

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7015625号
(P7015625)

(45)発行日 令和4年2月3日(2022.2.3)

(24)登録日 令和4年1月26日(2022.1.26)

(51)国際特許分類

G 0 6 F	11/36 (2006.01)	F I	G 0 6 F	11/36	1 6 4
G 0 6 F	8/34 (2018.01)		G 0 6 F	8/34	
G 0 6 F	8/60 (2018.01)		G 0 6 F	8/60	
			G 0 6 F	11/36	1 1 2

請求項の数 23 (全20頁)

(21)出願番号	特願2018-550499(P2018-550499)
(86)(22)出願日	平成29年3月21日(2017.3.21)
(65)公表番号	特表2019-521402(P2019-521402)
A)	
(43)公表日	令和1年7月25日(2019.7.25)
(86)国際出願番号	PCT/US2017/023426
(87)国際公開番号	WO2017/200633
(87)国際公開日	平成29年11月23日(2017.11.23)
審査請求日	令和2年3月5日(2020.3.5)
(31)優先権主張番号	62/337,156
(32)優先日	平成28年5月16日(2016.5.16)
(33)優先権主張国・地域又は機関	米国(US)
(31)優先権主張番号	15/417,635
(32)優先日	平成29年1月27日(2017.1.27)
	最終頁に続く

(73)特許権者	508011614 ティプコ ソフトウェア インク アメリカ合衆国、カリフォルニア州 9 4 3 0 4、パロ アルト、ヒルビュー ア ベニュー 3 3 0 3
(74)代理人	100104411 弁理士 矢口 太郎
(72)発明者	マルティネズ、フランシスコ アメリカ合衆国、9 4 3 0 4 カリフォ ルニア州、パロ アルト、3 3 0 3 ヒル ビュー アベニュー
(72)発明者	レジスター、マイケル アメリカ合衆国、9 4 3 0 4 カリフォ ルニア州、パロ アルト、3 3 0 3 ヒル ビュー アベニュー

最終頁に続く

(54)【発明の名称】 プロセスアプリケーションを開発及び診断するためのステップバック機構

(57)【特許請求の範囲】**【請求項1】**

複数のデバイスにおいてインスタンス化されたプロセルアプリケーションを診断および更新するためのシステムであって、このシステムはプロセッサを有し、このプロセッサは、

記憶リソースと通信可能に接続され、この記憶リソースに格納されたアプリケーションコード命令を実行するように構成され、それによって、以下の工程：

状態遷移情報をデバイスにローカルに格納する工程であり、この状態遷移情報は、インスタンス化されたプロセスエンジンによってインスタンス化されたプロセスアプリケーションの実行されたステップに関連付けられたものであり、前記インスタンス化されたプロセスアプリケーションは、誤った結果を生成したものである、前記格納する工程と；

フロー定義を遠隔診断プラットフォームに送信する工程と；

前記遠隔診断プラットフォームにおいて、前記フロー定義と前記状態遷移情報を受信する工程であり、前記フロー定義は、プロセスフローを記述し、前記状態遷移情報は、前記インスタンス化されたプロセスアプリケーションに関連付けられた前記プロセスフローのステップにおいて変更されたデータ値の情報を含むものである、前記受信する工程と；

前記状態遷移情報を使って、前記デバイスの前記インスタンス化された前記プロセスエンジンに関連付けられたプロセスエンジンを再生成する工程と；

前記プロセスフローの選択されたステップに関連付けられた1つ以上の入力データ値を変更する工程と；

前記変更された 1 つ以上の入力データ値を使用して、前記選択されたステップから前記プロセスフローを実行する工程と；
 プロセスアプリケーションの前記選択されたステップに関連付けられた少なくとも 1 つのデータレコードを更新して、前記変更された 1 つ以上の入力データ値と、前記プロセスフローのステップに関連付けられた状態情報を反映する工程と；
 前記プロセスアプリケーションを再度コンパイルする工程と；
 デバイス識別子を使用して前記デバイスに前記プロセスアプリケーションを再度デプロイする工程と；
 を行う、システム。

【請求項 2】

請求項 1 記載のシステムにおいて、前記プロセッサは、前記プロセスフローを記述するフロー定義および前記プロセスフローの各ステップにおける前記プロセスフローの状態を記述する状態情報を受け取る工程を行うように構成されるものである、システム。

【請求項 3】

請求項 2 記載のシステムにおいて、前記フロー定義が `flow.json` ファイルである、システム。

【請求項 4】

請求項 2 記載のシステムにおいて、前記プロセッサは、前記フロー定義および前記状態情報に基づいてプロセスアプリケーションを再構成する工程を行うように構成され、前記プロセスアプリケーションは前記プロセスフローを含むものである、システム。

【請求項 5】

請求項 2 記載のシステムにおいて、前記プロセッサはさらに、前記プロセスアプリケーションを実行可能ファイルにコンパイルする工程を行うように構成される、システム。

【請求項 6】

請求項 5 記載のシステムにおいて、前記プロセッサはさらに、前記コンパイルされた実行可能ファイルを前記状態情報を生成したデバイスにデプロイする工程を行うように構成される、システム。

【請求項 7】

請求項 2 記載のシステムにおいて、前記状態情報は、前記状態情報を生成したデバイス及び前記システムとは別のデータベースに格納されるものである、システム。

【請求項 8】

請求項 2 記載のシステムにおいて、前記フロー定義は、前記状態情報を生成した前記デバイス、前記状態情報を格納するデータベース、及び前記システムとは別のデータベースに格納されるものである、システム。

【請求項 9】

請求項 2 記載のシステムにおいて、前記状態情報を生成したデバイスは、状態サービスを格納するデータベースに状態値をアップロードするものである、システム。

【請求項 10】

請求項 9 記載のシステムにおいて、前記状態値は、状態遷移の発生時にアップロードされるものである、システム。

【請求項 11】

請求項 9 記載のシステムにおいて、前記状態情報を生成した前記デバイスから前記状態情報を格納する前記データベースに前記状態値が定期的にアップロードされるものである、システム。

【請求項 12】

請求項 9 記載のシステムにおいて、前記状態情報を生成した前記デバイスと前記状態情報を格納する前記データベースとの間のネットワーク接続に障害が生じた場合、再接続時に前記状態情報を前記データベースにアップロードされるものである、システム。

【請求項 13】

10

20

30

40

50

請求項 1 記載のシステムにおいて、前記プロセッサは、前記プロセスフローを含むプロセスアプリケーションを実行するために必要なプロセスエンジンコンポーネントの最小セットを決定する工程を行うように構成されている、システム。

【請求項 14】

請求項 1 記載のシステムにおいて、前記プロセスアプリケーションをプリコンパイルして、前記プロセスアプリケーションを実行するために必要な最小限のステップのセット及び不要なステップを決定する工程を、有するものであるシステム。

【請求項 15】

請求項 14 記載のシステムにおいて、プロセスステップは、前記必要なステップの任意のステップが選択可能であり、前記必要なステップの前記選択されたステップから実行可能なものである、システム。

10

【請求項 16】

請求項 1 記載のシステムにおいて、前記プロセスアプリケーションは、複数のデバイス識別子を使って複数のデバイスにデプロイされるものである、システム。

【請求項 17】

複数のデバイスにおいてインスタンス化されたプロセルアプリケーションを診断および更新するための方法であって、

状態遷移情報をデバイスにローカルに格納する工程であり、この状態遷移情報は、インスタンス化されたプロセスエンジンによってインスタンス化されたプロセスアプリケーションの実行されたステップに関連付けられたものであり、前記インスタンス化されたプロセスアプリケーションは、誤った結果を生成したものである、前記格納する工程と；

