

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6935972号
(P6935972)

(45) 発行日 令和3年9月15日(2021.9.15)

(24) 登録日 令和3年8月30日(2021.8.30)

(51) Int.Cl. F I
G05B 23/02 (2006.01) G05B 23/02 301Z

請求項の数 10 外国語出願 (全 122 頁)

<p>(21) 出願番号 特願2016-198941 (P2016-198941) (22) 出願日 平成28年10月7日 (2016.10.7) (65) 公開番号 特開2017-76387 (P2017-76387A) (43) 公開日 平成29年4月20日 (2017.4.20) 審査請求日 令和1年10月7日 (2019.10.7) (31) 優先権主張番号 62/239, 620 (32) 優先日 平成27年10月9日 (2015.10.9) (33) 優先権主張国・地域又は機関 米国 (US) (31) 優先権主張番号 15/274, 244 (32) 優先日 平成28年9月23日 (2016.9.23) (33) 優先権主張国・地域又は機関 米国 (US)</p>	<p>(73) 特許権者 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 78681-7430 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 エマーソン プロセス マネージメント (74) 代理人 100079049 弁理士 中島 淳 (74) 代理人 100084995 弁理士 加藤 和詳</p>
---	---

最終頁に続く

(54) 【発明の名称】 分散型工業システムにおけるソース非依存クエリ

(57) 【特許請求の範囲】

【請求項 1】

複数の電子データソースからの、プロセスプラントのオペレーションと関連付けられるプロセスプラントデータにアクセスするためのシステムであって、

第1の電子データソース及び第2の電子データソースであって、各々が、前記プロセスプラントのオペレーションと関連付けられる前記プロセスプラントデータを記憶する、有形の非一時的記憶媒体を備える、第1の電子データソース及び第2の電子データソースと

、前記第1及び第2の電子データソースと通信可能に連結されるプロセッサと、

前記プロセッサと通信可能に連結し、かつ命令を記憶する有形の非一時的記憶媒体を備える、プログラムメモリと、を備え、前記命令が、前記プロセッサによって実行されたとき、前記プロセッサに、

標準化クエリフォーマットを活用する標準化クエリを受信することと、

前記標準化クエリに基づいて、第1のデータソースに特異的なクエリを生成することであって、前記第1のデータソースに特異的なクエリが、前記第1の電子データソースと関連付けられる第1のクエリフォーマットを活用する、生成することと、

前記標準化クエリに基づいて、第2のデータソースに特異的なクエリを生成することであって、前記第2のデータソースに特異的なクエリが、前記第2の電子データソースと関連付けられる第2のクエリフォーマットを活用する、生成することと、

前記第1のデータソースに特異的なクエリを実行させて、前記第1の電子データソー

10

20

スから第 1 のデータのセットを獲得することと、

前記第 2 のデータソースに特異的なクエリを実行させて、前記第 2 の電子データソースから第 2 のデータのセットを獲得することと、

前記第 1 及び第 2 のデータのセットを含む集合データフレームを生成することと、を行わせる、システム。

【請求項 2】

前記標準化クエリが、前記第 1 の電子データソースの第 1 の指標と、前記第 2 の電子データソースの第 2 の指標とを含み、

前記第 1 の指標が前記第 1 のクエリフォーマットを特定し、前記第 2 の指標が前記第 2 のクエリフォーマットを特定する、請求項 1 に記載のシステム。

10

【請求項 3】

前記標準化クエリフォーマットが、前記第 1 のクエリフォーマットのクエリ構文と、前記第 2 のクエリフォーマットのクエリ構文との両方と別個のクエリ構文を活用し、

前記標準化クエリフォーマットの前記クエリ構文が、前記第 1 の電子データソースからの前記第 1 のデータのセット、または前記第 2 の電子データソースからの前記第 2 のデータのセットのいずれについても、獲得するように直接実行可能ではない、請求項 1 又は請求項 2 に記載のシステム。

【請求項 4】

前記標準化クエリが、第 1 のデータセット及び第 2 のデータセットのデータエントリに対応する期間を特定する時間範囲の指標を含む、請求項 1 ~ 請求項 3 の何れか 1 項に記載のシステム。

20

【請求項 5】

前記時間範囲が複数の期間を含み、かつ前記複数の期間のうちの少なくとも 2 つが、データが照会されない除外期間によって分離される、請求項 4 に記載のシステム。

【請求項 6】

前記時間範囲が複数の期間を含み、かつ前記時間範囲が、前記第 1 のデータセットと関連付けられる第 1 の期間と、前記第 2 のデータセットと関連付けられる第 2 の期間とを特定する、請求項 4 に記載のシステム。

【請求項 7】

前記第 1 の電子データソースがリレーショナルデータベースであり、前記第 2 の電子データソースが非リレーショナルデータベースである、請求項 1 ~ 請求項 6 の何れか 1 項に記載のシステム。

30

【請求項 8】

前記標準化クエリが、集合データフレームについてのサンプルレートの指標を含む、請求項 1 ~ 請求項 7 の何れか 1 項に記載のシステム。

【請求項 9】

第 1 のデータセットが、第 1 のサンプルレートを有するデータを含み、

第 2 のデータセットが、第 2 のサンプルレートを有するデータを含み、

前記集合データフレームを生成することが、前記第 1 及び第 2 のサンプルレートを整合することを含む、請求項 1 ~ 請求項 8 の何れか 1 項に記載のシステム。

40

【請求項 10】

前記第 1 及び第 2 のサンプルレートを整合することが、所望のサンプルレートを達成するために必要とされる、サンプルされていない時間に対応するデータ点を追加することを含む、請求項 9 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、広義には、プロセスプラント及びプロセス制御システムに関し、より具体的には、プロセスプラント及びプロセス制御システムによって生成されるリアルタイムデータのリアルタイムパフォーマンス監視及び分析に関する。

50

【 0 0 0 2 】

関連出願の相互参照

本出願は、2015年10月9日に出願された、DISTRIBUTED INDUSTRIAL PERFORMANCE MONITORING AND ANALYTICSと題される、米国仮出願第32/239,620号の利益を主張し、この仮出願の全体が、ここに、参照により本明細書に明示的に組み込まれる。本出願は、(i)2013年3月4日に出願された、「BIG DATA IN PROCESS CONTROL SYSTEMS」と題される米国出願第13/784,041号、(ii)2014年2月6日に出願された、「COLLECTING AND DELIVERING DATA TO A BIG DATA MACHINE IN A PROCESS CONTROL SYSTEM」と題される米国出願第14/174,413号、(iii)2014年8月11日に出願された、「SECURING DEVICES TO PROCESS CONTROL SYSTEMS」と題される米国出願第14/456,763号、(iv)2014年3月17日に出願された、「DATA MODELING STUDIO」と題される米国出願第14/216,823号、(v)2014年1月31日に出願された、「MANAGING BIG DATA IN PROCESS CONTROL SYSTEMS」と題される米国出願第14/169,965号、(vi)2014年3月14日に出願された、「DETERMINING ASSOCIATIONS AND ALIGNMENTS OF PROCESS ELEMENTS AND MEASUREMENTS IN A PROCESS」と題される米国出願第14/212,411号、(vii)2014年3月14日に出願された、「DISTRIBUTED BIG DATA IN A PROCESS CONTROL SYSTEM」と題される米国出願第14/212,493号、(viii)2014年10月6日に出願された、「STREAMING DATA FOR ANALYTICS IN PROCESS CONTROL SYSTEMS」と題される米国出願第14/506,863号、(ix)2014年10月6日に出願された、「REGIONAL BIG DATA IN PROCESS CONTROL SYSTEMS」と題される米国出願第14/507,188号、(x)2014年10月6日に出願された、「DATA PIPELINE FOR PROCESS CONTROL SYSTEM ANALYTICS」と題される米国出願第62/060,408号、及び(xi)2014年10月6日に出願された、「AUTOMATIC SIGNAL PROCESSING-BASED LEARNING IN A PROCESS PLANT」と題される米国出願第14/507,252号に関連し、これらの出願の全開示が、ここに、参照により本明細書に明示的に組み込まれる。

【背景技術】

【 0 0 0 3 】

物理的物質または生産物を製造、精製、変形、生成、または生産するための、化学、石油、工業、または他のプロセスプラントにおいて使用されるもの等の分散型プロセス制御システムは、典型的には、アナログバス、デジタルバス、またはアナログ/デジタル結合バスを介して、あるいは無線通信リンクまたはネットワークを介して、1つ以上のフィールドデバイスと通信可能に連結される、1つ以上のプロセスコントローラを含む。例えば、バルブ、バルブポジショナ、スイッチ、及び送信器(例えば、温度センサ、圧力センサ、レベルセンサ、及び流量センサ)である場合があるフィールドデバイスは、プロセス環境内に配置され、概して、バルブの開放または閉鎖、温度もしくは圧力といったプロセスパラメータ及び/または環境パラメータの測定等の物理的またはプロセス制御機能を実行して、プロセスプラントまたはシステム内で実行中の1つ以上のプロセスを制御する。公知のFieldbusプロトコルに準拠するフィールドデバイス等の、スマートフィールドデバイスは、制御計算、アラーム機能、及びコントローラ内で一般に実装される他の制御機能もまた実行し得る。プロセスコントローラは、これもまた同様に典型的にはプラント環境内に配置されるが、フィールドデバイスによって行われるプロセス測定を示す信号

及び/またはフィールドデバイスに関する他の情報を受信し、例えば、プロセス制御判断を行い、受信した情報に基づき制御信号を生成し、HART（登録商標）、Wireless HART（登録商標）、及びFOUNDATION（登録商標）Fieldbusフィールドデバイス等の、フィールドデバイスで実行される制御モジュールまたはブロックと連携する、異なる制御モジュールを実行するコントローラアプリケーションを実行する。コントローラ内の制御モジュールは、通信回線またはリンクを通じてフィールドデバイスに制御信号を送信し、それによって、プロセスプラントの少なくとも一部分の運転を制御して、例えばプラント内で実行または遂行中の1つ以上の工業プロセスの少なくとも一部分を制御する。例えば、コントローラ及びフィールドデバイスは、プロセスプラントのプロセス制御システムによって制御されているプロセスの少なくとも一部分を制御する。

10

【0004】

フィールドデバイス及びコントローラからの情報は、制御室もしくはより厳しいプラント環境からは離れた他の場所に典型的には配置される、オペレータワークステーション、パーソナルコンピュータもしくはコンピューティングデバイス、データヒストリアン、レポートジェネレータ、集中データベース、または他の集中管理コンピューティングデバイス等の、1つ以上の他のハードウェアデバイスに対して、通常、データハイウェイまたは通信ネットワークを通じて利用可能にされる。これらのハードウェアデバイスの各々は、典型的には、プロセスプラントにわたって、またはプロセスプラントの一部にわたって集中化される。これらのハードウェアデバイスは、例えば、オペレータが、プロセス制御ルーチンの設定の変更、コントローラもしくはフィールドデバイス内の制御モジュールの運転の修正、プロセスの現在の状態の閲覧、フィールドデバイス及びコントローラによって生成されたアラームの閲覧、担当者の訓練もしくはプロセス制御ソフトウェアの試験を目的としたプロセスの運転のシミュレーション、構成データベースの保守及び更新等の、プロセスの制御ならびに/またはプロセスプラントの運転に関する機能を実行することを可能にし得るアプリケーションを実行する。ハードウェアデバイス、コントローラ、及びフィールドデバイスによって活用されるデータハイウェイは、有線通信バス、無線通信バス、または有線及び無線通信バスの組み合わせを含み得る。

20

【0005】

例として、Emerson Process Managementによって販売されている、DeltaV（商標）制御システムは、プロセスプラント内の多様な場所に配置された異なる装置内に記憶され、それら異なる装置によって実行される複数のアプリケーションを含む。1つ以上のワークステーションまたはコンピューティングデバイス内に備わる、構成アプリケーションは、ユーザによる、プロセス制御モジュールの作成または変更、及びデータハイウェイを経由した、これらのプロセス制御モジュールの、専用分散型コントローラへのダウンロードを可能にする。典型的には、これらの制御モジュールは、通信可能に相互接続された機能ブロックで構成され、これらの機能ブロックは、それに対する入力に基づき制御スキーム内で機能を実行し、出力を制御スキーム内の他の機能ブロックに提供するオブジェクト指向プログラミングプロトコル内のオブジェクトである。また、構成アプリケーションは、データをオペレータに対して表示するため、かつプロセス制御ルーチン内の設定点等の設定のオペレータによる変更を可能にするために表示アプリケーションが使用するオペレータインターフェイスを、構成設計者が作成または変更することを可能にし得る。各専用コントローラ、及び一部の場合においては、1つ以上のフィールド装置は、実際のプロセス制御機能を実装するために、それらに割り当てられてダウンロードされた制御モジュールを実行するそれぞれのコントローラアプリケーションを記憶及び遂行する。表示アプリケーションは、1つ以上のオペレータワークステーション（またはオペレータワークステーション及びデータハイウェイと通信可能に接続された1つ以上のリモートコンピューティングデバイス）上で遂行され得るが、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経由してデータを受信し、このデータをユーザインターフェイスを使用して、プロセス制御システム設計者、オペレータ、またはユーザに表示して、オペレータのビュー、エンジニアのビュー、技術者のビュー等の、いくつかの異なる

30

40

50

るビューのうちのいずれかを提供し得る。データヒストリアンアプリケーションが、典型的には、データハイウェイにわたって提供されたデータの一部または全てを収集及び記憶するデータヒストリアンデバイスに記憶され、それによって遂行される一方で、構成データベースアプリケーションは、現在のプロセス制御ルーチン構成及びそれに関連するデータを記憶するためにデータハイウェイに接続される、更に離れたコンピュータで実行され得る。あるいは、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配置されてもよい。

【0006】

プロセスプラントまたはプロセス制御システムにおいて、異常状態または故障の証拠が見出された場合（例えば、アラームが発生した場合、あるいはプロセス測定値またはアクチュエータが過剰なばらつきを有することが見出された場合）、オペレータ、器械技術者、またはプロセス技術者は、典型的には、システム、及びシステムを通るプロセスのフローパスによって制御されているプロセスについての彼または彼女の知識と共に、分析ツールを使用して、異常状態または故障の証拠の生成に対して寄与した可能性のある上流の測定値及びプロセス変数の判断を試みる。例えば、オペレータは、プロセス制御デバイス（例えば、フィールドデバイス、コントローラ等）の出力から経時的に取り込まれているデータの履歴ログを、Delta V（商標）バッチ分析製品または連続データ分析ツールに供給して、異常状態または故障状態に対する様々なプロセス変数及び/または測定値の寄与を判断することを試みる場合がある。典型的には、ユーザは、どの履歴データログ及び/または他の時系列データを分析ツールに供給するかを決定し、プロセスについての彼または彼女の知識に基づいて、候補となる上流因子（例えば、測定値、プロセス変数等）を特定する。続いて、これらのデータ分析ツールは、主構成要素分析（PCA）、または部分最小二乗法、線形回帰等の他の分析技法を活用して、どの候補となった上流因子が、下流の予測される品質パラメータに影響を与えているかを判断する。したがって、分析ツールによって提供される出力の精度及び有効性はユーザの知識に基づき、すなわち制限されており、したがって、異常状態または故障の源に対する完全または正確な洞察を提供していない場合がある。

【0007】

加えて、このような分析は典型的にはプロセスからオフラインで実行されており、したがって、プロセスは、分析が行われている間に変化または移動する場合がある。例えば、典型的なプロセスプラントは、通常、1日に1回または2回の特定の分析のサイクル（例えば、データ収集及び分析サイクル）を実行するが、結果が分析され、任意の規定的アクションがプラント内で開発及び実装されるのは、分析が実行された後しばらく経ってからである。したがって、分析結果の精度が疑わしい場合があるだけでなく、そこから開発された規定的アクションも、最適ではないか、あるいは現在実行中のプロセスに対しては最早当てはまらない可能性がある。

【0008】

更に、現在公知のプロセス制御プラント及びプロセス制御システムのアーキテクチャは、限られたコントローラ及びデバイスのメモリ、通信帯域幅、ならびにコントローラ及びデバイスのプロセッサ能力によって強く影響される。例えば、現在公知のプロセス制御システムアーキテクチャにおいては、コントローラ内のダイナミック及びスタティック不揮発性メモリの使用は、通常、最小限にされるか、あるいは少なくとも慎重に管理される。結果として、システム構成中（例えば、演繹的に）、ユーザは、典型的には、コントローラ内のどのデータをアーカイブまたは保存すべきか、それが保存される頻度、及び圧縮が使用されるか否かを選択する必要がある。コントローラは、それに応じて、この制限されたデータ規則のセットで構成される。結果的に、トラブルシューティング及びプロセス分析において有用であり得るデータは、多くの場合、アーカイブされず、それが収集される場合、有用な情報は、データ圧縮に起因して失われている可能性がある。

【0009】

その上更に、工業またはプロセス制御プラントのデータセットは、現在のデータ処理分

10

20

30

40

50

析アプリケーションでは不十分であるという地点まで、そのサイズにおいて絶え間なく増加し続けている。典型的には、公知の分析技法は単にデータから値を抽出しようと試みるのみであり、その値が抽出されるデータセットの具体的なサイズまでは扱わず、特に、非常に大きいデータセット（プラントで生成された全てのプロセスデータ等）上でシームレスに動作することはない。更に、公知の分析技法は、典型的には、ストリーミングまたはストリーミングされるデータを取り扱うことができない。

【 0 0 1 0 】

上に考察した現在公知のプロセスプラント監視及び分析、ならびにプロセス制御システムの制限、ならびに他の制限は、例えばプラントの運転中、トラブルシューティング中、及び/または予測モデリング中に、プロセスプラントまたはプロセス制御システムの運転及び最適化において、それら自身を望ましくない形で顕在化させ得る。概して、リアルタイムの、現在の工業プロセスパフォーマンスデータを用いるリアルタイム分析は、公知の監視及び分析ツールでは不可能である。

【 発明の概要 】

【 0 0 1 1 】

分散型工業プロセスパフォーマンス監視及び/または分析用の技法、システム、装置、構成要素、及び方法が本明細書に開示される。そのような技法、システム、装置、構成要素、及び方法は、工業プロセス制御システム、環境、及び/またはプラントに対して適用することができ、これらは本明細書においては交換可能に、「自動化」、「工業制御」、「プロセス制御」、または「プロセス」システム、環境、及び/またはプラントとも呼ばれる。典型的には、そのようなシステム及びプラントは、分散型の様式で、物理的物質または生産物を製造、精製、変形、生成、または生産するように動作する、1つ以上のプロセスの制御を提供する。概して、該技法、システム、装置、構成要素、及び方法は、データ監視及び/またはデータ分析エンジン（本明細書においては交換可能に「分散型データエンジン」、「データエンジン」、または「DDE」とも呼ばれる）を、工業プロセスを制御するように提携して運転するデバイス内に、分散型の様式で埋め込むことを含む。例えば、分散型データエンジンは、プロセスプラントまたはプロセス制御システム内で実行中のプロセスを制御するように他のデバイスと共に動作するデバイス（例えば、フィールドデバイス、コントローラ、I/Oカード等のプロセス制御デバイス）内に製造されてもよく、かつ/あるいは分散型データエンジンは、そのようなデバイスとローカルで連結されるか、または直接連結されてもよい。追加的なデータエンジンが、通信ノード、ワークステーション、または他のオペレータインターフェイス、サーバの中等、プロセスプラント内に含まれる他のデバイスに埋め込まれるか、あるいはその中に製造されてもよい。一部の構成においては、データエンジンは、プロセスプラント内の様々な通信リンクに接続されるか、あるいは別様に、プラント内のプロセスの実行または制御中に送信されたリアルタイムデータへのビューまたはウィンドウを有するように、プロセスプラントに接続される。

【 0 0 1 2 】

データは、分散型制御システム（DCS）、プログラマブルロジックシステム（PLS）、及びプロセス制御安全衛生監視システム（SHM）において見られる従来型の通信システムの外側に典型的には存在する、1つ以上のデータ通信チャネル及びネットワークを例えば使用することで、様々な埋込式データエンジン間でストリーミングされる。そのような従来公知のプロセス制御通信ネットワーク及びリンクとの混乱を回避するために、データエンジン間で通信をストリーミングするために活用される通信チャネル及びネットワークは、本明細書においては交換可能に、「データ分析通信チャネル」、「データ分析チャネル」、「データ分析通信ネットワーク」、または「データ分析ネットワーク」と呼ばれる。ストリーミングされるデータは、データエンジンによって閲覧または観察されるリアルタイムデータを含み得る。例えば、分散型データエンジンが従来型のプロセス制御システム通信リンクに接続される（したがって、トラバースするデータを閲覧している）場合、データエンジンは、従来型の通信リンクをトラバースしているデータのコピーを、デ

10

20

30

40

50

ータ分析ネットワークを介して、1つ以上の他のデータエンジンにストリーミングし得る。別の例においては、分散型データエンジンがプロセス制御デバイスの中に製造されるか、またはそれに埋め込まれている場合、データ分析ネットワークを用いてデータエンジンによってストリーミングされる分析データは、デバイスによって受信、生成、または別様にプロセスされたデータのコピーを含み得る。追加的または代替的に、ストリーミングされるデータは、分析結果、規定的アクション等の、デバイスにおいてローカルで実施された1つ以上の分析に対応するデータを含み得る。このアーキテクチャによって、分析サービスが、ローカルでバインドされ、データソース付近で、またはデータソースにおいて提供されるが、一方で同時に、より大規模の分析の提供が可能となり、それによって、下でより詳細に説明されるように、帯域幅の使用及びシステムにまたがる処理サイクルを最小化しながら、タイムリーな結果及び最適化が提供される。

10

【0013】

ある態様においては、分散型工業プロセス監視及び分析システムは、プロセスを制御するように動作しているプロセスプラント内に埋め込まれた、複数の分散型データエンジン（DDE）を含む。DDEの各々は、制御されているプロセスの結果としてそれぞれデータを生成している、プロセスプラント内のそれぞれの1つ以上のデータソースに連結される。加えて、DDEの各々は、各DDEが連結されるそれぞれの1つ以上のデータソースによって生成されているデータを記憶する。本システムはまた、複数のDDE間の分析データのストリーミングをサポートし、複数のDDEにおいて記憶されるデータに関するクエリの送信をサポートするデータ分析ネットワークも含む。

20

【0014】

別の態様においては、方法には、プロセスを制御するように動作しているプロセス制御プラントのデータ分析ネットワーク内に含まれるクラスタの存在を告知することが含まれる。本方法はまた、その告知に対して応答し、プラントによるプロセスの制御から得られる連続データを生成するデータソースを登録することも含む。加えて、本方法は、データソースによって生成された連続データを受信すること、及びデータソースによって生成された連続データの少なくとも一部を、データ分析ネットワークを介して、データコンシューマにストリーミングすることを含む。本方法の1つ以上の部分が、例えば、クラスタによって実施されてもよい。

【0015】

更に別の態様においては、方法には、プロセスを制御するように動作しているプロセス制御プラントのデータ分析ネットワーク内に含まれる集中型クラスタの存在を告知することが含まれる。本方法はまた、その告知に対して応答し、プロセスを制御する結果として生成される、ストリーミングされる連続データを記憶するように構成されるローカルクラスタを登録することも含む。ローカルクラスタは、プロセスプラント内に埋め込まれ、プロセスを制御するように動作しながら、連続データの少なくとも一部分を生成する1つ以上のデータソースと通信可能に連結する。加えて、本方法は、ローカルクラスタからの、ストリーミングされる連続データの少なくとも一部分を受信すること、ローカルクラスタから受信したストリーミングされる連続データに1つ以上のデータ分析機能を実施すること、及び1つ以上のデータ分析機能の出力を、データ分析ネットワークを介して、データ

30

40

【0016】

別の態様においては、本システムは、連続データストリームにバインドされる、データ分析モジュールを含む。連続データストリームは、プロセスを制御するように動作しているプロセスプラント内に含まれるデータソースによってリアルタイムで生成される内容を有し、連続データストリームの内容は、プロセスを制御するオペレーションの結果として生成される。本システムは更に、連続データストリームに対してリアルタイムで動作するデータ分析モジュールから得られる、リアルタイムで生成される連続出力を表示するユーザインターフェイスを含む。ユーザインターフェイスに表示される連続出力は、1つ以上

50

の予測値の連続アップデートを含む。

【 0 0 1 7 】

更に別の態様においては、工業プロセス制御用のパフォーマンス監視及び分析システムは、ユーザ制御部のセットと、データモデルを表すデータダイアグラムのユーザによる作成を可能にするキャンバスとを提供するプラットフォームを含む。プラットフォームは更に、ユーザがデータダイアグラムを評価またはコンパイルして入力データセットに対し実行して、それによって出力データを生成することを可能にする。入力データセットは、プロセスを制御するオンラインプロセスプラントから得られる時系列データ（及び任意選択で他のデータ）を含む。本明細書で使用する場合、用語「時系列データ」は概して、典型的には1つ以上のデータソースによってある時間間隔にわたって生成される、一連のデータ点、値、またはセットを指す。

10

【 0 0 1 8 】

データダイアグラムは、ワイヤのセットで相互接続されたデータブロックのセットを含み、このワイヤのセットを介してデータがデータブロック間で移送される。データブロックのセットの各データブロックは、それぞれのデータオペレーションに対応し、0、1、または複数の入力コネクタを含み、0、1、または複数の出力コネクタを含む。それぞれの入力データは、入力コネクタ（複数可）を介して各データブロックで受信され、それぞれのデータブロックは、それぞれの入力データに対してそのそれぞれのデータオペレーションを実施する各データブロックから得られるそれぞれの出力データを、出力コネクタ（複数可）を介して提供する。更に、入力コネクタは、データブロックのセットの第1のデータブロックに含まれ、データブロックのセットに含まれる少なくとも1つのデータブロックのそれぞれのデータオペレーションは、データ分析機能を含み、データダイアグラムの異なる部分は、非同期的かつ別個にコンパイル可能であり、実行可能である。一部の実施形態においては、例えば入力コネクタが構成されておらず、出力コネクタが構成されていない場合、入力データ及び出力データは、データブロックによって内部で処理される。

20

【 0 0 1 9 】

ある態様においては、プロセス制御環境においてリアルタイム分析を実施するための方法は、グラフィカルユーザインターフェイスに第1のダイアグラムを作成することを含む。第1のダイアグラムは、プロセッサに、プロセス制御環境によって前もって生成及び記憶されたデータを操作させるように動作可能である、第1のプログラミング命令を表し得る。本方法はまた、第1のダイアグラムによって表される第1のプログラミング命令をコンパイルすること、コンパイルされた第1のプログラミング命令を実行して、結果として生じる第1の出力を作成すること、及び結果として生じた第1の出力を評価して、結果として生じた第1の出力を作成するのに使用されたモデルの予測値を決定することも含む。加えて、本方法は、第1のダイアグラムから第2のダイアグラムを自動的に生成することを含む。第2のダイアグラムは、プロセッサに、ライブデータソースにバインドさせ、少なくとも、結果として生じた第1の出力を作成するのに使用されたモデルを使用することで、ライブデータソースから受信したデータを評価させるように動作可能である、第2のプログラミング命令を表し得る。更に、本方法は、第2のダイアグラムによって表される第2のプログラミング命令をコンパイルすること、及びコンパイルされた第2のプログラミング命令を実行して、プロセス制御環境のオペレーションの態様を予測することを含む。

30

40

【 0 0 2 0 】

別の態様において、プロセス制御環境においてリアルタイム分析を実施するためのシステムは、プロセスプラント内で動作する複数のプロセス制御デバイスと、制御ネットワークを介して複数のプロセスデバイスと通信可能に連結するコントローラとを含む。本システムはまた、プロセス制御環境のオペレーションのデータを記憶する、有形の非一時的記憶媒体を備えるビッグデータストレージノードと、ビッグデータストレージノードと通信可能に連結するプロセッサと、プロセッサと通信可能に連結するプログラムメモリとを含む。このプログラムメモリは、プロセッサによって実行されたとき、プロセッサに、編集

50

キャンパスを含むグラフィカルユーザインターフェイスを表示させ、その編集キャンパス上に配置して第1のダイアグラムを作成するための、複数のグラフィカルプログラミング要素の選択を受信させるという命令を記憶する、有形の非一時的記憶媒体を備え得る。第1のダイアグラムは、ビッグデータストレージノードに記憶されたデータをプロセッサに操作させるように動作可能な、第1のプログラミング命令を表し得る。加えて、その命令がプロセッサによって実行された場合、プロセッサに、コンパイルされた第1のプログラミング命令を実行させて、結果として生じる第1の出力を作成させ、その結果として生じた第1の出力に基づいて予測的モデルを作成させることができる。その命令がプロセッサによって実行された場合、プロセッサには更に、第1のダイアグラムから第2のダイアグラムを自動的に生成させることができる。第2のダイアグラムは、プロセッサに、ライブデータソースにバインドさせ、少なくとも、結果として生じた第1の出力を作成するのに使用されたモデルを使用することで、ライブデータソースから受信したデータを評価させるように動作可能である、第2のプログラミング命令を表し得る。その命令がプロセッサによって実行された場合、プロセッサにはなおも更に、第2のダイアグラムによって表される第2のプログラミング命令をコンパイルさせ、コンパイルされた第2のプログラミング命令を実行させて、プロセス制御環境のオペレーションの態様を予測させることができる。

10

【0021】

別の態様においては、プロセス制御環境においてデータ分析を実施するための分析サービスは、ブロック定義のライブラリを含む。このライブラリは、有形の非一時的媒体に記憶することができ、ライブラリ内に記憶される各ブロック定義は、(i)プロセス制御環境のデータに関するアクションを実施するためのターゲットアルゴリズムと、(ii)1つ以上のブロックプロパティとを含み得る。この分析サービスは更に、有形の非一時的媒体に記憶される機械可読命令のセットを含み、この機械可読支持のセットは、プロセッサによって実行されたとき、(1)ディスプレイを介してキャンパスをユーザに呈示し、(2)ブロック定義のライブラリをユーザに呈示し、(3)1つ以上の対応するブロック定義のうちの1つ以上の選択を受信し、(4)1つ以上のブロック定義に対応する1つ以上のブロックをキャンパス上に配置して、1つ以上のブロックを含むモジュールを作成し、(5)(i)1つ以上のブロック、または(ii)モジュール、または(iii)1つ以上のブロック及びモジュールのいずれかに関して、1つ以上のプロパティの構成を促進し、かつ(6)(i)ブロックのうちの1つ以上、または(ii)モジュールの評価をもたらすように、動作可能である。

20

30

【0022】

加えて、分析サービスは、プロセッサ上で動作する実行サービスを含み、ここで、この実行サービスは、(1)1つ以上のブロックまたはモジュールを評価する要求を受信し、(2)1つ以上のブロックまたはモジュールに対応する命令をコンパイルし、(3)コンパイルされた命令を実行するために1つ以上のジョブプロセスを作成し、その1つ以上のジョブプロセスに、コンパイルされた命令を実行させ、かつ(4)1つ以上のジョブプロセスの結果を受信するように構成される。分析サービスはまた、1つ以上のジョブプロセスを実行する1つ以上のプロセッサを、1つ以上の分散型データエンジンと連結する通信ネットワークも含む。各分散型データエンジンは、プロセス制御環境内で生成されるデータを記憶する、有形の非一時的記憶媒体を備え得る。

40

【0023】

実際、プロセスプラントのオペレーションによって生成されるデータ、またはそれに関するデータは、リレーショナルデータベースまたは非リレーショナルデータベース等の複数のデータ記憶に記憶され得る。これらのデータ記憶は、様々なデータ構造及びクエリメカニズムを活用する場合があります、それにより、異なるデータ記憶のデータにアクセスするために異なるクエリ構文が必要とされる。様々なフォーマットを用いるデータ記憶へのデータアクセスを促進するための、標準化クエリについて本明細書では記載する。標準化データクエリフォーマットを活用するこの標準化クエリは、データ記憶のデータにアクセス

50

するために必要とされる情報を含むが、標準化クエリは、そのようなデータを獲得するように直接実行可能ではない場合がある。代わりに、データソースに特異的なクエリが、標準化クエリに基づいて生成される。これには、標準化クエリからクエリパラメータを抽出することと、特定のデータソースと関連付けられるデータソースに特異的なクエリフォーマットを活用して、1つ以上のデータソースに特異的なクエリを生成することが含まれ得る。このデータソースに特異的なクエリは、それらのそれぞれのデータソースにアクセスし、そこからデータを選択するように実行することができ、このデータはその後、標準化クエリで示されるデータを任意の所望のフォーマットで呈示するデータフレームを生成するようにフォーマットされ得る。これには、データソースに特異的なクエリによって獲得されたデータに関する、サンプリングレート、データプロパティ、または他のデータ特性を整合することが含まれ得る。一部の実施形態においては、プロセス制御または分析における更なる使用のために、異なるデータソースに特異的なクエリフォーマットを活用する、複数のデータソースからのデータを、集合データフレームへと組み合わせることができる。

10

【0024】

別の態様においては、電子データソースから時系列データを獲得するための命令を記憶する方法、システム、及びコンピュータ可読媒体について記載され、データの獲得元である電子データソースの指標を受信すること、データが獲得される時間範囲の指標を受信すること、獲得されるべきデータの特性の指標を受信すること、電子データソースと関連付けられる、データソースに特異的なクエリフォーマットを決定すること、獲得されるべきデータの時間範囲及び特性の指標に基づいて、データソースに特異的なフォーマットに従う、データソースに特異的なクエリを生成すること、データソースに特異的なクエリを実行させて、電子データソースからデータを選択すること、ならびに/または電子データソースからデータを受信することを含む。データの時間範囲及び特性の指標は、電子データソースからは独立する標準化クエリフォーマットで表されてもよく、電子データソースは、リレーショナルデータベースまたは非リレーショナルデータベースであり得る。

20

【0025】

本方法は、標準化クエリフォーマットを活用する標準化クエリを含み得る。時間範囲の指標及び特性の指標は、標準化クエリとして受信することができる。電子データソースの指標もまた、標準化クエリとして受信することができる。標準化クエリは更に、JavaScript Object Notation (JSON) 等の、データまたはファイルがデータを返すためのフォーマットを示し得る。標準化クエリはまた、JSONでフォーマットされたファイルでもあり得る。標準化クエリフォーマットは、電子データソースによって活用される本来の構文とは別個の構文を活用してもよい。標準化クエリフォーマットで活用されるそのような構文は、電子データソースからデータを獲得するように直接実行可能ではない場合がある。

30

【0026】

時間範囲の指標は、少なくとも1つの開始時間と、少なくとも1つの終了時間とを含み得る。時間範囲の指標は、獲得されるべきデータのサンプルレートを同様に含み得る。獲得されるべきデータの特性の指標は、プロセスプラントのオペレーションに関する1種類以上の測定値の指標を含み得る。この1種類以上の測定値は、プロセスプラント内に配置される1つ以上のフィールドデバイスからの測定値を含み得る。1種類以上の測定値の指標は、データと関連付けられる1つ以上のタグ、エイリアス、及びデータタイプを含み得る。

40

【0027】

本方法は更に、データが電子データソースから受信される際のフォーマットの指標を受信することを含み得、データは、示されるフォーマットで電子データソースから受信され得る。一部の実施形態においては、データが獲得される際のフォーマットの指標を受信して、電子データソースから受信したデータを、その示されるフォーマットに変換する場合がある。

50

【 0 0 2 8 】

データソースに特異的なクエリフォーマットは、電子データソースの種類に基づいて決定され得る。データソースに特異的なクエリを生成することは、獲得されるべきデータの示される時間範囲と、示される特性とを、データソースに特異的なクエリフォーマットにマッピングすることを含み得る。データソースに特異的なクエリを実行させることは、データソースに特異的なクエリを電子データソースのインターフェイスに送信することを含み得、それにより、このインターフェイスが、データソースに特異的なクエリを用いて、電子データソースを照会する。

【 0 0 2 9 】

電子データソースから受信したデータを含む、データフレームが生成され得る。このデータフレームは、複数のデータ点を含み得る。各データ点は、時間範囲内の時間と関連付けられ得る。各データ点は更に、開始時間後の、サンプル期間の整数倍と整合するか、あるいは開始時間である、時間範囲内の時間と関連付けられ得る。

【 0 0 3 0 】

別の態様においては、電子データソースからデータを提供するための命令を記憶する方法、システム、及びコンピュータ可読媒体について記載され、データ要求エンティティから、標準化クエリフォーマットを使用する標準化クエリを受信すること、標準化クエリからクエリパラメータを抽出すること、抽出したクエリパラメータに基づいて、データソースに特異的なクエリフォーマットで、データソースに特異的なクエリを生成すること、データソースに特異的なクエリを実行して、電子データソースからデータを獲得すること、及び/または獲得したデータをデータ受信エンティティに提供することを含む。標準化クエリフォーマットは、電子データソースによって活用される、データソースに特異的なクエリフォーマットとは異なり得る。電子データソースは、リレーショナルデータベースまたは非リレーショナルデータベースであり得る。データ要求エンティティは、本明細書で考察される Data Analytics Studio 等の分析プログラム内のクエリブロックであり得る。データ受信エンティティは、データ要求エンティティと同じエンティティであってもよい。標準化クエリは、電子データソースからデータを獲得するように直接実行可能ではない構文を活用する場合がある。

【 0 0 3 1 】

標準化クエリからクエリパラメータを抽出することは、データのある期間及びデータパラメータを決定することを含み得る。この期間は、開始時間と終了時間とを有し得、データパラメータは、電子データソースから獲得されるべきデータの種類または特性を示し得る。標準化クエリからクエリパラメータを抽出することは更に、サンプリングレートを決定することを含み得る。

【 0 0 3 2 】

獲得したデータを要求エンティティに提供することは、その期間と関連付けられるデータ点のセットを返すことを含み得る。各データ点は、データ点と関連付けられる値（複数可）に関する時間を示すタイムスタンプを含み得る。そのようなタイムスタンプは、開始時間である時間か、または開始時間後のサンプリングレートの期間の整数倍である時間かに限定され得る。完全な時系列を達成するために、1つ以上のデータ点が、開始時間後のサンプリングレートの期間の整数倍を示すタイムスタンプを伴うデータセットに対して追加されてもよい。追加されるデータ点は、これら1つ以上の追加されるデータ点の各タイムスタンプの最も直前の時間と関連付けられる、電子データソースのデータエントリの値を使用し得る。データ点のセットは、標準化クエリで特定されたフォーマットに基づいて、フォーマットされ得る。

【 0 0 3 3 】

データパラメータによって示される、データの種類または特性は、測定の種類またはある種類の測定デバイスからの測定値であり得る。データの種類または特性は更に、具体的な測定デバイスを示してもよく、この測定デバイスは、プロセスプラント内に配置されるフィールドデバイスであり得る。データパラメータは更に、獲得されるべきデータのタグ

10

20

30

40

50

、データ要求エンティティに提供されるべきデータのエイリアス、及び/またはデータ要求エンティティに提供されるべきデータのデータフォーマットタイプを示し得る。

【0034】

データをデータ要求エンティティに提供することは、獲得されたデータを含むデータフレームを、データ要求エンティティに送信することを含み得る。標準化クエリは、データフレーム用のフォーマットの指標を含み得る。データフレームは、データフレーム用のフォーマットの指標に従ってフォーマットされ得る。標準化クエリは、データフレームが、データ受信エンティティにJSONファイルとして提供されることを示し得る。

【0035】

別の態様においては、複数の電子データソースからのプロセスプラントデータにアクセスするための命令を記憶する方法、システム、及びコンピュータ可読媒体について記載され、標準化クエリフォーマットを活用する標準化クエリを受信すること、標準化クエリに基づいて、第1のデータソースに特異的なクエリを生成すること、標準化クエリに基づいて、第2のデータソースに特異的なクエリを生成すること、第1のデータソースに特異的なクエリを実行させて、第1の電子データソースから第1のデータのセットを獲得すること、第2のデータソースに特異的なクエリを実行させて、第2の電子データソースから第2のデータのセットを獲得すること、及び/または第1及び第2のデータのセットを含む、集合データフレームを生成することを含む。第1及び第2のデータのセットは各々、プロセスプラント内の1つ以上のフィールドデバイスによって測定された情報を含む、複数のデータ点を含み得る。

【0036】

第1のデータソースに特異的なクエリは、第1の電子データソースと関連付けられる第1のクエリフォーマットを活用し得、第2のデータソースに特異的なクエリは、第2の電子データソースと関連付けられる第2のクエリフォーマットを活用し得る。標準化クエリは、第1の電子データソースの第1の指標と、第2の電子データソースの第2の指標とを含み得る。一部の実施形態においては、第1の電子データソースがリレーショナルデータベースであってもよく、第2の電子データソースが非リレーショナルデータベースであってもよい。第1の指標が第1のクエリフォーマットを特定してもよく、第2の指標が第2のクエリフォーマットを特定してもよい。一部の実施形態においては、第1及び第2の指標は、データ列の指標に対してプリペンドされ得る。

【0037】

標準化クエリフォーマットは、第1のクエリフォーマットのクエリ構文と、第2のクエリフォーマットのクエリ構文との両方と別個のクエリ構文を活用し得る。標準化クエリフォーマットのクエリ構文は、第1の電子データソースからの第1のデータのセット、または第2の電子データソースからの第2のデータのセットのいずれについても、獲得するように直接実行可能ではない場合がある。例えば、標準化クエリは、JSONファイルの1つ以上のオブジェクトまたは配列に含まれ得る。

【0038】

標準化クエリは、第1のデータセット及び第2のデータセットのデータエントリに対応する期間を特定する時間範囲の指標を含み得る。時間範囲は複数の期間を含んでもよく、これらの複数の期間は、データが照会されない除外期間によって分離されてもよい。時間範囲はまた、第1のデータセットと関連付けられる第1の期間と、第2のデータセットと関連付けられる第2の期間とを特定し得る。標準化クエリはまた、集合データフレームについてのサンプルレートの指標も含み得る。

【0039】

第1のデータセットは、第1のサンプルレートを有するデータを含み得、第2のデータセットは、第2のサンプルレートを有するデータを含み得る。そのような事例においては、集合データフレームを生成することは、第1及び第2のサンプルレートを整合することを含み得る。第1及び第2のサンプルレートを整合することは、所望のサンプルレートを達成するために必要とされる、サンプルされていない時間に対応するデータ点を追加する

10

20

30

40

50

ことを含み得る。そのような追加されるデータ点は、追加されるデータ点に対して時間的に直前の、サンプル済みのデータの所与の値であり得る。第1及び第2のデータサンプルレートを整合することは、サンプル済みのデータ点を取り除くことを同様に含み得る。

【0040】

そのような方法を実践するためのシステムは更に、1つ以上のデータソースと通信可能に連結される1つ以上のプロセッサと、1つ以上のプロセッサによって実行されたとき、コンピュータシステムに、上記のシステム機能の一部または全てを実施させるというコンピュータ可読命令を記憶する1つ以上のプログラムメモリとを含み得る。コンピュータ可読媒体が、コンピュータシステムの1つ以上のプロセッサによって実行可能な、類似のコンピュータ可読命令を記憶して、それによってコンピュータシステムに上記のシステム機能の一部または全てを実施させてもよい。本方法、システム、またはコンピュータ可読媒体の追加的または代替的態様が、下の詳細な説明においてより完全に記載されるように実践され得る。

10

【0041】

加えて、新規の、周波数分析の分析技法が開示され、本明細書に記載されるシステム、方法、装置、及び技法のいずれかによって提供され得る。新規の、周波数分析の分析技法は、ストリーミングされるデータを分析して、プロセスプラントまたはプロセス制御システムにおける早期警戒故障検出を提供することができる。より具体的には、この周波数分析の分析技法は、故障、異常、パフォーマンスの低下、ターゲットパフォーマンスレベル、望まれない状態、及び/または所望の状態の、特定された先行インジケータに対応するプロセス変数の新しいセットを作成することができ、ストリーミングされるプロセスデータにローリングFFTを実施することで、新しいプロセス変数の時系列データを決定することができる。ローリングFFTは、ストリーミングされるプロセスデータを時間領域から周波数領域に変換することができ、ここで、プロセス変数の新しいセットの値が決定され得る。新しいプロセス変数の決定された値は、監視のために、時間領域へと再度転換され得る。監視される時間領域データ内における先行インジケータの存在の検出は、予測される故障、異常、パフォーマンスの低下、及び/または他の望まれない状態の兆候を生成及び/または存在させ得、プロセスプラントのオペレーションに対する変化をもたらし、それによって、望ましくない不調、事象、及び/または状態の影響を回避、防止、及び/または軽減し得る。同様に、監視される時間領域データ内における先行インジケータの存在の検出は、パフォーマンスのターゲットレベル等の予測される所望の状態の兆候を、一部の場合においては、所望の状態の発生が予測される時間または時間間隔と共に、生成及び/または存在させ得る。

20

30

【図面の簡単な説明】

【0042】

【図1】分散型の、埋込式工業プロセスパフォーマンス監視及び/または分析エンジンを含む、例示的システムプロセスプラントまたはプロセス制御システムを例証するブロック図である。

【図2A】分散型工業プロセスパフォーマンス監視/分析用の例示的システムのブロック図である。

40

【図2B】例示的プロセス制御システムに関連する、例示的分散型工業プロセスパフォーマンス監視分析システムの配列の実施形態である。

【図3】例示的工業プロセスパフォーマンス監視及び/または分析エンジンのブロック図である。

【図4A】様々なデータブロックの特色及び相互接続を例証する、例示的データダイアグラムである。

【図4B】例示的なData Analytics Studioのユーザインターフェイスである。

【図4C】データブロック定義ライブラリ、データブロック定義、データモジュール、データブロック、データブロックインスタンス、及びワイヤ間の例示的關係性である。

50

- 【図 4 D】データモジュールの下書きまたは作成の例である。
- 【図 4 E】データモジュールが作成される、Data Studio キャンパスのズームイン領域である。
- 【図 4 F】異なる種類のデータブロック用の、例示的データブロック定義テンプレートである。
- 【図 4 G】異なる種類のデータブロック用の、例示的データブロック定義テンプレートである。
- 【図 4 H】異なる種類のデータブロック用の、例示的データブロック定義テンプレートである。
- 【図 4 I】データダイアグラムの評価、編集、または展開の例示的フローである。 10
- 【図 4 J】標準及びカスタムビジュアライゼーションの呈示を支持する例示的アーキテクチャである。
- 【図 4 K】複合データブロックが作成される例示的シナリオである。
- 【図 4 L】例示的オフラインデータダイアグラムである
- 【図 4 M】図 4 L の例示的オフラインデータダイアグラムの転換から生成された、オンラインデータダイアグラムである。
- 【図 4 N - 1】例示的な、Data Analytics Dashboard のユーザーインターフェイスである。
- 【図 4 N - 2】例示的な、Data Analytics Dashboard のユーザーインターフェイスである。 20
- 【図 4 O】プロセスプラントのパフォーマンスを制御及び最適化する制御ループの、高次ブロックダイアグラムである。
- 【図 4 P】ローカライズしたデータ分析サービスを提供するための例示的方法である。
- 【図 4 Q】ローカライズしたデータ分析サービスを提供するための例示的方法である。
- 【図 5 A】本発明による Analytics Service の例示的アーキテクチャのブロック図である。
- 【図 5 B】図 5 A に描写したアーキテクチャの例示的データサービスエンティティをより詳細に例証するブロック図である。
- 【図 5 C】現在説明している実施形態による、プログラミングキャンパスに配置されるブロックのプロパティダイアログを呈示するための方法を例証するフローチャートである。 30
- 【図 5 D】例示的オフライン図がプログラミングキャンパスに構成される、例示的プログラミングキャンパスである。
- 【図 5 E】図 5 D のオフライン図に対応する例示的オンライン図である。
- 【図 5 F】モジュールの例示的プロパティダイアログである。
- 【図 5 G】プロパティがオフライン及びオンライン運転の両方に関する、オフラインモジュールのブロックの例示的プロパティダイアログである。
- 【図 6 A】例示的な分析的クエリシステムのブロック図である。
- 【図 6 B】標準化クエリを用いる、例示的な分析構成のブロック図である。
- 【図 6 C】例示的標準化クエリである。
- 【図 6 D】例示的標準化クエリ方法の流れ図である。 40
- 【図 6 E】例示的標準化クエリ方法の流れ図である。
- 【図 7 A】ある時間間隔中の、精製所のフレアシステムの例示的プロセスフロー測定値のグラフである。
- 【図 7 B】別の時間間隔中の、図 7 A の例示的プロセスフロー測定値のグラフである。
- 【図 7 C】データにフレア事象を含む、精製所の例示的プロセスフローデータのグラフである。
- 【図 7 D】図 7 C のプロセスフローデータにローリング FFT を適用して決定された時系列データのグラフである。
- 【図 7 E】フレア事象周囲の図 7 D の周波数領域データを分析するために、PCA を用いた結果のグラフである。 50

【図7F】別のフレア事象周囲の図7Dの周波数領域データを分析するために、PCAを用いた結果のグラフである。

【図7G】プロセスプラントにおいて、早期故障検出を提供するための例示的方法の流れ図である。

【発明を実施するための形態】

【0043】

分散型工業パフォーマンス監視及び分析用の技法、システム、装置、構成要素、及び方法が本明細書に開示される。概して、該新規パフォーマンス監視及び分析技法は、1つ以上の工業プロセスをリアルタイムで制御するように運転する、工業プロセスプラント、プロセス制御環境、及び/またはプロセス制御システムについての知識の発見及びアクション可能な知識を提供する。典型的には、そのようなプロセスプラントまたは制御システムによって実行される工業プロセスは、その中で制御されて、それによって物理的物質または生産物を製造、精製、変形、生成、または生産する。そのようなプロセスプラントまたは制御システムの例としては、製油所、製紙工場、化学的製造、薬学的製造、食品加工、及び配送等が挙げられる。

10

【0044】

どのプロセス制御システム、プラント、または環境においても、最も重要な関心事はその安全性、信頼性、及び経済性である。無制御のプロセス、故障、破損、及び/または防止できる人的過誤は、爆発、火事、危険な化学物質の放出、環境への被害、設備の損失、及び/または人命損失につながり得るため、どのプロセス制御システムまたはプラントにとっても、運転の安全性は特に重要である。設備及びプロセス運転の信頼性は、プラントの経済性及び収益性を維持及び/または向上するために重要である。更に、プラントの経済性及び収益性は、プラントの運転を調整してパフォーマンスを向上させることによって改善できる。

20

【0045】

プロセス制御システムで起こる故障は、その安全性、信頼性、及び/または経済性に影響を及ぼし得る。故障は、複数の事象が組み合わさって、供給原料組成の未知の変化、設備の劣化、設備の破損、及び異常（あるいは誤った）ユーザの運転アクション等、プロセス制御システムが対応するようには設計されていない状況が形成されたときに概して発生する。故障の他の例としては、計画外の設備の操業停止、化学物質の大気中への放出、圧力安全バルブのリフティング、設備の温度暴走、及びパフォーマンス劣化が挙げられる。無論、故障は全体的または大規模事象に限定されるものではなく、故障は、経済的影響、安全性への影響、及び/または環境影響を有するか、それらにつながり得る、任意の異常事象または事象の組み合わせであり得る。

30

【0046】

典型的には、故障が検出されたとき（例えば、プロセス制御システムによって自動的に検出されたとき）、オペレータインターフェイスにはアラームが発生する。次いで、オペレータは、その故障の根本原因の診断及び修正アクションを取ることを試みる。したがって、故障を管理する際の重要な因子の一部としては、早期の故障の検出及び誤報の低減が挙げられ、これには例えば、故障の適時及び確実な検出、故障の根本原因の診断、及び故障の源である計器、コントローラ、プロセス、及び/または設備を通常運転に戻す修正アクションを実行することが含まれ得る。

40

【0047】

公知のデータ監視及び分析ツールは、プロセスプラントを通常運転領域に保ち、故障の発生に対する第1の防衛線を提供することを試みている。だが、そのようなツールは、典型的には、プロセスプラント内の単一ループまたは非常に限定された範囲にしか注意を向けていない。それらのツールには、何百何千のプロセス変数が非常に速い速度で観察される典型的な近代的プロセス制御プラントの全体を（あるいは、その大部分ですら）処理することはできない。更に、公知のデータ監視及び分析ツールは、各ツールの直近の限定された範囲の外側の状況によって引き起こされた、劣化した設備のパフォーマンス、撤去さ

50

れたフィールド設備、または運転状態における大きな変化といった特別な事象を処理することができない。

【 0 0 4 8 】

一方で、本明細書に開示される新規の分散型工業プロセスパフォーマンス監視及び分析技法、システム、装置、構成要素、及び方法は、プロセスプラント全体から単一ループ、あるいは単一のデバイスに至るまで、任意のサイズの範囲の工業プロセス監視及び分析を処理（例えば、監視及び/または分析）することができる。実際、一部の構成においては、本明細書に開示される新規技法は、複数のプロセスプラント（例えば、データが入手可能である場合、単一の企業によって、あるいは異なる企業によって保有及び運転されている複数の製油所）に同時に対処することができる。概して、本明細書に開示される新規パフォーマンス監視及び分析技法は、早期検出を示す知識、及び/またはプロセスプラント及び制御システム内で起こり得る潜在的な故障の事前の警告を発見及び提供し、それによって、その故障の発生を防止するための規定的アクションまたは修正アクションを取るのに十分な時間が確保される。一部の状況においては、本明細書に開示される新規技法はまた、潜在的な故障の発生を防止するため、及び/またはそれらの発生の影響を制限するための、規定的でアクション可能な知識を発見及び提供する。更に、本明細書に開示される新規技法は、プラントの効率性に対する潜在的な改善を示す知識を発見または提供し、ならびにその効率性の改善を実現するためのアクション可能な知識を発見または提供する。

10

【 0 0 4 9 】

分散型工業プロセス監視及び分析を有する例示的プロセス制御システム

20

上で考察したように、本明細書に記載される新規の分散型工業プロセス監視及び分析技法の少なくとも一部を含むかあるいは支持する、プロセスプラント、プロセス制御システム、またはプロセス制御環境は、1つ以上の工業プロセスをリアルタイムで制御するように運転する。したがって、本プロセスプラントまたは制御システムは、その中に、物理的機能（バルブの開放または閉鎖、温度、圧力、ならびに/または他のプロセスパラメータ及び/もしくは環境パラメータの測定等）を実行して、プロセスプラントまたはシステム内で実行中のプロセスを制御する、1つ以上の有線もしくは無線プロセス制御デバイス、構成要素、または要素を含み得る。本プロセスプラントまたはプロセス制御システムは、例えば、1つ以上の有線通信ネットワーク及び/または1つ以上の無線通信ネットワークを含み得る。本プロセスプラントまたは制御システムは、連続データベース、バッチデータベース、及び他の種類のヒストリアンデータベース等の集中データベースを含み得る。

30

【 0 0 5 0 】

例証のため、図1は、本明細書に記載される分散型工業プロセス監視及び分析技法のいずれかまたは全てを含むかあるいは支持する、例示的プロセスプラントまたはプロセス制御環境5の詳細なブロック図である。プロセス制御システム5は、プロセス制御プラントまたは環境5に含まれるか、それと統合されるか、あるいはそれによって支持される、分散型工業プロセス監視及び分析システムの複数の分散型データエンジンを含む。（分散型工業プロセス監視及び分析システムについてのより完全な説明は、後の節において提供される。）工業プロセス監視及び分析システムの分散型データエンジンの各々は、プロセスプラント5のある構成要素または部分と、直接的または間接的に接続される（例えば、物理的に接続されるか、無線で接続される）。例えば、分散型データエンジンは、一元的なエンティティを形成するようにプロセスプラント5の特定のデバイスまたはノードに埋め込まれるかまたはその中に製造されてもよく、データエンジンは、プラント5の特定のデバイスまたはノードに接続または連結されてもよく、あるいはデータエンジンは、プラント5の従来型の通信リンクに接続または連結されてもよい。図1においては、丸で囲んだ「DDE」が、それぞれの例示的分散型データエンジンを表す。

