(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4672663号
(P4672663)
(45)発行日　平成23年4月20日(2011.4.20)　(24)登録日　平成23年1月28日(2011.1.28)

(51)Int.Cl.　F I
G06Q 10/00 (2006.01)　G06F 17/60 162C

請求項の数 37 (全 34 頁)

(21)出願番号	特願2006-526859 (P2006-526859)
(86)(22)出願日	平成16年1月16日 (2004.1.16)
(65)公表番号	特表2007-518145 (P2007-518145A)
(43)公表日	平成19年7月5日 (2007.7.5)
(86)国際出願番号	PCT/US2004/001205
(87)国際公開番号	WO2005/038544
(87)国際公開日	平成17年4月28日 (2005.4.28)
審査請求日	平成18年12月18日 (2006.12.18)
(31)優先権主張番号	10/666,446
(32)優先日	平成15年9月19日 (2003.9.19)
(33)優先権主張国	米国(US)

前置審査

(73)特許権者　594120847
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　７８７５９　テキサス　オースティン　リサーチ　パーク　プラザ　ビルディング　１１１　リサーチ　ブルーバード　１２３０１
(74)代理人　100079049
弁理士　中島　淳
(74)代理人　100084995
弁理士　加藤　和詳
(74)代理人　100085279
弁理士　西元　勝一

最終頁に続く

(54)【発明の名称】プロセス制御及び安全システムのソフトウェアオブジェクトの承認のための統合型電子署名

(57)【特許請求の範囲】

【請求項１】

ソフトウェアオブジェクト設計環境及び承認手段を含む安全計装システムにおいて用いられるソフトウェアオブジェクト承認方法であって、

前記承認手段が、前記ソフトウェアオブジェクト設計環境におけるソフトウェアオブジェクトに対する変更を検出し、前記検出した変更に基づき、前記安全計装システム内において前記ソフトウェアオブジェクトを実装する前に承認を得ることが必要な１グループのエンティティを選択し、前記１グループのエンティティを表す電子的識別情報を予め格納されたルックアップテーブルから取得することと、

前記承認手段が、前記１グループのエンティティを表す電子的識別情報に基づき、前記１グループのエンティティ内の各エンティティに前記ソフトウェアオブジェクトの検査の要求を電子的に送信することと、

前記承認手段が、前記１グループのエンティティ内の各エンティティから前記ソフトウェアオブジェクトの承認又は不承認に関する電子的な標示を受け取ることと、

前記承認手段が、前記１グループのエンティティ内の各エンティティから前記ソフトウェアオブジェクトを承認する電子的な標示を受け取るまで、前記安全計装システムにおける前記ソフトウェアオブジェクトの実装を防止することと、

を含み、

前記承認手段が前記１グループのエンティティを選択することが、前記ソフトウェアオブジェクト設計環境に入力された、前記ソフトウェアオブジェクトに個別に関連するリス

ク低減ファクタを判定し、該リスク低減ファクタに基づき前記１グループのエンティティを選択することを含むことを特徴とする方法。

【請求項２】

前記承認手段が前記ソフトウェアオブジェクトの検査の要求を電子的に送信することが、前記１グループのエンティティ内の各エンティティに通信ネットワークを介して電子的に通知することを含むことを特徴とする、請求項１記載の方法。

【請求項３】

前記承認手段が前記要求を電子的に送信することが、前記１グループのエンティティ内の各エンティティに電子メールメッセージを送ることを含むことを特徴とする、請求項２記載の方法。

【請求項４】

前記承認手段が、前記１グループのエンティティ内の各エンティティが前記ソフトウェアオブジェクトを承認する電子的な標示を受け取った場合には、前記ソフトウェアオブジェクトを前記安全計装システムにダウンロードすることを可能にすることを含むことを特徴とする、請求項１～３のいずれか一項に記載の方法。

【請求項５】

前記承認手段が前記１グループのエンティティを選択することが、前記リスク低減ファクタから安全計装レベルを判定し、判定された該安全計装レベルに基づき前記１グループのエンティティを選択することを含むことを特徴とする、請求項１記載の方法。

【請求項６】

前記承認手段が前記１グループのエンティティを表す電子的識別情報を取得することが、前記安全計装レベルから前記１グループのエンティティ内の人の数を判定することを含むことを特徴とする、請求項５記載の方法。

【請求項７】

前記承認手段が前記１グループのエンティティを表す電子的識別情報を取得することが、前記安全計装レベルから前記１グループのエンティティ内の人の職位を判定することを含むことを特徴とする、請求項５記載の方法。

【請求項８】

前記承認手段が、前記１グループのエンティティ内の１つ以上のエンティティから受け取った、前記ソフトウェアオブジェクトの承認又は不承認に関する前記電子的な標示をログに記録することを更に含むことを特徴とする、請求項１～７のいずれか一項に記載の方法。

【請求項９】

前記承認手段が、承認を得ることが必要な前記１グループのエンティティを選択することが、設計者に前記電子的識別情報の選択を催促することを含むことを特徴とする、請求項１～８のいずれか一項に記載の方法。

【請求項１０】

前記承認手段が、承認を得ることが必要な前記１グループのエンティティを表す電子的識別情報を取得することが、前記１グループのエンティティ内の人の数を判定することを含むことを特徴とする、請求項１記載の方法。

【請求項１１】

前記承認手段が前記ソフトウェアオブジェクトの実装を防止することが、前記ソフトウェアオブジェクトがプロセス制御システムの制御環境にダウンロードされることを防止することを含むことを特徴とする、請求項１～１０のいずれか一項に記載の方法。

【請求項１２】

前記承認手段が、オーバーライドキーを予め格納し、前記１グループのエンティティ内の各エンティティから前記ソフトウェアオブジェクトを承認する前記電子的な標示を受け取る前に、ユーザが前記オーバーライドキーを入力することにより前記ソフトウェアオブジェクトを前記安全計装システムにダウンロードすることを可能にすることを含むことを特徴とする、請求項１～１１のいずれか一項に記載の方法。

【請求項１３】
前記承認手段が前記ソフトウェアオブジェクト設計環境におけるソフトウェアオブジェクトに対する変更を検出することが、前記ソフトウェアオブジェクト設計環境に含まれると共にソフトウェアオブジェクトのコンフィグレーション情報を格納したコンフィグレーションデータベースをモニタすることを含む、請求項１～１２のいずれか一項に記載の方法。
【請求項１４】
前記承認手段が前記ソフトウェアオブジェクト設計環境における前記ソフトウェアオブジェクトに対する変更を検出することが、前記ソフトウェアオブジェクトに対する変更が行われた場合に前記ソフトウェアオブジェクトと関連付けられたバージョン番号を変更することを含むことを特徴とする、請求項１～１３記載の方法。
【請求項１５】
前記承認手段が前記ソフトウェアオブジェクト設計環境における前記ソフトウェアオブジェクトに対する変更を検出することが、前記ソフトウェアオブジェクト設計環境において新たなソフトウェアオブジェクトが作成されたときを検出することを含むことを特徴とする、請求項１～１３記載の方法。
【請求項１６】
前記承認手段が前記ソフトウェアオブジェクト設計環境における前記ソフトウェアオブジェクトに対する変更を検出することが、前記ソフトウェア設計環境においてユーザが承認手順を開始したときを検出することを含むことを特徴とする、請求項１～１３記載の方法。
【請求項１７】
前記承認手段が、前記ソフトウェアオブジェクトを承認する電子的な標示を受け取った場合、前記ソフトウェアオブジェクトに対するテストが期限を過ぎたときを判定するために前記ソフトウェアオブジェクトをモニタすることを更に含むことを特徴とする、請求項１～１６のいずれか一項に記載の方法。
【請求項１８】
前記承認手段が、前記ソフトウェアオブジェクトに対するテストが期限を過ぎたと判定した場合、前記ソフトウェアオブジェクトに対する新たなリスク低減ファクタを計算し、前記ソフトウェアオブジェクトに対する前記新たなリスク低減ファクタと元のリスク低減ファクタとを比較することを含むことを特徴とする、請求項１７記載の方法。
【請求項１９】
前記承認手段が、前記ソフトウェアオブジェクトに対する前記テストが期限を過ぎたと判定した場合に、ユーザに送られるアラーム信号を生成することを更に含むことを特徴とする、請求項１７記載の方法。
【請求項２０】
前記承認手段が、前記ソフトウェアオブジェクトに対する前記テストが期限を過ぎたと判定した場合に、ユーザに送られる作業命令を生成することを更に含むことを特徴とする、請求項１７記載の方法。
【請求項２１】
ソフトウェアオブジェクト設計環境と、承認手段と、プロセッサとを含むプロセス制御システムにおいて用いられるソフトウェアオブジェクト承認システムであって、
コンピュータ読み取り可能媒体と、
前記コンピュータ読み取り可能媒体に格納されたソフトウェアと、
を含み、
前記ソフトウェアが前記プロセッサにより実行されることにより、
前記承認手段が、前記ソフトウェアオブジェクト設計環境におけるソフトウェアオブジェクトに対する変更を検出し、前記変更に基づいて、プロセス制御システム内において前記ソフトウェアオブジェクトをオンラインで実装する前に承認を得ることが必要な１グループのエンティティを選択し、前記１グループのエンティティを表す電子的識別情報を予

め格納されたルックアップテーブルから取得し、
前記承認手段が、前記１グループのエンティティを表す電子的識別情報に基づき、前記１グループのエンティティ内の各エンティティに前記ソフトウェアオブジェクトの検査の要求を電子的に送信し、
前記承認手段が、前記１グループのエンティティ内の各エンティティから前記ソフトウェアオブジェクトの承認又は不承認に関する電子的な標示を受け取り、
前記承認手段が、前記１グループのエンティティ内の各エンティティから前記ソフトウェアオブジェクトを承認する電子的な標示を受け取るまで、前記プロセス制御システムにおける前記ソフトウェアオブジェクトの実装を防止する
ように、前記プロセス制御システムを動作させ、
前記承認手段が、前記ソフトウェアオブジェクト設計環境に入力された、前記ソフトウェアオブジェクトに個別に関連するリスク低減ファクタを判定し、該リスク低減ファクタに基づき前記１グループのエンティティを選択することによって、前記１グループのエンティティを表す電子的識別情報を取得する
ことを特徴とする、ソフトウェアオブジェクト承認システム。
【請求項２２】
前記承認手段が、前記１グループのエンティティ内の各エンティティに電子メールメッセージを送ることによって、前記要求を電子的に送信することを特徴とする、請求項２１記載のソフトウェアオブジェクト承認システム。
【請求項２３】
前記承認手段が、前記リスク低減ファクタから安全計装レベルを判定し、判定された該安全計装レベルに基づき前記１グループのエンティティを選択することによって、前記１グループのエンティティを表す電子的識別情報を取得することを特徴とする、請求項２１記載のソフトウェアオブジェクト承認システム。
【請求項２４】
前記承認手段が、前記安全計装レベルに基づき前記１グループのエンティティ内の人の数を判定することによって、前記１グループのエンティティを表す電子的識別情報を取得することを特徴とする、請求項２３記載のソフトウェアオブジェクト承認システム。
【請求項２５】
前記承認手段が、前記安全計装レベルに基づき前記１グループのエンティティ内の人の職位を判定することによって、前記１グループのエンティティを表す電子的識別情報を取得することを特徴とする、請求項２３記載のソフトウェアオブジェクト承認システム。
【請求項２６】
前記承認手段が、前記１グループのエンティティ内の１つ以上のエンティティから受け取った、前記ソフトウェアオブジェクトの承認又は不承認に関する前記前記電子的な標示のログを記録することを特徴とする、請求項２１～２５のいずれか一項に記載のソフトウェアオブジェクト承認システム。
【請求項２７】
前記承認手段が、設計者に前記電子的識別情報の選択を催促することによって、承認を得ることが必要な前記１グループのエンティティを表す電子的識別情報を取得することを特徴とする、請求項２１～２６のいずれか一項に記載のソフトウェアオブジェクト承認システム。
【請求項２８】
前記承認手段が前記ソフトウェアオブジェクト設計環境におけるソフトウェアオブジェクトに対する変更を検出することが、前記ソフトウェアオブジェクト設計環境に含まれると共にソフトウェアオブジェクトのコンフィグレーション情報を格納したコンフィグレーションデータベースをモニタすることを含む、請求項２１～２７のいずれか一項に記載の方法。
【請求項２９】
前記承認手段が、前記ソフトウェアオブジェクト設計環境において新たなソフトウェア

オブジェクトが作成されたときを検出することによって、前記ソフトウェアオブジェクト設計環境における前記ソフトウェアオブジェクトに対する変更を検出することを特徴とする、請求項２１～２８記載のソフトウェアオブジェクト承認システム。

【請求項３０】

前記承認手段が、前記ソフトウェア設計環境においてユーザが承認手順を開始したときに、前記ソフトウェアオブジェクトに対する変更を検出することを特徴とする、請求項２１～２８記載のソフトウェアオブジェクト承認システム。

【請求項３１】

前記承認手段が、前記ソフトウェアオブジェクト設計環境における前記ソフトウェアオブジェクトに対する変更を検出した場合に、前記ソフトウェアが、前記ソフトウェアオブジェクトと関連付けられたバージョン番号を変更することを特徴とする、請求項２１～２８記載のソフトウェアオブジェクト承認システム。

【請求項３２】

前記承認手段が、前記ソフトウェアがオーバーライドキーを予め格納し、前記１グループのエンティティ内の各エンティティから前記ソフトウェアオブジェクトを承認する前記電子的な標示を受け取る前に、ユーザが前記オーバーライドキーを入力することにより、前記ソフトウェアオブジェクトを前記プロセス制御システムにダウンロードすることを可能にすることを特徴とする、請求項２１記載のソフトウェアオブジェクト承認システム。

【請求項３３】

前記前記承認手段が、前記ソフトウェアオブジェクトを承認する電子的な標示を受け取った場合に、前記ソフトウェアオブジェクトに対するテストが期限を過ぎたときを判定するために前記ソフトウェアオブジェクトを更にモニタすることを特徴とする、請求項２１記載のソフトウェアオブジェクト承認システム。

【請求項３４】

前記承認手段が、前記ソフトウェアオブジェクトに対するテストが期限を過ぎた場合に、前記ソフトウェアオブジェクトに対する新たなリスク低減ファクタを計算し、前記ソフトウェアオブジェクトに対する前記新たなリスク低減ファクタと元のリスク低減ファクタとを比較することを特徴とする、請求項３３記載のソフトウェアオブジェクト承認システム。

【請求項３５】

前記承認手段が、前記ソフトウェアオブジェクトに対するテストが期限を過ぎたと判定された場合に、ユーザに送られるアラーム信号を更に生成することを特徴とする、請求項３４記載のソフトウェアオブジェクト承認システム。

【請求項３６】

前記承認手段が、前記ソフトウェアオブジェクトに対するテストが期限を過ぎたと判定された場合に、ユーザに送られる作業命令であって前記ソフトウェアオブジェクトに対して行われるべきテストを特定する作業命令を更に生成することを特徴とする、請求項３４記載のソフトウェアオブジェクト承認システム。

【請求項３７】

前記ソフトウェアオブジェクトが、前記プロセス制御システムにおいて安全性手順を実装するために用いられる安全システムソフトウェアオブジェクトであることを特徴とする、請求項２１記載のソフトウェアオブジェクト承認システム。

【発明の詳細な説明】

【技術分野】

【０００１】

関連出願

本願は、２００２年８月２日に出願された「プロセス制御システムのソフトウェアオブジェクトの承認のための統合型電子署名（Integrated Electronic Signatures for Approval of Process Control System Software Objects）」という名称の同時係属中の米国特許出願第１０／２１１，９０３号による優先権を主張する一部継続出願であり、その全開

示を本願明細書に参照することにより明示的に組み込む。

【０００２】

本願は、プロセス制御システムに関し、具体的には、プロセス制御システムにおいて用いられるソフトウェアオブジェクトの承認に関する。

【背景技術】

【０００３】

プロセス制御システムは、一般的に、或る製造プロセス又は他の制御プロセスを実行するために用いられる何セットもの機器を含む。この何セットもの機器はコントローラに接続され、コントローラは、製造プロセス又は制御プロセスを達成するために機器を何らかの方法で操作するための、プロセス制御ソフトウェアの命令を含む。プロセス制御ソフトウェアは、様々な制御機能のいずれかを実行するためにコントローラ（又は他のコンピュータ）上で実行されるソフトウェアオブジェクトとして実装され得る。例えば、或るプロセス制御の状況においては、ソフトウェアオブジェクトは、一般的に様々なタイプのプロセス工程に関する複数の異なるフェーズを実装するよう構成され得る。具体的には、混合フェーズを実装するソフトウェアオブジェクトは、プロセスの混合工程を行うハードウェアと関連付けられ得る。１つのフェーズを実装する各ソフトウェアオブジェクトは、何らかの個別の機能を実行すると共に、より複雑な制御手順を実装するために他のオブジェクトと通信することが理解されよう。