20

フロー定義を遠隔診断プラットフォームに送信する工程と；

前記遠隔診断プラットフォームにおいて、前記フロー定義と前記状態遷移情報を受信する工程であり、前記フロー定義は、プロセスフローを記述し、前記状態遷移情報は、前記インスタンス化されたプロセスアプリケーションに関連付けられた前記プロセスフローのステップにおいて変更されたデータ値の情報を含むものである、前記受信する工程と；

前記状態遷移情報を使って、前記デバイスの前記インスタンス化された前記プロセスエンジンに関連付けられたプロセスエンジンを再生成する工程と；

前記プロセスフローの選択されたステップに関連付けられた 1 つ以上の入力データ値を変更する工程と；

30

前記変更された 1 つ以上の入力データ値を使用して、前記選択されたステップから前記プロセスフローを実行する工程と；

プロセスアプリケーションの前記選択されたステップに関連付けられた少なくとも 1 つのデータレコードを更新して、前記変更された 1 つ以上の入力データ値と、前記プロセスフローのステップに関連付けられた状態情報を反映する工程と；

前記プロセスアプリケーションを再度コンパイルする工程と；

デバイス識別子を使用して前記デバイスに前記プロセスアプリケーションを再度デプロイする工程と；

、を有する方法。

【請求項 18】

請求項 17 記載の方法は、前記プロセスフローを記述するフロー定義および前記プロセスフローの各ステップにおける前記プロセスフローの状態を記述する状態情報を受け取る工程をさらに含むものである。

40

【請求項 19】

請求項 18 記載の方法において、前記フロー定義および前記状態情報に基づいてプロセスアプリケーションを再構成する工程をさらに有し、前記プロセスアプリケーションは前記プロセスフローを含むものである。

【請求項 20】

請求項 19 記載の方法は、前記プロセスアプリケーションを実行可能ファイルにコンパイルする工程をさらに有するものである。

50

【請求項 2 1】

請求項 2 0 記載の方法は、前記コンパイルされた実行可能ファイルを前記状態情報を生成したデバイスにデプロイする工程をさらに有するものである。

【請求項 2 2】

複数のデバイスにおいてインスタンス化されたプロセルアプリケーションを診断および更新するためのコンピュータ指令を有するコンピュータプログラムが格納されたコンピュータ可読記憶媒体であって、前記コンピュータ指令は、

状態遷移情報をデバイスにローカルに格納する工程であり、この状態遷移情報は、インスタンス化されたプロセスエンジンによってインスタンス化されたプロセスアプリケーションの実行されたステップに関連付けられたものであり、前記インスタンス化されたプロセスアプリケーションは、誤った結果を生成したものである、前記格納する工程と；

フロー定義を遠隔診断プラットフォームに送信する工程と；

前記遠隔診断プラットフォームにおいて、前記フロー定義と前記状態遷移情報を受信する工程であり、前記フロー定義は、プロセスフローを記述し、前記状態遷移情報は、前記インスタンス化されたプロセスアプリケーションに関連付けられた前記プロセスフローのステップにおいて変更されたデータ値の情報を含むものである、前記受信する工程と；

前記状態遷移情報を使って、前記デバイスの前記インスタンス化された前記プロセスエンジンに関連付けられたプロセスエンジンを再生成する工程と；

前記プロセスフローの選択されたステップに関連付けられた 1 つ以上の入力データ値を変更する工程と；

前記変更された 1 つ以上の入力データ値を使用して、前記選択されたステップから前記プロセスフローを実行する工程と；

プロセスアプリケーションの前記選択されたステップに関連付けられた少なくとも 1 つのデータレコードを更新して、前記変更された 1 つ以上の入力データ値と、前記プロセスフローのステップに関連付けられた状態情報を反映する工程と；

前記プロセスアプリケーションを再度コンパイルする工程と；

デバイス識別子を使用して前記デバイスに前記プロセスアプリケーションを再度デプロイする工程と；

を含むものである、コンピュータ可読記憶媒体。

【請求項 2 3】

請求項 2 2 記載のコンピュータ可読記憶媒体は、前記プロセスフローを記述するフロー定義および前記プロセスフローの各ステップにおける前記プロセスフローの状態を記述する状態情報を受け取る工程をさらに有するものである。

【発明の詳細な説明】**【技術分野】****【0 0 0 1】****(関連出願)**

本出願は、すべての目的のために参照により本明細書に組み込まれる、2016年5月16日付出願の米国仮特許出願第 62 / 337,156 号、発明の名称「STEPBACK MECHANISM TO DEVELOP AND DIAGNOSE PROCESS APPLICATIONS」に対する優先権を主張する。

【背景技術】**【0 0 0 2】**

プロセスアプリケーションを実行するプロセスエンジンを、デバイスにデプロイすることができる。プロセスアプリケーションは特定の方法で動作するように設計されているが、何らかの未知の理由によりプロセスアプリケーションが期待どおりに動作しないことがある。可能な解決策の 1 つは、単に装置を交換することであるが、このような解決策は実用的でないか、またはコスト高になり、またプロセスアプリケーションに関する根本的な問題を解決しない。

この出願の発明に関連する先行技術文献情報としては、以下のものがある（国際出願日以

10

20

30

40

50

降国際段階で引用された文献及び他国に国内移行した際に引用された文献を含む)。

(先行技術文献)

(特許文献)

(特許文献 1) 米国特許出願公開第 2005 / 0251794 号明細書

(特許文献 2) 米国特許出願公開第 2013 / 0335980 号明細書

(特許文献 3) 米国特許第 5,675,803 号明細書

(特許文献 4) 米国特許第 6,202,199 号明細書

(特許文献 5) 米国特許第 7,685,253 号明細書

(特許文献 6) 米国特許出願公開第 2008 / 0281944 号明細書

(特許文献 7) 米国特許出願公開第 2017 / 0169114 号明細書

(特許文献 8) 米国特許出願公開第 2016 / 0112262 号明細書

(特許文献 9) 米国特許出願公開第 2004 / 0078105 号明細書

(特許文献 10) 米国特許第 9,569,255 号明細書

(特許文献 11) 米国特許出願公開第 2006 / 0074736 号明細書

(特許文献 12) 米国特許出願公開第 2001 / 0032108 号明細書

(特許文献 13) 米国特許出願公開第 2016 / 0124742 号明細書

(特許文献 14) 米国特許出願公開第 2012 / 0271926 号明細書

(特許文献 15) 米国特許出願公開第 2012 / 0079013 号明細書

(特許文献 16) 米国特許出願公開第 2009 / 0164606 号明細書

(特許文献 17) 米国特許出願公開第 2018 / 0069774 号明細書

(特許文献 18) 米国特許出願公開第 2017 / 0090995 号明細書

(特許文献 19) 米国特許出願公開第 2017 / 0255886 号明細書

(特許文献 20) 米国特許出願公開第 2014 / 0282367 号明細書

【図面の簡単な説明】

【0003】

本発明の様々な実施形態が、以下の詳細な説明および添付図面に開示される。

【図 1】図 1 は、ステップバック機構を使用してプロセスアプリケーションを開発および診断するためのシステムの実施形態を示すブロック図である。

【図 2 A】図 2 A は、プロセスアプリケーション開発プラットフォームの一実施形態を示すブロック図である。

【図 2 B】図 2 B は、プロセスアプリケーション診断プラットフォームの一実施形態を示すブロック図である。

【図 3】図 3 は、プロセスアプリケーションの一実施形態を示すフローチャートである。

【図 4】図 4 は、デバイス上で動作するプロセスエンジンによって実行されるプロセスアプリケーションに関連付けられた状態情報を報告するためのプロセスの実施形態を示すフローチャートである。