40

【 0 0 5 1 】

先に考察したように、分散型データエンジンは、自動的にプロセス制御データを生成及び/または受信して、プロセスプラント環境5においてプロセスをリアルタイムで制御する機能を実行することを主な機能とする、プロセス制御デバイス内に埋め込まれてもよい

50

。例えば、それぞれのデータエンジンは、プロセスコントローラ、フィールドデバイス、及びI/Oデバイス内に埋め込まれてもよく、あるいはその中に製造されてもよい。プロセスプラント環境5においては、プロセスコントローラは、フィールドデバイスによって生み出されたプロセス測定値を示す信号を受信し、この情報を処理して制御ルーチンを実行し、制御信号を生成し、これらの制御信号が、従来型の有線または無線プロセス制御通信リンクまたはネットワークを通じて他のフィールドデバイスに送信されて、プラント5内のプロセスの運転を制御する。典型的には、少なくとも1つのフィールドデバイスが物理的機能を実行して（例えば、バルブの開放または閉鎖、温度の増加または減少等）、プロセスの運転を制御し、一部の種類のフィールドデバイスが、I/Oデバイスを用いることでコントローラと通信する。プロセスコントローラ、フィールドデバイス、及びI/Oデバイス10は有線であっても無線であってもよく、任意の数及び組み合わせの、有線及び無線のプロセスコントローラ、フィールドデバイス、及びI/Oデバイスがプロセスプラント環境またはシステム5内に含まれてもよく、それぞれの分散型データエンジンを各々含んでもよい。

【0052】

例えば、図1は、プロセスコントローラ11を例証し、このプロセスコントローラは、入力/出力(I/O)カード26及び28を介して、有線フィールドデバイス15~22と通信可能に接続され、無線ゲートウェイ35及びプロセス制御データハイウェイまたはバックボーン10（1つ以上の有線及び/または無線通信リンクを含んでもよく、例えばEthernet10 20プロトコル等の、任意の所望の、または好適な、または通信プロトコルを用いて実装され得る）を介して、無線フィールドデバイス40~46と通信可能に接続される。ある実施形態において、コントローラ11は、1つ以上の通信プロトコル、例えば、Wi-Fiまたは他のIEEE 802.11に準拠する無線ローカルエリアネットワークプロトコル、移動体通信プロトコル（例えば、WiMAX、LTE、もしくは他のITU-R対応プロトコル）、Bluetooth（登録商標）、HART（登録商標）、Wireless HART（登録商標）、Profibus、FOUNDATION（登録商標）Fieldbus等をサポートする、任意の数の他の有線または無線通信リンクの使用等によって、バックボーン10以外の1つ以上の通信ネットワークを用いて無線ゲートウェイ35と通信可能に接続される。本明細書で言及する場合、バックボーン10及びこれらの他の通信ネットワークは、「従来型の」プロセス制御通信ネットワークの例30である。

【0053】

コントローラ11は、例として、Emerson Process Managementによって販売されている、DeltaV（商標）コントローラであり得るが、フィールドデバイス15~22及び40~46のうち少なくとも一部を用いて、バッチプロセスまたは連続プロセスを実行するように運転してもよい。ある実施形態においては、プロセス制御データハイウェイ10に対して通信可能に接続されるのに加えて、コントローラ11はまた、例えば標準的な4~20mAデバイス、I/Oカード26、28、及び/またはFOUNDATION（登録商標）Fieldbusプロトコル、HART（登録商標）プロトコル、Wireless HART（登録商標）プロトコル等の任意のスマート40通信プロトコルと関連する、任意の所望のハードウェア及びソフトウェアを用いて、フィールドデバイス15~22及び40~46のうち少なくとも一部とも通信可能に接続される。図1においては、コントローラ11、フィールドデバイス15~22、及びI/Oカード26、28は有線デバイスであり、フィールドデバイス40~46は無線フィールドデバイスである。無論、有線フィールドデバイス15~22及び無線フィールドデバイス40~46は、将来的に開発される任意の規格またはプロトコルを含む、任意の有線または無線プロトコル等の、任意の他の所望の規格（複数可）またはプロトコルに準拠してもよい。

【0054】

図1のプロセスコントローラ11は、1つ以上のプロセス制御ルーチン38（例えば、50

メモリ32内に記憶されている)を実行または監督するプロセッサ30を含む。プロセッサ30は、フィールドデバイス15~22及び40~46、ならびにコントローラ11と通信可能に接続されている他のノードと通信するように構成される。本明細書に記載される任意の制御ルーチンまたはモジュール(品質予測及び故障検出モジュールまたは機能ブロック)は、そのように所望される場合は、その一部を異なるコントローラまたは他のデバイスによって実装または実行させてもよいことに留意されたい。同様に、プロセス制御システム5内で実装される、本明細書に記載される制御ルーチンまたはモジュール38は、ソフトウェア、ファームウェア、ハードウェア等を含む、任意の形態をとり得る。制御ルーチンは、オブジェクト指向プログラミング、ラダーロジック、シーケンシャルファンクションチャート、ファンクションブロックダイアグラムを用いるか、あるいは任意の他のソフトウェアプログラミング言語またはデザイン方式を用いること等によって、任意の所望のソフトウェアのフォーマットで実装できる。制御ルーチン38は、ランダムアクセスメモリー(RAM)または読み取り専用メモリー(ROM)等の、任意の所望の種類のメモリ32に記憶され得る。同様に、制御ルーチン38は、例えば1つ以上のEPROM、EEPROM、特定用途向け集積回路(ASIC)、または任意の他のハードウェアもしくはファームウェア要素にハードコードされてもよい。したがって、コントローラ11は、任意の所望の様式で制御ストラテジまたは制御ルーチンを実装するように構成することができる。

【0055】

一部の実施形態において、コントローラ11は、一般的に機能ブロックと呼ばれるものを用いて制御ストラテジを実行し、各機能ブロックは、全体的な制御ルーチンのオブジェクトまたは他の部分(例えば、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共に動作して、プロセス制御システム5内でプロセス制御ループを実行する。制御ベースの機能ブロックは、典型的には、送信器、センサ、もしくは他のプロセスパラメータ測定デバイスと関連するもの等の入力機能、PID、ファジー論理等の制御を実行する制御ルーチンと関連するもの等の制御機能、またはバルブ等のあるデバイスの運転を制御して、プロセス制御システム5内で物理的機能を実行する出力機能のうちの1つを実行する。無論、ハイブリッド及び他の種類の機能ブロックが存在する。機能ブロックはコントローラ11内に記憶され、それによって実行されてもよく、これは典型的には、これらの機能ブロックが標準的な4~20mAデバイス及びHART(登録商標)デバイス等のある種のスマートフィールドデバイス用に使用されるかあるいはそれと関連するときに成り立ち、あるいは機能ブロックは、フィールドデバイスそのものの内部に記憶され、それによって実装されてもよく、これはFOUNDATION(登録商標)Fieldbusデバイスの場合に成り立ち得る。コントローラ11は、1つ以上の制御ループを実装し得、機能ブロックのうちの1つ以上を実行することで実行され得る、1つ以上の制御ルーチン38を含んでもよい。

【0056】

有線フィールドデバイス15~22は、センサ、バルブ、送信器、ポジショナ等の任意の種類デバイスであり得、一方でI/Oカード26及び28は、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する、任意の種類デバイスであり得る。図1においては、フィールドデバイス15~18は、アナログ回線またはアナログデジタル結合回線を通じてI/Oカード26と通信する、標準的な4~20mAデバイスまたはHART(登録商標)デバイスであり、一方で、フィールドデバイス19~22は、FOUNDATION(登録商標)Fieldbus通信プロトコルを用いて、デジタルバスを通じてI/Oカード28と通信する、FOUNDATION(登録商標)Fieldbusフィールドデバイス等のスマートデバイスである。しかし、一部の実施形態においては、有線フィールドデバイス15、16、及び18~21のうちの少なくとも一部、ならびに/またはビッグデータI/Oカード26、28のうちの少なくとも一部は、プロセス制御データハイウェイ10を用いて、かつ/あるいは他の好適な制御システムプロトコル(例えば、Profibus、DeviceNet、Foundation Fieldbus

10

20

30

40

50

、ControlNet、Modbus、HART等)を用いることで、コントローラ11(及び/または他のビッグデータノード)と追加的または代替的に通信する。

【0057】

図1に示されるように、コントローラ11、I/Oカード26及び28、ならびに有線フィールドデバイス15~16、18~21の各々は、丸で囲んだ「DDE」で示される、それぞれの埋込式分散型データエンジンを含み、これは、1つ以上のデータ分析通信チャネル及び/またはネットワーク(図1には図示せず)を介して、他の分散型データエンジンと通信する。

【0058】

図1においては、無線フィールドデバイス40~46は、WirelessHART(登録商標)プロトコル等の無線プロトコルを用いて、従来型の無線プロセス制御通信ネットワーク70を介して通信する。そのような無線フィールドデバイス40~46は、1つ以上の他のデバイス、または(例えば、この無線プロトコルもしくは別の無線プロトコルを用いて)無線で通信するようにも構成されている、プロセス制御データ分析通信ネットワーク112のノードと直接通信し得る。無線で通信するようには構成されていない1つ以上の他のノードと通信するために、無線フィールドデバイス40~46は、プロセス制御データハイウェイ10または別の従来型のプロセス制御通信ネットワークに接続される無線ゲートウェイ35を活用してもよい。

【0059】

無線ゲートウェイ35は、無線通信ネットワーク70の様々な無線デバイス40~58へのアクセスを提供し得る。具体的には、無線ゲートウェイ35は、無線デバイス40~58、有線デバイス11~28、及び/またはプロセス制御プラント5の他のノードもしくはデバイスの間における通信連結を提供する。例えば、無線ゲートウェイ35は、プロセス制御データハイウェイ10を用いることで、かつ/あるいはプロセスプラント5の、1つ以上の他の従来型の通信ネットワークを用いることで、通信連結を提供することができる。

【0060】

有線フィールドデバイス15~22と同様に、無線ネットワーク70の無線フィールドデバイス40~46は、プロセスプラント5内で、物理的制御機能、例えば、バルブの開放もしくは閉鎖、またはプロセスパラメータの測定値の取得を実行することができる。しかしながら、無線フィールドデバイス40~46は、ネットワーク70の無線プロトコルを用いて通信するように構成される。したがって、無線ネットワーク70の、無線フィールドデバイス40~46、無線ゲートウェイ35、及び他の無線ノード52~58は、無線通信パケットの生産者であり、かつ消費者である。

【0061】

一部のシナリオにおいては、無線ネットワーク70は非無線デバイスを含んでもよい。例えば、図1のフィールドデバイス48は、レガシである4~20mAデバイスであってもよく、フィールドデバイス50は、従来型の有線HART(登録商標)デバイスであってもよい。ネットワーク70内で通信するために、フィールドデバイス48及び50は、無線アダプタを介して無線通信ネットワーク70と接続されるか、あるいはこの無線アダプタまたは52bとヒストライズしてもよい。無線アダプタ52a、52bは、Foundation(登録商標)Fieldbus、PROFIBUS、DeviceNet等の他の通信プロトコルをサポートし得る。更に、無線ネットワーク70は、1つ以上のネットワークアクセスポイント55a、55bを含んでもよく、これらは、無線ゲートウェイ35と有線で通信する別個の物理的デバイスであってもよいし、あるいは無線ゲートウェイ35と共に一体型デバイスとして提供されてもよい。無線ネットワーク70はまた、無線通信ネットワーク70内のある1つの無線デバイスから別の無線デバイスにパケットを転送するために、1つ以上のルータ58を含んでもよい。無線デバイス40~46及び52~58は、無線通信ネットワーク70の無線リンク60を通じて、かつ/あるいはプロセス制御データハイウェイ10を介して、互いと、かつ無線ゲートウェイ35と通信し

10

20

30

40

50

得る。

【0062】

図1に示されるように、アダプタ52a、アクセスポイント55a、及びルータ58の各々は、丸で囲んだ「DDE」で示される、それぞれの埋込式分散型データエンジンを含み、これは、1つ以上のデータ分析通信チャネル及び/またはネットワーク(図1には図示せず)を介して、他の分散型データエンジンと通信する。

【0063】

一部の実施形態においては、プロセス制御システム5は、Wi-Fiまたは他のIEEE 802.11に準拠する無線ローカルエリアネットワークプロトコル、WiMAX(Worldwide Interoperability for Microwave Access)、LTE(Long Term Evolution)、もしくは他のITU-R(International Telecommunication Union Radio communication Sector)対応プロトコル等の移動体通信プロトコル、近距離無線通信(NFC)及びBluetooth等の短波長無線通信、または他の無線通信プロトコルといった、他の無線プロトコルを用いて通信する1つ以上の他の無線アクセスポイント72を含む。典型的には、そのような無線アクセスポイント72は、無線ネットワーク70とは異なり、かつ無線ネットワーク70とは異なる無線プロトコルをサポートする、それぞれの従来型無線プロセス制御通信ネットワークを通じた、手持ち式または他の携帯用コンピューティングデバイス(例えば、ユーザインターフェイスデバイス73)による通信を可能にする。例えば、無線または携帯用ユーザインターフェイスデバイス73は、プロセスプラント5内のオペレータによって活用される、移動式ワークステーションまたは診断試験設備であってもよい。一部のシナリオにおいては、携帯用コンピューティングデバイスに加えて、1つ以上のプロセス制御デバイス(例えば、コントローラ11、フィールドデバイス15~22、または無線デバイス35、40~58)もまた、アクセスポイント72によってサポートされる無線プロトコルを用いて通信する。

【0064】

一部の実施形態においては、プロセス制御システム5は、直近のプロセス制御システム5の外側にあるシステムへの1つ以上のゲートウェイ75、78を含む。典型的には、そのようなシステムは、プロセス制御システム5によって生成された情報、またはプロセス制御システム5の運転が基づく情報の需要者または供給元である。例えば、プロセス制御プラント5は、直近のプロセスプラント5を別のプロセスプラントと通信可能に接続するために、ゲートウェイノード75を含んでもよい。追加的または代替的に、プロセス制御プラント5は、実験室システム(例えば、実験室情報管理システムまたはLIMS)、オペレータラウンドデータベース、マテリアルハンドリングシステム、保全管理システム、製品在庫制御システム、生産スケジューリングシステム、気象データシステム、出荷及び取扱システム、包装システム、インターネット、別のプロバイダのプロセス制御システム、または他の外部システム等の、外部のパブリックシステムまたはプライベートシステムと直近のプロセスプラント5を通信可能に接続するように、ゲートウェイノード78を含んでもよい。

【0065】

図1に示されるように、アクセスポイント72、ユーザインターフェイスデバイス73、及びゲートウェイ75は、丸で囲んだ「DDE」で示される、それぞれの埋込式分散型データエンジンを各々含み、これは、1つ以上のデータ分析通信チャネル及び/またはネットワーク(図1には図示せず)を介して、他の分散型データエンジンと通信する。

【0066】

図1は有限数のフィールドデバイス15~22及び40~46、無線ゲートウェイ35、無線アダプタ52、アクセスポイント55、ルータ58、ならびにプロセスプラント5内に含まれるプロセス制御通信ネットワーク70と共に単一のコントローラ11を例証するのみであるが、これは例証的かつ非限定的実施形態であるに過ぎないことが留意される

10

20

30

40

50

。任意の数のコントローラ 11 がプロセス制御プラントまたはシステム 5 内に含まれてもよく、コントローラ 11 のうちのいずれが、任意の数の有線または無線デバイス及びネットワーク 15 ~ 22、40 ~ 46、35、52、55、58、及び 70 と通信して、プラント 5 内でのプロセスを制御してもよい。

【0067】

例示的な分散型工業プロセスパフォーマンス監視 / 分析システムのアーキテクチャ

図 2 A は、分散型工業プロセスパフォーマンス監視 / 分析用の例示的システム 100 のブロック図を含み、このシステムは本明細書においては交換可能に Data Analytics System (DAS) とも呼ばれる。DAS 100 は、例えば図 1 のプロセスプラント 5 と共に運転してもよく、例証の簡便性のために本明細書ではそのように考察する。しかしながら、DAS 100 の少なくとも一部分は、プロセスプラント 5 以外の他のプロセスプラント及び / またはプロセス制御システムと共に運転してもよいことが理解される。

【0068】

概して、DAS 100 はローカライズしたパフォーマンス監視及び / または分析をサポートするが、一方で同時に、プロセスプラント環境に関する、大規模 (例えば、システム全体及び / またはプロセスプラント 5 の複数のデバイスもしくはノードにわたって) パフォーマンス監視、データマイニング、及びデータ分析をサポートする。このため、システム 100 は複数の分散型データエンジン (DDE) を含み、その例は、図 2 A においては参照番号 102 a ~ 102 e で示され、図 2 B においては参照番号 102 f ~ 102 h で示される。本明細書で使用する場合、参照番号「102 x」は、DDE 102 a ~ 102 h のうちのいずれか 1 つ以上を指す。図 2 A に例証される分散型データエンジンのうちの少なくとも一部は、図 1 に例証される分散型データエンジンに対応する。例えば、図 2 A のデータエンジン 102 a は、図 1 の無線ゲートウェイ 35 内に含まれるデータエンジンであり、図 2 A のデータエンジン 102 b は、図 1 のコントローラ 11 に埋め込まれたデータエンジンである。加えて、システム 100 は、図 1 には示されていない他の分散型データエンジン 102 を含む。例えば、分散型データエンジン 102 c はデータハイウェイ 10 に接続されており、分散型データエンジン 102 d はプロセスプラント 5 の集中ビッグデータアライアンス 108 に埋め込まれており、分散型データエンジン 102 e はビッグデータクラウドノード 110 に埋め込まれており、ここでビッグデータクラウドノード 110 は、プロセス制御プラント 5 のデータ分析の必要性を処理することができる (また一部の構成においては、他のプロセス制御プラントを処理することもできる)。無論、システム 100 は 5 つのデータエンジン 102 a ~ 102 e、または 8 つのデータエンジン 102 a ~ 102 h に限定されず、任意の数の分散型データエンジンを含むことができ、これらのデータエンジンのうちの少なくとも一部は、それらのそれぞれのデータソース内に埋め込まれるかまたはその中に製造され (例えば、プロセスプラント 5 のそれぞれのプロセス制御デバイスに)、かつ / あるいはこれらのデータエンジンのうちの少なくとも一部は、プロセスプラント 5 のある他のデータソース (例えば、構成要素、部分等) に別様に接続される。

【0069】

一部の構成においては、一部の分散型データエンジン 102 x は従来型のプロセス制御通信ネットワークを用いることで他のデータエンジンと少なくとも一部の情報を通信し得るが、上述の通り、典型的には、分散型データエンジンは、従来型のプロセス制御プラント通信ネットワーク (例えば、図 1 のバックボーン 10、無線ネットワーク 70、I/O カード 28 とデバイス 19 ~ 22 との間の有線接続等) のうちのいずれを使用しても他の分散型データエンジンと通信することはない。しかし、概して、データエンジン 102 x は、従来型のプロセス制御ネットワークとは異なる、別個の 1 つ以上のデータ分析通信ネットワーク 112 を用いることで、他のデータエンジン 102 x と通信する。分析データは、例えば、分析ストリーミングサービス、ストリーミング及び / またはキューイングプロトコルの使用、かつ / あるいは、カスタムストリーミングソース、Flume、HDF

10

20

30

40

50

S、ZeroMQ、Kafka、Microsoft Message Bus、MQTT、AMQP、RabbitMQ等のストリーミングをサポートするメッセージングブローカーまたはシステム115の使用、前述の「STREAMING DATA FOR ANALYTICS IN PROCESS CONTROL SYSTEMS」と題される米国出願第14/506,863号に記載されるもの等の、カスタム特化履歴オブジェクト通信プロトコル(custom specialized history object communications protocol)の使用、または別の好適な通信プロトコルの使用によって、分散型データエンジン間またはネットワーク112のノード間でストリーミングされる。図2Aにおいて、ネットワーク112は、明瞭性のために完全には描写されていないが、データ分析通信ネットワーク112のいくつかの分岐が図2Aには表されている。したがって、ある実施形態においては、データ分析通信ネットワーク112の少なくとも一部分は、ある意味において、プラント5内の従来型のプロセス制御通信ネットワークの上のオーバーレイネットワークであり得る。例えば、DDE102xを備える物理的設備、及びデータ分析ネットワーク112の少なくとも一部は、プロセスプラント5の構内に配置されてもよく、一部の場においては、プロセスプラント5の設備の中に配置されてもよい。

10

【0070】

別の実施形態においては、データ分析通信ネットワーク112の少なくとも一部分は、図2Bに示されるように、従来型のプロセス制御通信ネットワークと並んで、あるいはほぼそれと並列して実装され得る。図2Bにおいては、プロセスプラント5は、例えばプラント5内の1つ以上のプロセスの制御を補助するために、従来型のプロセス制御通信ネットワークと通信可能に接続される、様々なプロセス制御デバイスまたは構成要素を含む。そのようなプロセス制御デバイス/構成要素のうち3つは、それらの中に埋め込まれており、それらと関連付けられているそれぞれのDDE102f、102g、及び102hを各々有し、DDE102f~102hの各々は、データ分析ネットワーク112と通信可能に接続されている。しかしながら、データ分析ネットワーク112をサポート及び/または構成する設備の大多数は、プロセスプラント5の設備内には配置されておらず、代わりにエンタープライズデータセンタに離れて配置され得る。

20

【0071】

別の実施形態においては、データ分析通信ネットワーク112の少なくとも一部分は、プロセスプラント5内の論理ネットワークとして実装されてもよい。この実施形態においては、例えば、プロセス制御データ及び分析データの両方が、論理的には独立したリンク、例えば1つの従来型のプロセス通信リンク及び1つのデータ分析通信リンクの外観を有する、同じ物理的通信リンクを通じて伝送されてもよい。

30

【0072】

しかし、データ分析ネットワーク112の実装形態に関係なく、データ分析ネットワーク112は、プロセス制御システム5内のデバイスまたはノードにおいて従来型のプロセス制御通信ネットワークと交差し、これらのデバイスまたはノードには、分散型データエンジン(例えば、データエンジン102a、102b、102f、102g、102h)が埋め込まれており、プロセス制御システム5内の他のデバイスまたはノードと、従来型のプロセス制御通信ネットワークを介して、かつ/あるいは従来型のプロセス制御通信ネットワークに送信されるデータへのウィンドウを持つことを目的としてそれと接続される分散型データエンジン(例えば、データエンジン102c)によって情報を通信する。

40

【0073】

データ分析ネットワーク112は、任意の所望の通信技術及び/または通信プロトコルを活用してもよい。データ分析ネットワーク112の少なくとも一部分は有線技術を活用してもよく、データ分析ネットワーク112の少なくとも一部分は無線技術を活用してもよい。しかし、概して、データ分析ネットワーク112は、Ethernetよりも、IPまたは他の非同期データパケットプロトコル等のデータパケット伝送プロトコル及び/または技術をサポートし得る。

50

【 0 0 7 4 】

より高次または上位層においては、データ分析ネットワーク 1 1 2 は、ストリーミングが可能なプロトコル、リアルタイムデータフィードを扱うメッセージブローカー、及び/または Apache Kafka 等の、リアルタイムデータフィード 1 1 5 を扱うメッセージングシステムを活用してもよい。図 2 A に示されるように、データ分析ネットワーク 1 1 2 によって提供されるストリーミング能力 1 1 5 は、分散型パフォーマンス監視/分析システム、すなわち DAS 1 0 0 が、データソースにローカルでバインドすること、及びデータソース付近で様々な分析サービスを提供することを可能にする（例えば、分散型データエンジン 1 0 2 x が埋め込まれている、図 1 の 1 1、1 8、2 6、2 8、7 2、3 5、5 2 a、5 5 a、7 2 等の、プロセスプラント 5 のデバイスまたはノード付近、またはそこにおいて）。同時に、システム 1 0 0 のストリーミング能力 1 1 5 はまた、より広い範囲にはより高次のデータ分析をサポートするために必要な特定のソース生成データしか、受信側のデータエンジンに通信される必要がないため、システム 1 0 0 によるより大規模な予測及び最適化も可能にする。例えば、データエンジン 1 0 2 d が、プロセスプラント 5 のコントローラの各々によって生成され、かつ特定の構成変化が各コントローラにおいてインスタンス化された後 1 時間以内に取り込まれたデータのみに対して特定のデータ分析を実行する場合、コントローラ 1 1 に埋め込まれたデータエンジン 1 0 2 b は、必要とされるデータのみ、エンジン 1 0 2 d に対してストリーミングする（例えば、特定の構成変化がインスタンス化された後 1 時間の間にコントローラ 1 1 によって生成された出力データしかデータエンジン 1 0 2 d にストリーミングせず、コントローラ 1 1 によって生成された他の出力データはストリーミングしない）。

10

20

【 0 0 7 5 】

図 2 A に示されるように、分散型パフォーマンス監視/分析システムすなわち DAS 1 0 0 の一部は、プロセス制御環境またはプラント 5 内にローカルで配置される。例えば、分散型データエンジン 1 0 2 a、1 0 2 b、1 0 2 c、及び 1 0 2 d は、プロセスプラント 5 の構内に配置される。また、図 2 A に示されるように、システム 1 0 0 の他の部分は、離れて配置される。例えば、データエンジン 1 0 2 e は、クラウド 1 1 0 に配置される。他のデータエンジン 1 0 2 x は、複数のプロセスプラントまたはプロセス制御システムを処理するサーババンク等、1 つ以上の離れた場所（図示せず）に配置されてもよい。無論、一部の構成においては、システム 1 0 0 は、ローカルで配置されたデータエンジンを省いてもよく、離れて配置されたデータエンジンを省いてもよい。すなわち、システム 1 0 0 は、完全にサブジェクトプロセス制御環境 5 の構内に（例えば、ローカルで）、完全にサブジェクトプロセス制御環境 5 の構外に（例えば、離れて）、あるいは構内及び構外の分散型データエンジンの組み合わせを用いて（例えば、ローカルで、かつ離れて）、パフォーマンス監視及び分析を（例えば、1 つ以上の分散型データエンジンを介して）提供してもよい。

30

【 0 0 7 6 】

工業制御システム用のパフォーマンス監視及び分析の例示の種類

概して、工業制御システムのパフォーマンス監視及び分析は、サブジェクト制御システムによって、（例えば、システムが動作して 1 つまたはプロセスを制御している間に、リアルタイムで）生成されたデータを集めることと、1 つ以上の分析機能もしくはオペレーションを、集めたデータの少なくとも一部に実施して、制御システムがどの程度良好に機能しているかについての知識を決定すること、または一部の場合においては、システムのパフォーマンスを向上するために取り得るアクションを決定することを含む。工業プロセス制御システムのデータ分析は、3 つの一般的なカテゴリ、すなわち記述的分析、予測的分析、及び規定的分析において考慮され得る。これらのカテゴリの分析についての一般的な考察が後に続く。

40

【 0 0 7 7 】

記述的分析は、プロセス制御システムまたはプラント内で何が起こったか、いつそれが起こったか、どのくらいの頻度でそれが起こったか、及び、起こったことからどのような

50

問題（複数可）がもたらされたかをユーザが発見することを可能にする。典型的には、記述的分析は、プロセスシステムまたはプラントを監視することで（例えば、帰納的に）収集されたデータに対して実施され、基本的計算、スケーリング、及び標準的統計等の技法を活用し得る。

【 0 0 7 8 】

規定的分析は、プロセス制御システムまたはプラント内のオペレーションをユーザが最適化することを可能にする。例えば、規定的分析は、何が最善の答えか？不確実性を前提とすると、何が最善の結果か？何が有意に異なり、何がより良好な選択肢か？といった質問にユーザが回答することを可能にする。予測的分析は、工業プロセス制御プラント及びシステムのプロセスオペレーションの重要な品質変数または重要なインジケータを特定、監視、及び制御することができる。加えて、予測的分析は、将来的な入力のセットまたは因果条件を与えられた場合、プロセスオペレーションの重要な品質変数または重要なインジケータに対して何が起こるかを特定することができる。次いで、予測値は、規定的分析によって活用されて、規定的アクションを生成し得る。典型的には、規定的分析は、制約に基づく最適化、及び多目的最適化等の技法を活用するが、公知のデータ監視及び分析ツールは初歩的であり、その範囲において著しく限定されている。

【 0 0 7 9 】

例えば、公知のデータ監視及び分析ツールによって、ユーザまたはオペレータは、プラントの設備またはユニットの単一の部分のパフォーマンスを、その設備またはユニットの単一の部分を監視することで（例えば、帰納的に）収集されたデータに基づいて動作する、制約に基づく最適化のツールを使用することで、最適化し得る。しかしながら、プロセス制御システムまたはプラントは設備及びプロセスユニットの多数の部分を含むため、全体的なプロセスは、複数のユニット間のプロセス相互作用に起因して、未だなお最適からは遠い可能性がある。プロセス制御システムの一部にまたがる、相互作用しているユニットまたは協調的最適化のための多目的最適化技法は、公知のデータ監視及び分析ツールにおいて利用可能であるが（例えば、第一原理に基づくモデリング）、プロセスの増加する複雑さに起因して、そのような協調的最適化技法は困難であり、多くの場合、（実行可能であったにせよ）有用であるには実行に時間がかかり過ぎる。例えば、公知の多目的最適化ツールは、設備またはユニットの複数の部分を監視することで（例えば、帰納的に）収集されたデータに基づいて動作して、最適な動作点を特定し、現在の動作点を最適な動作点まで移動させる軌跡を決定する。しかしながら、データ収集及びそれに基づく計算の各サイクルは非常に時間がかかる場合があり、それにより、規定的アクションが特定される頃までには、プロセスはそのオペレーションにおいて遥かに移動しており、結果として、特定された規定的アクションは最早最適ではないか、あるいは無効、逆効果、かつ/または危険ですらあり得る。更に、ユーザが、多目的最適化ツールに入力される変数の量を制限することによって、計算のサイズ及び範囲（したがって、計算に必要とされる時間）の縮小または限定を試みる場合、ユーザは、どの変数を入力として選択するかについて人間による判断を下すことになるが、これは範囲を限定するのみならず、不正確である可能性もあり、したがって、誤った、または不正確な結果をもたらし得る。

【 0 0 8 0 】

更に、公知のデータ監視及び分析ツールは、多くの場合、重要な品質変数のアイデンティティ及び値をオンデマンドで提供することができず（例えば、利用可能な測定技法の制限に起因して）、タイムリーで適切な結果を提供することができない。例えば、プロセス制御システムの予測的分析のために現在使用されている技法は、オフライン技法（例えば、実験室の）またはオンライン技法（例えば、アナライザによる）のいずれかを含むが、これらの両方が、それらそれぞれの欠点を有する。実験室内試験が招く有意な遅延は、いかなる出力も、現在のプロセス状態にとって最適以下、または無効にすらしてしまうため（例えば、被測定信号を、フィードバック信号として使用することはできない）、重要な品質変数のオフラインでの決定は望ましくない。公知のオンラインアナライザには問題点が多く、効果であり、頻繁かつ高コストの整備を必要とするため、重要な品質変数を決定

10

20

30

40

50

するためのオンラインアナライザの使用もまた望ましいとは言えない。このような制限が、生成物の品質、廃棄物の生成、及びオペレーションの安全性に深刻な影響を及ぼし得る。

【 0 0 8 1 】

更に、大規模の監視状況においては、従来型の集中型パフォーマンス監視スキームは、深刻な制限を有する。好適なパフォーマンス監視システムは、耐故障性能力、動作信頼性、及び経済性を有するべきであるが、集中型パフォーマンス監視スキームが大規模監視状況に適用される場合、これらの全てが損なわれてしまう。具体的には、集中型監視装置は、プロセスの全てのユニットの故障を同時に監督する。集中型監視装置は全ての測定変数をその計算に使用するものであり、1つの変数が利用できないか、あるいは1つの通信チャンネルが遮断された場合、全体の監視システムが機能を停止する可能性があるため、あるユニットに故障が見つかり、集中型監視装置は同じ期間中に起こる他のユニットからの更なる故障を検出するその能力について制限されてしまう。加えて、集中型監視装置のリアルタイム能力は、変数の最も低いサンプリングレートによって制限され、これが監視システムの効率性を低減する。更に、プラントにおける異なる変数は、(同じユニット内ですら)非常に異なる動態のスケールを有し、単一の監視スキームは、通常、全体のプロセスまたはユニットの全ての変数に対しては適用されず、特に、変数が互いに影響を及ぼし合うような状況、例えば、相互作用するプロセスユニットにまたがる状況においては最適以下である。なおも更に、プロセス制御システム内に地理的分布が存在する場合(例えば、油送管等の、様々なプロセスユニット間が長距離である場合)、各ユニットは典型的には別個のパフォーマンス監視装置を備え付けられ、それによってローカルの(しかし全体的ではない)パフォーマンス監視、ならびに送信遅延、データ損失、及びバッテリーの制限という問題がもたらされる。

【 0 0 8 2 】

現在使用されている初歩的であり、単変量であり、かつ/あるいは集中型の予測的監視及び分析技法とは異なり、また記述的分析及び規定的分析の帰納的性質とは異なり、本明細書に記載される、工業制御システム及びプラント用の新規の分散型パフォーマンス監視及び分析システム、方法、装置、構成要素、及び技法は、ユーザが、任意の時間点においてプロセス制御システムまたはプラント内で現在起こっている物事を監視することと、現在のデータに基づいて、次または後に起こる可能性が高い物事をリアルタイムで予測することを可能にする。本明細書に開示される新規技法は、「今何が起こっているか?」、「次に何が起こるか?」、「この傾向が続くとどうなるか?」等の疑問に、プロセスデータの特性に関するユーザまたは人間の先入観をほぼまたは全く伴わずに、ユーザが回答することを可能にする。

【 0 0 8 3 】

本明細書に記載される、工業制御システム用の分散型パフォーマンス監視及び分析についての本新規技法は、概して、プラットフォーム(例えば、図2Aに示されるシステムアーキテクチャ100)と、このプラットフォームに加えて、あるいはこのプラットフォームと併せて動作するアプリケーションとを含む。このプラットフォームは、プロセス制御システムまたはプラントが1つ以上のプロセスを制御するように動作する前、動作している間、及び動作した後に、設備及びプロセスの知識で拡張されるデータ駆動型の方法を活用して、例えば、プロセス制御システムまたはプラント(例えば、プロセス制御システム5)の評価、監視、及びデータマイニングをサポートする。一部の実装形態においては、このプラットフォームは、複数の制御システムまたはプラントをサポートし得る。

【 0 0 8 4 】

このプラットフォームに依存して、あるいはこれと併せて動作するパフォーマンス監視/分析アプリケーションは、例えば、推測される測定値、設備監視、故障検出、プロセス予測、因果関係、他の監視アプリケーション、及び/または他の分析アプリケーションを含み得る。これらのアプリケーションによって活用され得る技法としては、データマイニング、最適化、予測的モデリング、機械学習、シミュレーション、分散型状態推定等が挙

10

20

30

40

50

げられる。したがって、監視／分析アプリケーションは、器具類、設備、制御、ユーザインタラクション、及びプロセス等の領域におけるものを含む、プロセス制御システム5の任意の数の任意の部分のパフォーマンス劣化及び故障を監視、予測、及び診断するために使用することができる。

【0085】

したがって、本明細書に記載される分散型工業パフォーマンス監視及び分析技法は、容易に利用可能なプロセス測定値から、重要なプロセス変数の連続的なオンライン評価を提供する、推論メカニズムを含み得る。データモデルを長期間にわたって維持するために、本システムは更に、データモデルを監視、調整、及び強化するための能力をサポートし得る。使用され得る方法としては、最も単純な場合においては、モデルを離調させることが

10

挙げられる（例えば、通常の動作範囲を、予測±3 sに増加させる）。より複雑な例としては、モデル切り替え及びモデル適応が挙げられる。

【0086】

実際、研究及びプロトタイプ試験プログラムの一部として、本発明者らは、本明細書に記載される技法を用いて、プロトタイプの工業プロセスパフォーマンスデータ監視及びデータ分析のフレームワークを確立して、非直線性、非ガウス分布、データサンプルの不規則性、プロセス制約、及び／または他の因子の存在の下、ソフトセンサ開発の問題に対処した。これらのプロトタイプ及び研究において、特殊化学品、バッチ、及び炭化水素プロセスについて、いくつかのソフトセンサによって生成されたデータの監視及び分析を行い、結果として得られた知識を、獲得されたプロセスを向上するために活用した。我々の工業上の提携先により、有意な経済的及び環境上の利点が報告された。我々の研究及びプロトタイプの例は、後の節において記載される。

20

【0087】

ここで図2Aに戻ると、本分散型工業パフォーマンス監視及び分析システム、すなわちDAS100は、ローライズしたパフォーマンス監視をサポートするために必要とされるツール、構成、及びインターフェイスを提供するが、一方で同時に、プロセスプラント環境、例えば図1のプロセスプラント5に関する、大規模データマイニング及びデータ分析をサポートする。例えば、システム100は、データ分析モデルを構成及び開発するためのユーザインターフェイスアプリケーション（例えば、Data Analytics Studio）と、モデルを実行するためのランタイムエンジン（これは、プロセス制御システムによって生成されるリアルタイムデータに基づいて、全体的または部分的に動作し得る）と、分析結果を表示するための、同じまたは別のユーザインターフェイスアプリケーション（例えば、Run-time Dashboard）とを含み得る。システム100は、先述したようなリアルタイム値（例えば、リアルタイム連続値）、事象収集、バッチデータ収集、オペレータラウンドデータ、LIMSデータ、サプライチェーンデータ及び／または気象データ等の外部データ、ならびに構造化データ及び非構造化データの両方を含む、プロセス制御システムと関連付けられる任意の他の種類のデータといった、複数の種類のデータソースに基づいて動作し、あるいはそれをサポートし得る。システム100は、記述的統計、ヒストグラムプロット、相関プロット等の、「常識にとらわれない」標準的分析のセットを提供し得る。加えて、システム100は、ユーザが所望の分析を作成するため、かつソースデータと、複数の異なるデータ分析を適用することにより生成されるn次データとの両方を閲覧するための構造化環境を提供して、プロセス制御システムの様々なデータセット内の潜在的な関係性を特定し、かつ／あるいは製品の能力、品質、及び他の特性を予測し得る。概して、システム100は、プロセス制御システムについての知識の発見、ならびにアクション可能な知識（例えば、プロセス制御システムのパフォーマンスを向上または最適化するため、故障を修正及び／または予測するため、信頼性を向上するため、安全性を向上するため、経済性を向上するため等）を提供することができ、これはデータモデルとして表され得る。

30

40

【0088】

図2Aに示されるように、システム100は高度に分散することができ、例えば、多数

50

の分散型データエンジン102xを含み得る。埋込式データエンジン102xのネットワーク112は、多数（例えば、数百、数千、または数百万）のセンサと、プロセス制御システムの情報のソースとを相互接続し得る。先述したように、データエンジン102xは、プロセス制御システムのデバイス及びノード（例えば、製造設備、プロセス制御デバイス、通信ノード、マテリアルハンドリングシステム、実験室システム、プラントのユーザ、及びプロセス自体）付近で、そこにおいて、またはその内部でクラスタ化される。したがって、分散型工業パフォーマンス監視及び分析システム100の埋込式の性質は、物理的なプロセス制御プラントに対して密接に連結される。しかしながら、プロセス制御システムデバイス及びノードの製造の基本構造への、データエンジン102xの埋め込み及び統合は、データエンジン102xをエンドユーザに対してほぼ不可視にし得る。典型的には、データエンジン102xは小型であり、ネットワーク112に無線で接続され、帯域幅が限定されており、限定されたエネルギー及び十分な熱放散の必要性等の物理的制約の下で動作するが、これは、データエンジン102xが、温度センサまたは他の種類のセンサ等のプロセス制御デバイス内に埋め込まれ得るためである。

10

【0089】

先述したように、分散型データエンジン102xは、ストリーミングプロトコル及び/またはキューイングプロトコルを用いることで、ネットワーク112を通じて他のデータエンジン102xと相互接続する。各分散型データエンジン102xは、1つ以上のデータ監視及び/またはデータ分析アプリケーションをサポートし得る。アプリケーションの莫大な数の接続されたクラスタは、リソースの勘定、故障の検出及び修正、システム管理等について、（決定的であるのではなく）統計的に正確なアルゴリズムの使用を必要とし、各クラスタが、ローカライズされたニーズに対して目的の機能性を提供し得る。

20

【0090】

したがって、分散型工業パフォーマンス監視及び分析システム100は、プロセス制御プラントの安全性、効率性、及び生産性を向上させ得る。例えば、システム100はプロセスパラメータを精密に制御することができ、したがってプロセス製造の総コストを低減し得る。加えて、プロセス制御環境5へのシステム100の統合は、無数にある利点のうちの一つを挙げるだけでも、精密なプロセス制御及びリアルタイムの品質保証による、より良好な製品品質及びより少ない廃棄物；プログラム可能なサブシステムの結果として、より柔軟性の高い、迅速に構成される生産ライン；より有効であり、予防的であり、より低いコストの整備につながる、システム正常性の監視；より良好な監視及び制御に起因する、より安全な作業環境；ならびに、スマートRFIDタグの使用等を通じた、より良好な部品組み立て技法を結果としてもたらし得る。

30

【0091】

また、分散型工業パフォーマンス監視及び分析システム100は、ユーザとプロセス制御システム5との間の人間-機械の相互作用を、プロセスプラント5内にリアルタイムの人間プラス機械の制御ループを生成する点にまで改善し得る。例えば、改善された人間-機械の相互作用は、オペレータ/整備/信頼性の誤りが存在しないことを確実にすることによって、ならびに事故を低減することによって、品質及び生産性を向上させ得る。更に、プロセス制御プラント5のサービスの利用可能性、信頼性、及び連続的品質は、システム100による、進歩した制御、冗長性、知的アラーム、自己診断、及び修復を通じて達成され得る。

40

【0092】

例示的工業パフォーマンスデータ監視/データ分析エンジン

ここで、分散型の、工業パフォーマンス分散型データ監視及び/または分析エンジン102xに注目すると、概して、分散型データエンジン102xは、1つ以上のデータソースによって、例えばデータエンジン102xが埋め込まれているデバイスもしくは構成要素によって、またはデータエンジン102xが連結もしくは接続されている構成要素（複数可）によって、生成または別様に観察される、全てまたは大部分のプロセス制御に関するデータ（例えば、ビッグデータ）を収集、観察、回収、受信、処理、記憶、キャッシュ

50

、及び/または分析する、データ分析通信ネットワーク112のノードであり得る。一部の状況においては、分散型エンジン102xは、追加的なデータ(例えば、それが実施する分析の結果)を生成し得、かつ/またはデータ分析ネットワーク112の他のノードに選択されたデータを送信もしくは転送し得る。本明細書で交換可能に使用する場合、用語「プロセス制御ビッグデータ」、「プロセスビッグデータ」、及び「ビッグデータ」は概して、プロセス制御システムまたはプラント(例えば、プロセス制御システムまたはプラント5)内に含まれ、それと関連付けられるデバイス及び/または構成要素(例えば、プロセス制御デバイス/構成要素と、分析デバイス/構成要素との両方)によって生成、受信、及び/または観察される全て(またはほぼ全て)のデータを指し、特に、全て(またはほぼ全て)のデータは、プロセス制御システムまたはプラントがリアルタイムで1つ以上のプロセスを制御するように実行している間に生成、受信、及び/またはされるものである。ある実施形態においては、プロセスプラント5内に含まれ、それと関連付けられる全てのデバイスによって生成され、それによって作成され、そこで受信され、またはそれによって別様に観察される全てのデータ(全てのプロセスデータ及び全ての分析データを含む)は、データ分析通信ネットワーク112内にビッグデータとして収集及び記憶される。プロセスプラント及びプロセス制御環境においては、時間の範囲及び特定のデータ点の存在または脱落は重大であり得るため、このビッグデータの収集及び分析は、安全性、信頼性、及び経済性の向上にとって重要である。例えば、ある特定の時間間隔内に、特定のデータ値が、プロセスプラントの受信側の構成要素に送達されない場合、プロセスは無制御になり得、これは火事、爆発、設備の損失、及び/または人命損失につながり得る。更に、プロセスプラント内で動作し、かつ/あるいはプロセスプラントの外部で動作する、異なる構成要素、エンティティ、及び/またはプロセス間の、複数及び/または複雑な時間基準の関係性が、動作効率、製品品質、及び/またはプラントの安全性に影響を及ぼし得る。

10

20

【0093】

分散型データエンジン102xによって生成、収集、観察、回収、受信、記憶、キャッシュ、処理、分析、及び/または転送されるプロセス制御ビッグデータは、プラント5内のプロセスにおいて直接活用されているデータ、またはそのプロセスを制御することから生成されているデータを含み得、例えば、コントローラ、入力/出力(I/O)デバイス、及びフィールドデバイス等のプロセス制御デバイスによって生成または使用される、一次のリアルタイム及び構成データを含み得る。追加的または代替的に、データエンジン102xは、そのような一次のプロセス制御データ、ならびにプロセスプラント5内の他のデータ、例えば、データ分析通信ネットワーク112及び/またはプラント5内の他の通信ネットワークのネットワーク制御に関するデータ、帯域幅を示すデータ、ネットワークアクセス試行、診断データ等を送達及び発送することに関するデータを、生成、収集、観察、処理、分析、記憶、受信、回収、キャッシュ、及び/または転送し得る。更に、一部のデータエンジン102xは、プロセス制御データ分析通信ネットワーク112が収集したプロセス制御ビッグデータを処理及び/または分析することによって、その内部で学習された、結果、学習、及び/または情報を示すデータを、生成、収集、観察、記憶、キャッシュ、回収、受信、及び/または転送し得る。典型的には、そのような分析結果、学習、及び/または学習された情報は、1つ以上のデータエンジン102xによって実施される分析から生成される。

30

40

【0094】

したがって、分散型データエンジン(例えば、DDE102x)は、本明細書においては交換可能に、「ビッグデータアプライアンス」、「ビッグデータアプライアンスノード」、または「アプライアンスノード」と呼ばれる。ほとんどの場合において、ビッグデータアプライアンスノード102xの分散型データエンジンは、リアルタイムで(例えば、ストリーミングを介して)ビッグデータを送信及び受信するため、ならびに一部の実施形態においては、後のストリーミングまたはプロセス制御データ分析通信ネットワーク112を通じた他の送達のためにリアルタイムビッグデータをキャッシュ及び/または記憶す

50

るために、マルチコアハードウェア（例えば、マルチコアプロセッサ）を含む。したがって、分散型データエンジン102xはまた、ビッグデータのキャッシュ及び/または記憶のためのメモリ（例えば、高密度メモリ）も含む。データエンジン102xによって送信、受信、ストリーミング、キャッシュ、収集、記憶、受信、回収、キャッシュ、及び/または別様に観察され得るリアルタイムデータの例としては、測定データ、構成データ、バッチデータ、事象データ、及び/または連続データ等のプロセス制御データが挙げられ得る。例えば、構成、バッチレシピ、設定点、出力、レート、制御アクション、診断、アラーム、事象、及び/またはそれに対する変更に対応するリアルタイムデータが、収集され得る。リアルタイムデータの他の例としては、プロセスモデル、統計、ステータスデータ、ネットワーク及びプラント管理データ、及び分析結果が挙げられ得る。本明細書に記載される技法のうちのいずれかまたは全てと共に活用できる、様々な種類の例示的ビッグデータアライアンス及びそれらの構成要素についての記載は、前述の米国特許出願第13/784,041号、同第14/174,413号、及び同第14/212,493号において見出され得るが、本明細書に記載される技法のうちのいずれかまたは全ては、他の好適なビッグデータアライアンスと共に活用できることが理解される。

10

【0095】

典型的には、分散型データエンジン102xは概して、プロセス制御プラントまたはシステム5内の1つ以上のデータソースによって生成または提供されており、かつ/あるいは別様にプロセス制御プラントまたはシステム5のリアルタイムオペレーションと関連付けられている、ビッグデータに基づいて動作する。例えば、DDE102xは、それが埋め込まれているデバイスにおいて受信され、かつ/あるいはそのデバイスによって生成されたデータ、またはその対応するデータソース（複数可）において受信され、かつ/あるいは生成されたデータを、収集及びタイムスタンプし得る。収集されたデータは、データエンジン102xのローカルメモリに、（少なくとも一時的に）保存され得る。一部の状況においては、少なくとも一部のデータは、前述の「STREAMING DATA FOR ANALYTICS IN PROCESS CONTROL SYSTEMS」と題される米国出願第14/506,863号に記載されるもの等の特化履歴オブジェクト通信プロトコル、またはKafka等の別の好適な通信プロトコルまたはメッセージングシステムを用いて、1つ以上の他のデータエンジン102xに移送またはストリーミングされ得る。

20

30

【0096】

一部の実施形態においては、1つ以上のデータエンジン102xは、リアルタイム連続値、事象収集、バッチデータ収集、オペレータラウンドデータ、及び/または他のデータを含む、多次元データに基づいて、大規模なデータマイニング及びデータ分析をサポートする。分散型データエンジン102xは、1つ以上のデータ分析を、そのローカルで収集したデータ及び/または他のDDE102xによって収集されたデータに基づいて実施するように構成され得る。例えば、DDE102xは、構造化データ（例えば、メモリ、リレーショナルデータベース、及び/または非リレーショナルデータベースに記憶されるか、あるいはストリーミングされる、時系列及び表データ）、ならびに非構造化データ（例えば、pdf）に基づいて動作する様々なツールを含み得る。加えて、DDE102xは、任意の1つ以上の所望のターゲット環境をサポートし、そこにおけるそれぞれのターゲットコード（例えば、Java、C#、Rスクリプト、Pythonスクリプト、Matlab（登録商標）スクリプト、Statgraphics等）を実行し得る。DDE102xは、製品の能力、品質、及び/または他の所望の特性を予測すること等、結果及び/または有用な情報を生成するために、学習アルゴリズム（例えば、部分最小二乗回帰、主構成要素分析等）、分類技法（例えば、ランダムフォレスト、パターン認識等）、及び/または他のデータ分析を実施し得る。ローカル分析の結果は、データエンジン102xのローカルメモリに記憶され得、それ自体が、追加的なビッグデータとして処理され得る。更に、一部のデータエンジン102xは、モデルと、モデルを実行するためのランタイムエンジンと、ユーザインターフェイスに結果を表示するためのダッシュボードとを構成

40

50

及び開発するためのインターフェイスを含み得る。例えば、分散型データエンジン102xは、前述の「DATA PIPELINE FOR PROCESS CONTROL SYSTEM ANALYTICS」と題される米国出願第62/060,408号に考察されるもの等の様式で、かつ/あるいは本開示の後の節において考察されるように、ランタイムについての分析を伴って構成され得る。表示される結果は、標準化記述的統計、ヒストグラム、相関プロット、及び/または様々なデータセット内の潜在的な関係性を特定できる他のデータ表現を含み得る。

【0097】

一部の場合においては、分散型データエンジン102xは、ホストデータソース内に組み込まれるか、その中に製造されるか、それに直接連結されるか、あるいは別様にそれと共在する（例えば、図2Aに示されるデータエンジン102a、102b、及び102c）。一部の場合においては、分散型データエンジン102xは、プロセス制御データ分析通信ネットワーク112の自立型ビッグデータノードであり得る（例えば、図2Aに示されるデータエンジン102d及び102e）。すなわち、これらの場合においては、データエンジン102xは、プロセス制御システムまたはプラント5のデータソース内に組み込まれたり、それと共在したりしてはいないが、1つ以上のデータソースによって生成されたデータを別様に観察し得る（例えば、分散型データエンジンが従来型のプロセス制御通信リンクと連結されている場合、データエンジン102c等）。埋込式であるかに自立型であるかに関わらず、データエンジン102xは、1つ以上のデータソースによってローカルで生成及び/または提供されたビッグデータを分析して、知識を発見または学習する。この学習した知識は、データエンジン102xに記憶され、ローカルでデータエンジン102xによって動作され、かつ/あるいはビッグデータとして他のデータエンジン102x、例えば受信側のビッグデータノードに提供または送信され得る。加えて、分散型データエンジン102xは、その既知のビッグデータまたは記憶されたビッグデータの一部分を、他のデータエンジン102x及び/または分析ネットワーク112の他のノード（例えば、ローカルまたはリモートのユーザインターフェイスノード）に対して提供し得る。

【0098】

図3は、例示的な分散型工業プロセスパフォーマンスデータ監視及び/またはデータ分析エンジン150の簡略化したブロックダイアグラムを含み、このインスタンスは、図2Aのプロセス制御データ分析通信ネットワーク112に含まれ得る（例えば、分散型データエンジン102x）。図3を参照すると、例示的な分散型データエンジン150は、ビッグデータをキャッシュ、記憶、及び/またはヒストライズするためのビッグデータ記憶領域155と、1つ以上のビッグデータアプライアンスのレシーバ160と、1つ以上のビッグデータアプライアンス要求サービサ165とを含む。ビッグデータアプライアンスレシーバ160の各々は、1つ以上のデータソース168からのデータを受信及び/または観察するように構成される。ある実施例においては、ビッグデータアプライアンスレシーバ160は、データハイウェイ10、Fieldbusネットワーク、Wireless HARTネットワーク等の従来型のプロセス制御通信ネットワークへのネットワークインターフェイスを介して、従来型のプロセス制御通信ネットワークを通じて移動するデータを受信及び/または観察する。追加的または代替的に、ビッグデータアプライアンスレシーバ160は、DDE150が組み込まれるか、その中に製造されているか、あるいはDDE150がローカルで、直接的に、及び/または別様に密接に連結されている、プロセス制御デバイスまたは他の種類のデバイス等のその対応するデータソース（複数可）168から、ローカルインターフェイスを介してデータを受信し得る。なおも追加的または代替的に、ビッグデータアプライアンスレシーバ160は、例えばデータ分析ネットワークインターフェイス175を介して、ビッグデータパケットを受信し得る。受信されるビッグデータパケットは、別のDDE150からストリーミングされたか、かつ/あるいはデータエンジン150が共に存在するビッグデータソースによって生成されている場合がある。受信/観察されるデータのソース（複数可）168に関係なく、ビッグデータアプ

10

20

30

40

50

ライアンスレシーバ160は、受信/観察されるデータパケット及び/またはメッセージを処理して、実質的データ及びその中で運搬されるタイムスタンプを回収し、また、データエンジン150のビッグデータ記憶領域155に、例えば時系列データとして、ならびに任意選択でメタデータとしても、その実質的データ及びタイムスタンプを記憶する。本明細書で使用する場合、用語「メタデータ」は概して、データについての統計情報、データについてのカテゴリ情報、要約情報、記述的情報、定義等の、データについてのデータを指す。ビッグデータ記憶領域155は、RAID(独立した複数のディスクからなる冗長配列: Redundant Array of Independent Disks)ストレージ、固体ストレージ、クラウドストレージ、高密度データストレージ、及び/またはデータバンクもしくはデータセンタストレージにとって好適であり、他のノードに対して単一または一元的な論理データ記憶領域またはエンティティの見かけを有し、ビッグデータをローカルで記憶及び/またはヒストライズするように構成し得る、任意の他の好適なデータストレージ技術等の、複数のローカル及び/またはリモートの物理データドライブまたはストレージエンティティを備え得る。

