

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7594163号
(P7594163)

(45)発行日 令和6年12月4日(2024.12.4)

(24)登録日 令和6年11月26日(2024.11.26)

(51)国際特許分類

G 0 6 F 9/44 (2018.01)

F I

G 0 6 F 9/44

請求項の数 20 (全36頁)

(21)出願番号	特願2020-553484(P2020-553484)	(73)特許権者	520262319
(86)(22)出願日	令和2年9月22日(2020.9.22)		ユーアイパス、インコーポレイテッド
(65)公表番号	特表2023-532823(P2023-532823 A)		U i P a t h , I n c .
(43)公表日	令和5年8月1日(2023.8.1)		アメリカ合衆国、ニューヨーク 1 0 0
(86)国際出願番号	PCT/US2020/051940		1 7 , ニューヨーク、ヴァンダービルト
(87)国際公開番号	WO2022/010513		アベニュー 1 , シックスティース フロア
(87)国際公開日	令和4年1月13日(2022.1.13)		1 V a n d e r b i l t A v e n u e
審査請求日	令和5年9月14日(2023.9.14)		, 6 0 t h F l o o r , N e w Y o
(31)優先権主張番号	16/925,544		r k , N Y 1 0 0 1 7 , U n i t e d
(32)優先日	令和2年7月10日(2020.7.10)	(74)代理人	S t a t e s o f A m e r i c a
(33)優先権主張国・地域又は機関	米国(US)		100180781
(31)優先権主張番号	16/924,910	(74)代理人	弁理士 安達 友和
(32)優先日	令和2年7月9日(2020.7.9)		100182903
	最終頁に続く	(72)発明者	弁理士 福田 武慶
			ダニエル ダインズ
			最終頁に続く

(54)【発明の名称】 第2のセッションで実行されるロボティックプロセスオートメーションロボットを介して第1のセッションで実行されるプロセスの自動化

(57)【特許請求の範囲】

【請求項1】

コンピュータによって実行される方法であって、
コンピューティングシステムのメインセッションにおいて、ロボティックプロセスオートメーション（RPA）ロボットを実行させることと、
前記メインセッションのRPAロボットによって、プロセス間通信（IPC）を介して、クライアントセッションにおいてRPAロボットをコールする又は起動することと、
前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッションのRPAロボットから一又は複数のメッセージを受け取ることと、
前記クライアントセッションのRPAロボットから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションのRPAロボットによって、前記メインセッションのRPAロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、
を含むことを特徴とする方法。

【請求項2】

前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッションにおいてアプリケーションをコールする又は起動することと、
前記アプリケーションのタスクの実行の間及び／又は前記タスクの実行の後、前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッション

のアプリケーションから一又は複数のメッセージを受け取ることと、

前記クライアントセッションのアプリケーションから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションのRPAロボットによって、前記メインセッションのRPAロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

をさらに含むことを特徴とする、請求項1に記載の方法。

【請求項3】

前記クライアントセッションのアプリケーションが実行を完了するまで、前記メインセッションのRPAロボットの実行を一時停止すること

をさらに含むことを特徴とする、請求項2に記載の方法。

【請求項4】

前記クライアントセッションのアプリケーションは、前記メインセッションのRPAロボットのワークフローにおいて、2回又はそれ以上コールされる又は起動されることを特徴とする、請求項2に記載の方法。

【請求項5】

前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッションにおいて追加のアプリケーションをコールする又は起動することと、

前記クライアントセッションの追加のアプリケーションから一又は複数のメッセージを受け取ることと、

前記クライアントセッションの追加のアプリケーションから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションのRPAロボットによって、前記メインセッションのRPAロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

をさらに含むことを特徴とする、請求項2に記載の方法。

【請求項6】

前記クライアントセッションのアプリケーションは、前記メインセッションのRPAロボットのワークフローのプロセス呼び出しアクティビティを介して、コールされる又は起動されることを特徴とする、請求項2に記載の方法。

【請求項7】

前記プロセス呼び出しアクティビティは、前記クライアントセッションのアプリケーションが前記クライアントセッションで実行されることを示すフラグを含むことを特徴とする、請求項6に記載の方法。

【請求項8】

前記タスクが完了した後、前記クライアントセッションのアプリケーションは実行し続けることを特徴とする、請求項2に記載の方法。

【請求項9】

前記メインセッションのRPAロボットと前記クライアントセッションのアプリケーションとの間のIPC通信が標準化された通信フォーマットであることを特徴とする、請求項2に記載の方法。

【請求項10】

前記クライアントセッションのRPAロボットが実行を完了するまで、前記メインセッションのRPAロボットの実行を一時停止することをさらに含むことを特徴とする、請求項1に記載の方法。

【請求項11】

前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッションにおいて追加のRPAロボットをコールする又は起動することと、

IPCを介して、前記クライアントセッションの追加のRPAロボットから一又は複数のメッセージを受け取ることと、

10

20

30

40

50

前記クライアントセッションの追加の R P A ロボットから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションの R P A ロボットによって、前記メインセッションの R P A ロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

をさらに含むことを特徴とする、請求項 1 に記載の方法。

【請求項 1 2】

前記クライアントセッションの R P A ロボットは、前記メインセッションの R P A ロボットのワークフローにおいて、2 回又はそれ以上コールされる又は起動されることを特徴とする、請求項 1 に記載の方法。

10

【請求項 1 3】

前記クライアントセッションの R P A ロボットは、前記メインセッションの R P A ロボットのワークフローのワークフロー呼び出しアクティビティを介して、コールされる又は起動されることを特徴とする、請求項 1 に記載の方法。

