

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7368374号
(P7368374)

(45)発行日 令和5年10月24日(2023.10.24)

(24)登録日 令和5年10月16日(2023.10.16)

(51)国際特許分類

F I

G 0 5 B 19/418 (2006.01)

G 0 5 B 19/418 Z

請求項の数 57 (全49頁)

(21)出願番号	特願2020-555407(P2020-555407)	(73)特許権者	512132022
(86)(22)出願日	平成31年4月18日(2019.4.18)		フィッシャー・ローズマウント システ
(65)公表番号	特表2021-521522(P2021-521522		ムズ, インコーポレイテッド
	A)		アメリカ合衆国 テキサス 7 8 6 8 1 -
(43)公表日	令和3年8月26日(2021.8.26)		7 4 3 0 ラウンド ロック ウェスト ル
(86)国際出願番号	PCT/US2019/028138		イス ヘナ ブルバード 1 1 0 0 ビルデ
(87)国際公開番号	WO2019/204616		イング 1 エマーソン プロセス マネー
(87)国際公開日	令和1年10月24日(2019.10.24)		ジメント
審査請求日	令和4年4月18日(2022.4.18)	(74)代理人	100096091
(31)優先権主張番号	62/659,325		弁理士 井上 誠一
(32)優先日	平成30年4月18日(2018.4.18)	(72)発明者	ジョシュア・ビー・キッド
(33)優先権主張国・地域又は機関	米国(US)		アメリカ合衆国 フロリダ州 3 3 7 0 1
			セイフティ ハーバー ウェストリー ス
			トリート 1 3 0 4
		(72)発明者	マーカス・エス・ストレンジウェイ・オ
			最終頁に続く

(54)【発明の名称】 品質レビュー管理システム

(57)【特許請求の範囲】

【請求項1】

プロセス機器によって実施されているプロセスの動作をレビューする際に使用するための品質レビュー管理システムであって、

ルールデータベースと、

ユーザインターフェースおよび前記ルールデータベースに通信可能に結合されたコンフィグレーションシステムであって、前記コンフィグレーションシステムは、ユーザが前記ユーザインターフェースを介して1つ以上のルールを作成することを可能にし、前記ルールの各々は、プロセス内の1つ以上の例外を識別し、前記コンフィグレーションシステムは、前記1つ以上のルールを前記ルールデータベース内に記憶する、コンフィグレーションシステムと、

前記ルールデータベース内の前記ルールを使用して、前記プロセス機器からのデータメッセージ内のデータを処理して、前記データメッセージ内の前記データが、前記ルールのうちの1つ以上によって定義された例外を含むかどうかを判断する例外エンジンであって、前記ルールのうちの1つに基づいて例外を識別すると、前記例外についての情報を例外レコードとして記憶する、例外エンジンと、を備える、品質レビュー管理システム。

【請求項2】

前記コンフィグレーションシステムは、ユーザ入力に基づいて、複数のルールを生成し、前記複数のルールの各々は、例外を識別するために満たされなければならない一組のコンディションを識別する、請求項1に記載の品質レビュー管理システム。

【請求項 3】

前記複数のルールの各々は、前記識別された例外の重大度をさらに識別する、請求項 2 に記載の品質レビュー管理システム。

【請求項 4】

前記複数のルールの各々は、前記例外についての情報をさらに識別する、請求項 2 に記載の品質レビュー管理システム。

【請求項 5】

前記複数のルールの各々は、前記例外を取り扱う、解決する、またはクローズするのに使用されるプロセスまたは手順をさらに識別する、請求項 2 に記載の品質レビュー管理システム。

【請求項 6】

前記複数のルールの各々は、品質レビュー技師が前記例外を理解および解決することを支援するために、例外レコードの一部として記憶されるプロセスデータまたは他のデータをさらに識別する、請求項 2 に記載の品質レビュー管理システム。

【請求項 7】

前記例外エンジンは、前記プロセス機器内のデータソースによって定期的に提供されるデータメッセージを受信する、請求項 1 に記載の品質レビュー管理システム。

【請求項 8】

前記データソースは、プロセス制御システム、バッチエグゼクティブ、またはオペレータワークフローアプリケーションのうちの 1 つである、請求項 7 に記載の品質レビュー管理システム。

【請求項 9】

前記コンフィグレーションシステムは、1 つ以上のルールを生成し、各ルールは、ルール名と、ルールタイプと、ルール重大度と、前記ルールに関連付けられた前記例外を解決または取り扱うためにとるべき 1 つ以上の手順またはステップと、前記例外が発生した時間の、前記ルールに関する前記例外が発生した前記プロセスの動作に関するプロセスデータと、を含む、請求項 1 に記載の品質レビュー管理システム。

【請求項 10】

前記例外エンジンは、識別された各例外に対するデータを、命令に関連付けられた例外ファイル内に、例外レコードとして記憶する、請求項 1 に記載の品質レビュー管理システム。

【請求項 11】

前記例外エンジンは、命令を実施する前記プロセスの前記動作中にリアルタイムで動作して、前記命令に対する例外ファイルを作成する、請求項 1 に記載の品質レビュー管理システム。

【請求項 12】

前記例外エンジンは、プロセスオペレータに、前記ルールのうちの 1 つ以上による例外の検出に基づいて、例外が発生した、または発生しようとしているというライブフィードバックを提供する、請求項 11 に記載の品質レビュー管理システム。

【請求項 13】

前記コンフィグレーションシステムは、例外を検出するための、異なるタイプの入力データに関連付けられた一組のルールテンプレートを含み、前記コンフィグレーションシステムは、前記ユーザが前記ユーザインターフェースを介して、前記一組のルールテンプレートのうちの 1 つを選択すること、ならびに前記ユーザが前記ユーザインターフェースを介して、前記選択されたルールテンプレートを使用して前記ルールが作成されるためのルールパラメータおよびルール入／出力を定義することを可能にする、請求項 1 に記載の品質レビュー管理システム。

【請求項 14】

前記コンフィグレーションシステムは、前記ユーザが前記ユーザインターフェースを介して、例外を検出するための論理式で 사용되는一組の入力データおよびルールパラメー

10

20

30

40

50

タに対して分析される前記論理式を定義することを可能にする、請求項 13 に記載の品質レビュー管理システム。

【請求項 15】

品質レビュー管理システムであって、

複数のルールを記憶するルールデータベースであって、前記複数のルールの各々が、プロセス内の 1 つ以上の例外に関連付けられたコンディションを識別するロジックを含む、ルールデータベースと、

前記ルールデータベース内の前記ルールを使用して、プロセス機器からのデータメッセージ内のデータを処理して、前記複数のルールのうちのいずれかによって定義された例外が存在するかどうかを判断する例外エンジンであって、前記ルールのうちの 1 つに基づいて例外を識別すると、前記例外についての情報を例外レコードとして記憶する、例外エンジンと、

10

ユーザが、前記例外エンジンによって識別された、かつ前記例外レコードとして例外データベース内に記憶された例外を取り扱うことを可能にするレビューインターフェースであって、前記例外データベースにアクセスし、1 つ以上の識別された例外に関する情報をユーザに前記ユーザインターフェースを介して提示し、各例外レコードに対して記憶されたデータを前記ユーザに提供し、前記ユーザが各例外レコードに関してアクションをとることを可能にする、レビューインターフェースと、を備える、品質レビュー管理システム。

【請求項 16】

前記レビューインターフェースは、前記ユーザに、各例外レコードについて、例外のタイプまたは例外の重大度を提供する、請求項 15 に記載の品質レビュー管理システム。

20

【請求項 17】

前記レビューインターフェースは、前記ユーザに、各例外レコードについて、前記例外が発生した時間に存在した 1 つ以上のプロセス変数、状態、または手順に関するデータを提供する、請求項 15 に記載の品質レビュー管理システム。

【請求項 18】

前記例外エンジンは、各例外について、前記例外時に処理されていた命令を記憶し、前記レビューインターフェースは、前記ユーザに、特定の命令に対する一群の例外を提供する、請求項 15 に記載の品質レビュー管理システム。

【請求項 19】

前記レビューインターフェースは、前記ユーザが例外を認識することを可能にする、請求項 15 に記載の品質レビュー管理システム。

30

【請求項 20】

前記レビューインターフェースは、前記ユーザが例外を承認またはクローズすることを可能にする、請求項 15 に記載の品質レビュー管理システム。

【請求項 21】

前記レビューインターフェースは、前記ユーザが例外レコードにさらなる情報で注釈を付けることを可能にする、請求項 15 に記載の品質レビュー管理システム。

【請求項 22】

前記レビューインターフェースは、前記ユーザが例外レコードを別の人に送信することを可能にする、請求項 15 に記載の品質レビュー管理システム。

40

【請求項 23】

前記レビューインターフェースは、前記レビュー者に、前記例外に基づいて実行される 1 つ以上の示唆されるまたは必須のアクションを提供する、請求項 15 に記載の品質レビュー管理システム。

【請求項 24】

前記レビューインターフェースは、特定の数またはタイプの人々による承認を得ることを前記ユーザに強制するか、または後で使用するために前記例外レコードと共に記憶される特定の情報が前記レビュー者から提供されることを要求する、請求項 23 に記載の品質レビュー管理システム。

50

【請求項 25】

前記レビューインターフェースは、前記処理された例外レコードをデータベース内に記憶し、前記ユーザが、前記例外データベース内の前記例外レコードをスクロールして、命令に対するすべての前記例外レコードが処理またはクローズされたかどうかおよびそれがいっつかであるか、前記レビューのステータス、または前記例外レコードに関する統計データを決定することを可能にする、請求項15に記載の品質レビュー管理システム。

【請求項 26】

前記レビューインターフェースは、前記ユーザが、1つ以上の異なる方法で例外レコードをグループ化することを可能にする、請求項15に記載の品質レビュー管理システム。

【請求項 27】

前記レビューインターフェースは、前記ユーザが、グループ全体の例外レコードについて1つ以上のアクションをとることを可能にする、請求項15に記載の品質レビュー管理システム。

【請求項 28】

前記例外エンジンは、プロセス機器からのデータメッセージ内のプロセスデータを、前記プロセス機器のオンライン動作中に受信し、かつ前記ルールデータベース内の前記ルールを使用して、前記プロセス機器のオンライン動作中に例外が存在するかどうかを判断する、請求項15に記載の品質レビュー管理システム。

【請求項 29】

前記例外エンジンは、前記プロセス機器のオンライン動作中に検出された例外を、ユーザインターフェースを介してユーザに通知する、請求項28に記載の品質レビュー管理システム。

【請求項 30】

ユーザインターフェースおよび前記ルールデータベースに通信可能に結合されたコンフィグレーションシステムをさらに含み、前記コンフィグレーションシステムは、ユーザが前記ユーザインターフェースを介して1つ以上のルールを作成することを可能にし、前記ルールの各々は、プロセス内の1つ以上の例外を識別し、前記コンフィグレーションシステムは、前記例外エンジンによって、前記1つ以上のルールを前記ルールデータベース内に記憶する、請求項15に記載の品質レビュー管理システム。

【請求項 31】

プロセス機器によって実施されているプロセスの動作を分析する方法であって、
ユーザが、プロセッサに結合されたユーザインターフェースを介して、1つ以上のルールのためのルール定義データを指定することを可能にすることと、
前記指定されたルール定義データに基づいて、プロセッサを使用して、1つ以上のルールを作成することであって、前記1つ以上のルールの各々が、プロセス内の1つ以上の例外を識別する、作成することと、
前記1つ以上のルールを前記ルールデータベース内に記憶することと、
前記プロセスの前記動作中に、プロセスデータを含む、前記プロセス機器からの1つ以上のデータメッセージを受信することと、
前記ルールデータベース内の前記ルールを使用して、前記プロセス機器からの前記データメッセージ内の前記プロセスデータを処理して、前記データメッセージ内の前記データが、前記ルールのうちの1つ以上によって定義された例外を含むかどうかを判断することと、
前記ルールのうちの1つに基づいて例外を識別すると、前記例外に関する情報を例外レコードとして別のデータベースに記憶することと、を含む、方法。

【請求項 32】

前記ユーザがルール定義データを指定することを可能にすることは、前記ユーザインターフェースを介して、異なるタイプのルールを定義する1つ以上のルールテンプレートを前記ユーザに提示することと、前記ユーザが、前記ルールを作成するために、前記1つ以上のルールテンプレートのうちの1つを選択して使用することを可能にすることと、を含

10

20

30

40

50

む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 3】

前記ユーザがルール定義データを指定することを可能にすることは、前記ユーザインターフェースを介して、前記選択されたルールテンプレートのための記入されるべき複数のデータフィールドを前記ユーザに提示して、前記ルールを定義することを含む、請求項 3 2 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 4】

前記ユーザがルール定義データを指定することを可能にすることは、前記プロセスの動作における例外を識別するために満たされなければならない一組のコンディションをユーザが指定することを可能にすることを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 5】

前記ユーザがルール定義データを指定することを可能にすることは、前記ユーザが前記識別された例外の重大度を識別することを可能にすることを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 6】

前記ユーザがルール定義データを指定することを可能にすることは、前記例外が前記プロセス内に存在することが判明したときに提示される、前記例外に関する情報を前記ユーザが識別することを可能にすることを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 7】

前記ユーザがルール定義データを指定することを可能にすることは、前記ユーザが前記例外を取り扱う、解決する、またはクローズするのに使用されるプロセスもしくは手順を識別することを可能にすることを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 8】

前記 1 つ以上のルールを作成することは、品質レビュー技師が前記例外を理解および解決することを支援するために、例外レコードの一部として記憶されるプロセスデータまたは他のデータを指定することを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 3 9】

前記プロセス機器から 1 つ以上のデータメッセージを受信することは、前記プロセス機器内のデータソースによって前記データメッセージを定期的に受信することを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 4 0】

前記プロセス機器から 1 つ以上のデータメッセージを受信することは、前記 1 つ以上のデータメッセージを、プロセス制御システム、バッチエグゼクティブ、またはオペレータワークフローアプリケーションのうちの 1 つから受信することを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 4 1】

前記 1 つ以上のルールを作成することは、ルール名と、ルールタイプと、ルール重大度と、前記ルールに関連付けられた前記例外を解決または取り扱うためにとるべき 1 つ以上の手順もしくはステップと、前記例外が発生した時間の、前記ルールに関する前記例外が発生した前記プロセスの動作に関するプロセスデータと、を含むように、前記 1 つ以上のルールの各々を作成することを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 4 2】

前記プロセス機器からの前記データメッセージ内の前記プロセスデータを処理することは、命令を実施する前記プロセスの動作中にリアルタイムで動作することを含み、前記例外に関する情報を例外レコードとして別のデータベースに記憶することが、識別された各

10

20

30

40

50

例外に対するデータを、例外レコードとして、前記命令に関連付けられた例外ファイル内に記憶することを含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 4 3】

プロセスオペレータに、前記ルールの中の 1 つ以上による例外の検出に基づいて、例外が発生した、または発生しようとしているというライブフィードバックを提供することをさらに含む、請求項 3 1 に記載の前記プロセスの前記動作を分析する方法。

【請求項 4 4】

プロセスで品質レビューを実行する方法であって、

複数のルールをルールデータベース内に記憶することであって、前記複数のルールの各々が、前記プロセス内の 1 つ以上の例外に関連付けられたコンディションを識別するロジックを含む、記憶することと、

前記プロセスの動作中に、プロセス機器からの 1 つ以上のデータメッセージを受信することと、

プロセッサデバイスを使用して、前記ルールデータベース内の前記ルールを使用して前記データメッセージを処理して、前記複数のルールのいずれかによって定義された例外が存在するかどうかを判断することと、

前記ルールの中の 1 つに基づいて例外を識別すると、前記例外に関する情報を例外レコードとして別のデータベースに記憶することと、

ユーザが、ユーザインターフェースおよび処理デバイスを介して、前記例外レコードとして前記例外データベース内に記憶されている識別された例外を取り扱うことを可能にすることと、を含み、前記可能にすることが、前記例外データベースにアクセスすること、1 つ以上の識別された例外に関する情報をユーザに前記ユーザインターフェースを介して提示すること、各例外レコードに対して記憶されたデータを前記ユーザに提供すること、および前記ユーザが各例外レコードに関してアクションをとることを可能にすること、を含む、方法。

【請求項 4 5】

1 つ以上の識別された例外に関する情報をユーザに前記ユーザインターフェースを介して提示することは、前記ユーザに、各例外レコードについて、例外のタイプまたは例外の重大度を提供することを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 4 6】

1 つ以上の識別された例外に関する情報をユーザに前記ユーザインターフェースを介して提示することは、前記ユーザに、各例外レコードについて、前記例外が発生した時間に存在した、1 つ以上のプロセス変数、状態、または手順に関するデータを提供することを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 4 7】

1 つ以上の識別された例外に関する情報をユーザに前記ユーザインターフェースを介して提示することは、前記ユーザに、処理されていた特定の命令の識別、および前記特定の命令に対する一群の例外を提供することを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 4 8】

前記ユーザが各例外レコードに関してアクションをとることを可能にすることは、前記ユーザが例外を電子的に認識することを可能にすることを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 4 9】

前記ユーザが各例外レコードに関してアクションをとることを可能にすることは、前記ユーザが例外を電子的にサインオフまたはクローズすることを可能にすることを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 0】

前記ユーザが各例外レコードに関してアクションをとることを可能にすることは、前記ユーザが例外レコードにさらなる情報で電子的に注釈を付けることを可能にすることを含

10

20

30

40

50

む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 1】

前記ユーザが各例外レコードに関してアクションをとることを可能にすることは、前記ユーザが例外レコードを別の人に電子的に送信することを可能にすることを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 2】

各例外レコードに対して記憶されたデータを前記ユーザに提供することは、前記ユーザに、各例外レコードに対して実行される、1 つ以上の示唆されるまたは必須のアクションを提供することを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 3】

前記ユーザが、前記例外データベース内の前記例外レコードをスクロールして、命令に対するすべての前記例外レコードが処理もしくはクローズされたかどうかおよびそれがいつであるか、前記レビューのステータス、または前記例外レコードに関する統計データを決定することを可能にする、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 4】

前記ユーザが、1 つ以上の異なる方法で例外レコードをグループ化すること、およびグループ全体の例外レコードについて1 つ以上のアクションをとること、を可能にすることを含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 5】

プロセス機器からのデータメッセージ内のプロセスデータを、前記プロセス機器のオンライン動作中に受信することと、前記ルールデータベース内の前記ルールを使用して、前記プロセス機器のオンライン動作中に例外が存在するかどうかを判断することと、を含む、請求項 4 4 に記載の品質レビューを実行する方法。

【請求項 5 6】

前記プロセス機器のオンライン動作中に検出された例外を、ユーザインターフェースを介してユーザに通知することを含む、請求項 5 5 に記載の品質レビューを実行する方法。

【請求項 5 7】

プロセッサデバイスによって、ユーザが前記ユーザインターフェースを介して1 つ以上のルールを作成することを可能にすることであって、前記ルールの各々が、プロセス内の1 つ以上の例外を識別する、可能にすることと、前記プロセス内の例外を判断する際に使用するために、前記1 つ以上のルールを前記ルールデータベース内に記憶することと、をさらに含む、請求項 4 4 に記載の品質レビュー管理システム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

関連出願

本出願は、2 0 1 8 年 4 月 1 8 日に出願された米国仮特許出願第 6 2 / 6 5 9 , 3 2 5 号、発明の名称「Quality Review Management System」の 3 5 U . S . C . § 1 1 9 (e) に基づく利益を主張し、その開示全体は、参照により本明細書に明示的に取り込まれる。

【0 0 0 2】

本発明は、概して、製造管理システムに関し、より具体的には、プラント、製造プラントで実施されるプロセス、バッチ、および他の作業の動作に関する高度な品質レビュー手順を可能にする品質レビュー管理システムに関する。

【背景技術】

【0 0 0 3】

加工プラントなどの製造プラントは、典型的には、一式の原材料または前処理済材料から1 つ以上の製品を生産する。多くの場合、生産された製品は特定の品質であるべきであるか、または顧客要件、業界規格、食品医薬品局 (F D A) 要件、安全性要件、表示要件などを満たすためのさまざまな異なる品質規格のうちの1 つに従って製造されねばならな

10

20

30

40

50

い。多くの場合、別個の品質レビュー担当者が、生産された製品が事前確立された一組の規格または要件を満たすこと、または特定の規格に合わせて製造されたことを保証することに関連付けられたさまざまな作業を実施する任務を与えられる。そのような作業には、製品を製造するのに使用される製造機器、材料、またはプロセスのさまざまなパラメータの状態または値をレビューして、適切な製造ステップが実施されたこと、製品が、特定の温度範囲、PH範囲、圧力範囲内など、さまざまな望ましいコンディション下で製造されたことを保証することが含まれ得る。当然のことながら、これらの品質レビュータスクは、製造される製品のタイプ、製品を製造するのに使用される製造プロセスまたは機器、満たされる品質規格などに依存する。そのため、品質レビューは、多くの製造環境で高度に専門化された作業であり、製品、製造プロセス、および品質規格の深い知識が必要である。

10

【0004】

一例として、FDAなど、さまざまな食品および医薬品の業界または組織によって高度に規制されているさまざまなタイプの食品または医薬品を製造するために、典型的には、バッチプロセスが使用される。これらの製品が製造される特定のプロセスステップおよびコンディションは、典型的には、生産された製品の安全性を保証するため、または製品が特定の表示基準を満たすこと（例えば、製品はグルテンフリー、コーシャーなどであること）を保証するために、何らかの方法で設定または規制される。一例として、製品を製造するのに使用されるプロセスステップには、製造プロセス中に原材料または中間材料に特定のプロセスステップが実施されること、製造プロセスの製造ステップ内または製造ステップ間に特定の洗浄手順がプロセス機器に実施されること、温度、圧力、PHバランスなどのさまざまなパラメータが、プロセス全体を通してまたはプロセスの肝心なときに、特定の範囲内または特定の閾値より上または下に維持されることなどが、要求または義務付けられ得る。結果として、多くの場合、製品製造業者は、製造プロセス中の製造機器の実際の動作に関するデータを収集および記憶して、生産された製品が適切な品質規格を満たすこと、例えば、製品が事前確立された手順またはコンディションに従ってまたはそれらの下で生産されたことを、規制当局または顧客に証明できるようにする必要がある。