【0099】

ビッグデータアプライアンス要求サービサ165の各々は、例えば、要求エンティティ、あるいはローカルもしくはリモートのデータ分析アプリケーション、ユーザインターフェイスアプリケーション、または別のアプリケーション等のアプリケーションの要求毎に、ビッグデータアプライアンス記憶領域155に記憶されている時系列データ及び/またはメタデータにアクセスするように構成される。例えば、ビッグデータアプライアンス要求サービサ165は、ビッグデータアプライアンス記憶領域155に記憶されている選択されたデータを、所与の時間間隔で、その選択されたデータのサブスクリバに対して公開し得る。別の例においては、ビッグデータアプライアンス要求サービサ165は、リモートで実行するデータ分析アプリケーションの要求毎に、ローカルのビッグデータアプライアンス記憶領域155内に記憶されているデータを回収し得る。DDE150に記憶されているデータにアクセスする追加的な例については、後の節において考察する。そのようなデータは、リレーショナルデータベースもしくは非リレーショナルデータベース、または他のデータ構造を含む、様々なデータフォーマットで記憶され得る。一部の実施形態においては、標準化クエリフォーマットが、これらのデータソースのいずれかにおけるデータにアクセスするために使用され得る。

【0100】

一部の実施形態においては、分散型データエンジン150は、メモリ155内に記憶されたビッグデータの少なくとも一部に対して、それぞれのデータ分析及び/または学習を実施するための、1つ以上のビッグデータアナライザ170を含む。ローカル分析及び/または学習の実行は、ユーザまたは別のノードによって生成されるコマンドまたは命令に対する応答として実施され得る。追加的または代替的に、ローカル分析及び/または学習の実行は、ユーザまたは他のノードからの、学習分析を開始及び/または実施するためのいかなる入力の使用も伴わずに、自動的及び/または自律的な様式で実施され得る。例えば、データ分析及び/または学習は、先に考察したもの等の様式で、前述の「DATA PIPELINE FOR PROCESS CONTROL SYSTEM ANALYTICS」と題される米国出願第62/060,408号に考察されるもの等の様式で、あるいは何らかの他の好適な様式で実施され得る。ある実施形態においては、ビッグデータアナライザ170は、新しい情報または知識を発見、検出、または学習するために、記憶されているデータに対して大規模なデータ分析(例えば、データマイニング、データ発見等)を、個別的または集合的に実施する。データマイニングは概して、大量のデータを調査して、普通ではない記録または複数のデータ記録群といった、新規または以前には知られていなかった興味深いデータまたはパターンを抽出するプロセスを伴う。ビッグデータアナライザ170はまた、記憶されているデータに対して大規模なデータ分析(例えば、機械学習分析、データモデリング、パターン認識、予測的分析、相関分析等)を実施して、記憶されているデータ内の潜在的な関係性または推論を予測、計算、または特定し

10

20

30

40

50

得る。

【 0 1 0 1 】

ある実施形態においては、複数のビッグデータアナライザ 170（及び/または少なくとも1つのビッグデータアナライザ 170の複数のインスタンス）が、分散型データエンジン 150のビッグデータ記憶領域 155に記憶されているデータを分析するように、かつ/あるいは他の分散型データエンジン 102xの1つ以上の他のビッグデータ記憶領域に記憶されているデータを分析するように、並列に、かつ/あるいは協同的に動作してもよい。更に、複数のビッグデータアナライザ 170は、ある種の協同的データ分析及び学習として、互いの中で、計算されたパラメータ及びモデル情報を共有、交換、または移送し得る。複数のビッグデータアナライザ 170は、同一のビッグデータノードに共在してもよく、あるいは異なるビッグデータノードに存在してもよい。本明細書に記載される技法のうちいずれかまたは全てと共に活用し得る、協同的データ分析の例は、前述の「DATA PIPELINE FOR PROCESS CONTROL SYSTEM ANALYTICS」と題される米国出願第 62/060,408号において見出されるが、任意の好適な共同的数据分析技法（複数可）が、本開示のいずれかの態様または全ての態様と共に活用され得る。ビッグデータアナライザ 170によって実施される分析の結果は、ビッグデータアプライアンス記憶領域 155に記憶されてもよく、かつ/あるいは要求エンティティまたはアプリケーションに返されてもよい。

10

【 0 1 0 2 】

ある実施形態においては、ビッグデータレシーバ 160、ビッグデータアプライアンス要求サービサ 165、及び/またはビッグデータアナライザ 170の少なくとも一部分が、1つ以上の集積回路、半導体、チップ、または他の好適なハードウェアに含まれるか、実装される。例えば、スペクトル分析を実施するビッグデータアナライザ 170は、前述の「AUTOMATIC SIGNAL PROCESSING - BASED LEARNING IN A PROCESS PLANT」と題される米国出願第 14/507,252号に記載されるもの等、ビッグデータノードに含まれる集積回路チップによって実装され得る。ある実施形態においては、ビッグデータレシーバ 160、ビッグデータアプライアンス要求サービサ 165、及び/またはビッグデータアナライザ 170の少なくとも一部分は、メモリに記憶され、かつ分散型データエンジン 150において動作するプロセスによって実行可能である、コンピュータが実行可能な命令を含む。例えば、ビッグデータアプライアンスレシーバ 160、ビッグデータアプライアンス要求サービサ 165、及び/またはビッグデータアプライアンスアナライザ 170の少なくとも一部は、1つ以上の非一時的で有形のメモリまたはデータストレージデバイスに記憶されている、それぞれの、コンピュータが実行可能な命令を含み、それぞれのビッグデータ機能のうち1つ以上を実施するように、1つ以上のプロセッサによって実行可能である。

20

30

【 0 1 0 3 】

一部の実施形態においては、少なくとも一部のビッグデータアナライザ 170は分散型データエンジン 150に含まれず、代わりに、同一のホストデータソースデバイスまたは構成要素において分散型データエンジン 150と共に共在し、かつデータエンジン 150と通信可能に接続される。例えば、記憶領域 155、レシーバ 160、及びサービサ 165を含むデータエンジン 150は、コンピュータが実行可能な命令の第1のセットによって実装され得、ビッグデータアナライザ 170は、半導体チップまたはコンピュータが実行可能な命令の第2のセットによって実装され得、これは、コンピュータが実行可能な命令の第1のセットと同一の非一時的で有形のメモリまたはデータストレージデバイスに、記憶されても、あるいは記憶されなくてもよい。一部の実施形態においては、ビッグデータアナライザ 170はデータエンジン 150に含まれず、同一のホストデータソースデバイスまたは構成要素においてデータエンジン 150と共に共在しないが、それでもなお、データエンジン 150と通信可能に接続される。例えば、ビッグデータアナライザ 170は、DDE 150ではない、データ分析ネットワーク 112の別のノード、例えばユーザインターフェイスノードまたは分析サーバに配置され得る。

40

50

【0104】

図3に示される分散型データエンジン150は更に、データエンジン150による、データ分析ネットワーク112を通じたペイロードビッグデータの送信及び受信、ならびにデータ分析ネットワーク112の他のデータエンジン及びノードとの通信(例えば、信号通信及び他の種類の通信)を可能にするように構成される1つ以上のネットワークインターフェイス175を含む。例えば、データエンジン150は、ネットワークインターフェイス175を用いることで、別のノードによって公開されている1種類のデータをサブスクライブし得、公開するノードによって生成される種類のデータ、及びデータエンジン150がサブスクライブされる種類のデータは、ネットワークインターフェイス175を介して受信され得る。

10

【0105】

先に考察したように、分散型データエンジン102xによって、プロセス関連データ、プラント関連データ、及び他の種類のデータ等の様々な種類のリアルタイムデータが、生成、収集、観察、回収、受信、記憶、キャッシュ、処理、分析、及び/またはストリーミングされ得る。プロセス関連データの例としては、プロセスがプロセスプラント5において制御されている間に生成される(ならびに、一部の場合においては、プロセスのリアルタイム実行の効果を示す)、連続、バッチ、測定、及び事象データが挙げられる。更に、プロセス関連データとしては、プロセス定義、構成データ及び/またはバッチレシピデータ等の配列データまたはセットアップデータ、構成に対応するデータ、プロセス診断の実行及び結果等が挙げられ得る。

20

【0106】

プロセスプラント5に関連するが、プロセスプラント5のプロセスを直接構成、制御、または診断するアプリケーションによっては生成され得ないデータ等のプラント関連データは、分散型データエンジン102xによって、ビッグデータとして生成、収集、観察、回収、受信、記憶、キャッシュ、処理、分析、及び/またはストリーミングされ得る。プラント関連データの例としては、振動データ、蒸気トラップデータ、プラントの安全性に対応するパラメータの値を示すデータ(例えば、腐食データ、ガス検出データ等)、プラントの安全性に対応する事象を示すデータ、機械の正常性に対応するデータ、プラント設備及び/またはデバイス等のプラントの資産に対応するデータ、構成に対応するデータ、設備、機械、及び/またはデバイス診断の実行及び結果、ならびに診断及び予後にとって有用であるデータが挙げられる。

30

【0107】

更に、プロセス制御ビッグデータネットワークバックボーンに関連する、ならびにプロセスプラント5の様々な通信ネットワークの、データハイウェイのトラフィックデータ及びネットワーク管理データ、ユーザトラフィック、ログイン試行、クエリ、及び命令等のユーザ関連データ、テキストデータ(例えば、ログ、運転手順、マニュアル等)、空間データ(例えば、位置情報データ)、ならびにマルチメディアデータ(例えば、有線TV、ビデオクリップ等)を含む、他の種類のデータは、データエンジン102xによって、ビッグデータとして生成、収集、観察、回収、受信、記憶、キャッシュ、処理、分析、及び/またはストリーミングされ得る。

40

【0108】

一部の実施形態においては、動的測定及び制御データが、分散型データエンジン102xによって、ビッグデータとして自動的に生成、収集、観察、回収、受信、記憶、キャッシュ、処理、分析、及び/またはストリーミングされ得る。動的測定及び制御データの例としては、プロセス動作における変化を特定するデータ、設定点等の運転パラメータにおける変化を特定するデータ、プロセス及びハードウェアアラームの記録、ダウンロードまたは通信障害等の事象が挙げられる。加えて、コントローラ構成、バッチレシピ、アラーム、及び事象等の静的データは、変化が検出されるとき、またはコントローラもしくは他のエンティティがデータ分析通信ネットワーク112に初めて追加されるときに、デフォルトで自動的に収集され得る。

50

【0109】

また、一部のシナリオにおいては、動的制御及び測定データについて説明または特定する少なくとも一部の静的メタデータが、メタデータにおける変化が検出されるとき、分散型データエンジン102xにおいて取り込まれる。例えば、コントローラ構成において、モジュールまたはユニットの測定及び制御データに影響を与え、コントローラによって送信されねばならない変更がなされる場合、関連付けられるメタデータのアップデートが、データエンジン102xによって自動的に取り込まれる。追加的または代替的に、外部システムまたはソース（例えば、気象予報、公的事象、企業の決定等）からデータをバッファリングするために使用される特別なモジュールと関連付けられるパラメータ、調査データ、及び/または他の種類の監視データが、データエンジン102xによって自動的に取り込まれ得る。

10

【0110】

一部の状況においては、エンドユーザによって作成された追加のパラメータが、分散型データエンジン102xに自動的に取り込まれる。例えば、エンドユーザは、モジュールに特別な演算を作成してもよく、または収集されるべきパラメータをユニットに追加してもよく、あるいはエンドユーザは、デフォルトでは通信されない標準的コントローラ診断パラメータの収集を欲し得る。エンドユーザが任意選択で構成するパラメータは、デフォルトのパラメータと同じ様式で通信され得る。

【0111】

工業パフォーマンス監視/分析用のユーザインターフェイスアプリケーション

20

先述したように、データ分析システム、すなわちDAS100は、1つ以上のユーザインターフェイスアプリケーションを含み得、これを介して、データに関連する人間-機械の相互作用が行われる。これらのユーザアプリケーションの例示的インスタンスの存在が、参照120a~120dによって図2Aに描写されている。本開示の本節及び図4A~4Qは、工業パフォーマンス監視/分析用のユーザインターフェイスアプリケーションについてより詳細に説明する。このユーザインターフェイスアプリケーションは、例えば、図1のプロセスプラントもしくはシステム5、図2Aの工業パフォーマンス監視/分析システム100、ならびに/または図3の監視及び分析エンジン150によって提供されるか、それらと共に運転し得る。しかしながら、本明細書に記載される工業パフォーマンス監視/分析用ユーザインターフェイスアプリケーションは、プロセス制御システム用の他の工業パフォーマンス監視及び分析システムによって提供されるか、あるいはそれらと共に運転してもよい。しかし、考察の容易さのために、工業パフォーマンス監視/分析ユーザインターフェイスアプリケーションは、図1、2、及び3を同時に参照しながら下において考察される。加えて、読解の容易さのために、Industrial Performance Monitoring/Analytics User Interface Applicationは、本明細書においては、一般的な目的及び/または他のユーザインターフェイスアプリケーションと区別するために大文字を用いて呼称され、また、本明細書においては、交換可能に、「Data Analytics User Application」、「Data Analytics User Interface Application」、「Data Analytics Application」、「DDE User Interface Application」、または「User Interface Application」とも呼ばれる。

30

40

【0112】

DDE User Interface Applicationは、ユーザが分散型工業プロセスパフォーマンス監視/分析システムすなわちDAS100と相互作用して、構造を定義し、データを照会し、ドラフトデータモデルを構築及び評価するためのインターフェイスを提供する。ドラフトデータモデルが完成したら、DDE User Interface Applicationは、データモデルのランタイムエンジンへのダウンロード、及びオンラインプロセス制御システムと共に運転するようなデータモデルの展開を可能にする。展開されたデータモデル（実行中またはオンラインデータモジュールと

50

も呼ばれる)は、DDE User Interface Applicationのランタイムダッシュボードを介してアクセス及び監視され得る。DDE User Interface Applicationはまた、実行中データモデルに対応するアラーム及び通知を生成することもできる。

【0113】

具体的には、DDE User Interface Applicationは、ユーザがデータモデルを作成、閲覧、及び修正することを可能にし、これらのデータモデルの各々が1つ以上の、入力データセットに対して実行または実施されるデータ分析(例えば、記述的、予測的、及び/または規定的分析)を定義する。データモデルは、オフラインモードで(例えば、データモデルが、オンラインまたは実行プロセス制御システムのライブデータソースから接続を切られている間に)ドラフトまたは作成され、このモードでは、データモデルは、本明細書においては「オフラインデータモジュール」と呼ばれる。概して、オフラインデータモジュールは、オフラインデータダイアグラムによって定義され、このオフラインデータダイアグラムは、ユーザが、DDE User Interface Applicationを介して、「ブロック」または「データブロック」のセットを選択し、ダイアグラム上でそれらのブロックを「ワイヤ」のセットを用いて所望の様式で相互接続することによって作成する。各データブロックは入力部を含み、ここを介して、データがブロック内に受信される。各データブロックはまた、特定の機能、アクション、アルゴリズム、及び/またはオペレーションを表すかあるいは定義し、これらは各データブロックによって、その入力データに基づいて実施され、それによって1つ以上の出力部を介して他のブロックに提供することができる出力データを生成する。各データブロックは、ブロック及びその相互接続ワイヤのオフラインダイアグラム表現を実行可能なコードにコンパイルし、実行するように、別個に評価することができ、各データブロックの実行の結果は、オフラインデータダイアグラムに呈示される。後の節において考察されるように、データブロックのコードへのコンパイル、及び後続のその実行は、異なるターゲット環境及び場所にまたがって、分配的に実施され得る。

【0114】

同様に、オフラインデータダイアグラムは、全体としても評価することができる。オフラインデータダイアグラムの評価は、データブロック及びワイヤをコンパイルすること、このコンパイルされたコードを実行して、入力データ及び出力データをワイヤにまたがって移送し、オフラインデータダイアグラムのブロック及びワイヤの特定の構成によって定義される機能、アクション、アルゴリズム、及び/またはオペレーションを実施することを含む。また、個別のデータブロックと同様に、オフラインデータダイアグラムのコンパイル、及びこのコンパイルされたオフラインデータダイアグラムの実行も、異なるターゲット環境及び場所にまたがって、分配的に実施され得る。

【0115】

更に、DDE User Interface Applicationは、ユーザがオフラインデータモジュールを「オンラインデータモジュール」へと転換することを可能にする。この転換は、データモデルのオンラインデータモジュールが、オンラインプロセス制御システムのライブデータソースにバインドするか、あるいはそれによって生成されているライブ(例えば、ストリーミング)データを別様に受信し、そこで定義されている1つ以上のデータ分析をライブデータに対して実施し、ユーザインターフェイス、ヒストリアン、または他のアプリケーションに出力を提供するようなものである。例えば、データモデルによって生成される出力データは、プロセスプラント及び/またはその中で制御されているプロセスに対応する、記述的、予測的、及び/または規定的情報またはデータを含み得る。

【0116】

具体的には、ユーザは、特定のデータモデルのオンラインデータダイアグラムを、その特定のデータモデルのオンラインデータダイアグラムへと転換することができる。オフラインデータダイアグラムと同様に、オンラインデータダイアグラムは、ワイヤのセットで

10

20

30

40

50

相互接続されたデータブロックのセットを含んでおり、また、オフラインデータダイアグラム及びオンラインデータダイアグラムは概して、同一のデータモデルに対応しているため、オンラインデータダイアグラムによって定義される機能、アクション、アルゴリズム、及び/またはオペレーションは、オフラインデータダイアグラムのものに対応する。しかしながら、オフラインデータダイアグラムとオンラインデータダイアグラムとの間では、主として、オンラインデータモジュールのオンラインプロセスプラントへの接続を適合させるため（しかし、必ずしもそのためだけではない）、ブロック及び相互接続のうちの少なくとも一部が異なる。

【 0 1 1 7 】

これもまたオフラインデータダイアグラムと同様に、オンラインデータダイアグラムは、データモデルのオンラインデータモジュールに対応する実行可能なコードへと、全体としてコンパイルすることができる。オンラインデータダイアグラムの特定のブロックのコンパイルは、特定のブロックの、オンラインプロセスプラント内のそれぞれのデータソース及びデータコンシューマとのバインディングの定義をもたらす。オンラインデータモジュールの展開は、これらのバインディングをインスタンス化し、実行可能なもののランまたは実行を引き起こし、それによって、オンラインデータモジュールが、オンラインプロセスプラントのオペレーションと併せて実行されるように、オンラインデータモジュールをオンラインプロセスプラントと統合する。オンラインデータダイアグラムのコンパイル、ならびに結果として得られたオンラインデータモジュールのバインディング及び実行は、異なるターゲット環境及び場所にまたがって、分配的に実施され得る。

【 0 1 1 8 】

特に有用な実施形態においては、データモデルのオンラインデータモジュールは、プラントで実行中のオンラインプロセスの進行中の制御の結果としてプロセスプラントのデバイスまたは構成要素によって生成される、連続リアルタイムデータを受信し、少なくとも部分的にはそれに基づいて動作する。例えば、オンラインデータモジュールは、プロセスプラント及びプロセス制御システムがプロセスを制御するように動作している間、オンラインプロセスプラント及び/またはプラントに含まれるプロセス制御システムによって生成される連続リアルタイム時系列データに基づいて動作する。データモデルは、連続リアルタイムデータストリームに基づいて継続的に動作し（例えば、データモデルによって定義された、その1つ以上のデータ分析機能及び（存在する場合は）他の機能を実施することによって）、結果または出力データのリアルタイムストリームを継続的に生成し、このリアルタイムストリームは、ユーザインターフェイスに表示され得（例えば、ローリング折れ線グラフまたは他の描写として）、データモデルの出力の現在のリアルタイム値を継続的に反映し得る。ある例においては、データモデルによって生成され、ユーザインターフェイスに表示されるデータ出力は、少なくとも1つの予測値及びその経時的な変動を含む。しかし、概して、データモデルは、プロセス制御プラントまたは環境5に関する故障検出、予測、及び規定という目的のために、プロセス制御プラントまたは環境5内の様々なデータソースによって生成された大量のデータに基づいて動作することが可能である。

【 0 1 1 9 】

ある実施形態においては、異なるインスタンスの `Application` を、様々なプラットフォーム（例えば、`Apple Macintosh`、`Linux`、`Windows` 等）で使用できるように、また時には同時に、様々なコンピューティングデバイスにおいて様々なユーザが使用できるように、`DDE User Interface Application` はウェブベースであり、ウェブブラウザを通じてアクセスされる。しかしながら、`DDE User Interface Application` は、ウェブベースの実装形態に限定されるものではなく、プラットフォームからは独立し、複数のユーザ及び/またはインスタンスまで同時に拡張され得る、任意の他の好適な実装形態を活用してもよい。

【 0 1 2 0 】

更に、`DDE User Interface Application` は本明細書に

10

20

30

40

50

において、同時に実行される DDE User Interface Application の複数のインスタンスを伴って、単一のプロセス制御プラントまたは環境 5 に適用されるものとして記載されているが、この構成は例証であるに過ぎず、限定的であることを意図しない。例えば、一部の構成においては、DDE User Interface Application は異なる場所で、異なるプロセスにおいて独立して運転する、複数のプロセス制御プラントまたは環境に対して適用され得る。例えば、サーバまたはコンピュータのバンクによって提供される単一の DDE User Interface Application が、その DDE User Interface Application の一部のインスタンスは製油所 A で実行され、一部のインスタンスが製油所 B で実行されるように、ある石油会社の複数の製油所で活用され得る。

10

【0121】

ともあれ、先に考察したように、DDE User Interface Application は、データブロックによってそれぞれの入力データに対して実施され得る機能、アクション、アルゴリズム、及び/またはオペレーションを、ブロックまたはデータブロックとして抽象化する。ブロックまたはデータブロックの例は、例えば、ロード、クリーン、操作、分析、視覚化等、それらそれぞれの機能によって指定される。各データブロックは、それぞれの入力データをロードまたは獲得し、その 1 つ以上のそれぞれの機能、アクション、アルゴリズム、及び/またはオペレーションを、獲得した入力データに対して実施し、1 つ以上のそれぞれの結果またはそれぞれの出力データを生成し得る。複数のデータブロックが所望に応じて相互接続されて（例えば、直列、並列、多対一、一対多等）、ドラフトデータモデルまたはオフラインデータモジュールを表すか、あるいはそれを定義するオフラインデータダイアグラムを形成し得、このドラフトデータモデル/オフラインデータモジュールは、オフラインデータモジュール（またはその一部）を、1 つ以上のデータセットのセットまたはオフラインデータソースに適用するか、あるいはそれに基づいて動作させることによって、実行または評価できる。例えば、オフラインデータモジュールの実行または評価中、オフラインデータダイアグラムは実行可能なコードへとコンパイルされ、特定されたデータセット（複数可）がオフラインモジュールに、あるいはその使用のために獲得及びロードされ、コンパイルされたオフラインモジュールの相互接続されたブロックが実行されて、それらそれぞれのオペレーションをそれらそれぞれの入力に対して実施し、それぞれの結果をそれらそれぞれの出力で提供し、その結果、開発または作成されているデータモデルに対応する全体的なオフラインデータモジュールによって生成される、1 つ以上の分析出力または結果につながる。ある実施形態においては、オフライン評価中、ファイルからデータセットをドラフトデータモデルへとロードするよりも、入力データを、試験ツール、環境、またはプラント等のストリーミングオフラインデータソースから受信し得る。更に、下でより詳細に説明されるように、オフラインデータモジュールまたはドラフトデータモデルは、それが開発されている間に、漸次評価してもよい。DDE User Interface Application においては、オフラインデータブロック及びオフラインデータモジュールを評価するためのプラットフォームまたはインフラストラクチャは Data Analytics Services であり、これについては後の節において記載する。

20

30

40

【0122】

オフラインデータモジュール（またはその一部）が評価された後、ユーザは、評価の結果をレビュー及び精査することができ、ドラフトモデル（またはその一部）は、適宜修正され、満足な結果が得られるまで再評価され得る。ユーザは、ドラフトデータモデル/オフラインデータモジュールを完成させ、それをオンラインデータモジュールへと転換し、そのオンラインデータモジュールを実行または展開して、プロセス制御環境またはプラント 5 のリアルタイムオペレーションに起因して生成されている、リアルタイム、オンライン、またはストリーミングされるライブデータに基づいて動作させ得る。例えば、実行中または展開済のオンラインデータモジュールまたはデータモデルは、プロセスプラントのオンラインオペレーションに起因して連続的に生成されているリアルタイムデータに基づ

50

いて連続的に動作し得、オンラインデータモジュールまたはモデルは、それ自体で、リアルタイム出力データまたはその実行の結果を連続的に生成し得、このリアルタイム出力データまたは実行の結果は、ユーザインターフェイスに表示され、連続的にアップデートされ得る。展開されたオンラインデータモジュール及び対応する結果のオンライン実行またはオペレーションは、DDE User Interface ApplicationのDashboardにおいて監視され得る。これもまた、後の節においてより詳細に記載される。

【0123】

先に考察したように、DDE User Interface Application内において、データブロック、データモジュール（オンラインかオフラインに関わらず）、及びそれらの一部は、ワークスペースまたはDDE User Interface Applicationによって提供されるキャンバス上に、ダイアグラムまたはデータダイアグラムとして、ユーザに対して視覚的及び/またはグラフィカルに表され得る。概して、各データダイアグラムは、ワイヤ（例えば、線で表され得る）のセットによって相互接続されるデータブロック（例えば、二次元形状で表され得る）のセットを含む。DDE User Interface Applicationは、典型的には、オペレーションの2つのモードを含み（しかし、一部の実施形態においては、より多いかより少ない数のモードが含まれ得る）、これらの両方が、データモジュールと、データモジュールに含まれるデータブロックとを表すためにデータダイアグラムを活用する。

【0124】

オペレーションのモードのうちの1つは、本明細書においては交換可能に、「Data Analytics Studio」、「Data Studio」、または「Studio」と呼ばれる。典型的には、Data Studioは、データ分析モデルを設計及び開発するためにユーザによって活用される。加えて、Data Studioは、ユーザがオフラインモジュールをオンラインモジュールへと転換すること、ならびに完成したオフラインモジュールを対応するオンラインデータモジュールとして展開することを可能にする。Data Studioは作業領域または描画キャンバスを提供し、その上で、ユーザは、データモデルまたはその一部の視覚的グラフィカル表現であるデータダイアグラムへとブロックを作成及び接続することによって、データモデルを開発することができる。データモデルがData Studio内で開発されている間は、このデータモデルはドラフトデータモデルまたはオフラインデータモジュールと呼ばれる場合があり、ドラフトまたはプロトタイプと考えられ得る。オフラインデータモジュールダイアグラムは、その結果として得られるオンラインデータモジュールよりも多い数のデータブロック及び/または接続を含み得、これは、オフラインモジュールに含まれるブロック及び接続の一部は、モジュールの異なる部分においてデータを分析及び閲覧するために、例えば、データモデルの特定の部分が所望される通りに評価しているかどうか、及び/または十分に有用かつ/もしくは予測的なデータを提供しているかどうかを分析及び確認する目的で使用され得るためである。概して、データモデルのオフラインダイアグラムは、（1）生データセットを探索及びクリーンにし、かつ（2）分類、回帰、クラスタリング、次元縮小等の所望の分析オペレーションのために構築、調整、及び評価され得る、ワークフローの表現である。オフラインであるデータモジュールは、それらが開発されている間に、漸次または継続的に評価してもよい。オフラインデータモジュールまたはデータモデルの評価プロセスの結果は、Data Studioを介してユーザに通信される。

【0125】

一部の場合においては、オフラインデータモジュールの実行または評価は、完了するのに長い期間がかかり得る。これらの状況においては、オフラインデータモジュールの評価プロセスのステータス及び進行は、DDE User Interface Applicationのオペレーションの別のモードを介してユーザに通信され得、この別のモードは、本明細書においては交換可能に、「Data Analytics Dashboard」、「Analytics Dashboard」、または単に「Dashboa

10

20

30

40

50

rd」と呼ばれる。Dashboardは、典型的には、(i)評価されているオフラインデータモジュール(例えば、ドラフトデータモデル)、及び/または(ii)オンラインデータモジュール(例えば、データモジュールとして展開されている完成したデータモデル)であるモジュールを閲覧及び/または管理するために、ユーザによって活用される。Dashboardはまた、それぞれのダイアグラムまたは視覚的グラフィカル表現を用いて、評価中のドラフトデータモデル及び展開されたオンラインデータモジュールを表す。展開されたデータモジュールは「オンライン」であると呼ばれる。これは、展開されたモジュールが、プロセス制御システムまたはプラント5のオンラインまたはランタイムオペレーションに起因して生成されているリアルタイムデータに基づいて実行されているためである。概して、データモジュールのオンラインダイアグラムは、プロセス制御システムまたはプラント5内の1つ以上のデータソース(例えば、ストリーミングデータソース)にバインドされて、リアルタイム記述、予測、及び/または規定を作成し、かつ/あるいはランタイム中またはランタイムにおいて連続的にデータモデルを調整するワークフローの表現である。展開されているかオンラインであるデータモジュールは、それらがDashboardから明確に終結されるまで、継続的に実行され得る。

10

【0126】

A. データモジュールダイアグラム

データモジュールの例示的ダイアグラム(オフラインかオンラインかに関わらず)は、概して、複数のデータブロックと、様々なブロックと一緒に相互接続して1つ以上のデータフローパイプラインを形成するワイヤとを含む。先述したように、データブロックは概して、ユーザがデータセットに対して適用したいと欲する機能またはオペレーションの抽象化である。例えば、ある特定のブロックは、データ記憶またはディスク上のファイルからデータセットをロードしてもよく、別の特定のブロックは、データセット内の全ての欠測値(例えば、別のパラメータ/変数が測定値を有する時間に対応する時間において、測定値を有さない値)を取り除いてもよく、更に別の特定のデータブロックは、ランダムフォレスト分析を実施してもよい。典型的には、異なるデータブロックは、データセットに対して実施され得る異なる機能またはオペレーションに対応し、したがって、様々なデータブロックはそれぞれ、対応する種類または名称、例えば「Load」、「Query」、「Fill」、「Columns」、「PCA(Principal Component Analysis(主構成要素分析))」、「PLS(Partial Least Squares(部分最小二乗法))」、「Explore」、「Write」等を有し得る。

20

30

【0127】

各ブロックは、ゼロ以上のそれぞれのプロパティを有し得る。ブロックのプロパティのセット(一部のブロックの場合は、空のセットであり得る)は、同一のブロックの種類の全てのインスタンスが同一のプロパティのセットを有するように、それぞれ、そのブロックの種類に対応する。一部のブロックについては、プロパティのデフォルト値は、DDE User Interface Applicationによって提供され得、一部のブロックについては、ユーザが、プロパティ値のうちの1つ以上を入力及び/または修正することができる。データブロック定義は、あるデータブロックの種類の、それぞれのプロパティ及び任意のデフォルトプロパティ値を定義する(一部のブロックについては、他の情報も)。データブロック定義は、DDE User Interface Applicationによって提供されるデータブロック定義ライブラリに記憶される。概して、データブロック定義ライブラリは、複数のデータ分析ユーザまたは技術者が、ライブラリによって提供されるリソースを用いて、データを同時に開発及び/または探査できるように、DDE User Interface Applicationの全てのオープンインスタンスに対して利用可能である。

40

【0128】

各ブロックは、1つ以上の他のブロックまたはデータソースからブロック内に流入するデータ(存在する場合)を特定する、0、1、または複数の入力コネクタを有する。加え

50

て、各ブロックは、ブロックから流出する（ならびに、潜在的には、1つ以上の受信側ブロックに、それらそれぞれの入力コネクタを介して流入する）データ（存在する場合）を特定する、0、1、または複数の出力コネクタを有する。様々なブロックの入力と出力との間の相互接続は、データモジュールダイアグラムにおいてはワイヤで表される。単純な計数装置の値から、それぞれが目的コードに対して数百万の値を含むデータフレームに至るまで、任意の種類のデータが、任意のワイヤを通じて流れるか、あるいは移送され得る。

【0129】

加えて、各ブロックは状態を有する。例えば、ブロックが初めて作成されるとき、そのブロックは、「構成」または「構成中」または「未構成」状態である。ブロックが構成された後、このブロックは「構成済」状態へと移る。オフラインブロックが評価されている間、そのオフラインブロックは「評価」状態である。オフラインブロックが評価された後、このブロックは、評価「成功」状態または評価「失敗」状態のいずれかへと移る。オンラインブロックが構成されている場合、またはデータソース及び/もしくはデータコンシューマにバインドされている場合、その状態は、「構成中」または「バインド中」である。オンラインブロックがプロセス制御システムに展開され、ライブデータに基づいて運転を始めた後、その状態は、「展開済」または「実行中」である。無論、他の状態もまたあり得る。典型的には、各ブロックは、その現在の状態を示す視覚的インジケータを含む。加えて、各データモジュールは、全体として、その現在の状態を示すインジケータを有し得る。

【0130】

可能性のある、様々なデータブロックの特色及び相互接続を例証する、例示的データダイアグラム200が図4Aに示される。この例示的データダイアグラム200は、ワイヤ205を介してFillNaNブロック202bに接続される、LoadDataブロック202aを含む。LoadDataブロック202aの表現は、その現在のブロック状態の指標208aと、そのブロックの種類または名称の指標210aと、ブロックの評価の結果の指標212aと、出力コネクタ215aとを含み、この出力コネクタを經由して、LoadDataブロックの評価の結果として生成されるデータの少なくとも一部が、ワイヤ205を介して受信側（このシナリオにおいては、FillNaNブロック202b）に送達される。

【0131】

FillNaNブロック202bは、LoadDataブロック202aからワイヤ205を通じて流れてくるデータを受信する、入力コネクタ218bを含む。LoadDataブロック202aと同様に、FillNaNブロック202bは、その現在のブロック状態の指標208bと、そのブロックの種類または名称の指標210bと、その評価の結果の指標212bと、出力コネクタ215bとを含み、この出力コネクタを經由して、FillNaNブロックの評価の結果として生成されるデータの少なくとも一部が、ワイヤ220を介して受信側またはコンシューマ（図示せず）に送達される。

【0132】

図4Aにおいては、LoadDataブロック202aは入力コネクタを有さないが、これは、LoadDataブロック202aが、データダイアグラム200の任意の他のデータブロックから出力されるデータを受信しないためであることに留意されたい。代わりに、LoadDataブロック202aは、1つ以上のデータソースからの1つ以上の入力データセットをロードまたは獲得するように構成され得る。これらのデータソースは、データファイル、データソース（例えば、Seqシステム）、リレーショナルデータベース、または非リレーショナルデータベース等のオフラインデータソースを含んでもよく、ならびに/あるいはデータソースは、分散型データエンジン202xによって生成されるデータストリーム等のオンラインまたはストリーミングデータソースを含んでもよい。

【0133】

B . Data Analytics Studio

先述したように、DDE User Interface Applicationのオペレーションのモードのうちの一つがData Analytics Studioである。Data Analytics Studioは、データモデルを設計、開発、閲覧、及び探査するために、ユーザによって活用され得る。図4Bは、Data Analytics Studio 240によって呈示される例示的ユーザインターフェイスを描写し、これは、ナビゲーションバー242と、ワークスペースまたは描画キャンバス245とを含む。ナビゲーションバー242は、制御部及びインジケータを提供し、これらの制御部及びインジケータを介して、ユーザは、オフライン及びオンラインデータモジュールを管理することができる。これは、例えば、新しいオフラインデータモジュールの作成、現在開放されており、キャンバス245に示されているオフラインまたはオンラインデータモジュールの特定、現在開放されており、キャンバス245に示されているデータモジュールのステータス（例えば、オフラインまたはオンライン）の容易な閲覧、現在開放されており、キャンバス245に示されているオフラインデータモジュールの保存/記憶、オフラインモジュールのオンラインデータモジュールへの転換、データモジュールのオフラインデータダイアグラムとオンラインデータダイアグラムとの間での閲覧の切り替え、オフラインデータモジュールの評価、オンラインデータモジュールの展開、他のデータモジュールへのブラウズ、及び他のそのようなモジュール管理機能等のアクションの、ユーザによる実施が可能になったことによるものである。したがって、Data Analytics Studio 240は、多数のユーザ制御部及びインジケータ248a~248nをその上に含み、限定されるものではないが：

- ・ユーザによる他のデータモジュールの検索及びブラウズを可能にする、モジュールナビゲーション制御部248a；
- ・キャンバス245で現在開放されているデータモジュールの識別子248b；
- ・キャンバス245で現在開放されているデータモジュールのビューがオフラインビューであるかオンラインビューであるかを示す、1つ以上のインジケータ248c、248d；
- ・ユーザが、キャンバス245で現在開放されているデータモジュールのオンラインビューとオフラインビューとを切り替えることができる、1つ以上の制御部248e、248f；
- ・ユーザが、キャンバス245で現在開放されているデータモジュールのプロパティを閲覧及び/もしくは定義することができる、ユーザ制御部248g；
- ・ユーザが、現在開放されているデータモジュールを保存することができる、ユーザ制御部248h；
- ・ユーザが、現在開放されているデータモジュールの少なくとも一部分を評価することができる、ユーザ制御部248i
- ・ユーザが、現在開放されているデータモジュールを展開することができる、ユーザ制御部248j
- ・現在開放されているモジュールのオペレーションのステータスを示す、インジケータ248k；ならびに/または
- ・1つ以上の他のユーザ制御部及び/もしくはインジケータ（図示せず）等が挙げられる。

【0134】

図4Bはまた、ユーザ制御部248mを例証しており、これを介して、ユーザは、データブロック定義ライブラリ（図4Bには図示せず）からブロック定義を閲覧、選択でき、かつ/あるいはデータブロック定義ライブラリに追加できる。図4Bに示される実施形態においては、ライブラリユーザ制御部248mは、キャンバス245上に配置されるように示されているが、他の実施形態においては、該制御部248mは、ナビゲーションバー242上または任意の所望の場所に位置付けることができる。

【0135】

10

20

30

40

50

実際、例示的 Data Analytics Studio 240 に示されるユーザ制御部及びインジケータの数、種類、場所/位置付け、形態因子、配置等は、多くの可能な実施形態のうちの一つであるに過ぎない。より多くの、またはより少ない数及び/または種類のユーザ制御部及び/またはインジケータが含まれてもよい。そのようなユーザ制御部及び/またはインジケータの異なる場所/位置付け、ならびに異なる形態因子、配置等が活用されてもよい。一部の実施形態においては、ナビゲーションバー 242 は省略されてもよく、ユーザが制御部にアクセスするための別のメカニズムが提供され得る（例えば、ポップアップウィンドウ、ドロップダウンメニュー等）。

【0136】

1. Data Analytics Studio - オフラインデータモジュール

Data Analytics Studio 240 においては、描画キャンバスまたはワークスペース 245 は、オフラインデータモジュールを開発、定義、及び評価することができる領域である。例えば、描画キャンバスまたはワークスペース 245 において、ユーザは、DDE User Interface Application のブロック定義ライブラリに（例えば、ユーザ制御部 248m を介して）アクセスして、その中に記憶されている様々なデータブロック定義を選択し、選択したブロック定義を相互接続（例えば、ワイヤでつなげる）して、データモデルのデータダイアグラムを形成することができる。加えて、描画キャンバスまたはワークスペース 245 において、ユーザは、特定のデータブロックインスタンスのプロパティの編集；評価の進行及びその結果を閲覧することを含む、一部分のドラフトデータモデルの評価；ならびに/またはオフラインデータモジュールに関する他のアクションの実施を行うことができる。

【0137】

先に考察したように、データダイアグラムにおいて活用され得る各データブロックは、データブロックの種類のプロパティに基づく。すなわち、所与の種類のブロックの特定のインスタンスは、その所与の種類のブロック定義に従うプロパティのセットを有するが、その特定のインスタンスは、例えば、オブジェクトクラスとオブジェクトクラスのインスタンスとに類似する様式で、同一の所与の種類のブロックの他のインスタンスとは異なる場合がある。これもまた先に考察したように、ブロック定義は、ブロック定義ライブラリに記憶されており、このブロック定義ライブラリは、キャンバス 245 上に置かれ得るユーザ制御部 248m を介してアクセスされる（例えば、表示または非表示で切り替えられる）。

【0138】

データブロック定義ライブラリ、データブロック定義、データモジュール、データブロック、データブロックインスタンス、及びワイヤ間の例示的關係性 250 の例証が図 4C に示される。図 4C に示されるように、ブロック定義は、DDE User Interface Application によって提供されるブロック定義ライブラリ 252 を通じて利用可能となる。一部の実施形態においては、異なるユーザ及び/またはユーザのグループに対して、特定のブロック定義ならびに/または他のデータ分析ライブラリリソース（及び/もしくはその一部）への異なるアクセス権（例えば、読み取り専用、読み書き等）が与えられ得る。

【0139】

ライブラリ 252 に記憶されているブロック定義 255 は、データモジュール 260 内で使用されるブロック 255 のインスタンス 258 を作成するために活用され得る。ブロック定義 255 の 1 つ以上のプロパティ 262 が、ブロックインスタンス 258 について具体的に定義され得る。ブロックインスタンス 258 は、ブロックインスタンス 258 へのデータが受信される、1 つ以上の入力コネクタ 265 を含むように構成または設計され得、また、ブロックインスタンス 258 は、1 つ以上の出力コネクタ 270 を含むように構成または設計され得、この出力コネクタを介して、出力データ（例えば、入力コネクタ 265 を介して受信した入力データに基づいて動作するブロックインスタンス 258 の評価から結果として生じるデータ）が、例えば、別のブロックインスタンス 258（書き込

10

20

30

40

50

むかまたは記憶するため)、ユーザインターフェイス等に提供される。ブロックインスタンス258の各入力コネクタ265は、データモジュール260の1本以上のワイヤ272を通じてデータを受信し得、ブロックインスタンス258の各出力コネクタ270は、データモジュール260の1本以上のワイヤ272を通じてデータを提供し得る。データモジュール260の各ワイヤ272は、特定のブロックインスタンス258の特定の出力コネクタ270と、別の特定のブロックインスタンス258の特定の入力コネクタ265との間で相互接続を提供し、それによって、2つのデータブロックインスタンスの間におけるデータの移送を可能にする。

【0140】

ある実施形態においては、ブロック定義は、機能またはアクションのカテゴリによって、ライブラリ252内で整理される。カテゴリは、サブカテゴリを含んでもよく、サブカテゴリは、サブサブカテゴリを含んでもよく、以降も同様である。例証的であるが非限定的である例においては、ブロック定義ライブラリ252は、少なくともブロック定義の5つのカテゴリ: Data Sources、Filters、Transformers、Visualizations、及びData Consumersを含む。

10

【0141】

Data Sourcesカテゴリに含まれるブロック定義は、典型的には、データモジュールの動作の基となる入力データを提供するデータソースに関連する、様々なアクションを定義する。例えば、Data Sourcesカテゴリのブロック定義としては、入力データセットを生成する「CreateDataSet」、オフラインまたはオンラインデータソースから既存のデータセットをロードまたは獲得する「LoadDataSet」、データベース(プロセス制御データベースまたは分析データベース等)からデータをロードまたは獲得する「LoadDB」、入力データセットをより長期的なデータストレージにヒストライズする「SaveDataSet」(例えば、その作成後に、入力データセットの試験のため等)等が挙げられ得る。更に、Data Sourcesカテゴリの一部は、1つ以上のサブカテゴリを含み得る。例えば、CreateDataSet及びLoadDataSetカテゴリは、各々、データセットが生成された特定のデータセットフォーマット及びまたは環境、例えばCSV(コンマ区切り)、Raspberry Pi、Seeq等についてのそれぞれのサブカテゴリを含み得る。概して、オフラインData Sourcesブロックは、データファイル、プロセス制御データベース、分析データベース等の静的データソースを獲得またはそれにアクセスするように構成される。他方で、オンラインData Sourcesブロックは、典型的には、オンラインData Sourceブロックの特定のインスタンスを、プロセスプラント内の1つ以上のオンラインデータソースと関連させるバインディング定義を伴って構成される。構成されたオンラインData Sourceのコンパイルは、オンラインData Sourceブロックが、それがバインドされている1つ以上のオンラインデータソースによって生成されたデータをストリーミングで受信するように、このバインディングをインスタンス化する。

20

30

【0142】

データソースカテゴリは、選択されたデータを1つ以上のデータソースからサーチまたは獲得することに関するブロック定義、例えば「QueryDataSource」または「Query」を有し得る。Queryブロックは、オフラインデータソース及び/またはオンラインデータソースに基づいて動作し得る。概して、Queryブロック定義は、データモジュールが、特定の種類のデータのアイデンティティ(例えば、列、タグ、または他の好適な識別子によって示される)を照会、獲得、または要求することを可能にし、かつ/あるいは特定の時間間隔(複数の離れた時間間隔でもよい)中に生成されたデータのみ照会、獲得、または要求することを可能にする。なおも更に、Queryブロックは、データが取り込まれている、または記憶されているフォーマット及び/または環境に関係なく、任意の種類 of データを照会することができる。クエリブロックの更なる詳細は、本出願の後の節において提供される。

40

50

【0143】

Filtersカテゴリに含まれるブロック定義は、典型的には、データセットに実施することができる様々なフィルタリング技法を定義する。例えば、Filtersカテゴリのブロック定義としては、「Clip」、「HighPass」、「LowPass」、「SGF（例えば、サビツキーゴレイフィルタリングの場合）」、「Exponential」、「Averaging」、「Wavelet」等が挙げられ得る。

【0144】

データブロック定義ライブラリ252の「Transformers」カテゴリに含まれるブロック定義は、典型的には、データセットまたはフィルタされたデータセットの内容を操作、分析、及び/または別様に転換する様々な技法を定義する。例えば、Transformersカテゴリとしては、入力データセットのクリーニングに対応するサブカテゴリのデータブロック定義、例えば、「FillNaN」（例えば、数字ではないデータセットのエントリを埋める）、「RmvOutlier」（例えば、異常値データを取り除く）、「CorrectBad」（例えば、入力データセット内で検出された不良データを修正する）、「ExcludeBad」（例えば、検出された不良データを除外する）等が挙げられ得る。加えて、Transformersカテゴリは、「Scale」、「Shift」、「Split」、「Merge」、「CenterNormal」、「DownSample」、「TimeOffset」、「Columns」等の、入力データセットの内容の操作に対応するサブカテゴリのデータブロック定義を含み得る。更に、一部の実施形態においては、Transformersカテゴリは、入力データセット内のデータの整合に対応するサブカテゴリのブロック定義、例えば、「TimeDelay」、「LagCorrect」等を含み得る。

【0145】

Transformersカテゴリは、「Sensitivity」、「Clusters」、「RandomForest」、「CBP（例えば、条件付きベイズ確率分析）」、「KMeans」、「FourierTransform」、「FastFourierTransform」、「PLS」、「PCA」等の、入力データセットを分析して知識を得て、それらの内容について学習することに対応する、サブカテゴリのデータブロック定義を含み得る。Transformersカテゴリの一部のサブカテゴリは、1つ以上のサブサブカテゴリを含み得る。例えば、Transformersカテゴリの「PCA」サブカテゴリは、「PCA_NIPALS」（PCA及び非線形反復部分最小二乗法）、「PCA_SVD」（PCA及び特異値分解）、「PCA_Test」等の、様々な主構成要素分析技法に対応するサブサブカテゴリのデータブロック定義を含み得る。

【0146】

Visualizationsカテゴリのブロック定義は、典型的には、データモジュールによって生成された出力を呈示するための様々な技法を定義する。例えば、Visualizationsカテゴリは、「LineGraph」、「Chart」、「BarGraph」、「ScatterChart」、「Histogram」、「DataGrid」、「DataCloud」、「Animation」等の、グラフィカル及び/または別様な視覚的表現に対応するサブカテゴリを含み得る。Visualizationsカテゴリは、「RoundSigFig」等の、特定のグラフィカル表現について出力データを準備することに対応するサブカテゴリを含み得る。

【0147】

Data Consumersカテゴリのブロック定義は、典型的には、データモジュールによって生成された出力データの様々なコンシューマまたは受信側に関するアクションを定義する。例えば、Data Consumersカテゴリは、ファイル、データベース、または他の静的記憶領域への出力データの書き込みまたは保存を引き起こす、サブカテゴリ「Write」を含み得る。そして、このサブカテゴリ「Write」は、出力データがどこにどのようにして書き込まれたり保存されたりするのかを示す、様々な種類の場所、データベース、データベースフォーマット、ファイルフォーマット等に対応する

、それぞれのサブカテゴリを有し得る。Data Consumersカテゴリは、出力データのストリーミング（例えば、分析データネットワーク112を介する）、またはデータのサブスクライバもしくは受信側（例えば、分散型データエンジン102x、オンラインデータモジュール、プロセス制御システムによって実行されているアプリケーション、ユーザインターフェイス等）への別様な公表もしくは送達を引き起こす、サブカテゴリ「Publish」を含み得る。一部の実施形態においては、オフラインのPublishブロックをそのオンラインバージョンに転換するために、Publishデータブロックは、ライブデータコンシューマ（例えば、データモデルの出力のサブスクリプションを有する、オンラインデータモジュール、実行中アプリケーション、プロセス制御デバイス、要素、または構成要素等）にバインドされるように構成され、バインドされる。ある実施形態においては、Data Consumersカテゴリは、サブカテゴリ「Convert2Control」を含み得、このサブカテゴリは、出力データを、従来型のプロセス制御通信ネットワークが理解するフォーマットの信号へと変換し、また、その信号を、従来型のプロセス制御通信ネットワークを介して、プロセス制御エンティティ、要素、デバイス、または構成要素へと送達して、プロセスプラント5における変化をもたらすか、あるいは引き起こす。サブカテゴリ「Convert2Control」は、例えば、変化信号（例えば、制御ループ入力、制御ループ構成、メッセージ優先度、パラメータ値等）の潜在的な受信側である、プロセスプラント5内の様々な従来型のプロセスプロトコルフォーマット（例えば、HART、PROFIBUS、WirelessHART等）及び/または様々なターゲットに対応する、サブサブカテゴリを含み得る。例えば、Convert2Controlブロックは、Convert2Controlブロックの出力を受信し、そのオペレーションを適宜修正する、プロセスプラント5内の特定のプロセス制御エンティティ、要素、デバイス、または構成要素にバインドするように構成される。

10

20

【0148】

データブロック定義ライブラリ252は、その中に他のデータブロック定義を含み得、それらの各々はカテゴリに属す場合または属さない場合がある。例えば、「Explore」ブロックは、オフラインモードでの、ユーザによる、以上の他のデータブロックによって生成された出力データの探査を可能にする。例えば、Exploreデータブロックは、その入力コネクタにおいて複数の異なるデータワイヤを受容し得、例えば、2つの入力データセットを経時的に整合し、1つの線グラフを別の線グラフの上に重ねることによって、各それぞれのデータセットの情報を他方のデータセットの情報の隣に表示させる、ヒストグラムを作成することによって、両方の入力データのセットを比較的な様式で含む視覚的表現を生成し得る。Exploreブロックは、ユーザが、例えば、x軸及び/またはy軸の縮尺を増加/減少させること、図表に表示される統計データを整列及び/またはフィルタリングすることによって、その出力のビジュアライゼーションを操作することを可能にする。Exploreブロックは、その機能が主として、ドラフトデータモデルによって生成された出力データをユーザが探査及び理解するのを可能にするため、オンラインの相対物を有さない場合があることに留意されたい。

30

【0149】

ユーザが新しいデータモデルを開発する場合、ユーザは、データブロック定義ライブラリ252を呼び出すことができる（例えば、ユーザ制御部248mを介して切り替えることによって）。ユーザは、ライブラリ252から所望のデータブロックの定義を、キャンバス245の所望の場所にドラッグアンドドロップすることで、キャンバス245に所望のデータブロックを追加することができる（それによって、キャンバス245上に描写される、開発中のデータモジュールにそのデータブロックを追加することができる）。（無論、ドラッグアンドドロップは、選択、位置付け、及び特定のデータブロックのデータモジュールへの組み込みを示す1つの方法に過ぎず、それらを行う多数の他の好適な方法が可能である。）ドラッグアンドドロップのオペレーションの後、選択したブロック定義はドラフトデータモジュールと関連付けられ、ドラフトデータモデルについてのそのデータ

40

50

ブロックのそれぞれのインスタンスが作成及び命名される。図4Dに例証される例示的シナリオにおいて、ドラフトModule A1は、開発プロセスの途中であり、その対応するドラフトのオフラインデータダイアグラムが、Data Studioキャンパス245上に示されている。インジケータ248cによって示されるように、Module A1は「オフライン」ステータスを有する。ユーザは既に、ブロックA1-B1、A1-B2、及びA1-B3をドラフトModule A1に追加しており、それらをワイヤA1-W1及びA1-W2を用いて接続している。ユーザは、ライブラリ252を制御部248mを介して呼び出し、ライブラリからブロックB4を選択し、ブロックB4をキャンパス245上にドラッグアンドドロップしており（破線で示されるように）、ブロックB4のインスタンスを「A1-B4」と命名するプロセスの途中である。ブロックB4のインスタンスを命名した後、ユーザは、ワイヤで、A1-B4をドラフトModule A1の1つ以上の他のデータブロックと相互接続し得る。例えば、ユーザは、A1-B4の入力コネクタ上でカーソルをホバリングし、クリックして、選択した入力コネクタへの新しいワイヤ接続を作成し得る。次いで、ユーザは、キャンパス245上の別のブロックインスタンスの所望の出力コネクタ上でクリックして、選択した入力コネクタを所望の出力コネクタに相互接続する新しいワイヤを作成し、それによって2つのブロックインスタンスを相互接続し得る。別の例においては、ユーザは、別のブロックインスタンスの出力コネクタ上でカーソルをホバリングし、クリックして、選択した出力コネクタへの新しいワイヤ接続を作成し、A1-B4の所望の入力コネクタ上でクリックして、相互接続するワイヤを作成することができる。無論、任意の好適なユーザ制御メカニズムが、ブロックインスタンス間におけるワイヤ相互接続を作成するために活用され得る。

【0150】

Data Studioキャンパス245において、ユーザは、特定のブロックインスタンスについて、ブロックの所与のプロパティの値を修正することができる。例証のため、図4Eは、Data Studioキャンパス245の一部分を描写しており、ここではデータモジュールB1が開発されている。ドラフトModule B1に含まれる2つのブロックインスタンス、すなわちB1-FillNaN及びB1-PCAが、キャンパス245上に位置付けられ、相互接続されている。B1-FillNaNは、図4Eには示されていないデータソースからその入力データセットを受信し、B1-PCAは、2つの出力、「Model」280a及び「Scores」280bを、それぞれのコンシューマまたは受信側ブロック（これも図示せず）に提供する。ユーザは、彼または彼女が、B1-PCAブロックインスタンスに対応するブロック定義プロパティの値の修正を所望することを示している（例えば、B1-PCAブロックインスタンス上でホバリングすることによって、B1-PCAブロック上でダブルクリックすることによって、あるいは任意の他の好適なユーザ制御メカニズムを用いることによって）。ユーザ制御部のアクティブ化にตอบสนองして、PCAブロック定義に対応する定義されたプロパティの一覧表を含むウィンドウ282が現れている（例えば、フローティングダイアログボックス、ポップアップウィンドウ、または何らかの他の好適な表現として）。次いで、ユーザは、ウィンドウ282を介して、B1-PCAの様々なブロックプロパティの値を、所望に応じて修正することができる。

【0151】

先に考察したように、各データブロックのプロパティは（存在する場合）、ブロック定義ライブラリ252に記憶されている、そのそれぞれのブロック定義によって定義されている。ブロック定義及びそれらそれぞれのプロパティの例証的（しかし非限定的である）な例が、図4F～4Hに示される。ブロック定義ライブラリ252は、図4F～4Hに示される例示的ブロック定義のうちの1つ以上を含んでもよく、これらの例示的ブロック定義をどれも含まなくてもよく、ならびに/または他のブロック定義を含んでもよいことが理解される。更に、図4F～4Hに示される各例示的ブロック定義のプロパティの数、種類、及びデフォルト値もまた、本明細書に記載される例とは相違してもよい。