【０００４】

例えば、バッチ処理で用いられる制御手順を定義又は作成する際には、エンジニアは、そのバッチ処理のフェーズ等といった特定の機能を実装するために、一般的（ジェネリック：generic）な制御論理を有するテンプレートソフトウェアオブジェクトを用いて作業を開始し得る。これらのテンプレートソフトウェアオブジェクトの一般的な性質に起因して、ソフトウェアオブジェクトは、特定のプロセス制御環境で用いられる前に、それらが実行する工程の詳細に基づいて修正又はカスタマイズされなければならない。例えば、一般的に混合設備を操作するよう構成される混合フェーズは、混合設備の或る特定の一部を特定の持続時間にわたって特定の速度で動作させるようカスタマイズされなければならない。フェーズをカスタマイズ又は修正するために、通常はレシピ（recipe）が用いられる。その名が暗示するように、レシピは、例えば、クッキーの製造、医薬品の製造、又は他のプロセスの制御等といった具体的なタスクを実行するために、プロセス制御ハードウェアにダウンロードされる命令のセットを含む。レシピは、一般的にフェーズよりも具体的であり、実際に、複数のフェーズの使用法を含む。例えば、クッキー製造レシピは、混合フェーズによって実行され得る混合工程を含み得る。しかし、混合フェーズとは異なり、クッキー製造レシピは、混合が行われるべき持続時間及び速度を特定する。従って、レシピは、混合フェーズの動作を定義するパラメータを特定する。

【０００５】

同様に、別個のプロセス制御システムによって制御されるプロセスプラントでは、安全性若しくは停止手順又は他の安全性関連機能を提供するために用いられる安全計装システムで用いるためのソフトウェアオブジェクトが作成され得る。プロセスプラント内の有害な、危険な、又は望ましくない状態を検出し、そのような状態が検出された際にプロセスの停止、プロセス内の流れの転換、電力の除去等といった何らかのアクションをとるために、安全計装システムは、一般的に、安全性関連ソフトウェアオブジェクトを用いてプログラムされた１つ以上の安全性コントローラを有し得る。

【０００６】

プロセス制御システムによって実行されるレシピの変更、又は、安全システムソフトウェアオブジェクトの変更が、そのプロセス制御システム又は安全システムの動作に大きく影響し得ることは、容易にわかるであろう。例えば、偶発的に変更された、又は別様で未許可で変更されたレシピをダウンロードすると、プロセスプラントの出力に悪影響を及ぼすことがあり、その結果、製品仕様に準拠しない製品が生じ、利益を損なう結果となる。クッキー等といった製品に対する（ソフトウェアオブジェクトとして実装され得る）レシ

ピが変更されると、明らかに欠陥（例えば、調理が不完全、チョコレートチップが不十分等）のあるクッキーを生じ得るが、全てのレシピ変更が、直ちに知覚可能な欠陥を有する製品を生じるわけではない。例えば、塩が多すぎるクッキーは、製造プロセス中には容易に発見されないこともあり得る。しかし、消費者は、クッキーが塩辛いことに気づき、製造者に苦情を訴えるかもしれず、それから、製造者はそのクッキーのレシピが許容できない方法で変えられたと判断して、そのクッキーを回収する事態にもなり得る。クッキー製造のようなケースでは、未許可のレシピ変更は、最悪でも顧客の不満を生じる程度であり得るが、一方、例えば、医薬品で用いられるレシピの未許可の変更は、より深刻な影響を有する場合がある。特に、薬の量や成分を変えるレシピ変更は、得られる薬を、効果が無い又は毒性があるものにし得る。更に、クッキーのチョコレートチップの量とは異なり、薬の成分が変わっても、その薬は、変更されていない即ち適正に製造された薬と同じ色及び粘稠度を有するように見える場合もあるので、薬の成分の変更は容易に検出できない。

【０００７】

同様に、安全システムで用いられるソフトウェアオブジェクトの未許可の又は不正確な変更は、有害な状態が検出されない、又は有害な状態に応答して不適切なアクションがとられるという結果を生じる場合があり、これは、プラント自体に有害であるのは勿論のこと、プラントの労働者やプラントの近所の住人の生命を脅かすことにもなり得る。更に、欠陥のある安全システムソフトウェアオブジェクトは、有害な状態を誤検出して、実際には有害な状態が存在しない場合にプラントを停止させることもあり得る。

【０００８】

プラントの製造ラインは、製造能力、時間、及び／又は材料にかなりの投資を伴うのが一般的であるので、誤ったレシピや安全システムの誤った作動により、進行中の製品を廃棄しなければならないことは、その製造ラン（稼働）を実行しているエンティティ（事業体）、及び、その製造ラン（稼働）によって製造される製品を受け取ることを期待している何らかの他のエンティティに対して、かなりの財務上の悪影響を及ぼし得る。例えば、ワイン、ビール、チーズ等といった発酵を伴う製品を製造するためのレシピは、何週間又は何ヶ月ものプロセス時間と、かなりの材料投資とを要することが多い。

【０００９】

一般的に、プロセス制御システムに対するレシピ、並びに安全システムソフトウェアオブジェクトを含む他のソフトウェアモジュール又はオブジェクトは、エンジニアや科学者によって書かれ、彼らは、例えば、研究グループや製造グループ等といった様々なエンティティに、そのレシピや安全システムソフトウェアがプロセス制御システムや安全システムにダウンロードされる前に、それらを承認するよう要求する。しかし、プロセス制御システムソフトウェアに対する承認プロセスは、承認を求めるメモや依頼書を回覧することによって行われるのが一般的であり、より略式の方法での入力を要求することによって行われる場合もある。更に、プロセスプラント内でオンラインで用いられるプロセス制御システムや安全システムに未承認のソフトウェアをダウンロードすることを防止するには、プロセス制御システム及びレシピ並びにそこで実装される他のソフトウェアオブジェクトの実際上の知識以外の障害が希に存在する。

【００１０】

油やガスの精製等といった、安全計装システムを用いる施設では、製造システムにおいてソフトウェアが実行され得るようになる前に、組織の１人以上の個人が、そのソフトウェアの特定の部分を自分が承認することを手作業で示すことを要求する標準操作手順（ＳＯＰ）を実施する試みが行われている。実際に、ドラフト標準（規格）ＩＥＣ６１５５１１は、複数の異なる安全度水準（ＳＩＬ）に対して複数の異なる承認レベルを定義することに加えて、そのような承認を必要とする。例えば、或る設備に対する検証プランは、ＳＩＬがレベル２以下である場合には同じ部門の同格者によって承認され、ＳＩＬがレベル３以上である場合には異なる部門の同格者によって承認される、安全計装システムによって実行される安全機能を要求するＳＯＰを含み得る。

【００１１】

周知のように、ＳＩＬは、或るシステムが、そのシステムが実行することになっている事柄をそのシステムが実行することになっている時に実行することにおいてどの程度良好な信頼性を有するかに関する、システムの保全性を定義するＩＥＣ６１５０８標準において定義されている測度である。より具体的には、ＳＩＬは、作動要求時の失敗の平均確率（ＰＦＤａｖｇ）によって定義される。ＰＦＤａｖｇの逆数に関係するリスク低減ファクタは、安全計装機能が用いられる前のプロセスリスクと、そのプロセス又は機器の一部に対して達成されなければならないリスクの「許容レベル」との差分を定める。基本的に、リスク低減ファクタは、安全計装機能を有しない「軽減されていないリスク」を、設定された「許容リスク」で除算したものである。リスク低減ファクタ及びＳＩＬは、共に、安全システム内の異なる各安全計装機能に対して定められ、個々の安全計装機能は、各ハザードシナリオについて、必要性を識別し、システムを安全な状態にするように動作するよう設計される。

【００１２】

安全計装機能の承認プロセスが実施されている設備では、承認は、手作業のアプローチ及び電子的アプローチを含む２つの可能なアプローチの一方を用いて処理されるのが一般的である。手作業のアプローチでは、承認されるべきソフトウェアオブジェクトのプリントされたコピー（即ち、安全機能ソフトウェアのプリントされたコピー）が、検査のために各承認者に手作業で渡され、全ての承認の署名が紙のフォーマットで手作業で集められる。電子的アプローチでは、ソフトウェアオブジェクトの電子バージョン（即ち、ソフトウェアを記述する一連の画面キャプチャ又は他のテキスト及びグラフィックに基づくドキュメンテーション）を、検査のために必要とされる各承認者に渡すために、電子文書管理システムが用いられる。このケースでは、全ての承認の署名は電子的フォーマットで集められる。

【００１３】

しかし、これらのアプローチには深刻な短所がある。具体的には、文書承認プロセスとソフトウェアが実行される安全計装システムとの間には直接のリンケージがない。文書承認プロセスは安全計装システムの外部で行われるので、承認者は、承認を求められているソフトウェアオブジェクトをそのネイティブ環境（例えば、安全計装システム内）で検査できず、その結果、検査される文書の正確さに関する問題が生じ得る。更に、文書承認プロセスは、安全計装システムやそれと関連付けられた設計システムの外部で行われるので、安全計装システム内には、全ての承認が整うまで未承認のソフトウェアオブジェクトを実行できないことを保証できる、又は、ソフトウェアオブジェクトに対して変更が行われた場合には、そのソフトウェアオブジェクトに対する全ての以前の承認が取り消される（即ち、そのソフトウェアが未承認の状態になる）ことを保証できる機構がない。

【００１４】

更に、別個の承認及び安全性設計システムでは、システムが、そのソフトウェアオブジェクトに対するソフトウェア開発監査トレイル内に承認の記録を維持することは不可能である。更に、承認システムは安全計装システムにリンクされていないので、両方のシステムがそれぞれ独立して検証されなければならず、これは時間がかかると共に繰返しの多い作業である。

【発明の開示】

【課題を解決するための手段】

【００１５】

プロセス制御システム及び安全システム環境内で作成される新たなソフトウェアオブジェクトの電子承認を実装及び管理するために、ソフトウェアオブジェクト承認システムは、プロセス制御又は安全システム環境、特に、プロセス制御又は安全システム設計環境と統合される。例えば、プロセス制御システム及び／又は安全システムにおいて用いられるシステム及び方法は、プロセス制御又は安全システム内でソフトウェアオブジェクトを実装する前に承認を得ることが必要な１グループのエンティティ（実体）を表す識別情報を電子的に生成する。このシステム及び方法は、識別情報内で表される各エンティティから

、ソフトウェアオブジェクトの承認に関する電子的な標示を受け取ることができ、識別情報内で表される各エンティティがそのソフトウェアオブジェクトを承認するまで、プロセス制御又は安全システムがそのソフトウェアオブジェクトを実装することを防止してもよい。更に、このシステム及び方法は、この電子的な標示に基づき、プロセス制御又は安全システムがソフトウェアオブジェクトを実装することを選択的に可能にする。

【００１６】

それに加えて又はその代わりに、プロセス制御又は安全システムにおいて用いられるソフトウェアオブジェクト承認システム及び方法は、複数のエンティティの少なくともの１つがソフトウェアオブジェクトを承認していないという電子的な標示を受け取ったことに応答して、又は、ソフトウェアオブジェクトが変更又は何らかの方法で改変された場合に、そのソフトウェアオブジェクトが承認されていないと判定することができる。その後、このシステム及び方法は、複数のエンティティの各々がソフトウェアオブジェクトを承認したという別の電子的な標示を受け取ったことに応答して、そのソフトウェアオブジェクトが承認されたと判定してもよく、そのような承認に応答して、このシステム及び方法は、そのソフトウェアオブジェクトをプロセス制御又は安全システムにダウンロードすることを可能にしてもよい。

【発明を実施するための最良の形態】

【００１７】

例えば、プロセス制御システムにおけるレシピや安全計装システムにおけるソフトウェアルーチン等といった、ソフトウェアオブジェクトの承認及びダウンロードを電子的に制御する方法及びシステムを以下に詳細に説明する。これらの方法及びシステムは、ソフトウェアオブジェクトの作者が、プロセス制御システム又は安全計装システム内においてオブジェクトがダウンロード又は実装される前に、そのソフトウェアオブジェクトを許可しなければならない検査者の各個人又はグループからの承認を自動的に且つ／又は電子的に取得するのを可能にするために用いられてよい。これらの方法及びシステムは、検査されるソフトウェアオブジェクトが可能な限り正確であるように、検査者がソフトウェアオブジェクトを、それが用いられる環境で検査することを可能にする。更に、この承認システムはプロセス制御又は安全システム設計環境と統合されているので、承認システムは、全ての承認が整うまで未承認のソフトウェアオブジェクトを実行できないこと、及び、ソフトウェアオブジェクトに対して変更が行われた場合には、そのソフトウェアオブジェクトに対する全ての以前の承認が取り消される（即ち、そのソフトウェアが未承認の状態になる）ことを確実にする機構を提供できる。更に、この統合型ソフトウェアオブジェクト承認システムは、プロセス制御又は安全システムが、承認の正確な記録を、そのソフトウェアオブジェクトに対するソフトウェア開発監査トレイル（記録）内に維持することを可能にする。

【００１８】

所望であれば、多くの異なる技術によって、ソフトウェアオブジェクトの検査者又は署名者に通知が行われてもよく、検査者又は署名者は、通知されたらソフトウェアオブジェクトを検査して、そのソフトウェアオブジェクトを承認又は拒絶してもよい。各検査者がソフトウェアオブジェクトを承認した場合には、そのソフトウェアオブジェクトは、プロセス制御システム又は安全計装システムにダウンロードされるために使用可能にされてもよい。更なる機能性として、様々な人々又はエンティティ（例えば、検査者、作者、ビジネスグループ、又はその他）がソフトウェアオブジェクトの承認ステータスをチェックすることを可能にすること、ソフトウェアオブジェクトが変更された場合には許可を自動的に取り消すこと、変更が行われた場合にはソフトウェアオブジェクトのバージョンを自動的に変更すること等が含まれてもよい。

【００１９】

以下、ソフトウェアオブジェクト承認システム及び方法を、例として、プロセス制御システムにおけるレシピの承認及びダウンロード、又は、安全システム環境におけるソフトウェアオブジェクトの承認及びダウンロードのために用いられるものとして説明するが、

本願明細書に記載されるシステム及び方法は、例えば、他のプロセス制御環境におけるユニット、フェーズ、グラフィック等といった、他のタイプのソフトウェアオブジェクトに対しても有用に用いられ得る。更に、本願明細書に例として記載されるソフトウェアオブジェクト承認システム及び方法は、一度に単一のソフトウェアオブジェクトを承認及びダウンロードするため、並びに／或いは、一群の関連する又は無関連のソフトウェアオブジェクトを同時に又は異なる時に承認及びダウンロードするために用いられてもよい。

【００２０】

更に、本願明細書に記載されるソフトウェアオブジェクト承認システム及び方法は、バージョン制御ソフトウェアとの関連において有益に用いられ得ることが容易にわかるであろう。１つの例示的なタイプのバージョン制御ソフトウェアは、「プロセス制御システムにおけるバージョン制御及び監査トレイル（Version Control and Audit Trail in a Process Control System）」という名称の米国特許第６，４４９，６２４号に開示されており、その全開示を参照することにより本願明細書に明示的に組み込む。

【００２１】

ここで図１を参照すると、例示的なプロセス制御システム１０は、イーサネット（登録商標）接続１５を介して複数のワークステーション１４に接続されたコントローラ１２を含む。コントローラ１２は、１組の通信ライン又はバス１９を介して、（全体を参照番号１６によって示される）プロセスと関連付けられた装置又は機器にも接続される。単なる例として、エマーソン・プロセス・マネジメント（Emerson Process Management）によって販売されているＤｅｌｔａＶ（登録商標）コントローラであってもよいコントローラ１２は、プロセス１６の所望の制御を実装するために、好ましくはオブジェクト指向プログラミング技術を用いて実装される１つ以上のプロセス制御ルーチン又はソフトウェアオブジェクトを実行するために、プロセス１６に隅なく分散されたフィールド装置及びフィールド装置内の機能ブロック等といった制御要素と通信できる。ワークステーション１４（例えば、パーソナルコンピュータであってもよい）は、コントローラ１２によって実行されるプロセス制御ルーチン又はソフトウェアオブジェクトを設計するため、そのようなプロセス制御ルーチン又はソフトウェアオブジェクトをダウンロードするためにコントローラ１２と通信するため、並びに、プロセス１６の動作中にプロセス１６に関する情報を受け取って表示するために、１人以上のエンジニア又は他のユーザによって用いられる設計ソフトウェア１７を含んでもよい。更に、承認ソフトウェア１８が、設計ソフトウェア１７を用いて設計又は修正された任意のレシピ又は他のプロセス制御ルーチン若しくはオブジェクトの承認を提供するために、設計ソフトウェア１７と通信可能に接続され、且つ、設計ソフトウェア１７と統合されてもよい。