20

【0005】

食品、石油、医薬品などのさまざまなタイプの製品を製造するのに使用される加工プラントは、概して、複雑で高度に自動化されている。特に、化学、石油精製、食品製造、医薬品製造、または他のプロセスに使用されるものなどの産業加工プラントは、概して、例えば、バルブポジショナ、スイッチ、センサ（温度、圧力、および流量センサなど）、タンク、ミキサ、ヒータなどであり得る1つ以上の現場デバイスに通信可能に結合された集中型または分散型プロセスコントローラを有する、1つ以上のプロセス制御ネットワークを含む。これらの現場デバイスまたは現場機器は、加工プラント内での物理的制御機能（バルブを開閉すること、タンク内の材料を攪拌または混合すること、コンテナを加熱することなど）を実行し得るか、加工プラントの動作を制御するために加工プラント内の測定値をとり得るか、または加工プラント内で任意の他の所望の機能を実行し得る。プロセスコントローラは、歴史的に、例えば、現場デバイスへのまたは現場デバイスからの4~20mA（ミリアンペア）の信号を搬送し得る1つ以上のアナログ信号ラインまたはバスを介して現場デバイスに接続されている。しかしながら、過去数十年において、プロセス制御業界は、プロセスコントローラと現場デバイスとの間の通信を実施するのに使用することができる、Foundation（登録商標）FIELDBUS（以降、「Fieldbus」）、HART（登録商標）、PROFIBUS（登録商標）、WORLDFIP（登録商標）、Device-Net（登録商標）およびCANプロトコルなど、標準、オープン、デジタル、またはデジタルとアナログの混合の多数の通信プロトコルを開発した。一般的に言えば、プロセスコントローラは、1つ以上の現場デバイスが作り出した測定値を示す信号および／または現場デバイスに関する他の情報を受信し、この情報を使用して、典型的には複雑な制御ルーチンを実施し、信号ラインまたはバスを介して現場デバイスへ送信される制御信号を生成し、それにより、加工プラントの動作を制御する。

30

40

50

【0006】

バッチプロセスで使用されるもののような特定のタイプのプロセス制御ネットワークは、典型的には、複数の複製機器セットを含み、各機器セットは、加工プラント内で本質的に同じ基本的機能を実行する同じまたは類似のハードウェアを有するように設計されている。したがって、例えば、クッキー製造プラントは、複数の混合機器セット、複数のベーキング機器セット、および複数の包装機器セットを有し得、個々の混合機器のいくつかまたはすべてを、並行に動作させ、ベーキング機器および包装機器のいくつかまたはすべてと直列に動作するように接続することができる。そのようなシステムでは、同じ全般的制御アルゴリズムまたはルーチンを使用して、任意の特定の複製機器セットの動作を制御して、（特定のバッチレシピによって定義されるような）同じ製品を製造するのが典型的である。したがって、製造される特定の量の特定のタイプの製品を指定または識別する命令によって定義される任意の特定のバッチランは、プラント内のさまざまな異なる機器セットの任意の組み合わせにおいて実施され得る。典型的には、そのような各バッチランまたは命令は、レシピを実施するための多数の異なるステップまたはステージを、第1のステージを終了してから第2のステージを開始するなど、順番に実行する特定の制御手順を含む。したがって、上述のクッキー製造プラントでは、バッチランまたは命令は、混合機器を制御するためのバッチ制御手順を実施し得、次いで、混合機器によって製造された製品にベーキング機器を稼働させる手順を実施し得、次いで、ベーキング機器によって生産された製品を包装するために包装機器を制御する第3の手順を実行し得、各ステップには有限の時間を要する。多くの場合、バッチランでは、各命令の一部として、プラント内のタンクまたは他の容器もしくは機器を洗浄する、空にする、充填するなどの手順も実施される。当然のことながら、各命令は、異なる原材料のセット、原材料に実施される異なるレシピ、異なるフローまたはバッチ手順、さらには異なる品質規格を要求し得る、異なる仕様セットを有し得る。

【0007】

プロセス制御の問題に携わる国際的組織である the International Society for Measurement and Control によって公布された1つのバッチ制御規格は、Batch Control Part 1: Models and Terminologyと題され、多くの場合、ISAS 88.01-1995規格と称されるもの、またはその改訂版の1つである（本明細書では「S88規格」と称される）。S88規格は、自動化バッチプロセスで使用するための機器および手順のモデルを定義し、それらのモデルおよびそれらの要素を参照する際に使用するための特定の用語を定義している。例えば、S88規格では、「バッチプロセス」を、1つ以上の機器類を使用して、有限の時間で、一定量の入力材料を、順序付けられた一組の処理作業に適用することにより、有限量の物質の生産につながるプロセスとして定義している。別の例として、「バッチ」は、S88規格によって、バッチプロセスの1回の実行によって生産される、または生産された物質と定義されている。

【0008】

上記したように、バッチ処理機器（例えば、バルブ、ヒータ、ミキサなどの制御可能な要素）は、バッチを製造するための事前定義された手順に従って、バッチプロセスまたはバッチラン中に動作させられる。そのようなバッチ処理機器はすべて、本明細書では、機器、機器モジュール、処理機器、および／または物理的要素と、同義的に称される。そのような物理的要素を動作させる手順は、S88規格では、「手順モデル」と称されることが多い。S88規格によれば、手順モデルは、手順の階層的なランク付けとして構造化され、最高レベルが下位レベルの各々を包含し、次に高いレベルがその下のレベルの各々を包含する、などとなる。特に関心対象のS88手順モデルのレベルには、降順で、「手順」、「単位手順」、「動作」、および「フェーズ」が含まれる。「手順要素」またはバッチ下位手順という用語は、本明細書では、S88手順モデルのこれらのレベルのうちのいずれかの任意の実施形態または実装形態、および一組のバッチ手順の任意の他の階層定義を指すために使用される。

【0009】

上記のように、関心対象の最高レベルのS88手順要素は、手順と称され、手順は、1つ以上の単位手順で編成される。各単位手順は、ひいては、1つ以上の動作で編成され、または編成することができ、動作は各々、ひいては、1つ以上のフェーズで編成することができる。さらに、S88手順モデルは、特定のアプリケーションにおける他の階層レベルの定義および使用を妨げるものではない。むしろ、S88規格と本明細書で言及される手順要素とは、自動化バッチプロセス制御において追従される手順を説明するための幅広い標準化モデルを提供することを意図としており、これらの要素は、S88規格によって定義される4つの手順要素に限定されない。

【0010】

バッチの異なる手順要素は、概して、実際には、パーソナルコンピュータ、ワークステーション、およびプログラマブルコントローラを含む、データ処理デバイスによって、およびその内部で実行されるコンピュータプログラムとして実装される。典型的な手順要素を実行すると、典型的には、データ処理デバイスの出力を物理的要素に直接、またはローカルエリアまたはワイドエリアネットワークを介して間接的に接続することにより、データ処理デバイスから、物理的要素を制御するために使用できる電氣的または光学的出力がもたらされる。手順要素は、少なくとも1つの物理的要素に関する「基本的制御」を呼び出すことにより、割り当てられたタスクまたは関連付けられたタスクを実行する。このタイプの制御は、典型的には、物理的要素の特定の所望の状態を確立および維持することに専用化される。基本的制御には、例えば、保存容器要素内の材料の流れを開始または維持すること、ポリ塩化ビニル反応器要素内の出発材料を加熱することなどが含まれる。実際には、手順モデルの下位レベル（すなわち、フェーズ）は、実際の物理的要素との実際の通信を実行し、それにより、基本的制御を呼び出すまたは実行する。手順モデルの上位レベルは、本質的に、手順モデルの組織および構造、ならびに物理的モデルを改善するための抽象体である。

【0011】

さらに、多くのバッチシステムは、バッチエグゼクティブを使用して、使用されている手順モデルに従ってプラント内の1つ以上のバッチの動作を制御する。バッチエグゼクティブは、バッチプロセスまたは作業の状態を記述する論理構造としての状態マシンモデルであるか、またはそれを使用し得る。状態マシンモデルは、多数のプロセス状態を、それらの状態間の遷移を引き起こすアクションと一緒に、記述または定義する。プロセスの状態マシンモデルは、その状態への先の遷移により、特定の状態にあると言われる。特定のイベントが発生するか、または特定のステータスが検知されると、状態マシンモデルは、特定のイベントまたは検知されたステータスに対応する別の状態へ遷移する。状態マシンモデルは、バッチプロセスの手順要素の動作を定義および実装するための有用な技法である。特に、状態マシンとして定義および実装された手順要素は、例えば、その関連付けられた状態マシンが古い状態から新しい状態に遷移したときに、アクションを開始する。

【0012】

当然のことながら、S88規格は、標準の状態マシンモデルに従う手順要素の定義および実装を可能とする。S88規格はこのアプローチを義務付けてはいないが、このアプローチは、プロセス制御業界で広く採用されており、さまざまな業者の製品間で高度な相互運用性を可能にしている。状態マシンモデルに従って定義および実装された手順要素を有するS88規格の1つの現在の商用アプリケーションは、Emerson Process ManagementのDelta V（商標）Batch製品である。Delta V（商標）Batchでは、サーバプログラムまたはバッチエグゼクティブプログラムが、さまざまな手順要素を実行するデータ処理デバイス上で稼働される。（「バッチエグゼクティブ」と称される）サーバまたはバッチエグゼクティブプログラムは、1つ以上の状態マシンモデルに従って手順要素の実行を調整して、手順、対応する単位手順、対応する動作、および対応するフェーズが、サーバプログラムによって、それぞれのステップを通して順序付けられるようにする。いずれの場合においても、フェーズがサーバプログラムによっ

10

20

30

40

50

て開始されたときなど、命令に関連付けられた特定のバッチランまたは特定のバッチプロセスの実施中に、フェーズは、プログラマブルコントローラ内のフェーズロジックインターフェースに開始要求を伝える。次いで、プログラマブルコントローラは、フェーズに対する実際の状態ロジックまたは制御ルーチンを実行し、プロセス機器への通信を介して必要なプロセス制御を提供する。

【0013】

プロセスの品質レビューに関する前述の説明から理解されるように、命令に対するバッチランが、適用可能な品質規格に従って満足の行くように動作または実行されたかどうかを判断することができるように、バッチランの処理を編成する履歴イベントを含め、バッチプロセスの動作を表すデータを収集することが望ましい。そのような履歴データは、品質レビューの目的だけでなく、品質制御の傾向を判断すること、またはバッチプロセスで使用される機器がいつ点検を必要とするかを判断することにも有用であり得る。多数のタイプのデータが、バッチプロセスの品質および進行をレビューする際に潜在的に有用である。1つのそのようなデータソースは、バッチの処理中にバッチプロセス内のさまざまなデータポイントによって生成される連続データである。データポイントは、バッチプロセスのいくつかの制御値または他のステータスもしくは測定値を反映する、単一の、そのような連続データのソースである。例えば、センサによって測定されたような、特定のレベルの材料流量または材料温度は、そのようなデータポイントであり得る。現在の制御バルブの設定、サンプルが採取された時間などは、他のデータポイントであり得る。そのような各データポイントは、それらに関連付けられたバッチプロセスアプリケーションによって経時的に検知または制御される、一連の連続的データ値を有し得る。バッチの処理中に生成されるすべてのそのような連続データの集計は、バッチ処理システムによってログに記録され、バッチデータベース内のバッチログファイルの一部として記憶されることがよくある。これらのバッチログレコードは、通常、タイムスタンプと現在値とを、データソースを識別するためのタグなど、データポイントの他の識別情報と共に含む。

【0014】

バッチプロセスの品質および進行をレビューする際に有用な別のタイプのデータは、手順モデルの実行に関してバッチプロセスを記述する情報（例えば、バッチプロセスの状態、バッチモデルの状態間の遷移など）に関するかまたはこれを含む、バッチ状態およびイベント情報である。例えば、手順モデルの特定のフェーズまたは特定の動作、単位手順もしくは手順の開始時間および終了時間を記述するバッチイベントは、イベント情報を構成する。イベント情報には、バッチプロセスの物理的要素によってまたはオペレータによって生成される情報を含む、プロセスイベントも含まれる。特に、プロセスの各機器モジュール、セルなどは、特定のフェーズの開始、停止、または稼働（つまり、特定の基本的制御アクションの実行）における1つ以上の特定の作業を示すプロセスイベントを生成し得る。プロセス機器によって認識されるアラームおよびイベントコンディションは、プロセスイベントのさらなる例である。プロセスイベントには、バッチプロセスの動作中に行われたバッチプロセスへのオペレータ変更に関する情報も含まれ得る。

【0015】

さらにその上、多くのバッチプロセスシステムは、ワークステーションなどの専用オペレータインターフェースデバイス上で動作するオペレータインターフェースプログラムを利用して、オペレータが、バッチランの現在の状態を閲覧すること、バッチラン内で手動のステップをとること、例えば、さまざまなパラメータを手動で設定すること、さまざまなバッチ手順、動作、フェーズなどの実行のためにプロセス機器を選択すること、バッチランまたは命令中の予想外の作業またはコンディションに関してメモすることなどを可能にする。1つのそのようなオペレータインターフェースプログラムは、処理されている各命令に対するバッチランを実施するのに使用されるバッチエグゼクティブまたはプロセス機器（例えば、プロセスコントローラ）からさまざまな情報を傍受または受信する、（Emerson Automation Solutionsが提供するSyncade Workflowプログラムなどの）ワークフロープログラムとして知られている。ワーク

フロープログラムは、バッチエグゼクティブまたはプロセス制御システムからのさまざまなタイプのデータをサブスクライブし、このデータをオペレータに提示して、オペレータが、バッチランの現在の状態を閲覧すること、バッチモデルの状態間の遷移を閲覧すること、バッチランに変更を行うこと、さまざまなバッチ手順、動作、フェーズ、または機器を手動で停止および開始させること、ならびにバッチラン内で発生した問題または予想外のイベントに対応することを可能にし得る。例えば、多くの場合、バッチ機器または原材料が予想外に利用不可能である、プロセスパラメータが予想範囲外または所望範囲外であるなど、予想外のエラーまたは問題がバッチラン中に発生する可能性がある。これらの場合、ワークフロープログラムは、プロセス内のアラームまたはイベントに対処するために、バッチステップまたは手順をスキップする、他の機器または材料を使用に指定する、加工プラント変数、設定点、またはパラメータを変更するなど、オペレータがどのように進行させるかについて決定を行い、実施することを可能にする。一般的に言えば、ワークフロープログラムは、バッチログファイル内に、オペレータのアクションを示すデータを、バッチログファイル内に記憶されている生バッチデータと共に記憶する。多くの場合、ワークフロープログラムはまた、オペレータが、アクション時にバッチで何が起こっていたか、オペレータがなぜ特定のアクションをとったかなどのメモを作成すること、または説明を記憶することを可能にし、これらのメモもバッチログファイルに記憶される。

【0016】

いずれの場合においても、多くの業界で、バッチランが発生し、製品または命令が完了した後、品質管理技師または他の品質管理担当者（本明細書では概して、品質管理技師と称される）が、製品が適切な品質規格を満たしていることを検証しなければならない。このレビューを実行するために、品質管理技師は、（典型的には、命令ごとに）命令のバッチランのバッチログファイルにアクセスし、閲覧して、予想されたパラメータ、ワークフロー、および範囲に従って、またはそれらの内で、バッチランが完了したことを保証する。概して、このレビューを実行するには、品質管理技師は、バッチデータファイル内の生データをスクロールして、バッチランの予想された動作に対する「例外」を探さなければならない。「例外」という用語は、本明細書では概して、製造プロセスまたはプラントにおいて、予想される手順、プロセス、ステップ、ワークフロー、範囲、値などからの逸脱を示すために使用される。例外は、例えば、1つ以上のプロセス変数もしくはパラメータが、予想範囲外もしくは所望範囲外であるか、または予想閾値もしくは所望閾値を上回るもしくは下回ること、バッチ手順、動作もしくはフェーズが予想されたシーケンスをスキップするもしくは実行しないこと、バッチ手順、動作もしくはフェーズが、予想外の時間に停止もしくは開始するか、または予想動作時間より長いもしくは短い時間がかかること、異なるもしくは予想外の材料がバッチ手順で使用されること、追加ステップもしくはアクションがバッチラン中にオペレータによって実施されること、レシピへの変更、バッチラン中にオペレータによってメモが生成されること、バッチラン中に発生したアラームおよびイベントなどであり得る。

【0017】

いずれの場合においても、品質管理技師は、バッチログファイル内に記憶された生データに基づいて例外を識別し、次いで、そのような各例外の影響または重大度、および例外を取り扱うまたは「クローズ」する任意の手順をとったらどうなるかを判断しなければならない。例えば、多くの場合、例外は、予想されるバッチランにおいて、実施されている品質規格に従って生産された製品の品質をまったくまたは著しく低下させない、最小または取るに足らない偏差を呈するのみであり得る。他の場合では、例外は、公認機関によるまたは顧客による後のレビューのために文書化される必要があり得るが、それらの影響は、製品の品質の著しい低下をもたらすほど重大ではない場合がある。さらに他の場合では、例外は、製品に1つ以上の追加手順を行って、所望の品質を保証するか、または製品に1つ以上のテストを実行して、製品の品質を保証する必要があるほど著しいまたは重大である場合がある。さらに他の場合では、品質管理技師は、例外の詳細に基づいて、製品を廃棄する必要がある、または製品を、より低品質または表示規格で販売する必要があると

判断する場合がある。

【0018】

バッチログファイルを調べて例外を探すプロセスは、非常に面倒で、問題が山積している。特に、バッチログファイル内のほとんどのデータは、例外を示すものではないので、品質管理技師は、かなりの（通常、ほとんどの）レビュー時間を、実際には任意の例外に関連付けられていないデータを探すことに費やす。さらにその上、品質管理技師は、例えば、重要なプロセス変数の所望の値または範囲、バッチ手順の予想される順序、さまざまな異なるバッチ手順の予想される停止および開始時間などを含む、例外につながるすべてのコンディションに留意しなければならない。バッチログファイル内に記憶されているバッチプロセスアラームおよびアラート、またはオペレータからのメモの存在は、例外の存在を示す可能性があるが、これは必ずしもそうであるとは限らず、当然のことながら、発生したすべての例外が、オペレータメモと共に文書化され得るか、または関連付けられたアラームまたはアラートが制御システムにおいて生成されるレベルに達し得るわけではない。理解されるように、決定される実際の例外は、したがって、レビューを実行する特定の品質管理技師のスキルと、満たされる特定の品質規格と、に依存する。

10

【0019】

さらにその上、品質管理技師が例外（例えば、プロセス値が範囲外など）を見つけた場合、技師は、通常、例外の重大度を判断するために、例外時のバッチのコンテキストと、例外に最善に応答するまたは取り扱う方法とを決定しなければならない。このプロセスは、例外時のバッチランまたはバッチ手順の状態についての追加データ、例外時の他のプロセス変数の値を取得すること、例外時にアラームまたはイベントが同じまたは他のプロセス機器で生成されたかどうかを判断することなどを含み得る。したがって、このプロセスでは、概して、品質管理技師が他の利用可能なデータアクセスシステムを使用して、例外時の追加プロセス状態情報を決定する必要がある。このデータ収集作業は、時間がかかる可能性があり、バッチ品質管理技師の側に追加ノウハウが必要である。

20

【0020】

例えば、バッチプロセスは、異なる設定値、設定などを使用して、異なる時間に、プラント内の異なる機器で稼働し得る、さまざまな異なる手順要素を有するので、バッチログファイルからバッチの動作を閲覧するために、特定の時間にバッチプロセスのスナップショットを取得し、そのデータを品質管理技師に表示することは簡単なタスクではない。代わりに、バッチランを閲覧するために、品質管理技師は、バッチの手順イベント（つまり、バッチに関連付けられた下位手順およびサブプロセス）に関する、さまざまな時間のバッチからのデータをレビューおよび分析し、それにより、例外時のバッチランの動作を理解することができる。さまざまなバッチデータは、典型的には、バッチランの動作中に自動的に収集および記憶されるが、これらの異なるタイプのデータは、概して、異なるサブシステムによって収集され、実際には、異なるデータベース内に記憶される場合がある。このことで、品質管理技師が任意の特定のバッチプロセスを包括的に閲覧することは困難になる。例えば、バッチプロセス内のセンサ、バルブなどの実際の現場デバイスから取得されるアラームデータおよびセンサ測定データなどのデータは、典型的には、タイムスタンプ付きデータとしてデータヒストリアンに記憶され、概してデータが収集された時間に基づいて、データヒストリアンから入手可能である。しかしながら、バッチエグゼクティブルーチンに関連付けられているものなどの異なるデータベースには、バッチランおよびバッチラン内のさまざまな異なる下位手順または手順要素の開始時間および終了時間が記憶され得る。これにより、品質管理技師が、バッチプロセスの他のさまざまなサブシステムにおいて、著しいデータ検索を行わずに、例外のコンテキストを理解することがより困難になる。

30

40

【発明の概要】

【0021】

（例外によるレビュー（RBE）システムとも称される）品質レビュー管理システムは、プロセス制御システム、ワークフローアプリケーション、バッチプロセスオペレータア

50

アプリケーション、バッチエグゼクティブアプリケーションなどのデータソースなど、他のさまざまな加工プラントデータソースによって収集されるような、1つ以上の製造プロセスの動作を分析するために使用され得る。品質レビュー管理システムは、これらのプロセス内の例外を自動的に検出し、検出された例外に関するデータを、整理され、簡単にレビューできる方法で記憶する。品質レビュー管理システムはまた、品質レビュー管理者または技師が、プロセスに関連付けられた例外を、使い易いインターフェースでレビューおよび取り扱う（解決する）ことを可能にする。