【0152】

10

20

30

40

50

図4Fにおいては、「LoadDB」データブロック285aのデータブロック定義テンプレートが、例えば、ユーザがライブラリ252からLoadDBブロック定義をキャンパス245上にドラッグし、続いて、彼または彼女が、LoadDBデータブロック285aのブロックプロパティ値を閲覧及び/または修正して、ブロック285aの特定のインスタンスを作成するのを所望することを示した結果として、DDE User Interface Applicationによって呈示されている。データモデルのオフラインダイアグラム内でLoadDBブロック285aが実施するアクションまたは機能としては、データモデルの運転が基づく特定のデータセットをロードすること(または別様に、そのデータセットの内容を獲得すること)が挙げられる。例えば、特定の入力データセットは、データベースまたはファイルからロードまたは獲得され得る。ユーザは、例えば、ドロップダウンメニュー、(例えば、プロセス制御データベース、またはプロセス制御システムと関連付けられる他のデータベースの)サーチ及び選択、自由形式テキスト入力等を介して、所望の名称、インジケータ、または値をDatabaseNameプロパティフィールド285b及びDataSetプロパティフィールド285cに入力することで、データモデルにロードされる特定の入力データセットを示すことができる。例えば、ユーザは、ドロップダウンメニューから所望のDatabaseName285bを選択することができ、所望のデータベース285bのその選択が、選択/特定されたデータベース285bのデータセット285cのみを含む、DataSetフィールド285cの対応するドロップダウンメニューの提供につながる。LoadDBブロック285aはまた、Plotプロパティフィールド285dも含み、この実装形態においては、Plotプロパティフィールドはブール値フラグであり、その値は、データベース285bからのデータセット285cのプロットが、LoadDBブロックインスタンスが評価されるときに、生成/呈示されるか否かを示す。Plot285dが「はい」に設定されている場合、ブロック評価の後に、ロードされたデータセット285cのプロットが、例えば、LoadDBブロックインスタンスのView Block Results領域212において、キャンパス245に示されるLoadDBブロックインスタンスのグラフィカル表現に表示される。LoadDBブロック285aの場合、Plotプロパティ285dは任意選択であり、その値はデフォルトでは「無し」となる(例えば、ブロックが評価されるとき、ロードされた入力データセットのプロットは生成/呈示されない)。ユーザが全ての所望の値をLoadDBデータブロック285aのプロパティフィールド285b、285c、及び285dに入力した後、ユーザは、修正したプロパティ値を保存して、それにより、LoadDBデータブロックテンプレート285aを、ユーザが作成/開発しているデータモデルにおいて使用されるLoadDBデータブロック285aの特定のインスタンスを作成するように構成することができる。ユーザは、例えば、キャンパス245に示されるブロックインスタンスのブロック名称フィールド210に、所望の名称(例えば、「LoadDB-1」)を入力することで、特定のインスタンスを命名することができる。

【0153】

図4Gは、DDE User Interface Applicationによって呈示される、「Columns」データブロック286aのデータブロック定義テンプレートを例証する。Columnsデータブロック286aの特定のインスタンスは、SelectedColumnsプロパティフィールド286bのそのプロパティ値によって特定されるように、ロード/獲得されたデータセットからのデータの選択されたグループ(本明細書においては、データの「列」または「タグ」で参照されるデータと呼ぶ)を抽出または獲得する。例えば、Columnsデータブロック286aは、その入力コネクタを介して、例えばLoadDBデータブロック285aのインスタンスから、データセット(または、ポインタ、インジケータ、もしくはデータセットに対する他の参照等の、その何らかの指標)を受信する。Columnsデータブロック286aのSelectedColumnsプロパティ286bは、ユーザが、データモデルの他のブロックがそれに基づいて動作することを所望する、入力データセットの1つ以上の列、タグ、または

10

20

30

40

50

他の部分を、ユーザが選択することを可能にする。例えば、図4Gに示されるように、ユーザは、Data set Aのタグのリストをスクロールして、所望のタグをハイライト表示させるプロセスの途中である。典型的には（必ずしもそうではないが）、選択された列またはタグは、入力データセットに含まれる全ての列またはタグのサブセットである。ユーザは、列またはタグ選択286b（また、存在する場合は、他のブロックプロパティ）を保存して、それにより、Columnsデータブロック286aを、特定のインスタンス、例えばユーザが作成/開発しているデータモデルにおいて使用される特定のインスタンスを作成するように構成することができる。ユーザは、例えば、そのブロック名称フィールド210に所望の名称を入力することで、Columns286aの特定のインスタンスを命名することができる。

10

【0154】

図4Hは、DDE User Interface Applicationによって呈示される、「PCA__NIPALS」データブロック287aのデータブロック定義テンプレートを例証する。PCA__NIPALSデータブロック287aは、データのセットに対して、主構成要素分析（PCA）と、非線形反復部分最小二乗法とを実施する。例えば、PCA__NIPALSデータブロック287aは、その入力コネクタを介して、Columnsデータブロック286aのインスタンスによって決定された、データの列もしくはタグ（または、そのインジケータもしくはそれに対する参照）を受信し得る。別の例においては、PCA__NIPALSデータブロック287aは、その入力コネクタを介して、LoadDB285aデータブロックによって、データモデルによる使用のためにロードされた、全体のデータセット（または、そのインジケータもしくはそれに対する参照）を受信し得る。PCA__NIPALSブロック287aのテンプレートは、所望される場合、PCAの構成要素の数287b及び信頼水準287cのユーザによる選択を可能にする。プロパティフィールド287b、287cの両方がPCA__NIPALSブロック287aについては任意選択であり、デフォルト値（例えば、PCA__NIPALSのブロック定義において定義される）またはゼロに設定され得る。図4Hに示される例示的シナリオにおいては、ユーザは、PCA__NIPALSの現在のインスタンスが、2つの構成要素287bと、少なくとも95%の信頼水準287cとを生成するように構成されることを示している。PCA__NIPALSデータブロック287aテンプレートは、ユーザが作成/開発しているデータモデルにおいて使用される特定のインスタンスを作成するように構成することができ（プロパティ値の修正を伴って、または伴わずに）、また、ユーザは、例えばそのブロック名称フィールド210に所望の名称を入力することで、特定のインスタンスを命名することができる。

20

30

【0155】

一部の種類のデータブロックは、前のブロックによって生成された出力データに基づいて動作する、コンシューマまたは受信側データブロックである。これらの種類のコンシューマデータブロックは、これらのコンシューマデータブロックを構成するために、評価されるべき前段階のブロックを必要とする場合がある。例えば、LoadDBブロック485aのインスタンスは、Columnsブロック486aのインスタンスにデータを提供するようにワイヤでつなげられ、LoadDBブロックインスタンス485aの評価はデータセットを提供し、このデータセットから、ユーザはColumnsブロックインスタンス486aを構成するための特定の列またはタグを選択し得る。

40

【0156】

ここでオフラインデータブロック及びデータモジュールの評価に注目すると、オフラインデータモジュールは、それが開発されている間に繰り返し評価することができ、評価するために完全に完成している必要はない。したがって、オフラインデータモジュールの評価は、非同期的に実施することができ、それにより、ユーザは、ドラフトデータモジュールの一部の評価、評価に基づくドラフトデータモジュールの修正、再評価、再修正、再々評価等を行うことができる。例えば、ユーザは、データモジュールの開始データブロックを作成し、その単一のデータブロックを評価し、次いで第2のデータブロックを第1のデ

50

ータブロックに接続し、第2のデータブロックのみを評価し（あるいは、第1及び第2のデータブロックの両方を全体として評価し）、更に4つのデータブロックを追加し、新規に追加したデータブロックのみを評価する（あるいは、6つのデータブロック全てを全体として評価する）等してもよい。すなわち、所望の数のデータブロックインスタンス及び/または関連付けられるワイヤが、オフラインまたはドラフトデータモジュールの作業中のデータダイアグラムに追加されたとき、キャンバス245に表されるモジュール（またはその一部）は、例えば、ナビゲーションバー242の評価ユーザ制御部248iをアクティブ化することによって評価できる。例えば、ユーザが、キャンバス245上に呈示されるブロック及びワイヤの全体を評価することを所望する場合、ユーザは、ユーザ制御部248iを単に評価し得る。一方で、ユーザが、キャンバス245上に呈示されるブロック及びワイヤのサブセットのみを評価することを所望する場合、ユーザは、所望のブロック及び/またはワイヤを選択し（例えば、クリック、投げ縄ツール、または他の好適なメカニズムで）、次いで、制御部248iをアクティブ化して、選択したブロック及びワイヤのセットを評価させ得る。

10

【0157】

図4Iは、オフラインデータダイアグラム288がどのように評価され得るかの例を例証する。オフラインデータダイアグラム288が作成または開発されている間、Data Studioキャンバス245上に示されるそのブロック及び相互接続の表現は、伝送ファイルまたはドキュメント290（本明細書においては、「構成ファイル」とも呼ばれる）に記憶され、この伝送ファイルまたはドキュメントは、JSON（Java Script Object Notation）または任意の他の所望のフォーマット等の、軽量データ交換フォーマットのものであり得る。したがって、グラフィカルオフラインデータダイアグラム288は、伝送ストレージファイルまたはドキュメント290にバインドされる。ユーザが、彼または彼女がオフラインデータダイアグラム288またはその一部が評価されるのを所望していることを示すとき（例えば、ユーザ制御部248iの活性化によって）、伝送ファイルまたはドキュメント290は、ターゲット実行環境の言語へとコンパイルされ、それによって、データダイアグラム288にとって実行可能なコード292が生成される。ある実施形態においては、伝送ファイルまたはドキュメント290は、複数の部分へと分割され得、これらの部分の各々が、異なるターゲット環境において実行可能である異なるターゲット言語へとコンパイルされる（例えば、異なるターゲット言語の実行可能なコード292の複数のチャンクまたはセグメントが、伝送ストレージファイルまたはドキュメント290から生成され得る）。オフラインデータダイアグラム288の実行可能なコード292が生成された後、コード292の実行を管理及び調整するためのプロセスが作成され、これは、異なるターゲット環境にまたがる場合またはまたがらない場合があり、ならびに/あるいは異なるプロセッサによって実施される場合または実施されない場合がある。

20

30

【0158】

オフラインデータダイアグラム285の様々なブロックが実行または評価されている間に、プロセスは、Data Studioに対してフィードバックを送信し得る。例えば、プロセスからのフィードバックに基づき、Data Studioはオフラインダイアグラム285の各データブロックのそれぞれのステータスインジケータ208をアップデートして、ブロックがコンパイルしているか、評価しているか、評価が成功裏に終了したか、または評価が不成功に終了したか（例えば、失敗したか）を示す。実際、概して、図4Iに例証されるコンパイル及び実行プロセスを通じて、フィードバックがData Studioに返され、ユーザインターフェイスに示され得る。例えば、プロセスからのフィードバックに基づき、Data Studioはモジュールステータスインジケータ248k及び/またはオフラインステータスインジケータ248cをアップデートして、モジュールが全体としてコンパイルを受けているか、評価を受けているか、評価が成功裏に終了したか、または評価が不成功に終了したかを示す。

40

【0159】

50

DDE User Interface Applicationにおいては、オフラインデータブロック、ダイアグラム、及びモジュールを評価するために使用されるインフラストラクチャ及びアクションは、Data Analytics Servicesによって提供される。例えば、評価ユーザ制御部248iのアクティブ化は、Data Analytics Servicesを発動させて、キャンパス245上で現在開発されているオフラインデータブロック、ダイアグラム、もしくはモジュール（またはそれらの一部）の評価を実施させ、これは、例えば、複数の異なるバックエンドプラットフォーム（例えば、複数の異なるターゲット言語、コンパイラ、プロセッサ、及び/またはターゲット環境）を活用し得る。したがって、DDE User Interface Applicationは、データブロック、ダイアグラム、またはモジュールを実装するために活用されるバックエンドプラットフォームと関わる（または、その知識を有する）必要性からユーザを保護する。すなわち、ユーザは、Data Studio及びDashboardの特色（例えば、データブロック定義ライブラリ252、キャンパス245、及び関連付けられるユーザ制御部）を、データダイアグラムを用いてデータモジュールを設計または操作するためだけでなく、それらの設計をリアルタイムかつバックエンドプラットフォームからは独立して展開するためにも活用することができ、このバックエンドプラットフォームは、Data Analytics Servicesによって、「秘密裏」に管理される。また、DDE User Interface Applicationアーキテクチャは、有利なことに、ユーザが直面するデータモジュール設計、評価、試験、及び展開機能性に影響を及ぼすことなく、追加的なバックエンドプラットフォーム及び/または構成要素が追加または削除されることを可能にする。

10

20

【0160】

Data Analytics Services及びその評価についてのサポートのより詳細な説明が、後の節において提供される。

【0161】

図4Aを参照すると、先に考察したように、オフラインモードにおけるデータモジュールの評価は、反復型プロセスであり得る。ユーザがドラフトデータモジュールに関してブロックを追加及び構成する際、ユーザは、追加されたブロックを評価することができ、各ブロックのステータスは、そのそれぞれのステータスアイコンによって反映される（例えば、図4Aに示されるインジケータ208）。ブロックの評価が不成功または失敗であった状況においては、ユーザがその失敗につながった任意の問題に対処することができるように、ユーザは、その失敗に関する詳細を獲得し得る（例えば、「失敗」ステータスインジケータ208を選択またはクリックすることによって）。ブロックの評価が成功した状況においては、ユーザは、例えばブロックのそれぞれのブロック結果閲覧アイコン212を選択またはクリックすることによって、ブロック評価の結果を閲覧することができる。ある実施形態においては、ユーザが「ブロック結果閲覧」212を選択したとき、モーダルダイアログまたはビジュアライゼーションダイアログが現れ得（例えば、ポップアップウィンドウ、フローティングダイアログボックス、または他の好適なフォーマットとして）、ユーザは様々なビジュアライゼーションを閲覧及び探査して、データブロック結果を見ることができる。

30

40

【0162】

実際、データブロックが実行または評価されるとき、データブロックは、（例えば、ワイヤ及び入力コネクタを介して）それに移送されたデータに基づいて動作するデータブロックの効果（複数可）をユーザが決定するのを補助するために有用であり得る、任意の種類の結果（複数可）を記憶し得る。記憶される結果（複数可）の種類は、典型的には、ブロックに対して特異的であり（また、一部の場においては、ブロックの特定のインスタンスに対して特異的であり）、ブロック開発者によって定義または決定され得る。これらの結果は、モーダルダイアログウィンドウまたはビジュアライゼーションダイアログウィンドウを介してユーザに示され得る。

【0163】

50

ブロック及び/またはブロックインスタンスに特異的な結果に加えて、DDE User Interface Applicationは、データブロックのうちの2つ以上（また、一部の場においては、大部分または全て）に対して適用可能であり得る標準ビジュアライゼーションを提供し得る。例えば、ブロックが実行されるとき、データセットの各列、タグ、または部分について平均、標準偏差、及び他のそのような統計が計算され、結果として生じるデータセットと共に記憶され得るようにブロックの実行の末期において、データの状態についての様々な統計の標準セットが収集され得る。特定のブロックインスタンスのビジュアライゼーションダイアログが呈示されるとき（例えば、それぞれのユーザ制御部212のアクティブ化を介して）、各列、タグ、または部分についての標準統計の計算セットがデータ記憶から回収され、ユーザに対して呈示される。次いで、ユーザは、目的の列/タグ/部分を選択し、Data Studioに、該列/タグ/部分の統計を表すそれぞれの図表または他の視覚的フォーマット（例えば、線図、散布図、ヒストグラム、データグリッド、データ要約グリッド、データの分散を示す計算された統計及びヒストグラム等）を生成するように要求することができる。ある実施形態においては、統計及びデータセットは独立的に記憶されるが、これは、結果として生じるデータセットから独立的に統計を記憶することが、有利なことに、DDE User Interface Applicationが、必要とされる量のデータのみをブラウザにロードすることを可能にするためである。

10

【0164】

標準的ビジュアライゼーション及びカスタムビジュアライゼーションを呈示するための例示的アーキテクチャ300が、図4Jに示される。例示的アーキテクチャ300においては、「Block1」302についてのデータブロックコードが実行され、結果として生じるデータセット305、計算された標準統計/他のビジュアライゼーション308、及びBlock1に特異的な結果310が生成され、DDE User Interface Applicationによって管理されるローカルまたはリモートの記憶領域312内に記憶される。Data Studioインスタンス315（例えば、ブラウザウィンドウ）においては、Block1グラフィックに表示される「ブロック結果閲覧」ユーザ制御部212をユーザが選択すると、Block1についての計算された統計308（例えば、標準セット及び/または任意のカスタムビジュアライゼーション）がData Studioインスタンス315にロードされ318、ユーザは、所望の列、タグ、または目的の部分を選択することができる。Block1の所望される列/タグ/部分をユーザが選択すると、ユーザによる閲覧及び探査のために、対応するデータがData Studioインスタンス315にロードされる320。

20

30

【0165】

経時的に、複数のデータモジュールが構成され、例えば、同じプロセス制御システム5内において、かつ/あるいは複数のプロセス制御システムを有するエンタープライズにまたがって、ブロック構成及び使用法の共通のパターンが起こり得る。そのような共通性が特定された場合、個別のデータブロックのセットをグループ化して、それらの共通の挙動をカプセル化する新しいブロック、例えば、複合ブロックを形成することが望ましくあり得る。一部のシナリオにおいては、例えば、特定のプロパティ値のセットが経時的に繰り返されていることが見出されるとき、またはユーザがカスタムデータオペレーションもしくは機能を定義することを所望するとき、カスタムの一元的なデータブロックを定義することが望ましくあり得る。カスタム及び/または複合データブロックは、それらが他のデータモジュールにおける使用について利用可能であるように、ライブラリ252内で作成及び記憶され得る。複合データブロックの作成を例証する例示的シナリオ330が、図4Kに示される。データダイアグラム332においては、ユーザは、特定の関係性（例えば、「Fill NaN」及び「Scale」）335を有する2つの一元的なデータブロックを選択して、新しい複合ブロックを形成する。Data Studioによって提供されるモーダルダイアログボックス、ポップアップウィンドウ、または他の好適なインターフェイスを用いて、ユーザは、所望の名称「Fill & Scale」をもつ新しい

40

50

複合ブロックを構成または定義し、この新しい「Fill & Scale」複合ブロックをデータブロック定義ライブラリ252に保存する(参照338)。「Fill & Scale」が定義及び保存されてしばらく経った後、個別の「Fill NaN」及び「Scale」データブロックを用いる代わりに、この新しい複合ブロックを別のデータダイアグラム339に組み込むことができる。

【0166】

図4Lは、上記の特色及び原理の一部を例証する例示的オフラインデータダイアグラム340を描写しており、これは、図4A、4B、及び4F~4Hを同時に参照しながら本明細書において説明される。図4Lにおいては、例示的オフラインデータダイアグラム340は、Data Studioのユーザキャンパス245において、ユーザによって作成された。具体的には、ユーザは、例えばキャンパス245の制御部248mをアクティブ化することでブロック定義ライブラリ252を呼び出し、LoadDBブロックプレート285aをキャンパス245上にドラッグアンドドロップした。更に、ユーザは、LoadDBブロック285aのこの特定のインスタンスを「LoadDB4M」(参照342a)と命名し、例えば、LoadDB4Mブロック342aのフィールド285c及び285bにおけるプロパティ値を設定することによって、Database MからDataset 4をロードするようにLoadDB4Mブロックインスタンス342aを構成した。加えて、図4Lにおいて明確に描写されていないが、ユーザは、ブロックLoadDB4M342aを評価する際、ロードされたDataset 4のプロットが生成され、LoadDB4Mブロック342aのView Block Resultsフィールド212aに呈示されるように、LoadDB4M342aのPlotプロパティ値285dを「True」に設定した。

10

20

【0167】

ユーザがLoadDB4Mブロックインスタンス342aを構成した後、ユーザは、LoadDB4Mブロックインスタンス342aの出力に、2つの受信側Columnsブロックインスタンス342b、342cを接続した。例えば、ユーザは、Columnsデータブロックプレート286aの2つの異なるインスタンスをキャンパス245上にドラッグアンドドロップし、これらのインスタンスをそれぞれ、「Columns X」(参照342b)及び「Columns Y」(参照342c)と命名した。更に、ユーザは、Columns Xブロック342b及びColumns Yブロック342cのそれぞれの入力の各々を、相互接続ワイヤを用いることで、LoadDB4Mブロック342aの出力に接続した。

30

【0168】

ユーザはまた、Columns Xブロック342b及びColumns Yブロック342cを、LoadDB4Mブロック342aの評価に基づくように構成した。具体的には、ユーザはまず、LoadDB4Mブロック342aを評価し(例えば、キャンパス245上のブロック342aの画像を選択し、「評価」ユーザ制御部248iをアクティブ化することによって)、それにより、LoadDB4Mブロック342aを、Database MからDataset 4をロードまたは獲得するようにコンパイル及び実行させた。LoadDB4Mブロック342aのPlotプロパティは「True」に設定されているため、LoadDB4Mブロック342aのこの評価はまた、ロードされたDataset 4のPlotを、例えば、LoadDB4Mブロック342aのView Block Resultsフィールド202a(図4Lには図示せず)に表示させる。このプロットを用いて、ユーザは、Dataset 4内のデータの様々な列、タグ、またはサブグループ分けを閲覧及び探査し、続いてColumns Xブロック342bを、Dataset 4内で「X」と標識またはタグ付けされたデータのサブグループまたは列を受信するように構成し、Columns Yブロック342cを、Dataset 4内で「Y」と標識またはタグ付けされたデータのサブグループまたは列を受信するように構成した(例えば、Columnsブロック342b、342cの各々について、それぞれ、Columnsブロックプレート286aのプロパティフィールド286bにおいて

40

50

、適切なサブグループ、列、またはタグインジケータを選択することによって)。したがって、Columns Xブロック342bの評価は、「X」と特徴付け、標識、またはタグ付けされるDataset 4からのデータのみがブロック342bにおいてロードまたはアクセスされることをもたらし、Columns Yブロック342cの評価は、「Y」と特徴付け、標識、またはタグ付けされるDataset 4からのデータのみがブロック342cにおいてロードまたはアクセスされることをもたらす。

【0169】

ユーザはまた、オフラインデータダイアグラム340において、Partial Least Squares (PLS)ブロックをドラッグ、ドロップ、及び構成した。具体的には、ユーザは、PLSブロックインスタンス342dを「PLS4M_X_Y」と命名し、このPLS4M_X_Yブロック342dを、Columns Xブロック342bの出力と、Columns Yブロック342cの出力とを、そのそれぞれの入力を受信するように構成した。PLS4M_X_Yブロック342cは、Columns Xブロック342b及びColumns Yブロック342cによって提供されるデータに対して、部分最小二乗機能、アクション、アルゴリズム、またはオペレーションを動作させるか、あるいは実施し、その結果(例えば、Columns XとColumns Yデータとの間の関係性に基づいて生成される、それぞれのPLSモデル)は、PLS4M_X_Yブロック342dの出力を介して、Explore4M_X_Yブロック342eへと提供される。他のブロック342a~342dと同様に、Explore4M_X_Yブロック342eは、ユーザがキャンパス245上にドラッグアンドドロップし、命名し、構成した、それぞれのExploreブロック定義のインスタンスである。

【0170】

オフラインダイアグラム340においては、Explore4M_X_Yブロック342eは、PLS4M_X_Yブロック342dによって生成された出力を受信するように構成されているが、342dに加えて、ユーザは、Explore4M_X_Yブロック342eも、例えばワイヤ342fを介して、Columns Yブロック342cの直接出力を入力として受信するように構成した。この構成によって、ユーザは、Explore4M_X_Yブロック342eを通じて、例えば1つ以上のビジュアライゼーション及び/または統計を呈示することで、PLS4M_X_Yブロック342cの出力とColumns Yブロック342cの出力との間の関係性を探索することが可能になる。例えば、典型的には、PLS4M_X_Yブロック342cの出力は、1つ以上の予測値を含む。Explore4M_X_Yブロック342eは、ユーザに、PLS4M_X_Yブロック342cの出力に含まれる1つ以上の予測値を、Columns Yブロック342cの実測値と比較させて、例えば、PLS4M_X_Yモデル342dにおけるColumns Xデータ342bの使用が、Columns Yデータ342cを十分に予測するかどうかを判定させる。

【0171】

このオフラインデータダイアグラムにおいては、PLS4M_X_Yブロック342d及びExplore4M_X_Yブロック342eのどちらも、それ自体の構成を完成させ得る前に、その直前のブロック(複数可)が評価されることを必要とするわけではないことが留意される。すなわち、PLS4M_X_Yブロック342d及びExplore4M_X_Yブロック342eの各々の構成は、他のブロックの評価からは独立的に、かつ任意の時間において実施することができる。しかしながら、オフラインデータダイアグラム340は1つ以上のデータパイプラインまたはデータフローを表すため、典型的には、各下流データブロックの評価は、その下流データブロックを評価できるようになる前に、その上流データブロック(複数可)を評価することを必要とする(無論、受信側または下流データブロックが、単に個別のブロック評価プロセスに関して、試験またはダミーの入力データセットを受信する場合を除く)。事実、一部のシナリオにおいては、ユーザは、全体のオフラインデータダイアグラム340を総括して構成し、かつ/あるいは全体のオフラインデータダイアグラム340を、ブロック毎または部分毎に評価するのではなく

10

20

30

40

50

、総括して評価することができる。

【0172】

2. Data Analytics Studio - オンラインデータモジュール

ここで図4Bに戻ると、データモデルのオフラインデータダイアグラムが、Data Studioにおいて、ユーザの思い通りに完成及び評価された後、このオフラインデータダイアグラムは、相当するそのオンライン形態へと翻訳または変換することができる。オフラインデータダイアグラムをそのオンライン形態へと転換するためには、ユーザは、ナビゲーションバー242のオンライン切り替え部またはユーザ制御部248fを選択またはアクティブ化し、それによって、DDE User Interface Applicationに、オフラインデータダイアグラムをそのオンライン相当物のデータダイアグラムへと転換させ、データモデルのオンラインデータダイアグラムをキャンパス245上に表示させればよい。ある特定のオフラインデータブロックは、オンライン相当物の定義を有する場合があります（例えば、「Load」データブロックのオフライン及びオンラインバージョン）、他のオフラインデータブロックは、異なるオンライン相当物を必要としない場合があるが、それでもなおオンラインデータモジュールに含まれ（例えば、「FillNaN」ブロックまたは「PCA_NIPALS」ブロック）、一方で、更に他のオフラインデータブロックは、オンラインデータモジュールにおいては省略される（例えば、「Explore」ブロック）。オフラインダイアグラムのそのオンライン相当物への転換中に実施される、インフラストラクチャ及びアクションは、Data Analytics Servicesによって提供される。例えば、ユーザ制御部248fのアクティブ化は、Data Analytics Servicesを呼び出して、オフラインデータダイアグラムの、その対応するオンラインデータダイアグラムへの転換を実施させる。Data Analytics Servicesがこの転換をいかに実施するかについてのより詳細な説明は、下のData Analytics Servicesの説明において提供される。

【0173】

データモジュールのオンラインデータダイアグラムがData Studioのキャンパス245上に呈示されるとき、ユーザは、オンラインデータモジュールを構成することができる。典型的には、オンラインデータモジュールの構成とは、例えば、オンラインデータモジュールを対応する入力データのデータソースに関連付けるバインディングを定義することによって、オンラインデータモジュールのために、入力データとして獲得される、プロセスの制御に関するライブプロセス制御データを示すことを含む。加えて、オンラインデータモジュールの構成には、オンラインデータモジュールによって生成される出力データが提供される場所（複数可）及び/またはコンシューマ（例えば、1つ以上のコンシューマDDE及び/またはアプリケーション、データ記憶、ファイル、ヒストリアン、プロセス制御デバイス、ルーチン、要素、構成要素等）を示すこと（例えば、バインディングを定義することで）が含まれ得る。オンラインデータモジュールのデータブロック、それらそれぞれのプロパティ、及び相互接続の構成及び/または修正は、ある実施形態においては、オフラインデータモジュールに関して上に考察した様式と類似する様式で実施される。

【0174】

ユーザがオンラインデータモジュールに伴う問題もしくは課題を特定した場合、または別様にオンラインデータモジュールの修正を所望する場合、ユーザは、例えば、オンライン切り替え部またはユーザ制御部248dを活用することで、オンラインデータモジュールをそのオフライン表現へとひっくり返すか、あるいは切り替えることができ、ユーザは更に、データモデルのオフラインデータダイアグラムを、上記のように修正及び評価することができる。後に、ユーザがオフラインデータダイアグラムをそのオンライン表現へとひっくり返すか、あるいは切り替えるとき、修正されたオフラインデータダイアグラムは、対応するデータモデルのオンラインデータダイアグラムへと転換する。

【0175】

10

20

30

40

50

図 4 M は、図 4 L の例示的オフラインデータダイアグラム 3 4 0 に対応するオンラインデータダイアグラム 3 4 5 を例証しており、図 4 M は、図 4 A、4 B、4 F ~ 4 H、及び図 4 L を同時に参照しながら下で考察される。図 4 M に描写される例示的シナリオにおいては、ユーザは、図 4 L のオフラインダイアグラム 3 4 0 はデータモデルを意図または所望の通りに定義しており、また、このデータモデルはオンラインプロセス制御システムへの展開にとって用意ができていと決定した。したがって、Studio がオフラインデータダイアグラム 3 4 0 (例えば、図 4 L に示されるもの) を閲覧している間に、ユーザは Data Studio のオンライン切り替え部 2 4 8 f をアクティブ化し、それによってオフラインデータダイアグラム 3 4 0 を対応するオンラインデータダイアグラム 3 4 5 へと転換し、このオンラインデータダイアグラムは、図 4 M に示されるように、Data Studio のキャンパス 2 4 5 上に呈示される。この転換は、DDE User Interface Application の Data Analytics Services によって実施され、これ及び他の転換を実施するために Data Analytics Services によって活用される特定のアクション、メカニズム、及びアーキテクチャは、後の節においてより詳細に説明される。

10

【0176】

オンラインデータダイアグラム 3 4 5 及びオフラインデータダイアグラム 3 4 0 は同一のデータモデルを表しているが、2 つのデータダイアグラム 3 4 5、3 4 0 のブロック及びワイヤのセットは異なっていることに留意されたい。例えば、Load DB 4 M ブロック 3 4 2 a の出力から始まり、Columns Y ブロック 3 4 2 e を横断し、Explore 4 M__X__Y ブロック 3 4 2 e の入力において終結する、オフラインデータダイアグラム 3 4 0 のデータパイプラインは、オンラインデータダイアグラム 3 4 5 からは省略されている。これは、このデータパイプラインが、オフラインデータダイアグラム 3 4 0 においてはオフライン PLS 4 M__X__Y ブロック 3 4 2 d を試験及び検証するために活用されており、今や PLS 4 M__X__Y ブロック 3 4 2 d の有効性はユーザの思い通りに実証されているので、オンライン環境においては必要でないためである。しかしながら、一部の実施形態においては、例えば、ユーザがオンライン環境においてオンラインデータモデルの継続的試験及び検証を行うことを所望する場合、この検証データパイプラインをオンラインデータダイアグラム 3 4 5 に含むことを選択してもよい。

20

【0177】

オフラインデータダイアグラム 3 4 0 とオンラインデータダイアグラム 3 4 5 との間の別の差異は Load DB 4 M ブロックである。オフラインデータダイアグラム 3 4 0 においては、Load DB 4 M ブロック 3 4 2 a は、Database M のオフラインデータソース Data set 4 からデータをロードするように構成される。一方で、オンラインデータダイアグラム 3 4 5 においては、Load DB 4 M ブロック 3 4 7 a は、プロセスプラント 5 内のオンラインストリーミングデータソース等のオンラインデータソースに接続できるブロックへと転換されている。ユーザは、Load DB 4 M ブロック 3 4 7 a を、所望のオンラインデータソースにバインドするように構成することができ、構成された Load DB 4 M ブロック 3 4 7 a のコンパイルが、バインディングをインスタンス化する。

30

40

【0178】

オンラインデータダイアグラム 3 4 5 においては、Columns X ブロック 3 4 7 b は概して、そのオフライン形態 3 4 2 b と類似するか、あるいは同等である。しかしながら、先に考察したように、PLS 4 M__X__Y ブロック 3 4 7 c のオンライン形態は、PLS 4 M__X__Y ブロック 3 4 7 c がオンライン環境において動作するために必要な入力、すなわち、Columns X ブロック 3 4 7 b の出力しか受信しない。

【0179】

加えて、オンラインデータダイアグラム 3 4 5 は、オフライン Explore 4 M__X__Y ブロック 3 4 2 e に対する相当物を有しない。これは、オフライン環境における Explore 4 M__X__Y ブロック 3 4 2 e の目的が、ユーザに、ドラフトデータダイアグ

50

ラム340がユーザの目標をどれだけ良好に達成したかについての所見を提供することであったためである。オンラインデータダイアグラム345においては、PLS4M__X__Yブロック347cの出力は、Data Consumersデータブロックの1種である、「Write」データブロック347dに提供される。Writeデータブロック347dは、オンライン環境におけるPLS4M__X__Yブロック347cのリアルタイム出力を、Writeブロック347dの構成において特定されるように、ファイルまたはデータベースに対して書き込ませる。無論、ユーザがそのように所望する場合、PLS4M__X__Yブロック347cの出力は、PLS4M__X__Yブロック347cの出力を他のData Consumerブロックの入力と相互接続することによって、データモデルの1つ以上の他のデータコンシューマに対して、追加的または代替的に提供することができる。例えば、PLS4M__X__Yブロック347cは、PLS4M__X__Yブロック347cの出力が、データ分析ネットワーク112に対して公開され、他のデータ分析アプリケーションによる使用にとって利用可能であるように、Publishブロックに接続することができる。別の例においては、PLS4M__X__Yブロック347cは、PLS4M__X__Yブロック347cの出力がオンラインプロセス制御システムにおいて変化を引き起こすように、Convert2Controlブロックに接続することができる。例えば、Convert2Controlブロックは、PLS4M__X__Yブロック347cの出力を、プロセスプラント5内のプロセス制御機能、ループ、またはアプリケーションに対する入力信号へと変換し、この入力信号を、対応するプロセス制御工業、要素、デバイス、または構成要素へと送達させるように構成することができる。

10

20

【0180】

3. データモジュールの展開

典型的には、オンラインデータモジュールは、オフラインデータモジュールの場合に可能であったように、非同期的には評価されない。むしろ、オンラインデータモジュールは、まず構成され（例えば、ライブデータソース及び/または1つ以上のデータコンシューマ/受信側にバインドされ）、次いで、連続的に実行され、オンラインプロセス制御システムと相互作用するように全体として展開される。ユーザは、例えば、ナビゲーションバー242の「展開」ユーザ制御部248jをクリックまたはアクティブ化して、それにより、Data Analytics Servicesにオンラインデータモジュールの展開を処理するように通知することで、データモジュールをオンラインプロセス制御システムに展開する。データモジュールを展開するために使用されるインフラストラクチャ及びアクションは、Data Analytics Servicesの説明においてより詳細に説明される。しかし、概して、オンラインデータモジュールが展開される時、オンラインデータモジュールは、例えば図4Iに例証されるような、評価されているオフラインデータモジュールの場合と同様の、コンパイル及び実行プロセスに従う。しかしながら、オンラインデータモジュールのターゲット環境（複数可）は、その対応するオフラインデータモジュールのターゲット環境（複数可）とは異なる場合があり、オンラインデータモジュールは、プロセス制御システムまたはプラント内のライブで他ソースにバインドされる。

30

【0181】

ある実施形態においては、DDE User Interface Applicationは、「ワンクリック」展開をサポートする。すなわち、オフラインデータモジュールがユーザの思い通りに構成されているとき、ユーザは、このデータモジュールをそのオンライン表現へと変換し、ライブデータソースからデータを読み取り、かつ書き込む役目を担う、オンライン表現のデータブロックに関するバインディングを構成する必要がない。代わりに、「ワンクリック」展開中は、ユーザが開始すると、サブジェクトデータモジュールにとって必要とされる任意のバインディングが、DDE User Interface Applicationによって自動的に決定及び構成される。例示的シナリオにおいては、ユーザは、例えば、DCSシステムの階層構造についての情報を回収する、Data Studioによって提供されるダイアログボックスまたはウィンドウを用い

40

50

ることで、特定のDCSコントローラに対して実行されるようにデータモジュールを割り当て得る。ユーザは、階層構造をブラウズし、データモジュールを適宜割り当て得る。続いて、DDE User Interface Application（例えば、Data Studio及び/またはData Analytics Services）が割り当てられたプロセス制御システム要素（このシナリオにおいては特定のDCSコントローラ）を活用して、必要なバイディングを自動的に決定する。ワンクリック展開を実施するために使用される例示的インフラストラクチャ及びアクションは、Data Analytics Servicesの説明においてより詳細に説明される。

【0182】

オンラインデータ分析モジュールが成功裏に展開された後、その実行によって生成される任意の結果として生じる値（例えば、記述的、予測的、及び/または規定的）を、例えば、DDE User Interface ApplicationのData Dashboardを介して、ユーザに対して示すことができる。ある実施形態においては、結果として生じる値は、少なくとも1つの予測値を含み、タグ（複数可）またはその予測値が適用されるプロセス制御要素についての、プロセス制御システム5によって生成される、対応するリアルタイムライブデータ値（複数可）もまた、予測値と併せて示され得る。

【0183】

C. Data Analytics Dashboard

先に考察したように、DDE User Interface Applicationは、オペレーションの第2のモード、Data Analytics Dashboardを含む。Dashboardは、典型的には、(i) 評価されているオフラインデータモジュール（例えば、ドラフトデータモデル）、及び/または(ii) オンラインプロセス制御システムまたはプラントに展開されているオンラインデータモジュールであるモジュールを閲覧及び/または管理するために、ユーザによって活用される。視覚的及び他の表現を用いて、Dashboardは、ユーザに、オフラインデータモデルの評価に対応するステータス及び他の情報の所見、ならびに展開されたオンラインデータモジュールの所見及びオンラインデータモジュールによって生成されているリアルタイムまたはライブ情報を提供する。概して、Data Analytics Dashboardモードは、ユーザに、ユーザがオンライン及びオフラインデータモジュールを監視及び管理できるインターフェイスを提供する。ユーザはまた、任意のオフラインデータモジュールの評価及び任意のオンラインデータモジュールの実行を、Dashboardから終結することもできる。概して、必ずしもそうではないが、Data Analytics Dashboardに表示される情報のうちの少なくとも一部は、DDE User Interface ApplicationのData Analytics Serviceによって、Dashboard 350に提供される。

【0184】

図4N-1は、DDE User Interface ApplicationのData Analytics Dashboardによって呈示される、例示的ユーザインターフェイス350を描写する。例示的Dashboardユーザインターフェイス350の第1の部分352は、DDE User Interface Applicationによって現在監視されているオンラインデータモジュール355の数の指標、及びオンラインデータモジュール355がそれに基づいて動作している、ライブデータストリーム358の数の指標等、現在のデータ分析の概要情報を提供する。先に考察したように、ライブデータストリーム358は、1つ以上の工業プロセスを制御するオンラインプロセス制御プラントまたは環境5から結果として生じるリアルタイムデータを連続的に生成している1つ以上のデータソースから受信され、実行中のオンラインデータモジュール355が、このライブデータストリーム358を受信するように構成されている。データ分析の概要情報はまた、その評価が現在DDE User Interface Applicationによって監視されている、オフラインデータモジュール360の数の指標

10

20

30

40

50

も含む。一部の実施形態においては、Dashboardユーザインターフェイス350は、オンラインまたはオフラインに関わらず、監視されているデータ分析モジュールに関連する、1つ以上の他の通知または他の情報362を含む。

【0185】

ユーザが概要制御部355、358、360、362のうちの1つをクリックするか、あるいは別様にアクティブ化すると、その選択に対応する追加的な要約情報が、Dashboard350の第2の部分365に呈示される。図4N-1においては、ユーザは、現在実行中のオンラインデータモジュール355についての要約情報を閲覧することを選択しており、これは、現在は2つ、すなわち参照368によって示されるDiagram2及び参照370によって示されるDiagram3が存在する。各実行中オンラインデータモジュール360、370について、Dashboard350の第2の部分365は、そのそれぞれの要約情報を示し、モジュールの名称372、モジュールが展開された開始時間375、最も直近の時間間隔にわたるモジュールの出力の傾向指標378（図4N-1に示されるように、これはグラフィカルであり得る）、最も直近の最新の予測もしくは予測値（複数可）380、及び/または他の情報（図示せず）を示す。特に留意すべきは、Dashboard350において、各オンラインデータモジュール368、370の傾向指標378は、それぞれのデータモジュール368、370の連続的に生成される出力を反映するように連続的にアップデートされ、それにより、ユーザがオンラインデータモジュールの実行及びその連続的に生成される結果をリアルタイムで監視することを可能にする。

10

20

【0186】

加えて、Data Analytics Dashboard350は、ユーザが、任意のデータモジュール360、378と関連付けられる任意の要約情報をクリックして、更なる詳細を、例えばポップアップウィンドウ、ダイアログボックス、または他の好適な表示メカニズム（図示せず）において閲覧することを可能にする。例えば、ユーザがDiagram2 368のNameをクリックする場合、Dashboard250は、Diagram2 368のオンラインデータモジュールダイアグラムと、Diagram2 368がバインドされている1つ以上の特定のデータストリーム358の指標とを呈示し得る。ユーザがDiagram2 368のStart情報をクリックする場合、Diagram2 368の作成者の指標、Diagram2 368を展開したユーザ、及び/またはDiagram2 368の所有者が呈示され得る。Diagram2 368のTrend378情報をクリックすると、ユーザは、例えば、閲覧する時間間隔の延長または短縮、傾向情報の、別の種類のビジュアライゼーション（例えば、棒図表、パイ図表、散布図等）への変換、所与の間隔にわたる傾向データに関する統計（例えば、平均、ローリング平均、最大値、最小値等）の適用及び/または獲得等によって、Diagram2 368の出力データ傾向を操作及び探査することが可能になる。同様に、Diagram2 368のLast Prediction380情報をクリックすると、ユーザは、例えば、最新の予測の時間及び予測がなされる頻度の閲覧、最新の予測と現在のライブデータ値との比較等によって、最新の予測値（複数可）を操作及び探査することが可能になる。

30

40

【0187】

ユーザが、ユーザ制御部360を選択して現在実行中のオフラインデータモジュール（図4N-1の例示的シナリオにおいては、これは存在しない）を閲覧すると、オフラインデータモジュールに対応する追加的な要約情報が、Dashboard350の第2の部分365に呈示される。概して、オフラインデータモジュールについての要約情報は、オンラインデータモジュールのものと同様であるが、関連付けられるライブデータストリームを示す代わりに、オフラインデータモジュールについての要約情報は、オフラインデータがそれに基づいて実行中である、試験データファイル（複数可）、またはオフラインデータソース（複数可）（例えば、試験ツール、デバイス、及び/または環境）によって生成されるデータストリームを示し得る。加えて、オフラインデータモジュールについての

50

要約情報は、その実行の現在のステータス、状態、及び/または進行を示し得る。

【0188】

一部のシナリオにおいては、ユーザは、Dashboardから(Data Studioからではなく)オフラインデータモジュールを構成及び/または展開することができる。例えば、ユーザは、Dashboardからオフラインデータモジュールの評価を監視している場合があり、そのオフラインデータモジュールが、その評価を成功裏に完了する。ユーザは、Dashboardのユーザ制御部(図示せず)をアクティブ化し得る。このユーザ制御部を介して、ユーザは、データモジュールを、ライブデータソース(複数可)と、オンラインデータモジュールがそれに基づいて実行される、ライブデータソース(複数可)によって生成されるデータとを参照するように構成することができる。オンラインデータモジュールの、データソース(複数可)にバインドするというこの構成は、Data Studioからではなく、Dashboardから開始されるという点を除いて、上に説明された構成と類似する。あるいは、ユーザは、Dashboardから、オフラインデータモジュール用のワンクリック展開を活用してもよい。

10

【0189】

ユーザが、ユーザ制御部358を選択して実行中のオンラインデータモジュールに現在バインドされているオンラインデータストリームと関連付けられる情報(図4N-1の例示的シナリオにおいては、これは3つ存在する)を閲覧すると、オンラインデータストリームに対応する追加的な要約情報が、Dashboard350の第2の部分365に呈示される(図示せず)。各ライブデータストリームについての要約情報は、データストリームの識別子、データストリームがそこから受信されるデータソースの指標、対応するプロセス制御システムのタグ、もしくはライブデータソースの他の従来型のプロセス制御システム識別子、データストリームのサブスクリプション(複数可)及び/もしくは公開についての情報、ライブデータストリームに基づいて現在実行中である1つ以上のオンラインデータモジュールの指標、ライブデータストリームの連続的にアップデートされるビジュアライゼーション(例えば、線グラフ、棒図表、散布図等、及び/もしくはその基本統計)、ならびに/または他の情報を含み得る。

20

【0190】

図4N-1に描写される例示的シナリオにおいて特に留意すべきは、データストリーム概要358が、現在オンラインデータモジュールがそれに基づいて動作しているライブデータストリームが3つ存在することを示している一方で、オンラインデータモジュール概要355は、実行中のオンラインモジュールは2つしか存在しないことを示していることである。したがって、実行中の2つのオンラインモジュール368、370のうちの一つは、ライブデータストリームのうちの2つに基づいて動作している。ユーザは、この関係性を、それぞれのモジュール368、370がバインドされているデータストリームの指標を提供する、各オンラインデータモジュール368、370の詳細を閲覧するようにナビゲートすることによって、あるいは各データストリーム358に基づいて動作している実行中のオンラインデータモジュールの指標を提供する、各データストリーム358の詳細を閲覧するようにナビゲートすることによって発見し得る。

30

【0191】

図4N-2は、DDE User Interface ApplicationのData Analytics Dashboardによって呈示され得る例示的ユーザインターフェイス382を描写する。ある実施形態においては、ユーザインターフェイス382は、図4N-1に示されるユーザインターフェイス350の第2の部分365に含まれる。ある実施形態においては、ユーザインターフェイス382は、フルスクリーンビューである。

40

【0192】

ユーザインターフェイス382は、1つ以上のタイルまたはカード385a~385jを含み、これらの各々は、それぞれの実行中のオンラインデータモジュールに対応し、これらは集合的に、カード385xのパレットまたはデッキと呼ばれる場合がある。パレッ

50

ト中のカード385xの各々は、図4N-1の実行中オンラインデータモジュール360、370に関して表示されていた情報と同様の情報、例えば、モジュールの名称、モジュールが展開される開始時間、最も直近の時間間隔にわたるモジュールの出力の傾向指標、最も直近の最新の予測もしくは予測値（複数可）、及び/または他の情報を含み得る。これもまた実行中オンラインデータモジュール360、370に関する表示と同様に、カード385xの各々の傾向指標は、そのそれぞれのオンラインデータモジュールの連続的に生成される出力を反映するように連続的にアップデートされ、それにより、ユーザがその実行及び連続的に生成される結果をリアルタイムで監視することを可能にする。

【0193】

カード385xの各々は、ユーザの所望に応じて、リサイズ可能かつ移動可能である。ある実施形態においては、特定のカード385xのサイズは、その特定のカードがフォーカスされるときに増加する。カード385xは、ユーザがその特定のカード385xをクリックするとき、その特定のカード385x上でホバリングするとき等、手動でフォーカスされ得る。追加的または代替的に、カード385xは、特定のデータモジュールが緊急の注意を必要とするとき（例えば、予測値が閾値を超えるとき、データモジュール自体がユーザ入力またはユーザ介入、カード385xの中での状態の重要度ランキング等を必要とするとき）、自動的にフォーカスされ得る。自動的に生成される、注意を必要とする他の指標は、ユーザインターフェイス382において、例えば、特定のカード385xに対する、何らかの他の処理のハイライトまたは提供、カード385xのセットの、それらそれぞれの緊急度に従ったサイズ調整、特定のカード385xの、パレットの前景または前
20
面への移動、カード385xの、注意の優先度の順位による配列によって示され得る。ある実施形態においては、ユーザは、特定の状況においてカード385xに対して自動的に適用される特定の処理を定義し得る。

【0194】

ここで、DDE User Interface Applicationのデータブロック定義ライブラリ252に記憶されているデータブロック定義255に戻ると（例えば、図4Cに示されるような）、ライブラリ252に記憶されているデータブロック定義は、そのスタートアップまたは初期化中に、DDE User Interface Applicationによって動的に発見され得る。ある実施形態においては、データブロック定義は、JSONファイルもしくはドキュメント、または他の好適なフォーマット
30
等の軽量データ交換フォーマットを用いて定義される。特定のデータブロックの定義ファイルまたはドキュメントは、ブロック定義の名称を含み、その特定のデータブロックを定義するブロックプロパティ及びコネクタについて詳述する。一部の場
合においては、特定のブロック定義は、オフライン表現についての定義、及びオンライン表現についての定義を提供し得る。

【0195】

DDE User Interface Applicationは継続的かつ動的にブロック定義を発見し得るため、追加的なブロック定義（例えば、カスタムデータブロック、複合データブロック、及び/または他のデータブロックに関する）が、DDE User Interface Applicationが実行中の任意の時間において、ライ
40
ブラリ252に追加され得る。ある実施形態においては、追加的なブロック定義は、本明細書においては「データブロック定義ラッパ」と呼ばれるユーティリティアプリケーションによって生成される。例えば、データブロック定義ラッパは、図4Kに関して先に説明したように、ユーザがカスタムまたは複合データブロックを保存するときに呼び出され得る。ブロック定義ラッパが実行されるとき、それは、追加的なブロック定義のインスタンスを作成し、追加的なブロック定義を定義する、対応するJSONドキュメントまたは類似物を作成及び記憶させる。例えば、追加的なブロック定義の作成されたインスタンスは、JSONドキュメントを生成し、それをライブラリ252に記憶する。図4Kを例示的シナリオとして参照すると、ユーザが複合ブロック定義338を保存すると、ブロック
50
定義ラッパが実行されて、複合ブロック定義338についてのJSONドキュメントまた

は類似物を生成及び記憶させる。

【0196】

データブロック（オフラインであるかオンラインであるかに関わらず）の構成中、DDE User Interface Applicationは、ユーザがデータブロックを成功裏に構成及び使用するのを補助するために必要な構成エクスペリエンスを達成するために、何らかの機能性（例えば、ビジネスロジック）を実施する必要がある。ブロック定義はDDE User Interface Applicationによって動的に発見されるため、この機能性は、DDE User Interface Application自体には埋め込まれていない。したがって、ある実施形態においては、特定のデータブロックに対して特異的である、任意の関連付けられた機能性もまた、DDE User Interface Applicationが初期化されるときに動的にロードされ得る。続いて、特定のデータブロックに対応する、動的にロードされた機能性に対する参照は、特定のデータブロックのブロック定義に維持され、その特定のブロックのブロック定義が構成されるとき、DDE User Interface Applicationによって使用される。ブロック定義と関連付けられる機能性に関する追加的な詳細は、本出願の後の節において提供される。

10

【0197】

プロセス制御システムと統合された工業プロセス監視及び分析システム

図1～3及び4A～4N-2の考察において見られるように、分散型工業プロセスパフォーマンス監視及び/または分析システム、すなわちDAS100は、工業プロセスプラントに関する、データ分析構成、データモデリング、データモデル実行、データビジュアライゼーション、及びストリーミングサービスをサポートするプラットフォームを提供する。先に考察したように、監視及び分析システム100の様々なノードがプロセス制御システムまたはプラント5内に埋め込まれるものの、典型的には、分析プラットフォーム100のデータ通信ネットワーク112は、プロセスプラント5の既存のDCS、PLS、衛生監視、及び他のシステムの外側に（また、それらから独立して）存在する。ある例示的構成においては、データ分析ネットワーク112は、プロセスプラント5の上にオーバーレイされ、それによって、プロセスプラント5の従来型のプロセス制御通信ネットワーク10、70等の大部分または全てからネットワーク112は分離される。従来型のプロセス制御通信ネットワークは、プラント5のプロセス及びオペレーションの制御部を制御、管理、及び/または監督するために、プロセス制御データの送信を継続するが、一方で、データ分析ネットワーク112は、プロセスプラント5のオペレーションに関する知識、例えば記述的知識、予測的知識、及び/または規定的知識の発見のために、分析データ（これは、従来型プロセス制御通信ネットワークを通じて送信されるデータのコピーを含み得る）を送信する。データ分析システム100のアーキテクチャは、データ分析システム100を、既存のレガシであるプロセスプラント5に容易に追加または統合させる。

20

30

【0198】

データ分析システム100によって発見される、プロセスプラント5についての知識は、プラント5のパフォーマンスを上昇させるため、ならびに故障、破損、及び他の望ましくない状態の発生を減少させるために、プロセスプラント5のパラメータ、設備、オペレーション、及び他の部分を修正するように活用され得る。有利なことに、データ分析システム100は、発見される知識がプロセスプラント5の現在のオペレーションを記述、予測、及び/または規定するように、プロセスプラントの現在のオンラインオペレーションに対応してリアルタイムでプロセスプラント5についての知識を発見する。したがって、データ分析システム100及びプロセスプラント5は、制御ループを形成して、プロセスプラント5のパフォーマンスを制御及び最適化する。

40

【0199】

プロセスプラント5のパフォーマンスを制御及び最適化する制御ループの例示的高次ブロックダイアグラム400が、図40に例証される。図40に示されるように、オンラインプロセスプラント5は、様々な種類の制御データ402を生成し、それに基づいて動作

50

し、従来型のプロセス制御通信ネットワーク（例えばネットワーク10、70等）を通じて制御データを通信して、1つ以上の工業プロセスを制御し、プロセスプラント出力405を生成する。分散型工業プロセスパフォーマンス監視/分析システム100は、プロセスプラント5が出力405を生成するように1つ以上のプロセスを制御した結果として生成される分析データ408を取り込む埋込式分散型データエンジン102xを介して、プロセスプラント5に接続され、それと統合される。先に考察したように、取り込まれる分析データ408は、一次及びより高次のプロセス制御データ402、ならびにDDE102xによってローカルで実施される分析機能の結果/出力、ならびに/またはデータ分析システム100のData Studioを介して開発されたオフライン及びオンラインデータ分析モジュールによって実施される分析機能の結果/出力を含み得る。一部の場

10

【0200】

データ分析システムまたはプラットフォーム100は、分析データ408を受信し、生成し、通信し、それに基づいて動作して、分析出力410を生成する。分析出力410は、プロセスプラント5の現在のオペレーションを記述する知識、プロセスプラント5の現在のオペレーションを前提とした場合の、故障、破損、時間間隔、パフォーマンス、事象等の発生を予測する知識、ならびに/またはプロセスプラント5の現在のオペレーションを前提とした場合の、現在のプラントオペレーションの望ましくない特性を軽減するた

20

【0201】

分析出力410に含まれる発見された知識の少なくとも一部は、Data Analytics User Interface Application等の、Human-Machine Interface (HMI) 412に提供され得る。HMI 412を介して、ユーザは、データ分析システム100によってプロセスプラント5に関して生成された、発見された知識410を探索し得る。一部の場

40

【0202】

一部の状況においては、分析出力410に含まれる発見された知識の少なくとも一部は、プロセスプラント5に対して直接提供されて、図40において矢印420で表されるように、1つ以上の規

50

の優先度に対する変更、または何らかの他の値、パラメータ、構成等に対する変更を含む、規定的アクションを含み得る。データ分析システム100は、いかなるユーザ介入418も必要とすることなく、この変更を自動的かつ直接ダウンロードするか、あるいは別様にこの変更をプロセスプラント5において行うことができ、それにより、発見された知識410に基づいて、プロセスプラント5のパフォーマンス及び出力405を自動的に最適化し、かつ/または故障、破損、及び他の望ましくない状態の発生を予防または阻止することができる。分析システム100が現在の分析データ408を処理し、プロセスプラント5の現在のオペレーションに基づいて、発見された知識410を生成する際、制御ループ400のこの脚は、オンラインプロセスプラント5を自動的にアップデートして、その現在のオペレーションをリアルタイムで改善することが留意される。したがって、規定的アクションを計算、決定、及び実施するのに数時間または数日間すら必要とする既存の分析技法とは対照的に、分析システム100によって生成される規定的アクションは、プラント5の現在の運転状態について最適化されており、プロセスプラント5に対して直ぐに適用され、一部の場においては、任意のユーザが切迫した、望ましくないまたは所望される状態に気付く前に適用される。