【００２２】

各ワークステーション１４は、コンフィグレーション設計アプリケーション等といったアプリケーションを格納すると共にプロセス１６のコンフィグレーションに関するコンフィグレーションデータ等といったデータを格納するための、メモリ２０を含む。各ワークステーション１４はプロセッサ２１も含み、プロセッサ２１は、ユーザがプロセス制御ルーチン又はソフトウェアオブジェクトを設計及び／又は修正すること、並びに、これらのプロセス制御ルーチン又はソフトウェアオブジェクトをコントローラ１２にダウンロードすることを可能にするために、アプリケーション１７及び１８を実行する。同様に、各コントローラ１２は、プロセス１６を制御するために用いられるコンフィグレーションデータ及びプロセス制御ルーチンを格納するメモリ２２を含むと共に、プロセス制御ストラテジー（計画）を実装するためにプロセス制御ルーチンを実行するプロセッサ２４を含む。コントローラ１２がＤｅｌｔａＶ（登録商標）コントローラである場合には、コントローラ１２はユーザに対して、ワークステーション１４の１つを介して、コントローラ１２内のプロセス制御ルーチンのグラフィカルな表示（プロセス制御ルーチン内の制御要素、及びこれらの制御要素がプロセス１６の制御を提供するよう構成された様子を示す）を提供してもよい。

【００２３】

図１のシステム１０は、ワークステーション１４の１つ以上が接続され得るネットワーク３０も含んでよい。ネットワーク３０は、例えば、インターネット、イントラネット、ローカルエリアネットワーク（ＬＡＮ）、ワイドエリアネットワーク（ＷＡＮ）、又はその他の任意の適切なネットワーク等といった、任意の適切なネットワークを用いて実装されてよい。図示されているネットワーク３０は有線接続を有するが、そのようなネットワークは無線ネットワークであってもよく、又は、有線部分及び無線部分の両方を含むネットワークであってもよいことは容易にわかるであろう。

【００２４】

ワークステーション１４には、ネットワーク３０を介して複数の端末３２が接続されてよい。各端末３２はメモリ３４を含んでよく、メモリ３４は、そこに格納された命令を実行するよう構成されたプロセッサ３６に接続される。例示的な一実施形態では、端末３２は、今日知られている従来のパーソナルコンピュータで使用可能であるのと同じ又はそれより高い処理能力及びメモリを含み得るパーソナルコンピュータ又は任意の類似の処理装置であってもよい。

【００２５】

図１のプロセス制御システム１０の説明に戻ると、コントローラ１２は、バス１９を介して、本願明細書ではＲｅａｃｔｏｒ＿０１、Ｒｅａｃｔｏｒ＿０２、及びＲｅａｃｔｏｒ＿０３と呼ぶ３組の類似の構成を有するリアクタと通信可能に接続される。Ｒｅａｃｔｏｒ＿０１は、反応器１００と、反応器１００に流体を提供する制御流体注入ラインに接続された２つの入力弁１０１及び１０２と、反応器１００から流体排出ラインを介して流出する流体を制御するために接続された出力弁１０３とを含む。温度センサ、圧力センサ、流体レベルメーター等といったセンサ、又は、電気ヒーター若しくはスチームヒーター等といった他の何らかの機器であってもよい装置１０５は、反応器１００内又は反応器１００の近傍に配置される。同様に、Ｒｅａｃｔｏｒ＿０２は、反応器２００と、２つの入力弁２０１及び２０２と、出力弁２０３と、装置２０５とを含む。同様に、Ｒｅａｃｔｏｒ＿０３は、反応器３００と、２つの入力弁３０１及び３０２と、出力弁３０３と、装置３０５とを含む。図１に示されるように、コントローラ１２は、バス１９を介して、弁１０１～１０３、２０１～２０３、及び３０１～３０３、並びに、装置１０５、２０５、及び３０５と通信可能に接続され、リアクタユニットに関する１つ以上の動作を行うために、これらの要素の動作を制御する。そのような動作は、例えば、反応器の充填、反応器内の材料の加熱、反応器の内容物の投棄、反応器のクリーニング等を含んでもよい。

【００２６】

図１に示されている弁、センサ、及びその他の機器は、例えば、フィールドバス装置、標準４－２０ｍＡ装置、ＨＡＲＴ（登録商標）装置等を含む任意の所望の種類又はタイプの機器であってよく、フィールドバスプロトコル、ＨＡＲＴ（登録商標）プロトコル、４－２０ｍＡアナログプロトコル等といった任意の公知の又は所望の通信プロトコルを用いてコントローラ１２と通信してもよい。更に、コントローラ１２には他のタイプの装置が接続されてもよく、それらはコントローラ１２によって制御されてもよい。また、プロセス１６と関連付けられた他の装置又はエリアを制御するために、コントローラ１２及びワークステーション１４には、イーサネット通信リンク１５を介して他のコントローラが接続されてもよく、そのような更なるコントローラの動作は、図１に示されるコントローラ１２の動作と任意の所望の方法で調和されてもよい。

【００２７】

一般的に、図１のプロセス制御システム１０は、バッチ処理を実装するために用いられてもよく、バッチ処理では、例えば、ワークステーション１４の１つ又はコントローラ１２がバッチ実行管理ルーチン４０を実行する。バッチ実行管理ルーチン４０とは、食料品、薬、又は他の医薬品等といった製品を製造するために必要な一連の異なる工程（一般的にフェーズと呼ばれる）を実行するために、１つ以上のリアクタユニット（並びに他の機器）の動作を指示する高レベルの制御ルーチンである。工程又はフェーズはソフトウェアオブジェクトを用いて実装されるのが一般的であり、ソフトウェアオブジェクトは、シス

テム１０内のプロセッサ２１及び２４の１つ以上にダウンロード可能であると共に、それらよるインスタンス作成及び実行が可能である。

【００２８】

バッチ実行管理ルーチン４０は、複数の異なるフェーズを実施するために、一般的にレシピと呼ばれるものを用いる。レシピとは、行われる工程、工程と関連付けられた量及び時間、並びに工程のシーケンス（順序）を特定するソフトウェアオブジェクトである。１つのレシピに対する工程は、例えば、反応器に適切な材料又は内容物を充填すること、反応器内の材料を混合すること、反応器内の材料を或る時間量にわたって或る温度まで加熱すること、反応器を空にすること、及び、次のバッチラン（実行）に備えて反応器をクリーニングすることを含み得る。各工程はバッチランの１つのフェーズを定め、バッチ実行管理ルーチン４０は、コントローラ１２に、これらの各フェーズに対する異なる制御アルゴリズムを実行させる。当然ながら、具体的な材料、材料の量、加熱温度及び時間等は、異なるレシピ毎に異なり得るものであり、従って、これらのパラメータは、生産又は製造される製品及び用いられるレシピに応じて、バッチラン毎に変化し得る。本願明細書に記載される制御ルーチン及びコンフィグレーションは、図１に示されるリアクタユニットにおけるバッチランに対するものであるが、制御ルーチンは、所望される場合には、他の任意の所望のバッチ処理ランを実行するため、又は、連続的なプロセスランを実行するために、他の所望の装置を制御するために用いられてもよいことを、当業者は理解するであろう。

【００２９】

高次レベルでは、動作の適切な関連部分において、ワークステーション１４の１つに位置する人又はエンティティは、レシピ又は他のソフトウェアオブジェクトを作成又は修正するために設計ソフトウェア１７を用いてもよく、承認ソフトウェア１８は、例えば、製造、エンジニアリング、品質保証、又は管理等といった様々な許可エンティティからの承認を電子的に要求してもよい。許可エンティティは、ワークステーション１４又は端末３２を用いて、問題のレシピ及び／又は他のソフトウェアオブジェクトを検査し、そのレシピ及び／又は他のソフトウェアオブジェクトを承認又は拒絶してもよい。問題のソフトウェアオブジェクトの承認又は拒絶は、そのオブジェクトの承認を要求した人又はエンティティに通信されてもよく、又は、そのソフトウェアオブジェクトを誰が承認し、誰が承認しなかったかを追跡記録可能な承認ルーチン１８に返されてもよい。ソフトウェアオブジェクトが、承認を得ることが要求された全てのエンティティ（又は、１グループのエンティティを定義する任意の予め設定されたサブセット）によって承認されたら、承認ルーチン１８は、そのソフトウェアオブジェクトが、プロセス制御システム１０内における実装又は実行のために、コントローラ１２の１つ又はバッチ実行管理ルーチン４０にダウンロードされるのを可能にしてもよい。

【００３０】

図２のオブジェクトツリーは、承認ソフトウェア１８と連動して動作する設計ソフトウェア１７を用いて作成及びダウンロードされることができる例示的なソフトウェアオブジェクトを示す。図２のソフトウェアオブジェクトは、（上述のレシピに加えて）本願明細書に記載される統合型電子承認システムを用いて作成及び承認されることができるソフトウェアオブジェクトの単なる例として提供されるものであり、このシステムを用いて他のソフトウェアオブジェクトが作成及び承認されてもよいものと理解される。図２において、ソフトウェアルーチンを用いて実装されるソフトウェアオブジェクトはボックスにより図示されており、オブジェクトの一般的なカテゴリー（又はオブジェクトタイプ）はツリーのオブジェクトの上方にボックスなしで示されている。図２に示されるように、プロセス制御システム１０は、例えば、プロセス制御プラント内における建物や他の地理的なエリア指定であり得る１つ以上のエリアを含む。図２のオブジェクトツリーでは、プロセス１６は、Ｂｕｉｌｄｉｎｇ＿０１、Ｂｕｉｌｄｉｎｇ＿０２、及びＢｕｉｌｄｉｎｇ＿０３と名付けられた３つのエリアオブジェクトを有する。各エリアオブジェクトは複数のプロセスセルに分割されてもよく、各プロセスセルは、そのエリア内で行われるプロセスの

それぞれ異なる局面に対応する。図２のエリアオブジェクトＢｕｉｌｄｉｎｇ＿０１は、Ｃｅｌｌ＿０１及びＣｅｌｌ＿０２と示される２つのプロセスセルオブジェクトを含むものとして図示されている。Ｃｅｌｌ＿０１は、例えば、Ｃｅｌｌ＿０２で用いられる製品コンポーネントを製造することに関連していてもよい。各セルオブジェクトは、そのプロセスセルで用いられるハードウェアの異なるカテゴリー又はグループ分けを識別する０個以上のユニットクラスを含んでもよい。一般的に、ユニットクラスは、１組の関連機器の共通のコンフィグレーションを保持する、名前を付けられたオブジェクトであり、より具体的には、同一ではないとしても非常に類似したプロセス計装を有するユニットの集合であり、各々は、プロセス内の同一ではないとしても非常に類似した機能を実行する。ユニットクラスオブジェクトは、一般的に、それらが属するプロセス制御システム内のユニットのタイプを記述する名前が付けられる。図２は、Ｍｉｘ＿Ｔａｎｋユニットクラス、Ｒｅａｃｔｏｒユニットクラス、及びＦｅｅｄ＿Ｔａｎｋユニットクラスを含む。当然ながら、大半のプロセス制御システム又はネットワークにおいては、例えば、乾燥器ユニット、フィードヘッダーユニット、及び、他の個々の又は論理的なハードウェアのグループ分けを含む、他の多くのタイプのユニットクラスが提供又は定義される。

【００３１】

図２のＲｅａｃｔｏｒユニットクラスに対して図示されているように、各ユニットクラスオブジェクトには、ユニットモジュールオブジェクト及びフェーズクラスオブジェクトが関連付けられてもよい。ユニットモジュールオブジェクトは、一般的に、名前が付けられたユニットクラス内の複製された（レプリカの）ハードウェアの或るインスタンスを特定し、一方、フェーズクラスは、一般的に、そのユニットクラスと関連付けられたユニットモジュールに適用可能なフェーズを特定する。より具体的には、ユニットモジュールオブジェクトは、単一のプロセスユニットに対する全ての変数及びユニットフェーズ（定義は後述する）を保持する、名前が付けられたオブジェクトであり、一般的に、特定のプロセス機器を識別する名前が付けられる。例えば、図２のＲｅａｃｔｏｒユニットクラスは、図１に示されるＲｅａｃｔｏｒ＿０１、Ｒｅａｃｔｏｒ＿０２、及びＲｅａｃｔｏｒ＿０３にそれぞれ対応するＲｅａｃｔｏｒ＿０１、Ｒｅａｃｔｏｒ＿０２、及びＲｅａｃｔｏｒ＿０３ユニットモジュールを有する。同様に、Ｍｉｘ＿Ｔａｎｋユニットクラス及びＦｅｅｄ＿Ｔａｎｋユニットクラスも、プロセス１６内の特定のハードウェア又は機器に対応する特定のユニットモジュールを有する。しかし、簡潔のために、図１にはＭｉｘ＿Ｔａｎｋユニットクラス又はＦｅｅｄ＿Ｔａｎｋユニットクラスと関連付けられた機器は示さない。

【００３２】

フェーズクラスは、同じユニットクラスに属する複数のユニット上及び複数の異なるユニットクラス上で実行可能なフェーズに対する共通のコンフィグレーションを保持する、名前が付けられたオブジェクトである。本質的に、各フェーズクラスは、同じ又は異なるユニットクラス内のユニットモジュールを制御するためにコントローラ１２によって作成されて用いられる異なる制御ルーチン（又はフェーズ）である。一般的に、各フェーズクラスは、ユニットモジュールに対して行われる動作を記述する動詞に従って名前が付けられる。例えば、図２に示されるように、Ｒｅａｃｔｏｒユニットクラスは、図１の反応器１００、２００、又は３００のいずれか１つを充填するために用いられるＦｉｌｌフェーズクラスと、図１の反応器１００、２００、又は３００のいずれか１つを加熱するために用いられるＨｅａｔフェーズクラスと、図１の反応器１００、２００、又は３００のいずれか１つを空にするために用いられるＤｕｍｐフェーズクラスと、図１の反応器１００、２００、又は３００のいずれか１つをクリーニングするために用いられるＣｌｅａｎフェーズクラスとを有する。当然ながら、このユニットクラス又は他の任意のユニットクラスと関連付けられた他の任意のフェーズクラスが存在してもよい。Ｆｉｌｌフェーズクラスは、Ｒｅａｃｔｏｒユニットクラス及びＦｅｅｄ＿Ｔａｎｋユニットクラスの両方と関連付けられており、従って、Ｒｅａｃｔｏｒユニットモジュール及びＦｅｅｄ＿Ｔａｎｋユニットモジュールに対する充填機能を実行するために用いることができる。

【００３３】
フェーズクラスは、一般的に、全体的なバッチ処理において必要な、そのバッチ処理に対するレシピによって定義される何らかの機能を実行するために、バッチ実行管理ルーチンによって呼び出され得るソフトウェアルーチン又はオブジェクトであると考えられてもよい。フェーズクラスは、基本的にバッチ実行管理ルーチン又は別のフェーズクラスからフェーズクラスソフトウェアルーチン又はオブジェクトに提供される入力である０個以上のフェーズ入力パラメータと、基本的にフェーズクラスルーチンからバッチ実行管理ルーチン又は別のフェーズクラスに渡される出力である０個以上のフェーズ出力パラメータと、そのフェーズクラスの動作や、そのフェーズクラスが何らかの方法で関連付けられている他のフェーズクラスに関する情報に関してユーザに対して表示されるメッセージであってもよい０個以上のフェーズメッセージと、このフェーズクラスに基づきフェーズ論理モジュール（ＰＬＭ）又はユニットフェーズ内でパラメータを作成させる０個以上のフェーズアルゴリズムパラメータとを含んでもよい。これらのフェーズアルゴリズムパラメータは、フェーズの実行中の一時的な格納場所又は変数として用いられ、ユーザ又はバッチ実行管理ルーチンに対して必ずしも可視ではない。フェーズクラスは、一般的にフェーズを実装するために用いられる制御ルーチンである１つ以上のフェーズアルゴリズム定義（ＰＡＤ）を含む。フェーズクラスは、０個、１個、２個、又はそれ以上のユニットクラスに対する連関リストも有し、このリストは、このフェーズクラス及びそれに従ってこのフェーズクラスのＰＡＤが適用され得るユニットクラスを定義する。Ｆｉｌｌフェーズクラスの連関リストは、Ｒｅａｃｔｏｒユニットクラス及びＦｅｅｄ＿Ｔａｎｋユニットクラスの両方を含む。