【0022】

より具体的には、品質レビュー管理システムは、バッチプロセスなどのプロセス内の1つ以上の例外を自動的に識別するために使用し得るルールを作成し、ルールデータベース内に記憶することを可能にするコンフィグレーションシステムを含む。ルールは、例外を識別するために満たさなければならない一組のコンディション、識別された例外の重大度、例外についての情報と、例外を取り扱う、解決する、またはクローズするのに使用されるプロセスまたは手順、品質レビュー技師が例外を理解および解決するのを支援するために例外レコードの一部として記憶されるべきプロセスデータまたはその他のデータなどを識別し得る。品質レビュー管理システムはまた、ルールデータベースのルールを使用して、プラントまたはプロセス制御システム、バッチエグゼクティブ、オペレータワークフローアプリケーションなどのデータソースによって定期的に提供されるデータメッセージなどのデータメッセージ内のデータを処理して、データメッセージ内のデータが、いずれかのルールによって定義されるような、例外を含むかどうかを判断する、例外エンジンも含む。例外エンジンは、例外を識別すると、例外を解決または取り扱うためにとる手順またはステップの名称、タイプ、重大度、および例外が発生した時間の、例外が発生したプロセスの動作に関するプロセスデータなど、例外についての情報を例外レコードとして記憶し得る。例外エンジンは、識別された各例外に対するこのデータおよび／または他のデータを、各命令に対する例外レコードとして、例外ファイル（例えば、例外データベース）内に記憶し得る。さらに、例外エンジンは、命令を実施しているプロセスの動作中にリアルタイムで動作して、命令に対する例外ファイルまたはデータベースを作成し、例外が発生した、または発生しようとしているというライブフィードバックをプロセスオペレータに提供し得る。

【0023】

したがって、1つの実施例では、品質レビューシステムは、ユーザが、例外エンジン内で実行して例外を定義または検出するためのコンフィグレーション可能な例外ルールを作成することを可能にする、コンフィグレーション環境を提供する。コンフィグレーション環境は、例外を検出するためのさまざまなタイプの入力データを処理し得る一組のルールテンプレートを記憶し得る。コンフィグレーション環境では、ユーザは、テンプレートのうちの1つを選択し、入力／出力を定義し、データセットに対して分析されるべき考えられる式を定義し得る。データがエンジンを通過するとき、設定されたルールに対してデータを分析し、いずれかのルール基準が満たされた場合、例外が作成される。次いで、例外は、すぐにチェックされ得る。例外製品ごとの典型的な既知のレビューは、システムに対して定義された例外を製品にハードコードする例外ソフトウェアプロバイダを必要とし、したがって、新しい例外を作成するにはソフトウェアまたはシステムの更新を必要とする。これらの更新は、ユーザが取得してインストールするのに時間がかかり、ユーザが自身のシステムをカスタマイズすることはできない。

【0024】

品質レビュー管理システムはまた、品質レビュー技師または他の担当者が、例外エンジンによって識別されたような、かつ命令に対する例外レコードとして例外データベース内に記憶されたような例外をレビューして対処することを可能にするレビューインターフェースも含む。特に、レビューインターフェースは、特定の命令に対して例外データベースにアクセスし、識別された各例外に関する情報を、品質レビュー技師に、整理され、理解し易い方法で提示し得る。レビューインターフェースは、品質レビュー技師に、各例外レ

コードに対する名称、タイプ、および重大度識別など、各例外レコードに対して記憶されたデータを提供し得る。レビューインターフェースは、レビュー者に、例外のタイプなど、例外レコードによって識別されたような例外についての記憶された情報、および例外が発生した時間に存在した、さまざまなプロセスまたはバッチ変数、状態、手順などに関するデータを提供し得る。この情報は、レビュー者が、例外、例外が発生した時間のプロセスのコンテキスト、および例外の重大度を容易に理解することを可能にする。さらにその上、レビューインターフェースは、レビュー者が各例外に関してさまざまなアクションをとることを可能にし得る。このようなアクションには、例外を認識すること、例外を承認またはクローズすること、なぜ例外を無視または承認することができるかなど、例外にさらなる情報で注釈を付けること、例外レコードを別の人へ送信することなどが含まれ得る。レビューインターフェースはまた、レビュー者にアクション案を提供し得、場合によっては、例外を解決またはクローズすることに関してとられるべき、特定のアクションを執行し得る。そのようなアクションには、例えば、特定の数またはタイプの人々（例えば、レビュー者のみ、レビュー者と管理者、2人のレビュー者など）による承認を要求することが含まれ得る。他のアクションには、顧客、品質レビュー機関などにより後に使用するための、例外レコードと共に記憶される、注釈、プロセス情報など、特定の情報がレビュー者から提供されることを要求することが含まれ得る。

【0025】

レビューインターフェースは、処理された例外レコードをデータベースまたはファイル内に記憶し得、レビュー者が例外データベース内の例外レコードをスクロールして、命令に対するすべての例外レコードが処理またはクローズされたかどうかおよびそれはいつであるか、レビューのステータス（例えば、例外レコードデータベース内にいくつかの未処理の例外が存在するか）、例外レコードに関する統計データ（例えば、ファイル内に各タイプまたは重大度の例外がいくつか存在するか）を決定することを可能にし得る。レビューインターフェースはまた、レビュー者が例外レコードを、タイプごと、重大度ごと、処理ステータスごと（例えば、レビューされていない、レビュー済みだがクローズされていない、クローズ済みなど）など、さまざまな方法でグループ化することを可能にし得る。さらにその上、レビューインターフェースは、レビュー者が、グループ内のすべての例外レコードをクローズすること、グループ内のすべての例外レコードに注釈を付けることなど、グループ全体の例外レコードに対して1つ以上のアクションをとることを可能にし得る。

【0026】

このようにして、品質レビュー管理システムは、バッチプロセスまたは他のプロセスの動作中にセットアップ（設定）および実行されて、例外を、プロセスラン中に発生したときに自動的に識別し得、プロセスオペレータに、例外が発生したこと、またはプロセスオペレータによってとられたアクションが例外となろうことを通知し得、特定のプロセス（例えば、バッチプロセス）に対して決定されたすべての例外を例外レコードデータベースに記憶し得る。品質レビュー管理システムはまた、品質レビュー技師が、現在必要なように、大量の生データをレビューする必要なく、例外データベース内の例外を迅速に決定、閲覧、および取り扱いすることを可能にする。同様に、品質レビュー管理システムは、さまざまな異なるタイプまたは重大度の例外が特定の方法で処理または取り扱いされることを保証するなど、例外取り扱いまたは解決プロセスにおいてさまざまなルールを執行し、それにより、品質レビュープロセスに一貫性を提供する。

【0027】

さらにその上、品質レビュー管理システムは、システムが、第三者のシステムまたはアプリケーションとインターフェースし、それらを例外について分析するために使用されることを可能にするプラグイン環境において実施され得る。1つの場合では、プラグインは、第三者のアプリケーションまたはデータソースから特定のデータを取得するように作成または構成され得、次いで、このデータを、1つ以上の例外ルールによって処理するために、例外エンジンに渡し得る。1つの実施例として、品質レビュー管理システムは、OPCインターフェースを使用するものなどのイベント監視システムとインターフェースして

、第三者のプロセス制御システムまたはプラントからデータを取得し、分析するために使用され得る。

【0028】

1つの実施例では、プラグインモジュールは、RBE例外エンジンを外部または第三者のシステム（例えば、制御システム、OPCインターフェース、保守管理システムなど）とインターフェースさせ、それにより、例外が、第三者または外部のソフトウェアシステムからのデータに全体的または部分的に基づいて生成されることを可能にするように作成される。プラグインは、第三者または外部のシステム内のまたはそれからの特定のデータにアクセスするように、およびこのデータをRBEエンジンに渡して、例外が作成されるべきかどうかを分析するように作成され得る。プラグインは、第三者のサーバ内、RBEサーバ内、または別のサーバ内で稼働され得、例外検出を実行してもしなくてもよい。したがって、プラグインは、単純な第三者のシステムデータ収集体とすることができ、またはデータを収集して追加的にいくつかの例外処理を実行し、例外または前処理済みの例外データをRBEエンジンに提供してもよい。

10

【0029】

既知のレビューシステムは、現在、制御システム、安全システム、保守管理システムなど、外部ソフトウェアプログラムとインターフェースすること、またはそれらから直接提供されたデータに基づいて例外を作成することはできない。プラグインにより、RBEシステムは、ランタイム中に動作している複数の異なるソフトウェアシステムを有する複雑なプロセス環境において、ずっと堅牢となる。

20

【0030】

さらにその上、1つの実施例では、RBEシステムは、RBE入力データ（すなわち、例外を定義または生成するのに使用されるプロセスデータ）を収集するときに、（例えば、プラグインを作成するときに事前定義され得る）一組のメタデータを収集するライブレビューコンポーネントを含み得る。メタデータは、任意の他のプロセスまたは環境データであり得、例外入力データと同時に収集される。このメタデータは、アクセスされた例外データに関連付けられたプロセススナップショットとして記憶される。RBEシステムが、収集された例外データに基づいて例外を検出したとき、RBEシステムはメタデータも例外の一部として記憶する。RBEシステムは、生成された例外を報告するときに、メタデータの一部またはすべてを追加的に表示し、それにより、ユーザまたはレビュー者に、プロセス内に例外が発生した時間のプロセス内のライブレビューを提供する。このシステムは、ユーザに、例外時のさまざまなプロセス状態／値などのクイックビューを提供し、ユーザが、なぜ例外が発生したのか、および例外の重大度をより迅速に理解することを可能にし、したがって、ユーザが例外をより迅速に処理することを可能にする。

30

【0031】

既知のプロセスレビュー製品は、生成された例外のリストをユーザまたはレビュー者に提供するが、レビュー者は、なおも各例外を分析して、例外を棄却することができるかどうか、または何らかの他の方法で対処しなければならないかを判断しなければならない。しかしながら、一般的に言えば、レビュー者は、例外をもたらしたデータが収集されたプロセスシステムに戻って、例外時のプロセスについて他のこと（例えば、プロセスはオンラインまたはオフラインだったか、例外時のバッチの状態など）を理解しなければならない。ライブレビュー機能により、例外レビューを実行する際にレビュー者にとって最も有用なプロセスデータを一般的に含むプロセススナップショットをユーザに自動的に提供でき、それは、レビュー者が、他のデータ収集システムに戻って、例外レビューを実行するためにユーザが必要とする情報を得る必要がないことを意味する。この機能はまた、レビュー者が、データが収集される他のプロセスシステムを使用してレビュー者が必要とするプロセスデータにアクセスする必要がないので、レビュー者は、概して、それらのシステムに精通している必要がないことも意味する。

40

【0032】

別の実施例では、RBEまたは品質レビューシステムは、第三者または外部のシステム

50

に結び付けられ得、メッセージを生成してRBEシステムに送信するイベントモニタを含み、次いで、RBEシステムは、これらのメッセージを使用して、例外処理または例外作成を実行できる。1つの場合では、イベントモニタは、OPCアラートおよびアラーム監視インターフェースに結び付けまたは結合され得、1つ以上のアラートまたはアラームがOPCインターフェースにいつ渡されたかを認識し得る。この時点で、イベントモニタは、アラートまたはアラーム情報と、恐らくは任意の他の所望のOPC収集データとを用いてメッセージを作成し、RBEシステムに届ける。次いで、RBEシステムは、メッセージを分析して、アラームおよびアラートに基づいて例外を生成する。典型的なレビューシステムは、第三者システムからのデータを分析せず、それらのいずれも、存在するOPCアラートおよびアラームインターフェースに接続されていない。そのため、これらのシステムは、例えば、プロセスのアラームおよびアラートに基づいて例外を生成することはできない。

【0033】

同様に、品質レビュー管理システムは、例外レコードのリストをレビュー者に電子ディスプレイまたはユーザインターフェースを介して提示するときに、新しいページングアルゴリズムを使用し得る。レビュープロセスにおける欠落レコードまたは重複レコードの提示を削減または排除する方法で動作するこのページングアルゴリズムは、インターフェースディスプレイに提供されている現在の表示ページのレコード内の実際のレコードを追跡するための1つ以上のアンカを記憶する。次いで、システムは、現在表示されているページに対するアンカを使用して、次のページの一部として表示する次のセットまたはレコードの開始位置を決定し、これは、ページングシステムが、レビュープロセス中に表示されるべきレコードリストにレコードが追加されても、またはそこからレコードが削除されても、適切に動作することを可能にする。

【0034】

1つの実施例では、レビューシステムは、生成された例外のリストをレビュー者に提示するときに、新しいページングアルゴリズムを実施する。概して、RBEシステムは、例外を作成／検出し、これらの例外を、SQLデータベースなどの構造化データベース内にレコードとして記憶する。次いで、例外のリストをレビュー者に対して表示画面に提示するとき（そして、多数の例外またはレコードが存在する可能性があるとき）、RBEシステムは、データベースを検索して、すべての関連例外レコードを探し、これらのレコードへのリンクのページをダウンロードし、次いで、ユーザがさまざまな例外レコードページをスクロールするときに、読み出したページの部分内のレコードをデータベースからダウンロードした順序で提示する。しかしながら、レビュー者が例外を処理し、ユーザインターフェース上に表示されているような例外リストをスクロールダウンするとき、レコードがデータベース内で変更され得、したがって、ページデータが古くなり得る。特に、レコードは、追加されることもあり得、またはデータベースから削除されることもあり得る。この場合、システムがレコードを表示画面に提示しようとするとき、ページ内で参照されているさまざまなレコードが欠落し、見て分かるデータの損失または重複を引き起こし得る。場合によっては、ユーザは、特定の回収されたページ上のレコードが削除されたことに起因して、あるページから次のページに移るレコードを見落とし得る。他の場合では、システムは、ダウンロードされたページ内にあった欠落レコードに基づいて、新しい表示ページをどこから開始するかについて、混乱する可能性があり得る。新しいシステムは、画面上に表示された最後のレコードから出発して、表示画面上に提示されたレコードをダウンロードするという点で、異なった動作を行う。つまり、元の回収されたレコードページに関連付けられた新しい表示ページを提示する代わりに、システムは、データベースにアクセスして、表示されたページ上の最後のレコードの直後に現在存在する一組のレコードを探し、それにより、新しい一組のレコードが画面上に表示される都度、表示された最初の新しいレコードは、スクロール時点の直前に表示されていた最後のレコードの直後のレコードであることが保証される。

【0035】

この表示システムは、ユーザが長時間にわたってさまざまなレコードページをスクロールダウンするレビュープロセスにおいて、レコードが順不同で表示され、または欠落されないことを保証するので、より正確である。さらに、RBEシステムは、ユーザがレコードをスクロールしているときに、新しい表示ページに収まる新しいレコードの数に関して、データベースにアクセスすることのみを必要とするので、より効率的であり、一方、ユーザがスクロールプロセスを実行するときに、関連レコードのすべてがアクセスされることをなおも保証する。

【図面の簡単な説明】

【0036】

【図1】オペレータがバッチプロセスを管理し、品質レビューのためにバッチログファイルを記憶することを可能にする、プロセス制御ネットワーク、バッチエグゼクティブ、およびワークフローアプリケーションを含む典型的なプロセスまたは製造プラントのブロック図である。

10

【図2】図1と同様であるが、プラント環境内で例外ベースの検出および品質レビューを実行する際に使用するための、コンフィグレーションアプリケーション、例外エンジン、および品質レビューインターフェースアプリケーションを有する品質レビュー管理システムを含む、プラント環境のブロック図である。

【図3】ユーザが例外エンジンによる使用のための1つ以上の例外ルールを作成することを可能にする、図2の品質レビュー管理システムのコンフィグレーションアプリケーションによって作成される例示的画面表示である。

20

【図4】例外処理のために複数のデータソースからのデータを分析するときの、図2の品質レビュー管理システムの例外エンジンの動作を示すフローチャートである。

【図5】品質レビュー管理システムからバッチプロセスのオペレータへの、プロセス内の実際のまたは潜在的な例外の検出をオペレータにリアルタイムで通知するライブフィードバックを示す例示的画面表示である。

【図6】プラグイン環境で実施された品質レビュー管理システムの動作を示す図である。

【図7】継続的プロセス環境で品質レビュー管理システムを使用して、第三者のアプリケーションおよびシステムの監視を実行する例示的イベント監視システムを示す。

【図8A】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図2の品質レビュー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

30

【図8B】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図2の品質レビュー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

【図8C】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図2の品質レビュー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

40

【図8D】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図2の品質レビュー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

【図8E】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図2の品質レビュー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

【図8F】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図2の品質レビュー

50

ー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

【図 8 G】品質レビュー者が例外ルールコンフィグレーションならびに例外レコード閲覧および取り扱いに関するさまざまな作業を実行することを可能にする、図 2 の品質レビュー管理システムのレビューインターフェースアプリケーションまたはコンフィグレーションアプリケーションによって作成される例示的画面表示を示す。

【図 9 A】例えばウェブ環境においてユーザにレコードのリストを提供するために使用される典型的なページング技術の動作を示す図である。

【図 9 B】例えばウェブ環境においてユーザにレコードのリストを提供するために使用される典型的なページング技術の動作を示す図である。

【図 10 A】一組のレコードにページングを実行するときに重複レコードまたは欠落レコードを削減または排除するための、図 2 の品質レビューインターフェースアプリケーションによって実施され得る新しいページング技術の動作を示す図である。

【図 10 B】一組のレコードにページングを実行するときに重複レコードまたは欠落レコードを削減または排除するための、図 2 の品質レビューインターフェースアプリケーションによって実施され得る新しいページング技術の動作を示す図である。

【発明を実施するための形態】

【0037】

背景として、図 1 は、典型的なまたは既存の品質レビュー管理アクションが実施される典型的なプラントまたはプロセス制御環境 10 を示す。プラント環境 10 は、通信可能にまとめて接続され、すべてが、バッチプロセスまたは連続プロセスなどの何らかの種類の製造プロセスを動作または実施するようにプログラムされている、コントローラ、現場デバイス（例えば、バルブ、センサ、送信器、タンク、ミキサなど）、I/O デバイスなどのさまざまなプロセス制御デバイスを含み得るプロセス制御ネットワーク 20 を含む。さらにその上、プラント環境 10 は、プロセス制御ネットワーク 20 とインターフェースして、プロセス制御ネットワーク 20 またはプロセス制御ネットワーク 20 内のデバイスに、各命令に関連付けられたバッチプロセスの単位手順、手順、動作、フェーズ、ステップなどのさまざまなバッチの異なるステージを実施することを含む、さまざまな命令またはバッチランを実施することなど、製造プロセスに関連付けられたさまざまなアクションをとらせ得る、（ワークステーションまたはサーバなどのコンピューティングデバイスのプロセッサ上に実装される）バッチエグゼクティブ 22 を含み得る。さらにその上、プロセス制御ネットワーク 20 およびバッチエグゼクティブ 22 の両方は、例えば、プラントの床または製造スペースから離れたビジネス環境に配置され得るさらなるネットワーク 24 に結合され得る。ネットワーク 24 は、コンピュータディスプレイデバイスまたはワークステーション 26 および 28 などのさまざまなデバイス、さまざまなデータベース 30、31 および 32、ならびに、そう所望であれば、（図 1 には示されていない）インターネットまたは他の通信ネットワークへのゲートウェイ、サーバ、ワイヤレスまたはハンドヘルドデバイスなどの他のコンピュータデバイスを通信可能に接続する通信バックボーン 25 を含み得る。デバイス 26、28、30、31、および 32 は、イーサネットネットワーク、インターネットまたは他のローカルエリアネットワーク（LAN）もしくはワイドエリアネットワーク（WAN）などの任意の所望の物理的通信ネットワークまたはバックボーン 25 を介して通信可能に接続され得る。同様に、プロセス制御システム 20 およびバッチエグゼクティブ 22 は、ネットワークバックボーン 25 を介して直接、またはそう所望であれば他の介在ネットワークを介して、さまざまなデバイスおよびデータベース 26、28、30、31、および 32 に接続され得る。

【0038】

この従来技術のシステムでは、ワークフローアプリケーション 40 は、オペレータインターフェースデバイス（ワークステーション 26、例えば）内に記憶され得、ワークフローアプリケーション 40 は、バッチエグゼクティブ 22 がプロセス制御ネットワーク 20 を制御して、さまざまな命令またはバッチランを実施しているときに、ワークステーショ

10

20

30

40

50

ン 2 6 のプロセッサによって実行されて、バッチエグゼクティブ 2 2 とインターフェースし得る。例えば、Emerson Process Management が販売する Syncade Workflow アプリケーションであり得るワークフローアプリケーション 4 0 は、バッチエグゼクティブ 2 2 からデータを取得して表示するように動作し得る。このデータには、バッチエグゼクティブ 2 2 によって取得されるような、進行中のバッチのステータスデータ、パラメータデータ、測定値データ、進行データ、バッチ手順モデルの状態データなどが含まれ得る。この情報またはデータには、実施されている命令、命令に関連付けられた材料およびプロセス（例えば、レシピおよび実施されるべきバッチ手順モデル）、バッチエグゼクティブ 2 2 および／またはプロセス制御ネットワーク 2 0 からのアラームおよびアラートデータなどの指標が含まれ得る。このデータのいずれかまたはすべてが、バッチエグゼクティブルーチンに典型的であるように、バッチログファイルデータベース 3 2 内にバッチログファイル 3 4 として記憶され得る。