【請求項 1 4】

前記ワークフロー呼び出しアクティビティは、前記クライアントセッションの R P A ロボットが前記クライアントセッションで実行されることを示すフラグを含むことを特徴とする、請求項 1 3 に記載の方法。

【請求項 1 5】

コンピュータによって実行される方法であって、

20

コンピューティングシステムのメインセッションにおいて、ロボティックプロセスオートメーション (R P A) ロボットを実行させることと、

前記メインセッションの R P A ロボットによって、プロセス間通信 (I P C) を介して、クライアントセッションにおいてアプリケーションをコールする又は起動することと、

前記メインセッションの R P A ロボットによって、I P C を介して、前記クライアントセッションのアプリケーションから一又は複数のメッセージを受け取ることと、

前記クライアントセッションのアプリケーションから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションの R P A ロボットによって、前記メインセッションの R P A ロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

30

を含むことを特徴とする方法。

【請求項 1 6】

前記メインセッションの R P A ロボットによって、I P C を介して、前記クライアントセッションにおいて追加のアプリケーションをコールする又は起動することと、

前記クライアントセッションの追加のアプリケーションから一又は複数のメッセージを受け取ることと、

前記クライアントセッションの追加のアプリケーションから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションの R P A ロボットによって、前記メインセッションの R P A ロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

40

をさらに含むことを特徴とする、請求項 1 5 に記載の方法。

【請求項 1 7】

前記クライアントセッションのアプリケーションは、前記メインセッションの R P A ロボットのワークフローのプロセス呼び出しアクティビティを介して、コールされる又は起動されることを特徴とする、請求項 1 5 に記載の方法。

【請求項 1 8】

コンピュータによって実行される方法であって、

コンピューティングシステムのメインセッションにおいて、ロボティックプロセスオー

50

トメーション（RPA）ロボットを実行させることと、

前記メインセッションのRPAロボットによって、プロセス間通信（IPC）を介して、クライアントセッションにおいてRPAロボットをコールする又は起動することと、

前記メインセッションのRPAロボットによって、IPCを介して、クライアントセッションにおいてアプリケーションをコールする又は起動することと、

前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッションのRPAロボット及び前記クライアントセッションのアプリケーションから一又は複数のメッセージを受け取ることと、

前記クライアントセッションのRPAロボット及び前記クライアントセッションのアプリケーションから受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションのRPAロボットによって、前記メインセッションのRPAロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されている一又は複数のアプリケーション及び／又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

を含むことを特徴とする方法。

【請求項19】

前記メインセッションのRPAロボットによって、IPCを介して、前記クライアントセッションにおいて、追加のアプリケーション、追加のRPAロボット、又はその両方をコールする又は起動することと、

IPCを介して、前記クライアントセッションの追加のアプリケーション、前記クライアントセッションの追加のRPAロボット、又はその両方から、一又は複数のメッセージを受け取ることと、

前記クライアントセッションの追加のアプリケーション、前記クライアントセッションの追加のRPAロボット、又はその両方から受け取られた前記一又は複数のメッセージからの結果を使用して、前記メインセッションのRPAロボットによって、前記メインセッションのRPAロボットのワークフローの少なくとも一部を完了するか、前記メインセッションで実行されているアプリケーション及び／又はアプリケーションオブジェクトと対話するか、又はその両方を行うことと、

をさらに含むことを特徴とする、請求項18に記載の方法。

【請求項20】

前記クライアントセッションのアプリケーションは、前記メインセッションのRPAロボットのワークフローのプロセス呼び出しアクティビティを介して、コールされるか起動され、或いは、前記クライアントセッションのRPAロボットは、前記メインセッションのRPAロボットのワークフローのワークフロー呼び出しアクティビティを介して、コールされるか起動され、或いは、これらの両方が行われることを特徴とする、請求項18に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、2020年7月9日に出願された米国非仮特許出願番号16／924,910の継続出願であり、その利益を主張する、2020年7月10日に出願された米国非仮特許出願番号16／925,544の利益を主張する。これらのより早く出願された出願の主題は、参照によりその全体が本明細書に組み込まれる。

【0002】

本発明は、概して、ロボティックプロセスオートメーション（RPA）、より詳細には、第2のセッションで実行されるRPAロボットを介して第1のセッションで実行されるプロセスの自動化に関する。

【背景技術】

【0003】

10

20

30

40

50

アテンディッド（操作要）自動化ロボットは、例えば、通常、ユーザと同じセッションでユーザによって操作されるコンピューティングシステムで実行される。RPAロボットは、ユーザと協力して、例えば、ユーザのコマンドで特定のタスクを実行する場合がある。しかし、アテンディッド自動化シナリオでは、RPAロボットがユーザのコンピューティングシステムを「テイクオーバーする（引き継ぐ）」可能性がある。ユーザは、ロボットがコンピューティングシステムと対話している間に他のアクティビティを実行したいと思うかもしれないが、ユーザはそうすることができない場合がある。つまり、ロボットは、ユーザが行うのと同じ方法で（例えば、マウスクリックやキーボード入力をシミュレートするなどして）、ユーザインタフェース（UI）を介してアプリケーションを制御する。

【0004】

オペレーティングシステム又はオペレーティングシステムで実行されているアプリケーションの完全な又は部分的なコピーを作成する様々な技術が存在する。エミュレータは何十年にわたり存在しており、アプリケーションをテスト、デバッグする機能を開発者に提供する場合がある。例えば、エミュレータは、開発ツールの実行を直接サポートしていないオペレーティングシステムを使用するモバイルアプリケーションをテスト、デバッグする機能を開発者に提供する。Android（登録商標）及びiOS（登録商標）はどちらも、それらのモバイルオペレーションシステム上で開発者用ツールをネイティブで実行できないため、開発マシンから実行されてAndroid（登録商標）又はiOS（登録商標）アプリケーションをテスト、デバッグできるエミュレータを提供する。