【図 5】図 5 は、プロセスアプリケーションを診断するためのプロセスの一実施形態を示すフローチャートである。

【図 6】図 6 は、プロセスアプリケーションを開発するためのプロセスの一実施形態を示すフローチャートである。

【図 7】図 7 は、プロセスアプリケーションを修正するためのプロセスの一実施形態を示すフロー

【図 8】図 8 は、プロセスを配備するためのプロセスの一実施形態を示すフローチャートである。

【発明を実施するための形態】

【0004】

本発明は、さまざまな形式で実施可能であるが、プロセス、装置、システム、組成物、コンピュータ可読記憶媒体上に具現化されたコンピュータプログラム製品、および／またはプロセッサ（プロセッサーに結合されたメモリに記憶された命令を実行するように構成されたプロセッサ等）が含まれる。本明細書では、これらの実装形態、または本発明がとり

10

20

30

40

50

得る他の形態は、技術と呼ぶことができる。一般に、開示されたプロセスのステップの順序は、本発明の範囲内で変更されてもよい。別段の記載がない限り、プロセッサまたはメモリなどのタスクを実行するように構成されたものとして記載されているコンポーネントは、所与の時間にタスクを実行するように一時的に構成された一般的なコンポーネント、またはそのタスクを行うために製造された特定のコンポーネントによって実行される。本明細書で使用する「プロセッサ」という用語は、コンピュータプログラム命令などのデータを処理するように構成された1つ以上のデバイス、回路、および/または処理コアを指す。

【0005】

本発明の1つ以上の実施形態の詳細を、以下に本発明の原理を説明する添付の図面と共に説明する。本発明はそのような実施形態に関連して説明されるが、本発明はいずれの実施形態にも限定されるものではない。本発明の範囲は、特許請求の範囲によってのみ限定され、本発明は、多数の代替、変形および等価物を含むものである。本発明の完全な理解を提供するために、多くの具体的な詳細が以下の説明に記載されている。これらの詳細は、例示のために提供されており、本発明は、これらの特定の詳細の一部または全部を伴わずに、請求項に従って実施することができる。明瞭化のために、本発明に関連する技術分野において周知の技術内容は、本発明が不必要に不明瞭にならないように詳細には記載していない。

10

【0006】

プロセスアプリケーションを開発し診断するためのステップバック機構が開示されている。いくつかの実施形態では、デバイス上で動作するプロセスエンジンの例では、プロセスアプリケーションを実行するように構成されているが、予期しない誤った挙動が起こる。プロセスアプリケーションは、1つ以上のプロセスフローを含む。各プロセスフローは一連のステップで構成されるが、たとえば、1つ以上のアクティビティの後に、メッセージを記録するアクティビティなどである。プロセスフローのステップの実行中に変更されたデータ値、すなわち状態遷移が記録され、ローカルに格納される。その後、変更されたデータ値が送信され、遠隔状態情報データベースに格納される。プロセスアプリケーションの定義およびプロセスエンジンのインスタンスに関連する記録された状態値は、プロセスアプリケーション診断プラットフォームによって得られる。プロセスアプリケーション診断プラットフォームは、ステップバック機構を含み、関連付けられた状態値で前記デバイス上で実行されたプロセスエンジンを再生成するように構成される。ステップバック機構は、オペレータがプロセスアプリケーションに関連する1つ以上のプロセスフローのステップをバックトラックすることを可能にする。オペレータは、プロセスフローの各ステップで記録された状態を検査するだけでなく、任意のステップで関連付けられた任意のデータ値を調整し、プロセスアプリケーションを始めから実行することなくプロセスフローの後続ステップへの調整データ値の影響を判定できます。

20

30

【0007】

オペレータは、所望の結果を達成するためにプロセスフローに対する1つ以上の調整を行うことができる。オペレータがプロセスフローに所望の結果を達成させるようにプロセスフローを調整した場合、プロセスアプリケーションを更新して調整を反映させることができる。その後、プロセスアプリケーションを再度コンパイルしてデバイスに再デプロイすることができます。これにより、装置を完全に交換することは高費用、または実用的でない場合があるが、完全に交換しなくても誤動作した装置を修理することができる。

40

【0008】

他の実施形態では、プロセスアプリケーションを実行するプロセスエンジンは、プロセスアプリケーション開発プラットフォームを使用して開発することができる。プロセスアプリケーション開発プラットフォームは、開発者がプロセスアプリケーションを作成できるように構成され、開発者がプロセスフローの任意のステップに対する1つ以上の調整を行い、プロセスアプリケーションを最初から再実行しなくとも、1つ以上の調整の後続のステップへの影響を決定することを可能にするステップバックメカニズムを含む。プロセス

50

フローが所望の結果を達成するように開発者がプロセスフローを調整した場合、プロセスアプリケーションを保存して調整を反映させることができる。プロセスアプリケーションをプリコンパイルして、プロセスアプリケーションを実行するために必要な最小限のコンポーネントセットを決定し、不要なコンポーネントを削除することが可能である。その後、プロセスアプリケーションを実行可能ファイルにコンパイルし、デバイスにデプロイすることができる。

【0009】

図1は、ステップバック機構を使用してプロセスアプリケーションを開発および診断するためのシステムの実施形態を示すプロック図である。図示の例では、システム100は、プロセスアプリケーション開発プラットフォーム110、プロセスアプリケーション診断プラットフォーム112、フロー定義データベース114、状態情報データベース116、デバイス118、およびネットワーク120を含む。10

【0010】

プロセスアプリケーション開発プラットフォーム110は、インテグレータ104がプロセスアプリケーションを呼び出すカスタムプロセスエンジンを作成することを可能にするプラットフォームである。インテグレータ104は、特定の目的のための1つ以上のプロセスフローからなるプロセスアプリケーションをプロセスアプリケーション開発プラットフォーム110を介して作成する、ソフトウェア開発者などのエンティティであってもよい。

【0011】

例えば、インテグレータは、装置がシステム更新を実行する、特定の機能を実行する、診断テストを実行する等ために、プロセスフローを作成することができる。各プロセスフローは、メッセージを記録するアクティビティの後に1つ以上のアクティビティが続くなど、一連のステップからなる。

【0012】

インテグレータは、プロセスフローを開始するトリガ、またトリガ後に呼び出される1つ以上のアクティビティを選択することができる。インテグレータは、第1のプロセスフローのアクティビティから分岐する第2のプロセスフロー、またはトリガーの出力に基づく第2のプロセスフローを作成することができる第2のプロセスフローを作成できる。トリガーおよびアクティビティは、ライブラリ、コンポーネント、および／または関数への参照を含むコードの個々のセットによって定義されてもよい。30

【0013】

開発者102は、プロセスアプリケーション開発プラットフォーム110を介してアクティビティおよびトリガを作成する個人または他のエンティティ（例えば、ソフトウェア開発者）であってもよい。開発者102は、アクティビティおよびトリガクリエータを使用してアクティビティおよび／またはトリガを作成することができる。開発者によって作成された各アクティビティおよびトリガは、Go、Python、C、C++などのプログラミング言語を介して作成される自己完結型のコード機能であってもよい。トリガおよびアクティビティは、複数の開発者によって作成されてもよく、プロセスフローを形成するためにトリガーおよびアクティビティがどのようにして後で他の人によって組み合わされるかを開発者のそれぞれが認識しない。40