【 0 0 3 9 】

一般的に知られているように、ワークフローアプリケーション 4 0 は、バッチエグゼクティブ 2 2 によって実施されている各命令に関してバッチエグゼクティブ 2 2 の動作を追跡し得、バッチオペレータがバッチエグゼクティブ 2 2 またはプロセス制御ネットワーク 2 0 内のまたはそれらに関するさまざまな動作またはアクションを実施することを可能にし得る。例えば、ワークフローアプリケーション 4 0 は、パラメータデータ、バッチ状態データ、バッチ状態変更データ、バッチアラームおよび／またはアラートなどの、バッチエグゼクティブ 2 2 からの特定タイプのデータをサブスクライブし得、このデータに基づいて、オペレータに、プラント 1 0 内のさまざまなコンディションを通知し得る。ワークフローアプリケーション 4 0 は、例えば、さまざまな異なるバッチ手順の開始および停止、またはバッチ手順モデルの状態変化をオペレータに通知し得、（欠落機器、欠落原材料などの予期しない問題によって起こり得る）バッチエグゼクティブエンジン 2 2 のさまざまな停止または他の動作コンディションをオペレータに通知し得、時間がずれていること、バッチエグゼクティブ 2 2 が予想していることに基づいて範囲外であり得るバッチパラメータなどをオペレータに通知し得る。ワークフローアプリケーション 4 0 は、既知のように、オペレータが、バッチランを再開するため、バッチランに対して異なる機器を選択するため、異なる原材料で動作させるため、バッチエグゼクティブに手順を開始、シャットダウン、またはスキップするように指示するため、またはバッチ手順モデル内の他の手順を実行するために、さまざまなアクションをとることを可能にし得る。一実施例として、バッチエグゼクティブ 2 2 は、洗浄されるべき機器がオフラインであり、したがって洗浄するには利用不可能である洗浄手順を実行しようと試みる場合がある。この場合、オペレータは、洗浄手順をスキップし得、それは、実施されている品質規格に厳密には準拠していない可能性があるが、バッチを、タイムリーにまたは割り当てられた時間枠内で完了するために必要であり得る。いずれの場合においても、ワークフローアプリケーション 4 0 は、オペレータが、オペレータがとったさまざまなアクションに注釈を付けることを可能にし得る。さらに、ワークフローアプリケーション 4 0 は、ワークフローアプリケーション 4 0 を介してオペレータによって提供された注釈またはメモ（例えば、何が起きたか、およびバッチプロセスの予想されるワークフロー外の特定のアクションをオペレータがなぜとったかを説明するために、オペレータによって提供されたメモまたはコメント）を記憶し得、ワークフローアプリケーション 4 0 を介して、オペレータがとったさまざまなアクションを示すデータをバッチログファイルデータベース 3 0 内に監視または制御されている命令に対するバッチログファイル 3 4 の一部として記憶し得る。各バッチログファイル 3 4 は、典型的には、バッチエグゼクティブ 2 2 から取得されるさまざまなデータを、そのデータに応答してオペレータがとったアクション、オペレータによってログファイル内に書き込まれたメモなどと共に、リスト化する。さらに、バッチエグゼクティブ 2 2 の動作中、バッチエグゼクティブ 2 2 は、データパケットをワークフローアプリケーション 4 0 に、バッチランまたは命令に関する現在データと共に送信する。ワークフローアプリケーション 4 0 は、バッチエグゼクティブ 2 2 （またはプロセス制御ネットワーク

10

20

30

40

50

20)からのさまざまなデータをサブスクライブし得、データに変更が生じたときなどに、このデータを定期的に受信し得る。さらに、各データセットは、データに関連付けられたバッチランまたは命令についてのさまざまな情報を識別する固有のパケット識別子を有し得る。このようにして、ワークフローアプリケーション40は、データパケットを処理して、命令を決定し、したがって、データを提供すべき適切なオペレータを決定し得る。

【0040】

既知のように、バッチランに対するバッチログファイルまたはバッチレポート34は、バッチラン中に生成され、バッチランまたは命令が完了した後、バッチログファイル34が完了する。その時点で、品質レビュー技師または他の担当者は、コンピュータまたはワークステーション28上で実行されている何らかの閲覧アプリケーションを介してバッチログファイル34にアクセスして、例えば、バッチレポートを分析し、バッチレポート内のデータおよび他の情報を行ごとに閲覧し得る。品質管理技師は、バッチラン内でいずれかの例外が発生したかどうか、これらの例外に対処する必要があるかどうか、これらの例外の存在に基づいて任意のアクションをとる必要があるかどうかを判断するために、バッチプロセスの動作中に記憶されているので、このデータを閲覧し得る。上記のように、このレビューは、多くの時間を要する傾向があり、手動で集中的に行われ、品質管理技師の側に多くの知識を必要とする。さらにその上、品質管理技師が費やす時間のほとんどは、バッチラン内の特定の例外または予想されるアクションからの逸脱に関連付けられていない、バッチレポートからのデータまたは情報を単に見ることであり、したがって、このレビューには時間がかかる。さらにその上、上記のように、品質管理技師がバッチログファイル34内のデータに基づいて例外を識別した場合、品質管理技師は、バッチヒストリアン30、プロセス制御ヒストリアン31など、他のデータベースのうちの1つに記憶されているバッチプロセスについての他の情報（例えば、プロセス変数値、バッチ手順モデル状態、レシピ情報など）を取得する必要があるし得る。この作業には、品質管理技師が他のデータアクセスアプリケーションを使用してそのデータにアクセスすることが必要であり得、したがって、技師が、これらの他のデータアクセスアプリケーションを使用して、効率的かつ正確に、そのデータにアクセスして閲覧するノウハウを有することが必要である。

【0041】

図2は、図1のプラント環境10と同様のプラント環境110を示し、同様の構造、アプリケーション、およびデータは共通の参照番号を有する。しかしながら、図2のプラント環境110は、品質管理技師（または他の担当者）が命令またはバッチラン（または複数の異なる命令またはバッチラン）に関連付けられた例外を識別およびレビューすることを支援するように、場合によっては、オペレータがバッチラン内でワークフロー動作を、命令中に生成される例外または例外レコードの数を減らす方法で実行するのを支援するように実行される品質レビュー管理システム112を含む。

【0042】

特に、図2のプラント環境110は、プロセス制御ネットワーク20およびバッチエグゼクティブエンジンまたはバッチエグゼクティブ22を含み、これらは、図1に関して上述したのと同様の方法で動作する。したがって、バッチエグゼクティブエンジン22および／またはプロセス制御ネットワーク20からのデータは、さまざまなデバイスに、さまざまなプロセス制御およびさまざまな異なる品質レビュー管理システム112のコンピュータまたはデバイスが接続されている通信ネットワークバックボーン25を介して提供される。この実施例では、バッチまたは他のプロセスデータは、ワークフローアプリケーション40を実行し、データを、バッチログファイル34を記憶するバッチログファイルデータベース32に提供するワークフローオペレータインターフェース26に提供される。しかしながら、図2のプラント環境110内の品質レビュー管理システム112は、プラント環境110内のコンピュータまたはプロセッサデバイス内に配置され、そこに記憶され、そこで実行されるさまざまなアプリケーションおよびデータベースを含む。特に、図2に示すように、品質レビュー管理システム112は、処理デバイス121のメモリ内に記憶されてそのプロセッサ上で実行されるコンフィグレーションアプリケーション120

と、サーバ１２３のメモリ内に記憶されてそのプロセッサ上で実行される例外エンジン１２２と、ルールデータベース１２４と、例外レコードデータベース１２６と、ワークステーション１２９などの処理デバイスのメモリ内に記憶されてその上で実行される品質レビューインターフェースアプリケーション１２８と、を含む。コンポーネント１２０、１２２、１２４、１２６、および１２８は、図２のコンピューティングデバイス２６、２８、３０、３１、および３２とは異なる別個のコンピューティングデバイスにおいて記憶および実行されるものとして示されているが、コンポーネント１２０、１２２、１２４、１２６、および１２８の任意のサブセットまたはすべては、デバイス２６、２８、３０、３１、および３２などの他のプロセス制御アプリケーションおよびデータベースを記憶および実行する同じ処理デバイスであり得る同じ処理デバイスにおいて記憶および実行されてもよい。

10

【００４３】

一般的に言えば、コンフィグレーションアプリケーション１２０は、コンフィグレーション技師、品質管理技師などのユーザが、特定の命令に関連付けられたバッチプロセスのバッチランなど、プロセスの稼働中に例外を検出するために例外エンジン１２２によって使用され得る一組のルール（本明細書では概して、例外ルールと称される）を作成および記憶することを可能にするように動作または実行され得る。コンフィグレーションアプリケーション１２０は、本質的に、ユーザが、例外を定義または検出するために例外エンジン１２２内で実行されるコンフィグレーション可能な例外ルールを作成することを可能にするコンフィグレーション環境を提供する。コンフィグレーションアプリケーション１２０は、ブール論理などの任意の所望のタイプのロジックまたは論理式を使用して例外を検出するために、さまざまなタイプの入力データを処理し得る一組のルールテンプレートを記憶し得る。コンフィグレーション環境では、ユーザは、ルールテンプレートのうちの１つを選択し得、ルールに対するデータ入力／出力を定義し得、データセットに対して分析されると考えられる式を定義し得る。データが例外エンジン１２２を通過するとき、例外エンジンは、設定されたルールに対してデータを分析して、例外が存在するかどうかを判断する。例外を検出すると、次いで、例外エンジン１２２は、例外レコードをデータベース１２６内に記憶する。

20

【００４４】

一実施例として、図３は、コンフィグレーションアプリケーション１２０が作成し、図２のワークステーション２８を介してユーザ（例えば、コンフィグレーション技師、品質レビュー技師など）に提供して、例えば、ユーザが１つ以上の例外ルール１５２を作成すること、および例外ルール１５２を例外ルールデータベース１２４に記憶することを可能にし得る、画面表示１５０を示す。特に、コンフィグレーションアプリケーション１２０は、ユーザが、異なるデータソースからのデータに適用されるべき異なるルールセットを作成することを可能にし得る。図２に示されたように、例外ルール１５２は、多数の異なるデータソースの各々について、異なる例外ルールセット１５２Ａ、１５２Ｂ、１５２Ｃ、１５２Ｄなどを含み得る。例えば、ルール１５２Ａは、ワークフローアプリケーション４０からのデータを分析するために作成されかつそれに関連付けられ得、ルール１５２Ｂは、イベントモニタアプリケーションからのデータを分析するために作成されかつそれに関連付けられ得、ルール１５２Ｃは、プラント環境１１０内で他のサービスを実行する第三者アプリケーションからのデータを分析するために作成され、かつそれに関連付けられ得る、などである。当然のことながら、各データソースに対して任意の数の例外ルール１５２を作成および記憶することができ、これらのルールは、それらのデータソースからのデータを分析するために、任意の数の異なる方法で設定し得る。例えば、単一のデータソースからの異なるタイプのラン（例えば、異なる製品を作成するバッチ）に対して、異なる例外ルールセットが作成され得る、異なる品質規格に対して、異なる例外ルールセットが作成され得る。同様に、特定のデータソースに関連付けられた例外ルールに順番を付け、どの例外ルールセットが１番目、２番目、３番目などに実行または分析されるか指定し得る。

30

40

50

【0045】

再び図3を参照すると、コンフィグレーションアプリケーション120は、ルールテンプレートセクション156、データソースセクション158、ルールリストセクション160、およびルールコンフィグレーションセクション162を含む多数の異なるセクションを含む画面を作成し得る。ルールテンプレートセクション156は、ユーザが新しいルールを作成するための出発点として使用し得る、例外ルールテンプレートまたはテンプレートとして記憶された以前に設定された例外ルールを示す一組のアイコンを含み得る。これらのテンプレートは、ルールデータベース124内に、またはコンフィグレーションアプリケーション120と同じデバイス内に、または任意の他のデータベースまたはコンピュータメモリ内に記憶され得る。データソースセクション158は、ユーザが例外ルールを作成し得るさまざまな異なるデータソースを示すアイコンを含み得る。アイコン158Aは、データソースから利用可能なデータ、データを通信するためにデータソースによって使用されるデータまたは通信フォーマット、データソースについての情報（例えば、製造業者、バージョンなど）、および／または任意の他の情報など、データソースについての詳細を閲覧するためにアクセス可能であり得る。ルールリストセクション160は、特定の選択されたデータソースに対するルールリストを記憶し、データソースからのデータを分析するときのルールの実行順序を指定し得る。

【0046】

動作中、ユーザは、データソースセクション158からデータソース158Aを選択して、ルールが作成されるべきデータソースを指定し得る。アプリケーション120は、ルールデータベース124にアクセスし、そのデータソースに対して以前に記憶されたルール（もしあれば）を見つけ、これらのルールをルールリストセクション160に、例えば実行順で、リスト化し得る。次に、ユーザは、テンプレートセクション156からテンプレートデータソースアイコン156Aを選択し、そのテンプレートアイコン156Aをルールコンフィグレーションセクション162にドラッグアンドドロップし、アプリケーション120に、選択されたテンプレートルールに対するコンフィグレーションデータを、コンフィグレーションセクション162のさまざまな部分またはフィールド内にロードさせることにより、データソースに対する新しいルールを作成し得る。所望であれば、ルールコンフィグレーションセクション162は、そのフィールドを空白にして、ユーザがテンプレートを使用せずに新しいルールを作成することを可能にし得る。さらにその上、ユーザは、ルールリストセクション160内のルールうちの1つを選択して、以前に作成されたルールを編集し得る。当然のことながら、アプリケーション120は、作成されるべきルールを選択または確立する他の方法を使用し得る（例えば、ドロップダウンメニュー、ポップアップ画面などを使用する）。

【0047】

次いで、コンフィグレーションアプリケーション120は、コンフィグレーション技師などのユーザが、ルールコンフィグレーションセクション162のさまざまなフィールドにおいてさまざまなルールコンフィグレーション情報を提供または編集することによって、ルールの詳細を指定、セットアップ、または作成することを可能にする。基本的に、アプリケーション120は、ユーザが、データソースによって監視されるプロセスの動作内の例外を検出および取り扱う目的で、データソースからのデータを処理するのに必要なルールコンフィグレーション情報を指定することを可能にする。概して、各ルールには、データソースからのデータを分析して例外を検出するために使用される、ブール論理などのロジックと、例外レコードをどのように記憶するかについての情報と、所望であれば、品質管理技師が、ルールの適用によって作成された例外レコードを処理または解決（例えば、クローズ）することをどのように可能にするかについての情報と、が記憶される。そのような例外ルールには、例えば、データソース（例えば、ワークフローアプリケーション40、バッチエグゼクティブ22、プロセス制御システム20など）からの1つ以上のパラメータが範囲外であるか、または特定の事前確立された閾値を上回るもしくは下回るかどうか；（例えば、ワークフローアプリケーション40を介して）オペレータによってメ

10

20

30

40

50

モが入力されたかどうか；命令に対するバッチランで異なる材料が使用されているかどうか；命令の1つ以上のバッチ手順に対して異なる機器が使用されているかどうか；1つ以上のバッチ手順の処理時間が、予想されたよりも長くまたは短くかかっているか；洗浄手順または濯ぎ手順などの命令に対するバッチラン内の特定のステップまたは手順のスキップまたは追加；プロセスシステムのオペレータまたは他のユーザが、プロセスで予想外のアクションをとったか、またはデータソースアプリケーションにメモを入力したかどうか、などを検出することが含まれ得る。当然のことながら、例外ルールは、例外が存在するかどうかを判断するために、データソースからの特定のデータに適用される任意の所望のロジックを指定し得る。

【0048】

したがって、図3の例示的画面150に示されているように、ルールコンフィグレーションセクション162は、例外ルールの名称を指定するのに使用され得る名称フィールド162Aと、ルールによって検出された例外の重大度または重要度のレベルを指定するのに使用され得る重大度フィールド162Bと、例外が存在するかどうかを検出するために例外エンジン122（図2）によって適用されるべきロジックを指定するために入力または使用され得るロジックフィールド162Cと、を含む。当然のことながら、ロジックフィールド162Cは、ブール論理パラメータまたは文字列のセット、特定の言語（C＋、HTMLなど）でのコーディング、何らかのロジック、If-Then論理文などを適用する任意の所望のカスタム表現シンタックスなど、ユーザがロジックを任意の所望の方法で入力または指定することを容認するまたは可能にし得る。さらにその上、ソースパラメータフィールドを使用して、フィールド162Cで提供されるようなロジックルールまたはロジックを適用するために、データソースから必要なあらゆるデータを指定し得る。本明細書ではデータソースメタデータまたは単にメタデータと称されるこのデータソースデータは、ルールに対してデータソースから受信されるべきデータの情報、フォーマット、および通信パラメータに関連付けられたすべてのデータを指定する。コンフィグレーションアプリケーション120は、ユーザがアイコン158Aのうちの1つを選択して、ルールが作成されている特定のデータソースに対するコンフィグレーション情報を取得して、どのデータソースメタデータ（情報および／または変数）がデータソースから入手可能であるか、これらの変数の名称、これらの変数のデータ形式（例えば、32ビット、64ビット、整数もしくはブールなどの変数タイプ、または文字列変数）などを決定することを可能にし得る。さらにその上、フィールド162Dは、ユーザが、データソースからの入力データがフィールド162Cで指定されたロジックに使用される前に、データソースからのメタデータに対して実行されるべき変換を指定することを可能にし得る。そのような変換は、数学的演算子（摂氏から華氏への変更、スケーリング、制限など）でもよく、変換は、（ロジックに使用されるさまざまな変数名をデータソース変数名に一致させるための）変数名変更でもよく、またはこれらの変換は、任意の他のタイプの変換でもよい。さらに、データソースからのメタデータは、プロセスパラメータ値、命令名、命令のための材料、命令のためのプロセスフロー、命令で使用されるプロセス機器、命令に適用されるワークフロー、命令で使用されるバッチ手順モデルなど、任意の所望の情報を含み得る。

【0049】

さらにその上、コンフィグレーション技師は、コンフィグレーションアプリケーション120を使用して、特定のルールセットの各ルールをデータソースからの特定のデータセットに適用する順序またはシーケンスを指定し得る。特に、例外エンジン122は、複数のルールをデータソースからの各データセットに適用し得、複数のルールの各々を特定のまたは所定の順序で適用して、特定のタイプの例外（典型的には、より重要なまたは重大な例外）を、他のタイプの例外（例えば、より重大でない例外）に先立ち、検出するようにセットアップされ得る。例えば、多くの場合、異なる例外ルールをデータソースからの特定のデータセットに適用すると、複数の異なる例外を検出する結果となり得る。しかしながら、バッチランまたは命令内の同じイベントまたは時間に関連付けられた複数の例外レコードの作成を防ぐように、データソースからの任意の特定のデータセットに対して最

10

20

30

40

50

も著しい例外のみを決定することが重要または望ましくあり得る。したがって、動作中に、例外エンジン 122 が、特定の順序でルールを実行し、次いで、最初に検出された例外のみに対する例外レコードを保存または作成することにより、さまざまなルールに基づいて例外を決定するように、ルールの順序（すなわち、例外エンジン 122 が例外ルールを特定のデータセットに適用する順序）を使用して、どの例外が最初に検出されるかを指定する。上記のように、ルールセクション 160 は、ソースに対するルールのリストを、これらのルールが実行される順序で記憶し得る。各ルールは、そのパラメータとして、この実行順序を示すシーケンス番号または順序番号を記憶または含み得る。いずれの場合においても、ユーザは、例えば、セクション 160 におけるルールリスト内のルールの順序を並べ替えることによって、データソースに対するルールの順序を変更し得る。ユーザは、例えば、ルールリスト 160 内のルールを選択し、そのルールをリスト 160 内で上または下にドラッグし、ルールをリスト 160 内の新しい位置にドロップして、ルールの実行順序を変更し得る。当然のことながら、アプリケーション 120 は、ユーザが、画面セクション 162 における順序フィールド 162E を変更するなど、他の方法でルールの順序を変更することを可能にし得る。

【0050】

さらにその上、所望であれば、画面セクション 162 は、コンフィグレーション技師が、ルールによって作成された例外レコードをクローズするために適用されるべきまたは適用されねばならない 1 つまたは恐らくそれ以上の取り扱いまたは解決手順を指定することを可能にし得る解決策フィールド 162F を含み得る。概して、各例外のタイプは、単一の解決手順を支持し得る。しかしながら、場合によっては、例外は、複数の解決手順を支持または含み得る。そのような解決手順には、例外を承認すること、例外レコードにさまざまな情報で注釈を付けること、例外レコードをレビューおよび／または承認のために他の担当者へ送信することなどが含まれ得る。場合によっては、各解決策は、単に、例外をクローズするのに必要な一組の署名の定義でもよく、または解決策は、単に、例外をクローズするのに必要な 1 つ以上の署名を執行してもよい。これらの場合、解決策は、署名に先立って完了されるべき他の作業（手順）を示す記述または一組の指示を含み得るが、これらの場合、解決策は、他の作業を監視または執行しなくてもよい。ルール内に指定された取り扱い手順は、事前設定されてもよく、例えばフィールド 162B で指定されるような例外の重大度に基づいて、デフォルト設定に設定されてもよい。

【0051】

さらにその上、画面セクション 162 は、ルールの適用によって作成された例外レコードと共に、またはその一部として、品質レビュー技師に提供されるべき情報を受理および記憶する情報フィールド 162G を含み得る。この情報は、このタイプの例外についての一般情報、通常の手続き、とるべきもしくは考慮すべき他のアクションについての指令もしくは示唆、またはルールの適用によって作成された例外レコードを閲覧または取り扱うときに品質レビュー技師にとって有用であり得る他の情報であり得る。多くの場合、フィールド 162G は、例外レコードの一部として記憶されるべきデータソースメタデータ、およびこのメタデータを表示する形式を指定するためにも使用され得る。概して、例外レコードを処理するとき、品質レビュー技師が例外時のプロセスのコンテキストを知ることが助けになり、データソースメタデータは、品質レビュー技師がこのコンテキストを理解するのを助ける情報として提供され得る。そのため、コンフィグレーションフィールド 162G は、例外レコードの一部としてどのデータソースメタデータが品質レビュー技師に提供されるべきかを示し得る。