【0005】

シミュレータにより、開発者は、開発マシンでの実行が困難又は不可能なアプリケーションの振る舞いを開発者がテスト、デバッグできる自身のローカルマシンでウィンドウをホストすることができる。例えば、シミュレータにより、開発者は、ボタンをクリックしてシミュレータを回転できる。これにより、シミュレータ内で実行されているアプリケーションに、このようなイベントにตอบสนองするアプリケーションの振る舞いをテスト、デバッグする目的でデバイスが回転されたことが通知される。別の一般的な例はマルチタッチである。多くの開発者マシンはタッチをサポートしていないため、シミュレータにより、開発者は、アプリケーションが複数のタッチポイントにどのようにตอบสนองするかをテスト、デバッグできる。Android（登録商標）及びiOS（登録商標）エミュレータもシミュレーション機能を提供する。さらに、Microsoft（登録商標）は、ユニバーサルWindowsプラットフォーム（UWP）アプリケーション用のシミュレータを提供する。

【0006】

仮想マシンは、マシン上で第2のオペレーティングシステムをホストし、ウィンドウを介して開かれ、監視されることが可能である。これは、完全に異なるオペレーティングシステムを実行し、ハードウェアをホストマシンと共有する。「ゲスト」マシンには、アプリケーションの「ゲスト」マシン自身のコピーがインストールされている必要があり、「ゲスト」マシンは、共通のリソースやファイルをユーザマシンと共有しない。

【0007】

Docker（登録商標）コンテナは、概念的には仮想マシンのハイブリッド形式である。実行される必要がある全てのアプリケーションが、ホストオペレーティングシステムで直接実行される不変のパッケージにパッケージ化される。パッケージは別のオペレーティングシステムの完全なコピーではないが、デフォルトではホストマシン上のアプリケーションやリソースを共有したりそれらにアクセスしたりすることはない。したがって、ユーザエクスペリエンスの観点からは、Docker（登録商標）コンテナは仮想マシンに類似するように感じられるが、技術的には、コンテナは完全に別のオペレーティングシステムで実行されていない。

【0008】

しかし、従来のエミュレータ、シミュレータ、仮想マシン（VM）、オペレーティングシステム（OS）レベルの仮想化を提供するハイブリッドVM（例えば、Docker（

10

20

30

40

50

登録商標) コンテナなど) は、ユーザと同じコンピューティングシステムで動作するアテンディッド自動化ロボットで発生する問題に対処していない。したがって、ユーザは、基本的に自身のコンピューティングシステムの見物人になり、ロボットの作業を監視し、ユーザとの対話(相互作用)を必要とするマシン上の他のアプリケーションと対話できなくなる。したがって、向上したアプローチが有益であり得る。

【発明の概要】

【0009】

本発明の特定の実施形態は、現在のRPA技術によってまだ完全に識別、認識、又は解決されていない、当該技術分野における課題及び必要性に対する解決方法を提供し得る。例えば、本発明の幾つかの実施形態は、第2のセッションで実行されるRPAロボットを介して第1のセッションで実行されるプロセスの自動化に関する。

10

【0010】

一実施形態において、コンピュータプログラムは、非一時的なコンピュータ読み取り可能な媒体に格納される。該コンピュータプログラムは、第1のセッションにおいて、プロセス間通信(IPC)ファシリテータ及びロボティックプロセスオートメーション(RPA)ドライバを実行させることと、第2のセッションにおいて、RPAロボットを実行させることとを少なくとも1つのプロセッサに実行させるように構成されている。該コンピュータプログラムはさらに、IPCファシリテータによって、IPCを介して、RPAロボットから一又は複数のメッセージを受け取ることと、RPAドライバを制御して、RPAロボットから受け取られた一又は複数のメッセージに基づいて、IPCファシリテータによって、第1のセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話させることとを少なくとも1つのプロセッサに実行させるように構成されている。

20

【0011】

別の一実施形態において、コンピュータによって実行される方法が、第2のセッションで実行されているRPAロボットによって、IPCを介して、第1のセッションで実行されているIPCファシリテータに、一又は複数のメッセージを送ることと、IPCファシリテータによって、IPCを介して、RPAロボットから一又は複数のメッセージを受け取ることとを含む。このコンピュータによって実行される方法はさらに、第1のセッションで実行されているRPAドライバを制御して、RPAロボットから受け取られた一又は複数のメッセージに基づいて、IPCファシリテータによって、第1のセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話させることを含む。

30

【0012】

さらに別の一実施形態において、コンピュータによって実行される方法は、第1のセッションで実行されているIPCファシリテータによって、IPCを介して、第2のセッションで実行されているRPAロボットから、一又は複数のメッセージを受け取ることを含む。このコンピュータによって実行される方法はさらに、第1のセッションで実行されているRPAドライバを制御して、RPAロボットから受け取られた一又は複数のメッセージに基づいて、IPCファシリテータによって、第1のセッションで実行されているアプリケーション又はアプリケーションオブジェクトと対話させることを含む。

40

【図面の簡単な説明】

【0013】

本発明の特定の実施形態の利点が容易に理解されるように、上記簡単に説明した本発明のより詳細な説明を、添付の図面に示す特定の実施形態を参照して行う。これらの図面は、本発明の典型的な実施形態のみを示すもので、その範囲を限定するものとみなされるべきではないことを理解されたい。本発明は、添付の図面の使用を通じて追加の特性及び詳細とともに記載され説明される。

