジェクトであって、  
前記更なる表示オブジェクトは、前記プロセスプラント内で実行される制御ルーチンを示す表示にユーザがアクセスすることを可能にする、一組の表示オブジェクト。

【請求項25】

前記表示オブジェクトには、ユーザにより定義可能な、前記表示オブジェクトの実行中における前記視覚的特徴の動作が含まれる、請求項14に記載の表示オブジェクト。

【請求項26】

前記動作は、アニメーションによって定義される、請求項25に記載の表示オブジェクト。

【請求項27】

プロセスプラントで使用するためのプロセス制御システムであって、  
プロセスを実行するために共に動作する複数の構成要素と、  
前記複数の構成要素に通信により接続されたプロセスコントローラと、  
前記プロセスコントローラに実装され、前記複数の構成要素の動作を制御する一つ以上の制御ルーチンと、  
プロセッサおよび表示エリアを含む表示デバイスと、  
請求項14～請求項16、請求項25、及び請求項26の何れか1項に記載の表示オブジェクトと、  
を含むプロセス制御システム。

## 【請求項 28】

プロセスプラントで使用するためのプロセス制御システムであって、  
プロセスを実行するために共に動作する複数の構成要素と、  
前記複数の構成要素に通信により接続されたプロセスコントローラと、  
前記プロセスコントローラに実装され、前記複数の構成要素の動作を制御する一つ以上の制御ルーチンと、  
プロセッサおよび表示エリアを含む表示デバイスと、  
請求項 17 ~ 請求項 24 の何れか 1 項に記載の一組の表示オブジェクトと、  
を含むプロセス制御システム。

## 【請求項 29】

プロセスプラントで使用するためのプロセス制御システムであって、  
プロセスを実行するために共に動作する複数の構成要素と、  
前記複数の構成要素に通信により接続されたプロセスコントローラと、  
前記プロセスコントローラに実装され、前記複数の構成要素の動作を制御する一つ以上の制御ルーチンと、  
プロセッサおよび表示エリアを含む表示デバイスと、  
請求項 1 ~ 請求項 13 の何れか 1 項に記載の表示エディタプログラムと、  
を含むプロセス制御システム。



## フロントページの続き

- (72)発明者 ハオ、テニソン  
 フィリピン国 クエゾン シティ 1114 エスエムエイチ ステーション カタリーナ ストリート 109
- (72)発明者 デ ガズマン、フランシス  
 フィリピン国 アンティポロ リーザル パインズ シティ エグゼクティブ ヴィレッジ、バーンハム ロード 5
- (72)発明者 カンプニー、ブルース  
 アメリカ合衆国 78653 テキサス州 メイナー レクター ループ 13814
- (72)発明者 ニクソン、マーク、ジェー .  
 アメリカ合衆国 78681 テキサス州 ラウンド ロック ブラックジャック ドライブ 1503
- (72)発明者 ギルバート、ステファン  
 アメリカ合衆国 78758 テキサス州 オースティン、マウンテン クウェイル ロード 10109

Fターム(参考) 3C223 AA01 BA03 BB02 BB06 BB11 CC02 DD03 GG01 HH04 HH05  
 HH08 HH15  
 5E555 AA02 AA05 AA23 AA28 AA79 BA02 BA42 BA69 BB02 BC18  
 CA02 CA18 CB02 CB34 CB38 CB39 CB40 CB45 CC03 CC11  
 DB04 DB05 DB11 DB18 DB41 DB53 DB56 DC13 DC21 DD01  
 DD07 EA17 FA00