【0052】

当然のことながら、コンフィグレーションアプリケーション 120 は、ルールコンフィグレーション情報のいずれかまたはすべてを、ルールデータベース 124 内に、例外ルールの一部として、任意の所望の方法で、記憶し得る。理解されるように、このコンフィグレーション情報は、データソースからの必要とされるデータのアイデンティティを含み得、このデータは、後述するように、データソースの動作中にデータソースから正しいデー

タまたは情報を取得するために使用することができる。

【0053】

当然のことながら、コンフィグレーションアプリケーション120を使用して、任意の数の例外ルールを作成し得、例外ルールは、データソースの方法によってデータソース上のルールデータベース124内に記憶され得、そうして、異なるデータソースに対して（または単一のデータソースに関連付けられた異なる命令に対して）、異なるルールまたはルールセットが作成および記憶され得る。いずれの場合においても、コンフィグレーションアプリケーション120は、ルールを作成するために使用され得、ユーザが（例えば、テンプレートルールから）新しいルールを作成することを可能にすることによってそれを行い得、ユーザが、ルールまたはルールによって検出されるべき例外の名称と、例外の存在またはデータに基づく標準からの逸脱を検出するために実施されるべき特定のデータソースからのデータに適用されるべきロジックと、ルールまたはルールが検出する例外のタイプの識別と、ルールが例外を検出するとレビュー者または他のユーザに提供される情報セット（例外の記述、例外を分析または承認するのに有用なデータまたは指示など）と、ルールによって検出された例外の重大度（例えば、重要性）の識別と、例外レコードの一部として記憶されるべき、例外時のプロセス（例えば、バッチラン）の状態に関するメタデータ（例えば、データソースからのデータ）と、例外を取り扱いまたは解決するのに使用され得るまたは使用されねばならないプロセスまたは手順（例えば、承認手順）と、例外レコードの一部として記憶される、または例外レコードの処理に使用される他のデータまたは情報と、を指定することを可能にし得る。

【0054】

重要なことには、このコンフィグレーションシステムまたはコンフィグレーションアプリケーション120は、システムに対して定義された例外を製品にハードコードするための例外ソフトウェアプロバイダを必要とし、したがって、新しい例外を作成するためにソフトウェアまたはシステムの更新を必要とする周知の製品よりも、著しい利点を提供する。これらの更新は、ユーザが取得してインストールするのに時間がかかり、ユーザが自身のシステムをカスタマイズすることはできない。

【0055】

一組の例外ルールが1つ以上のデータソースに対して作成され、これらのルールがルールデータベース124に記憶された後、例えば、品質レビュー管理システム112は、その後、データソースの動作中に例外エンジン122をリアルタイムで実施して、根底にあるプロセスの動作がデータソースアプリケーションによって監視、制御、または影響されている結果として発生し得る例外を検出し得る。特定のデータソースからのデータに稼働または実行するようにセットアップされたとき、例外エンジン122は、定期的にまたは断続的に、プロセスを監視または制御するプロセスおよび／またはアプリケーションのさまざまな動作パラメータを示すデータセット（例えば、メタデータ）をデータソースから受信する。各データセットについて、例外エンジン122は、データソースに対して作成された例外ルールのロジックを取得および適用し、それによって、データソースによって実施、管理、または監視されているプロセスにおける例外を検出する。データソースとインターフェースまたは通信するとき、例外エンジン122は、何らかの特定の1つまたは複数のアクションがプロセスまたはデータソース内で発生したとき、および／または他の設定された時間に、データセットをデータソースから定期的に受信し得る。例外エンジン122は、データソースに対する各例外ルール内のデータソースコンフィグレーションデータに基づきそのようなデータをサブスクライブしてもよく、そのようなデータに対するデータソースをさまざまな時点にポーリングしてもよく、またはこれらの通信技術の組み合わせを実施してもよい。このようにして、ワークフローアプリケーション40などのデータソースが、新しいデータを受信するように、および（例えば、オペレータが、例えばバッチエグゼクティブ22からのバッチデータを分析するのを支援するために）オペレータとインターフェースするように動作するとき、ワークフローアプリケーション40はまた、データを（定期的または別様に）例外エンジン122に送信し、次いで、例外エンジ

ン 1 2 2 は、ワークフローデータソースアプリケーション 4 0 に対する例外ルールを使用してデータを分析して、1 つ以上の例外が根底にあるプロセスで発生したかどうかを判断する。

【 0 0 5 6 】

このプロセスの間、ロジックエンジンである例外エンジン 1 2 2 は、データソースから送信されたデータをパースし、データベース 1 2 4 内のルール 1 5 6 A などの適切なルールセット内のロジックを適用し、それにより、ルールに従ってデータを分析して 1 つ以上の例外を検出する。例外エンジン 1 2 2 は、ルールをデータに、ルールデータベース 1 2 4 内に記憶されているようなルールセット内に指定された順序で 1 つずつ適用してもよく、ルールによって決定されるようないかなる例外も検出してもよく、または例外が特定のデータセットにおいて任意のルールによって検出されたとき、データの処理を停止してもよい。例外を検出すると、例外エンジン 1 2 2 は、そのデータセットに対する例外レコードを作成し、例外レコードをデータベース 1 2 6 内の例外レコードファイル 1 7 0 (図 2) 内に記憶する。当然のことながら、新しい各データセットがワークフローアプリケーション 4 0 (または他のデータソース) によって例外エンジン 1 2 2 に提供されると、例外エンジン 1 2 2 はそのデータを例外について処理し、例外が存在するかどうかを判断し、次いで、適宜、例外を作成して、例外レコードファイル 1 7 0 内に記憶する。例外エンジン 1 2 2 は、例えば、識別または検出された例外の名称、例外の重大度、適用されるレビューまたは解決手順、データソースから送信されたデータの一部としてのオペレータが入力したメモ、例外についての一般的情報を含む、例外レコードに関連付けられたさまざまな異なる種類のデータを記憶し得る。さらにその上、例外エンジン 1 2 2 は、例外レコードの一部として、例外時のバッチまたは他のプロセス機器の動作を何らかの方法で定義するまたはそれに関連するメタデータを提供し得る。このメタデータは、例外エンジン 1 2 2 にデータソースからのデータの一部として提供されてもよく、あるいは例外エンジン 1 2 2 によって、データソース (例えば、ワークフローアプリケーション 4 0) から、またはデータソースとは別個に、そのデータのデータベースもしくは他のソースから取得されてもよい。例外レコードに記憶されているメタデータは、例外が検出された時間に実施されていたプロセスに関連付けられた何らかの動作パラメータ、値、状態、または他の情報を定義する任意のデータであり得る。このメタデータは、例えば、バッチ状態情報、プロセスから測定または決定されたようなプロセスパラメータ値、プロセス機器情報、プロセス機器からのアラームもしくはアラート情報、または任意の他の情報であり得る。さらに、このメタデータは、バッチログファイル 3 4、プロセス制御データベースもしくはヒストリアン 3 1、プロセス機器、バッチヒストリアン 3 0 など、他のソースから取得されてもよく、または例外エンジン 1 2 2 によって処理された初期データの一部として、もしくは例外エンジン 1 2 2 による問い合わせに回答して、データソース自体から直接提供されてもよい。一般的に言えば、記憶されたメタデータは、例外が検出された時間のプロセスのスナップショット (例えば、プロセス状態情報、プロセス機器情報、プロセス変数値、バッチ手順モデル状態など) を提供する。このメタデータは、例外レコードの一部として表示されて、品質管理技師が例外のコンテキスト (例えば、例外時にプロセスで何が起こっていたのか) を理解し、それにより、例外および例外の重要性をより良く理解することを可能にし得る。

【 0 0 5 7 】

理解されるように、例外エンジン 1 2 2 は、リアルタイムで、すなわち、データソースと並行して進行して動作して、データソースからのデータを例外について分析し、検出された例外に対する例外レコードを例外データベース 1 7 0 に記憶する。例外エンジン 1 2 2 は、新しいデータパケットセットを、ワークフローエンジン 4 0 などのデータソースから、任意の時間の間、受信するように動作し得る。例えば、例外エンジン 1 2 2 は、特定のバッチプロセスが実行されるときにワークフローアプリケーション 4 0 からのデータを分析し得、バッチプロセスまたは命令の完了時に停止し得る。当然のことながら、例外エンジン 1 2 2 は、例えば、同じデータソース (例えば、ワークフローアプリケーション 4

10

20

30

40

50

0)からの複数の異なるバッチランに対する例外を同時に分析および検出するように動作してもよく、かつ／または異なるデータソースからのデータを同時に分析してもよい。これらのすべての場合において、例外エンジン122は、単一のデータソースの異なる各バッチランに対して、および／または異なる各データソースに対して、例外レコードを同時に作成および記憶し得る。したがって、例外エンジン122は、異なるバッチランが同時に発生していること、または異なるソースからのデータが同じ時間にわたって処理されていることを理解し得、これらのランおよびプロセスを別々に追跡して、各バッチラン、命令、および／またはデータソースに対して異なる例外データベース170を作成し得る。

【0058】

図4は、複数の異なるデータソース180、182、184、および186における例外エンジン122の動作を示す。データソース180～186は、同じタイプのデータソース（例えば、異なるバッチを実行しているまたは異なる製品を製造しているワークフローアプリケーション40の異なるインスタンス）でもよく、かつ／またはイベント監視アプリケーションまたはシステム、第三者の専有システムまたはアプリケーションなどの異なるタイプのデータソースでもよい。この場合、例外エンジン122および／またはデータソース180～186は、データソースメタデータを含むデータパケット188を例外エンジン122に定期的を送受信するように設定されている。これらのデータパケット188は、データソースに対するルール内のメタデータコンフィグレーション情報によって指定されるようなデータソースからのデータを含むようにフォーマット化され得、これらのデータパケット188は、データソースの識別子、データソースによって実施される命令、データソースのメタデータなどを含み得る。

【0059】

例外エンジン122は、異なるデータソース180、182、184、および186から各データパケット188を受信し、これらのパケットが受信されると、順番に処理する。各パケット188について、例外エンジン122は、（データパケット188内のデータによって指定されるような）データソースおよび命令のアイデンティティを決定し得、そのデータソースに対して適切なルールセットをルールデータベース124から取得し得、取得したルールを事前定義された順序でデータパケット188内のデータに適用して、例外が存在するかどうかを判断し得る。例外エンジン122が、例外を検出せずにデータソースに対するすべてのルールを処理した場合、次いで、例外エンジン122は、（同じまたは異なるデータソースからの）次のデータパケット188を待つか、または処理を開始する。一方、例外エンジン122がデータソースメタデータへのルールロジックの適用に基づいて例外を検出した場合、例外エンジン122は、命令に対する例外レコード180を作成し、その例外レコードを、例外レコードデータベース126内に、命令に対する例外レコードファイルまたはデータベース170として記憶する。つまり、特定の命令に対するすべての例外レコードは、その命令に対する例外レコードセットとして記憶され得る。いずれの場合においても、図4のダイアグラムで示されるように、例外エンジン122は、さまざまな異なるデータソースからのデータに、および／または（同じまたは異なるデータソースからの）さまざまな異なる命令に対して、同時に動作して、各命令に対する例外データベースを作成し得る。

【0060】

例外エンジン122の1つの追加的利点は、リアルタイムで、すなわち、根底にあるプロセスの動作中に動作し、したがって、例えば、例外が発生したときに、例外の発生をリアルタイムで検出できることである。結果として、例外エンジン122の1つの特徴は、例外の発生を検出すると、例外エンジン122が即時またはリアルタイムで、データソースにいるまたはデータソースに関連付けられたオペレータまたは何らかの他の担当者に、例外の存在を通知できることであり、これは、オペレータまたは他の者がプロセス内でアクションをとって例外を軽減または回避することを可能にし得る。場合によっては、現在のプロセスのアクションまたは状態に基づいて、将来発生する可能性のある例外を検出するように例外ルールを設定し得る。この場合、例外エンジン122は、起こり得るまたは

10

20

30

40

50

起こりそうな例外の発生を検出し得、例外が発生する可能性が高いことをオペレータまたは他のユーザにリアルタイムで通知し得る。この通知により、オペレータまたは他のユーザには、例外が発生する前に、将来の例外の根本原因である何らかのアクションを取り消すか、何らかのパラメータを変更する能力が与えられる。したがって、このライブフィードバックを使用して、オペレータまたは他のプロセス監視または制御担当者が、例外につながるコンディションを回避または逆転するアクションをとることを可能にすることで、例外を回避または即座に軽減できる。

【0061】

図5は、例えば、図2のワークフローアプリケーション40によって、ワークフローアプリケーション40の通常動作の一部としてオペレータに提供され得る画面表示200を示す。この場合、表示200は、プロセス情報をプロセスまたはワークフロー図202の形態でオペレータに提供し得、オペレータが入力フィールド204を介してプロセス内で1つまたは複数の何らかのアクションをとることを可能にし得る。この場合、オペレータは、画面200内の入力フィールド204を使用して、特定のプロセスパラメータの設定点を変更し得る。しかしながら、オペレータが新しい設定点を入力すると、またはそうした直後に、ワークフローアプリケーション40は、データパケットを例外エンジン122に、この変更を示すデータと共に送信する。この実施例では、例外エンジン122は、（このプロセスパラメータは品質の目的で重要なプロセスパラメータであり得るので）プロセスパラメータの値が特定の範囲内にあるかどうかを判断するルールを実施するようにプログラムされ得る。例外エンジン122は、ルールを実行すると、設定点が変更された場合、重要なパラメータが所望の範囲外に設定され、したがって、例外となることを検出し得る。次いで、例外エンジン122は、直ちに、ライブフィードバックメッセージを介してアプリケーション40に通知を送信し得、それが、通知またはアラームがオペレータに提供される結果となり得る。例えば、図5に示すように、ポップアップボックス206が、ライブフィードバックメッセージに基づいて画面200内に作成され、オペレータに、企図されたアクションまたはとられたばかりのアクションが例外レコードを作成する結果となる（または結果となった）ことを伝え得る。ポップアップボックス206内のこのメッセージは、オペレータに、（すでに発生している場合は）例外を回避または軽減するためのアクションをとるように指示し得る。したがって、理解されるように、データソースに対してルールデータベース124内に作成および記憶されるようなルールは、実際の例外を検出すること、および／または将来起こりそうな、例外につながるコンディションを検出することに関係し得る。さらに、例外エンジン122のライブフィードバック機能を使用して、ユーザが、将来の例外の発生を回避すること、または現在検出されている例外につながるコンディションを軽減または逆転することを可能にし得る。この通知はプロセスの動作に対してリアルタイムであるので、このライブ品質フィードバック技術を使用して、例外の影響を逆転または軽減できる可能性ははるかに高い。

【0062】

図2は、品質レビュー管理システム112の動作を、システム112のさまざまな部分が専用サーバおよびデータベース内で実行される加工プラント環境110において示しているが、品質レビュー管理システム112は、システムをより運搬可能にし、モバイルまたは分散型デバイスでより簡単に使用でき、分散型環境での保守管理および実施が容易で、かつ第三者のシステムまたはアプリケーションで使用するために簡単に設定できるプラグイン環境で実施され得る。特に、図6は、プラグインを使用して実施された品質レビュー管理（QRM）システム300を示す。図6の品質レビュー管理システム300は、さまざまなデータソース304A、304B、304Cなどと、ルールデータベース124と、コンフィグレーションアプリケーション122とに通信可能に結合された（サーバまたは他のコンピューティングデバイス内に記憶され、その上で実行され得る）品質レビューサービスデバイス302を含む。さらにその上、品質レビュー管理サービスデバイス302は、プラント環境の内または外のワークステーション内、ラップトップ、電話などのモバイルデバイス内、または任意の他のデバイス内など、任意の所望のコンピューティン

10

20

30

40

50

グデバイス内に記憶され、実行され得る、さまざまなプラグイン310、312、314、316などに結合されている。プラグイン312～314は、任意の特定の命令またはデータソースに対して、必要に応じて作成（インスタンス化およびスピノフ化）され得る。この場合、各プラグイン310～316は、特定のデータソースおよび／またはデータソース内の特定の命令に対して作成されて、例外検出のためにそのデータソースからのデータを分析し得る。命令またはアプリケーションが完了または終了すると、プラグインは、破壊され得る。

【0063】

一般的に言えば、QRMサービスデバイス302は、コンフィグレーションアプリケーション122に応答して、特定のデータソースまたは特定のデータソースからの特定の命令に対して、特定のプラグインを作成し得る。各プラグイン310～316は、評価のためにプラグイン310～316に送信される情報のデータのデータおよびデータ形式と、データソースまたは命令に対する例外レコードファイルまたはデータベースを記憶する場所と、プラグイン310～316の動作に必要な任意の他の所望のコンフィグレーションデータをプラグイン310～316に通知するために、プラグイン310～316のコンフィグレーションの一部として提供された、コンフィグレーション情報を含み得る。さらに、所望であれば、プラグイン310～316のためのコンフィグレーションデータは、受信データに基づいて適用または分析するための1つ以上のルールを含み得る。当然のことながら、各プラグイン310～316は、同じデータソースまたは異なるデータソースに関連付けられる（それらに対して設定される）ことができ、そこで、各プラグイン310～316のためのコンフィグレーション情報は、データソースまたはプラグイン310～316の使用に応じて、類似または大幅に異なる可能性がある。各プラグイン310～316のコンフィグレーション情報は、QRMサービスデバイス302、または動作中の第三者のアプリケーションまたはソースにサブスクライブするまたはそれらから受信されるデータを指定してもよく、プラグイン310～316は、QRMサービスデバイス302または他のアプリケーションもしくはソースに登録して、そのデータを受信してもよい。概して、プラグイン310～316は、単に、データソースから特定のデータを取得するように動作し得、そこで、どのデータがデータソースから来るか、およびそのデータの形式を理解するように設定されればよい。次いで、プラグイン310～316は、データソースからデータを受信して処理し、受信したデータを、データソースに対して作成されたような例外ルールで使用するための形式にし得る。このようにして、プラグイン310～316は、任意のタイプのデータソース、さらには第三者のまたは専有のデータソースのためのデータ収集および解釈アプリケーションとして動作し得る。

【0064】

さらに、各プラグイン310～316は、動作中に、プラグイン310～316の動作を定義または制御するランタイム処理コンフィグレーションを含み得る。このランタイムセクションは、データソースに対する例外ルール内のルールまたはロジックを使用して、データソースから受信したデータを分析し、例外レコードを作成して、例外レコードデータベースまたはログ170内に記憶するロジックパーサーを含み得る。本質的に、プラグイン310～316のランタイムセクションは、インスタンスごとに図2の例外エンジン122を実施する。したがって、上記したように、プラグインモジュール310～316は、例外エンジン122を外部または第三者のシステム（例えば、制御システム、OPCインターフェース、保守管理システムなど）とインターフェースさせ、それにより、例外が、第三者または外部のソフトウェアシステムからのデータに全体的または部分的に基づいて生成されることを可能にするように作成され得る。1つの実施例では、プラグインは、第三者または外部のシステム内のまたはそれらからの特定のデータにアクセスするように、かつこのデータを例外エンジン122に渡して、例外が作成されるべきかどうかを分析するように作成される。しかしながら、プラグイン310～316は、第三者サーバ内、QRMサービスデバイス302内、または別のサーバ、モバイルデバイス、もしくは他の処理デバイス内に記憶され、実行されてもよい。

【0065】

一実施例として、図6のコンフィグレーションアプリケーション122を使用して、プラグイン310～316によって分析されるべくデータソースから受信されるデータのフォーマット、プラグイン310～316がどのくらいの頻度でデータソースからデータを受信するべきか、命令およびデータソースについての他の情報のアイデンティティ、プラグイン310～316が処理に使用されるアプリケーションのタイプなどを指定することを含む、1つ以上のプラグイン310～316を作成し得る。

【0066】

さらに、プラグインモジュール310～316は、そう所望であれば、データ取得機能およびランタイム例外処理機能を伴って作成し得る。いったん作成されると、プラグイン310～316は、データソースに近いデバイスまたはデータソースなど、任意の好都合な場所または処理デバイスに記憶され、実行され得る。さらに、プラグイン310～316は、データサービスデバイス302（または任意の他のサーバもしくはデバイス）に登録して、データソース304のうちの1つなどの適切なデータソースからデータを受信し得る。次いで、QRMサービスデバイス302は、適切なデータについてデータソース304をサブスクライブまたはポーリングし得、データソースからデータパケットを受信すると、データおよびデータソースに対する1つ以上の例外ルールを、そのデータを分析するために使用されるプラグインモジュール310～316に提供し得る。次いで、プラグイン310～316は、そのロジックエンジン内の1つまたは複数のルールを使用して、例外に対するデータを分析し、例外を検出すると、データソースに対する例外レコードを例外レコードデータベース170内に記憶し得る。したがって、理解されるように、QRMサービスデバイス302は、任意のデータまたは異なるソースから取得されたデータを使用して、任意の所望のデバイス上で互いに独立して動作できる、プラグイン310～316のためのデータブローカとして作用する。

【0067】

理解されるように、品質レビュー管理システム128は、ソースごとにプラグインまたはプラグインモジュールの使用を実施して、本明細書に記載のように、例外エンジンの機能を実行して、品質レビュー管理システムが、異なるデータソースに対して、異なる品質レビュー基準に対して、容易にコンフィグレーション可能であること、および／またはシステムのさまざまな異なる部分がプラント全体に広がる異なるコンピュータ内で実行される分散型環境で容易に実行されることを可能にする。この場合、各さまざまなタイプのデータソースに対して異なるプラグインを作成でき、各プラグインは、特定のデータソースからの例外検出を実行するのに使用されるように関連付けられた、それ自体のルールセットを有することができる。