【0014】

【図1】本発明の一実施形態によるロボティックプロセスオートメーション(RPA)システムを示すアーキテクチャ図である。

50

【0015】

【図2】本発明の一実施形態による、デPLOYされたRPAシステムを示すアーキテクチャ図である。

【0016】

【図3】本発明の一実施形態による、デザイナとアクティビティとドライバとの関係を示すアーキテクチャ図である。

【0017】

【図4】本発明の一実施形態によるRPAシステムを示すアーキテクチャ図である。

【0018】

【図5】本発明の一実施形態による、RPAロボットのセッション間の自動化を容易にするように構成されたコンピューティングシステムを示すアーキテクチャ図である。

10

【0019】

【図6】図6A～図6Gは、本発明の一実施形態による、ロボットセッションで実行されるRPAロボット、IPCファシリテータ、及びRPAドライバを使用して、ユーザセッションでフォームを完成させる一例を示す。

【0020】

【図7】図7A～図7Gは、本発明の一実施形態による、直接変数変更によりロボットセッションで実行されるRPAロボットを使用して、ユーザセッションでフォームを完成させる一例を示す。

【0021】

20

【図8】本発明の一実施形態による、ロボットセッションで実行されているRPAロボット、IPCファシリテータ、及びRPAドライバを介してユーザセッションで実行されるプロセスの自動化のためのプロセスを示すフローチャートである。

【0022】

【図9】本発明の一実施形態による、直接変数変更によりロボットセッションで実行されるRPAロボットを介してユーザセッションで実行されるプロセスの自動化のためのプロセスを示すフローチャートである。

【0023】

【図10】本発明の一実施形態による、メインセッションアプリケーションと対話する複数のクライアントセッションRPAロボットの一例を示す。

30

【0024】

【図11】本発明の一実施形態による、クライアントセッションでアテンディッド自動化RPAロボットを実行するプロセスを示すフロー図である。

【0025】

【図12】本発明の一実施形態による、メインセッションロボットM1、クライアントセッションアプリケーションA1、及び一対のクライアントセッションロボットC1、C2間のマルチロボット共同ワークフローの実行を示すフロー図である。

【0026】

【図13】本発明の一実施形態による、メインセッションRPAロボットとクライアントセッションRPAロボットの並行した実行を示すフロー図である。

40

【0027】

【図14】本発明の一実施形態による、メインセッションアプリケーションのトリガに基づくクライアントセッションRPAロボットの実行を示すフロー図である。

【0028】

【図15】本発明の一実施形態による、セッション間の自動化を実行するためのプロセスを示すフローチャートである。

【0029】

特に示さない限り、同様の符号は、添付の図面全体を通して一貫して対応する特徴を示す。

【発明を実施するための形態】

50

【0030】

幾つかの実施形態は、第2のセッションで実行されるRPAロボットを介して第1のセッションで実行されるプロセスの自動化に関する。例えば、フォーム（例えば、ウェブページ、電子メールアプリケーション（例えば、Outlook（登録商標）など）、スプレッドシートアプリケーション（例えば、Excel（登録商標）など）、顧客関係管理（CRM）システムアプリケーション（例えば、Salesforce（登録商標）など）、エンタープライズリソース管理（ERM）システム、サプライチェーン管理システム、特注のコンピュータアプリケーション、モバイルアプリケーション、入力可能なフィールドを有する別のアプリケーションなど）が第1のセッションで表示されてもよい。幾つかの実施形態において、第1のセッションは、ユーザセッション（本明細書ではメインセッション又は親セッションともいう）であってもよい。第1のセッションでアプリケーションのためのデータを取得し且つ／又は該データと対話する一又は複数のRPAロボットは、一又は複数の別のセッション（本明細書ではロボットセッション、第2のセッション、又は二次セッションともいう）で実行されてもよい。特定の実施形態において、ロボットセッションは、第1のセッションの子セッションであってもよい。

10

【0031】

幾つかの実施形態において、クライアントセッションで実行されているRPAロボットは、プロセス間通信（IPC）を介してメインセッションで実行されているRPAドライバにメッセージを送る。RPAロボットからのIPC通信に基づいて、RPAドライバは、コンピューティングシステムとの様々なオペレーティングシステム（OS）及び／又はアプリケーションプログラミングインタフェース（API）レベルの対話を実施してもよい。例えば、ドライバは、マウスの移動、ボタンのクリック、メニューの操作、フィールドへのテキストの入力、ウィンドウの開閉、ウィンドウの移動及び／又はサイズ変更、或いは、本発明の範囲から逸脱することなく任意の他の適切な対話（操作）を実装してもよい。

20

【0032】

このような機能を実装するために、ドライバは、IPCファシリテータを含むか、IPCファシリテータによって制御されるか、呼び出されてもよい。幾つかの実施形態において、IPCファシリテータは、別個のアプリケーション又はプロセスであってもよいし、或いは、ドライバのサブプロセスであってもよい。IPCファシリテータは、クライアントセッションでRPAロボットからのIPCメッセージをリスンすること（例えば、トリガをリスンすること）、RPAロボットとの間で通信を送受信すること、RPAロボットの実行ステータスを監視すること、これらのうちの組み合わせなどを行ってもよい。ファシリテータはまた、RPAロボットからのIPC通信に基づいて、ドライバに、親セッションで所望の機能を実装させてもよい。特定の実施形態において、IPCファシリテータは、所与のRPAロボットのためのスクリプトで構成されてもよい。幾つかの実施形態において、IPCファシリテータは、親セッションで実行される機能のためにカスタム設計されてもよい。