【0014】

オペレータ106は、プロセスアプリケーションを診断する人または他のエンティティ（例えば、ソフトウェア開発者）であってもよい。オペレータ106は、フロー定義データベース114から検索するファイルと、状態情報データベース116から検索するプロセスエンジンに関連付けられた関連する記録済み状態遷移とを選択することによって、プロセスエンジンが故障したインスタンスを復活させることができる。いくつかの実施形態において、前記ファイルは、flow.jsonファイル、XMLファイル、またはフロー定義を格納するその他のファイルが可能である。プロセスエンジンに関連付けられた関連する記録済み状態遷移は、プロセスエンジンが故障したインスタンスを実行したデバイス

10

20

30

40

50

に固有の識別子を介して識別することができる。オペレータ 106 は、任意のステップでプロセスアプリケーションに関連する状態値をユーザインターフェースを介して修正し、プロセスアプリケーションの開始からプロセスアプリケーションを実行する必要なく、修正された状態が後続のステップに及ぼす影響を見ることができる。オペレータ 106 は、元の状態値の代わりに修正された状態値を出力するようにステップが構成されるように、プロセスアプリケーションを更新することができる。

【0015】

プロセスアプリケーション診断プラットフォーム 112 は、オペレータ 106 がデバイスに関連付けられたフロー定義ファイルをフロー定義データベース 114 から、また前記デバイスに関連付けられた関連する記録済み状態遷移を状態情報データベース 116 から、検索することを可能にするプラットフォームである。特定のフロー定義は 1 つ以上のデバイスに関連付ける方法で格納可能である。フロー定義ファイルは、プロセスアプリケーションに関連する期待される動作のメタデータ記述である。いくつかの実施形態では、フロー定義は、プロセスアプリケーションのトリガとアクティビティとの関係、およびトリガおよびアクティビティの実行方法を記述する。いくつかの実施形態では、フロー定義は、プロセスアプリケーションのアクティビティとプロセスアプリケーションの動作との間の関係を記述する。いくつかの実施形態では、フロー定義ファイルは、`flow.json` ファイル、`app.json` ファイル、XML ファイル、またはフロー定義を格納する他のファイルであってもよい。関連する記録済み状態遷移は、プロセスアプリケーションの各ステップにおいて、プロセスアプリケーションのトリガおよびアクティビティに関連付けられた値を記述する。プロセスエンジンに関連付けられた関連する記録済み状態遷移は、プロセスエンジンが故障したインスタンスを実行したデバイスに固有の固有の識別子を介して識別することができる。プロセスアプリケーション診断プラットフォーム 112 は、フロー定義ファイルと関連する記録済み状態遷移との組み合わせを使用して、デバイス 118 などのデバイスにデプロイされたプロセスエンジンを再作成するように構成される。プロセスアプリケーション診断プラットフォーム 112 は、プロセスアプリケーションの任意のステップからプレセスアプリケーションを再生成するように構成されている。プロセスアプリケーション診断プラットフォーム 112 は、プロセスアプリケーションの任意のステップに関連付けられたデータ値を調整する指示を受け取り、前記データ値を調整するように構成される。プロセスアプリケーション診断プラットフォーム 112 は、選択されたステップから、ステップの検索されたデータ値の代わりに調整されたデータ値を有するプロセスアプリケーションを再生するように構成される。いくつかの実施形態では、プロセスアプリケーション診断プラットフォーム 112 は、プロセスアプリケーションを故障前のステップに再生するように構成される。

【0016】

いくつかの実施形態では、プロセスアプリケーション開発プラットフォーム 110 およびプロセスアプリケーション診断プラットフォーム 112 を、開発者 102、インテグレータ 104、および / またはオペレータ 106 がアクセス可能な単一のプロセスアプリケーション開発および診断プラットフォームに結合することができる。

【0017】

フロー定義データベース 114 は、プロセスアプリケーションの動作を記述するファイルを格納するデータベースである。いくつかの実施形態では、フロー定義データベース 114 は、プロセスアプリケーションのトリガ、プロセスアプリケーションのアクティビティ、プロセスアプリケーション内のトリガとアクティビティとの間の関係、および / またはプロセス内のアクティビティ間の関係を記述するフロー定義ファイルを格納する。いくつかの実施形態では、ネットワーク 120 を介してアクセス可能なフロー定義データベース 114 のインスタンスが多数存在する。いくつかの実施形態では、フロー定義ファイルは、`flow.json` ファイル、`app.json` ファイル、XML ファイル、またはその他のフロー定義を格納するファイルが可能である。

【0018】

10

20

30

40

50

状態情報データベース 116 は、デバイス 118 などのデバイス上で実行されているプロセスエンジンに関連付けられた状態を格納するデータベースである。いくつかの実施形態では、各ステップにおけるプロセスアプリケーションのすべての状態値が状態情報データベース 116 に格納される。いくつかの実施形態では、プロセスアプリケーションのステップ中に変更される状態値が状態サービスデータベース 116 に格納される。いくつかの実施形態では、各プロセスアプリケーションステップの完了時に状態値が格納される。いくつかの実施形態では、状態値は、各プロセスアプリケーションのステップの開始時に格納される。

【0019】

デバイス 118 は、プロセスエンジンを実行するターゲットデバイスである。例えば、デバイス 118 は、モバイルデバイス、相互接続デバイス、IoT デバイス、汎用コンピュータ、またはプロセスエンジンを実行する任意のデバイスであってもよい。プロセスアプリケーションがコンパイルされた後、実行可能なプロセスエンジンをデバイス 118 にデプロイすることができる。プロセスエンジンは、一般的なプロセスエンジンまたはカスタムプロセスエンジンとすることができます。プロセスエンジンは、デバイス 118 によって実行されると、デバイスの挙動を制御する。いくつかの実施形態では、デバイス 118 は、プロセスアプリケーションの各ステップが完了した後のすべての状態値を記憶する。いくつかの実施形態では、デバイス 118 は、プロセスアプリケーションの各ステップが完了した後の状態遷移を格納する。いくつかの実施形態では、デバイス 118 は、プロセスアプリケーションの各ステップが完了した後に、状態情報データベース 116 に状態遷移を送信する。他の実施形態では、デバイス 118 は、状態情報データベース 116 に状態遷移を周期的に送信する。他の実施形態では、デバイス 118 は、要求に応じて状態情報データベース 116 に状態遷移を送信する。いくつかの実施形態では、デバイス 118 がネットワーク 120 への接続性を失ったとき、デバイス 118 は、ネットワーク 120 への接続を再確立するときに状態情報データベース 116 に状態遷移を送信する。

10

20

【0020】

ネットワーク 120 は、プロセスアプリケーション開発プラットフォーム、プロセスアプリケーション診断プラットフォーム 112、フローサービスデータベース 114、状態サービスデータベース 116、およびデバイス 118 と相互接続する。ネットワーク 120 は、ローカルエリアネットワーク、有線ネットワーク、無線ネットワーク、インターネット、インターネット、または任意の他の適切な通信ネットワークのうちの 1 つ以上を備える。

30

【0021】

図 2A は、プロセスアプリケーション開発プラットフォームの一実施形態を示すブロック図である。いくつかの実施形態では、プロセスアプリケーション開発プラットフォーム 110 を使用して、プロセスアプリケーション開発プラットフォーム 210 を実装することができる。

【0022】

プロセスアプリケーション開発プラットフォーム 210 は、プロセスアプリケーションワークスペース 205、プリコンパイラ 206、アクティビティおよびトリガクリエータ 207、コンパイラ 208、および / またはデータベース 209 を含むことができる。プロセスアプリケーション開発プラットフォーム 210 は、1 つ以上のコンピュータデバイスを使用して実施可能であり、コンピュータ、マルチプロセッサシステム、マイクロプロセッサベースのシステム、クラウドベースのシステム、専用装置、上述のシステムまたは装置のいずれかを含む分散コンピューティング環境、または 1 つ以上のプロセッサー、前記プロセッサーに接続された、プロセッサーに指令を提供するように構成されたメモリーを含むハードウェア / ソフトウェア / ファームウェアの他の適切な組み合わせ等を含む。