【0068】

さらに、プラグインモジュール310～316は、バッチまたは命令処理環境で例外処理および検出を実行してもしなくてもよい。場合によっては、例えば、1つ以上のプラグイン310～316は、プラント内のプロセスまたはイベント監視を実行または拡張するため、または連続プロセス製造プラント内などのプラント内で品質レビューを実行するために作成され得る。したがって、プラグイン310～316は、単純な第三者システムデータ収集体であってもよく、またはデータを収集し、追加的に何らかの例外処理を実行してもよく、または例外データまたは前処理された例外データをスタンドアロン例外エンジンに提供してもよい。既知の品質レビュー製品は、制御システム、安全システム、保守管理システムのように、インターフェースすること、または外部ソフトウェアプログラムから直接提供されるデータに基づいて例外を作成することはできないので、プラグインを使用して品質レビューを実施することは有利である。さらに、プラグインモジュール環境の使用により、本明細書に記載の品質レビュー管理システム112は、ランタイム中に動作する複数の異なるソフトウェアシステムを有する複雑なプロセス環境においてはるかに堅牢となる。

【0069】

図7は、本明細書に記載の品質レビュー例外処理技術を使用するデータまたはイベント監視システムの一実施例を示す。特に、図7は、連続プロセスまたはプロセス機器を実行し得る加工プラント400の高レベル図を示す。図7に示されるように、加工プラント400は、例えば、連続プロセス（原油精製プロセスなど）、またはバッチプロセス、または両方を実行または実施することに関連付けられ得るプロセス制御機器402を含む。加工プラント400は、プロセス制御機器402に接続された1つ以上のバッチエグゼクティブ404と、プロセス制御機器402の1つ以上のセクションまたは部分を監視する1つ以上のオペレータインターフェース406と、を含み得る。さらにその上、バッチエグゼクティブ404およびオペレータインターフェース406は、プラント400のためのコンフィグレーションおよび制御情報を記憶し得るマスタープラント制御デバイス410に接続され得る。さらにその上、プラント400は、例えばデータベースアプリケーション、監視アプリケーション、プラント分析アプリケーションなどを含む他のアプリケーションを実行させ得る1つ以上のアプリケーションサーバ411を含み得る。

【0070】

多くの場合、プラント400内のプラントアセットは、動作中に大量のデータをプラントから収集する専有のまたは第三者の制御、監視、およびアセット管理システムを実行させる。さらに、オペレータ監視アプリケーションおよびインターフェース406、バッチエグゼクティブ404または他の制御アプリケーションも、本質的に優先され得る。プロセス制御技術では、OPCインターフェースを使用して、さまざまな第三者のまたは専有のアプリケーションからデータを取得することが知られている。したがって、オペレータインターフェース406の各々は、OPCアラームおよびイベント（OPCa e）412またはOPCデータアクセス（OPCd a）インターフェース412を含み、例えば、（プロセス機器402内など、プラント400内で生成されるような）アラームおよびイベントデータ、または外部システムによって使用される他のプロセスデータの、検出および使用を可能にし得る。追加的に、場合によっては、1つ以上の監視アプリケーション414が、アプリケーションサーバ411または別個のイベントモニタサーバ416内など、1つ以上のプラントデバイス内に記憶され得る。イベント監視アプリケーション414は、1つ以上のバッチエグゼクティブ404とインターフェースして、プロセスイベント情報を（必ずしもOPCインターフェースを介してではないが）取得するために使用される既知のアプリケーションであり得、かつ／または1つ以上のOPCa eまたはOPCd aインターフェース412とインターフェースして、アラームおよびイベントデータまたは他のデータをプロセス制御ネットワーク402から取得するように設定され得る。イベントモニタアプリケーション414は、従来から、プラントアセットからイベントおよびアラーム情報などのプロセスデータを取得するため、およびプロセス制御分析システム、保守管理システムなどのさまざまなシステムにおいて、後のレビューまたは使用のために、そのデータをデータベース内に記憶するために使用されている。

【0071】

しかしながら、本明細書に記載の例外処理構造は、イベント監視アプリケーション414から取得されるようなアラームおよびイベントならびに他の情報を処理して、特定のプラントコンディションがプラント機器402内またはその動作中に発生したかどうかを判断するために使用され得る。そのようなコンディションには、例としてのみ、特定のイベントもしくはアラームが臨界レベルに達したとき、特定のセットもしくは組み合わせのアラームもしくはイベントが一緒に発生したとき、さまざまなプロセスコンディションが存在するとき、または品質レビュー技師、安全管理技師など、何らかのユーザに留意される必要があり得る問題を示す任意の他の一組のコンディションが含まれ得る。この機能を実行するためには、図7のイベントモニタプラグイン450を使用して、プラント400内の1つ以上のイベント監視アプリケーション414、プラント400内の1つ以上のOPCa eおよび／またはOPCd aインターフェース412などからのデータにアクセスし得る。イベント監視プラグイン450は、プラント400内のイベント監視アプリケーション414、OPCa eおよびOPCd aインターフェース412、または他のデータソ

10

20

30

40

50

ースからデータを受信し得、例外ルール処理を実行して、プラント400内の特定の、典型的にはより複雑なコンディションの存在を検出し得る。すなわち、品質レビュー技師、安全管理技師、保守管理技師、プロセス制御技師、または他の者が知りたいかもしれないが、プロセス制御または監視システムがセットアップされ、プラント400を稼働させているとき、これらのシステムによって別様に検出されないコンディションが、プラント400に存在し得る。概して、これらのコンディションは、プラント400内で生成される単一のアラームまたはイベント通知によって単純には決定され得ないが、特定の臨界レベルを超えるアラーム、同時にまたはほぼ同時に発生する特定の数の何らかのタイプのアラーム、プラント400内の1つ以上の他のイベントに関連する特定のタイプまたは重大度のアラームなど、プラント400内のより複雑な一組のコンディションの存在によって決定される必要があり得る。理解されるように、プラグインモジュール450は、プラント400の動作におけるこれらの「例外」を決定するためのロジックを実施するルールで動作する（本明細書に別様に記載されるような）例外エンジンへのアクセスを含むかまたは提供し得る。したがって、図7のイベントモニタ450は、第三者アプリケーション（イベント監視アプリケーション414）またはシステム（イベントモニタサーバ416）またはインターフェース（OPCaeインターフェース412）からデータを収集し得、このデータを、一組の事前設定されたルールを使用して処理して、品質レビューの目的で、拡張されたイベント検出またはプロセス監視または例外処理を実行し得る。この場合、（図7の点線で示されるように、プラント環境とは異なる環境に配置され得る）イベント監視アプリケーション450は、ルール分析の結果を、後のレビューのために、監視インターフェース452、および／またはSQL（Sequel）データベースなどのデータベース454に提供し得る。本質的に、イベント監視プラグイン450は、プラント機器402によって収集され、プラント400内の第三者のアプリケーションまたはインターフェースに提供されるイベント監視データに基づいて、プラント機器402の動作における「例外」を検出するように動作する。このタイプの監視は、品質レビュープロセスの一部として実行されない場合でも、この監視はプラント400の動作における例外を検出するので、本明細書ではやはり例外処理と称される。

【0072】

いずれの場合においても、イベントモニタ450は、1つ以上の第三者または外部のシステムのいずれかに結び付けられ得、メッセージを生成してルールエンジン（図示せず）に送信し得、次いで、ルールエンジンは、これらのメッセージを使用して、監視メッセージ、通知、アラームなどの形態の例外処理または例外作成を実行し得る。1つの場合では、内部イベントモニタ450がOPCaeインターフェース412に結び付けまたは結合されて、1つ以上のアラートまたはアラームがOPCaeインターフェース412から渡されたときを認識し得る。このとき、イベントモニタアプリケーション414またはOPCインターフェース412は、アラートまたはアラーム情報と、恐らくは他の所望のOPC収集データとを用いてメッセージを作成し、外部イベント監視システム450に届ける。次いで、システム450は、メッセージを分析して、メッセージ内のデータに基づいて例外を生成し得る。既知の品質レビューシステムは第三者システムからのデータを分析せず、存在するOPCアラートおよびアラームインターフェースに接続されていないので、この動作は有利である。そのため、これらのシステムは、例えば、プロセスのアラームおよびアラートに基づいて例外を生成することができない。

【0073】

一般的に言えば、図7のイベント監視システム450は、プロセス制御システム402からアラームおよびイベントデータを収集して処理し、例外処理を使用して拡張監視システムを生成するように、多数の異なる方法で設定することができる。特に、ユーザは、上述したようなプラグインであり得るイベント監視システムまたはモジュール450を、例えば、図7のOPCaeおよびOPCdaインターフェース412によって収集されるように、データ（例えば、イベントおよびアラームデータ、メッセージなど）をサブスクライブして収集するように、設定することができる。次いで、このデータを収集するイベン

ト監視サーバ416は、このデータをデータパケットでイベント監視アプリケーション450に、（OPCインターフェース412、バッチエグゼクティブ404、プロセスコンピュータ410などから取得された）タイミング情報、プロセス制御メタデータと共に、送信することができる。次いで、イベント監視システム450は、上述したように、データに一組のルールを適用して、収集されたイベントおよびアラームデータとルールロジックとに基づいて、イベント監視の例外が存在するかどうかを判断することができる。例外が決定された場合、イベント監視システム450は（ルールに基づいて）メッセージを作成し、そのメッセージを、ユーザに表示するために監視インターフェース452へ、およびまたは、例えばSequenceサーバ内のデータベースもしくはメッセージキュー454へ送信し得る。当然のことながら、例外を生成したルールは、メッセージの内容、メッセージの形式、メッセージ内に置かれる情報、メッセージが送信されるインターフェースなどを指定することができる。追加的に、または代わりに、イベント監視システム450は、例外レコードを作成して、記憶および後の読み出しのために、監視データベースに送信することができる。（監視コンディションレコードである）この例外レコードは、ルールコンフィグレーションに基づく任意の所望の情報を含み得、例えば、イベントの記述、イベントの重大度、イベントを取り扱うための指示、イベント時のプロセスメタデータなどを含むことができる。理解されるように、この場合、イベントに対するルールロジックは、イベント監視システム450内（またはそれに結合された例外エンジン内）で処理され、図6のようなプラグイン環境内もしくはそれを使用して、または図2のような従来のサーバ環境内で、処理することができる。

【0074】

別の場合では、イベント監視システム450は、そのコンフィグレーションに基づいて、例えば、特定のコンディションまたはコンディションの組み合わせを検出するように、次いで、これらのコンディションが満たされたときにイベント監視システム450にメッセージを送信するように、OPCインターフェース412を設定することができる。この場合、イベント監視システム450は、本質的に、ルールのロジックを実行するように、そのロジックが満たされたときにのみメッセージまたはデータをイベント監視システム450に送信するように、OPCaeまたはOPCdaインターフェース412をプログラムする。特定の一組のコンディションまたはロジックが満たされたことを示すメッセージと共にOPCaeインターフェース412からメッセージを受信すると、イベント監視システム450は、単に、通信を作成してイベント監視インターフェース452およびデータベース454に送信して、ユーザにコンディションを通知すること、および例外レコードをデータベース454に記憶することができる。この場合、イベント監視システムに対するルールのロジックは、OPCインターフェース412、または既知の技術を使用して、加工プラント400内に特定のコンディションが存在するときのみメッセージを送信するように構成され得る、外部イベント監視アプリケーション414、416内で、全体的または部分的に実行される。

【0075】

したがって、図7の例示的システムでは、本明細書に記載の品質レビュー管理システムの例外ベースの処理は、例えば、ワークフローアプリケーション40と同じデータ形式を使用しない第三者システムからのデータにおいて例外を分析、レビュー、または検出するために使用され得る。図7の場合、品質レビュー管理システムは、1つ以上のOPCインターフェース412を介してバッチまたは連続プロセス制御システムとインターフェースして、さまざまなイベント、アラーム、パラメータ設定などの存在に基づく、監視ベースの例外を決定する、イベント監視システムとして使用されるときに示されているが、本明細書に記載の品質レビュー管理システムは、任意の他の所望のインターフェースを介して第三者のプロセス機器、サーバ、アプリケーションなどに接続することができる。

【0076】

図2に戻って参照すると、品質レビューインターフェースアプリケーション128は、品質レビュー技師などのユーザが、特定のデータソースからの、1つ以上のバッチラン、

命令、プロセス動作などのための1つ以上の例外レコードデータベース170にアクセスすることを可能にするように実行され得る。品質レビューインターフェースアプリケーション128は、例えば、ユーザが、例外データベース170内の各例外レコードに1つずつ踏み込んで、検出された例外を閲覧すること、および例外レコードをクローズするための任意の必要なアクションをとることなど、これらの例外を解決することを可能にし得る。品質レビューアプリケーション128は、インターフェースデバイスのインターフェース画面を介して、例外の名称、例外のタイプ、例外の重大度、例外レコードに対して記憶されたバッチメタデータまたはスナップショット情報、例外レコードが作成されることに関連付けられたまたはその原因となった、オペレータによって作成および記憶されたメモ、データソースによって提供されたその他の情報など、各例外レコードに対するデータまたは情報を提示し得る。この情報は、品質レビュー者が、例外と、例外発生時のプロセス（例えば、バッチラン）のコンテキストとを迅速に理解して、例外と例外の重要性とをより良く理解することを可能にする。品質レビューインターフェースアプリケーション128はまた、このタイプの例外についての一般的情報、このタイプまたは重大度の例外を取り扱うまたは承認するために実行される必要のあるステップなど、さまざまな他のタイプの情報を構造化された方法でユーザに提示し得る。さらにその上、品質レビューインターフェースアプリケーション128は、例外を承認または取り扱うために、ユーザが、例外を検出した元の例外ルールによって定義されるような、さまざまな手順のうちの1つを実行するなど、例外を承認またはクローズするのに必要な1つ以上のさまざまなステップをとることを可能にし得る。品質レビューアプリケーション128は、例えば、ユーザが画面上のチェックボックスをクリックして例外を認識および承認することを可能にしてもよく、例外をクローズするために、1人以上の他の人々に例外を閲覧および承認させることを含むプロセスを要求または執行してもよい。さらにその上、品質レビューインターフェースアプリケーション128は、例外の承認を得るため、例外について質問する、または例外についてのデータもしくはメモを提供するため、例外についてのより多くの情報を決定するため、または例外を取り扱うもしくはクローズするのに必要な手順を実施するために、電子メールまたは他のネットワーク通信を介して例外レコードを他の人々に送信してもよい。

【0077】

図8A～図8Gは、例えば、命令の品質レビューを実行するプロセスにおいて、レビュー者が例外レコードを見て操作することを可能にするために、品質レビューインターフェースアプリケーション128によって作成され得るさまざまな例示的画面表示を示す。一実施例として、図8Aは、レビュー者が、レビューする一組のさまざまな命令または例外レコードファイルのうちの1つを選択するために使用し得る画面表示500を示す。画面500は、（例えば、図2の例外レコードログデータベース126に記憶されているような）例外レコードログ170が存在するさまざまなデータソースを（例えば、ドロップダウンメニュー形式で）リスト化するセクションまたはフィールド502を含む。さらに、セクション504は、セクション502において選択されたデータソースに対して存在する一組の命令または例外レコードファイル170をリスト化し得る。したがって、図8Aの例示的画面500では、フィールド502はワークフローデータソースをリスト化し、セクション504は、ワークフローアプリケーションによって管理される命令またはバッチランに対する多数のさまざまな例外レコードファイルをリスト化している。さらにその上、セクション506は、セクション504で選択されたような特定のファイル（例えば、命令）内の例外レコードをリストし得る。ユーザは、リスト506をスクロールダウンして、例外レコード506A～506Nのうちの異なるものを選択し得る。さらにその上、画面500のセクション508は、セクション506で選択されたような例外レコード506A～Nに対する情報を提示し得る。この実施例では、ユーザは例外レコード506Bを選択し、アプリケーション128は、選択された例外レコード506Bに対する情報を例外レコードログデータベース126から取得し、その情報を、ユーザによる使用および操作のために表示している。セクション508は、名称、タイプ、重大度、例外に対す

10

20

30

40

50

る情報、および例外に対するレビューまたは取り扱い手順など、選択された例外レコードに対するさまざまな情報を含み得る。

【0078】

重要なことには、例外レコードに対して表示される情報は、例外作成時にデータソースから収集されたような、プロセスまたはデータソースのメタデータ510を含み得る。上述したように、品質レビュー管理システム112は、データソースから入力データを収集するときに、（例えば、プラグインもしくはルールを作成するときに事前定義され得る、または特定のデータソースによって提供されることが分かっているデータであり得る）一組のメタデータを収集する。概して、収集されるメタデータは、ルールを作成するときに設定されるデータ変換によって定義される。データ変換は、データソースからのメタデータに適用され、プラグインは、この変換されたメタデータを、例えばHTMLで出力し、そうして、メタデータは、レビューアプリケーションにおいて閲覧可能である。任意のプロセスまたは環境データであり得るこのデータソースメタデータは、概して、例外入力データと同時に収集され、アクセスされた例外データに関連付けられたプロセススナップショットとして、例外に対して作成された例外レコードの一部として記憶される。そのため、例外エンジン122が、収集された例外データに基づいて例外を検出すると、例外エンジン122は、データソースメタデータも例外レコードの一部として記憶する。次いで、品質レビューインターフェースアプリケーション128は、このメタデータの一部または全部を、例えば、生成された例外を報告する画面500のフィールド510内に表示し、それにより、レビュー者に、プロセス内で例外が発生した時間のプロセス内へのライブビューを提供する。したがって、この機能は、レビュー者に、例外時のさまざまなプロセスの状態／値などのクイックビューを提供し、レビュー者が、なぜ例外が発生したか、例外のコンテキストなどをより迅速に理解することを可能にし、したがって、レビュー者が、例外をより迅速に処理すること、またはどのように解決するかを知ることが可能にする。この動作により品質レビューはより容易になるが、一方、既知の品質レビュー製品は、概して、生成された例外のリストをレビュー者に提供し、レビュー者は、各例外を分析して、例外を却下できるか、または他の方法で取り扱わねばならないかを判断し、これらの既知の製品では、レビュー者は、典型的には、例外をもたらしたデータが収集されたプロセスシステムに戻って、例外時のプロセススナップショットデータを得る必要がある。画面500のメタデータフィールド510によって提供されるライブビュー機能により、レビュー者には、例外レビューを実行する際にレビュー者にとって最も有用なプロセスデータを一般的に含むプロセススナップショットが自動的に提供され、これは、レビュー者が、他のデータ収集システムに戻って、ユーザが例外のレビューおよび取り扱いを実行するのに必要な情報を得る必要はないことを意味する。この機能は、レビュー者が、例外レコードを解決するためにレビュー者が必要とするプロセスデータにアクセスするために、それらのデータ収集システムを使用することを概して必要としないので、レビュー者は、プロセスまたはデータソースのメタデータが一般的に収集される他のプロセスデータ読み出しシステムに精通している必要が概してないことも意味する。

【0079】

さらにその上、図8Aに示されるように、セクション512は、ユーザに、例外を取り扱うまたは解決する方法に関する一組のオプションまたは情報、例えば、レビューステップまたは手順、例外をクローズするために実行するアクションのチェックリスト、例外をクローズするためにさまざまなユーザによって記入または署名される必要がある承認フィールドなどを提供し得る。

【0080】

図8Bは、タイプがワークフローコメントの例外レコードに対する情報の一部を示す例示的画面表示520を示す。ここで、画面セクション506内のワークフロー例外は、選択された命令に対してリストされた唯一の例外である。画面520のセクション508に示されているように、ワークフローコメントは、ステータスが新しい、重大度が低であり、（例外を閲覧および／または処理するのに使用され得る第三者システムへの参照を定

義するのに使用され得る) 訂正および予防措置 (CAPA) 識別子を有さない、または関連付けられておらず、かつどのグループにも割り当てられていない。しかしながら、セクション 508 は、(各例外は固有の ID を有するので) 例外 ID、ソース指標、および例外の作成日/時を表示し、これらのデータのすべては例外レコードの一部として記憶されている。さらにその上、セクション 508 は、例外レコードの一部として記憶された情報でもあり得る、例外の一般的記述を含む。しかしながら、このデータは、レビューアプリケーション 128 の一部として記憶されてもよく、例外タイプに基づいてもよい。

【0081】

同様に、画面セクション 508 の下部は、この特定の例外に対して収集されたメタデータを示す。この場合、メタデータは、例外が作成されたときにワークフローアプリケーション内にユーザが見ていたものを反映しており、レビュー者が、例外が発生したときのプロセスのコンテキストを理解するのに使用され得る。

【0082】

さらにその上、図 8C の例示的畫面 530 は、例外レコードの取り扱いまたはクローズを実行するために、図 8B の画面セクション 508 内でレビュー者に提供され得る解決策フィールドのリストを提供する。ここで、解決策は、所望通りにコンフィグレーション可能であり得る(したがって、例外に対するルールを生成するために、同様の画面をコンフィグレーションアプリケーション 120 内でも使用し得る)。いずれの場合においても、図 8C に示すように、解決策フィールドは、解決策の名称またはタイプ 532 (例えば、標準) と、記述 534、1 つ以上のステージ名称 536 (例えば、製造または製造監督者) と、重大度 538 (例えば、低) と、例外をクローズまたは解決するために 1 つ以上のステージの担当者が必要とする署名 539 と、を含み得る。この場合、図 8C に表示されている例示的例外レコードの解決策は、例外をクローズまたは解決するために、例外は、(製造ステージの) 2 人のオペレータおよび(製造監督者ステージの) 1 人の監督者によって署名されなければならないことを示す。これらの取り扱い手順は、署名ボックス、チェックボックスなどの形態で、図 8A のセクションまたはフィールド 512 内で実施され得る。さらに、レビュー者は、例外におけるこれらの手順を例外ごとに変更でき得るか、または例外をグループ化して、グループに基づいて対策を講じ得る。