30

【0033】

幾つかの実施形態において、クライアントセッションにおけるRPAロボットは、コンピューティングシステム上のメモリに記憶された変数を直接変更してもよい。そして、このような変数にアクセスする親セッションで実行されているアプリケーションが、表示が更新されたとき、又は、アプリケーションがそのUIをリロードしたときに、表示された値を更新してもよい。例えば、Excel（登録商標）スプレッドシートの変数が、クライアントセッションで実行されているRPAロボットによって直接変更されてもよく、これにより、親セッションで表示される値も変更されるであろう。

40

【0034】

幾つかの実施形態において、RPAロボットの動作は、RPAロボットが実行されているときにユーザが他のアプリケーション又はインスタンスを使用することを妨げず、しかし、RPAロボットによって行われるデータ変更は、ユーザには、アプリケーションの表

50

示がユーザセッションウィンドウで更新されるように見えてもよい。しかし、特定の実施形態において、RPAロボットは、ユーザインタフェースを有しないアプリケーションと対話してもよい。

【0035】

幾つかの実施形態において、プロセスは次のように動作してもよい。ユーザは、ボタンをクリックするか、或いは、ユーザセッションの子セッション（例えば、ユーザがユーザのコンピューティングシステムを起動したときに起動される親セッション又はメインセッション）でRPAロボットを起動させてもよい。特定の実施形態において、例えば、ロボットセッションは、既に実行されていてもよいし、或いは、ロボットトレイからユーザによって別個に起動されてもよい。ユーザは、子セッションでロボットにそのワークフローを実行させ、ロボットは、親セッションで実行されているIPCファシリテータと通信することで、メインセッションで一又は複数のアプリケーションと対話してもよい。次に、IPCファシリテータは、RPAドライバに親セッションで所望の機能を実行させてもよい。

10

【0036】

幾つかの実施形態において、ロボットセッションで実行されているRPAロボットは、ワークフローに関連する少なくとも1つのアプリケーションが起動されていない場合、動作を一時停止してメッセージを提供してもよく、IPCファシリテータは、ロボットが動作を継続できるようにアプリケーションを起動してもよい。ロボットがそのワークフローの実行を開始すると、IPCファシリテータ及びRPAドライバを介して変更が行われた場合に、アプリケーションで表示されているデータへの変更がメインセッションウィンドウで可視であってもよい。例えば、クライアントセッションでRPAロボットによって提供されたデータに基づいてドライバがデータを入力すると（例えば、IPCを介して、フラットファイル又はデータベースにデータを保存するなど）、データが表示されてもよい。

20

【0037】

非限定的な例として、幾つかの実施形態において、RPAロボットをクライアントセッションで起動させる、又は、クライアントセッションで既に実行されているRPAロボットと対話するショートカットを、Salesforce（登録商標）で作成してもよい。ユーザがショートカットをクリックすると、RPAロボットはRPAプロセスをそのRPAワークフローで設計されているように実行することを開始する。その実行の結果（データベースからの情報の取得、人工知能（AI）／機械学習（ML）モデルの呼び出し及び結果の取得、複数のレガシーシステムからの情報の取得など）に基づいて、RPAロボットはIPCファシリテータと対話し、IPCファシリテータは、親セッションで実行中のSalesforce（登録商標）アプリケーションインスタンスと適宜対話する。これにより、実行中のSalesforce（登録商標）アプリケーションインスタンスにおける表示される情報の一部が変更されてもよく、例えば、RPAドライバが対話を実行するようにユーザに見えてもよい（可視であってもよい）。

30

【0038】

特定の実施形態において、クライアントセッションで実行されているRPAロボットが、親セッションにおけるアプリケーションを、IPCファシリテータを介して開いてもよい。次に、IPCファシリテータは、RPAドライバを介して、アプリケーションに関連するデータにアクセスし、追加、変更、及び／又は削除を行ってもよい。このようにして、RPAロボットが親セッションで実行されていない場合でも、RPAロボットは、親セッション自体で実行されているRPAロボットと同様に見えるが機能的に異なる方法で、親セッションアプリケーションで対話を生じさせることができる。

40

【0039】

幾つかの実施形態において、ユーザは、ロボットに、ウェブサイトアクセスして何らかの情報を収集するクライアントセッションでワークフローを実行させてもよい。次に、RPAロボットは、この情報をIPCファシリテータに提供し、IPCファシリテータは、RPAドライバに、例えば、収集された情報をメインセッションで可視であるスプレッ

50

ドシートに入力させる。特定の実施形態において、例えばSalesforce（登録商標）などのアプリケーションがメインセッションで開かれる。次に、ユーザは、クライアントセッションで、現在のクライアントIDを読み取り、ウェブサイト（例えば、HubSpot（登録商標）など）にアクセスして、クライアントとウェブサイトとの対話に関する情報を収集する自動化（自動動作）を実行させる。しかし、本発明の範囲から逸脱することなく、アテンディッドRPAロボットは、RPAファシリテータを介して任意の適切なアプリケーションと対話してもよく、且つ／又は、任意の適切なソース（例えば、データベース、別のアプリケーションなど）からデータを取得してもよい。