40

【0023】

プロセスアプリケーションワークスペース 205 は、インテグレータ 104 などのインテグレータがプロセスアプリケーションに含めるべき 1 つ以上のプロセスフローを作成する

50

ことを可能にするユーザインタフェースである。いくつかの実施形態では、1つ以上のプロセスフローは、ビジュアルエディタまたはテキストエディタによって作成することができる。プロセスフローには、1つ以上のアクティビティが続くトリガが含まれる。他の実施形態では、プロセスフローは、1つ以上のアクティビティから構成される。このようなプロセスフローは、プロセスフロー定義の一部ではないトリガに関連付けることができる。プロセスフローが実行されるとき、プロセスフローは1つ以上のステップから構成される。プロセスフローの各ステップでは、各トリガーと1つ以上のアクティビティには関連した状態がある。プロセスアプリケーションワークスペース205は、インテグレータがプロセスフローの任意のステップで任意のトリガおよび／または任意のアクティビティに関連付けられた1つ以上のデータ値を修正し、修正されたデータに基づいて残りのステップを実行することを可能にするステップバック機構を含む。プロセスアプリケーションワークスペース205は、ユーザがトリガおよび／またはアクティビティに関連付けられたデータレコードを修正および格納することを可能にするように構成される。データレコードは、トリガーおよび／またはアクティビティによって使用されるデータ値を含む。

【0024】

プリコンパイラ206は、インテグレータによって作成されたプロセスアプリケーションの1つ以上のプロセスフローを分析し、プロセスアプリケーションを実行するために必要な最小限のコンポーネントセットを決定するように構成されている。コンパイラ208は、プロセスアプリケーションを実行可能ファイルにコンパイルするように構成される。得られたコンパイルされたコードはパッケージ化され、実行可能なバイナリとして（例えば、ダウンロードによって）デプロイされるように利用可能にされる。実行可能ファイルは、デバイス118などのデバイス上で実行されると、デバイス118がどのように動作するかを制御するプロセスエンジンである。データベース209は、開発者102などの1人または複数の開発者によって作成されたアクティビティおよびトリガを格納するように構成される。データベース209は、アプリケーション開発プラットフォーム210の内部のメモリおよび／または記憶装置、アプリケーション開発プラットフォーム210に付属したメモリおよび／または記憶装置、プロセスアプリケーション開発プラットフォーム210にネットワーク接続を介して接続された任意のメモリおよび／またはストレージデバイス、またはそれらの任意の組み合わせが可能である。ネットワーク接続は、有線または無線接続であってもよい。

【0025】

図2Bは、プロセスアプリケーション診断プラットフォームの一実施形態を示すブロック図である。いくつかの実施形態では、処理アプリケーション診断プラットフォーム212は、処理アプリケーション診断プラットフォーム112を使用して実装することができる。

【0026】

プロセスアプリケーション診断プラットフォーム212は、プロセスアプリケーション診断ワークスペース215、プリコンパイラ216、コンパイラ218、および／またはデータベース219を含むことができる。プロセスアプリケーション診断プラットフォーム210は、1つ以上のコンピューティングデバイスを使って実装可能であり、これにはコンピュータ、マルチプロセッサシステム、マイクロプロセッサベースのシステム、クラウドベースのシステム、専用装置、上記のシステムまたは装置のいずれかを含む分散コンピューティング環境、または他の適切なハードウェア／ソフトウェア／ファームウェアの組み合わせであって1つ以上のプロセッサと、プロセッサに結合され、プロセッサに命令を提供するように構成されたメモリとを含むもの等が含まれる。

【0027】

プロセスアプリケーション診断ワークスペース215は、オペレータ106などのオペレータが、フロー定義データベースからデバイスにデプロイされたプロセスアプリケーションに関連付けられた1つ以上のプロセスフローを検索し、それに関連する1つ以上の状態値を状態情報データベースから検索することを可能にするユーザインタフェースである。前記検索されたプロセスフローは、デバイス上で実行されたものである。プロセスアプリ

10

20

30

40

50

ケーション診断ワークスペース 215 は、ステップバック機構を含み、プロセスフローの特定のステップでプロセスフローに関連付けられた 1 つ以上のデータ値を変更する 1 つ以上の入力を受け取るように構成される。プロセスアプリケーション診断ワークスペース 215 は、前記修正された値で特定のステップからのプロセスフローを実行するようにさらに構成される。

【 0028 】

プロセスアプリケーション診断ワークスペース 215 は、格納された状態値の代わりに修正された状態値を出力するようにプロセスアプリケーションを更新するようにさらに構成される。プロセスアプリケーション診断ワークスペース 215 はさらに、更新されたプロセスアプリケーションをデータベース 219 またはリモートフロー定義データベースに格納するように構成される。更新されたプロセスアプリケーションは、所望の出力を生成するためにトリガおよび / またはアクティビティによって使用される 1 つ以上の変更されたデータ値を含むデータレコードを含む。

【 0029 】

プリコンパイラ 216 は、プロセスアプリケーションの 1 つ以上のプロセスフローを分析し、プロセスアプリケーションを実行するために必要なコンポーネントの最小セットを決定するように構成されている。コンパイラ 218 は、プロセスアプリケーションを実行可能ファイルにコンパイルするように構成される。得られたコンパイルされたコードはパッケージ化され、実行可能なバイナリとして（例えば、ダウンロードなどによって）再配置されるように利用可能にされる。実行可能ファイルは、デバイス 118 のようなデバイス上で実行されると、デバイスがどのように動作するかを制御するプロセスエンジンである。データベース 219 は、オペレータ 106 などの 1 つ以上のオペレータによって変更されたアクティビティおよびトリガを格納する。データベース 219 は、プロセスアプリケーション開発プラットフォーム 212 内のメモリおよび / またはストレージデバイス、プロセスアプリケーション開発プラットフォームに取り付けられたメモリおよび / またはデバイス、またはネットワーク接続を介してプロセスアプリケーション開発プラットフォーム 212 に接続された任意のメモリおよび / または記憶装置、または上記の任意の組み合わせを含む。ネットワーク接続は、有線または無線接続であってもよい。いくつかの実施形態では、変更されたプロセスアプリケーション定義は、フロー定義データベース 114 などのリモートデータベースに送信される。

【 0030 】

図 3 は、プロセスアプリケーションの一実施形態を示すフローチャートである。いくつかの実施形態では、プロセスアプリケーション開発ワークスペース 205 などのプロセスアプリケーション開発ワークスペースを使用してプロセスアプリケーション 300 を開発することができる。他の実施形態では、プロセスアプリケーション診断ワークスペース 215 などのプロセスアプリケーション診断ワークスペースを使用してプロセスアプリケーション 300 を診断することができる。

【 0031 】

プロセスアプリケーションは、1 つ以上のプロセスフローから構成される。いくつかの実施形態では、プロセスフローは、トリガおよび 1 つ以上のアクティビティから構成される。いくつかの実施形態では、1 つ以上のアクティビティから構成され、プロセスフロー定義の一部ではないトリガに関連付けられる。アクティビティの非限定的な例には、カウンタアクティビティ、ログアクティビティ、リセットアクティビティ、制約付きアプリケーションプロトコル（C o A P）アクティビティ、およびハイパーテキスト転送プロトコル（H T T P）アクティビティが含まれる。