10

【0203】

無論、プロセスプラント5内における規定的アクションの全体としての最適な実施のためには、制御ループ400は、自律的脚部420及びHMI制御脚部418の両方を必要とする。例えば、置換バルブ、センサ、または他の構成要素の設置等、一部の規定的アクションは、人間の介入を必要とする。

20

【0204】

先に考察したように、データ分析プラットフォーム100は、分析データが、複数の埋込式分散型データエンジン(DD E)102xの間でストリーミングされることを可能にする。プロセスの制御の結果として、プロセスプラント5によって生成される任意の種類的一次データがストリーミングされ得る。例えば、プロセスを制御する目的のために、従来型のプロセス制御通信ネットワークを介して典型的には送信される、プロセス制御デバイス(例えば、コントローラ、I/Oカード、フィールドデバイス等)によって生成される信号に含まれるプロセス制御データもまた、データ分析の目的のために、データ分析ネットワーク112においてストリーミングされ得る。したがって、プロセス制御信号、センサ信号等に含まれる内容またはデータは、コピー及び/または複製され、データ分析ネットワーク112を通じてストリーミングされ得る。更に、従来型のプロセス制御通信ネットワークを介して生成及び送信される、二次(またはより高次の)プロセス制御データ(例えば、プロセス制御データベースまたはヒストリアン内にヒストライズされるデータ、プロセス制御構成、プロセス制御ユーザインターフェイスコマンド、プロセス制御ネットワーク管理信号、及びプロセスの制御の結果として、従来型のプロセス制御通信ネットワークを介して従来的には送信及び受信される任意の他の信号)が、データ分析の目的のために、データ分析ネットワーク112においてストリーミングされ得る。加えて、データ分析ネットワーク112は、計算されたデータ、または従来型のプロセス制御データ及び/もしくは計算される分析データに対して実施された分析機能もしくは技法の結果をストリーミングし得る。

30

40

【0205】

結果的に、データ分析プラットフォーム100は、プロセスプラント5のデータソースにローカルでバインドすることによって、ならびに/またはプロセスプラント5内のデータソース付近でデータ分析サービスまたは機能性を提供することによって、部分的にプロセスプラント5と統合するが、一方で同時に、より大規模の予測及び最適化を提供する。外側ループ予測及び最適化をサポートするために必要であるデータのみが、分析システム100の、より高次の/より有能な分析サーバ及びクラウドプラットフォームに通信される必要がある。

【0206】

先に考察したように、データ分析システム100の複数のDD E102xのうちの一部

50

は、プロセスプラント5内の物理的デバイスに埋め込まれる。例えば、複数のDDE102xのうちの一部は、各々、それぞれのプロセス制御デバイス（例えば、コントローラ11、I/Oカード26、28、フィールドデバイス15~16、18~21、42A、42B、44等）ならびに/またはそれぞれのネットワークルーティング及び管理デバイス（例えば、ゲートウェイ35、75、78、アダプタ52a、ルータ58、アクセスポイント55a、72等）と一体である。ある実施形態においては、プロセスプラント5の一元的デバイスまたは構成要素に統合されるのではなく、DDEは、スコープ、監視装置、またはボヤ（voyeur）として従来型プロセス通信リンクまたはネットワーク（例えば、ネットワーク10、70、PLSネットワーク、安全衛生監視システム等）に接続される自立型デバイスであってもよく、データ分析を目的としてデータ分析ネットワーク112を通じてストリーミングするために、従来型プロセス通信リンクまたはネットワークを介して送信されている、選択されたプロセス制御データをコピー/複製し得る。

【0207】

一部のシナリオにおいては、2つ以上のDDEが、データソースが登録及び接続するローカルデータ分析クラスタを形成し得、このローカルデータ分析クラスタを介して、登録されたデータソースで生成された分析データが、データ分析ネットワーク112に、分析データとして受信され得る。ある例においては、DDEのセットは、ローカルクラスタを形成し、そのローカルクラスタの存在を告知する。ローカルクラスタは、データ分析ネットワーク112を介して相互接続される複数のDDEをそのローカルクラスタが含んでいたとしても、データ分析ネットワーク112の一元的ノードまたは単一のDDEインスタンス150として現れる。新しいデータソースがプロセスプラント5の構内においてアクティブ化または起動される際、データソースは、ローカルクラスタの告知をスキャンまたは探す。データソースはローカルクラスタの告知を検出し、ローカルクラスタに登録し、それによってデータ分析ネットワーク112に接続し得る。続いて、登録されたデータソースは、分析システム100から、（例えば、それらのローカルクラスタを介して）帯域幅を要求し得、それらのローカルクラスタに、データ分析ネットワーク112の別のノードに、あるいは分析システム100に直接、データを公開し得る。ローカルクラスタは分散型データエンジン150のインスタンスとして現れるため、ローカルクラスタは、ローカルで生成された分析データ（例えば、登録されたソース及び/または他のソースからの分析データ）を、それぞれのビッグデータレシーバ160を介して受信し、それぞれのローカルの永続ストレージ155に、ローカルで生成された分析データを記憶する。加えて、ローカルクラスタは、例えばそれぞれのビッグデータ要求サービサ165を介して、要求アプリケーション（例えば、別のデータ分析アプリケーション、データモジュール、ユーザインターフェイス、Data Studio、別のアプリケーション等）に、記憶された分析データの少なくとも一部を提供し得る。一部の場においては、ローカルクラスタは、例えばそれぞれのビッグデータ要求アナライザ170を介して、登録されたデータソースから受信した、記憶されているデータ、及び/またはデータ分析ネットワーク112内でストリーミングされる他のデータに対して、それぞれのローカルデータ分析機能を実施し得る。ローカルデータ分析機能の結果または出力は、それぞれのビッグデータストレージ155にローカルで記憶され、かつ/あるいは、例えばローカルクラスタのそれぞれの1つ以上のデータ分析ネットワークインターフェイス175を介して、ローカルクラスタの外部の1つ以上の他のDDE、例えば別のローカルDDEもしくはローカルクラスタ、集中型データクラスタ、またはサブスクライブしているアプリケーションへとストリーミングされ得る。

【0208】

一部の実施形態においては、ローカルクラスタは、それら自身を、より高次の分析データを受信及び記憶し、それぞれのより高次のデータ分析機能を実施する、より集中型のクラスタに登録する。例えば、ある特定の集中型クラスタ（例えば、リージョナルクラスタ）は、プラント5の特定の領域内の全てのローカルクラスタから、ストリーミングされるデータを受信し得、それに対してデータ分析を実施し得る。別の例においては、ある特定

10

20

30

40

50

の集中型クラスタは、プラント5内の特定の種類の構成の全ての制御ループから、ストリーミングされるデータを受信し得、それに対してデータ分析を実施し得る。ローカルクラスタと同様に、集中型クラスタは、データ分析ネットワーク112内のDDE150の単一のノードまたはインスタンスとしての見かけを有する。

【0209】

結果的に、ローカリゼーションの様々な層または次元におけるDDE及び/またはDDEのクラスタを提供するという、分散型工業プロセスパフォーマンス監視及び分析システム100の能力は、プロセスプラント5のデータソース付近における故障検出及び予測の実行を可能にし、それにより、オンラインデータソースによって生成されるリアルタイムデータに対応する、リアルタイム故障検出及び予測を提供する。そのような多層またはクラスタ化アーキテクチャはまた、システム100全体にわたる分析データのストレージを可能にし、実際、分析システム100はプロセスプラント5内に埋め込まれた部分を含むため、プロセスプラント5全体にわたる分析データのストレージを可能にする。更に、データ分析システム100は、任意のDDEにおいて記憶されているデータにアクセスする能力を提供し、分析構成、診断、ならびにパフォーマンス、故障、及び破損の監視に関する集中型システムを提供する。

【0210】

ある実施形態においては、1つ以上の集中型クラスタが、ユーザまたはシステムアドミニストレータが、複数のDDE102xからの分析データのバックアップまたはヒストライズに関する規則を確立するための集中ロケーションを提供する。ある例においては、1つ以上の集中型クラスタは、プロセスプラント5の構内に配置され、プロセスプラント5を全体として処理する。別の例においては、1つ以上の集中型クラスタは、クラウドに配置またはホストされ、プロセスプラント5を全体として処理する。一部の構成においては、プロセスプラント5のバックアップまたはヒストライズを提供する集中型クラスタ（複数可）は、他のプロセスプラントもサービスする。概して、先に考察したように、分析データは、そのそれぞれのビッグデータ記憶領域155の各DDEまたはクラスタにおいて永続的に記憶される。一部の例においては、DDEまたはクラスタにおいて受信された、またはそれによって生成された一部の分析データは、より迅速なアクセスのために、そのDDEまたはクラスタにおいてキャッシュされ得る（例えば、サブスクリバの数、公開または送信の頻度等の1つ以上の判定基準に基づいて）が、概して、全てのローカルの分析データが、DDEまたはクラスタにおける永続的ストレージ155に移動される。定期的に、あるいは予め定義された時間において、各DDEにおいて永続的に記憶されている分析データは、例えば、データ分析ネットワーク112を介してアクセス可能である集中型分析データヒストリアンにおける、長期データ分析ストレージに、バックアップ、保存、またはヒストライズされ得る。例えば、データ分析ヒストリアンは、プラント全体のデータ分析ヒストリアンであってもよく、ならびに/またはクラウドにホストされてもよい。しかし、典型的には、データ分析ヒストリアンは、どのプロセス制御データヒストリアンからも独立し、分離されている。1つ以上の集中型クラスタにアクセスするユーザインターフェイスを介して、ユーザまたはシステムアドミニストレータは、各特定のDDE記憶領域155がどのくらいの頻度でバックアップされるか、バックアップはいつ起こるか、分析データがどこにヒストライズされるか等を示す、規則のセットを定義することができる。

【0211】

ある実施形態においては、1つ以上の集中型クラスタが、ユーザまたはシステムアドミニストレータが、データ分析システム100のクラスタ、ノード、及び/またはDDE102x内における分析データの複製に関する規則を確立するための集中ロケーションを提供する。例えば、複製規則のセットは、どの特定の分析データが、データ分析ネットワーク112のどの特定のノード内において複製されるか、ならびにどのくらいの頻度で特定の分析データが特定のノード内においてアップデートされるかを定義することができる。例えば、複製規則のセットは、ノードB~Dがそれらそれぞれのデータ分析機能を実施す

10

20

30

40

50

る際に、複製された結果を利用可能であるように、どのくらいの頻度で、ノードB～Dにおいて、ノードAによって実施された分析機能Aの結果を（例えば、ストリーミングによって）複製するかを定義し得る。ヒストライズ及びバックアップの規則と同様に、複製規則のセットは、1つ以上の集中型クラスタにアクセスするユーザインターフェイスを介して、ユーザまたはシステムアドミニストレータによって定義され得る。例えば、システムアドミニストレータは、ユーザインターフェイスを活用して、複製規則のセットを管理することができ、複製規則のセットは、1つ以上の集中型クラスタにおいて記憶され得る。分析データの複製の管理を提供する1つ以上の集中型クラスタは、分析データのバックアップ/ヒストライズの管理を提供するクラスタのセットと同じであってもよく、あるいは異なるクラスタのセットであってもよい。

10

【0212】

ある実施形態においては、1つ以上の集中型クラスタは、ローカルクラスタ、他の集中型クラスタ、及び/または個々のDDE102xが活用できる、ロケーションサービス、ストリーミングサービス（例えば、図1のストリーミングサービス115）、データディクショナリ、及び/または他の機能性を提供して、それらの間での分析データの交換を確立する。ロケーションサービス、ストリーミングサービス、データディクショナリ、及び/または他の機能性を提供する1つ以上の集中型クラスタは、分析データ複製の管理及び/または分析データバックアップ/ヒストライズの管理を提供するクラスタのセットと同じであってもよく、あるいは異なるクラスタのセットであってもよい。システムアドミニストレータまたはユーザは、ユーザインターフェイスを活用して1つ以上の集中型クラスタにアクセスして、クラスタ間及びデータ分析ネットワーク112のノード間における分析データの交換を確立及び提供するために使用される、サービス、ディクショナリ、及び他の機能性を管理し得る。

20

【0213】

図4Pは、ローカライズしたデータ分析サービスを提供する例示的方法450を描写する。ある実施形態においては、分散型データエンジン102xまたはDDEのクラスタは、方法450の少なくとも一部分を実施する。概して、方法450の少なくとも一部分は、システム100等の分散型工業パフォーマンス監視及び/または分析システムの任意の部分によって実施され得る。実際、方法450は、本明細書に記載されるシステム、装置、デバイス、及び/または技法のうちいずれかの任意の部分を用いて行われ得る。しかし、考察の容易さのために、方法450は、一元的なDDEインスタンス150として現れる、分析システム100の1つ以上のDDE102xのクラスタに関して下で説明される。しかしながら、これは多数の実施形態のうちの一つであるに過ぎず、限定的であることを意図しない。

30

【0214】

例示的方法450は、クラスタによる、クラスタの存在の告知を含む（ブロック452）。クラスタは、オンラインであり、プロセスを制御するように動作しているプロセス制御プラント5のデータ分析ネットワーク112に含まれ、クラスタの告知は、データ分析ネットワーク112を通じて、1つ以上の他のネットワークを通じて、Bluetooth、NFC等の近距離無線プロトコルを介して、あるいは任意の他の好適な告知手段を介して、送信され得る。

40

【0215】

方法450は更に、クラスタの告知へのデータソースからの応答を、クラスタで受信することを含む。例えば、データソースからの応答は、クラスタに登録するというデータソースの要求であり得、方法450は、データソースをクラスタ及び/またはデータ分析ネットワーク112に登録することを含み得（ブロック455）、それにより、データソースはデータ分析ネットワーク112と接続される。データソースは、プロセスプラント5に含まれ、かつオンラインプロセスプラント5によるプロセス制御の結果としてデータを生成する、デバイスまたは構成要素であり得る。ある例においては、データソースは、制御されているプロセスから結果として生じる、連続時系列データを生成する。

50

【0216】

加えて、方法450は、データソースによって生成されたデータの、クラスタによる受信を含む(ブロック458)。例えば、データソースは、その生成したデータを、データ分析ネットワーク112を介してクラスタにストリーミングし得る。追加的または代替的に、クラスタは、データソースによって生成及び公開される特定のデータをサブスクライブしてもよい。一部の実施形態においては、クラスタは、例えば、本出願の後の節において説明されるクエリメカニズムを用いることによって、特定のデータについてデータソースに照会し得る。データソースが、典型的には1つ以上の従来型のプロセス制御通信ネットワーク(例えば、ネットワーク10、70)を介してプロセス制御データを通信するデバイスまたは構成要素である場合、データソースは、この1つ以上の従来型のプロセス制御通信ネットワークを通じて通常通りプロセス制御データを通信し続ける一方でまた、データ分析ネットワーク112を介して、通信されるプロセス制御データの内容のコピーまたは複製をストリーミングし得る。

10

【0217】

データソースによって生成されたデータを受信する際、クラスタは、例えば受信したデータが1つ以上のキャッシュ条件を満たす場合、受信したデータの少なくとも一部をキャッシュし得る。しかし、受信したデータのいずれかがキャッシュされるか否かに関わらず、受信したデータの全体が、クラスタの永続的ストレージ155にローカルで記憶される。

【0218】

方法450は、データソースによって生成されたデータの少なくとも一部を、データ分析ネットワーク112を介して、クラスタによってデータコンシューマへとストリーミングすることを含む(ブロック460)。データコンシューマは、例えば、別の分析ノードもしくはクラスタ、データ分析アプリケーション、ユーザインターフェイス及び/もしくはユーザインターフェイスアプリケーション、オンラインデータ分析モジュール、別のアプリケーション、ならびに/またはデータベースもしくはデータヒストリアンであり得る。データコンシューマは、データソース及び/またはデータソースによって生成されるデータの少なくとも一部を含む特定のデータストリームに対してバインドされ得る。データコンシューマによる受信を所望される特定のデータは、典型的には、時系列データであり、連続データであり得る。

20

30

【0219】

データコンシューマは、例えば本出願の後の節において説明されるクエリメカニズムを活用することによって、データソースによって生成される特定のデータについてクラスタを照会している場合があり、ストリーミング(ブロック460)は、データコンシューマによって生成されたこのクエリに対する応答であり得る。ある実施形態においては、データコンシューマは、データソースによって生成され、データソースまたはクラスタによって公開されている特定のデータをサブスクライブし得る。ともあれ、クラスタは、データソースによって生成された所望のデータを、そのキャッシュまたはそのローカルの永続的データストレージ155から受信し、そのデータを、データ分析ネットワーク112を介してデータコンシューマにストリーミングする(ブロック460)。

40

【0220】

一部の実施形態においては、方法450は、データソースによって生成されたデータの少なくとも一部分に対して、クラスタによって、1つ以上のデータ分析機能または技法を実施することを含む(図4Pには図示せず)。1つ以上の分析機能または技法は、記述的分析、予測的分析、及び/または規定的分析を含み得る。1つ以上の分析機能または技法は、例えば、オンラインデータ分析モジュール、別のデータ分析アプリケーション、またはユーザの要求毎に、クラスタによって実施され得る。別の例においては、1つ以上の分析機能または技法は、クラスタによって自動的に開始及び実施され得る。1つ以上の分析機能または技法の実行の結果は、要求しているパーティに返される場合があり、ならびに/またはクラスタにおいてローカルで、永続的データストレージ155に記憶される場合

50

がある。一部の実施形態においては、1つ以上の分析機能または技法は、クラスタに登録される複数のデータソースによって生成され、クラスタに記憶（及び任意選択でキャッシュ）されているデータに対して実施される。

【0221】

クラスタは、プロセスプラント5内のデータソースによって生成される一次データを受信し、プロセスプラント5内のこれらのデータソース付近に配置されるため、クラスタは、本明細書においては「ローカル」クラスタと呼ばれる。一部の実施形態においては、方法450は、リージョナルクラスタ、広域クラスタ、プラントワイドクラスタ等の集中型クラスタによって生成される告知をローカルクラスタが検出することを含む。ローカルクラスタは、それ自身を集中型クラスタに登録することができ、そのキャッシュ及び/またはその永続的データストレージ155に記憶されているローカル分析データの少なくとも一部を集中型クラスタにストリーミングし得る。例えば、集中型クラスタにストリーミングされる分析データは、データソースによって生成されるデータ及び/またはローカルクラスタによって実施される1つ以上の分析機能の結果を含み得る。ローカルクラスタは、集中型クラスタによって開始されるクエリ毎に、集中型クラスタに対してデータをストリーミングしてもよく、ならびに/またはローカルクラスタは、集中型クラスタによってサブスクライブされているデータを公開してもよい。

【0222】

一部の実施形態においては、方法450は、クラスタのローカルの永続的ストレージ155に記憶されているデータをヒストライズまたはバックアップすることを含む（図4Pには図示せず）。データのバックアップはクラスタによって開始されてもよく、あるいはデータのバックアップは集中型分析データヒストリアンによって開始されてもよい。ある実施形態においては、クラスタ及び集中型分析データヒストリアンは、パブリッシャ/サブスクライバの関係性を有し得る。クラスタからデータをバックアップするタイミング及び/またはその量は、ある実施形態においては、システム100のシステムアドミニストレータによって定義され得る。バックアップされるデータは、長期ストレージ及びバックアップのために、クラスタから分析データヒストリアンに送信及び/またはストリーミングされ得る。

【0223】

図4Qは、ローカライズしたデータ分析サービスを提供する例示的方法470を描写する。ある実施形態においては、分散型データエンジン102xまたはDDE102xのクラスタは、方法470の少なくとも一部分を実施する。概して、方法470の少なくとも一部分は、システム100等の分散型工業パフォーマンス監視及び/または分析システムの任意の部分によって実施され得る。実際、方法470は、本明細書に記載されるシステム、装置、デバイス、及び/または技法のうちのいずれかの任意の部分を用いて行われ得る。しかし、考察の容易さのために、方法470は、一元的なDDEインスタンス150として現れる、システム100の1つ以上のDDE102xのクラスタに関して下で説明される。しかしながら、これは多数の実施形態のうちの1つであるに過ぎず、限定的であることを意図しない。

【0224】

例示的方法470は、集中型クラスタによる、集中型クラスタの存在の告知を含む（ブロック472）。データソースは典型的には集中型クラスタに直接は登録しないため、集中型クラスタは、図4Pのローカルクラスタとは異なる。むしろ、多くの場合、ローカルクラスタまたは他の種類のクラスタが集中型クラスタに登録し、そのような他のクラスタから受信するデータに対して、集中型クラスタがデータ分析機能を実施する。しかし、ローカルクラスタと同様に、集中型クラスタは、プロセスを制御するように運転する際にオンラインであるプロセス制御プラント5のデータ分析ネットワーク112に含まれ、集中型クラスタの告知は、データ分析ネットワーク112を通じて、送信、ストリーミング、及び/またはブロードキャストされ得る。本明細書で使用する場合、「集中型」クラスタは、他のクラスタが登録しているクラスタを指す。集中型クラスタは、制御ループ（例え

10

20

30

40

50

ば、「制御ループ」クラスタ)、プロセスプラントのある領域(例えば、「広域」クラスタ)、プロセスプラントのあるリージョン(例えば、「リージョナル」クラスタ)、全体のプロセスプラント(例えば、「プラントワイド」クラスタ)、複数のプロセスプラント(「エンタープライズ」クラスタ)等処理し得る。典型的には、集中型クラスタは、単一の、一元的 D D E インスタンス 1 5 0 としての見かけを有する、データ分析ネットワーク 1 1 2 を介して相互接続する複数の D D E を含む。

【 0 2 2 5 】

方法 4 7 0 は更に、集中型クラスタの告知への別のクラスタからの応答を、集中型クラスタで受信することを含む。例えば、他方のクラスタからの応答は、集中型クラスタに登録するという他方のクラスタの要求であり得、方法 4 7 0 は、他方のクラスタを集中型クラスタに登録することを含み得る(ブロック 4 7 5)。登録するクラスタは、図 4 P のローカルクラスタ等のローカルクラスタであり得、あるいは登録するクラスタは、別の集中型クラスタであり得る。

10

【 0 2 2 6 】

加えて、方法 4 7 0 は、登録されたクラスタからストリーミングされるデータを集中型クラスタにおいて受信することを含む(ブロック 4 7 8)。例えば、登録されたクラスタは、その記憶及び/またはキャッシュしているデータの一部分を、データ分析ネットワーク 1 1 2 を介して集中型クラスタにストリーミングし得る。したがって、ストリーミングされるデータは、典型的には、例えば連続データであり得る時系列データを含む。追加的または代替的に、集中型クラスタは、登録されたクラスタによって公開される特定のデータをサブスクライブしてもよい。一部の実施形態においては、集中型クラスタは、例えば、本出願の後の節において説明されるクエリメカニズムを用いることによって、特定のデータについて、登録されたクラスタに照会する。ストリーミングされるデータは、登録されたクラスタによって生成されたデータを含み得、ならびに/またはデータソース及び/もしくは他のクラスタから登録されたクラスタにストリーミングされたデータを含み得る。

20

【 0 2 2 7 】

一部の実施形態においては、集中型クラスタは、例えば受信したデータが 1 つ以上のキャッシュ条件を満たす場合、それが受信するストリーミングされたデータの少なくとも一部をキャッシュする。集中型クラスタにおいて受信されるデータのいずれかがキャッシュされるか否かに関わらず、受信したデータの全体が、集中型クラスタの永続的ストレージ 1 5 5 にローカルで記憶される。

30

【 0 2 2 8 】

ブロック 4 8 0 においては、方法 4 7 0 は、受信され、ストリーミングされたデータ(例えば、そのキャッシュ及び/またはその永続的記憶領域 1 5 5 に記憶されているデータ)の少なくとも一部分に対して、1 つ以上のデータ分析機能または技法を実施することを含む。1 つ以上の分析機能または技法は、記述的分析、予測的分析、及び/または規定的分析を含み得る。例えば、1 つ以上の分析機能または技法は、オンラインデータ分析モジュール、別のデータ分析アプリケーション、またはユーザの要求毎に、集中型クラスタによって実施され得る。別の例においては、1 つ以上の分析機能または技法は、集中型クラスタによって自動的かつ/または定期的実施され得る。1 つ以上の分析機能または技法の実行の結果または出力は、要求しているパーティに返される場合があり、ならびに/または集中型クラスタにおいてローカルで、永続的データストレージ 1 5 5 に記憶される場合がある。一部の実施形態においては、1 つ以上の分析機能または技法は、複数の登録されたクラスタによって生成され、集中型クラスタに記憶(及び任意選択でキャッシュ)されているデータに対して実施される。

40

【 0 2 2 9 】

方法 4 7 0 は、1 つ以上の分析機能または技法の結果または出力の少なくとも一部を、データ分析ネットワーク 1 1 2 を介して、集中型クラスタによって 1 つ以上のデータコンシューマへとストリーミングすることを含む(ブロック 4 8 2)。1 つ以上のデータコン

50

シューマは、例えば、別の分析ノードもしくはクラスタ、データ分析アプリケーション、ユーザインターフェイス及び/もしくはユーザインターフェイスアプリケーション、オンラインデータ分析モジュール、別のアプリケーション、ならびに/またはデータベースもしくはデータヒストリアンを含み得る。1つ以上のデータコンシューマは、集中型クラスタ、特定の分析結果/出力、及び/または特定の分析結果/出力を含む特定のデータストリームに対してバインドされ得る。

【0230】

1つ以上のデータコンシューマは、例えば本出願の後の節において説明されるクエリメカニズムを活用することによって、特定の分析結果/出力について集中型クラスタを照会している場合があり、ストリーミング(ブロック482)は、1つ以上のデータコンシューマによって生成されたこのクエリに対する応答であり得る。追加的または代替的に、データコンシューマは、集中型クラスタによって公開される特定の分析結果/出力をサブスクライブしてもよい。ともあれ、集中型クラスタは、所望のデータを、そのキャッシュまたはそのローカルの永続的データストレージ155から受信し、その所望のデータを、データ分析ネットワーク112を介して1つ以上のデータコンシューマにストリーミングする。一部の場合においては、所望の分析結果/出力データは、1つ以上のデータ分析機能または技法により、その生成時にストリーミングされる。

【0231】

一部の実施形態においては、方法470は、1つ以上の追加的なローカルクラスタを集中型クラスタに登録することと、この1つ以上の追加的なローカルクラスタからそれぞれの分析データを受信することとを含む(図4Qには図示せず)。そのような実施形態においては、1つ以上のデータ分析機能または技法を実施すること(ブロック480)は、集中型クラスタにおいて複数のローカルクラスタから受信した分析データに対してデータ分析機能または技法を実施することを含み得る。

【0232】

一部の実施形態においては、方法470は、集中型クラスタのローカルの永続的ストレージに記憶されているデータ(例えば、ビッグデータ記憶領域155に記憶されているデータ)をヒストライズまたはバックアップすることを含む(図4Qには図示せず)。データのバックアップは集中型クラスタによって開始されてもよく、あるいはデータのバックアップは集中型分析データヒストリアンによって開始されてもよい。ある例においては、集中型分析データヒストリアンにおける集中型クラスタは、パブリッシャ/サブスクライバの関係性を有し得る。集中型クラスタからデータをバックアップするタイミング及び/またはその量は、その集中型クラスタにおいて、または別の集中型クラスタにおいて、システム100のシステムアドミニストレータによって定義され得る。例えば、システムアドミニストレータは、ユーザインターフェイスを活用して、システム100内におけるバックアップに関する規則のセットを管理し得、この規則のセットは、集中型クラスタに記憶され得る。バックアップされるデータは、長期ストレージ及びバックアップのために、集中型クラスタ152及び分析データヒストリアンから送信及び/またはストリーミングされ得る。

【0233】

追加的または代替的に、一部の実施形態においては、方法470は、データ分析システム100のクラスタ、ノード、及び/またはDDE102x内における分析データの複製に関する規則のセットを提供することを含む。例えば、複製規則のセットは、どの特定の分析データが、データ分析ネットワーク112のどの特定のノード内において複製されるか、ならびにどのくらいの頻度で特定の分析データが特定のノード内においてアップデートされるかを定義することができる。例えば、複製規則のセットは、ノートB~Dがそれぞれそれぞれのデータ分析機能を実施する際に、複製された結果を利用可能であるように、どのくらいの頻度で、ノードB~Dにおいて、ノードAによって実施された分析機能Aの結果を(例えば、ストリーミングによって)複製するかを定義し得る。ヒストライズ及びバックアップの規則と同様に、複製規則のセットは、1つ以上の集中型クラスタにおいて

10

20

30

40

50

、システム100の管理されるシステムによって定義され得る。例えば、システムアドミニストレータは、ユーザインターフェイスを活用して、複製規則のセットを管理することができ、複製規則のセットは、それらの1つ以上の集中型クラスタにおいて記憶され得る。

【0234】

一部の実施形態においては、方法470は、別の集中型クラスタによって生成される告知を集中型クラスタが検出することを含む。例えば、告知するクラスタは、プラント5の構内に配置される、リージョナルクラスタ、広域クラスタ、またはプラントワイドクラスタであり得る。一部の場合においては、告知するクラスタはクラウドにホストされ得、プラント5の一部分、プラント5の全体、または複数のプラントに関して分析サービスを提供し得る。集中型クラスタは、それ自身を、告知するクラスタに登録することができ、そのキャッシュ及び/またはその永続的データストレージ155に記憶されている分析データを、告知するクラスタにストリーミングし得る。例えば、集中型クラスタは、集中型クラスタが実施した分析機能の結果の少なくとも一部をストリーミングし得る。集中型クラスタは、その集中型が登録されている告知するクラスタにデータをストリーミングしてもよく、ならびに/または集中型クラスタは、告知するクラスタによってサブスクライブされているデータを公開してもよい。

10

【0235】

一部の実施形態においては、方法470は、他のローカル及び集中型クラスタ、ならびに個々のDDE102xが使用できる、ロケーションサービス、ストリーミングサービス（例えば、図1のストリーミングサービス115）、及び/またはデータディクショナリを提供して、それらの間での分析データの交換を確立することを含む（図4Qには図示せず）。

20

【0236】

工業パフォーマンス監視/分析のサービス

Data Studioによる、オフラインデータダイアグラムのオンラインデータダイアグラムへの転換、オフライン及びオンラインブロック及びモジュールの両方の実行、ならびにダイアグラムによって表される分析の機能性は、データ分析システム(DAS)100等の工業パフォーマンス監視/分析システムによって提供される、Analytics Serviceによって実現される。Analytics Serviceは、本明細書においては交換可能に、「Service」、「Services」、または「Analytics Services」と呼ばれる。本明細書に記載されるAnalytics Servicesのいずれかの部分または全ての部分が、図1、2、3、及び/または4A~4Qのいずれかの部分または全ての部分と併せて、かつ/あるいは図1、2、3、及び/または4A~4Qにそれぞれ対応する本開示の節に記載される任意の数の特色及び/または技法と併せて動作し得る。

30

【0237】

概して、Analytics Serviceは、（例えば、Data Studio及び/またはDashboardにおける）オフラインダイアグラムのコンパイル、実行、及び/または転換が可能であり、（例えば、Data Studio及び/またはDashboardにおける）オンラインダイアグラムのコンパイル、実行、及び/または作成が可能である。Analytics Serviceは、特定の種類（複数可）のデータの知識を必ずしも必要とせず、ならびにAnalytics Serviceのユーザが、分析を実施する低水準プログラミングを行うことを必要とせずに、大きなセットのデータ、すなわちBig Dataの探査を容易にする。Analytics Service 500の例示的ブロックダイアグラムアーキテクチャが、システム100の一部分の文脈において、図5Aに示される。より具体的には、Analytics Service 500は、複数のコンピュータ実装プロセスのうちの1つであり、これらのコンピュータ実装プロセスの各々が、様々なソフトウェアエンティティ（例えば、1つ以上の有形の非一時的コンピュータ媒体に記憶される、コンピュータが実行可能な命令及びデータ）と

40

50

関連付けられる。下で詳細に説明されるように、Analytics Service 500は、システム100の様々なエンティティによって生成及び/または記憶されるデータを介して、多くの部分で、より大きなシステム100と相互作用する。

【0238】

上記のように、DDE User Interface Applicationは、実施形態においては、ウェブベースのアプリケーションである。図5Aは、そのような実施形態において、すなわち、ウェブベースのDDE User Interface Applicationを伴って実装される、Analytics Service 500の実施形態を描写する。その最も広い意味においては、この様式で実装されるAnalytics Service 500は、クライアント側の、サーバ側の、及びバックエンドのコンピュータ実装プロセスを含む。例えば、図5Aに描写されるように、Analytics Service 500は、ウェブクライアントプロセス502、ウェブサーバプロセス504、実行サービスプロセス506、及び1つ以上のジョブプロセス508を含む。様々なプロセス502、504、506、508は、1つ以上のプロセッサにおいて、1つ以上のワークステーションまたはサーバにおいて、1つ以上の物理的及び/または地理的ロケーションにおいて実行され得る。すなわち、ウェブクライアントプロセス502のあるインスタンスが、ウェブサーバプロセス504が実行されているサーバから離れたワークステーションにおいて実行され得る一方で、ウェブクライアントプロセス502の別のインスタンスが、ウェブサーバプロセス504が実行されているサーバと同じサーバ上で実行されてもよく、実際、同じプロセッサ上で動作してもよい。別の例としては、実行サービスプロセス506が、プロセス制御環境の領域内のワークステーションにおいて実行され得る一方で、1つ以上のジョブプロセス508は、プロセス制御環境から離れて配置される分散型プロセス環境（例えば、サーバファーム）の1つ以上のプロセッサにおいて実行され得る。簡単に言えば、Analytics Service 500は、様々なプロセス502、504、506、508が、同じプロセッサ、ワークステーション、プロセッサのバンク、1つ以上の仮想マシン、及び/または同じ物理的もしくは地理的ロケーションにおいて実行される必要性、または実行されない必要性が存在しないほど、フレキシブルであるように設計される。有利なことに、このアーキテクチャは、Analytics Service 500によって提供されるデータ分析へのリモートアクセス、ならびに巨大プロセッサ及び計算リソースを活用して分析を実施する能力を促進するのみならず、また、Analytics Service 500が、複数のプロセスプラントの地理的ロケーションがどれだけ異なり得ようと、それらにまたがってデータを消費及び/または分析することも可能にする。

【0239】

ウェブクライアントプロセス502は、一般に理解されている原理とそれほど異なりはしないが、上記のDDE User Interface Applicationの説明に則った原理に従って動作する。例えば、ウェブクライアントプロセス502は、例えば、DDE User Interface Application（例えば、Data Analytics Studio 240のキャンパス245及びユーザ制御部248a~n、Data Analytics Dashboard等）をユーザに呈示する閲覧エンティティ510；ブロック状態等のアプリケーションデータの、ユーザインターフェイスが活用し得る形態への移送及び翻訳を管理し、かつオンラインデータモジュールのオペレーションからのフィードバックを提供し、ならびに/または別様にそのオペレーションを示すビューを生成する、ビューモデルエンティティ512；ウェブクライアントプロセス502にダウンロードされ、かつユーザがDDE User Interface Applicationと相互作用するために使用するクライアントワークステーションまたはデバイスに存在する、ソフトウェアエンティティ（すなわち、ウェブベースのアプリケーション）である、アプリケーションエンティティ514；ならびにDDE User Interface Applicationに、及びそこからデータを受け渡す、データサービスエンティティ516を含む、様々なソフトウェアエンティティを含む

10

20

30

40

50

【0240】

データサービスエンティティ516は、例えば、ユーザ入力及び要求に対する応答として、実行された様々なジョブから返されるデータを受信する。上記のように、ならびに下の更なる詳細においては、DDE User Interface Applicationは、異種データ（もしくは既知の関係性のデータ）間の関係性を発見するために使用される（例えば、Data Analysis Studioにおける）探査モード、またはリアルタイム（もしくは準リアルタイム）の連続分析を実施して、プロセス制御環境において少なくとも1つの値を予測するために使用される（例えば、Dashboardにおける）予測モードのいずれかで、プロセス制御環境からのデータ（及び、一部の場
10
合においては、プロセス制御環境によって現在生成されているデータ）に基づく様々な分析の実行を要求し得る。本明細書で使用する場合、語句「リアルタイム（real time）」または「リアルタイムの（real-time）」は、「プロセスプラントの現在のオペレーションを調整するためまたは別様にそれに影響を及ぼすために有用であり続けるように、十分に同時期であること」を意味する。例えば、1つ以上のブロックがキャンパス245上に配置され、それらのブロックのうちの1つ以上が実行される場合、その実行の結果は、データサービスエンティティ516を介してアプリケーションエンティティ514へと戻され（例えば、ウェブサーバプロセス504からのウェブソケットプロトコ
20
ル接続を用いて）、表示エンティティ510を介してユーザに対して表示され得る。

【0241】

同時に、データサービスエンティティ516は、他のプロトコル（例えば、HTTP）を介してウェブサーバプロセス504と通信して、アプリケーションエンティティ514のオペレーションに必要な様々な種類の情報を要求及び受信し得る。例として、アプリケーションエンティティ514が最初に実行される場合、それは、HTTPプロトコルを用いるデータサービスエンティティ516を介して、ブロック定義ライブラリ252からの最新のブロック定義255を要求及び受信し得る。あるいは、アプリケーションエンティティ514は、当初は、利用可能なブロック定義255のカテゴリ及びリストのみを受信し得、ユーザによって選択されたとき、アプリケーションエンティティ514は、HTTP
30
プロトコルを用いるデータサービスエンティティ516を介して、ユーザによって選択された特定のブロック定義255を要求及び受信し得る。

【0242】

ここで、ウェブサーバプロセス504に注目すると、1つ以上のサーバ通信エンティティ518が、ウェブクライアントプロセス502と通信する。上記のように、サーバ通信エンティティ（複数可）間の通信は、HTTP、HTTPS等のプロトコル及びウェブソケットプロトコルを実装し得るが、本出願にとって好適である任意の他のプロトコルが実装され得るため、記載される特定のプロトコルは限定的であるものとみなされるべきではない。ウェブクライアントプロセス502との通信に加えて、サーバ通信エンティティ518は、キャンパス245に配置される1つ以上のブロック（すなわち、ダイアグラムのプログラミング要素）に従って要求される命令（例えば、データ分析、データローディング、データフィルタリング等）を実行する1つ以上のプロセスからデータを受信し得る。
40
データは、プロセスから、例えばサーバ通信エンティティ518のウェブソケット（図示せず）を介して受信され得、これもまたウェブソケットプロトコルを介して、ウェブクライアントプロセス502に通信され得る。

【0243】

データサービスエンティティ520は、ウェブクライアントプロセス502に対して、様々なブロックと関連付けられる命令を実行するプロセスから受信するデータ以外のデータを提供する。例えば、データサービスエンティティ520は、ブロック定義ライブラリ252からのブロック定義255を回収し、それをアプリケーションエンティティ514に通信し得、ならびに/またはブロックの実行についてのステータス情報を、そのステータスが表示エンティティ510によって表示され得るように、実行サービスプロセス50
50

6 からアプリケーションエンティティ 5 1 4 に通信し得る。

【 0 2 4 4 】

図 5 B に注目すると、データサービスエンティティ 5 2 0 がより詳細に描写される。具体的には、データサービスエンティティ 5 2 0 は、ブロック定義ライブラリ 2 5 2 のブロック定義 2 5 5 のコピー（またはそれに対するポインタ）と、データプラグインのセット 5 2 3 とを含む。データプラグイン 5 2 3 の各々は、アプリケーションエンティティ 5 1 4 から呼び出されて特定のタスクを実施し得るインターフェイスに対して適合する、コンピュータが実行可能な命令のセットである。プラグイン 5 2 3 の各々は、アプリケーションエンティティ 5 1 4 からは独立しており、アプリケーションエンティティ 5 1 4 が、プラグイン 5 2 3 をロードすることを特定するメタデータと遭遇したときに、アプリケーションエンティティ 5 1 4 によってロードされる。プラグイン 5 2 3 は、アプリケーション及び/または Analytics Service 5 0 0 が動作している動作環境に対して特異的なロジックを含み得る。すなわち、Analytics Service 5 0 0 を実装するプロセス制御環境用のプラグインのセット 5 2 3 は、Analytics Service 5 0 0 を実装する金融サービス用のプラグインのセットとは異なる場合があり（Analytics Service 5 0 0 は、特定の種類（複数可）のデータの知識を必ずしも必要とせず、ならびに Analytics Service 5 0 0 のユーザが、分析を実施する低水準プログラミングを行うことを必要とせずに、大きなセットのデータ、すなわち Big Data の探査を容易にすることに留意されたい）、また、実際、Analytics Service 5 0 0 を実装する異なるプロセス制御環境用のプラグインのセットとも異なる場合がある。

10

20

【 0 2 4 5 】

アプリケーションエンティティ 5 1 4 からのブロック定義 2 5 5 の分離は、アプリケーションエンティティ 5 1 4 が、顧客の要件に基づいて、異なるブロック定義を使用及び展開することを可能にする。アプリケーション及び/または動作環境に対して特異的なロジックは、アプリケーションエンティティ 5 1 4 にハードコードされていないため、ブロック定義は、アプリケーションエンティティ 5 1 4 の根本的なオペレーションに影響を及ぼすことなく、作成、追加、修正、及び/または除去することができる。アプリケーションエンティティ 5 1 4 からは別個にプラグイン 5 2 3 及びブロック定義 2 5 5 を提供することによって、アプリケーションエンティティ 5 1 4 は、特定の用途にとって必要とされる特定のブロック定義 2 5 5 及びプラグイン 5 2 3 に関わらず、類似の機能で動作し得る。

30

【 0 2 4 6 】

図 5 A を再度参照すると、上記のように、アプリケーションエンティティ 5 1 4 は、キャンバス 2 4 5 上における、図 4 L のオフラインダイアグラム 3 4 0 等のオフラインダイアグラムの作成を容易にするように動作する。オフラインダイアグラムを作成するために、ユーザは、1 つ以上のブロックを選択してキャンバス 2 4 5 上に配置し（例えば、ドラッグアンドドロップによって）、データがそれらのブロック間で通信されるように、それらのブロックを共に「ワイヤでつなげる」。ブロックの各々は、ブロック定義ライブラリ 2 5 2 のブロック定義 2 5 5 のうちの 1 つとして記憶されるテンプレートブロックのインスタンスである。ユーザがユーザ制御部 2 4 8 m をアクティブ化したとき、アプリケーションエンティティ 5 1 4 は、表示エンティティ 5 1 0 に、ブロック定義 2 5 5 の利用可能なカテゴリ及び/または利用可能なブロック定義 2 5 5 を表示させ、ここからユーザはブロックを選択して、キャンバス 2 4 5 上に配置することができる。ある実施形態においては、ブロック定義ライブラリ 2 5 2 は、実行サービスプロセス 5 0 6 もしくはウェブサーバプロセス 5 0 4 のうちの 1 つまたはそれら両方によって共有されるメモリデバイスに記憶されており、ブロック定義ライブラリ 2 5 2 のメタデータ（例えば、ブロック定義の名称、ブロック定義のカテゴリ等）は、アプリケーションエンティティ 5 1 4 に送信されるが、ブロックがキャンバス 2 4 5 上に配置されるように選択されるときにのみ、ブロック定義はアプリケーションエンティティ 5 1 4 に送信される。あるいは、ブロック定義ライブラリ 2 5 2 は、アプリケーションエンティティ 5 1 4 がウェブクライアントにおいてイ

40

50

インスタンス化され、ウェブクライアントプロセス 502 の実行中、終始ウェブクライアントに存在し続ける場合、その全体がアプリケーションエンティティ 514 に送信される。ブロック定義ライブラリ 252 のマスターコピーは、プロセス制御環境内のデータベース（図示せず）、ネットワークを介してアクセス可能なクラウドストレージデバイス、ワークステーション等において存在し得る。

【0247】

ブロック定義 255 は、各ブロックの機能性に従ってカテゴリ化され得る。図 5 A は、描写されるいくつかのカテゴリ 521 の各々の内で分類されたブロック定義 255 を描写するが、図 5 A は、任意の特定のカテゴリ 521 にカテゴリ化されたブロック定義 255 が必ず一緒に記憶されること、または必ず他のブロック定義 255 とは別個に記憶されることを提案することを意図してはいない。これは一部の実施形態においては当てはまる場合もあるが、他の実施形態においては、ブロック定義 255 はいかなる特定の順番でも記憶されておらず、代わりに、ブロック定義が属するカテゴリを示し得るメタデータで各々がタグ付けされ得る。任意の所与のブロック定義 255 に関するメタデータはまた、例としてであり限定ではないが、入力及び出力の数、ブロックの実行にとって必要とされるプロパティ、各プロパティに関する UI 制御タイプ、各プロパティに関する制御がいつ（すなわち、どのような条件の下に）有効化/無効化されるか、プロパティに関するプラグイン情報、プロパティのデフォルト値等を示し得る。

【0248】

全てではないが多くの実施形態においては、ブロックは、ブロックがコンパイル及び実行され得る前に特定され得る、または特定されねばならないプロパティを含む。ブロック定義 255 はアプリケーションエンティティ 514 に構築されていないため、ユーザがそこから特定のブロックに関するプロパティを選択し得る値は、アプリケーションエンティティ 514 がインスタンス化される場合、予め決定できず、かつ/あるいはアプリケーションエンティティ 514 に提供できない。図 5 C は、キャンバス 245 上に配置されたブロックに関するプロパティダイアログを呈示するための方法 551 を描写するフローチャートである。ブロックがキャンバス 245 上に配置されるとき、アプリケーションエンティティ 514 は対応するブロック定義 255 をデータサービスエンティティ 520 から、あるいは実施形態においてはデータベース 529 から回収する（ブロック 553）。その後、アプリケーションエンティティ 514 は、例えば、ユーザがブロックをダブルクリックしたとき、キャンバス 245 上に配置されたブロックに関するプロパティダイアログを表示するためのコマンドを受信する（ブロック 555）。

【0249】

アプリケーションエンティティ 514 がブロックに関するプロパティを示すためのコマンドを受信すると（ブロック 555）、アプリケーションエンティティ 514 は、ブロック定義をプロパティ毎に構文解析する。各プロパティについて、アプリケーションエンティティ 514 はそのプロパティに関するメタデータを回収し（ブロック 557）、プロパティに関するメタデータを構文解析して、プラグイン情報を探す（ブロック 559）。そのプロパティに関して、アプリケーションエンティティ 514 がプラグイン情報を見つける場合（ブロック 561）、アプリケーションエンティティ 514 は、特定されたプラグインを発見し、それをロードする（例えば、データサービスエンティティ 520 のプラグイン 523 から）（ブロック 563）。アプリケーションエンティティ 514 は、プロパティに関するメタデータによって特定されたプラグインを呼び出して、必要とされるプロパティ値を入手する（ブロック 565）。一方で、アプリケーションエンティティ 514 がプロパティに関するプラグイン情報を見つけなかった場合（ブロック 561）、アプリケーションエンティティ 514 は、プロパティに関するメタデータにおいて、プロパティのデフォルト値（複数可）を見つける（ブロック 567）。

【0250】

いずれにせよ、プロパティに関するメタデータによって特定されたプラグインを呼び出して、プロパティ値を入手するか（ブロック 565）、あるいはプロパティに関するデフ

10

20

30

40

50

ォルト値（複数可）を見つけることで（ブロック567）、アプリケーションエンティティ514は、プロパティに関するメタデータにおいて、UI制御タイプ（テキストボックス、ドロップダウン選択、ラジオボタン、チェックボックス等）を見つけ（ブロック569）、その値（複数可）及び制御をプロパティダイアログに追加する（571）。追加的なプロパティがブロック定義に存在する場合（ブロック573）、アプリケーションエンティティ514がこの次のプロパティに関するメタデータを回収し（ブロック557）、本方法は、追加的なプロパティが存在しなくなるまで継続する（ブロック573）。追加的なプロパティがブロック定義において見つからない場合、アプリケーションエンティティ514は、ユーザがプロパティを設定するためのプロパティダイアログを表示する（ブロック575）。

10

【0251】

ある実施形態においては、かつ/あるいは一部のブロック定義255に関しては、ブロックの1つ以上の第1のプロパティが、ブロックの1つ以上の第2のプロパティが構成されるまで、非アクティブまたは無効化状態において表示され得る。1つ以上の第2のプロパティを構成する際、アプリケーションエンティティ514は、方法551の1つ以上のステップを再実行し得、ならびに/または第1のプロパティと関連付けられるプラグイン523のうちの一つ以上を再実行し得る。

【0252】

例として、ユーザが、「ロード」ブロックをキャンバス245上に配置したと仮定する。アプリケーションエンティティ514は、ブロック定義255を回収する。ある実施形態においては、ロードブロックについてのブロック定義255は、アプリケーションエンティティ514にそのブロックに関するメタデータを提供し、具体的には、ロードブロックが3つのプロパティ：ブロックの名称と、選択されたデータベース（データベースの名称）と、選択されたデータベースから引き出す、選択されたデータセット（データセットの名称）とを有することを示す。プロパティ構成ダイアログがトリガされると（例えば、ロードブロックをダブルクリックすることによって）、アプリケーションエンティティ514は、何を表示すべきが決定する必要がある。アプリケーションエンティティ514は、ブロックの名称のプロパティに関するメタデータ内にプラグイン情報が存在しないことを決定し得、代わりに、デフォルト値、例えば「load<##>」を見つける。アプリケーションエンティティ514はまた、ブロックの名称のプロパティに関するメタデータ内で示される、ブロックの名称についての制御タイプが、テキストボックスを特定することを見つけ得る。アプリケーションエンティティ514はデフォルト値（例えば、「load01」）を解釈し、それをダイアログ内にテキストボックスとして配置する。

20

30

【0253】

ブロック定義を構文解析するアプリケーションエンティティ514は、データベースの名称のプロパティに関するメタデータを見つける。データベースの名称のプロパティに関するメタデータを確認することで、アプリケーションエンティティ514は、そのプロパティに関するプラグイン情報を見つけ、適宜、特定されたプラグインを発見、ロード、及び呼び出す。特定されたプラグインは、特定のアプリケーション及び/または動作環境について（例えば、特定のプロセス制御環境について）書かれているが、データセットがロードされ得るデータベースの名称（及び関連付けられるロケーション）のリストを発見するようにプログラムされている。データベースの名称のリストを受信すると、アプリケーションエンティティ514は、データベースの名称のプロパティに関するメタデータを構文解析し、ドロップダウン選択制御部で表示されるべきであることを決定する。アプリケーションエンティティ514は、プラグインによって返されたデータベースの名称のリストと共に、ドロップダウン選択制御部をプロパティダイアログに追加する。

40

【0254】

ブロック定義を構文解析するアプリケーションエンティティ514は、データセットの名称のプロパティに関するメタデータを見つける。データセットの名称のプロパティに関するメタデータは、プラグインを特定し、データセットの名称のプロパティがドロップダ

50

ウン選択制御部であることを示すが、また、この制御部が、データベースの名称がユーザによって選択されるまで無効化されなければならないことを示す。ロードブロックに関するブロック定義において更なるプロパティが見つからなくなると、アプリケーションエンティティ514は、ダイアログを表示する。ユーザが、データベース名称ドロップダウン選択制御部からデータベースの名称のうちの1つを選択すると、アプリケーションエンティティ514は、(データブロック定義から)データセット名称プロパティに関するメタデータにおいて特定されるプラグインを呼び出し、選択されたデータベースの名称を呼び出されたプラグインに引数として渡すようになっている。プラグインは、選択されたデータベースの名称の中で利用可能であるデータセットの名称を返し、データセットの名称のプロパティに関するドロップダウン選択制御部を、利用可能なデータセットの名称で埋め、今度は制御部を無効化ではなく有効化して、ダイアログを再度表示する。その後、ユーザはデータセットの名称を選択して、データブロックの構成を完了することができる。

10

【0255】

無論、あるブロックがキャンバス245上に配置されると、任意の必要とされるプロパティを特定するように構成された後、任意の下流ブロック(すなわち、このブロックの出力が「ワイヤでつながれ」た入力を有する、任意のブロック)が、このブロックから有効な入力を得て、有効な出力を生成するように、このブロックは実行され得る(すなわち、ブロック定義において特定されるプログラミング命令が実行され得る)。例えば、図4Lに示されるオフラインデータダイアグラム340に関して、LoadDB4Mブロック342aは、ブロック、Columns X342b及びColumns Y342cが構成され得る前に評価または実行されねばならない。これは、ブロック、Columns X342b及びColumns Y342cの構成は、LoadDB4Mブロック342aの出力からのユーザ選択に基づくためである。一方で、更に図4Lを参照すると、PLS4M_X_Yブロック342dが、ブロック、Columns X342b及びColumns Y342cの評価とは独立して構成できる一方で、ブロック、Columns X342b及びColumns Y342cは、PLS4M_X_Yブロック342dが、オフラインデータダイアグラム340にロードされる入力データセットに基づいて評価され得る前に、評価されなければならない。

20

【0256】

ブロック及び/またはモジュールの実行及び評価は非同期的である。これにより、ユーザは、ブロックまたはモジュールが評価されている間に、それらの修正を継続することが可能になる。例えば、ユーザは、1つのブロックの実行を開始する一方で、1つ以上の他のブロックの編集及び/または構成を継続し得る。有利なことに、別個にコンパイル可能かつ実行可能である各ブロックは、下流ブロックを選択及び/または構成及び/または実行する前に評価することができ、その結果を探查することができる。結果として、特定のブロックの予期される出力の演繹的知識は必要ではなく、また、任意の特定のブロックの結果を見る前に、データフローの次のブロックが何になるかを知る必要もない。

30

【0257】

オフラインまたはオンラインダイアグラムの任意の実行は、実行サービスプロセス506及びジョブプロセス508の機能性に関与する。一般的に、先に記載したように、ジョブプロセス508は、実行サービスプロセス506及び/もしくはウェブサーバプロセス506が実行されている1つ以上のプロセッサにおいて(同じプロセッサ(複数可)である場合またはそうでない場合がある)、ウェブクライアントプロセス502が実行されている1つ以上のプロセッサにおいて、ならびに/またはクラウドコンピューティングプラットフォームにおけるプロセッサ等の1つ以上の別個のプロセッサにおいて実行され得、1つ以上のプロセッサに基づいて構成される1つ以上の仮想マシンにおいて実行され得る。また、各ジョブプロセス508は、例えば、ターゲット環境用に構成された専用の仮想マシンにおいて各ジョブプロセス508を実行することによって、異なるターゲット環境において実行され得る。ジョブプロセス508の各々は概して、実行されるコードの一部を表す。一部の事例においては、特定のジョブプロセス508が、特定のブロックと関

40

50

連付けられるターゲットスクリプト/コードを実行し、一方で他の事例においては、単一のブロックと関連付けられる命令が、複数のジョブプロセス508における実行のために、より小さいスクリプト/コードのセグメントへと分割される。

【0258】

具体的には、単一のブロックと関連付けられる命令が、異なるジョブプロセスによる実行のために小さなスクリプト/コードのセグメントへと分割されるとき、これらの異なるジョブプロセスは、協働して何らかの形態の並列化を実現し得る。本明細書で使用する場合、「並列コンピューティング」は、あるタスクをより小さいユニットへと分割し、それらを並列で実施する一般的実践を意味するように使用され、「マルチスレッド処理」は、複数のスレッドを実行するソフトウェアプログラムの能力（リソースが利用可能である場合）を意味し、「分散型コンピューティング」は、複数の物理的または仮想マシンにまたがって処理を広げる能力を意味する。原則的に、分散型コンピューティングは、制限無しにスケールアウトすることができる。これらの概念の見地から、分散型プラットフォームにおいて命令を実行することは、分散型モードで命令を実行することと同じではないことを理解されたい。分散型処理をサポートする明確な命令が無い場合、それらの命令はローカルで実行されることになる。