【0040】

幾つかの実施形態のアプリケーションには、エミュレータ、シミュレータ、VM、OSレベルの仮想化を提供するハイブリッドVM（例えば、Docker（登録商標）コンテナなど）が含まれるが、これらに限定されない。幾つかの実施形態は、アテンディッド自動化プロセスによって制御されているアプリケーションのUIを含むウィンドウとして一または複数のロボットセッションを作成し、ホストする。特定の実施形態において、ロボットが対話するアプリケーションのインタフェースのみが示される。幾つかの実施形態において、クライアントセッションウィンドウは全く示されない。本明細書で使用される場合、「ウィンドウ」は、本発明の範囲から逸脱することなく、メインUI内に示されるUIを表すウィンドウ、コンピューティングシステムの第2のディスプレイの第2の画面、仮想デスクトップ、隔離された環境（即ち、該環境内で起動された全てのアプリケーション（「子」という）のUIを描画し、（ホストセッションのコンテキストにおいて）それらを実行するウィンドウ（「ホスト」という）など）などに適用されてもよい。複数のセッションを実行することにより、ユーザが第1のセッション（例えば、親セッションなど）と対話している間、ロボットがロボットのセッションで動作できる。したがって、ロボットがそのデータと対話している間、ユーザは様々なアプリケーションと対話することが可能であってもよい。

【0041】

幾つかの実施形態において、本発明の範囲から逸脱することなく、任意の数のロボットの任意の所望の数のセッションが作成されて使用されてもよい。例えば、ユーザは第1のセッションで動作（操作）してもよく、第1のロボットは第2のセッションで動作してもよく、第2のロボットは第3のセッションで動作してもよいなど。特定の実施形態において、複数のロボットが単一のセッションで動作してもよく、RPAファシリテータを介して一又は複数の共通のアプリケーションと対話するために交代する可能性があってもよい。

【0042】

セッションを作成する機能は、例えば、Windows（登録商標）ターミナルサービスの子セッションを介して実装されてもよく、これにより、ユーザがログアウトする必要なく、ユーザ自身のマシンにセッションを作成できる。新しく作成されたセッションは、子ウィンドウとして表示され、ユーザのセッションに存在するアプリケーションを含み、起動する。つまり、ユーザとロボットの分離がUIレベルで生じる。例えば、ファイルが削除された場合、これはコンピューティングシステムで実行されている全てのセッションにわたり生じる。

【0043】

特定の実施形態が、ロボティックプロセスオートメーション（RPA）に使用されてもよい。図1は、本発明の一実施形態による、RPAシステム100を示すアーキテクチャ図である。RPAシステム100は、開発者がワークフローを設計及び実装することを可能にするデザイナー110を含む。デザイナー110は、アプリケーション統合、並びにサードパーティアプリケーション、管理情報技術（IT）タスク、及びビジネスITプロセスの自動化のためのソリューションを提供してもよい。デザイナー110は、ビジネスプロセスのグラフィック表現である自動化プロジェクトの開発を容易にしてもよい。簡単に言うと、デザイナー110はワークフロー及びロボットの開発とデプロイメントを容易にする。

【0044】

自動化プロジェクトは、開発者が、本明細書において「アクティビティ」として定義される、ワークフローで開発されたカスタムセットのステップ間の実行順序及び関係を制御できるようにすることで、ルールベースのプロセスの自動化を可能にする。デザイナー110の実施形態の1つの商業的な例は、UiPath Studio（商標）である。各アクティビティには、例えばボタンのクリック、ファイルの読み込み、ログパネルへの書き込みなどのアクションが含まれていてもよい。幾つかの実施形態において、ワークフローがネストされ又は埋め込まれてもよい。

【0045】

一部の種類のワークフローには、シーケンス、フローチャート、FSM、及び／又はグローバル例外ハンドラが含まれてもよいが、これらに限定されない。シーケンスは、線形プロセスに特に適している可能性があり、ワークフローを混乱させることなく、あるアクティビティから別のアクティビティへのフローを可能にする。フローチャートは、より複雑なビジネスロジックに特に適している可能性があり、複数の分岐論理演算子によって、より多様な方法で決定の統合及びアクティビティの接続を可能にする。FSMは、大規模なワークフローに特に適している可能性がある。FSMは、実行時に有限数の状態を使用してもよく、それらの状態は、条件（即ち、遷移）又はアクティビティによってトリガされる。グローバル例外ハンドラは、実行エラーが発生したときのワークフローの振る舞いを決定したり、プロセスをデバッグしたりするのに特に適している可能性がある。

【0046】

ワークフローがデザイナー110で開発されると、ビジネスプロセスの実行は、デザイナー110で開発されたワークフローを実行する一又は複数のロボット130を調整するコンダクタ120によって調整される。コンダクタ120の実施形態の1つの商業的な例は、UiPath Orchestrator（商標）である。コンダクタ120は、環境におけるリソースの作成、監視、及びデプロイメントの管理を容易にする。コンダクタ120は、サードパーティのソリューション及びアプリケーションとの統合ポイント又は集約ポイントの1つとして機能してもよい。

【0047】

コンダクタ120は、全てのロボット130を管理して、集中ポイントからロボット130を接続して実行してもよい。管理可能なロボット130の種類には、アテンディッドロボット132、アンアテンディッドロボット134、開発ロボット（アンアテンディッドロボット134と同様であるが、開発及びテストの目的で使用される）、及び非生産ロボット（アテンディッドロボット132と同様であるが、開発及びテストの目的で使用される）が含まれるが、これらに限定されない。アテンディッドロボット132は、ユーザイベントによってトリガされ、又は自動的に実行されるようにスケジュールされて、同じコンピューティングシステム上で人間と一緒に動作する。アテンディッドロボット132は、集中プロセス展開及びロギング媒体のためにコンダクタ120と共に使用されてもよい。アテンディッドロボット132は、人間のユーザが様々なタスクを達成するのを助け、ユーザイベントによってトリガされてもよい。幾つかの実施形態において、プロセスは、この種のロボットのコンダクタ120から開始されることができず、且つ／又は、ロックされた画面の下で実行できない。特定の実施形態において、アテンディッドロボット132は、ロボットレイから又はコマンドプロンプトから開始できるのみである。幾つかの実施形態において、アテンディッドロボット132は、人間の監督下で動作するべきである。