プロセスアプリケーションを呼び出すデプロイされたプロセスエンジンは、誤った結果を生成する可能性がある。例えば、プロセスフローのアクティビティは、第三者のソフトウェアからの出力を入力として使用することができる。第三者ソフトウェアは、第三者ソフトウェアが異なる値、すなわちアクティビティに対する異なる入力値を出力するよう更新される可能性がある。その結果、アクティビティは、アクティビティが出力すると予想

10

20

30

40

50

される状態値とは異なる状態値を出力する。

【0032】

図示の例では、プロセスアプリケーション300は、プロセスフローAおよびプロセスフローBを含む。プロセスフローAは、トリガ302で始まり、アクティビティ304、アクティビティ306、およびアクティビティ308が後続する一連のステップからなる。プロセスフローBは、トリガ310で始まり、アクティビティ312、アクティビティ314、およびアクティビティ316が後続する一連のステップから構成される。プロセスフローは一般にトリガで始まり、先行するトリガーとアクティビティの出力を入力として使用する1つ以上のアクティビティが後続する。

【0033】

プロセスフローAは、逐次プロセスフローの一例である。プロセスフローAは、プロセスアプリケーション開発プラットフォームで開発することができる。プロセスフローAは、デバイスにデプロイされたプロセスエンジンに関連付けられた検索されたフロー定義と、デバイス上で実行されるプロセスエンジンのインスタンスに関連付けられた検索された状態情報に基づいて、プロセスアプリケーション診断ワークスペースによって再生成することもできる。プロセスフローの第1ステップにおいて、トリガ302は、アクティビティ304に入力される状態値を出力することができる。プロセスフローの第2ステップにおいて、アクティビティ304は、アクティビティ306に入力される状態値を出力することができる。プロセスフローの第3ステップにおいて、アクティビティ306は、アクティビティ304の出力に基づいて状態値を出力することができる。プロセスフローの第4のステップにおいて、アクティビティ308は、アクティビティ306の出力に基づいて状態値を出力することができる。各アクティビティは、先行するアクティビティまたはトリガの出力に加えて、1つ以上の入力データ値に基づいた状態値を出力することが可能である。例えば、アクティビティ304によって出力される状態値は、トリガ302の出力および1つ以上の追加入力データ値に基づくことができる。1つ以上の追加の入力データ値は、アクティビティ304によって出力された状態値を調整するように調整することができる。

【0034】

アクティビティに関連付けられた1つ以上の追加入力データ値は、プロセスアプリケーション開発ワークスペースで調整することができる。例えば、アクティビティ306が特定の状態を出力することが望ましいかもしれないが、何らかの理由でアクティビティ306が所望の状態を出力しない。いくつかの実施形態では、アクティビティ304は、アクティビティ304に関連する状態値が、アクティビティ306に所望の状態とは異なる値を出力させるため、所望の状態を出力しないことがある。プロセスアプリケーション開発ワークスペースは、アクティビティ304に関連付けられた1つ以上の入力データ値を調整する入力を受け取り、1つ以上の調整された入力データ値でプロセスフローを第2ステップからシミュレートするように構成される。例えば、ステップ2において、アクティビティ304は、「X」の状態を出力するように構成される。ステップ3において、アクティビティ306は「X」の状態を受け取り、「A」の状態を出力する。ステップ4において、アクティビティ308は「A」の状態を受け取り、「B」の状態を出力する。アクティビティ304に関連付けられた1つ以上の入力データ値は、アクティビティ304が「Y」の状態を出力するように構成されるようにアクティビティ304を調整することができる。ステップ3では、アクティビティ306は「Y」の状態を受け取り、「C」の状態を出力する。ステップ4では、アクティビティ308が「C」の状態を受け取り、「D」の状態を出力する。プロセスアプリケーション開発ワークスペースは、インテグレータが、ステップ1からプロセスフローを開始することなく、アクティビティ304の状態を調整した場合の影響を判定することを可能にする。これにより、選択したステップの前にあるステップを実行するためにシステム資源を使用する必要がなくなるので、プロセスアプリケーションが故障したインスタンスを診断するために必要な時間が短縮される。いくつかの例では、プロセスアプリケーションは数百のステップから構成されるので、最初からプ

10

20

30

40

50

ロセスアプリケーションを実行すると、選択されたステップ、例えばプロセスアプリケーションの途中、あるいは最後の方からプロセスアプリケーションを実行した時よりも多くの計算資源を必要とする。プロセスアプリケーション開発プラットフォームはまた、プロセスフローの変更されたバージョンを保存するように構成されている。

【0035】

プロセスフローBは、複数の経路を有するプロセスフローの一例である。プロセスフローBは、プロセスアプリケーション開発プラットフォームで開発することができる。プロセスフローBは、プロセスアプリケーション診断ワークスペースによって、デバイスにデプロイされたプロセスエンジンに関連付けられた検索されたフロー定義と、前記デバイスで実行されたプロセスエンジンのインスタンスに関連付けられた、検索された状態情報に基づいて再生成可能であり、オペレータがトリガまたはアクティビティに関連付けられた1つ以上の入力変数を調整して、関連するトリガまたはアクティビティが所望の状態を生成するようにさせることを可能にする。

10

【0036】

図示の例では、ステップ2において、アクティビティ312は、ステップ3においてアクティビティ314に状態値を出力させる状態を出力するが、ステップ3においてアクティビティ316には状態値を出力させない状態を出力することができる。アクティビティ312は、元々、アクティビティ316に状態値を出力させる状態値を出力するようにプログラムされたものであってもよい。アクティビティ312に関連付けられた1つ以上の入力データ値は、アクティビティ316に状態値を出力させるように調整することができる。プロセスアプリケーション診断ワークスペースは、オペレータがステップ1からのプロセスフローを開始する必要なく、アクティビティ状態312の調整の影響を判定することを可能にする。これにより、選択したステップの前にあるステップを実行するためにシステム資源を使用する必要がないため、プロセスアプリケーションが故障したインスタンスを診断するために必要な時間が短縮される。示されている例では、変更された入力でステップ2からプロセスフローを開始することができる。

20

【0037】

プロセスフローCは、トリガのないプロセスフローの一例である。プロセスフローCは、アクティビティ320, 322、および324から構成される。プロセスフローCは、トリガ(図示せず)に関連付けることができるが、トリガはプロセスフローCの定義の一部ではない。プロセスフローCは、プロセスフローAに類似するが、トリガはプロセスフローCの定義の一部ではない。

30

【0038】

図4は、デバイス上で動作するプロセスエンジンによって実行されるプロセスアプリケーションに関連する状態情報を報告するためのプロセスの実施形態を示すフローチャートである。いくつかの実施形態では、図4のプロセス400は、デバイス118のような装置によって実施することができる。

【0039】

402において、実行可能バイナリが受信される。実行可能バイナリは、デバイス上で実行されると、デバイスの動作を制御するプロセスエンジンである。プロセスエンジンは、プロセスアプリケーションで構成される。

40

【0040】

404において、プロセスアプリケーションのインスタンスがデバイス上で実行される。プロセスアプリケーションの各ステップの後、各ステップに関連付けられた状態情報がデバイスに記憶される。

【0041】

406において、状態情報がリモートサービスデータベースに報告される。状態情報は、デバイスに関連付けられた識別子と、関連する状態情報が変更されたときのタイムスタンプとを含む。いくつかの実施形態では、デバイスは、プロセスアプリケーションの各ステップが完了した後の状態遷移を格納する。いくつかの実施形態では、デバイスは、プロセ