【0083】

図 8D は、一実施例として、選択された範囲外パラメータの例外に対する解決策フィールド 542 を有する画面セクション 508 を含む画面表示例 540 を示す。再び、ここでも、画面 540 は、ステータス(アクティブ)、重大度(低)、CAPA 識別子、および、もしあれば、グループ指定を含む、選択された例外レコードについての何らかの情報を含む。解決策セクション 542 は、レビュー者が閲覧してファイルを例外レコードに添付することを可能にするフィールド 544 (これは、後で、命令に対する例外レコードが顧客、FDA などによってレビューされるときに有用であり得る) と、レビュー者が例外レコードにコメントまたはメモを追加することを可能にするコメントフィールド 546 (ここでは、SA 運用管理者が、例外レコードのクローズを引き受けるとコメントしている) と、を含む。当然のことながら、フィールド 546 を使用して、追加コメントを追加することができる。追加的に、画面 540 の画面セクション 508 は、例外レコードをクローズするのに必要なさまざまな署名者が使用し得る署名およびクローズフィールド 548 を含む。さらにその上、画面セクション 508 は、どんなアクションがとられたか、誰がアクションをとったか、およびアクションの時間と日にちを含む、レコードを処理する間に例外レコードに対してとられたアクションを示す履歴フィールド 549 を含み得る。この情報は、レビューインターフェースアプリケーション 128 によって獲得され、例外レコードの一部として記憶され得る。

【0084】

同様に、品質レビューインターフェースアプリケーション 128 によって提示されるさまざまな画面は、命令または命令グループについての統計情報を含み得る。例えば、画面 540 のセクション 550 は、閲覧されている命令の指標 551 と、クローズされた、命

10

20

30

40

50

令に関連付けられた例外レコードの数の指標 5 5 2 と、命令の完了ステータスの指標 5 5 3（この場合、命令はまだ完了しておらず、なおも処理されている）と、を含む。セクション 5 5 0 は、レビュー者が、命令または命令の例外レコードに署名して検証するためのリンク 5 5 3 などとも提供し得る。

【0085】

同様に、図 8 E の画面 5 6 0 は、品質レビューアプリケーション 1 2 8 が、命令ごとまたはグループごとに、決定し、レビュー者に提供し得る、追加統計情報を示す。図 8 E の画面 5 6 0 は、セクション 5 6 2 が 1 1 個の例外レコードについてのさまざまな統計情報を選択された順序で表示していることを除いて、図 8 D の 5 4 0 の画面と同様である。この場合、ボックス 5 6 4 は、各ステータスカテゴリ（新しい、アクティブ、クローズ済）にある、命令に対する例外レコードの数を示す。さらに、棒グラフ 5 6 6 は、各重大度カテゴリごとの、命令に対する例外レコードの数を示す（5 つは低であり、3 つは中であり、2 つは高であり、そして 1 つは新しい）。さらにその上、棒グラフ 5 6 8 は、各例外タイプにおける命令の例外の数を示す（9 つは範囲外パラメータタイプであり、2 つはコメントタイプである）。当然のことながら、品質レビューアプリケーション 1 2 8 は、例外レコードについての他のタイプの統計情報を提供してもよく、命令グループ、命令またはグループ内のレコードのサブセットに対してなど、他の方法で統計情報を提供してもよい。

【0086】

さらにその上、品質レビューインターフェースアプリケーション 1 2 8 は、レビュー者が例外レコードのうちのさまざまなものをグループにまとめ、グループとしてのレコードのさまざまなアクション（コメントする、パラメータを変更する、承認するなど）をとることを可能にし得る。一実施例として、図 8 D および図 8 E の画面 5 4 0 および 5 6 0 の各々は、例外レコードのリスト内に、3 つのレコードグループ、すなわち第 1 のグループ、第 2 のグループおよびテスターのリスト 5 7 0 を含む。アプリケーション 1 2 8 は、例えば、例外レコードリスト 5 0 6 内の例外レコードを選択し、選択した例外レコードをリスト 5 7 0 内のグループにドラッグアンドドロップすることにより、レビュー者が例外レコードをグループに追加することを可能にし得る。当然のことながら、アプリケーション 1 2 8 は、レビュー者が、任意の好適な方法で所望通りに、新しいグループを作成すること、グループを削除することなどを可能にし得る。

【0087】

さらにその上、アプリケーション 1 2 8 は、レビュー者がグループごとに例外レコードに対して 1 つ以上のアクションをとることを可能にし得、これはレビュー者の時間を節約する。図 8 F は、レビュー者がグループアクションのためにグループテスターを選択した例示的畫面 5 8 0 を示す。この選択の結果として、アプリケーションは、グループテスター内の例外レコードのグループに対してとられる可能性のあるアクションのポップアップウィンドウ 5 8 2 を表示する。そのようなアクションには、（ドロップダウン入力ボックス 5 8 4 を介して）グループ内の例外の重大度を変更すること、（ドロップダウン入力ボックス 5 8 6 を介して）グループ内の例外レコードの CAPA 識別子を変更すること、（入力ボックス 5 8 8 を介して）グループ内の各例外レコードにコメントを加えること、（ボックス 5 9 0 を介して）グループ内の各例外レコードに文書を添付すること、およびリンク 5 9 2 を介してグループ内の例外レコードに署名してクローズすることが含まれ得る。当然のことながら、アプリケーション 1 2 8 は、レビュー者が選択された例外レコードグループに対して任意の他の 1 つまたは複数のアクションをとることを可能にし得、インターフェースデバイスを介して任意の他の方法で、そうし得る。

【0088】

所望であれば、品質レビューインターフェースアプリケーション 1 2 8 は、レビュー者が、実際にはグループに関連付けられていない例外レコードに対してグループアクションをとることを可能にし得る。例えば、図 8 G の例示的畫面表示 5 9 2 に示されるように、レビュー者は、リスト 5 0 6 内のさまざまな例外レコードのチェックボックスを選択し得る（図 8 G の実施例では、最初の 2 つの例外レコードがそのようにチェックされている）

。次いで、アプリケーション 128 は、これを、チェックされた例外レコードの両方に対してアクションをとる試みとして認識し得、図 8 F のさまざまな編集フィールドを含むポップアップ画面 594 を提供し得る。しかしながら、この場合、ポップアップボックス 594 は、図 8 G のポップアップボックス 594 の上部に示されるように、それらを既存のグループに追加すること、または新しいグループを作成することなど、選択された例外レコードに対してとることができるグループ化アクションを含み得る。

【0089】

本明細書に記載のような品質レビューインターフェースアプリケーション 128 は、図 8 A、図 8 B、および図 8 D～図 8 G のレコードリスト 506 内に提供されるような例外レコードなど、さまざまなレコードのページを、異なるページ内に重複レコードを提示すること、またはレコードページ間を移動するときに表示内のレコードを欠落させることを削減または排除する方法で、レビュー者がスクロールすることを可能にするのに有利なページングアルゴリズムまたは技術を使用し得る。概して、ブラウザを使用して、例えばネットワーク接続などを介してアクセス可能な SQL データベースなど、外部の構造化データベースに記憶されている、整理されたレコードのリストを表示する Web ベースまたはブラウザベースの表示システムは、データベース内の関連レコードを検索し、次いで、表示ページに収まる数のレコード、例えば一度に 10 個のレコードを提示する。したがって、これらのシステムは、典型的には、まず、返されたリストの最初の 10 個の位置にあるレコードを表示する。ユーザが現在表示されているリスト以外のレコードをさらに見たいときは、ブラウザは同じ検索でデータベースに連絡し、ユーザがレコードリストをスクロールダウンした場合は、次のレコードセット（例えば、次の 10 個の位置のレコード）を取得し、ユーザがレコードリストをスクロールアップした場合は、前のレコードセット（例えば、前の 10 個の位置のレコード）を取得する。当然のことながら、多数の例外またはレコードが存在する可能性があり、そこで、ユーザがレコードリストをスクロールダウンまたはスクロールアップするとき、インターフェースは多数のレコードページをスクロールせねばならない。したがって、表示システムのブラウザは、問い合わせに関連付けられたすべての関連レコードを決定し、1 番目のページを、リスト内の最初の X 個のレコードとして表示する（例えば、X が 10 の場合、レコード 1～10）。ユーザが 1 番目のページにないレコードを見たい場合、システムは 2 番目のページを、リスト内の 2 番目のセットの X 個のレコードとしてロードする（例えば、リストの位置 11～20 にあるレコード）。次いで、システムは、リストを X 個の位置だけスクロールアップまたはスクロールダウンして、前のまたは次のレコードページを提示する。システムは、連続する X 個のレコードのセットを各新しいページとしてロードするだけなので、レコードリストが静的であり頻繁に変化しないときは、このページングアルゴリズムは良好に機能する。

【0090】

しかしながら、レコードの一部のパラメータが変更されたために一部のレコードが消える、または検索から外れたとき、または新しいレコードがデータベース内に作成されたときなど、リストまたは検索中のレコードが変更されたときは、検索によって見つかった元のレコードリストは変更され、これらの変更は、レビュー者が 1 つのレコードページをレビューしている間で、次のレコードページを呼び出す前に発生する可能性がある。この状況では、レビュー者が、レコードを処理し、そしてユーザインターフェース上に表示されているようなレコードリストをスクロールダウンすると、データベース内のレコードが変更され得、したがって、ページデータが古くなる可能性がある。特に、レコードが、データベースに追加されるかもしれない、またはデータベースから削除されるかもしれない（あるいは、レコードが、検索基準に従う検索によってそのレコードはもはや返されないように変更されるパラメータを含み得る）。結果として、表示システムが後続のページに対するレコードを表示画面に表示しようとするとき、元の検索で参照されたさまざまなレコードが欠落し、見て分かるデータの損失または重複を引き起こし得る。場合によっては、ユーザは、特定の回収されたページ上のレコードの削除のために、あるページから次のページに移動するレコードを見落とすことがある。他の場合では、システムは、元の返され

10

20

30

40

50

たレコードリスト内にあった欠落レコードに基づいて、新しい表示ページをどこから開始するかについて混乱する可能性がある。さらに別の場合は、表示システムは、データベースへの新しいレコードの追加に基づいて、同じレコードを複数のページに提示する可能性がある。アプリケーションによって128提示されるような、命令に対する例外レコードログ内の例外レコードの場合のように、レビュー者がリスト内のありとあらゆるレコードを閲覧および承認することが重要であるレビューシステムでは、これらの状況はどれも望ましくない。

【0091】

図9Aおよび図9Bは、データベース内に記憶されたレコードの検索から返されたようなレコードを表示する典型的なブラウザベースの表示インターフェースであって、表示作業中に、レコードが、データベースに追加もしくはデータベースから削除される、またはそのパラメータがデータベース内で変更され得るインターフェースにおける、上記の問題を示す。特に、図9Aは、時間T1においてデータベースの検索から返された（レコードR1～RNを含む）関連レコードリスト700を示し、レコードR1～RNは、命令に対するすべてのレコードであり得るか、または命令に対するすべてのオープンレコード、特定の重大度のすべてのレコードなど、何らかの他の検索基準に基づいたある順番のレコードリストであり得る。いずれの場合においても、表示システム（例えば、ユーザインターフェースにおけるブラウザ）は、第1の時間T1に、リスト700を受信し、リスト700の最初の7つのレコードにアクセスし、これらの7つのレコードをユーザインターフェース上の表示画面704内に最初のレコードページとして提示し得る。したがって、表示画面704は、レコードR1～R7を検索からの最初のレコードページとして含む。表示システムは、ユーザが次のレコードページを要求したときなど、後の時間T2に、データベースにアクセスして、関連レコードリストを取得し得る。データベース内のレコードに変更が加えられていないと仮定すると、同じレコードリスト700が時間T2に返され、表示システムは、単に、リスト上の2番目のセットの7つの位置で見つかったレコードセット（レコードR8～R14となる）にアクセスして、ディスプレイ708内の2番目のレコードページとして表示する。図9Aに示されるこの動作は、データベースからの順序付けられたレコードリスト700が時間T1から時間T2まで静的なままである場合、良好に機能する。

【0092】

しかしながら、1番目のページ704が提示される時間T1と、2番目のページ708が提示される時間T2との間に、レコードが関連レコードリストから削除された（例えば、1つ以上のレコードが、そのパラメータを変更されて、そのレコードがレコード検索において見つけれなくなかった）場合、表示システムは、レコードの表示をうっかりしてスキップし得る。図9Bは、1番目のページ704がディスプレイ内に提示された時間T1の後で、2番目のページ708が提示される時間T2の前に、レコードR4が検索リストから削除された（または抜けた）例を示す。この場合、表示システムのブラウザは、時間T1に、図9Aのデータベース内で見つけれられたようなリスト700に基づいて、最初の7つのレコードR1～R7を取得してページ1704の内に表示する。しかしながら、表示システムのブラウザが図9Bの2番目のページ708のためのレコードにアクセスしようとする、ブラウザは、単に、返されたリスト710内の2番目のセットの7つのレコード、すなわち9Bのリスト710内の位置8～14にあるレコードにアクセスして使用する。このとき、レコードR4は削除されたので（クローズされ、関連レコードリストから削除またはドロップされた可能性がある）、1番目のページが表示されたときには図9Aのリスト700内の8番目の位置に元々あったレコードR8は、時間T1と時間T2との間のレコードR4の削除のために、今は、図9Bのリスト710内の7番目の位置にあるので、表示システムは、レコードR9～R15を2番目のレコードページとしてディスプレイ708内に表示する。そのため、レコードR8は、図9Bのディスプレイの1番目または2番目のページ704または708のいずれにも示されず、そこで、レビュー者には見落とされる。当然のことながら、ページングプロセス中にレコードがリストま

10

20

30

40

50

たはデータベースに追加されたときに同様の状況が発生するが、この場合、レコードの追加により、あるページに関連付けられた位置に元々あった特定のレコードが、後の時間には異なるページに関連付けられた位置に置かれたために、1つ以上のレコードが、重複され、または連続ページ内に提示される可能性がある。

【0093】

この状況は、変更することができるパラメータを有する、または新しいレコードを追加することができる例外レコードのページをスクロールするときには、関連レコードがディスプレイ内でスキップされ、または関連レコードがディスプレイの異なるページ内で重複される状況をもたらすので、不利である。

【0094】

しかしながら、本明細書に記載の品質レビューアプリケーション128は、ディスプレイのさまざまなページ内にどのレコードを提示するかを決定するための異なるページング技術を使用し得、この新しいページング技術は、異なるページ間を移動するときの重複レコードおよび／または欠落レコードを削減または排除する。特に、品質レビューアプリケーション128は、データベースから返されたレコードリスト内の固定もしくは事前設定位置またはレコード位置を使用してページングを実行するのではなく、代わりに、ごく最近表示されたページ内の最初のレコード、または最後のレコード、または最初と最後の両方のレコードなど、1つ以上のレコードをマークし、次いで、そのマークまたはアンカを使用して、ディスプレイの次のページの一部として表示するレコードセットを見つけ出す動的ページングアルゴリズムを使用する。このようにして、ページが表示された後に、レコードが、関連レコードリストに追加された、または関連レコードリストから削除された場合、次のページに表示されるレコードセットは、新しいページがロードされた時点で最後に表示されたレコードのうちの1つに隣接する、リスト内のレコード（例えば、前のページの最初または最後のレコード）になる。

【0095】

図10Aおよび図10Bは、品質レビューインターフェースアプリケーション128によって使用される新しいページングアルゴリズムまたは技術の動作を示す。この場合、時間T1および時間T2にデータベース検索から返されるような、図10Aおよび図10Bの関連レコードリスト700および710は、それぞれ図9Aおよび図9Bに示されたような同じ関連レコードセットを含む。図10Aに示すように、インターフェースアプリケーション128は、ディスプレイ内の最初の7つのレコードとして1番目のページ704を提示するが、1番目のページ704内に表示された最後またはボトムレコード（R7）を、現在表示されているページ内の最後またはボトムレコードを示す（アンカ1ボトムと称される）アンカでマークする。アプリケーション128はまた、ディスプレイ704内の最初またはトップレコード（R1）を、現在表示されているページ内のトップまたは最初のレコードを示す（本明細書ではアンカ1トップと称される）トップアンカでマークし得る。次いで、アプリケーション128が次のレコードページであるページ2をロードすると、アプリケーション128は、返されたレコードリストを検索して、ページ1のボトムアンカ（アンカ1ボトム）を探すが、返されたリスト700または710内のこのレコードが見つかる位置は関係ない。次いで、アプリケーション128は、リスト700内の、アンカ1ボトムというページ1のボトムアンカでマークされたレコードの直後の7つのレコードを、ページ2内に表示するレコードリストに使用する。次いで、アプリケーション128は、アンカ1ボトムというアンカを削除し得るが、次のレコードページのための追加アンカ（すなわち、アンカ2トップおよびアンカ2ボトム）をマークまたは記憶して、ディスプレイ708のページ2内に表示されたトップおよびボトムレコードをマークし得る。次いで、アプリケーション128は、ユーザがページ2のリストをスクロールダウンしたときに、アンカ2ボトムでマークされたレコードを使用して、どのレコードを次のページ（ページ3）内に表示するかを決定するか、または（例えば、ユーザがページ2のレコードリストをスクロールアップして、ページ1に戻る場合）アンカ2トップでマークされたレコードを使用して、どのレコードを前のページ内に表示するかを決定する。

10

20

30

40

50

理解されるように、このページング技術は、レコードにアンカまたはマーカを付けるか、現在の表示ページのさまざまな位置にあるレコードを示すアンカまたはマーカを記憶し、これらのアンカを使用して、返されたリスト内の実際のレコードの位置に関係なく、返されたレコードセット内のどのレコードを次のページの最初（または最後）のレコードに使用するかを決定する。そのため、レコードは、図 9 A および図 9 B に示された方法におけるページング表示を混乱させることなく、異なるページ表示間で、データベースまたは返されたリスト 700 内で、位置が上下に動くことができる。

【0096】

一実施例として、図 10 A は、時間 T1 と時間 T2 との間にデータベースに変更が加えられず、1 番目のページ 704 内のレコード R1 ~ R7 および 2 番目のページ 708 内のレコード R8 ~ R14 が提示される結果となるときにアンカの使用の動作を示す。図 10 B は、これらのアンカを使用して、返されたレコードリストが図 10 A のリスト 700 である時間 T1 と、返されたレコードリストが図 10 B のリスト 710 である時間 T2 とにページングを実行するときの、レコードリスト 710 内のアンカの使用の動作を示す。特に、ページ 1 の表示 704 は、ページ 1 が作成された時間 T1 における図 10 A のデータベース内のレコードリスト 700 の状態に基づくレコード R1 ~ R7 を含む。しかしながら、アプリケーション 128 は、ページ 1 をロードすると、レコード R1 をアンカ 1 トップというアンカでマークし、レコード R7 をアンカ 1 ボトムというアンカでマークした。次に、時間 T2 において、表示アプリケーション 128 が、ページ 2 を表示するために（レコード R4 はもはやリストにないので今や変更されている）レコードリスト 710 を取得したとき、アプリケーション 128 は、返されたリスト 710 を検索してアンカ 1 ボトムレコードとしてマークされたレコードを探し、リスト 710 内でこのレコードを見つけると、単に、次の 7 つのレコード（この場合、レコード R8 ~ R14）を、レコード R8 ~ R14 はリスト 710 内の 7 番目 ~ 13 番目の位置にあるにも関わらず、ページ 2 のディスプレイ 708 としてロードして表示する。当然のことながら、アプリケーション 128 は、ページごとに任意の数のアンカをマークすることができるが、概して、現在のページ内のトップおよびボトムのレコードを識別するためにページごとに 2 つのアンカを有し、次いで、後続のページをロードするときに、それらのアンカに隣接するまたは隣り合うレコードをロードする。2 つのアンカの使用は、リスト内のレコードの状態が変化した（そのようなレコードがリストに追加されたまたはリストから削除された）ときに、ディスプレイ内の欠落レコードまたは異なるページ内の重複レコードを排除または削減する方法で、アプリケーション 128 がページをスクロールアップまたはスクロールダウンすることを可能にする。

【0097】

アプリケーション 128 は、特定のページが表示されている間に、（ブラウザまたはブラウザデバイス内などの）トップおよびボトムアンカ位置にあるレコードの、別個のアンカ変数での指標を記憶すること、現在表示されているレコードページ内のあるアンカ位置に現在あるレコードに関連付けられたマーカをデータベース内に実際に記憶すること、現在のページ内の 1 つ以上のレコードをマーカとして記憶すること、一時的マーカをデータベース内のマークされたレコードの直前または直後に記憶するなど、任意のマーキング技術を使用することができる。追加的に、マーカまたはアンカは、アプリケーション 128、アプリケーション 128 によって使用されるブラウザ、レコードが記憶されるデータベースなどに記憶することができる。さらに、アプリケーション 128 は、新しいページがアクセスされ表示されると、アンカ位置またはマーカを移動または変更する。さらにその上、アプリケーション 128 は、レコードに対するアンカとして任意の値を使用し得る。例えば、アプリケーション 128 は、レコードの時間／日付スタンプなど、アンカがアンカとして位置するレコードのパラメータを使用し得る。場合によっては、レコードに対するアンカは、レコードの検索パラメータもしくは検索パラメータの何らかの組み合わせに基づくかまたはそれらを示してもよく、またはレコード名、識別番号など、レコードの何らかの他の固有の値であってもよい。同様に、場合によっては、マーカに関連付けられた