【0048】

アンアテンディッドロボット134は、仮想環境で操作不要で実行され、多くのプロセスを自動化できる。アンアテンディッドロボット134は、遠隔実行、監視、スケジューリング、及びワークキューのサポートの提供を担当してもよい。幾つかの実施形態において、全てのロボットの種類のデバッグをデザイナー110から実行してもよい。アテンディッドロボットとアンアテンディッドロボットの両方が、メインフレーム、ウェブアプリケーション、VM、エンタープライズアプリケーション（例えば、SAP（登録商標）、S

10

20

30

40

50

alesForce（登録商標）、Oracle（登録商標）などによって生成されたもの）、及びコンピューティングシステムアプリケーション（例えば、デスクトップ及びラップトップアプリケーション、モバイルデバイスアプリケーション、ウェアラブルコンピュータアプリケーションなど）を含むがこれらに限定されない様々なシステム及びアプリケーションを自動化してもよい。

【0049】

コンダクタ120は、プロビジョニング、デプロイメント、バージョンニング、コンフィギュレーション、キューイング、監視（モニタリング）、ロギング、及び／又は相互接続性の提供を含むがこれらに限定されない様々な機能を有してもよい。プロビジョニングには、ロボット130とコンダクタ120（例えば、ウェブアプリケーションなど）との間の接続の作成及び保守が含まれてもよい。デプロイメントには、実行のために割り当てられたロボット130へのパッケージバージョンの正しい配信を保証することが含まれてもよい。バージョンニングには、幾つかの実施形態において、幾つかのプロセス又はコンフィギュレーションの固有のインスタンスの管理が含まれてもよい。コンフィギュレーションには、ロボット環境及びプロセスコンフィギュレーションの保守及び配信が含まれてもよい。キューイングには、キュー及びキューアイテムの管理の提供が含まれてもよい。監視には、ロボット識別データの追跡及びユーザ権限の維持が含まれてもよい。ロギングには、データベース（例えば、SQLデータベースなど）及び／又は他のストレージメカニズム（例えば、ElasticSearch（登録商標）など。これは、大規模なデータセットを記憶してすばやくクエリを実行する機能を提供する）へのログの記憶及びインデックス付けが含まれてもよい。コンダクタ120は、サードパーティのソリューション及び／又はアプリケーションのための通信の集中ポイントとして機能することで、相互接続性を提供してもよい。

【0050】

ロボット130は、デザイナー110に組み込まれたワークフローを実行する実行エージェントである。ロボット130の幾つかの実施形態のうち1つの商用的な例は、UiPath Robots（商標）である。幾つかの実施形態において、ロボット130は、デフォルトで、Microsoft Windows（登録商標）サービスコントロールマネージャ（SCM）が管理するサービスをインストールする。その結果、そのようなロボット130が、ローカルシステムアカウントでインタラクティブなWindows（登録商標）セッションを開き、Windows（登録商標）サービスの権限を有することができる。

【0051】

幾つかの実施形態において、ロボット130は、ユーザモードでインストールされてもよい。そのようなロボット130の場合、これは、ユーザのもとでロボット130がインストールされて、そのユーザと同じ権利をロボット130が有することを意味する。この特徴は、高密度（HD）ロボットで利用可能であってもよく、各マシンの最大限の活用を確実にしてもよい。幾つかの実施形態において、任意の種類のロボット130が、HD環境で構成されてもよい。

【0052】

幾つかの実施形態におけるロボット130は、それぞれが特定の自動化タスク専用である幾つかのコンポーネントに分割される。幾つかの実施形態におけるロボットコンポーネントには、SCM管理のロボットサービス、ユーザモードのロボットサービス、エグゼキュータ、エージェント、及びコマンドラインが含まれるが、これらに限定されない。SCM管理のロボットサービスは、Windows（登録商標）セッションを管理、監視してコンダクタ120と実行ホスト（即ち、ロボット130が実行されるコンピューティングシステム）との間のプロキシとして機能する。このようなサービスは、ロボット130の資格情報を託され、これを管理する。コンソールアプリケーションは、ローカルシステムのもとでSCMによって起動される。

【0053】

幾つかの実施形態におけるユーザモードロボットサービスは、Windows（登録商標）セッションを管理、監視し、コンダクタ120と実行ホストとの間のプロキシとして機能する。ユーザモードロボットサービスは、ロボット130の資格情報を託され、これを管理してもよい。SCM管理のロボットサービスがインストールされていない場合、Windows（登録商標）アプリケーションが自動的に起動されてもよい。

【0054】

エグゼキュータは、Windows（登録商標）セッションのもとで所定のジョブを実行してもよい（即ち、エグゼキュータはワークフローを実行してもよい。エグゼキュータは、モニタ毎のドット／インチ（DPI）設定を認識していてもよい。エージェントは、システムトレイウィンドウで利用可能なジョブを表示するWindows（登録商標）Presentation Foundation（WPF）アプリケーションであってもよい。エージェントはこのサービスのクライアントであってもよい。エージェントは、ジョブの開始又は停止を要求し、設定を変更してもよい。コマンドラインはそのサービスのクライアントであってもよい。コマンドラインは、ジョブの開始を要求可能なコンソールアプリケーションであり、その出力を待つ。