10

【0259】

タスクを並列化する能力は、タスク自体の定義に内在するプロパティである。一部のタスクは、各ジョブプロセスによって実施される計算が全ての他のジョブプロセスから独立しており、所望の結果のセットが各ジョブプロセスからの結果の単純な組み合わせであるため、並列化が容易である。これらのタスクを「当惑するほど並列である」と称する。

20

【0260】

一部のタスクは、並列化により多くの労力を要する。具体的には、そのような「直線的に並列である」タスクとは、1つのジョブプロセスによって実施される計算は別のジョブプロセスによって実施される計算から独立しているが、所望の結果のセットは各ジョブプロセスからの結果の線形の組み合わせであるというタスクのことである。例えば、「平均計算」ブロックは、単一のデータのセットが複数の物理的メモリデバイスにまたがって記憶されている、分散型データベースにおける値のセットの平均を計算する命令によって定義され得る。より具体的な例としては、平均計算ブロックが、一連の300個のバイオリアクタにまたがる平均温度を計算すると仮定する。これらのバイオリアクタの各々は、関連付けられるコントローラのメモリに温度データを記憶する。平均計算ブロックの実行は、300個のバイオリアクタについての温度値のセットを複数のジョブプロセスの各々に割り当て、次いで、複数のジョブプロセスの各々によって決定された平均の加重平均として、「総」平均を計算することによって並列化され得る。

30

【0261】

データが意味のある方法で整理されなければならないタスクは、並列化がより困難である。「データ並列」タスクとは、各ジョブプロセスによって実施される計算が、各ジョブプロセスがデータの「正しい」チャックを有する限り、全ての他のジョブプロセスから独立しているタスクである。これには、例えば、上記と同じ300個のバイオリアクタ内において交差効果が存在しないと仮定すると、これらのバイオリアクタの各々について独立した温度平均を計算すること（それら300個全てにまたがる平均ではなく）が含まれ得る。このタスクは、ジョブプロセスの各々が、これらのバイオリアクタのうちの単一のものに関する温度データを有する場合に並列化できる。

40

【0262】

しかしながら、そのようなタスクがデータ並列であるためには、データは、タスクと整合するチャンクとして整理されていなければならない。これは、上記のシナリオの300個のバイオリアクタの各々が、その温度データを、単一の対応するデータベースまたはロケーションに記憶している場合に成り立ち得る。しかしながら、300個のバイオリアクタ全てに関するデータが単一のデータベースに記憶されており、データベース内でいずれの方法でも整理されていなかった場合、このデータは、何らかの種類の分析が起こり得る

50

前に、再編成を必要とする。本明細書に記載されるシステムにおいては、データの再編成は、分析を実施するように構築された、ダイアグラム内の他のブロック（例えば、データフィルタリングブロック、クエリブロック等）によって取り扱われ得る。

【0263】

ジョブプロセス508の作成及び管理、ならびに様々なプロセッサ内におけるジョブプロセスの並列化及び分散は、実行サービスプロセス506の担当である。モジュールまたはブロックが実行される場合、そのモジュールまたはブロックがオンラインダイアグラムの一部として、またはオフラインダイアグラムの一部として実行されるかに関わらず、そのモジュールまたはブロックの構成は、ウェブサーバプロセス504から実行サービスプロセス506に送信される。実施形態においては、この構成は、JSON (JavaScript Object Notation) ファイルとして送信されるが、この構成ファイルは、言語に依存しない、非同期のブラウザ/サーバ通信にとって好適である、任意のデータフォーマットを活用し得る。構成ファイルは、ダイアグラム602によって表される分析（またはその一部）が実行される環境の固有のソースコード、ならびに、その実行にとって必要とされる環境プロパティ及び変数の値を含む。

10

【0264】

ブロック定義は、任意の特定の環境における、そのブロックの実行を必要とせず、ターゲット環境は、そのブロック（またはそのブロックを含むモジュール）の構成の際に選択されるため、Application、ブロック定義、または記憶されている、もしくは実行中のブロック及び/もしくはモジュールを書き換える必要無しに、新しいターゲット環境が、Analytics Serviceにおいて利用可能であるものに追加され得る。

20

【0265】

同様に、新しいブロック定義255は、予め構築されたモジュールのいずれにも影響を及ぼすことなく、あるいはAnalytics Service500全体または具体的にはアプリケーションエンティティ514のオペレーションのいずれも変更することなく、任意の時間に、ブロック定義ライブラリ252に追加することができる。図5Bを再度参照すると、新しいブロック定義の作成及び追加は、実施形態においては、アルゴリズム定義ラッパ525によって実現される。アルゴリズム定義ラッパ525においては、ユーザまたは開発者は、定義ジェネレータ527を使用して、アルゴリズムを開発することによって定義を作成し得る。定義ジェネレータ527は、定義255'を生成し、その定義255'を、例えばデータベース529内のブロック定義ライブラリ252に記憶する。本説明の後の節においてより明確になるように、各ブロック定義255は、対応するデータプラグイン523を有し得、そのデータプラグイン523は、Analytics Service500が把握しているロケーション（例えば、データベース531）に記憶され得る。データサービスエンティティ520は、アプリケーションエンティティ514が初期化/インスタンス化されるときに、ブロック定義255及びデータプラグイン523を回収し得、それらをアプリケーションエンティティ514に提供し得る。

30

【0266】

あるいは、新しいブロック定義を作成するよりもむしろ、一部の実施形態においては、ユーザによって定義されたブロック定義は、関連付けられるアルゴリズムを有さない場合もあるが、代わりに、ユーザがユーザ独自のコードを配置することができる、プロパティを含み得る。すなわち、ブロック定義255のうちの一つは、ユーザが、ユーザ独自のアルゴリズムをプロパティとしてブロックに入力することを可能にし得る。

40

【0267】

本明細書で使用する場合、用語「パラメータ (parameter)」及び「パラメータ (parameters)」は、それぞれ、構成ファイルに含まれ、かつ構成ファイル経由で実行サービスにパラメータとして渡される、ブロック（複数可）及び/またはモジュール（複数可）のプロパティ (property) またはプロパティ (properties) を指す。パラメータとして構成ファイルで渡されるプロパティとしては、例とし

50

て、ファイルパス、データセットの名称、選択される列、PCA分析の結果において所望される構成要素の信頼水準及び数、ならびにコンパイラエンティティ526が実行可能なコードをコンパイルするために必要とされ、ジョブプロセス508がコンパイルされた命令を実行するために必要とされる任意の他の情報が挙げられ得る。すなわち、ブロックまたはモジュールの全てのプロパティが実行のために必要とされるわけではない。例えば、ブロックまたはモジュールの一部のプロパティは、そのブロックまたはモジュールがData Analytics Studioにおいてどのように表示されるかを特定するか、あるいは別様にそれに関連し得、そのデータがどのように収集、操作、または処理/分析されるかについては影響を及ぼさない。

【0268】

ある実施形態においては、構成ファイルは、ブロック及び/またはモジュールの定義、ならびにまた、ブロック/モジュールをコンパイル及び実行するために必要とされる任意のプロパティも含む。ある代替的实施形態においては、構成ファイルは、ブロックの識別コード及び必要とされる構成パラメータのみを含み、ブロック定義は、メモリから（例えば、ブロック定義ライブラリ252から）回収される。とにかく、構成パラメータは、ブロック定義に従って変化し得る。一部のブロックは構成パラメータを有さない場合があり、一方で他のブロックは、1つ、2つ、または多数の構成パラメータを有し得る。加えて、構成パラメータは必須である場合があり、あるいは任意選択である場合がある。例えば、データロードブロック（データのセットをロードするブロック）またはクエリブロック（データセット内で特定のデータをサーチするブロック）は、データがロードまたは照会される場所を特定するデータパスを必要とし得る。しかしながら、列選択ブロックは、デフォルト選択として「全ての列」を有する場合があり、したがって、列のサブセットの選択の、具体的な主体的構成を必要とはしない場合がある。

【0269】

構成ファイルにおける他のパラメータとしては、環境パラメータが挙げられ得る。構成ファイルは、例としてであり限定ではないが、例えばワークフローもしくはその一部がオフラインで実行されるか、またはオンラインで実行されるかを特定する実行モード；ターゲット実行環境（例えば、Python、PySpark、Native Spark等）；実行がローカルであるか、または分散されるか；分散型環境の構成；ワークフローまたはワークフローの一部がバインドされるストリーミングデータソースの指標；新しいジョブを作成するか、または持続的ジョブにバインドし、その実行を継続するかのオプション；持続的ジョブにまたがるデータをキャッシュするオプション；実行中のモデルの精度を評価する目的で、オンラインダイアグラムの実行に検証変数を含むオプション；あるいは他の環境変数を含み得る。

【0270】

ブロック、モジュール、及び/または環境パラメータの構成は、様々な方法のうちのいずれかで起こり得る。上記のように、例えば、多くのパラメータが、それぞれのブロックまたはモジュールにおいて設定されるプロパティに従って、構成ファイルに渡される。ダイアグラムを構築する際、（常にではないが）多くの場合、ユーザは、下流ブロックを構成及び/または実行し得る前に、特定のブロックを構成及び実行することが必要である。これは、ユーザが、データをロードするために第1のブロックを使用して、第2の下流ブロックをデータから列を選択するために使用する場合に当てはまる。ロードブロックは、ロードブロックを実行し得る前に構成されなければならない（例えば、データをロードするパス、ロードするデータの選択等を伴って）、データ内で利用可能な列が、ユーザによる列の選択が可能であるように第2のブロックに把握される前に、実行されなければならない。

【0271】

代替的または追加的に、1つ以上のパラメータが、ブロックまたはモジュールの実行について要求される際、構成ファイルに追加され得る。これは、例えば、ブロックまたはモジュールのプロパティのうちの1つについて、ユーザが特定しない場合に成り立ち得る。

10

20

30

40

50

これはまた、1つ以上のプロパティが、実行に関して要求される際にのみ特定されるようにプログラムされている場合にも当てはまり得る。実施形態においては、例えば、Data Analytics Studioは、ブロックまたはモジュールがいつ実行のために選択されるか、実行のターゲット環境、及び/またはブロックもしくはモジュールを実行するターゲットプロセッサもしくはプロセッサのバンクをユーザに特定させる。更に、一部の実施形態においては、構成ファイル中の1つ以上のパラメータが、Data Analytics Studio環境自体から渡され得る。好ましい実行環境、処理が起り得る/起こらなければならない物理的ロケーションに対する制限（例えば、クラウド実行の防止または要求）、及びその他等のパラメータが、アプリケーションから、アプリケーション内で実行される全てのブロック/モジュールについて、アプリケーション自体において特定され得る。これらのパラメータは、任意の特定のブロックまたはモジュールの実行が要求される際、構成ファイルに渡され得る。

10

【0272】

いずれにせよ、図5Aをなおも参照すると、ジョブリスナエンティティ522は、ブロック（複数可）またはモジュール（複数可）を実行するという、構成ファイルによる実行要求を受信し、実行が要求されているブロック（複数可）及び/またはモジュール（複数可）に対応する、1つ以上のジョブマネージャエンティティ524を作成する。ジョブマネージャエンティティ524は、構成ファイルを受信し、特定されるターゲット実行環境に従って、ダイアグラムコンパイラエンティティ526のうちの1つ以上に構成ファイルを送信して、特定されるターゲット環境内のプロセッサによって実行可能な命令へとコンパイルさせる。ダイアグラムコンパイラエンティティ526は、ダイアグラムコンパイラエンティティ526の各々が、いくつかの特定されるターゲット環境のうちのいずれかへと構成ファイルをコンパイルするように動作可能であるように設計されてもよく、あるいはダイアグラムコンパイラエンティティ526の各々が、ただ1つの具体的なターゲット環境を特定する構成ファイルをコンパイルするように動作可能であるように設計されてもよい。いずれにしても、ジョブコンパイラエンティティ526の出力は、特定されるターゲット環境内のプロセッサによって実行可能である、コンピュータ可読命令である。

20

【0273】

ダイアグラムコンパイラエンティティ526に構成ファイルを送信するジョブマネージャエンティティ524は、コンパイルされた出力を、ダイアグラムコンパイラエンティティ526からジョブ実行部エンティティ528に送信する。ジョブ実行部エンティティ528は、ジョブのターゲット環境等に関する特定された構成パラメータに従って、コンパイルされた出力を実行するのに、どのリソース（複数可）（例えば、プロセッサリソース）が利用可能であるか、ジョブが分散されるべきか否か、ジョブがローカルであるべきかまたはリモートであるべきか、ならびに限定されるものではないが入力データのソース（複数可）及び出力データの行先（複数可）を含む、任意の/または他の変数を決定し、その後特定されたリソースを用いてジョブプロセス508を作成する。各ジョブプロセス508は、コンパイルされた出力を特定のターゲット環境において実行して、特定のターゲットアルゴリズム（すなわち、コンパイルされた命令によって特定された）を実装する。実施形態においては、ジョブプロセス508の一部または全てが、分散型データエンジン150に埋め込まれるビッグデータアナライザ170によって実行され得る。

30

40

【0274】

したがって、このフレキシブルなアーキテクチャにおいては、以下のシナリオのうちの少なくとも各々が可能である。単一のブロックまたはモジュールが、分散型の様式で複数のプロセッサにまたがって、複数のジョブプロセス508として実行されるシナリオ、単一のブロックまたはモジュールが、単一のプロセッサにおいてシリアルで、複数のジョブプロセス508として実行されるシナリオ、単一のブロックまたはモジュールが、単一のプロセッサにおいて、単一のジョブプロセス508として実行されるシナリオ、複数のブロックが、分散型の様式で複数のプロセッサにまたがって、対応する複数のジョブプロセス508として実行されるシナリオ等。また、概して、単一のブロックまたはモジュール

50

に対応するジョブプロセス508は、同一のターゲット環境（例えば、Python、PySpark等）において実行されることが企図されるが、モジュール内の異なるブロック、またはプロジェクト内の異なるモジュールに関する全てのプロセスが、単一のターゲット環境において動作するという要件は存在しない。一部の事例においては、例えば、ある特定のターゲット環境が、モジュール内の1つのブロックと関連付けられるターゲットアルゴリズムを実現するのにより良好に適し得る一方で、モジュール内の他のブロックは、異なるターゲット環境においてより効率的に実行され得るため、ブロックは、それぞれのプロパティにおいて、異なるターゲット環境を特定するように構成され得、最終結果は、異なるブロックと関連付けられるジョブプロセス508は、異なる環境において実行されるというものになる。

10

【0275】

なおも更に、多くの事例においては、特定のモジュールまたはプロジェクトと関連付けられるブロックは、全て分散型であるか、あるいは全てローカルである、ジョブプロセス508において実行されることが予期される。しかしながら、上の記載の見地から、モジュールまたはプロジェクトの1つ以上のブロックをローカルで実行し、一方でモジュールまたはプロジェクトの他のブロックを1つ以上のリモートのプロセッサ（例えば、クラウドコンピューティング環境）において実行することで効率性を獲得できることは明白であろう。例えば、第1のブロックが、Analytics Service 500からは離れて配置されるデータベース（または他のデータストレージ）から特定されるデータを回収するようにクエリを実施し、第2のブロックが、そのクエリによって返されるデータに対して分析を実施するというダイアグラムを考慮されたい。このクエリを、データベースから離れたプロセッサにおいて実施するよりも、データベースに対してローカルであるプロセッサにおいて（例えば、データベースの埋込式DDEにおいて）ジョブプロセス508を作成して、その埋込式DDEに対してローカルであるデータにクエリを実施する方がより効率的であり得る。これは特に、ユーザが現在いるプロセス制御環境以外のプロセス制御環境に関するデータをユーザが照会するとき等、データベースが別の地域にある場合に当てはまり得る。次いで、クエリの結果は、同一のジョブプロセス（及びプロセッサ）、同一のプロセッサ内の異なるジョブプロセス、または異なるプロセッサ内の異なるジョブプロセス（例えば、ローカルのプロセッサ）のいずれかにおいて、第2のブロックに従って分析され得る。

20

30

【0276】

ジョブ実行部エンティティ528はまた、特に分散型処理アプリケーションにおいて、ジョブプロセス508の記録を取り、データ依存性の記録を取る。したがって、各ジョブプロセス508が完了する度に、出力データがジョブ実行部エンティティ528に返される。ジョブ実行部528は、そのデータの一部または全てが別のジョブプロセス508に受け渡されるか、かつ/あるいはウェブクライアントプロセス502による、アプリケーションエンティティ514における表示のために、ウェブサーバプロセス504に戻されるかを決定する。ジョブプロセス508の実行中、ジョブ実行部エンティティ528は、ステータス情報を受信し得、このステータス情報は、アプリケーションエンティティ514における表示のために、ウェブサーバプロセス504に戻され得る。ジョブプロセス508はまた、特にジョブプロセス508が、明確に停止されるまでは連続的に実行されるオンラインダイアグラム（またはオンラインダイアグラムの一部分）を実行しているときに、ジョブ実行部エンティティ528に予測的データを返し得る。予測的データは、予測的データが生成される度に、ジョブ実行部エンティティ528に戻され得、アプリケーションエンティティ514における表示のためにウェブサーバプロセス504に渡され得るが、一方で、予測的データは同時に、プロセス制御環境内のロケーション（例えば、DDEストレージロケーション、制御機能を実施するコントローラ等）に戻されて、オンラインダイアグラム内のブロックのうちの1つの関数として書き込まれ得ることに留意されたい。実施形態においては、ステータス、状態、及び/または予測的データは、ジョブプロセス508と、ジョブ実行部エンティティ528との間で、ジョブ作成中に確立されるバ

40

50

ックチャンネルストリーミング通信チャンネルを経由して受け渡されて、ジョブプロセス508が、互いに、かつ/あるいはジョブ実行部エンティティ528と通信することを可能にする。

【0277】

下の実施例を通じて、Analytics Service 500を通るデータのフローは、少なくとも、構成されるブロック及び/またはモジュール、ブロックまたはモジュールの各々について構成されるプロパティ、ならびに実行されるダイアグラムがオフラインダイアグラムであるか、またはオンラインダイアグラムであるかに、ある程度依存することが明白になるであろう。

【0278】

実施例1 - オフラインダイアグラムの作成

図5Dは、ブロック604a~604g及びワイヤ606a~606hを含む、構成されたオフラインダイアグラム602をその上に有する、例示的キャンバス600を描写する図である。図5Aを再度参照すると、キャンバス600は、ウェブクライアントプロセス502によって表示され、特に、アプリケーションエンティティ514と協働する表示エンティティ510によって表示される。アプリケーションエンティティ514は、データサービスエンティティ516を介して、データ(ブロック定義ライブラリ252を含む)を受信するが、このデータサービスエンティティ516は、サーバ通信エンティティ518を介して、ウェブサーバプロセス540と連通している。

【0279】

概して、図5Dに描かれるようなオフラインダイアグラム602を構築するためには、2つの方法が存在する。第1に、ユーザは、ブロック定義切り替え部248mを切り替えて、ブロック定義ライブラリ252を開くことができる(図4Dを参照)。ブロック定義ライブラリ252から、ユーザは、ブロック604a~604gを1つずつ選択し、ブロック604a~604gをキャンバス600上に配置することができる。次に、ユーザは、様々なブロックを接続するために、ワイヤ606a~606hを配置することができる。ブロック604aの出力608aとブロック604bの入力610bとの間にワイヤ606aを配置、ブロック604bの出力608bとブロック604cの入力610cとの間にワイヤ606bを配置、ブロック604bの出力608bとブロック604dの入力610dとの間にワイヤ606cを配置、ブロック604bの出力608bとブロック604eの入力610eとの間にワイヤ606dを配置、ブロック604cの出力608cとブロック604fの入力610f2との間にワイヤ606eを配置、ブロック604dの出力608dとブロック604gの入力610gとの間にワイヤ606fを配置、ブロック604dの出力608dとブロック604fの入力610f1との間にワイヤ606gを配置、及びブロック604fの出力608f1とブロック604gの入力610gとの間にワイヤ606hを配置。このようにブロック604a~604g及びワイヤ606a~606hを配置した後、オフラインダイアグラム602は未構成のまま留まる。

【0280】

あるいは、ユーザは、ブロック604a~604gを1つずつ選択及び配置し、各ブロックをキャンバス600上に配置する度に各ブロック604a~604gについてのプロパティを構成し、任意選択で(また、大部分の事例において)、ブロックについてのプロパティを構成した後、ブロック604a~604gの各々を実行することができる。

【0281】

ユーザは、ブロックをクリックし、ブロックに関するブロックプロパティウィンドウ612において、ブロックと関連付けられるプロパティを特定することによって、ブロック604a~604gの各々を1つずつ構成することができる(プロパティの構成前にブロック604a~604gを配置し、ワイヤで繋げるか、あるいは1つずつ配置し、次のブロックを配置する前に各々についてのプロパティを構成するかに関わらず)。図5Dにおいては、ブロック604a、すなわちクエリブロックが選択され、クエリブロック604aに関するブロックプロパティウィンドウ612が表示されている。図5Dに描写される

10

20

30

40

50

例示的キャンバス600においては、クエリブロック604aに関するブロックプロパティウィンドウ612は、ユーザがクエリを入力することができる、クエリプロパティ入力領域614を含む。クエリ言語の詳細な説明は、明細書の後の節における考察のために保留するものの、現在の目的としては、クエリプロパティ入力領域614に入力されたクエリが、クエリブロック604aの実行に際して照会されるデータのロケーションを特定し得ると仮定するのに十分である。ブロックプロパティは、ブロックと関連付けられる構成ファイル内のパラメータとして記憶される。構成ファイルは、ウェブクライアントプロセス502を実行するウェブクライアントに存在し得るか、あるいはウェブサーバプロセス504を実行するウェブサーバに配置され得る。

【0282】

クエリブロック604aのプロパティを構成した後、ユーザは、典型的には、次の下流ブロックを配置する前、ならびに/または次の下流ブロック(この場合、ブロック604b)のプロパティの構成を開始する前に、ブロック604aを実行する。ブロック604aを実行するために、ユーザは、評価ユーザ制御部248iをアクティブ化する。そうすることによって、ブロック604aと関連付けられる構成ファイルは、ウェブサーバプロセス504のデータサービスエンティティ520を介して、実行サーバプロセス506に送信される。構成ファイルは、実行サーバプロセス506によって、例えばJSONファイルとして受信され、ジョブリスナエンティティ522が、ジョブマネージャエンティティ524を作成して実行要求を処理する。ジョブマネージャエンティティ524は、適切なダイアグラムコンパイラエンティティ526に、構成ファイル中のパラメータによって特定されるターゲット環境にとって好適な、実行可能な命令(すなわち、コンパイルされたクエリコード)へと、構成ファイルをコンパイルさせる。ダイアグラムコンパイラエンティティ526は、コンパイルされたコードを含む出力ファイルを返し、また、コンパイルされたコードが適しているターゲット環境を示すメタデータを返し得る。(あるいは、ジョブマネージャエンティティ524は、コンパイルプロセス中、ターゲット環境の知識を保持してもよい。)ジョブマネージャエンティティ524は、出力ファイルをジョブ実行部エンティティ528に受け渡し、このジョブ実行部エンティティが、ターゲット環境に関する構成ファイルのパラメータに従って、コードがローカルで実行されるべきか、またはリモートで実行されるべきか、コードが分散型システムで実行されるべきか等を決定し、出力ファイルに含まれる命令を実行するプロセッサ及びメモリリソースを選択及び確保する。次いで、ジョブ実行部エンティティ528は、ジョブプロセス(複数可)508を作成して、出力ファイルを実行する。

【0283】

出力ファイルの実行が完了したとき(オフラインブロックまたはモジュールの場合において)、または結果が利用可能になると(オンラインモジュールの場合)、これらの結果はジョブ実行部エンティティ528に返される。この事例においては、保留中の更なる実行は存在しなかったため(ダイアグラムはオフラインダイアグラムであり、実行したばかりのブロック、すなわちクエリブロック604aは構成された唯一のブロックであるためである)、結果は、サーバ通信エンティティ508を介してウェブサーバプロセス504に返され、最終的には、ウェブクライアントプロセス502におけるアプリケーションエンティティ514に返される。出力データは、クエリによって返されたデータの全てを含み得るが、(限定されるものではないが)クエリによって返されたデータの各列の平均値、標準偏差、及び中央値等の、クエリデータについての様々な統計的情報(メタデータ)も含み得る。実施形態においては、メタデータ及び結果のデータは、互いからは独立して記憶される。これによって、有利なことに、ユーザは、結果として生じたデータの全てのロード(時間及び/またはリソース、すなわちコンピュータメモリ集約的であり得る)を必ずしも伴わずに、結果として生じたデータについての情報をレビューすることが可能となる。ユーザは、ブロック上のブロック結果閲覧制御部212a(図4Aを参照)をクリックすることで、返されたデータ及び/または返されたデータについてのメタデータを閲覧することができる。デフォルトビューは、結果に含まれる変数の各々についての結果に

10

20

30

40

50

関するメタデータを含み得、ユーザが、結果として生じたデータの1つ以上の選択される部分を選択して、データの全てをロードする必要無しに、ロード、ならびに所望される場合、閲覧すること（例えば、グラフ、図表、表等の形態で）を可能にし得る。加えて、アプリケーションエンティティ514への結果の返却は、ワイヤ606aによって、データ及び/またはメタデータが利用可能にされる、次の下流ブロック（この場合、ブロック604b）のプロパティの構成を可能にする。

【0284】

ブロック（例えばブロック604a）の実行は、次の下流ブロック（例えばブロック604b）を構成するために、全ての事例において必要とされるわけではないことに留意されたい。下流ブロックを構成及び/または実行できるようになる前に、あるブロックを構成及び/または実行しなければならないかどうかは、本開示全体の見地から理解されるように、データ依存性に依存することになる。

【0285】

実行中、ジョブプロセス508は、様々な分散型データエンジン150と通信して、ビッグデータストレージ155に記憶されているデータを受信またはそれにアクセスすることにも留意されたい。ジョブプロセス508が埋込式ビッグデータアナライザ170において実行されている事例においては、ビッグデータアナライザ170は、ビッグデータストレージ155に対して、直接または1つ以上のビッグデータ要求サービサ165を介して、通信可能に連結され得るあるいは、ジョブプロセス508が、埋込式ビッグデータアナライザ170の一部ではないプロセッサにおいて実行されている事例においては、ジョブプロセス508は、ネットワークを介してDDEと通信可能に連結され得、ビッグデータ要求サービサ165を介して、データを要求し得る。

【0286】

オフラインダイアグラム602のブロック604b~604gの各々のプロパティの構成及び実行は同じ様式で継続し、極めて詳細に述べる必要はない。

【0287】

A. 探査及びモデル化のためのオフラインダイアグラムの使用

Offlineダイアグラムを扱っている、DDE User Interface Applicationのユーザは概して、以下の2つのうちの一方、または両方を行うことを期待している。（1）様々なデータのセットを探査して、様々なデータセット及び/または事象間の関係性を探究及び発見すること、ならびに（2）リアルタイムの連続的予測能力を実装することを目的として、それらの関係性のモデルを作成すること。後者を実現するには、前者への取り組みに費やされた時間の何らかの尺度が必要となる。すなわち、（全てではないが）多くの事例においては、ユーザは、データセットを探査して、様々な分析ツール（主構成要素分析、フーリエ解析、回帰分析等）を用いてデータ間の関係性を見つけ、予測値を有するように考えられる関係性が見つかった場合、ユーザは、その分析をモデルとして実装し、それをプロセスのリアルタイム連続分析に採用する。

【0288】

実施形態においては、Analytics Service 500は、リアルタイムデータもしくはヒストライズされたデータ、または両方であるかに関わらず、データの分析を自動的に実行して、プロセス環境内の変則、異常状態、データ間の関係性、及び/または特定のデータセットの特色を発見する、ツールを含み得る。ある実施形態においては、ブロック定義ライブラリ252は、データを観察し、そのデータ中の不規則なスパイクまたは下落等の、レビューしているデータセットの文脈における変則を発見するブロック255を含む。オフラインダイアグラム中のそのような変則発見ブロックは、ヒストライズされたデータを入力として受信して、例えばPLS、PCA、及び他の種類の分析を含む上記の技法のうちの一部を更に用いて、探査するのに興味深くあり得るヒストライズされたデータの特色を探ることができる。代替的または追加的に、オンラインダイアグラム中の変則発見ブロックは、ヒストライズされたデータ及びリアルタイムデータの両方を受信し、ヒストライズされたデータをリアルタイムデータと比較して、ヒストライズされたデ

ータに対するリアルタイムデータ内の変則を探ることができる。どちらの場合でも、変則発見ブロックの出力は、変則的事象が起こった時間、変則的データのソース（複数可）（例えば、どの測定値及び/またはデバイスがその変則と関連付けられるか）等のうちの1つ以上を含み得る。

【0289】

ある実施形態においては、ブロック定義ライブラリ252は、様々なデータ間の関係性を自律的に発見するブロック255を含む。関係性発見ブロックは、ランダムまたは準ランダムのデータのセット（時間で整列されている）を選択し得、これらのデータのセットに対していくつかの異なる分析を実行して、ユーザにとって興味深くあり得る関係性、特に因果関係または予測的關係を探ることができる。無論、データのセットの真ランダムな選択は、有用な結果を多く生み出す可能性は低い。しかしながら、データセットの選択は、例えば、物理的ロケーション、プラントの階層構造、またはデータ間の関係性が存在する可能性があることを示唆し得る他の証拠によって左右され得る。

10

【0290】

一部の実施形態においては、関係性発見ブロックは、変則発見ブロックと協働して、データ間の関係性を発見し得る。例として、変則発見ブロックは、様々な変則が様々なデータセットで起こる時間を出力し得る。変則発見ブロックは、ある実施形態においては、変則的事象の時間を記憶し、事象変則的事象が異なるデータセットにおいて同時または近接した時間で起こる場合、変則発見ブロックの1つの出力は、互いに対して時間的に近接して変則的事象を経験したそれらのデータセットに対応するデータセットのリストであり得、変則発見ブロックの別の出力は、それらの変則が起こった時間のリストであり得る。関係性発見ブロックは、変則発見ブロックの出力を入力として受信し得、関連する期間の関連するデータセットを、可能性としては、他のデータセット及び/またはより長い期間（例えば、変則が起こった時間の前後でいくらか長い時間延長した期間）と共にロードして、様々な種類の分析をそのデータに適用し、データ間の因果関係または少なくとも予測的關係を探究し得る。

20

【0291】

あるいは、関係性発見ブロックの1つ以上のプロパティは、ユーザがデータの関係性を見つけることを希求する、特定のデータセット（例えば、特定のプロセス変数についての値）を示し得る。関係性発見ブロックは、物理的ロケーション、プラントの階層構造、論理関係等によって関連する他のデータと、特定されたデータセットとの間の関係性を自律的に探究し、ある特定の判定基準（例えば、95%の予測信頼度、0.75の相関値等）を満たす1つ以上の因果関係または予測的關係が見つかるまで、そのデータに対して異なる分析を実施し得る。

30

【0292】

無論、関係性発見ブロックまたは変則発見ブロック等の任意の自動的ブロックは、オフラインダイアグラムまたはオンラインダイアグラムにおいて動作可能であるように企図される。

【0293】

いずれにせよ、典型的には、ユーザは、キャンバス600上のダイアグラム602等のオフラインダイアグラムを、キャンバス245上に作成する。ダイアグラムの作成は、1つ以上のブロック604をキャンバス600上に配置し、各ブロック604のプロパティを1度に1つのブロックずつ構成し、各ブロック604を、次のプロパティを構成する前に実行することで、概して上記のように進行する。ユーザは、ブロック604と関連付けられるデータ要約ユーザ制御部をクリックして、ブロックのデータについての様々な統計（例えば、平均値、中央値、標準偏差等）を見ることで、ダイアグラムにおける任意の点で、データを批判的に眺めることができ、あるいは、データを探索するために、ブロック604の出力608に（ワイヤ606を介して）ブロックを接続することもできる（例えば、図5Dのブロック604e）。ユーザが、探索ブロックが接続されているブロックまで、それを含んで、オフラインダイアグラム中のブロックを実行したと仮定すると、ユー

40

50

ずは、その探査ブロックを使用して、データを様々なグラフで閲覧したり、データと関連付けられるメタデータを見たり等することができる。例えば、オフラインダイアグラム 602 においては、探査ブロック 604 e は、データがブロック 604 b によって処理された後のデータをユーザが見ることを可能にする。同様に、探査ブロック 604 g は、PLS ブロック 604 f 及び Column ブロック 604 d からデータを受信する。後者の場合、探査ブロック 604 g は、ユーザが、PLS モデルの出力を視覚化し、その出力を、PLS モデルを作成するのに使用された (ブロック 604 d からの) その変数の実測値と比較することを可能にし得る。ユーザは、データのグラフを見ることで、PLS モデルがプロセスにおける何らかの予測値を有するように、プロセスを正確に表しているかどうかを判定することができ得る。

10

【0294】

実施形態においては、ブロック 255 と関連付けられるデータ要約ユーザ制御部は、ブロックの種類及び機能に対して特異的である。データをロードするブロック (例えば、クエリブロックまたはロードブロック) は、例えば、データ要約ユーザ制御部が、アクティブ化されたときに、クエリによってロードまたは返されるデータを特徴付ける、様々な統計的データの表示をもたらすように、プログラムされ得る。このデータは、実施形態においては、データに含まれる各データソースについての平均値、中央値、標準偏差、最大値、及び最小値を含み得る。対照的に、データに対して分析 (例えば、PLS または PCA 分析) を実施するブロックは、データ要約ユーザ制御部がアクティブ化されるとき、異なるデータを表示し得る。分析ブロックは、1 つ以上の R - 二乗値、PLS 及び / または PCA 分析に関する係数、分散値、観察カウント (例えば、特定のデータソースに関して、時系列値がいくつ含まれているか)、ならびにユーザがどのデータ (すなわち、どのデータソースからのデータ) を閲覧するかを選択できるカスタマイズ可能なグラフを表示し得る。探査ブロックはまた、実施形態においては、そのデータ入力 (複数可) が接続するデータ出力 (複数可) に応じて、異なる方法で振る舞う。すなわち、探査ブロックにおいて表示される、ならびに / または表示にとって利用可能であるデータのフォーマット及び種類は、探査ブロックへの入力を生成するブロックの種類に依存し得る。

20

【0295】

図 5 D を参照すると、オフラインダイアグラム 602 は単に、多くの可能性のうちの 1 つのダイアグラム構成に過ぎないが、その機能をここで一般的に説明する。オフラインダイアグラム 602 は、クエリブロック 604 a で開始され、このクエリブロックの目的は、特定のデータのセットを探し、それを分析のためにロードすることである。この特定のデータのセットとは、単に 1 つの例を提供するだけだが、3 つの特定の時間において異常なばらつきを呈示した特定のプロセス変数に関する、ヒストライズされたプロセス制御データであり得る。しかしながら、概して、クエリブロック 604 a は、限定されるものではないが、ビッグデータストレージ 155 の任意のデータ、データソース 168 のうちの 1 つ以上に記憶及び / またはキャッシュされているデータ、気象データソース、サプライチェーンデータソース、デリバラブルな追跡データソース等の外部データソースに記憶されているデータを含む、任意の記憶されているデータを照会できる。実施形態においては、データベースまたはドキュメント記憶 (図示せず) は、Analytics Service と関連付けられるオンライン及び / またはオフラインダイアグラムについて記載するドキュメント (例えば、JSON ドキュメント) を記憶し、この場合、クエリブロック 604 a は、そのデータベースまたはドキュメント記憶を、実施される計算の種類、計算において使用されるデータのソース (複数可)、結果の質等に関して照会するように構成され得る。後者の種類のクエリは、有利なことに、ダイアグラムの履歴または知識ベースの構築を可能にし得る。クエリブロック 604 a は、そのプロセス変数及びいくつかの他のプロセス変数の値を選択するように構成することができ、更には、サンプルレート、ならびに / または特定のプロセス変数及び他のプロセス変数の値のみを、例えば、3 つの事象の各々の 1 時間前から 1 時間後に対応する時間の範囲内で選択するように構成することもできる。すなわち、クエリは、離れた複数の時間範囲を選択し、任意の数の変数 (

30

40

50

このうち少なくとも1つは、ユーザが予測を所望するものである)について、それらの時間範囲中に生成されたデータを探し得る。

【0296】

いったん実行されると、クエリブロック604aによって受信されたデータは、Fillブロック604bによって使用され得る。例示的ダイアグラム602においては、Fillブロック604bが、変数が値を有しないが、その他の変数のうちの1つは値を有した時間に対応する、変数についてのデータを埋め得る。すなわち、照会された値のうちの1つが、別の値よりも高頻度でサンプリングされている場合、Fillブロック604bは、より高頻度のサンプリングされた変数の頻度に釣り合うように、より低頻度のサンプリングされた変数について、値を挿入し得る。実施形態においては、Fillブロック604bは、より低頻度のサンプリングされた変数の値を外挿し得る。例えば、より低頻度のサンプリング変数のサンプル間に、より高頻度のサンプリング変数の値が4つ発生する場合、ブロック604bは、より低頻度のサンプリング変数の2つの連続する値(例えば、1.0及び6.0)の間での差異を見つけ、4で割り、「欠けている」4つの値を連続的により大きくなるか、またはより小さくなる値(例えば、2.0、3.0、4.0、及び5.0)で埋めてもよく、これにより、より高頻度のサンプリング変数の全ての値に対して、より低頻度のサンプリングされた値について、対応する値が存在することになる。(無論、リアルタイム分析においては、後の値は未だ知られていないため、この方法は、値を埋める方法として妥当ではない。)他の実施形態においては、Fillブロック604bは、単に、より低頻度のサンプリングされた値について最も直近の値で、「欠けている」値を埋める場合がある。上で使用される実施例においては、欠けている値は、各々、1.0という値で埋められる。

【0297】

データを見つけ、値を挿入して、欠けている値の無いデータセットを作成した後、Fillブロック604bの出力は、3つのブロック、探査ブロック604e、第1の列ブロック604c、及び第2の列ブロック604dに対して提供される。探査ブロック604eについては上に記載されている。第1及び第2の列ブロック604c及び604dは、それぞれ、データから変数(列)を引き出すように動作する。データは、各列が変数を表す表に記憶され、各列内の行は、異なる時間におけるそれぞれの変数の値を表す(すなわち、時系列データ)。第1の列ブロック604cは、例えば、異常なばらつきを呈示したプロセス変数以外の変数についてのデータを含む列の全て(すなわち、変数の全て)を選択し得、一方で、第2の列ブロック604dは、例えば、異常なばらつきを呈示したプロセス変数についてのデータを含む列を選択し得る。

【0298】

列ブロック604c及び604dの各々の出力は、PLSブロック604fに送信される。ブロック604fの入力610f2は、ある実施形態においては、独立変数、説明変数、または入力変数の値を受け入れるように構成され得、一方で、ブロック604fの入力610f1は、従属変数、被説明変数、または出力変数の値を受け入れるように構成され得る。モデルブロック(例えば、オフラインダイアグラム602におけるPLSブロック604f)に対する入力の特定の配列及び/または機能がダイアグラム602において示されているが、入力の配列及び機能は、ブロックの機能、入力の種類、入力の数等に応じて異なり得る。

【0299】

ブロック604fの出力608f1は、探査ブロック604gにワイヤでつながれる。したがって、探査ブロック604fは、PLSブロック604fによって出力された値、及び異常なばらつきを呈示した変数の値を受信することになる。ユーザは、探査ブロック604gを用いることで、ブロック604fの出力が、任意の所与の時間において従属変数と比較してどうであるかを見ることができ、予測値を決定することができる。ブロック604fの出力608f2は、図5Dの他のどのブロックにもワイヤでつながれていないが、ブロック及びワイヤの一部のプロパティを例証するものである。具体的には、出力6

10

20

30

40

50

08f2は、オブジェクト出力である。オブジェクト出力は、コードを出力するように動作可能であり、この事例においては、例えば、オブジェクト出力は、プログラムされる通り、モデル、データ、及び出力を含む感度モデルを出力する。無論、これは、出力608f2に接続されるワイヤが、そのワイヤを通じてモデルを必ず運搬できることを意味する。そのワイヤを新しいブロッククラウドの入力に接続することで、例えば、上流のダイアグラム全体を伴うブロックがプログラムされて、機能性及びデータの全てを含むブロックが作成される。

【0300】

したがって、現在明白であるように、ユーザは、オフラインダイアグラム602を反復的に修正し、結果を探索して、所与の変数について所望の予測値を有するダイアグラムを構築することができる。そのようにする際、ユーザは、異なるデータのセット、同じデータのセットの異なるサブセット、異なるモデリング/分析技法等を使用できる。換言すれば、ユーザは、異なるデータのセットを照会またはロードすることができ(例えば、オフラインダイアグラム602における、ブロック604aの交換、または修正、または追加)、異なる変数の予測を試みることで、データを別様に分割でき(例えば、ブロック604c及び604dの各々において選択される列を変更することによって)、異なる種類の分析(PCA、PLS等)、及び/または特定の分析について異なるプロパティを試みること等ができる(例えば、ブロック604fを交換または修正することで)。

【0301】

上の実施例の見地から、DDE User Interface Application及び付随するAnalytical Servicesが、膨大な量のデータが収集される多くのシステムにおいて、好都合な有用性を有することが明確となろう。プロセス制御システムに関しては、プロセス制御環境は、比較的近年、収集及び記憶するデータの量を拡大しており、一部の事例においては、環境内で生成される全てのデータを含むようになっている。過去のシステムもかなりのデータを生成していたが、それらのシステムはプロセス制御に関してそのデータの僅かな部分しか使用しておらず、例えば、それをプロセス制御アルゴリズムに供給し、その一部をオペレータワークステーションに表示していた。相対的にデータのほとんどが、後の分析及び/または使用のために記憶されることはなかった。1秒間に1回サンプリングされるある変数の値は、例えば、事後分析に対する必要性和、ストレージ及びネットワーク能力の限界とのバランスを取るために、1分間に1回しか記憶されない場合がある。更に、記憶されたデータは、多くの場合圧縮され、信頼性が低いデータを結果としてもたらした。

【0302】

対照的に、現在のシステムは、プラントにおいて生成されるデータの遥かに多くを記憶する。実施形態においては、そのようなシステムは現在、サンプルが記録される度に全ての変数値を記憶し、また、オペレータ入力、オペレータ対オペレータのメッセージング、ビデオ等、かつてヒストライズされたことのないデータも記憶する場合がある。本明細書に記載されるDDE User Interface Application及びAnalytical Servicesは協働して、他のデータ(例えば、他のプラントからのデータ、気象データ等)も加えたこれらのデータの全ての探索、及び以前には見過ごされるか確認不可能であった可能性がある関係性の発見を容易にする。様々なプロセス値間の関係性、ならびにプロセス値の、及びプロセス値間の相互作用を発見することの結果として、プラントエンジニア、オペレータ、及び保守要員は、プロセスプラントをより良好に設計、構築、運転、及び整備することができ、これはひいては、より安価、より効率的、より容易に運転及び整備され、より良好に製品を生産し、環境に対する負の影響がより少なく、かつ職員及び周辺のコミュニティにとってより安全であるプロセスプラントにつながる。

【0303】

B. オンラインダイアグラムの作成

無論、発見された関係性をプロセスのオペレーションの改善に向けて採用し得る1つの

方法は、発見された関係性を用いて、リアルタイム連続予測を実施することによるものである。具体的には、値の1つのセットを、別の値（複数可）（または何らかの事象（複数可）の発生）を予測するために使用できるように、プロセスの履歴データにおいて、プロセス値間または他のデータ間の1つ以上の関係性を発見することで、発見された関係性を使用して、プロセスからのリアルタイムデータを見て、同じ値（複数可）（または同じ事象（複数可）の発生）を予測することができる。DDE User Interface Application及びAnalytic Servicesが、下に記載されるように、予測的分析を実施するための、発見された関係性の使用を容易にする。

【0304】

DDE User Interface Applicationは、それがオフラインダイアグラム（オフラインダイアグラム602等）をオンラインダイアグラム（すなわち、少なくとも1つのリアルタイム値を用いて、プラントオペレーションのある状態を予測するもの）に変換できるようにさせる機能性を含む。上記のように、オンラインダイアグラムは、少なくとも1つのリアルタイムデータソースにバインドする（完全に、ヒストライズされたデータにのみバインドするのではなく）という点においてオフラインダイアグラムとは異なり、リアルタイムの連続予測的出力を提供し、この出力を、制御アルゴリズムにおいて閲覧、記憶、及び/または使用して、アラーム、アラートをトリガし、プロセスプラントのオペレーションにおける変更をもたらすことができる。

【0305】

オフラインダイアグラムが少なくとも1つのモデル生成ブロックを含む限り、DDE User Interface Applicationのユーザは、Online切り替えユーザ制御部248fをアクティブ化することができ、オフラインダイアグラムは、自動的にオンラインダイアグラムへと変換され、キャンバス245上に表示されることになる。再度図5Aを参照すると、オフラインダイアグラムからオンラインダイアグラムへの変換は、実施形態においては、ウェブサーバプロセス504のダイアグラムトランスフォーマエンティティ530を経由して実現される。

【0306】

その最も単純な実施形態においては、ダイアグラムトランスフォーマエンティティ530は、モデル生成ブロック（例えば、オフラインダイアグラム602中のPLSブロック604f）の場所を突き止め、あらゆる出力ワイヤ（例えば、ワイヤ606h）を取り除き、モデル生成ブロックの出力と、書き込みブロックの入力との間にワイヤを接続する。書き込みブロックは概して、モデルから出力された値を、データストレージロケーション、プロセス制御アルゴリズム（例えば、コントローラもしくはプロセス制御デバイスの機能ブロックにおいて、制御アルゴリズムを実行する）に、及び/またはDashboardに書き込む。ダイアグラムトランスフォーマエンティティ530はまた、オフラインデータローディングブロックを、対応するオンラインデータローディングブロックと交換し、このオンラインデータローディングブロックは、ヒストライズされたデータのバッチをロードするのではなく、プロセス制御環境からの少なくとも1つのリアルタイムデータソースにバインドする。

【0307】

しかしながら、様々な他の実施形態において、ダイアグラムトランスフォーマエンティティ530は、オフラインダイアグラムにおいて単に2つのブロックを交換すること以上のことを行うように動作可能である。一部の実施形態においては、ダイアグラムトランスフォーマエンティティ530は、オフラインダイアグラムからブロックを積極的に取り除いて、オンラインダイアグラムを作成する。例えば、モデルを作成するために、モデルブロックが2つの入力を必要とする場合（例えば、一連のX変数及びY変数）、これら2つの入力は異なるブロックで生成されることになる。しかしながら、開発したモデルを実装するとき（すなわち、オンラインダイアグラムにおいて）、モデルが1つの入力しか有さない場合、先には他方の入力にデータを提供していたブロックは最早必要ではなく、取り除くことができる。別の例としては、ダイアグラムトランスフォーマエンティティ53

10

20

30

40

50

0 は、オンラインダイアグラムを作成するとき、探査ブロック（例えば、図 5 D の探査ブロック 6 0 4 e）を取り除き得る。

【 0 3 0 8 】

一部の実施形態においては、ダイアグラムトランスフォーマ 5 3 0 は、各ブロックの種類についてオフライン及びオンラインのスキーマを提供するスキーマライブラリ（図示せず）を参照し得、これらのスキーマは、どの入力及び出力がブロックのオンラインバージョン及びオフラインバージョンと関連付けられるか、どのデータソース（複数可）にブロックがアクセスするか等を定義する。代替的实施形態においては、ブロック定義ライブラリ 2 5 2 の各ブロック定義 2 5 5 は、ブロックに関するオンライン及びオフラインのスキーマの両方を定義する。

10

【 0 3 0 9 】

一部の実施形態においては、ダイアグラムトランスフォーマエンティティ 5 3 0 は、オフラインダイアグラムのオンラインダイアグラムへの変換中に、任意選択のオペレーションを実施する、または実施しないように構成することができる。例えば、任意選択のオペレーションの一部としては、限定されるものではないが、予測値をプロセス制御アルゴリズムへの入力として使用できるように、予測値にタグ付けすること、予測値の連続グラフを D a s h b o a r d 上に出力すること、及び分散型データエンジンに予測値を書き込むことが挙げられ得る。一部の場合作いいては、ダイアグラムトランスフォーマエンティティ 5 3 0 によって生成される出力の少なくとも一部は、分析及び/または制御システムビジュアライゼーション（フェイスプレート、T r e n d i c o n 等）のために提供され得る。

20

【 0 3 1 0 】

データローディングブロック（ロード、クエリ等）がバインドするリアルタイムソース、モデルブロックの出力が公表されるロケーション（複数可）、及び/または出力データと関連付けられる仮想タグもまた、実施形態においては、所与の実施形態が各オプションを実装する程度に各々構成可能である。実施形態においては、各々は、ウェブクライアントプロセス 5 0 2 に存在するアプリケーションエンティティ 5 1 4 の実施形態において、例えば、モジュールプロパティ及び/またはブロックプロパティとして構成される。例えば、あるモジュールのモジュールプロパティは、図 5 F に概して描写されるように、データ公表ロケーション及びリアルタイムデータバインディングロケーションを含み得る。図 5 F においては、モジュールプロパティダイアログ 6 3 0 は、ユーザがモジュールプロパティ制御部 6 3 2 をアクティブ化したとき、モジュールについて表示される。モジュールプロパティダイアログ 6 3 0 は、モジュール名称フィールド 6 3 4 を含み、これによってユーザは、モジュールの名称を入力することができる。モジュールプロパティダイアログ 6 3 0 はまた、P o s t フィールド 6 3 6 も含み、これによってユーザは、実行中にオンラインダイアグラムによって出力されるデータがどこに書き込まれるべきか（すなわち、公表されるべきか）を特定することができる。データが公表されるロケーションは、データがヒストライズされるストレージロケーションか、公表された値を制御アルゴリズムに対する入力として使用して、プロセスプラントのオペレーションを制御する、プロセス制御環境内の制御モジュールか、またはその両方であり得る。ユーザ制御部 6 3 6 a は、利用可能なロケーションをグラフィカルなフォーマット（例えば、ツリーフォーマット、ディレクトリ構造、任意選択のロケーションのリスト等）で閲覧することで、データが公表されるロケーションをユーザが選択することを可能にする、エクスプローラウィンドウを開き得る。同様に、フィールド 6 3 8 は、オンラインダイアグラムの実行中に、オンラインデータのソースとしてバインドするデータパスをユーザが特定することを可能にする。ユーザ制御部 6 3 8 a は、同様に、データがそこから受信されるロケーションをユーザが選択することを可能にする、エクスプローラウィンドウを開く。

30

40

【 0 3 1 1 】

あるいは、オフラインブロックは、ブロックのオフラインバージョンに関するプロパティを含み得、ブロックのオンラインバージョンは、ブロックのオンラインバージョンにつ

50

いて、同様のプロパティを含み得る。例えば、ロードブロックは、ブロックのオフラインバージョンにおいて、「ソースパス」（例えば、\DDE_path\device_tag）プロパティを有し得るが、ブロックのオンラインバージョンにおいては、「デバイスタグ」（例えば、\unit1\device_tag）プロパティを有し得、これが、ロードブロックによる、オフラインブロックのソースパスに記憶されたデータに対応するソースからのリアルタイムデータの読み込みを可能にする。同様に、オンラインダイアグラムの書き込みブロックは、出力データが書き込まれるべきロケーションを特定するプロパティを有し得る。

【0312】

更に別の代替形態として、今度は図5Gを参照すると、オフラインダイアグラム640中のブロックは、各々、オフラインオペレーション及びオンラインオペレーションの両方についての構成を有し得る。例として、ロードブロック642が選択されてもよく、関連するブロックプロパティダイアログ644を有してもよい。ブロックプロパティダイアログ644は、ブロック名称フィールド646を含み、これによってユーザは、ブロックの名称を入力することができる。ブロックプロパティダイアログ644はまた、オフラインソースフィールド648も含み、これによってユーザは、オフラインダイアグラム中で、ロードされるべきデータがどこで見つかるかを特定することができる。ロードされるデータが記憶されるロケーションは、例えば、データがヒストライズされるストレージロケーションであり得る。ユーザ制御部648aは、利用可能なロケーションをグラフィカルなフォーマット（例えば、ツリーフォーマット、ディレクトリ構造、任意選択のロケーションのリスト等）で閲覧することで、そこからデータがロードされるロケーションをユーザが選択することを可能にする、エクスプローラウィンドウを開き得る。同様に、フィールド649は、オンラインダイアグラムの実行中に、オンラインデータのソースとしてバインドするデータパスをユーザが特定することを可能にする。ユーザ制御部649aは、同様に、データがそこから受信されるロケーションをユーザが選択することを可能にする、エクスプローラウィンドウを開く。無論、制御部648、648a、649、649aは、ブロック定義において特定される任意の制御部であり得、先に記載されたように、適切なプラグインの実行を呼び出し得る。

【0313】

実施形態においては、一方では、オフラインブロックがそこからデータを回収/照会する、記憶されているヒストライズされたデータに対するパスと、他方では、オンラインブロックがバインドし得る、リアルタイムデータの対応するソースとの関係性は、データソースマネージャモジュール532によって管理される。データソースマネージャモジュール532は、ウェブサーバプロセス502の一部として統合されてもよく、ダイアグラムトランスフォーマンティティ530の一部として含まれてもよく、あるいは単に、オフラインダイアグラムをオンラインダイアグラムに転換するとき、ダイアグラムトランスフォーマンティティ530が参照するルックアップテーブルであってもよい。少なくとも一部の実施形態においては、データソースマネージャモジュール532は、ビッグデータ要求サービス165と協働して、オンラインブロックの要件に従って、リアルタイムデータのストリームを設定する。

【0314】

図5Eは、図5Dのオフラインダイアグラム602に対応するオンラインダイアグラム622を描写する。オンラインダイアグラム622は、ブロック624a、624b、624c、624f、及び624h、ならびにワイヤ626a、626b、626e、及び626jを含む。ブロック及び/またはワイヤが、オフラインダイアグラム602中のものに対応する場合、ブロックと関連付けられる文字は同じものである。

【0315】

オンラインダイアグラム622のオペレーション及び実行は、上記のオフラインダイアグラム602の場合と概して同じ方法で進行する。プロパティの全てが構成されれば、オンラインダイアグラム622（及び構成ファイル（複数可））に記憶される、関連付けられ

10

20

30

40

50

るパラメータ)は、展開ユーザインターフェイス248jをエンゲージすることで、「展開」できる。制御部248jのアクティブ化は、構成ファイルを実行サービスプロセス506に送信し、ここで、構成ファイルは、構成ファイル中のパラメータに従ってコンパイルされる。しかしながら、1つ以上のジョブプロセス508として実行される場合、データ分析ネットワークインターフェイス175を介してビッグデータストレージ155からデータをダウンロードまたは照会する代わりに、ジョブプロセス508は、例えば、ビッグデータレシーバから、またはビッグデータ要求サービス165のうちの一つによってセットアップされるストリームを通じて直接、リアルタイムデータを受信する。また、出力データをジョブ実行部528に送り戻して、ダッシュボード上での表示のために、ウェブサーバプロセス504のサーバ通信エンティティ518を介して、アプリケーションエンティティ514にのみ送信するのではなく、ジョブプロセス508は、出力データを、ビッグデータストレージ155、及び/またはデータソースのうちの一つ以上(例えば、コントローラ、他のDDE等)に公表してもよい。