50

スアプリケーションの各ステップが完了した後に、状態情報データベースに状態遷移を送信する。他の実施形態では、デバイスは、状態遷移を状態情報データベースに周期的に送信する。他の実施形態では、デバイスは、要求に応じて状態情報データベースに状態遷移を送信する。いくつかの実施形態では、デバイスがネットワークへの接続を失った際、デバイスは、ネットワークへの接続を再確立すると、状態情報データベースに状態遷移を送信する。

【 0 0 4 2 】

図5は、プロセスアプリケーションを診断するためのプロセスの一実施形態を示すフローチャートである。いくつかの実施形態では、図5のプロセス500は、図5のプロセスは、プロセスアプリケーション診断プラットフォーム112などのプロセスアプリケーション診断プラットフォームによって実行することができる。

10

【 0 0 4 3 】

示された例では、プロセス500は、デバイスにデプロイされたプロセスエンジンに関連するフロー定義と、デバイスにデプロイされたプロセスエンジンのインスタンスに関連する状態情報とを受け取る502で開始する。いくつかの実施形態では、フロー定義はリモートデータベースから受信され、デバイスにデプロイされたプロセスエンジンのインスタンスに関連する状態情報は、別個のリモートデータベースから受信される。フロー定義は、プロセスエンジンによって実行されるプロセスアプリケーションの動作を記述し、プロセスアプリケーションに関連付けられた1つ以上のプロセスフローを含む。状態情報は、各ステップでプロセスエンジンによって実行されるプロセスアプリケーションの状態を記述し、各ステップにおけるプロセスフローのトリガおよびアクティビティに関連する状態値を含む。

20

【 0 0 4 4 】

504において、デバイスにデプロイされたプロセスエンジンによって実行されるプロセスアプリケーションのインスタンスが再作成される。再作成されると、プロセスアプリケーションはどのステップからでも実行でき、関連する状態値を決定できる。

506において、所望の出力を達成するようにプロセスアプリケーションが変更される。例えば、プロセスアプリケーションの任意のステップを選択することができ、選択されたステップに関連付けられた1つ以上の入力データ値を変更して、選択されたステップの出力状態を調整することができる。選択されたステップ調整が後続のステップの状態に及ぼす影響を決定することができる。いくつかの実施形態では、アクティビティまたはトリガに関連する定義を修正して、所望の出力結果を生成することができる。

30

【 0 0 4 5 】

508において、プロセスアプリケーションを更新して、プロセスアプリケーションに所望の結果を達成させる変更を反映させ、さらにメモリに保存する。更新されたプロセスアプリケーションは、所望の出力を生成するためにトリガおよび／またはアクティビティによって使用される1つ以上の変更されたデータ値を含むデータレコードを含む。

【 0 0 4 6 】

図6は、プロセスアプリケーションを開発するためのプロセスの一実施形態を示すフローチャートである。いくつかの実施形態では、図6のプロセス600は、図6に示すプロセスアプリケーション開発プラットフォーム110は、プロセスアプリケーション開発プラットフォーム110のようなプロセスアプリケーション開発プラットフォームによって実行することができる。

40

【 0 0 4 7 】

図示の例では、プロセス600は、アクティビティおよびトリガのセットが受信されて、プロセスアプリケーションの1つ以上のプロセスフローが作成される、602で開始する。インテグレータには、例えば、F1ologo（登録商標）インターフェースなどのユーザインターフェースを介して、アクティビティおよびトリガを選択する機能が提供される。いくつかの実施形態では、インテグレータは、提供される利用可能なアクティビティおよびトリガのリストから1つ以上のアクティビティおよびトリガを選択する。他の実施形態では

50

、インテグレータは、利用可能なアクティビティおよびトリガをローカルデータベースから検索し、検索の検索結果から1つ以上のアクティビティを選択することができる。他の実施形態では、インテグレータは、利用可能なアクティビティおよびトリガを遠隔データベースで検索し、遠隔データベースから1つ以上のアクティビティおよびトリガを選択することができる。他の実施形態では、インテグレータは、コマンドラインインターフェースを介して、利用可能なアクティビティおよびトリガを検索し、1つ以上のアクティビティおよびトリガを選択することができる。

【0048】

604において、プロセスアプリケーションのトリガおよび／またはアクティビティに関連付けられた1つ以上の入力データ値が、所望の状態を生成するように変更される。例えば、プロセスアプリケーションの任意のステップを選択することができ、選択されたステップに関連付けられた1つ以上の入力データ値を変更して、選択されたステップの出力を調整することができる。選択されたステップの調整が後続のステップの状態に及ぼす影響を判定することができる。

10

【0049】

606において、プロセスアプリケーションが格納される。いくつかの実施形態では、プロセスアプリケーションはローカルデータベースに格納される。他の実施形態では、プロセスアプリケーションは、リモートデータベースに格納される。

【0050】

図7は、プロセスアプリケーションを修正するためのプロセスの一実施形態を示すフローチャートである。いくつかの実施形態では、図7のプロセス700は、プロセスアプリケーション開発プラットフォーム110などのプロセスアプリケーション開発プラットフォームによって実装することができる。いくつかの実施形態では、プロセス700は、プロセスアプリケーション開発プラットフォーム110のようなプロセスアプリケーション開発プラットフォームによって実装することができる。図7のプロセスは、ステップ506またはステップ604の一部を実行するように実施することができる。

20

【0051】

図示の例では、プロセス700は、ステップの選択が受信される702から始まる。プロセスフローは、ステップ1からステップnで完了する1からnステップで構成される。プロセスフローの各ステップにおいて、関連するトリガまたはアクティビティは、関連する状態値を有する。プロセスフローの任意のステップが選択可能である。

30

【0052】

704において、各ステップに関連する状態値が表示される。いくつかの実施形態では、プロセスエンジンが動作している任意の時点における任意のステップの状態値を表示することができる。

【0053】

706において、選択されたステップに関連付けられた状態が修正される。たとえば、選択されたステップの状態は、1つ以上の入力データ値に基づく。選択されたステップの状態は、先行するステップの出力に基づくこともできる。1つ以上の入力データ値を変更して、選択されたステップの出力状態を調整することができる。

40

【0054】

708において、変更された入力変数を用いて、選択されたステップからプロセスフローを実行することができる。例えば、プロセスフローをステップ1から実行する必要はなく、変更された選択されたステップから実行することができ、後続のステップに関連する状態値の変化を観察することができる。いくつかの実施形態では、中断することなくプロセスフローを最後まで実行することができる。各プロセスステップに関連する状態値を表示することができる。他の実施形態では、プロセスフローは段階的に実行される。プロセスフローは、ユーザインターフェースから受信したユーザ入力に応答して、プロセスフローの後続のステップに進むことができる。各ステップの出力状態を表示することができます。いくつかの実施形態では、プロセスフローは最後まで実行することができるが、ユーザイ

50

ンタフェースから受信したユーザ入力を介していつでも中断することができる。割り込み発生時の各ステップの出力状態を表示することができる。

【 0 0 5 5 】

図 8 は、プロセスエンジンをデプロイするためのプロセスの一実施形態を示すフローチャートである。いくつかの実施形態では、図 8 のプロセス 8 0 0 は、図 8 のプロセスは、プロセスアプリケーション開発プラットフォーム 1 1 0 などのプロセスアプリケーション開発プラットフォーム、またはプロセスアプリケーション診断プラットフォーム 1 1 2 などのプロセス診断プラットフォームによって実装することができる。