【００３４】
図３は、図２に示されているオブジェクトの幾つかのより詳細なバージョンと、これらのオブジェクト間の相関関係とを示す。図３には、２つのユニットクラス、即ち、Ｒｅａｃｔｏｒユニットクラス５０及びＦｅｅｄ＿Ｔａｎｋユニットクラス５２が示されている。Ｒｅａｃｔｏｒユニットクラス５０は、１つのユニットモジュール５４、即ちＲｅａｃｔｏｒ＿０１を有する。他のユニットモジュールが存在してもよいが、単に図３には図示しない。ユニットモジュール５４は、Ｒｅａｃｔｏｒユニットクラス５０と関連付けられた幾つかのリアクタパラメータを定義する。即ち、Ｒｅａｃｔｏｒ＿０１の容量（Ｃａｐａｃｉｔｙ）は３００であること、及び、Ｒｅａｃｔｏｒ＿０１は撹拌器（Ａｇｉｔａｔｏｒ）を含まないことを定めている。同様に、Ｒｅａｃｔｏｒユニットクラス５０には、Ｆｉｌｌフェーズクラス５６及びＤｕｍｐフェーズクラス５８を含む２つのフェーズクラスが関連付けられている。Ｆｉｌｌフェーズクラス５６は、２つのエイリアス名（Ａｌｉａｓ Ｎａｍｅｓ）、即ち、＃ＩＮＬＥＴ＿ＶＡＬＶＥ＃及び＃ＬＥＶＥＬ＃を用いて設計されたＰＡＤを含む（その右側にグラフィカルな形態のＳＦＣとして図示されている）。これらのエイリアス名は、実際には、Ｆｉｌｌフェーズクラス５６のＰＡＤ内に図示されているボックス内で用いられるが、或いは、ＰＡＤの論理内のどこか別の場所で用いられてもよい。Ｆｉｌｌフェーズクラス５６は、更に、ＴＡＲＧＥＴ＿ＬＥＶＥＬとして定義される入力（Ｉｎｐｕｔｓ）と、ＦＩＮＡＬ＿ＬＥＶＥＬとして定義される出力（Ｏｕｔｐｕｔｓ）とを含む。エイリアス名は、番号記号（＃）によって区切られて即ちマークされて示されているが、他の任意の識別子を用いてフェーズのインスタンス化の際に置換されるべきエイリアス名を定義してもよい。同様に、Ｄｕｍｐフェーズクラス５８は、その右側にグラフィカルな形態で図示されている、＃ＯＵＴＬＥＴ＿ＶＡＬＶＥ＃及び＃ＬＥＶＥＬ＃のエイリアス名を有するＰＡＤと、ＲＡＴＥとして定義される入力（Ｉｎｐｕｔｓ）と、ＦＩＮＡＬ＿ＬＥＶＥＬとして定義される出力（Ｏｕｔｐｕｔｓ）と、ＡＣＴＵＡＬ＿ＲＡＴＥとして定義される（ＰＡＤによって用いられる）アルゴリズムパラメータ（Ａｌｇｏ Ｐａｒａｍｅｔｅｒｓ）（これは、ＰＡＤの実行中に一時的な格納場所として用いられてもよい）とを含む。

【００３５】
図１～図３は、従来のプロセス制御機能を実行するためにソフトウェアオブジェクトが

作成されて用いられ得るプロセス制御システムを示しているが、図４は、プロセス制御システム及び安全計装システムのいずれか又は両方における同じ又は別のソフトウェアオブジェクトを作成、変更、及び承認するために用いられ得る、統合された設計及び承認ソフトウェアを含む統合型のプロセス制御システム及び安全計装システムを示す。具体的には、図４に示されるように、プロセスプラント１１０は、プロセス制御システム１１２によって提供される制御をモニタ及びオーバーライドすることによりプロセスプラント１１０の安全であると思われる動作を最大化する安全計装システム（ＳＩＳ）として一般的に動作する安全システム１１４（点線で示される）が統合された、プロセス制御システム１１２を含む。プロセスプラント１１０は、１つ以上のホストワークステーション、コンピュータ、又はユーザインターフェイス１１６（任意のタイプのパーソナルコンピュータ、ワークステーション、ＰＤＡ等であってよい）も含み、これらは、プロセス制御オペレータ、保守作業者、安全性エンジニア等といったプラント作業者によってアクセス可能である。図４に示される例では、２つのユーザインターフェイス１１６が、共通の通信ライン又はバス１２２を介して、２つの別個のプロセス制御／安全制御ノード１１８及び１２０と、コンフィグレーションデータベース１２１とに接続されている。通信ネットワーク１２２は、任意の所望のバスに基づく又はバスに基づかないハードウェアを用いて、任意の所望の有線又は無線通信構成を用いて、イーサネットプロトコル等といった任意の所望の又は適切な通信プロトコルを用いて、実装されてよい。

【００３６】

一般的に、プロセスプラント１１０の各ノード１１８及び１２０は、複数の異なる装置が取り付けられるバックプレーン上で提供されてもよいバス構成を介して相互に接続された、プロセス制御システム装置及び安全システム装置を含む。図４では、ノード１１８は、プロセスコントローラ１２４（コントローラの冗長ペアであってもよい）と、１つ以上のプロセス制御システム入出力（Ｉ／Ｏ）装置１２８、１３０、及び１３２とを含むものとして図示されており、一方、ノード１２０は、プロセスコントローラ１２６（コントローラの冗長ペアであってもよい）と、１つ以上のプロセス制御システムＩ／Ｏ装置１３４及び１３６とを含むものとして図示されている。各プロセス制御システムＩ／Ｏ装置１２８、１３０、１３２、１３４、及び１３６は、図４ではフィールド装置１４０及び１４２として示されている１組のプロセス制御関連フィールド装置と通信可能に接続される。図４のプロセス制御システム１１２は、プロセスコントローラ１２４及び１２６、Ｉ／Ｏ装置１２８～１３６、並びにコントローラフィールド装置１４０及び１４２で概ね構成される。

【００３７】

同様に、ノード１１８は、１つ以上の安全システムロジックソルバー１５０、１５２を含み、一方、ノード１２０は、安全システムロジックソルバー１５４及び１５６を含む。各ロジックソルバー１５０～１５６は、メモリ１７９に格納された安全論理モジュール１５８を実行するプロセッサ１５７を有するＩ／Ｏ装置であり、安全システムフィールド装置１６０及び１６２に制御信号を提供するため、及び／又は、安全システムフィールド装置１６０及び１６２から信号を受け取るために、通信可能に接続される。更に、各ノード１１８及び１２０は、メッセージ伝搬装置（ＭＰＤ）１７０又は１７２を含み、これらは、リング型バス接続１７４（図４にはその一部のみを示す）を介して相互に通信可能に接続される。図４の安全システム１１４は、安全システムロジックソルバー１５０～１５６、安全システムフィールド装置１６０及び１６２、ＭＰＤ１７０及び１７２、並びにバス１７４で概ね構成される。

【００３８】

単なる例として、エマーソン・プロセス・マネジメント（Emerson Process Management）によって販売されているＤｅｌｔａＶ（登録商標）コントローラ又は他の任意の所望のタイプのプロセスコントローラであってもよいプロセスコントローラ１２４及び１２６は、Ｉ／Ｏ装置１２８、１３０及び１３２（コントローラ１２４に対して）、Ｉ／Ｏ装置１３４及び１３６（コントローラ１２６に対して）、並びにフィールド装置１４０及び１４

２を用いて、（一般的に制御モジュールと呼ばれるものを用いて）プロセス制御機能性を提供するようプログラムされる。具体的には、各コントローラ１２４及び１２６は、そこに格納された又は別様で関連付けられた１つ以上のプロセス制御ルーチン（これらはソフトウェアオブジェクトであり、相互接続されたソフトウェアオブジェクトの集合で構成されてもよい）を実装又は監視すると共に、プロセス１１０又はプロセス１１０の一部を任意の所望の方法で制御するために、フィールド装置１４０及び１４２並びにワークステーション１１４と通信する。フィールド装置１４０及び１４２は、センサ、弁、トランスミッタ、ポジショナ等といった任意の所望のタイプのフィールド装置であってよく、例えば、幾つかを挙げれば、ＨＡＲＴ（登録商標）や４－２０ｍＡプロトコル（フィールド装置１４０に対して図示されている）、ＦＯＵＮＤＡＴＩＯＮ（登録商標）フィールドバスプロトコル（フィールド装置１４２に対して図示されている）等といった任意のフィールドバスプロトコル、又は、ＣＡＮプロトコル、Ｐｒｏｆｉｂｕｓ（登録商標）プロトコル、ＡＳ－Ｉｎｔｅｒｆａｃｅプロトコルを含む任意の所望の公開、非公開、又はその他の通信又はプログラミングプロトコルに準拠してよい。同様に、Ｉ／Ｏ装置１２８～１３６は、任意の適切な通信プロトコルを用いる任意の公知のタイプのプロセス制御Ｉ／Ｏ装置であってよい。

【００３９】

図４の安全ロジックソルバー１５０～１５６は任意の所望のタイプの安全システム制御装置であってよく、これらはプロセッサ１５７とメモリとを含み、メモリは、フィールド装置１６０及び１６２を用いて安全システム１１４と関連付けられた制御機能性を提供するためにプロセッサ１５７上で実行されるよう構成された安全論理モジュール１５８を格納する。当然ながら、安全性フィールド装置１６０及び１６２は、上述したような任意の公知の又は所望の通信プロトコルに準拠した又はそれを用いる任意の所望のタイプのフィールド装置であってよい。具体的には、フィールド装置１６０及び１６２は、別個の専用の安全性関連の制御システムによって従来の方法で制御されるタイプの、安全性関連のフィールド装置であってもよい。図４に示されるプロセスプラント１１０では、安全性フィールド装置１６０は、ＨＡＲＴ（登録商標）や４－２０ｍＡプロトコル等といった専用又はポイント・ツー・ポイント通信プロトコルを用いるものとして図示されており、一方、安全性フィールド装置１６２は、フィールドバスプロトコル等といったバス通信プロトコルを用いるものとして図示されている。安全性フィールド装置１６０は、閉止弁、遮断スイッチ等といった任意の所望の機能を実行してよい。

【００４０】

各ノード１１８及び１２０では、コントローラ１２４及び１２６をプロセス制御Ｉ／Ｏカード１２８、１３０及び１３２、又は１３４及び１３６、安全ロジックソルバー１５０、１５２、１５４、又は１５６、並びにＭＰＤ１７０又は１７２に接続するために、共通バックプレーン１７６（コントローラ１２４、１２６、Ｉ／Ｏ装置１２８～１３６、安全ロジックソルバー１５０～１５６、並びにＭＰＤ１７０及び１７２を通る破線で示される）が用いられる。コントローラ１２４及び１２６はバス１２２にも通信可能に接続されると共に、各Ｉ／Ｏ装置１２８～１３６、ロジックソルバー１５０～１５６、並びにＭＰＤ１７０及び１７２がバス１２２を介していずれかのワークステーション１１６と通信するのを可能にするために、バス１２２に対するバス制御管理システム（bus arbitrator）として動作する。

【００４１】

各ワークステーション１１６は、プロセッサ１７７と、メモリ１７８とを含み、少なくともの１つワークステーションは、プロセッサ１７８上で実行されるよう構成された１つ以上のコンフィグレーション、承認、診断及び／又は表示アプリケーションを格納することが理解されよう。図４の分解図では、ワークステーション１１６の１つには、コンフィグレーションアプリケーション１８０、承認アプリケーション（又はルーチン）１８１、及び表示アプリケーション１８２が格納されて図示されており、ワークステーション１１６の別の１つには、診断アプリケーション１８４が格納されて図示されている。しかし、

所望であれば、これらの及びその他のアプリケーションは、それぞれ異なるワークステーション１１６で又はプロセスプラント１１０と関連付けられた他のコンピュータで格納及び実行されてもよい。一般的に、コンフィグレーションアプリケーション１８０は、安全性エンジニアにコンフィグレーション情報を提供し、安全性エンジニアがプロセスプラント１１０の一部又は全ての要素を構成（設計）して、そのコンフィグレーションをコンフィグレーションデータベース１２１に格納するのを可能にする。コンフィグレーションアプリケーション１８０によって行われるコンフィグレーション動作の一部として、安全性エンジニアは、プロセスコントローラ１２４及び１２６に対する制御ルーチン又は制御モジュール（即ち、ソフトウェアオブジェクト）を作成又は変更してもよく、いずれか又は全ての安全ロジックソルバー１５０～１５６に対する安全論理ソフトウェアモジュール１５８を作成してもよく（安全ロジックソルバー１５０～１５６において又はコントローラ１２４及び１２６においても用いられる、入力、ＶＯＴＥＲ（多数決回路）、及び他の機能ブロックの作成及びプログラミングを含む）、承認ルーチン１８１を介して適切な許可を受け取った後で、これらの異なる制御及び安全性モジュールを、プロセスコントローラ１２４及び１２６並びに安全ロジックソルバー１５０～１５６の適切なものに、バス１２２及びコントローラ１２４及び１２６を介してダウンロードしてもよい。同様に、他のプログラム及び論理を作成し、承認ルーチン１８１を介して適切な許可を受け取った後で、Ｉ／Ｏ装置１２８～１３６、任意のフィールド装置１４０、１４２、１６０及び１６２等にダウンロードするために、コンフィグレーションアプリケーション１８０を用いてもよい。所望であれば、プロセス制御システム１１２及び安全システム１１４の設計活動を互いから分離するために、プロセス制御システム１１２及び安全システム１１４の各々に対して、別個のセットのコンフィグレーション及び承認ルーチンが存在してもよい。

【００４２】

表示アプリケーション１８２は、プロセス制御オペレータ、安全性オペレータ等といったユーザに対して、プロセス制御システム１１２及び安全システム１１４の状態に関する情報を含む１つ以上の表示を所望であれば別個のビュー又は同じビューで提供するために用いられてもよい。例えば、表示アプリケーション１８２は、アラームの標示を受け取ってオペレータに対して表示するアラーム表示アプリケーションであってもよい。所望であれば、そのようなアラーム表示アプリケーションは、共に本特許の譲受人に譲渡されると共にここに参照することにより本願明細書に明示的に組み込まれる「アラーム優先度調整を含むプロセス制御システム（Process Control System Including Alarm Priority Adjustment）」という名称の米国特許第５，７６８，１１９号、及び、「プロセス制御ネットワークにおける統合型アラーム表示（Integrated Alarm Display in a Process Control Network）」という名称の米国特許出願第０９／７０７，５８０号に開示されている形態をとってもよい。しかし、これらの特許のアラーム表示又はアラームバナーは、プロセス制御システム１１２及び安全システム１１４の両方からアラームを受け取って、統合型アラーム表示として表示し得るものであり、システム１１２及び１１４の両方からのアラームは、アラーム表示アプリケーションを実行するオペレータワークステーション１１４に送られ、複数の異なる装置からのアラームとして認識可能であることが理解されよう。同様に、オペレータは、アラームバナーに表示される安全性アラームを、プロセス制御アラームと同様に扱ってもよい。例えば、オペレータ又はユーザは、アラーム表示を用いて、安全性アラームを認識し、安全性アラームを止める等してもよく、それによりオペレータ又はユーザは、その安全性アラームに関する対応するアクションをとるために、バス１２２及びバックプレーン１７６を介した通信を用いて、安全システム１１４内の適切なプロセスコントローラ１２４、１２６にメッセージを送るだろう。同様に、他の表示アプリケーションは、プロセス制御システム１１２及び安全システム１１４の両方からの情報又はデータを表示してもよく、システム１１２及び１１４の１つからの任意のデータをプロセス制御システムに対して従来の方法で提供される１つのディスプレイ又はビューに統合できるように、これらのシステムは同じタイプ及び種類のパラメータ、セキュリティ、及びリファレンシング（参照関係）を用いてもよい。同様に、これらの表示アプリケーション

（これらはソフトウェアモジュールである）に対する変更又は新たな表示アプリケーションが作成されて、本願明細書により詳細に記載される電子署名プロセスを介して適切な許可を受け取った後に実装されてもよい。

【００４３】

いずれにしても、アプリケーション１８０、１８１、１８２及び１８４、並びに他の任意のアプリケーションは、各プロセスコントローラ１２４及び１２６並びに各安全システムロジックソルバー１５０～１５６に別個のコンフィグレーション及び他の信号を送ってもよく、各プロセスコントローラ１２４及び１２６並びに各安全システムロジックソルバー１５０～１５６からのデータを受け取ってもよい。これらの信号は、プロセスフィールド装置１４０及び１４２の動作パラメータの制御に関するプロセス制御関連のメッセージを含んでもよく、安全性関連のフィールド装置１６０及び１６２の動作パラメータの制御に関する安全性レベルメッセージを含んでもよい。安全ロジックソルバー１５０～１５６は、プロセス制御メッセージ及び安全性メッセージの両方を認識するようプログラムされてもよいが、安全ロジックソルバー１５０～１５６は、２つのタイプのメッセージを区別でき、プロセス制御関連のコンフィグレーション信号によってプログラムされ得る又は影響され得ることはない。一例では、プロセス制御システムの装置に送られるプログラミングメッセージは、安全システムの装置によって認識される特定のフィールド又はアドレスを含んでもよく、これらの信号が安全システム装置をプログラムするために用いられるのを防止する。