【0316】

工業パフォーマンス監視/分析用のクエリ言語

工業パフォーマンス監視/分析システム100が動作する、プロセス制御システムまたは環境と関連付けられるデータソースは、典型的には時系列データを提供するが、他の種類のデータも使用され得る(例えば、1つ以上のプロセスプラント5において別個に実行される複数のバッチからの横断的データ)。時系列データは、フィールドデバイス15~22及び40~46を含む、プロセスプラント5内の様々な種類の測定デバイスからの様々な種類のデータ測定値を含み得る。データソースは、それらのフォーマットにおいて、一般的に知られているものから著作権のあるフォーマット、例えば、OSISoftPI、DeltaVHistorian、SEEQ、FF3、及び/またはスプレッドシートに手動で捕捉されたフォーマットに至るまで、広い範囲にわたって異なり得る。一部のデータソースがリレーショナルデータベースを含み得る一方で、他のデータソースは非リレーショナル(NoSQL)データベースを含み得る。なおも更なるデータソースは、データベースではない場合があり、代わりにファイルディレクトリまたはドキュメント(例えば、XMLドキュメント)内のテキストを用いてデータを記憶する。クエリ構文における差異に加えて、データソースの多様性が、データが記憶される方法の差異により、根本的に異なるクエリ構造を必要とし得る。例えば、Mongo等のドキュメント指向非リレーショナルデータベースは、MySQL等のリレーショナルデータベースにおける、SQLクエリを通じてアクセス可能な表形式ではなく、ドキュメントに基づいてデータを記憶する。したがって、異なる種類のデータソースに記憶されるデータについてのクエリは、フォーマット及び構文の差異に加えて、別個の構造及び規則を採用する。様々なデータソースの下のクエリメカニズムの各々を使用して、その中に記憶されるデータにアクセスするのではなく、本明細書に記載される工業パフォーマンス監視/分析システム及び技法(ならびに、特に、DDE User Interface Application)は、図6Aに示されているように、標準化クエリを活用して、データソース702a~702fの各々とインターフェイスで接続する。

【0317】

図6Aは、クエリ実行サービス706を介してDDE User Interface Applicationのクエリブロック708と通信可能に接続される、様々なデータソース702a~702f及びデータプロバイダ704a~704dを示す、ブロックダイアグラムを例証する。データソース702a~702fの各々は、コンピュータ可読フォーマットでデータを記憶する、電子データソースである。一部のデータソース702e及び702fは、内部データ接続またはネットワークデータ接続を通じて等、介入データプロバイダ704を伴わずに、クエリ実行サービス706とインターフェイスで接続してもよい。他のデータソース702a~702dは、1つ以上のデータソースプロバイダ704a~704dを介して、クエリ実行サービス706とインターフェイスで接続する。データプロバイダ704a~704dは、それぞれのデータソース702a~702d

10

20

30

40

50

に、またはそこから、データにアクセス、データをサーチ、整列、読み込み、及び/または書き込むように構成され得る。一部の実施形態においては、データプロバイダ704a~704dは、データソース702a~702dの、それぞれのデータソースに特異的なクエリフォーマットを活用する、データソースに特異的なクエリを受信し得る。他の実施形態においては、データソースプロバイダ704a~704dは、カスタマイズされたクエリ言語を用いるクエリブロック708から標準化クエリ709を受信し、その標準化クエリを、ターゲットデータソース702a~702dの特定のクエリメカニズムを活用するそれぞれのデータソースに特異的なクエリフォーマットへと変換するように構成され得る。データプロバイダ704a~704dまたはデータソース702e~702fは、特定のデータソース702に関して、データソースに特異的なクエリを処理及び実行する、
10 実行エンジン（図示せず）を含み得る。実行エンジンは、データソース702自体の一部であってもよく、あるいは実行エンジンは、データソース702と関連付けられるデータプロバイダ704（例えば、データソース702bと関連付けられるデータプロバイダ704b）の一部であってもよい。各データソース702は、データベース、またはデータを含む複数のJavaScript Object Notation (JSON) ファイルを含むドキュメント記憶等のドキュメント記憶であってもよい。一部の実施形態においては、データソース702a~702fは、プロセスプラント5のフィールドデバイス15~22及び40~46によって生成されるデータストリーム等のライブデータソースも含み得る。そのようなライブデータソースは、得られたデータを構文解析し、具体的に
20 要求されたデータの一部分のみを保持または抽出することによって、照会され得る。

【0318】

データソース702a~702fは、任意の既知のフォーマットまたは今後開発されるフォーマットでデータを記憶するように構成され得る。データソース702a、702b、及び702eは、リレーショナルデータベースに記憶されたデータを表すように、データベースとして示されている。そのようなリレーショナルデータベースの例としては、MySQL、または表形式でデータを記憶し、SQLでフォーマットされたデータのクエリを活用する他のデータベースが挙げられる。データソース702c、702d、及び702fは、NoSQLデータベースまたは非データベースデータソース等の非リレーショナルデータベースを表すように、ファイルまたはデータエントリのコレクションとして示されている。非リレーショナルデータベースの例としては、表形式ではなく、ドキュメント
30 に基づいてデータを記憶する、MongoDBまたはCouchDB等のドキュメント指向データベースが挙げられる。極端に大きいデータまたは複合データセットの記憶またはサーチは、非リレーショナルデータベースを用いることでより効率的になるため、このようなデータベースが、ビッグデータ分析の場合頻繁に使用される。しかしながら、そのような非リレーショナルデータベースのデータは概して、表のフォーマット(SQLデータベースの基礎を形成する)では整理されていないため、そのような非リレーショナルデータベースの照会は、異なる技法及び異なる構文を要する。SQLの照会は広く使用され、かつ周知であるため、これもまたデータストレージ及び分析のために頻繁に使用されている。データソースに特異的なフォーマットを活用するデータソースに特異的なクエリへと
40 変換される標準化クエリ709のための標準化クエリフォーマットを使用することにより、本明細書に開示される本発明は、ユーザが、同じクエリフォーマットでSQLまたはNoSQLデータベース内のデータにアクセスすることを可能にする。標準化クエリフォーマットは更に、複数の異なる種類のデータソース702からのデータを、一貫したデータ構造及びフォーマットを有する1つのデータセットへと変換することを可能にする。したがって、クエリブロック708は、標準化クエリ709を用いて、異なるデータ構造、フォーマット、及びクエリメカニズムを有するデータソース702を接続する、スーパーコネクタとして働き得る。

【0319】

クエリブロック708は、上で考察したように、Data Analytics Studio内に実装されて、データソース702a~702fから獲得されるべきデータを
50

特定するブロックであり得る。例えば、クエリブロック708は、標準化クエリ709を特定するプロパティを含む、1つ以上のブロックプロパティを有する、構成ファイルであり得る。標準化クエリ709は、クエリブロック708によって参照される別個のファイル(Javascript Object Notationファイル)であってもよい。クエリブロック708は、代替的に、Distributed Data Engine 150のうちのいずれかの内部で動作するプロセスまたはルーチン等の、標準化クエリ709を含むか、またはそれを示す任意のオブジェクトであり得る。クエリブロック708は、標準化クエリ709を受信し(ユーザ選択または入力によって等)、標準化クエリ709は、標準化クエリフォーマットを活用する。標準化クエリフォーマットを活用することによって、データソース702a~702fは、ユーザ、技術者、またはデータソース702a~702f内のデータの具体的な種類もしくは構造の知識を必要とするデータ要求エンティティを伴わずに照会することができる。標準化クエリフォーマットは、一部のデータソースによって活用されている既存のクエリフォーマットであってもよく、またはどのデータソースによっても直接活用はされていない異なるクエリフォーマットであってもよい。後者の場合、標準化クエリフォーマットの構文を活用する標準化クエリは、データソース702a~702fからデータを獲得するように、直接実行可能または実装可能であるわけではない。標準化クエリは、データがそこから獲得される、データソース702a~702fのうちの1つ以上(または、データプロバイダ704a~704dのうちの1つ以上)を特定し得る。あるいは、クエリブロック708は、データソース702a~702fのうちの1つ以上を特定するプロパティを含み得る。

10

20

【0320】

クエリ実行サービス706は、クエリブロック708から標準化クエリ709を受信し、データソース702a~702fのうちの1つ以上を照会させる。クエリブロック708からの標準化クエリ709の受信は、データソース702a~702fから獲得されるべきデータを示すクエリパラメータを含むファイルを受信することを含み得る。データソース702a~702fを照会させることは、クエリパラメータを抽出し、これらのクエリパラメータに基づいて、1つ以上のデータソースに特異的なクエリを生成することを含み得る。次いで、データソースに特異的な各クエリは、実行サービス706によって実行されてもよく、あるいはデータソース702e~702fまたはデータプロバイダ704a~704dに送信されて、クエリパラメータによって示されるデータを獲得するように実行され得る。そのようなクエリパラメータは、1つ以上のデータソース702から獲得されるべき具体的なデータを示す場合があり、このデータは、被測定データ(フィールドデバイス15~22及び40~46からの測定値)、被測定データから計算もしくは別様に導出されたメトリクス、及び/またはデータソース702もしくはそこに記憶されているデータに関するメタデータであり得る。例えば、そのようなメタデータは、データソース702に記憶されるデータの種類、ソース、または質の指標を含み、データに対して実施される計算の種類を含み得る。そのようなメタデータは、システムのデータソース702からプロセスモデルまたは知識ベースを開発する際に有用であり得る。

30

【0321】

クエリ実行サービス706は、上に考察した実行サービスプロセス506であり得る。クエリ実行サービス706は、データサービス520、ジョブリスナ522、ジョブマネージャ524、ジョブ実行部528、またはデータソースマネージャ532を含む、分析サービス500の構成要素のうちのいずれを含んでもよい。例えば、クエリブロック708は、ユーザからの入力で、アプリケーション514によって作成することができ、JSONファイルで記憶される標準化クエリ709を含む。ユーザの要求または他のトリガする事象を受信した際、アプリケーション514は、標準化クエリ709を、ウェブサーバプロセス504から、またはそれを通して、実行サービスプロセス506のジョブリスナ522へと通信させてもよい(この実行サービスプロセス506は、実行サービス706として働き得る)。ジョブリスナ522は、標準化クエリ709を1つ以上のジョブマネージャ524に受信させ得、これが更に、1つ以上のジョブ実行部528に、標準化クエ

40

50

リ709を、1つ以上のジョブプロセス508と関連付けられる1つ以上のデータソースに特異的なクエリへと変換させ得る。次いで、ジョブプロセス508は、データソースに特異的なクエリを用いて、1つ以上のデータソース702（すなわち、データソース168またはビッグデータストレージ155）を照会させ得る。その後、1つ以上のデータソース702を照会することによって獲得されたデータは、ジョブプロセス508、実行サービスプロセス506、データサービス520、及び/またはアプリケーション514によって受信され得る。

【0322】

一部の実施形態においては、クエリ実行サービス706は、標準化クエリ709を、データソース702eまたは702fに本来備わる、データソースに特異的なフォーマットを活用する、データソースに特異的なクエリへと変換する。代替的实施形態においては、クエリ実行サービス706は、照会する1つ以上のデータソース702a~702fを決定し得、標準化クエリ709を、データソースに特異的なクエリへと変換するために、1つ以上のデータプロバイダ704a~704dに提供し得る。データプロバイダ704は、データソース702から分離されてもよく、またはデータソース702と組み合わせられてもよい。データプロバイダ704a及び704dは、それぞれ、データソース702a及び702dと通信可能に接続されるが、それでも分離されているように示される。対照的に、データプロバイダ704b及び704cは、それぞれ、データソース702b及び702cを含むか、またはそれと組み合わせられているように示される。例えば、データプロバイダ704bは、サーバ（データヒストリアン等）、またはデータソース720bがデータベースとして記憶される、データベースインターフェイスプログラムであってもよい。別の例としては、データプロバイダ704aも同様に、サーバ、またはサーバと通信可能に接続される外部メモリデバイスに記憶されているデータベース等、外部データソース702aと接続されるデータベースインターフェイスプログラムであってもよい。更に別の例としては、データプロバイダ704a~704dは、クエリ実行サービス706から、標準化クエリ709または標準化クエリ709からのクエリパラメータをそれらが受信したときに、標準化クエリ709をデータソースに特異的なクエリへと変換するアルゴリズムを含むジョブプロセス508であってもよい。

【0323】

プロセス制御及び分析における標準化クエリの使用を例証するために、図6Bは、Data Analytics Studioにおけるクエリブロック708のブロックダイアグラムを例証する。クエリブロック708は、1つ以上のデータソース702a~702fからプロセスデータを獲得する標準化クエリ709と関連付けられる。例えば、ユーザはクエリブロック708のフィールドに、標準化クエリフォーマットの標準化クエリ709を定義し得る。標準化クエリ709は、クエリブロック708と共に記憶され得、あるいはそれは別個のファイルに記憶されてもよい。例えば、クエリブロック708は、標準化クエリ709を含むJSONでフォーマットされたファイルの識別子を記憶するクエリプロパティを含む構成ファイルであってもよい。事象の発生時（時間、プロセス制御状態、またはユーザ選択の発生等）、標準化クエリ709は、クエリブロック708から特定され、要求されるデータをデータソース702a~702fのうちの1つ以上から獲得するように、クエリ実行サービス706に送信され得る。標準化クエリ709によって特定されたデータ、またはそのようなデータに関する情報（例えば、要約情報、検証メタデータ等）は、クエリブロック708または別のデータ受信エンティティへと返され得る。そのような他のデータ受信エンティティとしては、Data Analytics Studioの別のブロック、ジョブプロセス508、クエリ実行サービス706、データサービス520、アプリケーション514、プログラムメモリ、またはプロセスプラント5と関連付けられるデータを更に分析、呈示、もしくは記憶し得る任意の他のサービス、ルーチン、プロセス、もしくはデバイスが挙げられ得る。上に考察したように、プロセスデータは、一部の実施形態においては、1つ以上のデータプロバイダ704を介して獲得されてもよい。獲得されたデータは、クエリブロック708、クエリ実行サービス706、

10

20

30

40

50

または他のデータ受信エンティティにおいて、所望のフォーマットで受信され得、あるいはクエリブロック708、クエリ実行サービス706、または他のデータ受信エンティティによってフォーマットされ得る。例えば、クエリ実行サービス706は、標準化クエリ709を1つ以上のデータソースに特異的なクエリへと変換させ、そのようなソースに特異的なクエリの結果を所望の様式でフォーマットさせることによって、データフレームを生成し得る。そのようなフォーマットされたデータフレームは、一部の実施形態においては、クエリブロック708に返されてもよい。1つ以上のデータソース702から得られたデータは、データソースに特異的なクエリによって回収された複数のデータ点を含み得、このデータ点は、プロセスプラント5内の測定値またはそのような測定値に由来する値等、プロセス変数に対応し得る。そのようなデータ点は、時系列における点を表し得、各点は、それと関連付けられるタイムスタンプを有する。あるいは、そのようなデータ点は、ロケーション、プロセスバッチ、または他の特定する特色と関連付けられる横断的データを表し得る。標準化クエリを実行することによって獲得されるデータは、その変数構造を示すデータフレームと呼ばれる場合があり、表のフォーマットを使用する場合または使用しない場合がある。

【0324】

クエリブロック708は、標準化クエリを用いて得られたデータフレームをフィルブロック710に提供してもよく、このフィルブロックが、フィルブロック710と関連付けられる規則に従って、データセット内の空または非数(NaN)のエントリを埋める。これは、データをサニタイズし、またはデータフレーム中の不正確なデータまたは誤りの指標を確認するため、ならびに本明細書の他の場所に記載されるように、所望のサンプリングレートに到達するのに必要なデータ点を追加するためになされ得る。次いで、データフレームは、更なる分析及び/またはプロセス制御のために使用され得る。例証されるように、1つ以上のデータのセットが、独立変数ブロック712及び従属変数ブロック714によって、データフレームから選択され得る。例えば、従属変数ブロック712は、フィルブロック710から受信したデータフレームから、データマッチングで特定したパラメータまたは特性(例えば、圧力または温度の測定値)またはデータ列(ここでは、データフレームがデータ表に対応する)に対応する、1つ以上の種類のデータを選択し得る。従属変数ブロック714はまた同様に、フィルブロック710から受信したデータフレームから、1つ以上の種類のデータを選択するために使用され得る。2つのブロック712及び714が示されているが、任意の数の類似するブロックを使用することができる。その後、分析ブロック716が、独立変数ブロック712及び従属変数ブロック714によって選択されたデータを、示されるように、ブロック間のコネクタによって受信し得る。分析ブロック716は、この分析ブロックがそのために構成されている、任意の種類の分析を実施し得る。例えば、分析ブロック716は、独立変数ブロック712のデータの、従属変数ブロック714のデータに対する影響を判定するために、部分最小二乗(PLS)分析を実施し得る。探査ブロック718は、クエリブロック708ならびにその他のブロック及び接続の適切な構成を試験するために、分析ブロック716及び従属変数ブロック714と接続されてもよい。この探査ブロック718は、ユーザが記憶または閲覧できる、グラフィカル、表形式、またはテキスト形式の出力を生成し得る。前述の説明は、ブロック708~718がある特定のアクションを取るものとして考察しているが、本明細書の他の場所に考察されるように、これらのブロックは代わりに、分析サービス500のウェブサーバプロセス504、実行サービスプロセス506、及び/またはジョブプロセス508にそのようなアクションを行わせてもよいことが理解されるであろう。

【0325】

標準化クエリ709の使用について更に実証するために、図6Cは、例示的標準化クエリフォーマットを活用する例示的クエリを例証する。この例示的クエリは、クロスプラットフォームな適合性を最大化するために、JSONに準拠するフォーマットを活用するものとして呈示されているが、任意の他のフォーマットが使用されてもよい(例えば、XMLベースのフォーマット、CSVベースのフォーマット等)。この例示的クエリは、クエ

10

20

30

40

50

りの名称で始まり、データフレームを返すのに使用されるフォーマットの指標 (JSONとして特定される)、クエリフォーマットのバージョンの指標が後に続く。「timeSelector」アレイは、データが返される時間の範囲を示す複数の時間枠を、「startTime」及び「endTime」オブジェクトで示す。例示的クエリにおいては、開始時間及び終了時間は、文字「T」で分けられている、Coordinated Universal Time (協定世界時)の日付データ及び時刻の観点から特定される。例示的クエリによって示されるように、時間枠の各々は、データが照会されていない除外期間によって分けられる場合があり、この除外期間は、時間枠のうちの1つの終了時間と、別の時間枠の開始時間との間の期間に対応する。「timeSelector」アレイに続いて、「sampleRateSecs」オブジェクトが、秒単位でサンプリングレートを特定し、これは、単位時間あたりにデータ点がいくつ獲得されるかを特定する。例示的クエリの最終部分は、データソース702から獲得されるべきデータ(すなわち、獲得されるべきプロセス変数)のパラメータを示す「columns」アレイである。この例示的クエリは、返される4種類または4列のデータを特定している。4列の各々は、データソース及び/またはデータソースのプロセス変数を特定する「tag」オブジェクトによって特定される(すなわち、「FT630B/DENS.CV」、「PT615/WIRED_PSIA.CV」、「TT6079/INPUT_1.CV」、及び「630.molefrac.c5」)。4列の各々は、データフレーム内の返されたデータを方式する「alias」オブジェクト、データのデータタイプ(例えば、浮動小数点値、倍長整数値、テキスト文字列等)を特定する「dataType」オブジェクト、「renderType」オブジェクト(例えば、値、総数等)、及び返されるデータのフォーマットを特定する「format」オブジェクト(すなわち、「0.###」は、小数点以下3桁の浮動小数点値からなるフォーマットを示す)

【0326】

これらの列と関連付けられる「tag」オブジェクトは、そこからデータを獲得するデータソース702を特定し得る。あるいは、例示的クエリがデータソース702を明示的に特定しない場合は、そのようなデータソース702a~702f(複数可)は、ブロック708の他のプロパティによって特定され得る。標準化クエリフォーマットを活用する類似のクエリにおいては、データソース702(複数可)が、オブジェクトによって、クエリにおいて明示的に示され得る。そのような指標は、ソースオブジェクトへのパス、またはデータソースの種類(例えば、MongoDB、CouchDB、SQL等)を示す値を含み得る。データソース702の指標は更に、適切な場合、関連付けられるデータプロバイダ704を示し得る。あるいは、データソースマネージャ532が、データプロバイダ704の指標、またはデータソース702によって活用される構造もしくはフォーマットを提供し得る。一部の実施形態においては、ソースの指標は更に、獲得されるべきデータの各パラメータまたは列について、データソース702の別個の指標を含み得る。それによって、標準化クエリフォーマットを活用するクエリは、複数の別個のデータソース702からデータを獲得できる。

【0327】

上述されるように、標準化クエリ709は、ターゲットとされる各データソース702について実行され得る、データソースに特異的なクエリへと変換される。そのような変換は、クエリ実行サービス706によって実施されてもよく、あるいはターゲットとされる各データプロバイダ704a~704dによって実施されてもよい。図6Dは、クエリ実行サービス706が標準化クエリ709を1つ以上のデータソースに特異的なクエリへと変換する、例示的クエリ方法740を例証しており、図6Eは、データプロバイダ704が標準化クエリ709をソースに特異的なクエリへと変換する、例示的クエリ方法760を例証する。

【0328】

方法740は、標準化クエリフォーマットを活用する標準化クエリの受信から始まり得る(ブロック742)。標準化クエリは、クエリブロック708、クエリ実行サービス7

10

20

30

40

50

06、またはデータプロバイダ704a~704dによって受信され得る。クエリは、上記の特色または要素のうちいずれを含んでもよい。クエリは、ユーザからの直接入力を通じて受信されてもよく、あるいはユーザ入力に基づいて標準化クエリを生成及び/または記憶し得る、プログラムブロックまたはルーチンから受信されてもよい。標準化クエリは、JavaScript Object Notation (JSON) に準拠するファイルまたはフォーマットで構築されてもよく、あるいは任意の他の好都合な言語または構文を用いて構築されてもよい。標準化クエリは、一部のデータソース702a~702fによって実行可能である構文を活用してデータを獲得してもよく (SQL等)、あるいは標準化クエリは、データソース702a~702fによって直接実行可能ではない構文を活用してデータを獲得してもよい (図6Cに例証される例示的クエリ等)。標準化クエリは、獲得されるべきデータ (例えば、データの時間枠及びパラメータ)、データが獲得される元のデータソース702 (例えば、データソース702a~702fもしくはデータプロバイダ704a~704d)、ならびに/またはデータが提供されるフォーマットを特定し得る。標準化クエリ709がクエリブロック708で受信される場合、この標準化クエリは、クエリを実行するというオプションのユーザ選択、または分析サービス500内の別のオブジェクト、ルーチン、ブロック、プロセス、サービス、もしくは機能による、クエリデータについての要求等の、トリガする事象が発生した際、クエリ実行サービス706に送られ得る。

10

【0329】

標準化クエリ709の受信後、クエリ実行サービス706 (またはデータプロバイダ704) は、データが獲得される元の1つ以上のデータソース702を決定し得る (ブロック744)。一部の実施形態においては、この決定は、1つ以上のデータソース702の指標の受信または特定を伴い得る。この指標は、各データ特性 (例えば、プロセス変数または列) にたいして特異的であってもよく、あるいは全体のクエリに対して一般的であってもよい。そのような指標 (複数可) は、標準化クエリ709内にオブジェクトまたはタグとして含まれてもよく、あるいはクエリブロック708の追加的なプロパティによって示されてもよい。例えば、データ列またはデータの種類についてのデータソース702の指標は、データ列またはデータの種別を特定するオブジェクトまたはタグに対してプリペンドされ得る。あるいは、クエリブロック708またはデータプロバイダ704は、1つ以上のデータソース702と別様に関連付けられてもよい。この関連付けは、1つ以上のデータソース702の指標を、標準化クエリ709から別個に受信することによって果たされ得る。例えば、クエリブロック708が、データソース702 (またはデータプロバイダ704) の選択を受信することで作成される場合、クエリブロック708は、データソースプロパティを介して、1つ以上のデータソース702 (または1つ以上のデータプロバイダ704) と関連付けられ得る。同様に、データプロバイダ704a~704dが、内在的に、あるいは構成中に作成される関連付けを通じて構成される場合、データプロバイダ704a~704dは、1つ以上のデータソース702a~702dと関連付けられ得る。

20

30

【0330】

1つ以上のデータソース及び標準化クエリ709に関する情報を用いて、1つ以上のデータソース702からデータを得るために、データソースに特異的なクエリが生成される (ブロック746)。データソースに特異的なクエリは、クエリ実行サービス706によって生成され、ブロック744において特定されたデータソースに基づいて、照会されるべき各データソース702a~702fに送信され得る (直接、あるいはデータソースプロバイダ704a~704dを介して)。あるいは、クエリ実行サービス706は、1つ以上のデータプロバイダ704a~704dの各々によってデータソースに特異的なクエリへと変換される標準化クエリ709を送信してもよく、これらのデータプロバイダは、ブロック744において特定されたデータソースに基づいて、データが獲得される元のデータソース702a~702dと関連付けられている。例えば、クエリ実行サービス706は、1つ以上のジョブプロセス508に、データプロバイダ704として標準化クエ

40

50

リ709を変換させ得る。クエリ実行サービス706またはデータプロバイダ704のいずれが標準化クエリ709からデータソースに特異的なクエリを生成するかに関わらず、データソースに特異的なクエリは、データソース702に記憶されているデータを獲得するために、データプロバイダ704またはデータソース702によって実行可能な、データソースに特異的なフォーマットを活用しなければならない。データソースに特異的なクエリフォーマットは、電子データソースに記憶されているデータを照会するか、あるいは別様にそれにアクセスするための、SQL、MongoDB、CouchDB等の、任意の既知の、または今後開発されるフォーマットまたは構文であり得る。

【0331】

データソースに特異的なクエリの生成は、標準化クエリ709中のクエリパラメータの特定を含み得る。クエリパラメータとしては、上に考察した例示的標準化クエリ中の「timeSelector」アレイによって特定される時間枠、「sampleRateSecs」オブジェクトによって特定されるサンプルレート、及び「columns」アレイ（特に「tag」オブジェクト）によって特定されるデータパラメータ等の、データの時間枠または他の特性と関連付けられるパラメータが挙げられ得る。これらのクエリパラメータは、種類もしくは測定値、測定デバイスの種類、または具体的な測定デバイス（フィールドデバイス15~22及び40~46）と関連付けられるプロセス変数の指標を含み得る。1つ以上の時間枠が、照会される各データソース702について特定され得、異なるデータソース、または異なるパラメータ（例えば、異なる列、異なるプロセス変数等）を有する、あるデータソース内のデータは、それらと関連付けられる異なる時間枠を有し得る。特定されたクエリパラメータは、クエリ実行サービス706またはデータプロバイダ704によって標準化クエリ709から抽出され、データソースに特異的なクエリの態様へと変換され得る。したがって、生成されたデータソースに特異的なクエリは、標準化クエリ709において特定された実質的パラメータを表すクエリ言語を含むが、そのような実質的パラメータは、データソースに特異的なクエリフォーマットのデータソースに特異的なクエリに含まれる。

【0332】

1つ以上の時間枠及びサンプリングレートが特定されている実施形態においては、データソースに特異的なクエリは、各時間枠によって特定される期間内の、特定された時間における複数のデータ点と関連付けられるデータを提供するように、生成され得る。各データ点は、データ点と関連付けられる特定された時間を示すタイムスタンプを有し得る。このタイムスタンプは、時間枠と関連付けられる開始時間における、あるいはその後の、かつ時間枠と関連付けられる終了時間において、あるいはその前に発生する、サンプリングレートの期間の整数倍である。複数の時間枠が、図6Cにおいて示されるように、標準化クエリによって特定され得る。そのような事例においては、データソースに特異的なクエリは、複数の時間枠の各々の間において、サンプリングレート（複数可）に対応するタイムスタンプを有するデータ点におけるデータを獲得するように、生成され得る。一部のそのような事例においては、複数の時間枠の全てのうちの一部について、別個のデータソースに特異的なクエリが生成され得る。1つ以上の時間枠がサンプリングレートを伴わずに特定される場合、複数のデータ点がタイムスタンプを伴って獲得され得るが、これらのタイムスタンプは、その時間枠内の任意の値を取り得る。したがって、時間枠は、各データ点が測定された時間を示すことになる（あるいは、根本的なプロセス値が測定された時間）。

【0333】

次いで、クエリ実行サービス706またはデータプロバイダ704は、1つ以上のデータソースに特異的なクエリを実行させて、特定されたデータソース702を照会する（ブロック748）。これには、データソースに特異的なクエリを、クエリ実行サービス706から、1つ以上のデータプロバイダ707a~707d及び/または1つ以上のデータソース702a~702fに送信することを含み得る。同様に、一部の実施形態においては、データプロバイダ704は、データソースに特異的なクエリをデータソースに送信し

10

20

30

40

50

て、要求されるデータを獲得し得る。更なる実施形態においては、データソースに特異的なクエリは、受信側のデータソース702の実行エンジンによって実行されてもよい。他の実施形態においては、データプロバイダ704が、データソース702内に記憶されているデータにアクセス、かつそれを操作することによって、データソースに特異的なクエリのうちの1つ以上を実行し得る。1つ以上のデータソースに特異的なクエリを実行する際に1つ以上のデータソースから返されるデータは、その後、データ受信エンティティに送信され、あるいはそれによって受信され得る。このデータ受信エンティティとしては、データプロバイダ704、クエリ実行サービス706、またはクエリブロック708が挙げられ得る。したがって、データ受信エンティティは、1つ以上のデータソース702から、要求されるデータを獲得する(ブロック750)。

10

【0334】

サンプリングレートが特定されている実施形態においては、データを獲得することとは、時間枠及びサンプリングレートによって示されるサンプル点の各々と調和するタイムスタンプを伴うデータ点をデータが含むということを確認することを含み得る。これは各時間枠及びサンプリングレートについてなされ得る。上述されるように、これらのサンプル点は、時間枠の開始時間における、あるいはその後の、かつ時間枠の終了時間において、あるいはその前に発生する、サンプリングレートの期間の整数倍に対応する。データソース702に記憶されるデータ点が、サンプル点と関連付けられる時間のうちの1つ以上に關するデータ点を含まない場合、サンプル点に対応するタイムスタンプを有する追加的なデータ点が追加されてもよい。これらの追加的なデータ点は、タイムスタンプに対して時間的に最も直前である、データソース702内のデータエントリの値と関連付けられる代入値であってもよい。例えば、データソース702は、時間7:01:5500(値V1を有する)、7:02:0500(値V2を有する)、及び7:02:5500(値V3を有する)についてのデータ点を含む場合があるが、標準化クエリにおいて示される時間枠は、7:02:0000(開始時間)から7:03:0000(終了時間)である場合があり、サンプリングレートの期間が0:00:5000である場合がある(0.5秒毎である、1つのサンプルのサンプリングレートに対応する)。そのような例においては、獲得されるデータ点は、7:02:0000、7:02:5000、及び7:03:0000のタイムスタンプを有することになり、これらはそれぞれ値V1、V2、及びV3を有する。したがって、タイムスタンプ、7:02:5000を有するデータ点については、このタイムスタンプ後の時間では別の値(時間7:02:5500のV3)の方がより近いが、最も直前の値V2(7:02:0500に測定)が使用される。加えて、所望のサンプル時間の間の時間において発生する過剰なデータ点は、得られたデータから除去または消去してもよい。したがって、獲得されたデータ中の各データ点(そのような調整または整合後)は、標準化クエリによって示される各時間枠内の開始時間における、あるいはその後の、サンプリングレートの期間の整数倍と関連付けられるタイムスタンプを有することになる。獲得されたデータ内のデータ点を整合する他の手段も、同様に使用できる。

20

30

【0335】

複数のデータソース702が同一の標準化クエリにおいて示される場合、データソース702の各々が、異なるタイムスタンプを有するデータ点を有する場合があり、またそのようなタイムスタンプと関連付けられる測定値が、異なるサンプリングレートで発生した可能性がある。例えば、第1のデータソース702a及び第2のデータソース702dが標準化クエリによって示される場合がある。データソース702aは、第1のサンプリングレート(例えば、毎秒1回の測定)を有するデータ点を記憶する第1のデータソースである可能性があり、データソース702dは、第2のサンプリングレート(例えば、毎秒4回の測定)を有するデータ点を記憶する第2のデータソースである可能性がある。獲得されたデータは、標準化クエリ内で特定されるパラメータに従って、上に考察したように、標準化クエリによって特定されるサンプリングレートの期間の整数倍である時間におけるデータ点を追加または除去することによって、整合できる。

40

【0336】

50

データ受信エンティティが1つ以上のデータソース702からデータを獲得するとき、データ受信エンティティは更に、標準化クエリ709と関連付けられるフォーマットパラメータまたは命令に従ってデータをフォーマットし得る(ブロック752)。そのようなフォーマットパラメータまたは命令は、標準化クエリ709に含まれてもよく、標準化クエリフォーマットと関連付けられてもよく、あるいはクエリブロック708の別個のプロパティにおいて特定されてもよい。一部の実施形態においては、これは、獲得したデータのフォーマットを、所望のフォーマットと調和するように調整することで、獲得したデータからデータフレームを更に生成することを含み得る。複数のデータソース702が照会されている場合、データフレームの生成は更に、複数のデータソース702の各々から獲得したデータを組み合わせ、集合データフレームを生成することを含み得る。例えば、第1のデータセットが第1のデータソース702aから照会され、第2のデータセットが第2のデータソース702dから照会される場合、第1及び第2のデータセットを組み合わせる集合データフレームが生成され得る。

10

【0337】

方法760は、データ要求エンティティから標準化クエリ709を受信することで始まり得る(ブロック762)。クエリブロック708またはクエリ実行サービス706等のデータ要求エンティティは、標準化クエリ709をデータプロバイダ704に送信することによって、1つ以上のデータソース702からのデータを要求し得る。データプロバイダ704は、1つ以上のデータソース702と関連付けられてもよく、あるいはネットワーク接続を介して様々なデータソース702からデータを獲得するように構成されてもよい。あるいは、データ要求エンティティは、データ分析システム100内の別のデータブロック、ソフトウェアルーチン、プロセス、またはサービスから標準化クエリ709を受信し得る。加えて、データ受信エンティティ(これは同様に、クエリブロック708、クエリ実行サービス706、ジョブプロセス508、データサービス520、アプリケーション514、プログラムメモリ、またはプロセスプラント5と関連付けられるデータを更に分析、呈示、もしくは記憶し得る任意の他のサービス、ルーチン、プロセス、もしくはデバイスであり得る)は、標準化クエリ709によって、あるいは別様に示され得る。そのようなデータ受信エンティティは、一部の事例においてはデータ要求エンティティと同じエンティティであってもよく、あるいはデータ要求エンティティとは別個であってもよい。明瞭性のために、例示的方法760についての以下の考察は、データプロバイダ704aが、データ要求エンティティ及びデータ受信エンティティとしてのクエリ実行サービス706から、標準化クエリ709を受信し、標準化クエリ709はデータソース702aからのデータを要求するものと仮定する。これは、本方法の重要な特色をより良好に例証するためになされるものであり、本開示の範囲を限定することを意図するものではない。当業者であれば、本明細書に考察される方法を軽度かつ普通に適応させるだけで、多数の代替的構成が容易に作成できることを理解するであろう。

20

30

【0338】

標準化クエリがデータプロバイダ704aにおいて受信されると、データプロバイダ704aは、標準化クエリ709からクエリパラメータを抽出する(ブロック764)。クエリパラメータには、時間と関連付けられるパラメータ(例えば、データが獲得される時間枠、サンプリングレート等)、データタイプもしくは特性と関連付けられるパラメータ(例えば、プロセス変数、表内の列、測定値、測定値からの計算値等)、またはデータの獲得元であるデータソース702aと関連付けられるパラメータ(例えば、データベースの指標、そこへのパス、もしくはその中の表)が含まれ得る。クエリパラメータの抽出は、標準化クエリ709内のオブジェクト、アレイ、または要素に基づいて1つ以上のパラメータを決定することを含み得る。一部の実施形態においては、データプロバイダ704aは更に、データがどのようにクエリ実行サービス706に返されるかを示すパラメータ(すなわち、要求されたデータをデータ要求エンティティに提供する際に使用するフォーマット、構造、タイミング、またはプロトコル)を抽出し得る。データプロバイダ704aは、抽出したクエリパラメータを、1つ以上のデータソースに特異的なクエリを生成す

40

50

る際及び/または獲得したデータをフォーマットしてデータ要求エンティティに提供する際に使用するため、揮発性メモリまたは不揮発性メモリに記憶し得る。

【0339】

次いで、データプロバイダ704aは、抽出したクエリパラメータに基づいて、データソースに特異的なクエリを生成し得(ブロック766)、このデータソースに特異的なクエリは、データソース702aと関連付けられるデータソースに特異的なクエリフォーマットを活用する。例えば、データソース702aは、MongoDBデータ構造またはフォーマットを活用する非リレーショナルデータベースであり得、この場合、データプロバイダ704aは、MongoDBのクエリ構文を活用するデータソースに特異的なクエリを生成して、標準化クエリ709から抽出されたクエリパラメータによって示されるデータを獲得する。代替的例としては、データソース702aは、MySQLを活用するリレーショナルデータベースであり得、この場合、データプロバイダ704aは、SQLクエリ構文を活用するデータソースに特異的なクエリを生成して、標準化クエリ709から抽出されたクエリパラメータによって示されるデータを獲得する。データソースに特異的なクエリを生成するために、データプロバイダ704aは、標準化クエリフォーマットと、データソースに特異的なクエリフォーマットとの間でマッピングを適用し得る。そのようなマッピングには、標準化クエリ709において表現されるクエリパラメータを、データソースに特異的なクエリフォーマットで実質的に等価のパラメータへと変換する、構文またはフォーマットへの調整が含まれ得る。一部の実施形態においては、データプロバイダ704aが、標準化クエリによって示されるクエリパラメータを、データソースに特異的なクエリのパラメータまたは要素に直接マッピングするように、クエリパラメータの抽出及びデータソースに特異的なクエリの生成は組み合わせられ得る。上に考察したように、データソースに特異的なクエリは、標準化クエリ709によって示される各時間枠内の開始時間における、あるいはその後の、サンプリングレートの期間の整数倍に対応するタイムスタンプを有するデータ点を返すように、生成され得る。

【0340】

データソースに特異的なクエリがいったん生成されると、データプロバイダ704aは、データソースに特異的なクエリを実行して、要求されたデータをデータソース702aから獲得し得る(ブロック768)。一部の実施形態においては、データプロバイダ704aは、データソースに特異的なクエリを実行するという要求をデータソース702aに送信してもよく、データソース702aは、データソースに特異的なクエリを実行し、結果として得られたデータをデータプロバイダ704aに返してもよい。あるいは、データプロバイダ704aは、データソース702a用のインターフェイスまたは実行エンジンとして働いてもよく、この場合、データプロバイダ704aは、データソースに特異的なクエリフォーマットの規則に従って、データソースに特異的なクエリを実行して、データソース702aに記憶されているデータにアクセス、それを分析、及び選択してもよい。データソースに特異的なクエリが、データプロバイダ704aによって実行されるか、あるいはデータソース702aによって実行されるかに関わらず、データソースに特異的なクエリの実行から結果として得られるデータは、データプロバイダ704aによって獲得される。

【0341】

一部の実施形態においては、データプロバイダ704aは、標準化クエリ709に基づいて、クエリの実行によって獲得されたデータをフォーマットし得る(ブロック770)。したがって、データプロバイダ704aは獲得されたデータを処理して、標準化クエリ709で表現されるデータについての要件に対応するように、獲得されたデータをフォーマットし得る。これには、クエリ実行サービス706が容易に使用できるフォーマットの獲得されたデータを含む獲得されたデータからデータフレームを生成することが含まれ得、獲得されたデータに、データフォーマット規則及びデータエイリアスを適用することが含まれる。例えば、標準化クエリ709は、データが、JSONファイルを用いるドキュメントで提供されるべきであることを特定してもよい。同様に、標準化クエリは、データ

10

20

30

40

50

が、表のフォーマットでデータ要求エンティティに返されるべきであること、またはデータ値が、小数点以下3桁の浮動小数点数としてフォーマットされるべきであることを特定してもよい。それらのフォーマット要件は、標準化クエリ709において特定されてもよいが、フォーマット要件の一部または全ては、標準化クエリフォーマット（またはそのバージョン）によって特定されてもよい。この様式においては、標準化クエリフォーマット（またはそのバージョン）を採用する標準化クエリは、常に、データを一貫したフォーマットで返すことになる。上に考察したように、獲得したデータのフォーマットは、標準化クエリ709によって示される各時間枠内の開始時間における、あるいはその後の、サンプリングレートの期間の整数倍に対応するタイムスタンプを有するように、データ点の時間を調整または整合することを含み得る。これには更に、適切なタイムスタンプを伴うデータ点の追加、または所望のサンプル時間の間の時間に発生する過剰なデータ点の除去が含まれ得る。

10

【0342】

獲得したデータがフォーマットされると、データプロバイダ704aは、フォーマットしたデータをデータ受信エンティティに提供する（ブロック772）。データプロバイダ704aは、要求されたフォーマットまたは任意の既知のフォーマットのデータフレームを、クエリ実行サービス706に、更なる分析のために送信し得る。更なる実施形態においては、獲得されたデータをフォーマットすることの一部または全ては、データプロバイダ704aが、データソースに特異的なクエリの実行から獲得された未フォーマットまたは部分的にフォーマットされたデータを、クエリ実行サービス706に送信した後、クエリ実行サービス706によって代わりに実施されてもよい。データが、データ要求エンティティによって受信され、フォーマットされたとき、フォーマットされたデータは、本明細書の他の場所に考察されるようにプロセスプラント制御または分析において使用できる。一部の実施形態においては、クエリ実行サービス706は更に、クエリブロック708またはジョブプロセス508等の別のデータ受信エンティティにデータを提供してもよい。

20

【0343】

概して、データソース702a~fのいずれかの部分または全ての部分、データソースプロバイダ704a~d、及びクエリブロック708、ならびに可変的にフォーマットされるデータソース702a~fから獲得するために活用されるクエリ言語は、図1、2、3、4A~4Q、及び/または5A~5Gのいずれかの部分または全ての部分と併せて、かつ/あるいは図1、2、3、4A~4Q、及び/または5A~5Gにそれぞれ対応する本開示の節に記載される任意の数の特色及び/または技法と併せて動作し得る。

30

【0344】

早期警戒故障検出のための周波数分析の分析技法

新規データ分析技法または機能（例えば、分散型工業プロセスパフォーマンス監視/分析システムまたはDAS100によって提供され得る）は、プロセスプラント5等のプロセス制御システムまたはプラントにおける早期警戒故障検出のための、周波数分析の分析技法である。該新規周波数分析の分析技法または機能は、本明細書においては、「ローリング高速フーリエ変換」または「ローリングFFT」と呼ばれ、本明細書に記載されるシステム、アーキテクチャ、方法、及び技法のうちのいずれかと併せて活用できる。例えば、ローリングFFT用のデータブロック定義が、データブロック定義ライブラリ252内に記憶され、また、オフラインで実行され、かつ/あるいはオンラインで実行されるデータモジュールにおいて活用できるようにユーザに対して利用可能にされ得る。追加的または代替的に、1つ以上の分散型データエンジン102xが、1つ以上のDDE102xにおいて獲得されたデータに対してローリングFFTを実行してもよい。しかし、ローリングFFTは、本明細書に記載されるシステム、アーキテクチャ、方法、及び技法と併せて使用することのみに限定されず、時系列データを生成する任意のシステム、アーキテクチャ、方法、及び/または技法と併せて活用してもよい。下に示されるように、ローリングFFT分析技法または機能は、記述的分析及び予測的分析両方の組み合わせである。

40

50

【 0 3 4 5 】

概して、プロセス制御システムまたはプラントにおいては、未来のプロセスプラントの挙動についての先行指標を提供するプロセスデータを、好ましくは予防的または軽減的アクションを起こすことを可能にする時間の枠内において発見できれば、異常、故障、パフォーマンスの低下、及び/または所望されないか、もしくは望ましくない状態を予防できる(あるいは、それらの影響を最小化できる)。そのようなプロセスデータには、例えば、圧力、温度、及びプラントを通過して移動する材料の流速の測定値データ、ならびに設備の部分に関する類似の情報が含まれ得る。そのようなプロセスデータとしてはまた、例えば、プロセスフローストリームの化学組成、及び設備のオン/オフ状態が挙げられ得る。概して、ローリングFFTを適用できるプロセスデータには、プロセスプラント内のプロセスを制御する結果として生成され、例えばサンプリング、データストリームの受信、データベース、データファイル、データソース(例えば、Seqシステム)、またはヒストリアンからの読み取り、照会等によって、任意の好適な様式で獲得される、任意のオンライン及びオフライン時系列データが含まれ得る。

10

【 0 3 4 6 】

プロセスプラントにおける先行インジケータを発見する1つの技法としては、プロセスデータの挙動を経時的に分析することが挙げられる。この挙動は多くの理由で変化し得るが、一部のシナリオにおいては、この変化は、異常、故障、パフォーマンスの低下、及び/または他の状態につながるプロセスの不調と関連付けられ得、したがって、そのような状態の先行インジケータとみなされ得る。本明細書に記載されるローリングFFT技法は、先行インジケータに対応する時系列データの第1のセットを周波数領域に変換し、続いてその周波数領域データに基づいて時系列データの第2のセットを生成し、その後これを監視及び使用して、プロセスプラントの異常、故障、パフォーマンスの低下、及び/または状態を予測し得る。

20

【 0 3 4 7 】

従来、プロセスプラントからのプロセスデータは、周波数分析技法に対する入力として使用するために、収集、受信、または別様に獲得されている場合がある。データは、温度、フロー、圧力、組成、及び/またはプロセスを制御するためのプロセスプラントの運転の結果として生成される他の連続信号等、プロセスプラントにおいて見出される既存のプロセス信号に関する任意のデータであり得る。従来、FFTは、通常2の累乗(例えば、 $2^{10} = 1024$)に基づく固定窓(例えば、データ点の特定の数)を用いて、獲得されたプロセスデータ中の重要な周波数の振幅を特定するために、そのプロセスデータに対して実施される。最新の計算方法は、このデータの窓の長さをユーザが定義することを可能にするが、この長さは多くの場合、利用可能なコンピュータのメモリの量によって制限されてしまう。FFTにおけるサンプルの数及びサンプリング周波数はまた、目的の最速周波数当たり少なくとも2つのサンプルを有するというナイキスト要件を満たさねばならない。更に、従来のFFTは、所望のデータ信号の数サイクルの周期的挙動に対して作用すべきである。

30

【 0 3 4 8 】

しかし、プロセスデータに適用されるFFTの多くの従来型の適用においては、目的の信号は経時的に変化しないと仮定される。しかしながら、ローリングFFTはこの仮定によって制約されない。実際、「ローリング」タイプのFFTは、有利なことに、信号に対する経時的な変化を捕捉して、信号に対するそれらの経時的な変化がいつ起こるかを特定できる。具体的には、ローリングFFTは、目的の信号または変数によって生成されるデータ(プロセスプラントの結果として生成された、測定値、感知された値、または他の信号)の窓に対してFFTを実施すること、及びその窓について周波数の振幅(例えば、ピーク振幅)を記録することを含む。次いで、この窓を、時間で1つのサンプル分前向きに動かし、FFTを再度実施して、結果を記録または記憶する。これを時系列データの最後まで継続する。FFTは、データセット中の全てのサンプル時間に対して実施されるため(一部の場

40

50

ルは除く)、1つ以上の目的の周波数の振幅(例えば、ピーク振幅)を含む、1つ以上の新しい時系列データセットが作成または生成される。目的の各振動数は、目的の振動数の振幅(例えば、ピーク振幅)に対応するそれぞれの時系列データを生成する、プロセスプラントのそれぞれの新しいプロセス変数に対応し得る。各新規プロセス変数によって生成される時系列データは、プロセスプラント内の、可能性のある異常、故障、または他の状態を予測するために、記憶、監視、及び/または分析され得る。

【0349】

したがって、目的の信号または変数について、予測される、所望されないプロセスプラント状態に対応する1つ以上の新しいプロセス変数が、プロセスプラント内で作成、生成、及び活用され得る。目的の信号または変数は、個別の信号または測定点であってもよく、あるいは個別の信号/点の信号または測定値の組み合わせ(例えば、和または他の組み合わせ)であってもよい。目的の変数はユーザによって定義されてもよく、あるいはデータブロック、データモジュール、及び/またはデータ分析機能によって自動的に決定されてもよい。

10

【0350】

上に考察したように、新しい時系列データセットのセットを決定するプロセスは、目的の信号または変数を時間領域から周波数領域に変換すること、周波数領域データを獲得すること、及び獲得された周波数領域データを転換して、新しいプロセス変数に対応する時系列データを決定することを伴う。時間領域に再度変換または転換することは重要である。これは、新しい時系列データセットを、元のプロセスデータ信号及び/または他の時系列プロセスデータと共に閲覧することを可能にするためである。したがって、新しい時系列データは、元の目的の信号もしくは変数によって生成されるデータ、及び/または他のプロセスデータに対して使用可能である分析技法(例えば、統計的分析、PCA、標準偏差等)と同じものを用いて探查することができ、一部の状況においては、元のプロセスデータ及び/または他のプロセスデータと併せて分析することができる。例えば、新しい時系列データ及び元の/他のプロセスデータの両方が、Transformersデータブロック521に入力として提供されてもよい。

20

【0351】

ローリングFFT分析技法及びその利点を例証するために、ローリングFFTが、精製所または化学プラント等のプロセスプラントにおける潜在的なフレア事象を検出する問題に適用される、例示的シナリオについて考慮されたい。概して、一部のプロセスプラントは、個別のプロセスユニットから過剰な蒸気を収集し、蒸気を大気中に放出する前に、収集した過剰な蒸気を燃焼させて毒性化学物質を除去するフレアリングシステムを有する。この過剰な蒸気またはガスの燃焼を、概して、「バーンオフ」または「フレア事象」と呼ばれる。一部のシナリオにおいては、バーンオフされる代わりに、フレアガスは圧縮され、供給材料または燃料ガスとして再利用される。しかし、フレアガス圧縮機の容量を超過するとフレア事象が発生するため、過剰な材料はフレアバーナを通じて放出される。典型的には、フレア事象の数及び/または頻度は、環境規制及び/または他の種類の規制によって管理されている。一部の計画的なフレア事象は許容されており、かつ必要であるが、プロセスプラントの提供者または企業は、計画外のフレア事象が余りにも頻繁に発生する場合、科料に処される可能性がある。したがって、プロセスプラントの提供者またはオペレータにとっては、現在の運転状態に基づいて、計画外のフレア事象が発生する、または発生する可能性が高いことを予測でき、また十分な先行時間を伴ってこの予測を獲得し、その間にオペレータが軽減的アクションを取って、計画外のフレア事象が起こるのを防止できることが望ましい。

30

40

【0352】

これは、現代の精製所及び化学プラントは多数の相互接続されたユニット(例えば、数十または数百にすら及ぶ相互接続されたユニット)を伴う複合システムであり、各ユニットそれ自体が大きなプロセスプラントとみなされ得るため、解決が困難な問題である。通常、これらのユニットは、一般的なフレアシステムに接続されている。これらのユニット

50

のうちのいずれもが、フレアシステムが取り扱うべき蒸気の潜在的なソースとなり得るため、どのユニット（複数可）がフレア状態に近いかを監視するのは困難であり得る。更に、いったんフレアが発生すると、どのユニットが原因であるかは直ちには明らかとならない。

【0353】

この状況に対処するために、本明細書に記載されるローリングFFT技法を使用することができる。図7Aは、ローリングFFTが適用された精製所によって生成された例示のプロセスデータを示す。例示のプロセスデータには、特定の時間間隔中に、精製所のフレアシステム内のユニットから獲得される、プロセスフロー測定値または信号1100が含まれる。確認できるように、プロセスフロー信号1100は本来周期的であり、その周期はほぼ1日である（例えば、その周期性が、毎日の加熱冷却サイクルに対応する場合があるため）。加えて、例示のプロセスデータには、同一の特定の時間間隔にわたる、フレアシステムのフレア圧力を示す、別の信号1102が含まれる。フレア圧力はフレアシステム内の集積的蒸気の測定値であり、この測定値は、例えば、過剰なガス及び蒸気を格納する圧縮機または他の容器のセンサから獲得され得る。図7Aにおいては、信号データ1100及び1102は、それらの経時的な挙動及び関係性が容易に視覚化されるように、時間で整合され、スケール変更されていることに留意されたい。ある実施形態においては、1つ以上のデータブロック、データモジュール、及び/または1つ以上のDDE102xが、信号データ1100及び1102を受信し、信号1100及び1102を図7Aに示される時間整合された様式で表示できるように、2つの信号1100及び1102の時間整合を実施する。

【0354】

図7Bは、異なる時間間隔についての同じ信号1100及び1102を示し、この期間中に、信号1102においてフレア事象1104が発生した（例えば、信号1102に対応するフレア圧力が高まり、圧力限界を超え、したがってフレア事象1104を引き起こした）。図7Bを見ると、プロセスフロー信号1100の挙動が、フレア事象1104の前に変化していることが明らかである。例えば、プロセスフロー信号1100の周期的挙動は、フレア事象1104の発生の2日から3日前に変化している。フレア事象1104に関する先行インジケータとしての、プロセスフロー信号1100における変化の特定は、例えば、PCA、相互相関関係、PLS回帰等のデータ分析技法を用いることで決定され得る。ローリングFFTを活用することによって、この変化/先行インジケータを、捕捉、特定、または定義することができ、特定された変化/先行インジケータに対応する新しい信号またはプロセス変数を、生成、定義、または作成することができ、新しいプロセス変数の時系列出力を決定することができる。したがって、先行インジケータの存在に関するローリングFFTによって決定される、新しいプロセス変数の時系列データを監視することによって、切迫したフレアを判定することができ、予防的及び/または軽減的アクションを取ることができるように、対応する警告をユーザ（例えば、エンジニア、プロセスオペレータ等）に呈示することができる。ある実施形態においては、新しく定義されたプロセス変数は、（例えば、タグの割り当て、及び/または新しいプロセス変数をプロセス制御データベースに別様に認識させることによって）精製所のプロセス制御システム内に組み込まれてもよく、従来型のプロセス制御監視設備及び技法を用いて監視してもよい。したがって、新しいプロセス制御変数に対応するステータス及び他の情報は、限度、アラーム、動向チャート等を伴うプロセス情報の任意の他の項目と同様に、ユーザに呈示することができる。

【0355】

ある実施形態においては、フレア事象を予測するために活用されるプロセスフロー信号は、個別のフローまたは測定値の組み合わせ（例えば、合計、加重平均、または他の好適な組み合わせ）である。図7Cは、ローリングFFT分析を適用することができるプロセス信号が、個別のプロセスフロー信号1106の合計に対する変化に対応する実施形態を示す。合計シグナル1106は、ある例においては、個別のフロー信号の測定値を足し合

10

20

30

40

50

わせることで作成できる。合計 1106 または信号の他の所望される組み合わせの生成は、作成及び分析されるべき新しい信号の数を低減し、これはひいては全体的な分析を簡略化させ得る。しかしながら、単一のプロセス信号を使用するか、あるいは信号の何らかの組み合わせを使用するかについて、制約は存在しない。図 7C はまた、表示される時間間隔中においてフレア事象 1108 及び 1110 を含む、フレア圧力信号 1102 を同時に示す。相互関係が容易に視覚化されるように、信号 1106 及び 1102 は時間整合されていることに留意されたい。プロセスフローの合計 1106 の挙動における変化は、フレア事象 1108 の直前において確認できるが、フレア事象 1110 の場合は、合計されたプロセスフロー 1106 における変化は、フレア事象 1110 の後に発生している。合計されたプロセスフロー 1106 の挙動が、例えばより長い時間間隔にまたがる場合、するか否か（また、する場合はどのようにするか）を決定するためには、更なる分析機能等を実施することによる、フロー 1102 とフロー 1106 との間の関係性の更なる分析が必要となり得る。