【 0 0 5 6 】

8 0 2において、プロセスアプリケーションのためのプロセスエンジンコンポーネントの最小セットが決定される。プロセスエンジンコンポーネントの最小セットは、プロセスアプリケーション用にインテグレータによって定義された 1 つ以上のプロセスフローをサポートするために実際に必要とされるプロセスエンジンコンポーネントの最小数である。例えば、プロセスフローは、トリガおよび／またはフローによって呼び出されない 1 つ以上のアクティビティを含むことができる。アクティビティまたはトリガには、プロセスアプリケーションによって使用されないコンポーネントが含まれる場合がある。そのようなアクティビティー、トリガー、およびコンポーネントは識別され、除去される。

10

【 0 0 5 7 】

トリガおよび／またはアクティビティは、プロセスアプリケーションを実行するために必要な 1 つ以上のトランスポートに関連付けられてもよい。トランスポートの例には、H T T P / H T T P S 、 M Q T T 、 C o A P などがある。実際にプロセスアプリケーションによって呼び出されるトランスポートが識別され、プロセスアプリケーションに追加される。

20

【 0 0 5 8 】

8 0 4において、プロセスアプリケーションに含まれるトリガおよびアクティビティは、実行可能バイナリにコンパイルされる。いくつかの実施形態では、プロセスアプリケーションによって参照されないトリガおよびアクティビティに関連付けられたプロセスエンジンコンポーネントは無視され、実行可能ファイルから除外される。プロセスアプリケーションによって実際に呼び出されるトランスポートは識別され、実行可能ファイルに追加される。

【 0 0 5 9 】

30

8 0 6において、プロセスエンジンを含む実行可能ファイルがデバイスにデプロイされる。未使用的プロセスエンジンコンポーネントは実行可能ファイルから削除されるため、デバイスによって実行されると、プロセスエンジンはより迅速にオンラインになる可能性があります。いくつかの実施形態では、実行可能ファイルは、無線接続を介してデプロイされ、デバイスにダウンロードされる。他の実施形態では、実行可能ファイルは、物理メモリデバイスにデプロイされ、デバイスに転送される。他の実施形態では、実行可能ファイルは、デバイスへの有線接続を介してデプロイされる。いくつかの実施形態では、実行可能ファイルは、誤出力を生成したデバイスとは異なるデバイスにデプロイされる。

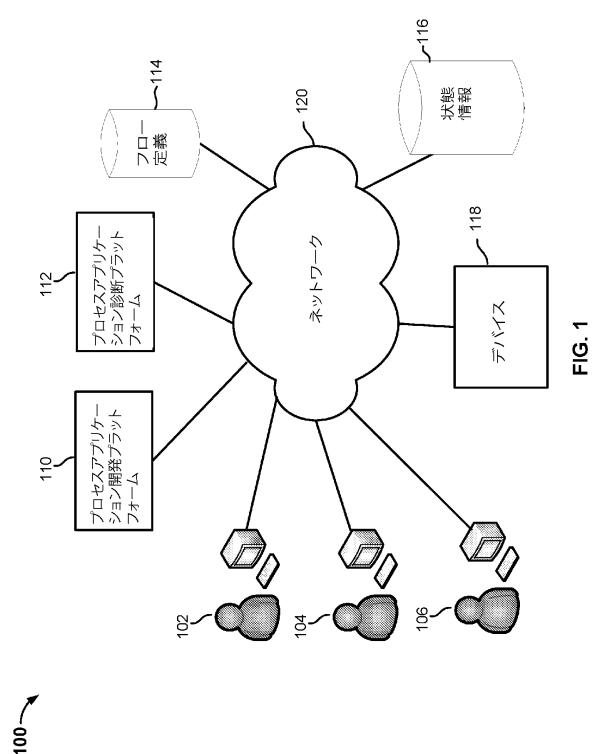
【 0 0 6 0 】

前述の実施形態は、理解を明確にするために詳細に記載しているが、本発明は、ここに記載された詳細に限定されない。本発明を実施する多くの代替方法がある。開示された実施形態は例示であり、限定期的ではない。

40

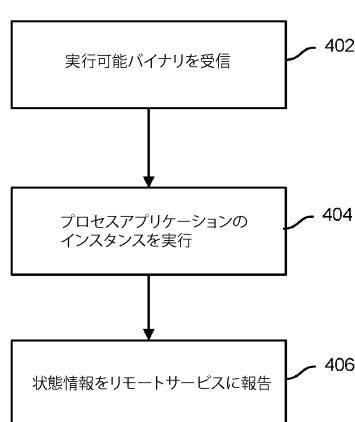
【図面】

【図 1】



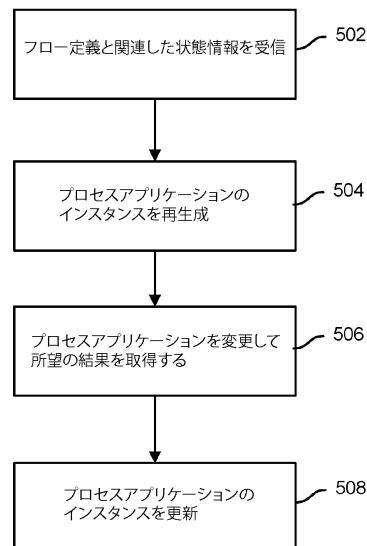
【図 4】

400 ↘



【図 5】

500 ↘



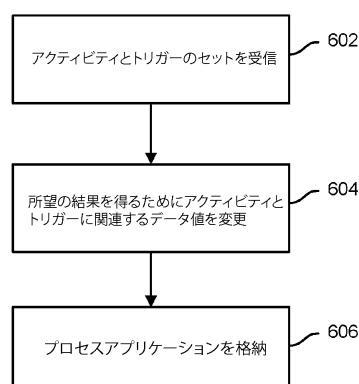
10

20

FIG. 4

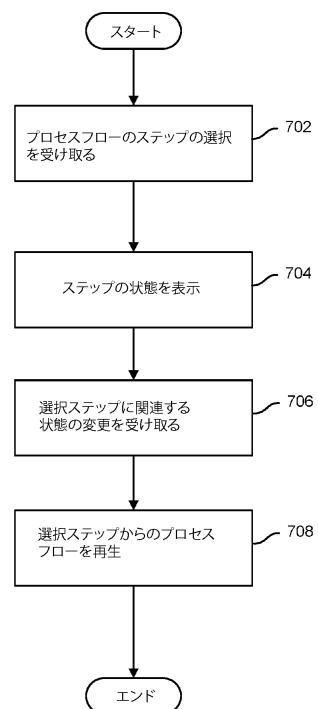
【図 6】

600 ↘



【図 7】

700 ↘



30

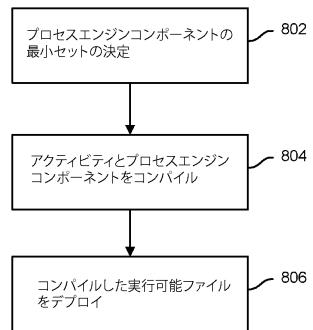
40

FIG. 7

50

【図 8】

800 ↘



10

20

FIG. 8

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(72)発明者 コズィカッソディ、ラジーブ

アメリカ合衆国、94304 カリフォルニア州、パロ アルト、3303 ヒルビュー アベニュー

審査官 中村 信也

(56)参考文献 特開2001-134464 (JP, A)

特開2012-038019 (JP, A)

特開2002-304295 (JP, A)

特開2010-211257 (JP, A)

特開2003-058395 (JP, A)

特開2005-018425 (JP, A)

(58)調査した分野 (Int.Cl., DB名)

G06F 11/07

G06F 11/28 - 11/36

G06F 8/00 - 8/38

G06F 8/60 - 8/77

G06F 9/44 - 9/445

G06F 9/451