【００４４】

所望であれば、安全ロジックソルバー１５０～１５６は、プロセス制御Ｉ／Ｏカード１２８～１３６に対して用いられるハードウェア及びソフトウェア設計と同じ又は異なるハードウェア又はソフトウェア設計を用いてもよい。プロセス制御システム１１２内の装置及び安全システム１１４内の装置に対して複数の代替技術を用いることで、共通の原因によるハードウェア又はソフトウェアの不具合が最小限になるか又は解消され得る。更に、ロジックソルバー１５０～１５６を含む安全システム装置は、それによって実装される安全性関連の機能に対して未許可の変更が行われる機会を低減又は解消するために、任意の所望の分離及びセキュリティ技術を用いてもよい。例えば、安全ロジックソルバー１５０～１５６及びコンフィグレーションアプリケーション１８０は、特定の権限レベルを有する人又は特定のワークステーションに位置する人がロジックソルバー１５０～１５６内の安全性モジュールに対する変更を行うことを必要としてもよく、この権限レベル又はワークステーションの所在は、コントローラ１２４及び１２６並びにＩ／Ｏ装置１２８～１３６によって実行されるプロセス制御機能に対する変更を行うために必要な権限又はアクセスレベル又は所在とは異なる。このケースでは、安全性ソフトウェア内で示される人々又は安全システム１１４に対する変更を行うことを許可されたワークステーションに位置する人々のみが、安全性関連の機能を変更する許可を有し、このことにより、安全システム１１４の動作が変造される機会が最小限になる。このようなセキュリティを実装するために、安全ロジックソルバー１５０～１５６内のプロセッサは、入力されるメッセージが適正な形態及びセキュリティを有するかを評価して、安全ロジックソルバー１５０～１５６内で実行される安全制御モジュール１５８に対して行われる変更に対するゲートキーパーとして動作することが理解されよう。

【００４５】

同様に、図４に示される承認ルーチン１８１は、プロセス制御システム１１２及び安全システム１１４のいずれか又は両方におけるソフトウェアオブジェクトに対して、許可を得ることが必要とされる他の個人（名前又は職位による）からの適正な許可無しに変更が行われることを防止するよう実装されてもよい。具体的には、承認ルーチン１８１は、コンフィグレーションアプリケーション又は表示アプリケーション内のソフトウェアモジュールに対して行われる変更を検出して、必要な人々からの適切な許可又は承認無しにこれらのモジュールをプロセス制御システム１１２又は安全システム１１４にダウンロードすることを防止してもよい。

【００４６】
各ノード１１８及び１２０においてバックプレーン１７６を用いることにより、安全ロジックソルバー１５０及び１５２並びに安全ロジックソルバー１５４及び１５６が、これらの各装置によって実装される安全機能を調和させるために、相互にデータを通信するために、又は、他の統合された機能を実行するために、ローカルに相互に通信することが可能になる。一方、ＭＰＤ１７０及び１７２は、プラント１１０の広く異なる場所に配置された安全システム１１４の各部分が、プロセスプラント１１０の複数の異なるノードにおいて調和した安全動作を提供するために、相互に通信することを可能にするよう動作する。具体的には、プロセスプラント１１０内の安全性関連機能の、割り当てられた優先度に従ったカスケード接続を可能にするために、ＭＰＤ１７０及び１７２は、バス１７４と連動して、プロセスプラント１１０の異なるノード１１８及び１２０と関連付けられた安全ロジックソルバーが相互に通信可能にカスケード接続されるのを可能にする。或いは、プラント１１０の別個のエリア又はノード内の個々の安全性フィールド装置への専用ラインを引かずに、プロセスプラント１１０内の異なる場所にある２つ以上の安全性関連機能がインターロック（連動）又は相互接続されてもよい。換言すれば、ＭＰＤ１７０及び１７２並びにバス１７４の使用により、安全性エンジニアは、プロセスプラント１１０にわたって分散された性質を有する安全システム１１４を、その複数の異なるコンポーネントが通信可能に相互接続されるように設計及び構成可能になり、異種の安全性関連ハードウェアが必要に応じて相互通信することが可能になる。この特徴は、必要に応じて又はプロセスプラント１１０に新たなプロセス制御ノードが追加される場合に、安全システム１１４に更なる安全ロジックソルバーを追加できるようにすることにより、安全システム１１４の拡張性も提供する。

【００４７】
一実施形態では、ロジックソルバー１５０～１５６は、機能ブロックプログラミングパラダイムを用いて、安全装置１６０及び１６２に関する制御動作を行うようプログラムされてもよい。具体的には、ロジックソルバー１５４の（メモリ１７９に格納されている）安全制御モジュール１５８ａの１つの拡大図に示されるように、安全制御モジュールは、プロセス１１０の動作中に実行されるよう作成されてロジックソルバー１５４にダウンロード可能な、１組の通信可能に相互接続された機能ブロック（各々がソフトウェアオブジェクトである）を含んでもよい。図４に示されるように、制御モジュール１５８ａは２つのＶＯＴＥＲ機能ブロック１９２及び１９４を含み、ＶＯＴＥＲ機能ブロック１９２及び１９４は、別の機能ブロック１９０と通信可能に相互接続された入力を有する。別の機能ブロック１９０は、例えば、アナログ入力（ＡＩ）機能ブロック、デジタル入力（ＤＩ）機能ブロック、又はＶＯＴＥＲ機能ブロック１９２に信号を提供するよう設計されたその他の機能ブロックであってもよい。ＶＯＴＥＲ機能ブロック１９２及び１９４は、１つ以上の別の機能ブロック１９１と接続された少なくとも１つの出力を有する。別の機能ブロック１９１は、アナログ出力（ＡＯ）機能ブロック、デジタル出力（ＤＯ）機能ブロック、原因及び結果論理（cause and effect logic）を実装する原因及び結果機能ブロック、安全装置１６０及び１６２の動作を制御するためにＶＯＴＥＲ機能ブロック１９２及び１９４から出力信号を受け取ってもよい制御及び診断機能ブロック等であってもよい。当然ながら、安全制御モジュール１５８ａは、任意の所望の機能性を実行するために任意の所望の又は有用な方法で構成された任意のタイプの機能ブロックを含むよう、任意の所望の方法でプログラムされてよい。これに加えて又はこれとは別に、ＡＩブロック又はＤＩブロックによって検出される１つ以上の事象の発生時に１つ以上の停止装置を起動することにより、そのような事象に応答する安全論理制御モジュールを提供するために、ＡＩ機能ブロック及びＤＩ機能ブロック等といった他の入力機能ブロックが安全システムの論理に直接接続されてもよい。制御モジュール１５８ａ及びその中の各機能ブロックは、安全システム１１２にダウンロードされ得ると共に必要に応じて動作中に変更され得る別個のソフトウェアオブジェクトであることが理解されよう。しかし、これらの作成又は修正動作が実行され得る前に、これらのソフトウェアオブジェクトは、承認ルーチン１８１を介し

て許可を受け取ることを必要としてもよく、このことについて、以下に詳細に説明する。
【００４８】
具体的には、作成又は変更された新たな又は修正された各ソフトウェアオブジェクトが、プロセス制御システム１１２又は安全システム１１４内にオンラインで実際にダウンロードされるか又は別様で実装される前に、そのソフトウェアオブジェクトに対して適切な権限又は検査が行われることを確実にするために、承認ルーチン１８１がコンフィグレーションアプリケーション１８０（又はプロセスプラント内で用いられる他の設計アプリケーション）と統合されるか又は通信可能に結合される。
【００４９】
ソフトウェアオブジェクト承認ルーチン１８１は安全計装システム（又はプロセス制御システム）と緊密に統合され、その結果、承認ルーチン１８１は安全計装システムのセキュリティを利用し得ることが理解されよう。更に、承認ルーチン１８１はプロセス制御又は安全システム設計又はコンフィグレーションソフトウェア１８０と同じ環境内にあるので、検査者は、承認対象のソフトウェアオブジェクトを検査するために、そのソフトウェアオブジェクトの開発に用いられたツールと同じツールを用いることができる。従って、検査されるソフトウェアオブジェクトの正確さに関する問題は生じない。更に、承認ルーチン１８１は、必要な承認が全て整うまで、承認を必要とするソフトウェアオブジェクトが制御又は安全環境にダウンロードされることを防止する。従って、プロセス制御又は安全計装システムは、承認されたソフトウェアオブジェクトのみが実行されることを保証することができる。しかし、所望であれば、承認ルーチン１８１は、この要求の実施をオーバーライドして未承認のソフトウェアオブジェクトをダウンロードする能力をユーザに提供するセキュリティキーを含んで（例えば、格納して）もよい。
【００５０】
動作中、承認ルーチン１８１は、ソフトウェアオブジェクトに対して行われている変更を検出してもよく、承認されたソフトウェアオブジェクトに対して何らかの修正が行われたら、そのソフトウェアオブジェクトのバージョン番号が増分されるようにしてもよい。更に、ソフトウェアオブジェクトのバージョン番号が増分された場合には、承認ルーチン１８１は、そのソフトウェアオブジェクトの状態を自動的に未承認の状態に変更する。その結果、ソフトウェアオブジェクトのこの新たなバージョンに対する全ての承認が再実行されるまで、ソフトウェアオブジェクトの新たなバージョンをプロセス制御システムにダウンロード又は別様で実装することができない。
【００５１】
更に、承認ルーチン１８１は、特定のソフトウェアオブジェクトの全ての承認及び／又は拒絶を、そのソフトウェアオブジェクトと関連付けられたコンフィグレーション監査トレイルにログとして記録してもよい。これらの特徴により、ソフトウェアオブジェクトの検証が、プロセス制御又は安全計装システムの検証の統合された一部として行われる。
【００５２】
図５Ａは、プロセス制御及び／又は安全システム内の承認機能を提供するために、承認ルーチン１８１によって、同じ又は異なるソフトウェアルーチンを用いて実行され得るステップを示すフローチャート１８５を示す。ブロック１８６では、承認ルーチン１８１は、ソフトウェア設計アプリケーション（例えば、コンフィグレーションルーチン１８０）又はコンフィグレーションデータベースをモニタし、いずれかのソフトウェアオブジェクトに対して変更が行われたか否か及び変更が行われた時を判定する。ブロック（又はルーチン）１８６は、既存のソフトウェアオブジェクトの変更、新たなソフトウェアオブジェクトの作成を検出するによって、又は、例えばソフトウェアオブジェクト設計者による承認プロセスの実施要求を受け取ることによって、ソフトウェアオブジェクトに対して変更が行われていることを判定してもよい。当然ながら、ブロック１８６は、ソフトウェアオブジェクトに対して全ての変更が行われる（例えば、ソフトウェア設計者が設計アプリケーション内の変更完了を示すボタンを選択することによって示される）まで、承認プロセスの実施を待ってもよい。いずれにしても、ブロック１８６が、ソフトウェアオブジェク

トに対する変更が行われたことを検出した時又はそれからしばらくした後に、ブロック１８７は、設計者に、承認手順を開始するよう自動的に促してもよい。

【００５３】

承認手順の間、（例えば、承認ルーチン１８１内の）ブロック１８８は、１グループのエンティティ（１人以上の人々を含んでよい）を判定してもよく、このプロセスの一部として、プロセス制御又は安全システムのコンフィグレーション中にユーザによって提供される又は別様で提供される情報に基づき、承認に必要な署名の数及びタイプ（職位等）を判定してもよい。一実施形態では、ブロック１８８は、必要な承認のタイプ及び数を判定するために、ユーザ又はコンフィグレーションエンジニア（例えば安全システムコンフィグレーションエンジニア等）によって入力されるリスク低減ファクタ（ＲＲＦ）を用いてもよい。この場合、ユーザは、特定の機能又はソフトウェアオブジェクトに対して達成されるべき所望のＲＲＦを入力してもよく、ブロック１８８は、このＲＲＦを用いて、要求されるＳＩＬレベルを判定する。次にブロック１８８は、（例えばルックアップテーブルを用いることにより）ＳＩＬレベルを用いて、必要な署名の性質（例えば、署名の数、同じ部門又は異なる部門の署名でなければならないか、検査者の業務又は職位のレベル又はタイプ等）を判定してもよい。なお、特定の各安全計装機能にはＳＩＬが関連付けられているので、特定のＳＩＬは或る安全計装機能に特有のものであり、安全機能毎に変化し得る。

【００５４】

いずれにしても、次のブロック１８９は、承認のためにソフトウェアオブジェクトを実際に送るべき実際の人々を判定する。所望であれば、対応するルックアップテーブルから、選択されたグループ内の検査者の名前、電子的アドレス、又は他の電子的連絡情報を取得してもよい。これに加えて又はこれとは別に、１グループの検査者及び関連付けられた電子的連絡情報は、コンフィグレーションエンジニアから直接、ソフトウェアオブジェクトに対する変更を行った人から、又はその他の任意の適切な人から、表示端末を介して取得されてもよい。このプロセスの一部として、ブロック１８９は、ソフトウェアオブジェクト設計者からの補助を用いて又は用いずに、任意の所望の又は所定の方法で、格納されている候補の名前又はオフィス（及びそれと関連付けられた電子的アドレス等）のリストを評価して、そのリストから選択してもよい。或いは、ブロック１８９は、何らかの監督者がユーザインターフェイスを介して、承認プロセスに必要なレベル及び部門に一致する特定の人を選択することを促してもよい。

【００５５】

実際の検査者を選択した後、ブロック１９０は、作成又は修正されたソフトウェアオブジェクトを選択された検査者に電子的に送り、検査者から、作成又は修正されたソフトウェアオブジェクトを承認するか否かに関するメッセージが返されるのを待つ。ブロック１９１は、全ての承認を受け取った時（及び受け取ったか否か）を判定するために、検査者からの応答をモニタしてもよい。修正されたソフトウェアオブジェクトを全ての検査者（又は所定数の検査者）が承認した場合には、ブロック１９２は、そのソフトウェアオブジェクトをダウンロード可能なものとしてマークしてもよい。しかし、全ての検査者からの応答を受け取っていない場合、又は、検査者の１人がそのソフトウェアオブジェクトを拒絶した場合には、承認ルーチン１８１は、そのソフトウェアオブジェクトにダウンロードのためのマーキングを行わず、これにより、そのソフトウェアオブジェクトがプロセス制御又は安全システムで用いられることを防止する。

【００５６】

当然ながら、承認ルーチン１８１は、全ての応答を受け取るまでループしてもよく、設計者又はユーザに任意の所望の方法で応答を提供してもよい。ブロック１９１が、特定の数の拒絶を受け取ったと判定した場合には、ブロック１９３は、そのソフトウェアオブジェクトを、そのソフトウェアオブジェクトのダウンロードを可能にするには修正又は承認者への再提出を要するものとしてマークしてもよい。

【００５７】

当然ながら、承認ルーチン１８１の任意の又は各ブロックは、そのソフトウェアオブジェクトに対する変更、そのソフトウェアオブジェクトに関連する収集された承認及び拒絶、及び承認プロセス関する他の任意の情報を、そのソフトウェアオブジェクトと関連付けられたコンフィグレーションデータベース又は監査トレイルにログとして記録してもよい。更に、承認ルーチン１８５は、検査者の識別情報、応答した検査者、新たな又は修正されたソフトウェアモジュールを承認した検査者及び／又は拒絶した検査者等といった、検査プロセスの状態に関する情報を追跡し、所望であれば設計者（又は他の検査者）に対して表示してもよい。所望であれば、ブロック１９１は、応答していない検査者にリマインダーを送ってもよい。

【００５８】

更に、ソフトウェアモジュールが承認されたら、そのソフトウェアオブジェクトと関連付けられた安全機能又は動作が依然としてその設計基準を満たしているか否かを判定するために、承認ルーチン１８５は、ＲＲＦをオンラインで、即ち、そのソフトウェアオブジェクトが属するプロセス制御又は安全システムの動作中に用いてもよい。具体的には、コンフィグレーション（構成管理）担当者（configurer）は、安全機能のどの要素が周期的テストを要するか、テスト期間はどのようであるべきか、及び、このテストが安全機能の全体的なＲＲＦにどのように影響するかを判定してもよい。これらの詳細は一般的に設計時に決定されるので、この機能を実施する際にはコンフィグレーション担当者にとって既知である。