レコードは、関連検索リストから外れるレコードであり得る。この場合、アプリケーション１２８は、（新しいページをロードするときに）新しいリスト内でアンカレコードが見つからないと、アンカを、現在表示されているページ内の、現在欠落しているアンカレコードに隣接するレコード（例えば、現在表示されているページ内の、ボトムアンカのすぐ上もしくは直前のレコード、または現在表示されているページ内の、トップアンカのすぐ下もしくは直後のレコード）に変更し、次いで、新しいアンカを使用して、返されたリスト内の、新しい表示ページを提示するためのレコードセットを決定する。

【００９８】

したがって、品質レビューインターフェースアプリケーション１２８によって使用されるページングアルゴリズムは、この新しいシステムは、返されたリスト内のレコードの位置に関係なく、前のページまたは画面に表示された最後のレコードから開始してレコードをダウンロードして新しい表示画面に提示するという点で、過去のページングシステムよりも良好に動作する。すなわち、元の回収レコードリスト内のレコード位置に関連付けられた新しい表示ページを提示する代わりに、システム１２８は、（リストをページダウンするときは）データベースにアクセスして、新しいリスト内で、現在表示されているページ上の最後のレコードの直後に現在存在しているレコードを探し、それにより、新しいレコードセットが画面上に表示される都度、最初に表示される新しいレコードは、新しい検索リスト（７１０）内で、前の検索（７００）からの前のページ内で表示されていた最後のレコードに直ぐ続くレコードである。この表示システムは、データベース内のレコードに変更が行われている間の長時間にわたって、ユーザがさまざまなレコードページをスクロールするレビュープロセスにおいて、レコードが順不同で表示されないこと、レコードが複数の表示ページ内で重複しないこと、およびレコードが欠落しないことを保証する助けとなるので、より正確である。さらに、レビューアプリケーション１２８は、ユーザがレコードをスクロールしているときに、データベースにアクセスして、新しい表示ページに収まる数の新しいレコードを探す必要があるだけであるが、一方、ユーザがスクロール処理を実行するときに関連レコードのすべてが確実にアクセスされることをなおも保証するので、より効率的である。

【００９９】

本明細書に記載の品質レビュー管理アプリケーションおよびバッチ実行エンジン、サーバアプリケーション、プラグインなどは、任意の所望の加工プラント環境において使用および実施することができ、任意の所望のタイプの加工プラント制御システム通信プロトコルを使用する任意の加工プラント制御システムで使用し得ることを理解されたい。本明細書に記載のアプリケーションおよびルーチンは、好ましくは、例えば、サーバ、ワークステーション、ハンドヘルドデバイス、または他のコンピュータに記憶されたソフトウェアにおいて実装されるが、これらのルーチンは、所望により、代替的または追加的に、ハードウェア、ファームウェア、特定用途向け集積回路、プログラマブル論理回路などにおいて実装され得る。ソフトウェアで実装される場合、ルーチンまたはアプリケーションは、磁気ディスク、レーザーディスク（登録商標）、EPROMまたはEEPROM、ソリッドステートまたはその他の記憶媒体などのコンピュータ読み取り可能なメモリ、コンピュータのRAMまたはROM、ハンドヘルドデバイスなどに記憶されてもよい。同様に、このソフトウェアは、例えば、電話線、インターネットなどの通信チャネルを介して、コンピュータ可読ディスクなどの可搬型媒体上で、などを含む、任意の既知の、または所望の配信方法を介してユーザまたはデバイスに配信され得る。

【０１００】

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、これらの例は例解的であるに過ぎず、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、本発明の趣旨およ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。

10

20

30

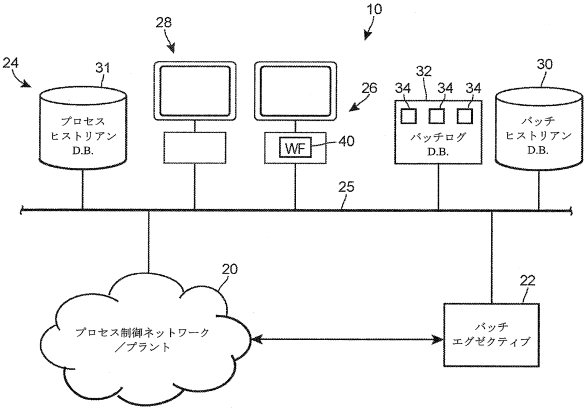
40

50

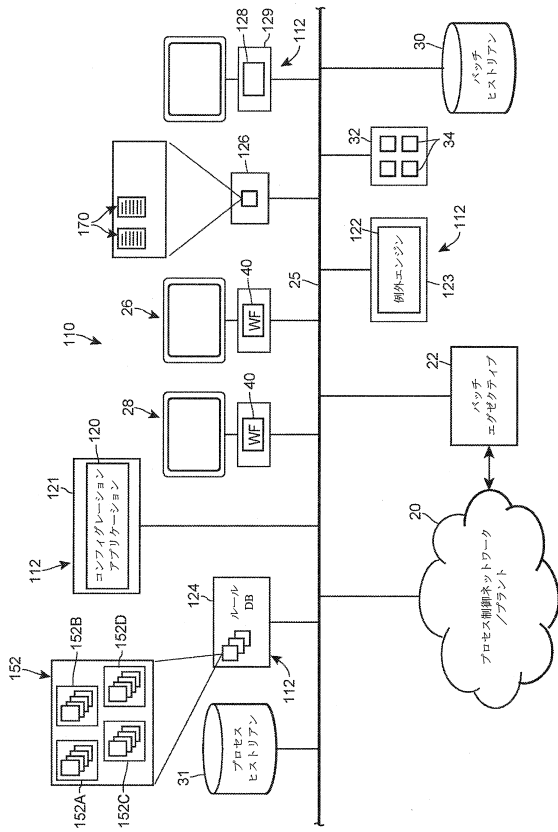
【図面】

【図 1】

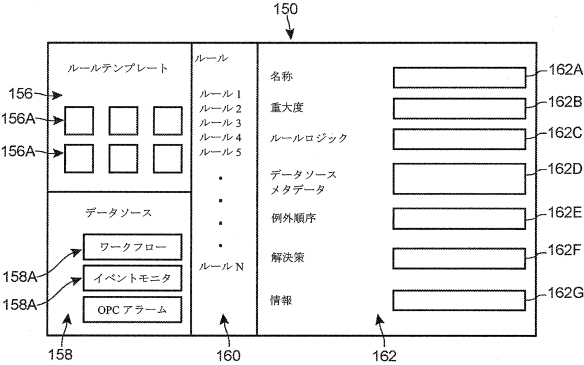
(先行技術)



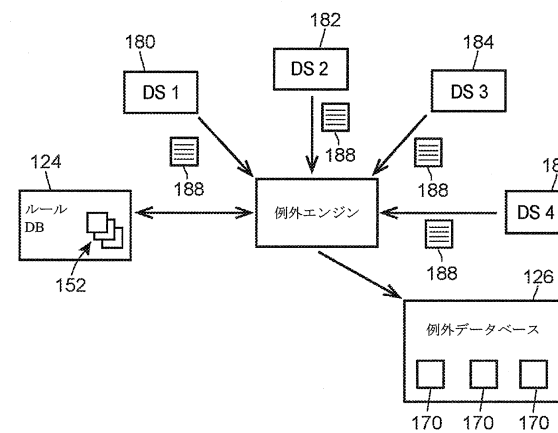
【図 2】



【図 3】



【図 4】



10

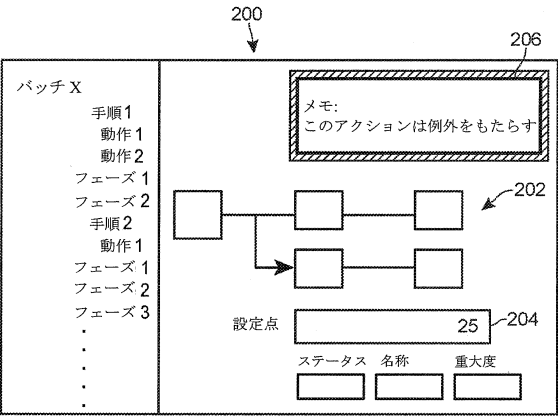
20

30

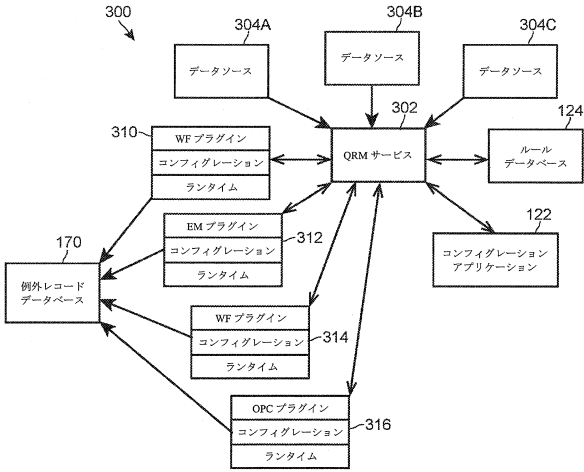
40

50

【図 5】

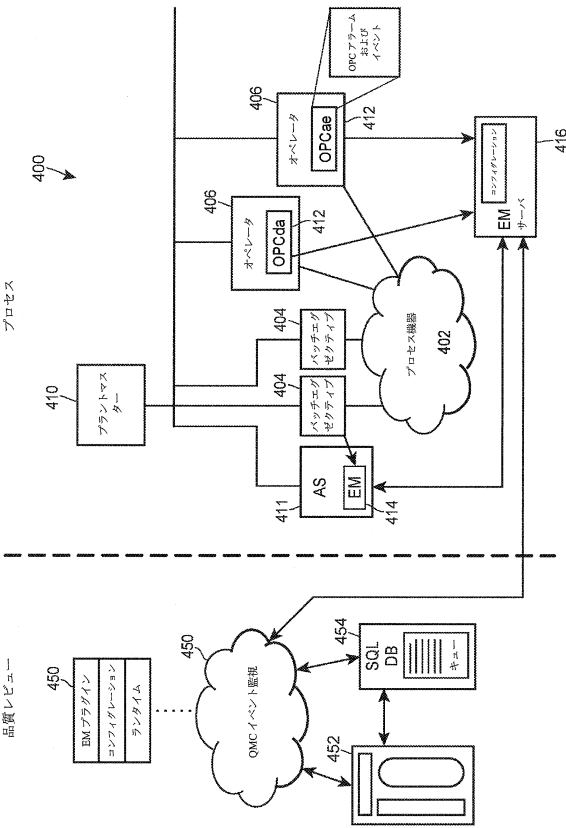


【図 6】

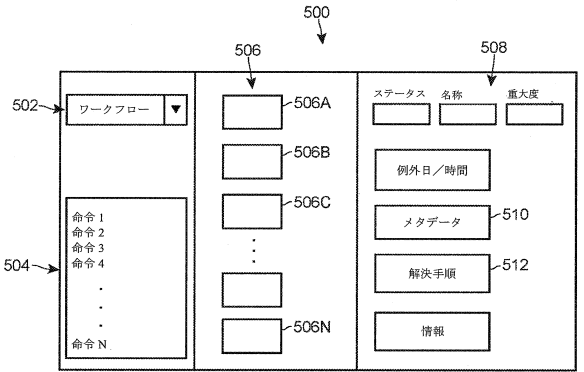


10

【図 7】



【図 8 A】



20

30

40

50

【図 8 B】

品質レビューマネージャ 概要 命令 530 506 508 運用管理者サム・ジョーンズ

パラメータが範囲外である
パラメータ 1036
ソース ワークフロー 2018-01-11 04:05:01
作成 2018-01-11 04:05:01
作成 2018-01-11 04:05:01

例外を管理する
合計 11 個の例外のうち 0 個が
クローズされた
命令を完了させる
1 個の関連ワークフローのうち
0 個が完了した
検証および署名
QA レビュー開始
LRD 20180226
QA レビュー承認

ワークフローコメント
ワークフロー 2018-02-28 02:24:27
作成 2018-02-28 02:24:27

パラメータに対する基本的表示オプション
WL_DISPLAY_OPTIONS.1
反復 1
ペアラント記述 MOP、単純 RA レンビ
この命令は、パラメータに対する基本的表示オプションを示す。
反復 1、パラメータの値を表示する。手動の入力/編集は許可されない。
任意選択 - 次の例外に適用するために使用される。必要である。
必須 - 命令を完了させるには、値が必要である。
任意しない - 命令が強制または取り消されなければ、表示しない。
1 個の関連ワークフローのうち 0 個が完了した。
必要でないパラメータの値を省略しようとする。入力に許可されない。
必要である。
必要でないパラメータの値を省略しようとする。必要でないパラメータに適用してエラー
を実行可能である。

【図 8 C】

品質レビューマネージャ 概要 命令 530 506 508 運用管理者サム・ジョーンズ

パラメータが範囲外である
パラメータ 1036
ソース ワークフロー 2018-01-11 04:05:01
作成 2018-01-11 04:05:01
作成 2018-01-11 04:05:01

例外を管理する
合計 11 個の例外のうち 0 個が
クローズされた
命令を完了させる
1 個の関連ワークフローのうち
0 個が完了した
検証および署名
QA レビュー開始
LRD 20180226
QA レビュー承認

ワークフローコメント
ワークフロー 2018-02-28 02:24:27
作成 2018-02-28 02:24:27

パラメータに対する基本的表示オプション
WL_DISPLAY_OPTIONS.1
反復 1
ペアラント記述 MOP、単純 RA レンビ
この命令は、パラメータに対する基本的表示オプションを示す。
反復 1、パラメータの値を表示する。手動の入力/編集は許可されない。
任意選択 - 次の例外に適用するために使用される。必要である。
必須 - 命令を完了させるには、値が必要である。
任意しない - 命令が強制または取り消されなければ、表示しない。
1 個の関連ワークフローのうち 0 個が完了した。
必要でないパラメータの値を省略しようとする。入力に許可されない。
必要である。
必要でないパラメータの値を省略しようとする。必要でないパラメータに適用してエラー
を実行可能である。

【図 8 D】

品質レビューマネージャ 概要 命令 540 506 508 運用管理者サム・ジョーンズ

パラメータが範囲外である
パラメータ 1036
ソース ワークフロー 2018-01-11 04:05:01
作成 2018-01-11 04:05:01
作成 2018-01-11 04:05:01

例外を管理する
合計 11 個の例外のうち 0 個が
クローズされた
命令を完了させる
1 個の関連ワークフローのうち
0 個が完了した
検証および署名
QA レビュー開始
LRD 20180226
QA レビュー承認

ワークフローコメント
ワークフロー 2018-02-28 02:24:27
作成 2018-02-28 02:24:27

パラメータに対する基本的表示オプション
WL_DISPLAY_OPTIONS.1
反復 1
ペアラント記述 MOP、単純 RA レンビ
この命令は、パラメータに対する基本的表示オプションを示す。
反復 1、パラメータの値を表示する。手動の入力/編集は許可されない。
任意選択 - 次の例外に適用するために使用される。必要である。
必須 - 命令を完了させるには、値が必要である。
任意しない - 命令が強制または取り消されなければ、表示しない。
1 個の関連ワークフローのうち 0 個が完了した。
必要でないパラメータの値を省略しようとする。入力に許可されない。
必要である。
必要でないパラメータの値を省略しようとする。必要でないパラメータに適用してエラー
を実行可能である。

【図 8 E】

品質レビューマネージャ 概要 命令 570 590 506 運用管理者サム・ジョーンズ

パラメータが範囲外である
パラメータ 1036
ソース ワークフロー 2018-01-11 04:05:01
作成 2018-01-11 04:05:01
作成 2018-01-11 04:05:01

例外を管理する
合計 11 個の例外のうち 0 個が
クローズされた
命令を完了させる
1 個の関連ワークフローのうち
0 個が完了した
検証および署名
QA レビュー開始
LRD 20180226
QA レビュー承認

ワークフローコメント
ワークフロー 2018-02-28 02:24:27
作成 2018-02-28 02:24:27

パラメータに対する基本的表示オプション
WL_DISPLAY_OPTIONS.1
反復 1
ペアラント記述 MOP、単純 RA レンビ
この命令は、パラメータに対する基本的表示オプションを示す。
反復 1、パラメータの値を表示する。手動の入力/編集は許可されない。
任意選択 - 次の例外に適用するために使用される。必要である。
必須 - 命令を完了させるには、値が必要である。
任意しない - 命令が強制または取り消されなければ、表示しない。
1 個の関連ワークフローのうち 0 個が完了した。
必要でないパラメータの値を省略しようとする。入力に許可されない。
必要である。
必要でないパラメータの値を省略しようとする。必要でないパラメータに適用してエラー
を実行可能である。

10

20

30

40

50

【図 8 F】

580

品質レビューマナーページ

命令

命令1
BOM-1
リリース済

例外を管理する
合計11個の例外のうち1個がクロ
ーズされた
9%

命令を完了させる
1個の関連ワークフローのうち0個
が完了した
0%

検証および署名
オペレーター

詳細

570 更新済、作成日
第1のグループ
第2のグループ
これを使用する
テスト
これはテストグループである

582 重大度
各例外の重大度を設定する
584
CAPA
各例外に関連付けられたCAPAを設定する
586
コメント
各例外にコメントを追加する
588
添付
各例外に添付を追加する
590
署名およびクローズ
署名をクローズするためには、要求されるすべての
署名が収集される必要がある

592

506

【図 8 G】

592

品質レビューマナーページ

命令

命令1
BOM-1
リリース済

例外を管理する
合計11個の例外のうち1個がクロ
ーズされた
9%

命令を完了させる
1個の関連ワークフローのうち0個
が完了した
0%

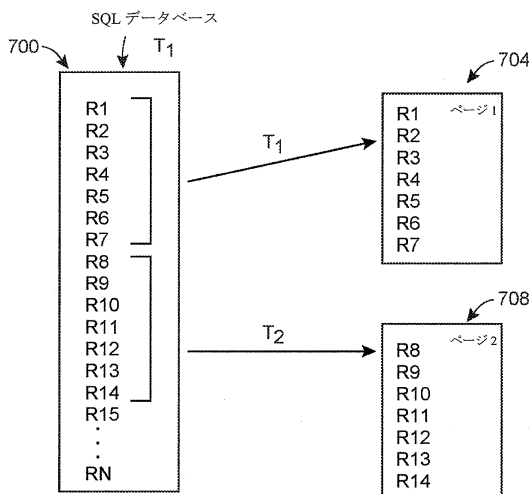
検証および署名
オペレーター

詳細

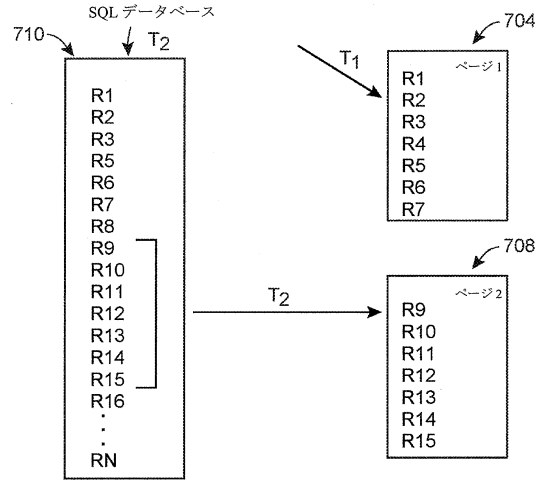
594 グループ
CAPA
コメント
添付
署名およびクローズ

506

【図 9 A】



【図 9 B】



10

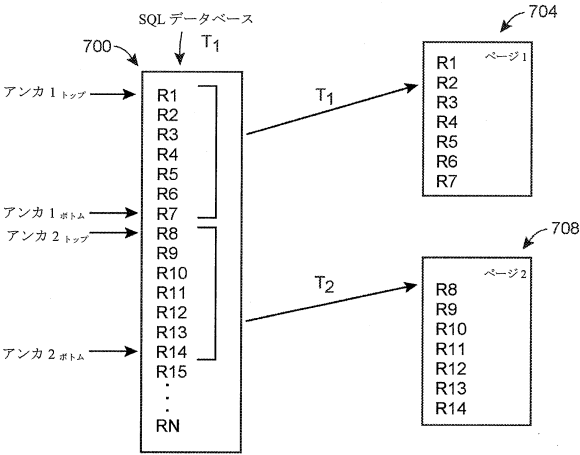
20

30

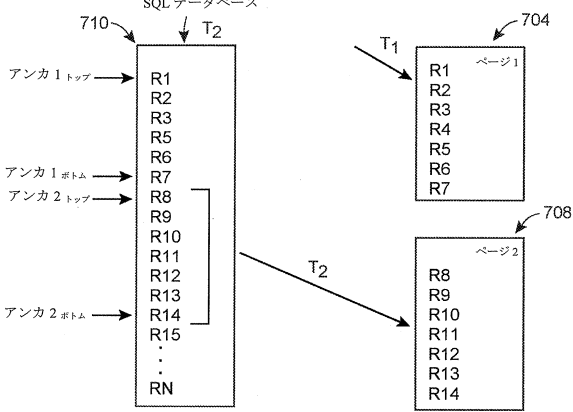
40

50

【図 10 A】



【図 10 B】



10

20

30

40

50

フロントページの続き

ーチャード
アメリカ合衆国 フロリダ州 3 4 2 2 9 オスプレイ オーク リベラ コート 6 1 9
(72)発明者 トッド・エー・クック
アメリカ合衆国 フロリダ州 3 3 7 0 1 セント・ピーターズバーグ 5号 7番ストリート 4 0 1
審査官 石田 宏之
(56)参考文献 特許第5 6 4 9 3 3 4 (J P, B 2)
特許第6 5 9 5 9 8 0 (J P, B 2)
特開2 0 0 8 - 0 4 6 7 4 6 (J P, A)
米国特許出願公開第2 0 2 0 / 0 1 2 6 1 6 7 (U S, A 1)
(58)調査した分野 (Int.Cl., D B名)
G 0 5 B 1 9 / 4 1 8