10

【0356】

図 7D は、ローリング FFT 技法を信号 1106 に適用したことの結果 1112 を示しており、この適用により、プロセスフロー信号の合計 1106 は、元の時間領域から周波数領域へと転換され、次いで再度時間領域へと転換される。特に、図 7D の信号 1112 は、信号 1106 の、具体的な目的の周波数（例えば、第 4 の周波数は 1 日当たり 4 周期に対応する）の振幅（例えば、時系列データ）に対応する。1 日当たり 4 周期に対応する第 4 の周波数が、新しいプロセス変数として定義され、その時系列ピーク振幅値が対応する周波数領域データから捕捉され、フレア圧力信号 1102 と併せて、信号 1112 として時間領域で表示されている。相互関係が容易に視覚化されるように、信号 1112 及び 1102 は時間整合されていることに留意されたい。図 7D に見られるように、信号 1112 の特定のピーク振幅 1113 が、フレア事象 1108 と関連付けられる。しかし、データ 1112 はノイズが混ざっているように見受けられ、偽陽性と呼ばれ得るものを含む（例えば、事象 1115 a、1115 b）。しかしながら、偽陽性は、注意報として使用することができ、ならびに / または回避された「近」フレア事象を表す場合があるため、過度に心配するには当たらない。

20

【0357】

信号データ 1112 を更に処理及びクリーンにするため、更なるデータ分析技法が実施されてもよい。例えば、図 7E においては、PCA が、1 日当たり 4 周期に対応する第 4 の周波数の周波数領域データに対して、その第 1 の主構成要素を決定するために適用されており、第 1 の主構成要素の時系列値が、信号 1118 として示されている。具体的には、図 7E は、フレア圧力信号 1102 のフレア事象 1108 周囲の時間間隔中における、第 1 の主構成要素 1118 の挙動の拡大表示を示している。信号 1118 及び 1102 は、それらの間の相互関係が容易に視覚化されるように、時間整合されていることに留意されたい。図 7E に見られるように、第 1 の主構成要素 1118 の値におけるスパイクが、フレア事象 1108 の前によく発生しており、そ麗江には、第 1 の主構成要素 1118 の値は有意に減少している。このスパイクを捕捉するために、信号 1118 をある期間保持するために、減衰フィルタを使用してもよい。ある実施形態においては、減衰フィルタは、その減衰フィルタのインスタンスが、各アプリケーションについて異なって調整可能であるように、データブロック定義及び対応するデータブロックプロパティによって定義され得る。例えば、減衰フィルタブロックの構成可能なプロパティは、信号の増加速度及び / または減衰速度を定義し得る。一部の実施形態においては、更なる主構成要素を決定し、技法の感度を上昇させるために使用してもよい。

30

40

【0358】

図 7F は、図 7D の別の部分の拡大表示を示す。具体的には、図 7F は、フレア圧力信号 1102 のフレア事象 1110 周囲の時間中における、第 1 の主構成要素 1118 の挙動をより詳細に示している。図 7F においては、フレア事象 1110 に対応する第 1 の主構成要素 1118 の値は、フレア事象 1108 の場合と比べると小さいが、基線と比較す

50

ると、値の有意な差異が依然として存在する。

【0359】

プロセス制御データにおいて先行インジケータを検出するために、ローリングFFT分析を使用する技法は、オフライン分析及びデータモデルを構築する試みにおいて使用できる。しかしながら、新しいプロセス変数（例えば、周波数、主構成要素、及び/または他の目的の高次データ）が特定及び定義されると、ローリングFFT分析は、オンラインプロセスプラントからのリアルデータのストリーミングに対して実施され得る。例えば、ローリングFFTデータブロックのインスタンスをオンラインデータモジュールに組み込み、プロセスプラント内のオンラインソースにバインドできるように、ローリングFFTをデータブロックとして定義し、ブロック定義ライブラリ252に記憶してもよい。したがって、ローリングFFT分析が動作するライブのストリーミングデータは、ユーザまたはプラントオペレータに対し、切迫した状態についてのリアルタイム予測/警告を提供し得る。一部の実施形態においては、ユーザは、他の分析機能ブロック（例えば、PCA、PLS、及び/または他の分析機能ブロック）に対して相互接続されるローリングFFTデータブロックを含むように、組み合わせデータブロックを定義してもよい。加えて、ユーザは、複数のプロセス信号を単一の信号へと組み合わせる（例えば、合計する、加重平均を計算する等）データブロックを定義して、ローリングFFTデータブロックへの入力及び/またはローリングFFTデータブロックを含む組み合わせブロックへの入力としてその単一の信号を活用してもよい。

10

【0360】

したがって、ローリングFFTデータ分析技法は、オフラインプロセスデータ及びオンラインプロセスデータについて活用できる、記述的分析技法の一例である。

20

【0361】

一部の実施形態においては、ローリングFFT分析技法に対応する新しいプロセス変数は、プロセスプラントに組み込まれ得る。例えば、新しいプロセス変数は、（例えば、それぞれのプロセス制御タグを割り当てることによって）定義することができ、プロセスプラントのプロセス制御データベースに記憶することができる。ある実施形態においては、新しいプロセス変数によって生成された時系列データは、プロセスプラントのプロセスの一部を制御するように動作する制御機能または制御ブロックへの入力として働くことができ、あるいはプロセスプラントにおける変化のトリガとして働き得る。

30

【0362】

更に、潜在的なフレア事象の検出は別として、所望されない状態または事象を予防するためにローリングFFTを適用できる他の状況としては、潜在的な圧力安全バルブのリフティング、潜在的な圧縮機のサージ、切迫したポンプの破損、切迫したバルブの破損、蒸留塔における冠水等のフローの不安定性、回転粉碎ミルにおける破損、油井及びガス井生産の不安定性等の警告が挙げられる。一部のシナリオにおいては、ローリングFFTは、プロセスプラント内の1つ以上のエンティティ、またはプロセスプラント全体の所望されないパフォーマンス測度を予防するために適用できる。

【0363】

図7Gは、プロセスプラント及びプロセス制御システムにおける早期故障検出を提供するための例示的方法1200のフローダイアグラムを示す。方法1200のうちの1つ以上の部分が、例えば、ブロック定義ライブラリ252のデータブロックによって実施できる。方法1200のうちの1つ以上の部分が、工業プロセスパフォーマンス監視/分析システム100のうちの1つ以上の部分、例えば、1つ以上のDDE102x、オフラインデータモジュール、オンラインデータモジュール等によって実施できる。無論、方法1200のうちの1つ以上の部分は、本明細書に記載されるデータ分析システム100のシステム、デバイス、及び装置以外のものによって実施できる。一部の実施形態においては、方法1200は、本明細書に記載されるステップよりも多いか、少ないか、あるいはそれらとは異なるステップを含み得る。

40

【0364】

50

ブロック1202において、方法1200は、プロセスプラントによるプロセス制御の結果として生成されるプロセス信号またはデータの開始セットを受信することまたは獲得することを含み得る。プロセス信号の開始セットは、プロセスプラントにおけるプロセスを制御するように動作する1つ以上のプロセス制御デバイスによって生成される時間領域データを含み得、かつ/あるいはプロセスの制御の結果としてプロセスプラント内の1つ以上の他の構成要素、デバイス、またはエンティティによって生成される時間領域データ（例えば、プロセスプラント内の様々なデータソースによって生成される、一次プロセスデータ）を含んでもよい。例えば、プロセス信号の開始セットは、温度、フロー、圧力、組成、状態等の値等、経時的なプロセス測定値の値を含み得る。一部の実施形態においては、獲得したプロセス信号またはデータの開始セットは、プロセスプラントによるプロセス制御の結果として生成される、診断結果、オペレータまたはユーザの一連のアクション、リンクまたはプロセス通信ネットワークの利用可能な帯域幅、データ分析の結果等の、経時的な二次または高次プロセスデータを含み得る。獲得したプロセス信号の開始セットは、例えば、オフラインプロセスデータ及び/またはオンラインプロセスデータを含み得る。

10

【0365】

ブロック1204においては、方法1200は、獲得したプロセス信号の初期セットに基づいて、プロセスプラント内で発生する（または発生した）異常、故障、パフォーマンスの低下、または他の所望されない/望ましくない状態の先行インジケータを決定することを含み得る。例えば、先行インジケータは、特定のプロセス制御信号の特定の周波数のピーク振幅におけるスパイク（例えば、図7Bに示されるスパイク1104）等の、異常、故障、パフォーマンスの低下、及び/または他の状態の発生の前に起こる、1つ以上のプロセス制御信号の挙動における変化であり得る。ある実施形態においては、先行インジケータは、PCA、PLS回帰、クラスタリング、相互相関関係等の、獲得したプロセス信号の開始セットのうち1つ以上に対して1つ以上の統計分析技法を用いることによって決定できる。ある実施形態においては、データ分析システム100の1つ以上のオフライン及び/またはオンラインデータブロック及び/またはデータモジュールが、1つ以上の獲得したプロセス信号に対して動作して、異常、故障、パフォーマンスの低下、及び/または他の状態の1つ以上の先行インジケータを決定し得る。

20

【0366】

ブロック1206においては、方法1200は、先行インジケータに対応する1つ以上の新しいプロセス変数のセットを、作成、定義、または生成することを含み得る。ある実施形態においては、特定の信号の挙動における特定の変化に対応する新しいプロセス変数が、作成、定義、または生成され得る。例えば、先行インジケータが発生し得る目的の周波数（例えば、図7Cの信号1106の周波数）が特定され、新しいプロセス変数として作成/生成され得る。一部の実施形態においては、1つ以上の新しいプロセス変数のセットの生成（ブロック1206）は、先行インジケータに対応する第1の新しいプロセス変数（例えば、図7Cの信号1106に対する変化）を定義/特定/生成することと、続いて、第1の新しいプロセス変数によって生成される時系列データ（単独で、あるいは他のプロセス変数によって生成される他の時系列データと共に）に対して、1つ以上の分析技法を実施して、先行インジケータと関連付けられる別の新しいプロセス変数を決定することを含む。例えば、図7Dを参照すると、図7Cの信号1106に対応する周波数領域データに対してPCAが適用されて、第1の主構成要素を決定し、第1の主構成要素が別の新しいプロセス変数として特定/定義され、その経時的な値は、信号1112によって図7Cに表される。

30

40

【0367】

ブロック1206の一部の実装形態において、データ分析システム100の1つ以上のオフライン及び/またはオンラインデータブロック及び/またはデータモジュールは、先行インジケータに対応する1つ以上の新しいプロセス変数を特定し、作成/定義/生成するように動作し得る。この1つ以上の新しいプロセス変数は、一部の場においては、タ

50

グ付けされ、かつ/あるいはプロセスプラントまたはプロセス制御システム内に記憶され得る。

【0368】

ブロック1208においては、方法1200は、プロセス信号の後続セットを獲得すること(例えば、ブロック1202の様式と類似する様式で)と、それにローリングFFTを実施して、1つ以上の新しいプロセス変数に対応する時系列データを決定することを含み得る。プロセス信号の後続セットは、例えば、オフライン及び/またはオンライン信号を含み得る。それに対してローリングFFTを実施するには、プロセス信号の後続セットのデータの第1の窓に対してFFTを実施し得、その中の周波数のピーク振幅の値が時系列データとして保存されてもよく、データの第2の窓に対してFFTを実施し得、その中の周波数のピーク振幅の値が時系列データとして保存されてもよく、以降同様に続き、それによって、プロセス信号の後続セット内に含まれる1つ以上の目的の周波数についてのそれぞれの時系列データが生成される。例えば、特定のプロセス信号の特定の目的の周波数が、新しいプロセス変数に対応する場合、ローリングFFTを、プロセス信号の後続セットに対して適用して、プロセス信号の後続セット内の特定の目的の周波数の挙動を経時的に獲得し得、例えば、新しいプロセス変数によって生成されるピーク振幅値を含む時系列データを経時的に獲得し得る。分析技法を実施して新しいプロセス変数を特定した状況においては(例えば、上に考察したように、新しいプロセス変数は、信号1106に対応する周波数領域データの第1の主構成要素に対応する)、1つ以上の分析技法もまた、ブロック1208において適用されて、そのような新しいプロセス変数の時系列データを決定できる。ある実施形態においては、ブロック1208は、1つ以上のオンライン及び/またはオフラインデータブロック及び/またはデータモジュール等、データ分析システム100のうちの1つ以上の部分によって実施され得る。

10

20

【0369】

ブロック1210においては、方法1200は、1つ以上の新しいプロセス変数によって生成される時系列データ(例えば、ブロック1208において獲得されたもの)を、先行インジケータの存在について監視することを含み得る。例えば、特定の目的の周波数のピーク振幅が、規模及び/または持続期間の閾値を超える場合、先行インジケータの存在を検出できる。ある実施形態においては、新しいプロセス変数は、それぞれのタグまたは他のインジケータによって特定され、プロセスプラントに組み込まれ、したがって、その時系列データ値の監視(ブロック1215)は、プロセスプラントの故障検出アプリケーション、アラーム取扱アプリケーション、及び/または他の監視アプリケーションによって実施され得る。追加的または代替的に、先行インジケータの存在を検出する、時系列データの監視(ブロック1215)は、データ分析システム100によって実施されてもよい。

30

【0370】

ブロック1212においては、方法1200は、故障、異常、事象、パフォーマンスの低下、所望されない状態、及び/または所望される状態等の発生が、ブロック1210における監視中に発見された、検出された先行インジケータの存在に基づいて予想されることについての指標を生成することを含み得る。一部の場においては、ブロック1212は、該状態の発生が予想される時間枠の指標を生成することを含む。例えば、動向チャート、アラーム、アラート、及び/または他の視覚的もしくは聴覚的インジケータが、ユーザインターフェイスによって生成及び呈示されて、獲得した時系列データ中の先行インジケータの存在に基づいて、プロセスプラントにおける切迫した異常、故障、パフォーマンスの低下、及び/または他の状態についてユーザまたはオペレータに警告し得る。

40

【0371】

一部の実施形態においては(図7Gには図示せず)、方法1200は、信号(制御信号または変化を示す他の信号等)を生成させ、検出された先行インジケータの存在に基づいて、オンラインプロセスプラントに提供させることを含み得る。例えば、データ分析システム100は、監視されているデータにおいて検出された先行インジケータの存在に基づ

50

いて、1つ以上の制御信号を自動的に生成し得(ブロック1210)、この1つ以上の制御信号を1つ以上のコントローラに対して自動的に提供して、プロセス制御プラントの少なくとも一部分の挙動を変更することができる。オンラインプロセスプラントに変化をもたらすようにそこに提供され得る他の信号としては、例えば、(例えば、設備、デバイス、ルーチン、もしくはアプリケーション等の一部の)パラメータ、値、構成、及び/もしくは状態に対する変更、またはプロセスプラント内もしくはそれと併せて実行されるアプリケーションに対する変更を示すトリガまたは信号が挙げられる。

【0372】

一部の実施形態においては(図7Gには図示せず)、方法1200は、1つ以上の信号を生成させ、データ分析システム100に提供させることを含み得る。例えば、方法1200は、データ分析システム100に対して、新しいプロセス変数の指標及びそれらそれぞれの識別子、決定された先行インジケータの指標、時系列データ(及び、それに適用される様々な分析技法の出力等の、そこから生成されるより高次のデータ)に対して実施される様々な分析技法の特性及び配列を提供して、更なる新しいプロセス変数、新しいプロセス変数によって生成される、監視される時系列データ、その中の先行インジケータの存在等を決定し得る。概して、方法1200は、方法1200の実行によって生成される任意のデータを、データ分析システム100に提供させることを含み得る。ある実施形態においては、データ分析システム100に提供される1つ以上の信号は、ストリーミングデータであり得る。

【0373】

無論、方法1200はブロック1202~1212に限定されない。一部の実施形態においては、更なるブロックが方法1200によって実施され得、かつ/あるいはブロック1202~1212のうちの一部は方法1200から省略されてもよい。更に、方法1200の実施形態は、図1、2、3、4A~4Q、5A~5G、及び6A~6Eのうちの一つの部分または全ての部分と併せて、かつ/あるいは本開示の他の節に記載される任意の数の特色及び/または技法と併せて動作し得る。

【0374】

上記の見地から、例えば、制御及び他の処理が多かれ少なかれ中心的に実施されていた集中型プロセス制御システムから、制御及び他の処理がプラント全体にわたる複数のコントローラ中に分散された次世代分散型制御システム(DCS)への経時的な発展をよく知る者ならば、本明細書に記載される新規Distributed Analytics System(DAS)100及び関連付けられる工業プロセスパフォーマンス監視及び分析方法、デバイス、装置、構成要素、及び技法が、プロセス制御パフォーマンス監視及び分析の次世代への同様の飛躍を提供することを認識するであろう。例えば、プロセス制御システムのオペレータワークステーションにおいて、分析設備においてオフラインで、あるいは初歩的分析ツールを用いることによる、標的を決め、焦点を絞った分析を実施するのではなく、本明細書に開示される分散型分析システム100、ならびに関連付けられる方法、デバイス、装置、構成要素、及び技法は、プロセスプラントの現在のオペレーションに関する記述、予測、及び規定が恒常的に利用可能であるように、分散型制御システムの様式と同様の様式で、プラント全体にわたり、更には企業及び/またはエンタープライズのレベルにおける包括的パフォーマンス監視及びリアルタイム分析を可能にする(例えば、図2Bを参照)。加えて、DAS100のモジュール式の分散型アーキテクチャは、プロセスプラント及び/またはプロセス制御システムのデータソースに対して所望されるほど近い、あるいはそこから所望されるほど遠く離れた、パフォーマンス及び分析機能の埋め込みを可能にし、更なるDDE102xをDAS100に対して所望される通りに、あるいは所望されるときに追加することを可能にする(例えば、よりローカライズされた処理馬力のため、プラント5の特定の領域またはロケーションにおいて分析を提供するため、プラント5が拡張される場合等)。更に、上で説明された通り、DAS100は、データモジュールの設計、定義、展開、及び監視を、それらが実行されている環境及びプラットフォームから分割または分離させ、それによって、分析及びデータモジュール/モ

10

20

30

40

50

デルのコーディング及びコンパイルについて心配する必要性からデータエンジニアまたはユーザを完全に開放する。なおも更に、この分離は、より高い柔軟性、及び異なるターゲット環境に対するシームレスな移行を可能にする。

【0375】

ソフトウェアに実装される場合、本明細書に記載されるアプリケーション、サービス、及びエンジンは、磁気ディスク、レーザディスク、固体メモリデバイス、分子メモリストレージデバイス、または他の記憶媒体、コンピュータもしくはプロセッサのRAMもしくはROM等の、任意の有形の、非一時的コンピュータ可読メモリに記録され得る。本明細書に開示される例示的システムは、他の構成要素の中でも、ハードウェア上で実行されるソフトウェア及び/またはファームウェアを含むように開示されているが、そのようなシステムは単に例示的であるに過ぎず、限定的であると見なされるべきではないことに留意されたい。例えば、これらのハードウェア、ソフトウェア、及びファームウェア構成要素のうちのいずれかまたは全てが、ハードウェアにのみ、ソフトウェアにのみ、あるいはハードウェア及びソフトウェアの任意の組み合わせで、埋め込まれ得ることが企図される。したがって、本明細書に記載される例示的システムは、1つ以上のコンピュータデバイスのプロセッサにおいて実行されるソフトウェアに実装されるように記載されているが、当業者ならば、提供される例が、そのようなシステムを実装するための唯一の方法であるわけではないことを容易に理解するであろう。

10

【0376】

したがって、本発明は具体的な例に関して記載されてきたが、これらの例は例証的である過ぎず、本発明の限定であることを意図せず、変更、追加、または削除が、本発明の趣旨及び範囲から逸脱することなく、開示される実施形態に対して行われ得ることが当業者には明らかであろう。

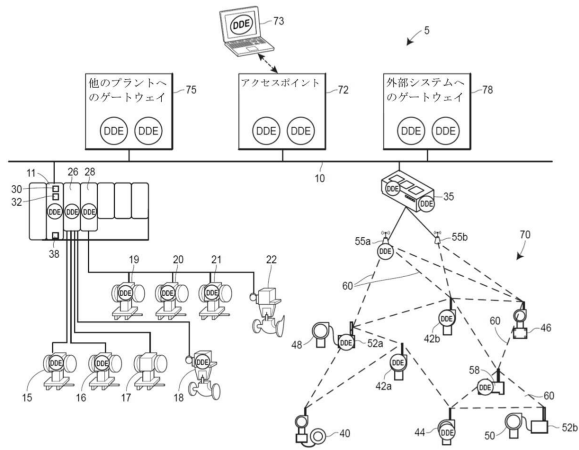
20

【0377】

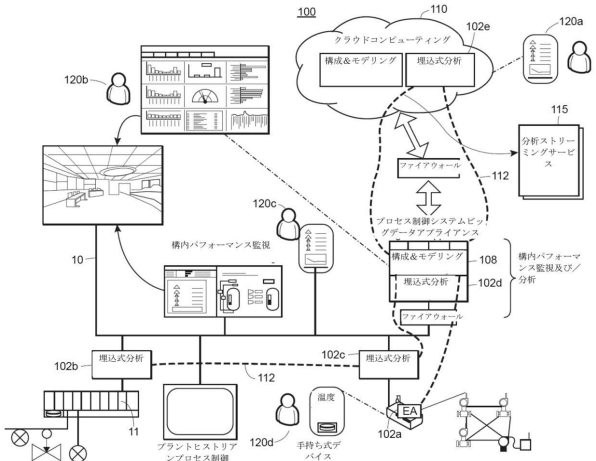
任意の具体的な実施形態の特定の特色、構造、及び/または特性は、他の特色の対応する使用を伴う、または伴わない選択された特色の使用を含む、任意の好適な様式で、かつ/あるいは1つ及び/またはそれ以上の他の実施形態との任意の好適な組み合わせで、組み合わせてもよい。加えて、本発明の本質的な範囲または趣旨に対して、特定の用途、状況、及び/または材料を適合させるように、多くの修正がなされてもよい。本明細書に記載及び/または例証された本発明の実施形態の他の変形及び/または修正が、本明細書の教示の観点から可能であり、本発明の趣旨または範囲の一部としてみなされるべきことを理解されたい。本発明のある特定の態様は、例示的態様として本明細書に記載されている。

30

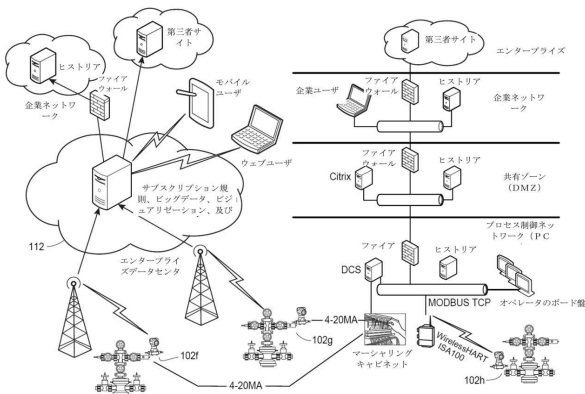
【図1】



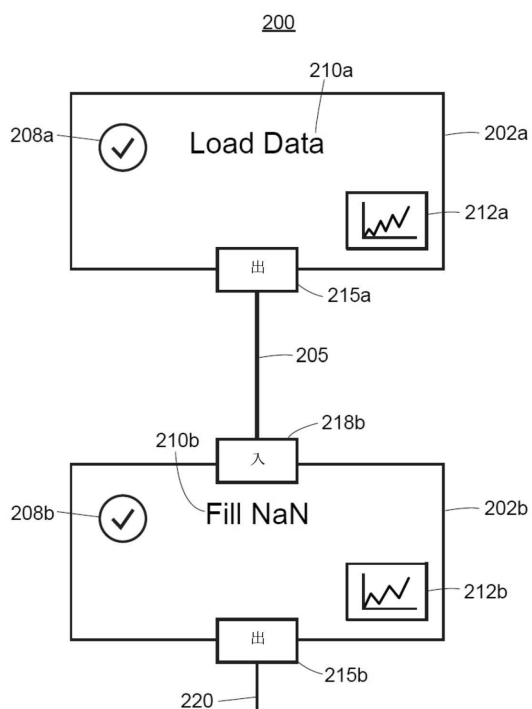
【図2A】



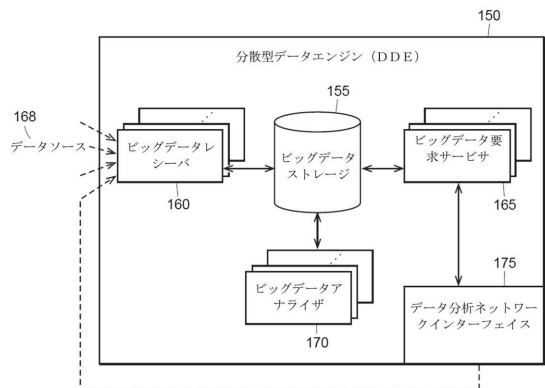
【図2B】



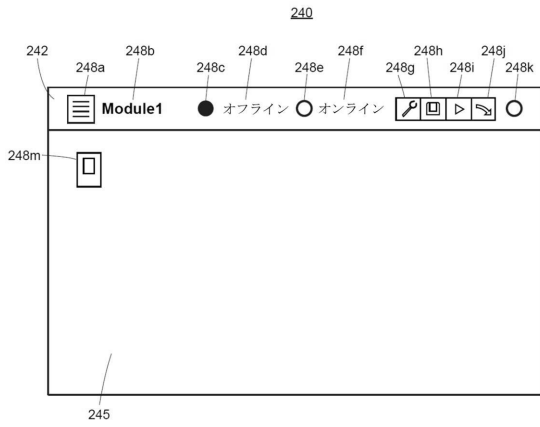
【図4A】



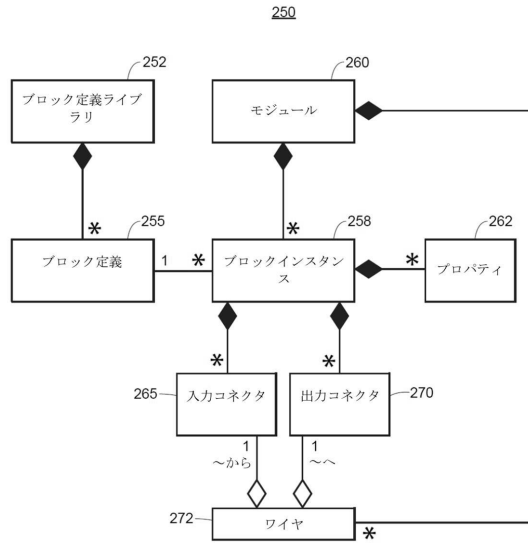
【図3】



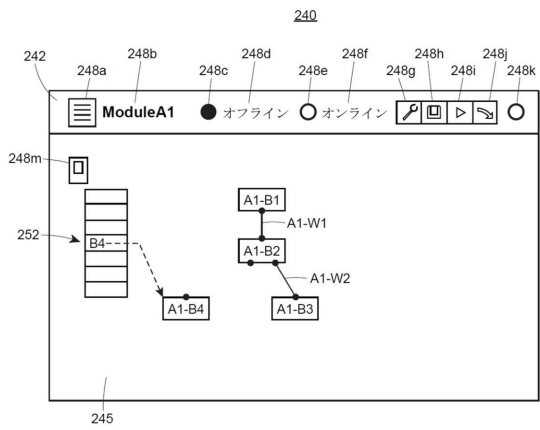
【図4B】



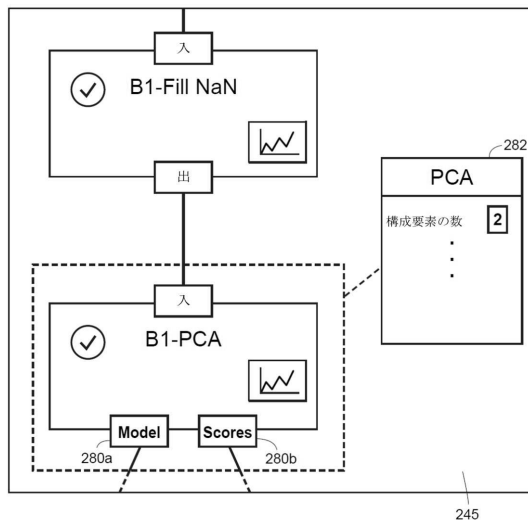
【図4C】



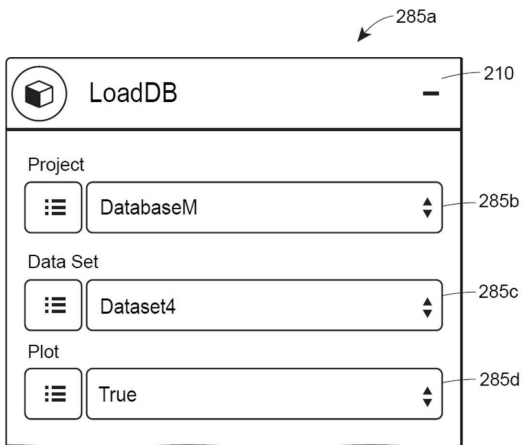
【図4D】



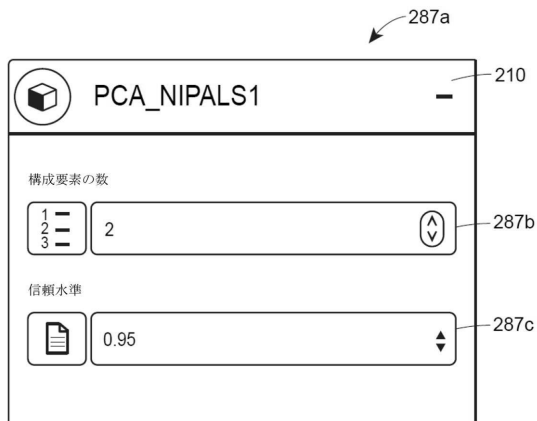
【図4E】



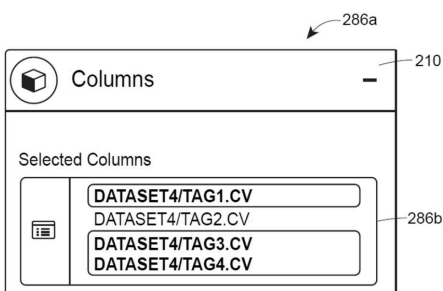
【 図 4 F 】



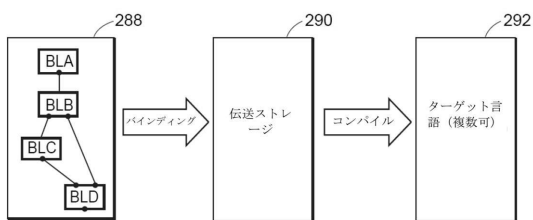
【 図 4 H 】



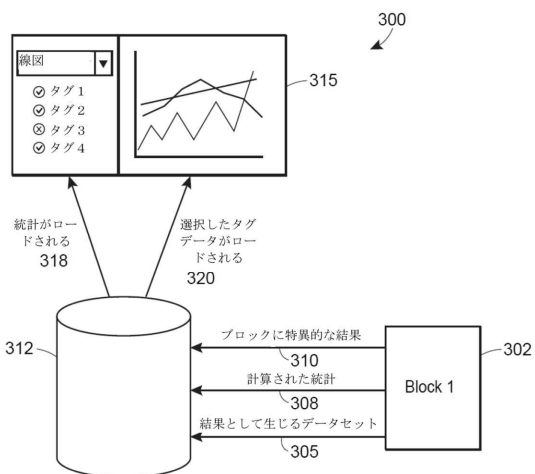
【 図 4 G 】



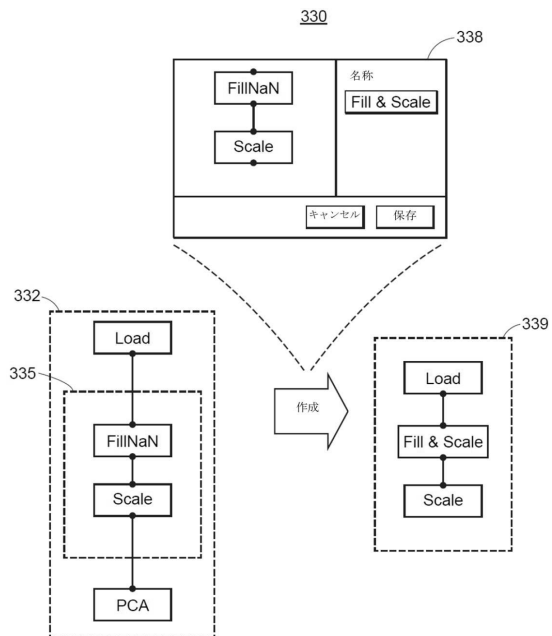
【 図 4 I 】



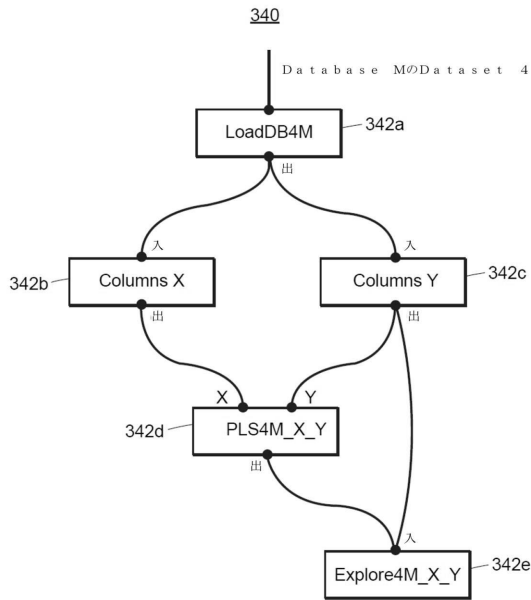
【 図 4 J 】



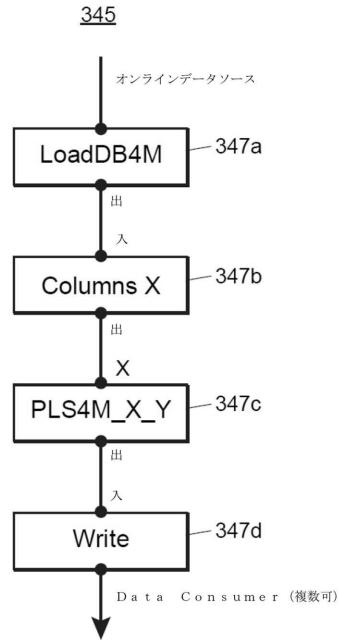
【 図 4 K 】



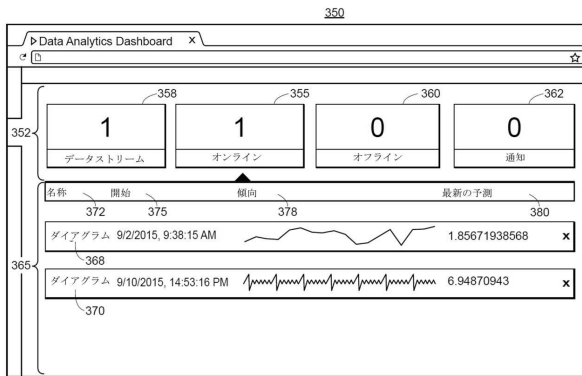
【図 4 L】



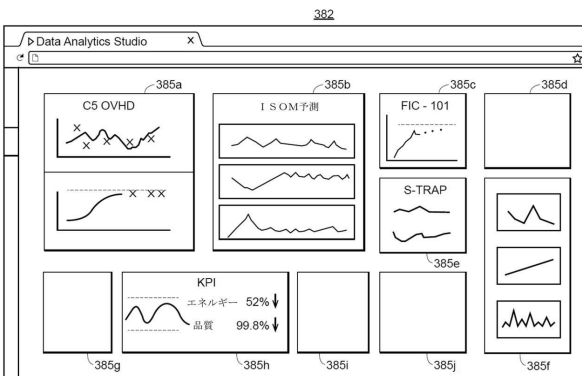
【図 4 M】



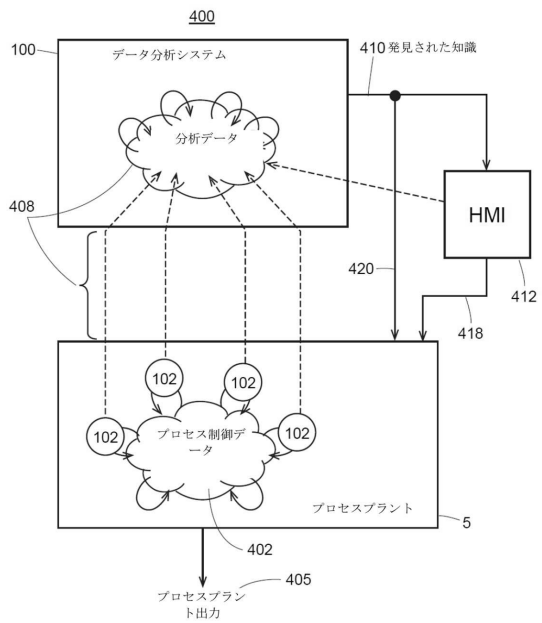
【図 4 N - 1】



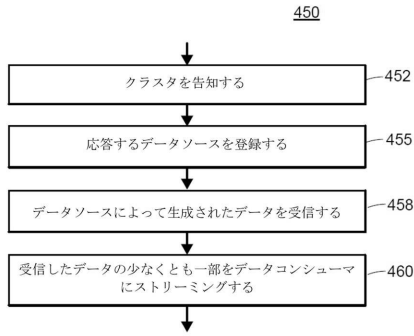
【図 4 N - 2】



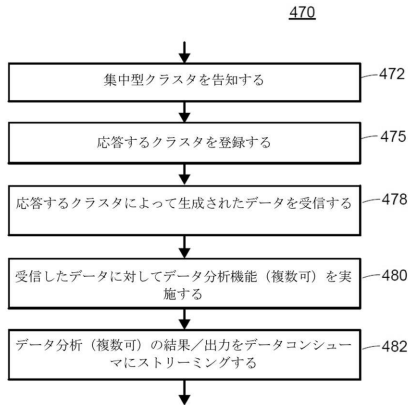
【図 4 O】



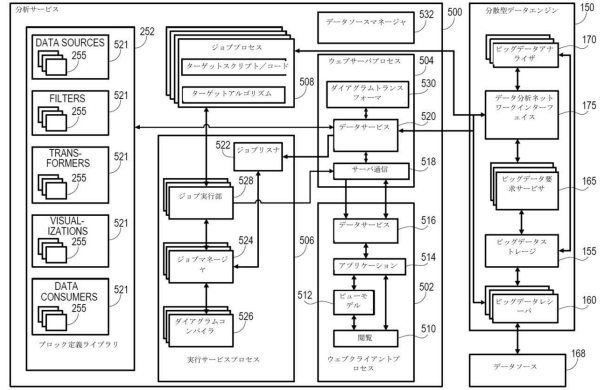
【図4P】



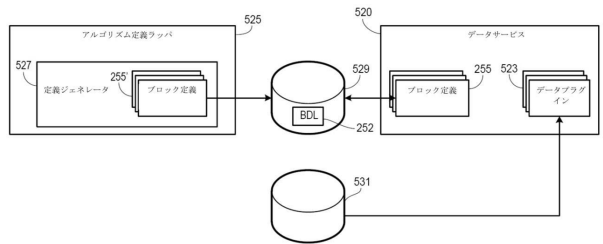
【図4Q】



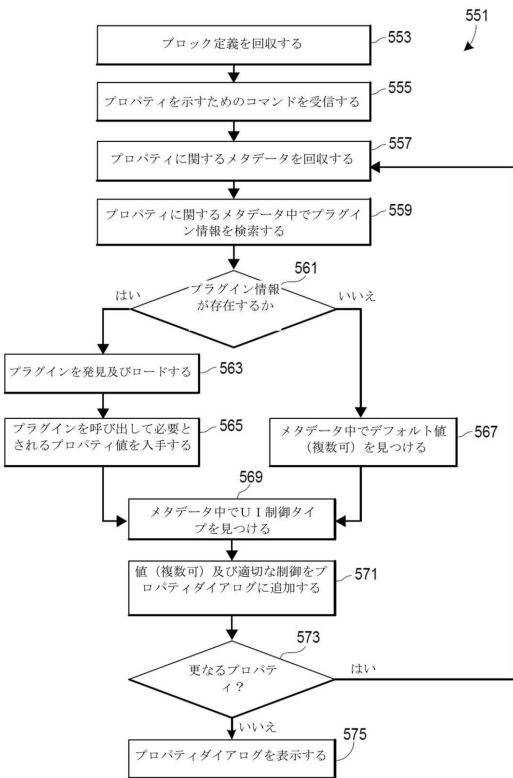
【図5A】



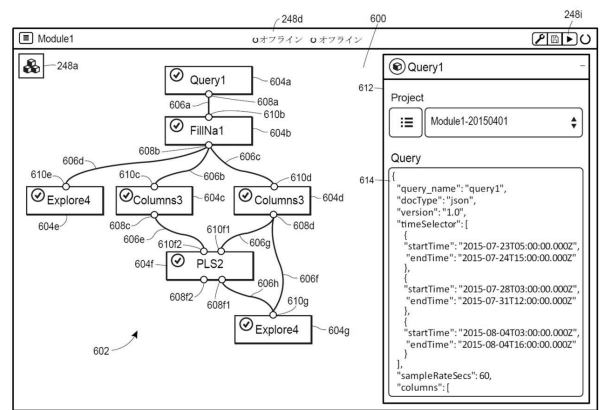
【図5B】



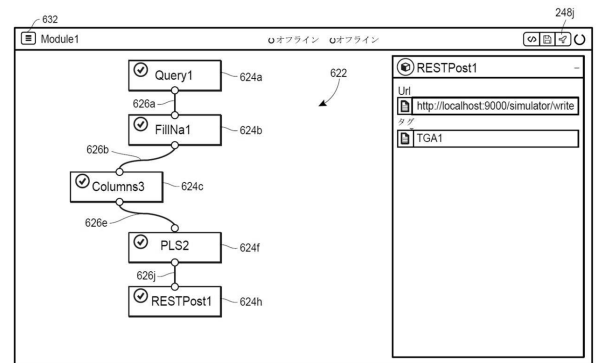
【図5C】



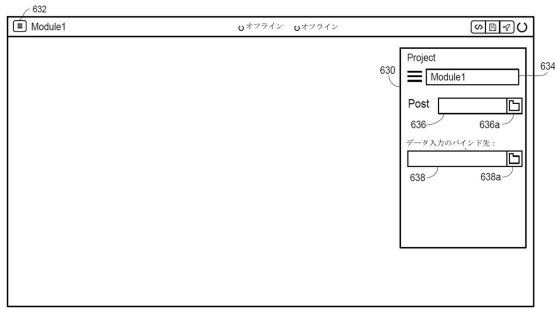
【図5D】



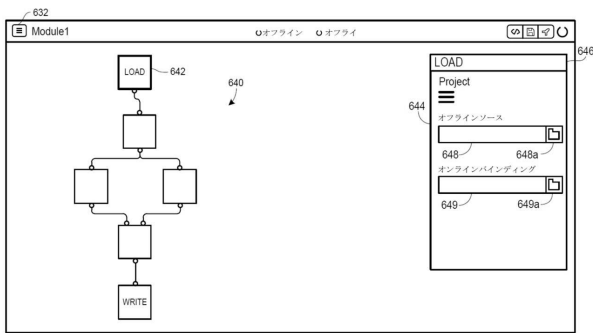
【図5E】



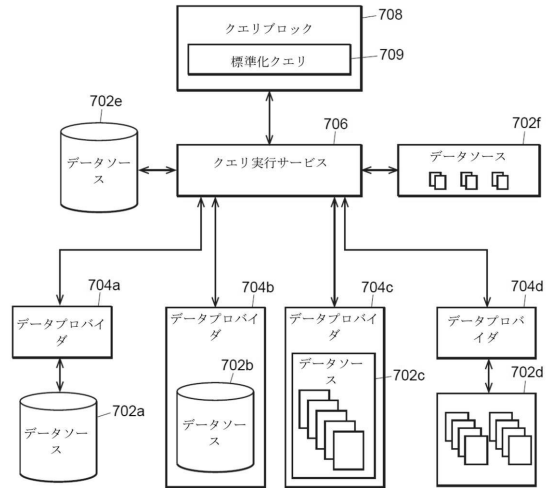
【図5F】



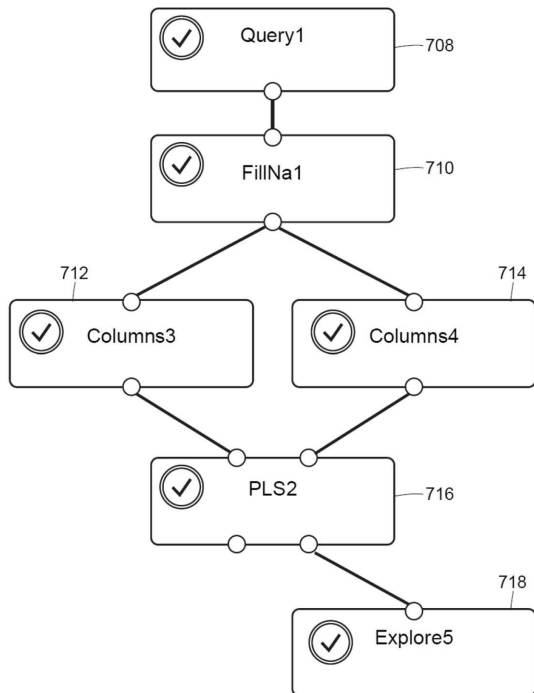
【図5G】



【図6A】



【図6B】



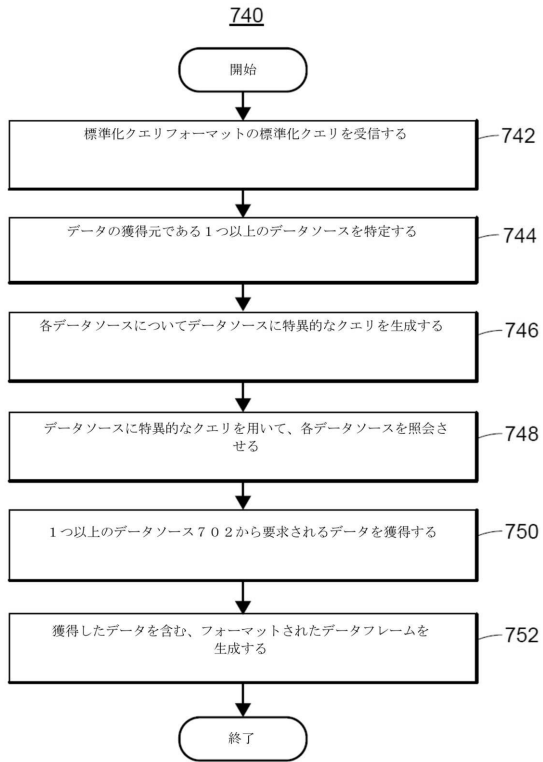
【図6C】

```

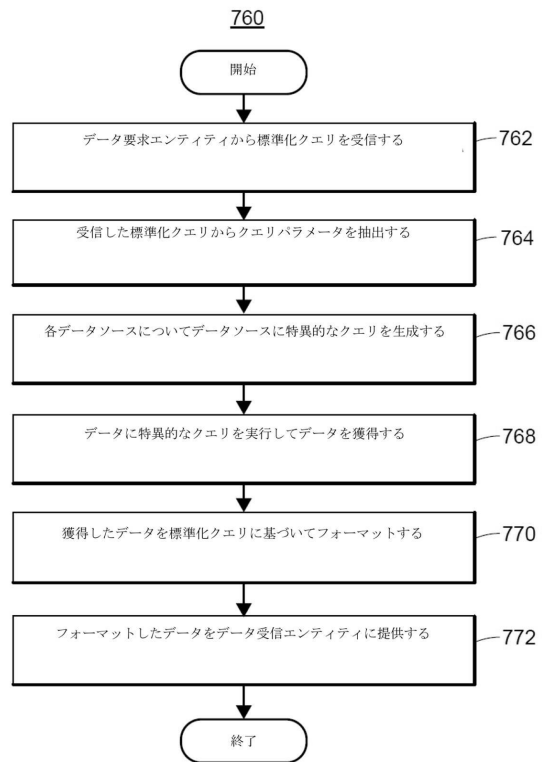
{
  "query_name": "query1",
  "docType": "json",
  "version": "1.0",
  "timeSelector": [
    {
      "startTime": "2015-07-23T05:00:00.000Z",
      "endTime": "2015-07-24T15:00:00.000Z"
    },
    {
      "startTime": "2015-07-28T03:00:00.000Z",
      "endTime": "2015-07-31T12:00:00.000Z"
    },
    {
      "startTime": "2015-08-04T03:00:00.000Z",
      "endTime": "2015-08-04T16:00:00.000Z"
    }
  ],
  "sampleRateSecs": 60,
  "columns": [
    {
      "tag": "FT630B/DENS.CV",
      "alias": "FT630B/DENS.CV",
      "dataType": "Float",
      "renderType": "VALUE",
      "format": "0.###"
    },
    {
      "tag": "PT615/WIRED_PSIA.CV",
      "alias": "PT615/WIRED_PSIA.CV",
      "dataType": "Float",
      "renderType": "VALUE",
      "format": "0.###"
    },
    {
      "tag": "TT6079/INPUT_1.CV",
      "alias": "TT6079/INPUT_1.CV",
      "dataType": "Float",
      "renderType": "VALUE",
      "format": "0.###"
    },
    {
      "tag": "630.molefrac.c5",
      "alias": "630.molefrac.c5",
      "dataType": "Float",
      "renderType": "VALUE",
      "format": "0.###"
    }
  ]
}

```

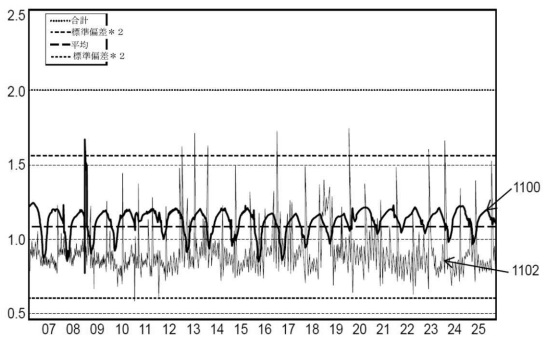
【図 6 D】



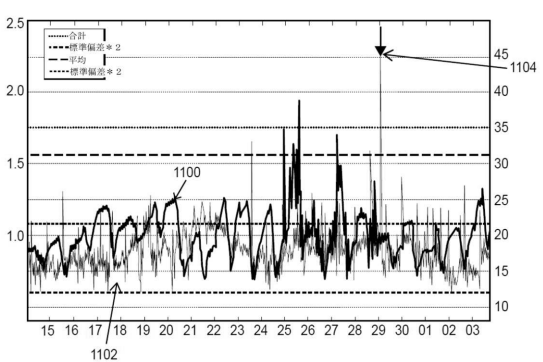
【図 6 E】



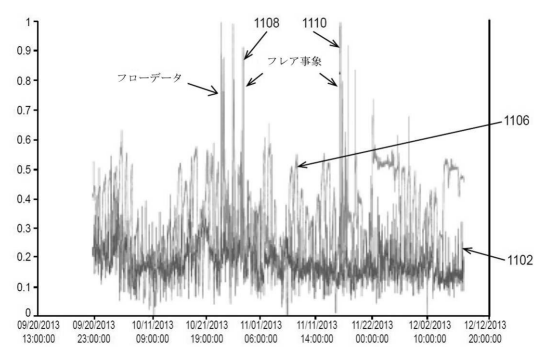
【図 7 A】



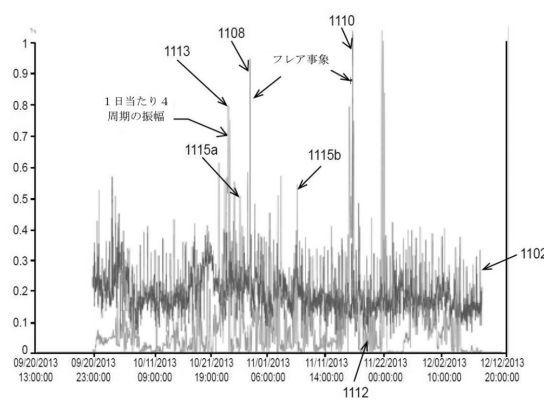
【図 7 B】



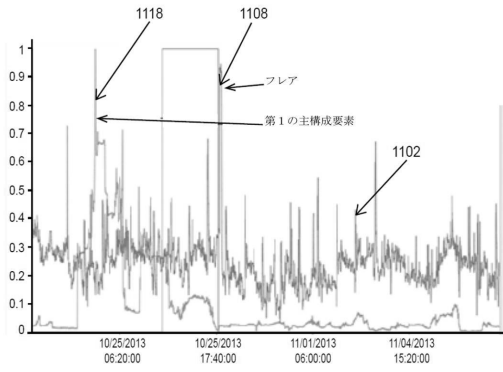
【図 7 C】



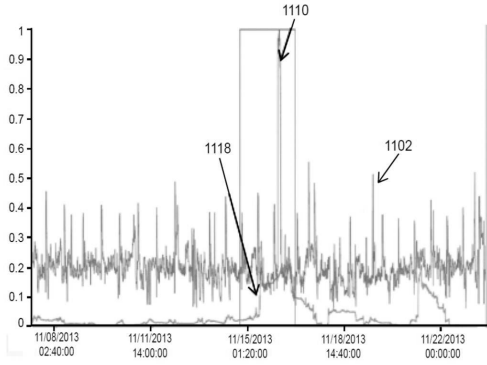
【図 7 D】



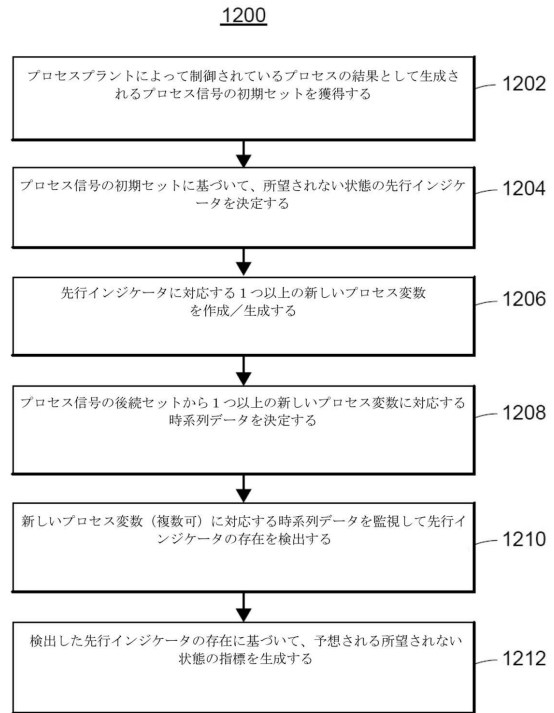
【図7E】



【図7F】



【図7G】



フロントページの続き

- (72)発明者 マーク ジョン、 ニクソン
アメリカ合衆国 テキサス州 78681 ラウンド ロック ブラックジャック ドライブ 1
503
- (72)発明者 エルパー ターハン、 エンバー
アメリカ合衆国 テキサス州 78745 オースティン ケスウィック ドライブ 7805
- (72)発明者 ノエル ホワード、 ベル
アメリカ合衆国 テキサス州 78745 オースティン ヒルトン ドライブ 6224
- (72)発明者 ジョシュア ブライアン、 キッド
アメリカ合衆国 フロリダ州 34695 セーフティ ハーバー ウエストリー ストリート
1304
- (72)発明者 ポール アール .、 マストン
イギリス国 エルイー19 3ディワイ ナーボロウ ザ パスチャーズ 59

審査官 大古 健一

- (56)参考文献 特開2008-269489(JP, A)
特開2008-250805(JP, A)

- (58)調査した分野(Int.Cl., DB名)
G05B 23/00 - 23/02