【００５９】

図５Ｂに示されるように、承認ルーチンのオンラインモニタ部１９４は、要素のテストをモニタするブロック１９５を含む。ブロック１９６によって判定される、特定のソフトウェアオブジェクトに対する周期的テストインターバルを過ぎた場合には、ブロック１９７は、（安全機能テストの実行の失敗に基づき）実際に達成されるＲＲＦを判定し、この修正されたＲＲＦを安全システムの監視要員やオペレータ等といったユーザに対して表示する。ブロック１９８は、この修正されたＲＲＦをターゲット（目標）ＲＲＦと比較して、ターゲットＲＲＦがもはや満たされない場合には、アラームを生成してもよい。所望であれば、ブロック１９９は、ＲＲＦをその設計ターゲット値より高く維持するために、又はＲＲＦをその設計レベルにできるだけ迅速に戻すために、テストインターバルの前又はテストインターバルを過ぎた後に、ＳＩＬ内の情報を用いて、（例えば、実行すべきテストを特定する）作業命令を自動的に生成してもよい。

【００６０】

図１には単一の承認ルーチン１８が示されており、図４には単一の承認ルーチン１８１が示されているが、検査者が設計者とは異なるワークステーションに位置していてもよいように、これらのルーチンのコンポーネントは他のコンピュータ又はワークステーションに格納されてもよいことが理解されよう。この場合には、検査者が設計端末においてルーチン１８又は１８１から承認メッセージを受け取るのを可能にするため、検査対象のソフトウェアオブジェクトを解析するためにプロセス制御又は安全システム内のソフトウェアを呼び出すため、並びに、承認又は拒絶と検査者が作成を望み得る任意のコメントとで応答するために、設計端末と通信可能に接続された（図１に示される検査者端末３２等といった）１組の検査者端末の各々に、類似の又は相補的なソフトウェアがインストールされる。

【００６１】

更に、以下に詳細に論じるように、承認ルーチン１８又は１８１は、検査者の選択、ソフトウェアオブジェクトの修正等といった複数の異なる機能を実行する任意の数のコンポーネントを有してよい。これらのコンポーネントの一部を図６～図２０との関連においてより詳細に論じるが、これらの代わりに又はこれらに加えて、他のルーチンを承認又は設計ルーチンと関連付けてもよいことが理解されよう。更に、図６～図２０では、プロセス制御システムで用いられるレシピの承認に関する複数のサブルーチンを論じるが、これらのルーチンは、安全システムのソフトウェアオブジェクト等といった他のソフトウェアオ

ブジェクトを承認するためにも用いられてよいことが理解されよう。

【００６２】

次に図６に移ると、ワークステーション１４のプロセッサ２１の１つ以上によって実行されてもよいレシピ編集ルーチン４００は、ブロック４０２で実行を開始し、ここで、ユーザ又はオペレータはレシピを作成又は修正する。これには、プロセス制御システム１０において用いられるレシピと関連付けられたソフトウェアオブジェクトの修正が含まれてよい。ユーザは、図１～図５との関連において記載された技術を用いて又はその他の任意の適切な技術を用いて、レシピ又は他のソフトウェアオブジェクトを作成又は修正してもよいことは、容易に認識されよう。レシピが作成又は適切に修正された後、制御はブロック４０２からブロック４０６に渡る。図７～図１３との関連において以下により詳細に論じるように、許可設定ルーチン４０４（図７）は、レシピ編集ルーチン４００の実行前に、又は、レシピ又は他のソフトウェアオブジェクトを修正及び／又はシステム１０にダウンロードする前の他の任意の時に、少なくとも１回実行されてよい。一般的に、許可設定は、レシピ又は他のソフトウェアオブジェクトを実施するために承認を得ることが必要な人々又はエンティティ（例えば、署名者）の特定や、署名者の削除又は修正を含み得るが、これらに限定されない。

【００６３】

ブロック４０６では、ブロック４０２で作成又は修正されたレシピを承認するために、許可設定の際に特定された各署名者からの承認が請求される。承認請求は、許可設定ルーチン４０４との関連においてレシピを検査するよう特定された署名者への電子メールの送信、特定された各署名者の承認ステータスを示す報告の実行、レシピを検査することになっている署名者へのインスタントメッセージの送信、又は、署名者への他の任意の適切な通信方法を介した通知の送信を含み得るが、これらに限定されない。ブロック４０６で各署名者からの承認を請求した後、レシピ編集ルーチン４００は実行を終了するか、又はレシピ編集ルーチン４００を呼び出した別のルーチンに制御を戻す。

【００６４】

許可設定ルーチン４０４のフロー図及びユーザインターフェイス画面をそれぞれ開示する図７及び図８との関連において、許可設定ルーチン４０４の更なる詳細を提供する。許可設定ルーチン４０４は、システム起動時に１回実行されるのが一般的であるが、その代わりに、許可設定ルーチン４０４は所望であれば１回以上実行されてもよい。一般的に、図７に示されるように、許可設定ルーチン４０４が実行されたら、ユーザは、キャンセル／ＯＫブロック４１０で、ルーチンの実行をキャンセルすることを選んでもよい。或いは、ユーザは、ブロック４１２、４１４、又は４１６のそれぞれにおいて、レシピ署名者を追加、削除又は修正することを選んでもよい。署名者の追加、削除又は修正後、制御はブロック４１２、４１４又は４１６から渡り、ユーザがブロック４１０で許可設定ルーチン４０４の動作をキャンセル若しくは終了させること、又は、再びブロック４１２～４１６を用いて署名者を追加、削除又は修正することを選択可能にする。ユーザがブロック４１０で許可設定ルーチン４０４の動作をキャンセル又は終了することを選んだ場合には、許可設定ルーチン４０４は終了する。

【００６５】

図８に示されるように、ユーザインターフェイス４２０はレシピ許可設定タブ４２２を含む。レシピ許可設定タブ４２２は、ユーザが、署名者を追加、修正、又は削除するためにインターフェイスボタン４２４、４２６又は４２８を選択することを可能にする。追加、修正、及び削除インターフェイスボタン４２４～４２８は、図７に示される追加、削除、及び修正ブロック４１２～４１６に対応する（これらのブロックによって実行される機能を呼び出すために選択され得る）。各ブロック４１２～４１６に関する更なる詳細を、インターフェイスボタン４２４～４２８に関する更なる詳細も含め、図９～図１３との関連において提供する。署名者が追加、修正又は削除されると、テキストボックス４３０に署名者のステータスが示される。図８に示されるように、テキストボックス４３０は、人又はエンティティの名前であってもよい署名者の名前を列挙する署名者記述列４３２を含

むと共に、承認へのアクセスを制御するために対応する署名者が有する必要がある機能ロックを列挙する機能ロック列４３４も含む。例えば、図８に示されるように、このレシピは、ＲＥＣＩＰＥ＿ＡＰＰＲＯＶＡＬ＿０１の機能ロックに対応するエンジニアリング（Ｅｎｇｉｎｅｅｒｉｎｇ）、製造（Ｐｒｏｄｕｃｔｉｏｎ）、及び品質保証（Ｑｕａｌｉｔｙ　Ａｓｓｕｒａｎｃｅ）によって検査及び承認される。

【００６６】

図８には２つのチェックボックス４３６及び４３８も示されており、これらは、レシピの許可を可能にすること、及び、含まれるレシピ（即ち、サブレシピ）への承認の波及を可能にすることに対応する。動作においては、チェックボックス４３６がチェックされた場合には、システムの「レシピ許可を可能にする」フィーチャ（機能）が使用可能にされると共に、許可設定プロセスが使用可能にされる。チェックボックス４３８がチェックされていない場合には、ユーザが承認を波及させる選択肢を有しないことを示す。逆に、チェックボックス４３８がチェックされている場合には、ユーザがサブレシピに承認を波及させる選択肢を有する。例えば、メインレシピは２つ以上のサブレシピで構成されても又は含んでもよく、サブレシピには、メインレシピと関連付けられた承認が自動的に波及されてもよい。当然ながら、このようにレシピに対する承認を自動的に波及させることで、特に多数のサブレシピを含むレシピに関しては、かなりの時間の節約になり得る。

【００６７】

ユーザインターフェイス４２０は、参照番号４４０及び４４２でそれぞれ示されるキャンセルインターフェイスボタン及びＯＫインターフェイスボタンも含む。インターフェイスボタン４４０及び４４２は、図７のキャンセル／ＯＫブロック４１０に対応し、ユーザが許可設定ルーチン４０４から出ることを可能にする。インターフェイスボタン４４０及び４４２は共にユーザが許可設定ルーチン４０４を終了することを可能にするものであるが、キャンセルインターフェイスボタン４４０は、レシピ許可設定に対して行われた変更を含まずに許可設定ルーチン４０４を終了させる。逆に、ＯＫインターフェイスボタン４４２は、ユーザが許可設定ルーチン４０４を終了することを可能にすると共に、ユーザインターフェイス４２０を用いて許可設定に対して行われた変更を保存する。

【００６８】

次に図９に移ると、追加ルーチンを表すブロック４１２の更なる詳細が提供される。追加ルーチン４１２はブロック４５０で実行を開始し、ユーザによって提供された機能ロック選択を受け取る。図１０に示されるように、グラフィカル・ユーザ・インターフェイス又はポップアップウインドウ４５２は、承認機能ロックボックス４５４を含んでよく、ここに、ユーザが承認機能ロックの名前を入力してもよい。例えば、図１０に示されるように、ボックス４５４は、選択された承認機能ロックがＲＥＣＩＰＥ＿ＡＰＰＲＯＶＡＬ＿０３であることの標示を含んでもよい。

【００６９】

図９に戻ると、ブロック４５０が機能ロック選択を受け取った後、ブロック４６０は、ユーザによって提供される署名者記述を受け取る。例えば、図１０のユーザインターフェイス４５２に示されるように、ユーザは、ブロック４６２に署名者記述を入力してもよい。例として、ブロック４６２には「Ｔｅａｍ　Ｌｅａｄｅｒ」という記述が示されており、これは、ユーザが、ＲＥＣＩＰＥ＿ＡＰＰＲＯＶＡＬ＿０３の承認機能ロックを有する署名者としてチームリーダーを追加することを所望することを示す。

【００７０】

ブロック４５０及び４６０で機能ロック選択及び署名者記述をそれぞれ受け取った後、制御はブロックに４６６に渡り、そこで、機能ロック又は署名者記述のいずれかが欠けているか否か、或いは、図１０の参照番号４７０及び４７２でそれぞれ示されるキャンセルインターフェイスボタン又はＯＫインターフェイスボタンが選択されたか否かが判定される。ロック又は記述が欠けている場合には、制御はブロック４６６からブロック４５０に渡る。或いは、ブロック４６６が、ユーザによってキャンセルインターフェイスボタン４７０又はＯＫインターフェイスボタン４７２が選択されたと判定した場合には、追加ルー

チン４１２は実行を終了し、制御を図７の許可設定ルーチン４０４に戻す。図８のユーザインターフェイス４２０に関して記載したように、キャンセルインターフェイスボタン４７０を作動させると、追加ルーチン４１２は、その実行中に行われた変更を保存せずに実行を終了する。逆に、既述のように、ＯＫインターフェイスボタン４７２を作動させると、追加ルーチン４１２は終了すると共に、追加ルーチン４１２の実行中に行われた変更を保存する。追加ルーチン４１２によって新たな承認者又は署名者が追加された場合には、以前に承認されたいかなるレシピ（即ち、最初に必要とされた全ての承認が受け取られたレシピ）も、新たに追加された署名者からの承認が取得されるまで、自動的に未許可になる。

【００７１】

図１１との関連において、削除ルーチン４１４の更なる詳細を提供する。削除ルーチン４１４は、図８のユーザインターフェイス４２０と関連して動作する。具体的には、削除ルーチン４１４は、ブロック４８０で実行を開始し、削除すべき署名者記述の選択を受け取る。ユーザは、図８のユーザインターフェイス４２０のテキストボックス４３０に示される署名者記述を選択することによって、そのような選択を提供してもよい。ユーザが削除すべき記述を選択した後、ユーザは、選択された署名者記述又は署名者を削除するという自分の意図を宣言するために、削除インターフェイスボタン４２８を作動させる。ブロック４８０の実行完了後、制御はブロック４８２に渡り、ユーザによって要求された削除に対する確認を受け取る。例えば、ユーザが削除すべき署名者記述を選択し、削除インターフェイスボタン４２８を作動させた後で、削除ルーチン４１４は、表示画面上でユーザに対して表示されるユーザインターフェイスのグラフィックを介して、ユーザに、選択された署名者記述の削除を望むことを確認するよう要求してもよい。そのようなグラフィックは、ＯＫインターフェイスボタン又はキャンセルインターフェイスボタンを含んでもよく、この場合、ＯＫインターフェイスボタンの作動（例えば、マウス、キーボード等を介した選択）により、ユーザが選択された署名者記述の削除を望むことが確認され、キャンセルインターフェイスボタンは、選択された記述の削除を中止する。ブロック４８２で削除の確認を受け取った後、削除ルーチン４１４は実行を終了し、制御を許可設定ルーチン４０４に戻す。

【００７２】

図１２及び図１３との関連において、図７の修正ルーチン４１６に関する更なる詳細が提供される。修正ルーチン４１６はブロック４８４で実行を開始し、ユーザから、修正すべき署名者記述の選択を受け取る。例えば、ユーザは、図８のＱｕａｌｉｔｙ Ａｓｓｕｒａｎｃｅという名前の署名者を選択してから、修正インターフェイスボタン４２６を作動させてもよい。修正インターフェイスボタン４２６の作動後、例えば、図１３に示されるユーザインターフェイス４８６等といったユーザインターフェイスがユーザに対して表示されてもよく、ユーザインターフェイスは、署名者記述ボックス４８８及び承認機能ロックボックス４９０を含んでもよい。ユーザインターフェイス４８６は、ＯＫインターフェイスボタン４９２及びキャンセルインターフェイスボタン４９４も含んでよい。修正ルーチン４１６が修正すべき署名者記述の選択（このケースでは、署名者Ｑｕａｌｉｔｙ Ａｓｓｕｒａｎｃｅが修正のために選択されている）を受け取った後、制御はブロック４８４からブロック４９６に渡る。ブロック４９６は、例えば、署名者名や承認ロック機能の変更、又はその他の任意の適切な変更等といった、署名者記述の修正を受け取る。例えば、ブロック４８８でユーザが署名者記述を提供した後、ユーザは署名者の名前を修正してもよく、又は、ブロック４９０に表示される承認ロック機能を修正してもよく、次にＯＫインターフェイスボタン４９２又はキャンセルインターフェイスボタン４９４を選択してもよい。上述したように、ＯＫインターフェイスボタン４９２を作動させると、署名者記述に対して行われた修正が保存される。逆に、キャンセルインターフェイスボタン４９４を作動させると、行われた変更を保存せずに、修正ルーチン４１６が終了する。いずれにしても、インターフェイスボタン４９２及び４９４のいずれかを作動させると、修正ルーチン４１６の実行が終了し、制御は図７の許可設定ルーチン４０４に戻る。追加ルーチ

ン４１２と同様に、署名者又は承認者を修正すると、その署名者の承認を要する以前に承認されたレシピは、自動的に未許可になる。

【００７３】

ここまで、署名者、レシピ検査者、又は承認者の追加、削除、及び修正の説明が提供された。記載されたルーチン又はこれらのルーチンとの関連において記載された機能性を具現化するルーチンは、図１の任意のワークステーション１４及び／又は端末３２において又は図４の任意のワークステーション１１６で実行されてよい。

【００７４】

先の図面及び説明は署名者の特定に関するものであったが、図１４～図１８は、署名者によって行われ得る検査、承認又は拒絶プロセスに関する。図１４～図１８に示されるルーチン及びユーザインターフェイスは、図１の端末３２及び／又はワークステーション１４上で、又は図４の任意のワークステーション１１６上で実行されてよい。具体的には、メモリ２０及び３４の１つ以上は、ルーチン内のブロックが代表する動作を実行するためにプロセッサ２１及び３６の１つ以上によって実行され得る命令を格納してもよい。

【００７５】

次に図１４に移ると、レシピ許可ルーチン５００はブロック５０２で実行を開始し、検査されるレシピに関する署名者及びステータス情報を表示する。例えば、図１５のユーザインターフェイス５０４は、署名者識別、ステータス、ユーザタイプ、時間、コメント、及びノードを表してもよい複数の列５０８～５１８を有するテキストボックス５０６を含んでよい。署名者列５０８は、レシピの承認に必要な署名を列挙し、ステータス列５１０は、各署名者の署名の状態を列挙する。例えば、署名ステータスは、ブランク若しくは未決（Ｐｅｎｄｉｎｇ）、承認（Ａｐｐｒｏｖｅｄ）、又は拒絶（Ｒｅｊｅｃｔｅｄ）であってもよく、ブランクステータス又は未決ステータスは、その署名者がまだレシピを検査していないことを表してもよい。ユーザ列５１２は、最新の署名の変更に対して責任があるユーザタイプを列挙する。時間列５１４は、最新の署名の変更が行われた時間を列挙する。コメント列５１６は、レシピの承認又は拒絶時に署名者によって作成された任意のコメントを列挙し、ノード５１８は、署名者がレシピを承認又は拒絶したシステムノードを表す。例えば、ノードは、図１の端末３２及び／又はワークステーション１４、又は図４のワークステーション１１６のいずれか１つであってもよい。ユーザインターフェイス５０４は、テキストブロック５０６に加えて、閉じる、承認、拒絶、及びクリアインターフェイスボタン５２０～５２６を含んでもよく、これらについて、図１４のレシピ許可ルーチン５００と関連して説明する。

【００７６】

ブロック５０２が署名者及びステータス情報を表示した後、ブロック５３０は、ユーザが任意のインターフェイスボタン５２０～５２６を選択することによって明らかにしてもよい署名者選択を受け取る。具体的には、ユーザが「閉じる」インターフェイスボタン５２０を作動させた場合には、レシピ許可ルーチン５００の制御はブロック５３０からブロック５４０に渡り、ユーザインターフェイス５０４を閉じて、レシピ許可ルーチン５００の実行を終了し、レシピ許可ルーチン５００を呼び出した任意のルーチンに制御を戻す。

【００７７】

或いは、ユーザが承認インターフェイスボタン５２２を作動させた場合には、制御は、ブロック５３０から承認ルーチンを表すブロック５５０に渡る。図１６に示されるように、承認ルーチン５５０はブロック５５２で実行を開始し、ユーザによって提供されたユーザ名及びパスワードを受け取る。図１７に一例が示されるユーザインターフェイス５５４は、ユーザが自分のユーザ名及びパスワードを入力し得るユーザ名ボックス５５６及びパスワードボックス５５８を含んでもよい。

【００７８】

ブロック５５２の実行完了後、制御はブロック５６０に渡り、承認中に作成されたユーザコメントを受け取る。例えば、図１７のユーザインターフェイス５５４は、コメントがキーボード入力され得るテキストボックス５６２を含んでもよい。ブロック５６０の実行

完了後、ブロック５６１は、そのユーザが許可されているか否かを判定する。ブロック５６１で行われる許可チェックは、ブロック５５２で受け取ったユーザ名及び／又はパスワードが有効であることを検証してもよく、それと共に／又は、そのユーザ名及びパスワードと関連付けられたユーザがそのような承認を行うことが許可されているか否かを検証してもよい。ブロック５６１でそのユーザが許可されていると判定された場合には、制御はブロック５６６に渡る。ブロック５６６は、承認を反映するためにステータス情報を更新する。例えば、テキストボックス５６２は「This one is ready for production」というテキストコメントを含み、これは、ブロック５６６の実行後に、レシピ承認時にＰｒｏｄｕｃｔｉｏｎの署名者によって作成された図１５のコメントとしても反映される。ブロック５６１で、ブロック５５２で受け取ったユーザ名及びパスワードのいずれか又は両方が許可されていないと判定された場合には、承認ルーチン５５０は終了する。

【００７９】

多くの上述のユーザインターフェイス画面との関連において記載したように、図１７のユーザインターフェイス５５４はＯＫインターフェイスボタン５６８及びキャンセルインターフェイスボタン５７０を含み、これらは、ルーチンの実行中に行われた変更を保存して又は破棄して、承認ルーチン５５０の実行を終了させるために用いられてもよい。更に、図１７に示されるように、任意の含まれる又はサブレシピにこの承認を波及させることをユーザが選ぶことを可能にするために、チェックボックス５７２が提供されてもよい。

【００８０】

図１４及び図１５に戻ると、ユーザが図１５の拒絶インターフェイスボタン５２４を作動させた場合には、制御はレシピ許可ルーチン５５０のブロック５３０からブロック５８０に渡る。ブロック５８０は拒絶ルーチンを表し、その更なる詳細は図１８に見出され得る。図１８に示されるように、拒絶ルーチン５８０の実行はブロック５８２で開始し、ここで、ユーザがユーザ名及びパスワードを入力し、制御はブロック５８４に渡る。ブロック５８４では、署名者がレシピの拒絶プロセス中に作成したコメントを入力してもよい。ブロック５８２及び５８４の動作は、ブロック５８２及び５８４はレシピの拒絶と関連して用いられることを除き、図１６に示される承認ルーチン５５０のブロック５５２及び５６０の動作と同様である。ブロック５８４の実行完了後、制御はブロック５８５に渡り、図１７に示されるブロック５６１で行われるものと同様の許可チェックを行う。ブロック５８５でユーザが許可されていると判定された場合には、制御はブロック５８６に渡る。

【００８１】

ブロック５８６は、ユーザによるレシピの拒絶を反映するために、ステータス情報を更新する。ステータス情報更新ブロック５８６は、署名者がレシピを拒絶した事実を反映するために図１５のユーザインターフェイス５０４上に反映される情報を生成してもよい。図示しないが、拒絶ルーチン５８０は、図１７のレシピ承認に用いられるユーザインターフェイス５５４と同様のグラフィカル・ユーザ・インターフェイスを用いてもよい。

【００８２】

再び図１４及び図１５に戻ると、ユーザが図１５のクリアインターフェイスボタン５２６を作動させた場合には、制御はレシピ許可ルーチン５００のブロック５３０からブロック５９０に渡る。ブロック５９０は、署名をクリアするために用いられてもよい。例えば、図１５に示される署名者の１つがユーザによって選択されて、インターフェイスボタン５２６を用いてクリアされてもよい。しかし、一旦レシピが、例えば、コントローラ１２（図１）又はワークステーション１４（図１）による実行のためにダウンロードされたら、承認署名の効果を撤回することはできない。換言すれば、一旦レシピ（又は他の任意のソフトウェアオブジェクト）がダウンロードされたら、署名（即ち、承認）をクリア又は拒絶することはできない。

【００８３】

上記説明は署名者の選択及びレシピの検査に関するものであるが、図１９に示されるように、ユーザインターフェイス６００は、プロセス制御システム１０内のレシピのステータスを報告するために用いられてもよい。例えば、ユーザインターフェイス６００は、レ

シピ名、製造（Ｐｒｏｄｕｃｔｉｏｎ）、エンジニアリング（Ｅｎｇｉｎｅｅｒｉｎｇ）、品質保証（Ｑｕａｌｉｔｙ　Ａｓｓｕｒａｎｃｅ）、及びチームリーダー（Ｔｅａｍ　Ｌｅａｄｅｒ）をそれぞれ表す複数の列６０２～６１０を含んでもよい。簡単に言えば、レシピ名列６０２は全ての未承認レシピを列挙し、列６０４～６１０は各検査者又は検査エンティティにおける各レシピのステータスを列挙する。例えば、ＯＰ＿ＣＨＡＲＧＥという名前のレシピは、製造、エンジニアリング、品質保証、及びチームリーダーの各々において未決である。一方、製造、エンジニアリング、品質保証、及びチームリーダーの各々は、ＰＲＣ＿ＰＡＩＮＴというレシピを承認しているが、品質保証はこのレシピを承認していない。従って、ＰＲＣ＿ＰＡＩＮＴレシピは依然として未承認である。ユーザインターフェイス６００は、ユーザインターフェイス６００を閉じるため又は列６０２～６１０に含まれる情報を示すためにユーザインターフェイス６００をプリントするために用いられ得る「閉じる」インターフェイスボタン６１２及び「プリント」インターフェイスボタン６１４を含んでもよい。

【００８４】

レシピが検査され、全ての署名者によって承認されたら、そのレシピは図１に示されるコントローラ１２の１つ以上又は図４のコントローラ１７６にダウンロードされてもよく、又はそれらの装置において実装されてもよい（或いは、他のソフトウェアオブジェクトが図４の安全論理モジュール１５０～１５６又はこれらのオブジェクトのダウンロードを必要とし得る任意のフィールド装置にダウンロードされてもよい）。図２０に示されるように、ダウンロードルーチン６３０は、ダウンロードを実行し得る１つの方法である。ダウンロードルーチン６３０はブロック６３２で実行を開始し、ダウンロードスクリプトを生成する。ブロック６３２でダウンロードスクリプトが生成されたら、制御はブロック６３４に渡り、レシピがチェックアウトされていないか（即ち、チェックインされているか）、又は、レシピがチェックアウトされている場合であっても、ユーザがレシピのダウンロードを可能にするキーを供給したかを判定する。ダウンロードルーチン６３０と関連して、米国特許第６，４４９，６２４号に開示されているソフトウェア等といったバージョン制御ソフトウェアが用いられてもよい。ブロック６３４が、レシピがチェックアウトされており、キーが提供されていないと判定した場合には、制御はブロック６３６に渡り、ダウンロードをキャンセルして、ダウンロードルーチン６３０の実行を終了する。或いは、レシピがチェックアウトされていないか、又はキーが提供されている場合には、制御はブロック６３４からブロック６３８に渡り、そのレシピが許可されているか、又はユーザが未許可のレシピのダウンロードを可能にする特別なキーを提供しているかを判定する。レシピの許可は、全ての署名者がそのレシピを承認していることの保証を含み得るが、これに限定されない。ブロック６３８が、そのレシピが許可されておらず、キーも提供されていないと判定した場合には、制御はブロック６３６に渡り、ダウンロードをキャンセルしてから、ダウンロードルーチン６３０を終了する。或いは、ブロック６３８が、そのレシピが許可されていると判定した場合、又はキーが提供されている場合には、制御はブロック６４０に渡り、ダウンロードラベルを設定する。ダウンロードラベルは、ダウンロードされる項目に添付される、時間、日付、バージョン、及びダウンロードの開始者（又はユーザ）を含む１つ以上のコメント文又は他の類似のテキスト情報であってもよい。更に、ダウンロードラベルは、ダウンロードされる個々の項目（例えば、レシピ）の詳細なリストを含む。次に、ブロック６４２で、レシピは、例えば、図１のコントローラ１２内で具現化されることができるランタイムシステムに送られる。ブロック６４２の実行後、ダウンロードルーチン６３０は実行を終了し、ダウンロードルーチン６３０を呼び出したルーチンに制御を戻す。

【００８５】

上記の説明から、現在承認されていないソフトウェアオブジェクトは、そのソフトウェアオブジェクトと関連付けられた全ての署名者又は承認者がソフトウェアオブジェクトを承認するまで、プロセス制御又は安全システムによってダウンロード又は実装できないことがわかる。従って、新たなソフトウェアオブジェクト又はレシピは、例えば、所定のリ

スト又はグループの人々及び／又は他のエンティティ（例えば、図７の許可設定ルーチン４０４によって生成されるリストの人々及び／又は他のエンティティ）によって承認されなければならない。更に、以前に承認されたソフトウェアオブジェクト又はレシピが修正されたら、そのソフトウェアオブジェクト又はレシピは自動的に未許可になり、従って、図２０のブロック６３８及び６４０に例として示されるように、修正されたソフトウェアオブジェクト又はレシピをダウンロードするには、全ての対応する署名者又は許可者によって再承認されなければならない。

【００８６】

本願明細書では、本発明の教示に従って構成された特定の装置を説明したが、本特許が包含する範囲はこれに限定されない。むしろ、本特許は、添付の特許請求の範囲に文言として又は均等論の下で正当に含まれる、本発明の教示の全ての実施形態を包含する。

【図面の簡単な説明】

【００８７】

【図１】プロセス機器の制御を実行する１つ以上の制御ソフトウェアオブジェクトを承認するための、統合型電子的ソフトウェアオブジェクト承認システムを含む、プロセス制御システムの部分ブロック図である。

【図２】図１のプロセス制御システムの一般的な論理階層又はコンフィグレーション（構成）を示すオブジェクト構成のブロック図である。

【図３】図２のオブジェクト構成の一部をより詳細に示す、より詳細なブロック図である。

【図４】統合型電子的ソフトウェアオブジェクト承認システムを含む安全システムがプロセス制御システムと統合されている例示的なプロセスプラントのブロック図である。

【図５Ａ】承認ルーチンの例示的なフロー図である。

【図５Ｂ】承認ルーチンの例示的なフロー図である。

【図６】ソフトウェアオブジェクト編集ルーチンの例示的なフロー図である。

【図７】許可設定ルーチンの例示的なフロー図である。

【図８】図７の許可設定ルーチンと関連付けられた例示的なユーザインターフェイスを示す図である。

【図９】追加ルーチンの例示的なフロー図である。

【図１０】図９の追加ルーチンと関連付けられた例示的なユーザインターフェイスを示す図である。

【図１１】削除ルーチンの例示的なフロー図である。

【図１２】修正ルーチンの例示的なフロー図である。

【図１３】図１２の修正ルーチンと関連付けられた例示的なユーザインターフェイスを示す図である。

【図１４】ソフトウェアオブジェクト許可ルーチンの例示的なフロー図である。

【図１５】図１４のソフトウェアオブジェクト許可ルーチンと関連付けられた例示的なユーザインターフェイスを示す図である。

【図１６】承認ルーチンの例示的なフロー図である。

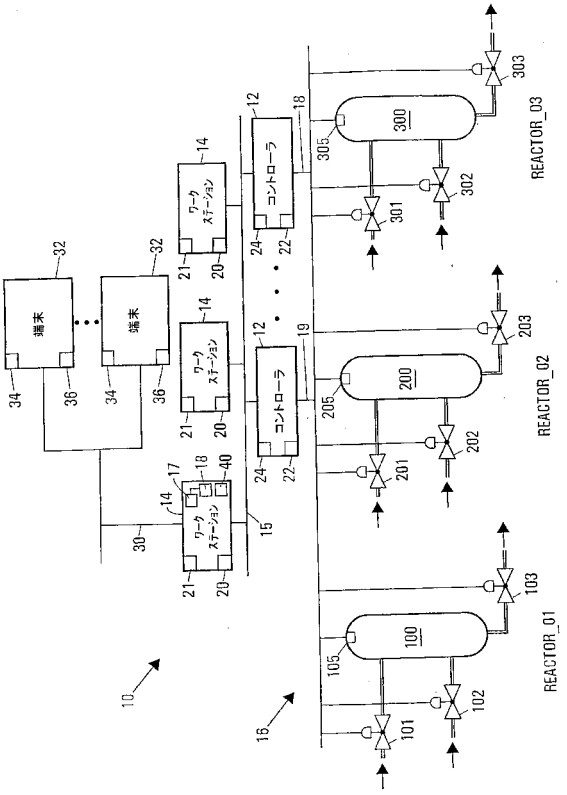
【図１７】図１６の承認ルーチンと関連付けられた例示的なユーザインターフェイスを示す図である。

【図１８】拒絶ルーチンの例示的なフロー図である。

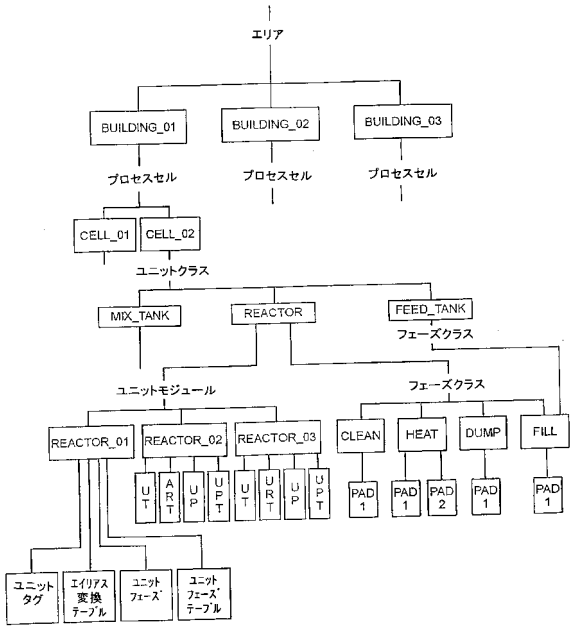
【図１９】未承認のソフトウェアオブジェクトのステータスを示す例示的なユーザインターフェイスを示す図である。

【図２０】ダウンロードルーチンの例示的なフロー図である。

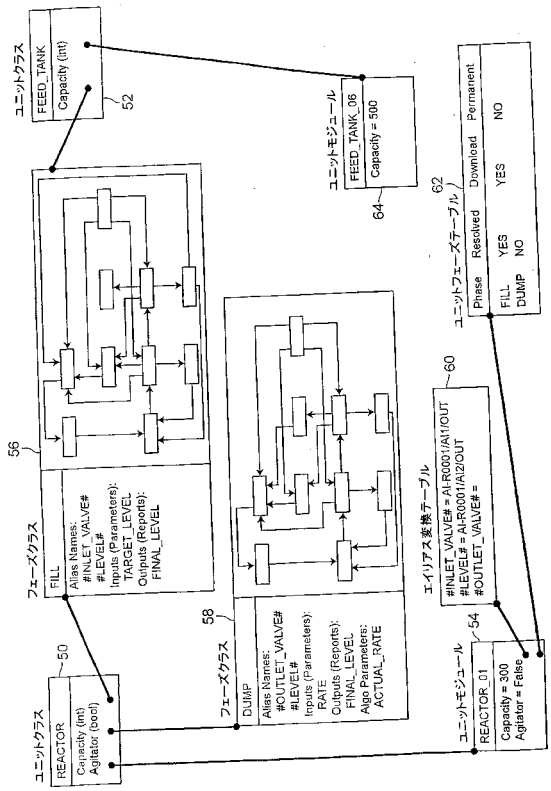
【図１】


REACTOR_03
REACTOR_02
REACTOR_01
コントローラ
ワークステーション
端末
300
200
100

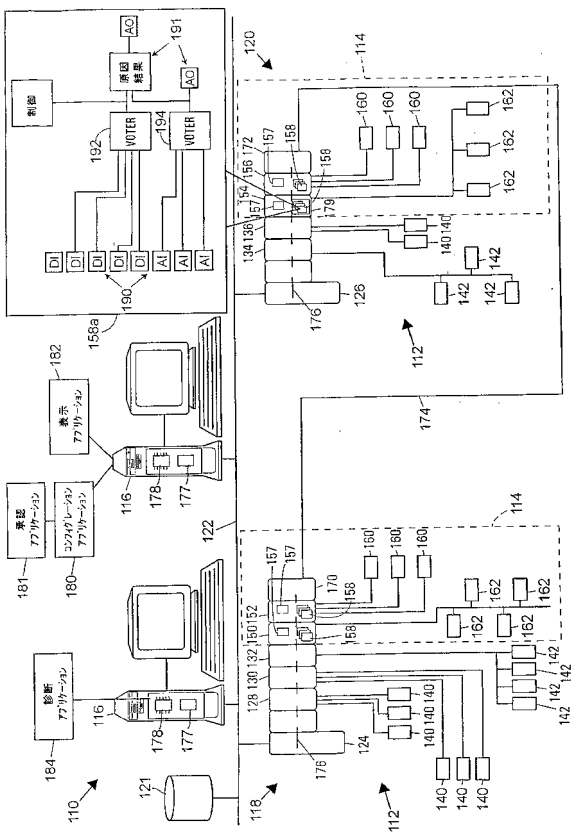
【図２】


エリア
BUILDING_01
BUILDING_02
BUILDING_03
プロセスセル
CELL_01
CELL_02
ユニットクラス
MIX_TANK
REACTOR
FEED_TANK
フェーズクラス
ユニットモジュール
REACTOR_01
REACTOR_02
REACTOR_03
CLEAN
HEAT
DUMP
FILL
PAD 1
PAD 2
ユニットタグ
エイリアス変換テーブル
ユニットフェーズ
ユニットフェーズテーブル

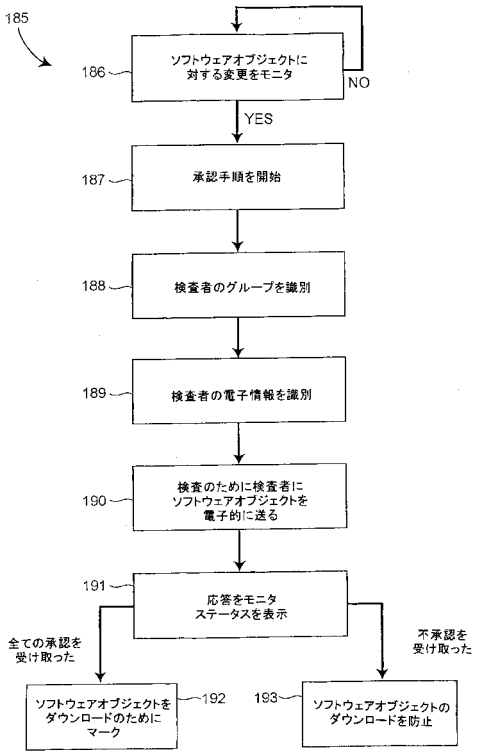
【図３】


ユニットクラス
FEED_TANK
Capacity (int)
ユニットモジュール
FEED_TANK_06
Capacity = 500
ユニットフェーズテーブル
Phase
Resolved
Download
Permanent
FILL
DUMP
YES
NO
フェーズクラス
FILL
Alias Names:
#INLET_VALVE#
#LEVEL#
Inputs (Parameters):
TARGET_LEVEL
Outputs (Reports):
FINAL_LEVEL
DUMP
Alias Names:
#OUTLET_VALVE#
#LEVEL#
Inputs (Parameters):
RATE
Outputs (Reports):
FINAL_LEVEL
Algo Parameters:
ACTUAL_RATE
エイリアス変換テーブル
#INLET_VALVE# = AI-R0001/AI1/OUT
#LEVEL# = AI-R0001/AI2/OUT
#OUTLET_VALVE# =
ユニットクラス
REACTOR
Capacity (int)
Agitator (bool)
ユニットモジュール
REACTOR_01
Capacity = 300
Agitator = False

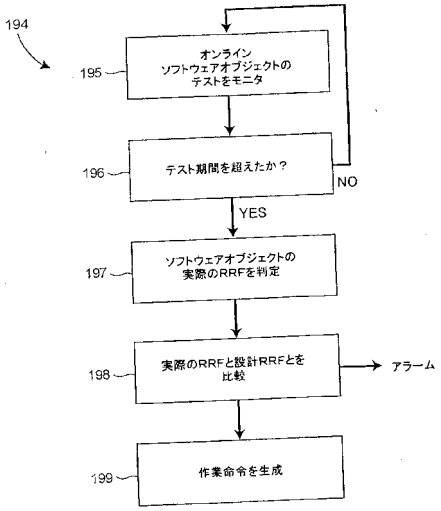
【図４】


制御
表示
アプリケーション
診断
VOTER
DI
AI
AO

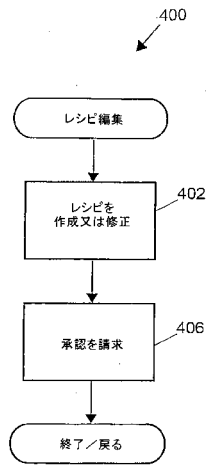
【図５Ａ】



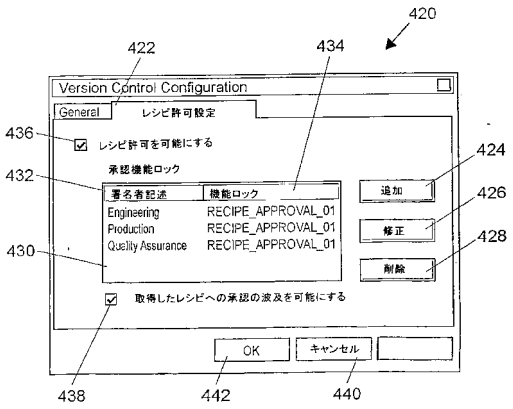
【図５Ｂ】



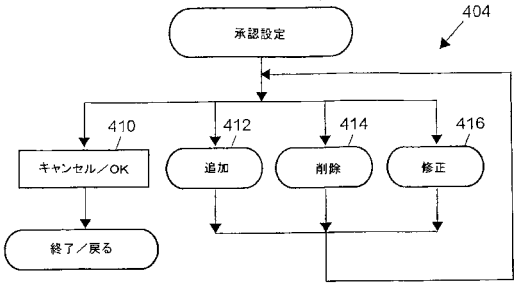
【図６】



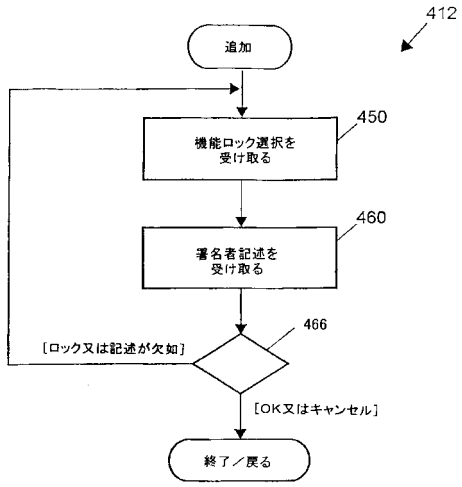
【図８】



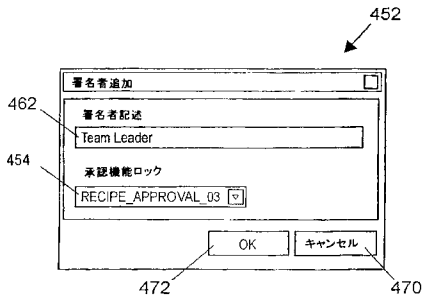
【図７】



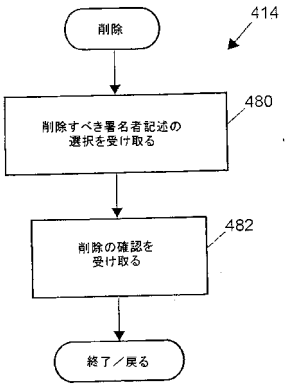
【図９】



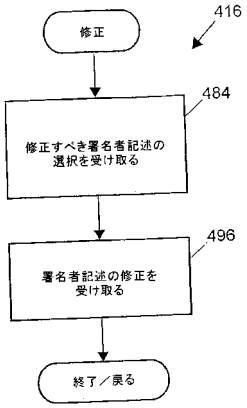
【図１０】



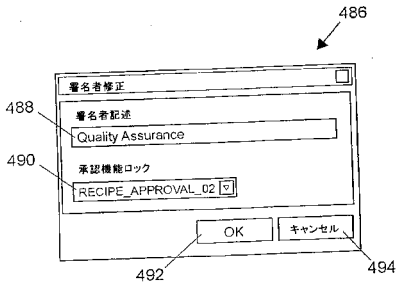
【図１１】



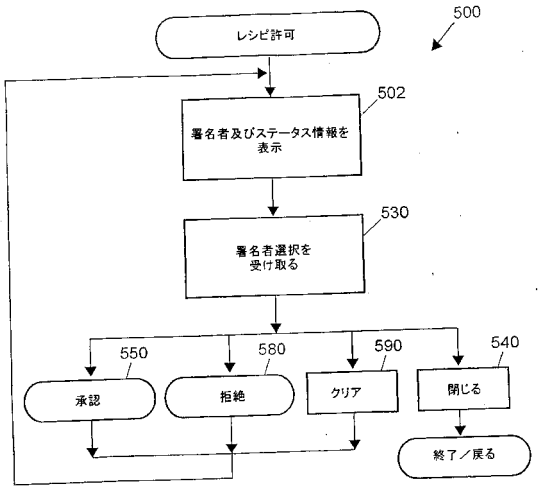
【図１２】



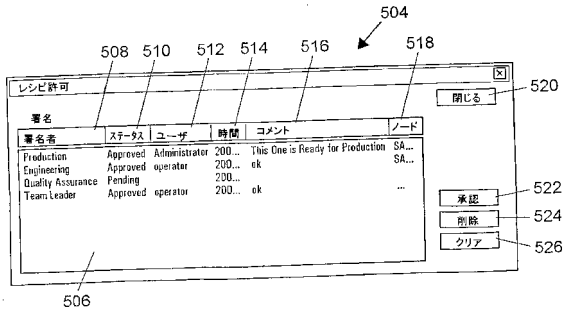
【図１３】



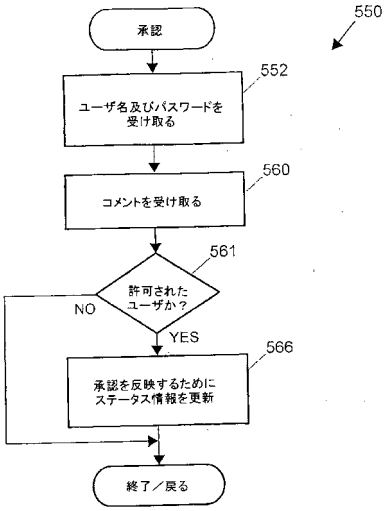
【図１４】



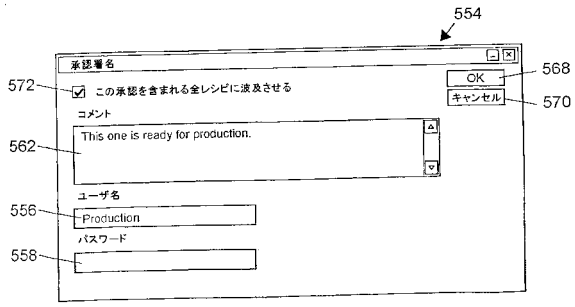
【図１５】



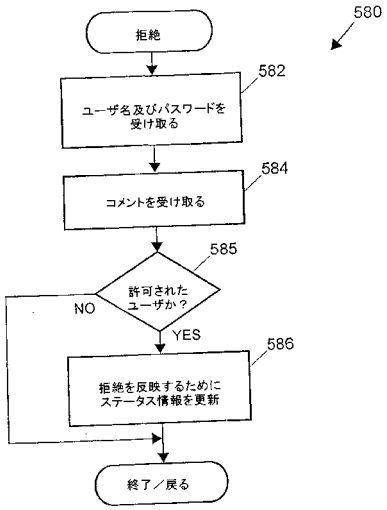
【図１６】



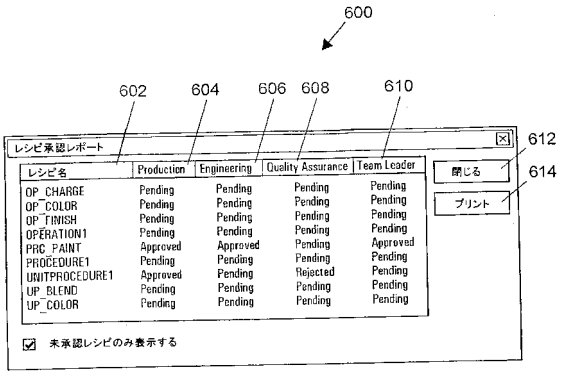
【図１７】



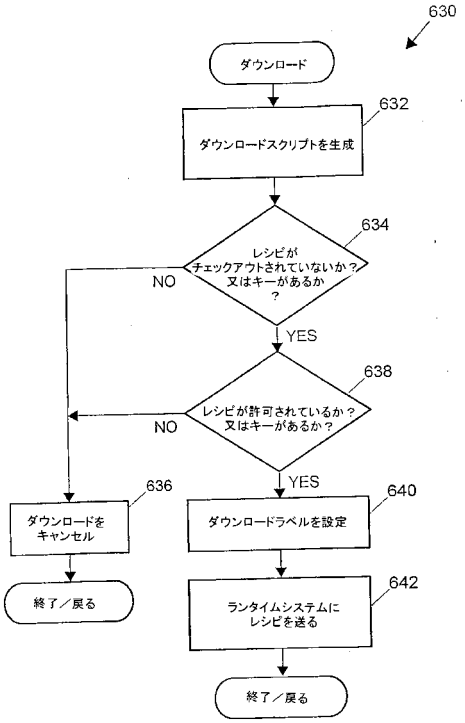
【図１８】



【図１９】



【図２０】



フロントページの続き

(72)発明者 ロー、ゲアリ、ケー.
アメリカ合衆国 ７８６２８－４３２０ テキサス州 ジョージタウン ミシェル コート １１０
(72)発明者 デイツ、デヴィッド、エル.
アメリカ合衆国 ７８７３１ テキサス州 オースティン マウンテン ヴィラ ドライブ ５９１５
(72)発明者 シュライス、トレヴァー、ダンカン
アメリカ合衆国 ７８７３０ テキサス州 オースティン リーニング ロック サークル ９１０８
(72)発明者 ナイドゥー、ジュリアン、ケー.
アメリカ合衆国 ７８６１３ テキサス州 シダー パーク メスキート ロード １７１１

審査官 佐藤 裕子

(56)参考文献 特開２００２－１１６８０１（ＪＰ，Ａ）
特開平１０－２２２３５１（ＪＰ，Ａ）
特開平０１－１６０１５８（ＪＰ，Ａ）
特開平０５－０３５４６０（ＪＰ，Ａ）
特開平１１－２７２７７７（ＪＰ，Ａ）
特開平７－１４６７８７（ＪＰ，Ａ）
特開平１０－２１４１１３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06Q 10/00-50/00