【0055】

上記で説明したようにロボット130のコンポーネントを分割することにより、開発者、サポートユーザ、及びコンピューティングシステムが、各コンポーネントの実行内容の実行、識別、及び追跡をより容易に行うことができる。このように、例えばエグゼキュータとサービスに異なるファイアウォールルールを設定するなど、コンポーネント毎に特別な振る舞いが構成されてもよい。幾つかの実施形態において、エグゼキュータは、モニタ毎のDPI設定を常に認識していてもよい。その結果、ワークフローが作成されたコンピューティングシステムの構成に関わらず、ワークフローが任意のDPIで実行されてもよい。幾つかの実施形態において、デザイナー110からのプロジェクトは、ブラウザのズームレベルから独立していてもよい。DPIを認識しない又は意図的に認識しないとマークされているアプリケーションの場合、幾つかの実施形態においてDPIが無効にされてもよい。

【0056】

図2は、本発明の一実施形態による、デプロイされたRPAシステム200を示すアーキテクチャ図である。幾つかの実施形態において、RPAシステム200は、図1のRPAシステム100であってもよいし、その一部であってもよい。クライアント側、サーバ側、又はこれらの両方が、本発明の範囲から逸脱することなく、任意の所望の数のコンピューティングシステムを含んでもよいことに留意されたい。クライアント側では、ロボットアプリケーション210が、エグゼキュータ212、エージェント214、及びデザイナー216を含む。しかし、幾つかの実施形態において、デザイナー216はコンピューティングシステム210で実行されていなくてもよい。エグゼキュータ212はプロセスを実行する。図2に示すように、複数のビジネスプロジェクトが同時に実行されてもよい。このような実施形態において、エージェント214（例えば、Windows（登録商標）サービスなど）は、全てのエグゼキュータ212の単一の接続ポイントである。このような実施形態における全てのメッセージは、コンダクタ230にログインされる。このコンダクタ230は、さらにデータベースサーバ240、インデクササーバ250、又はこれらの両方を介して、それらのメッセージを処理する。図1に関して上記で説明したように、エグゼキュータ212はロボットコンポーネントであってもよい。

【0057】

幾つかの実施形態において、ロボットは、マシン名とユーザ名との間の関連付けを表す。ロボットは同時に複数のエグゼキュータを管理してもよい。同時に実行されている複数の対話型セッションをサポートするコンピューティングシステム（例えば、Windows（登録商標）Server 2012など）では、複数のロボットが同時に実行され、それぞれが一意のユーザ名を使用する個別のWindows（登録商標）セッションで実行されてもよい。これを上記のHDロボットという。

【0058】

エージェント214はまた、ロボットのステータスを送り（例えば、ロボットがまだ機能していることを示す「ハートビート」メッセージを定期的に送り）、実行されるパッケージの必要なバージョンをダウンロードすることも担当する。幾つかの実施形態において、エージェント214とコンダクタ230との間の通信は、常にエージェント214によって開始される。通知シナリオにおいて、エージェント214は、コンダクタ230によって後で使用されるWeb Socketチャネルを開き、ロボットにコマンド（例えば、開始、停止など）を送ってもよい。

【0059】

サーバ側では、プレゼンテーション層（ウェブアプリケーション232、Open Data Protocol（オープンデータプロトコル）（ODa）Representative State Transfer（リプレゼンタティブステートトランスファー）（REST）Application Programming Interface（アプリケーションプログラミングインタフェース）（API）エンドポイント234、通知・監視236）、サービス層（API実装／ビジネスロジック238）、及び永続層（データベースサーバ240及びインデクササーバ250）が含まれる。コンダクタ230には、ウェブアプリケーション232、ODa REST APIエンドポイント234、通知・監視236、及びAPI実装／ビジネスロジック238が含まれる。幾つかの実施形態において、コンダクタ230のインタフェースで（例えば、ブラウザ220を介して）ユーザが実行する殆どのアクションが、様々なAPIを呼び出すことで実行される。このようなアクションには、本発明の範囲から逸脱することなく、ロボットでのジョブの開始、キュー内のデータの追加／削除、操作不要で実行するジョブのスケジューリングなどが含まれてもよいが、これらに限定されない。ウェブアプリケーション232は、サーバプラットフォームのビジュアル層である。このような実施形態において、ウェブアプリケーション232は、ハイパーテキストマークアップ言語（HTML）及びJavaScript（JS）を使用する。しかし、本発明の範囲から逸脱することなく、任意の所望のマークアップ言語、スクリプト言語、又は任意の他のフォーマットが使用されてもよい。このような実施形態において、ユーザは、コンダクタ230を制御するための様々なアクションを実行するため、ブラウザ220を介してウェブアプリケーション232からウェブページと対話する。例えば、ユーザは、ロボットグループを作成し、ロボットにパッケージを割り当て、ロボット毎に且つ／又はプロセス毎にログを分析し、ロボットを起動、停止させるなどしてもよい。

【0060】

ウェブアプリケーション232に加えて、コンダクタ230には、ODa REST APIエンドポイント234を公開するサービス層も含まれる。しかし、本発明の範囲から逸脱することなく、他のエンドポイントが含まれてもよい。REST APIは、ウェブアプリケーション232とエージェント214の両方によって使用される。このような実施形態において、エージェント214は、クライアントコンピュータ上の一又は複数のロボットのスーパーバイザである。

【0061】

このような実施形態におけるREST APIは、コンフィギュレーション、ロギング、監視、及びキューイングの機能をカバーする。幾つかの実施形態において、コンフィギュレーションエンドポイントが使用されて、アプリケーションユーザ、権限、ロボット、アセット、リリース、及び環境を定義、構成してもよい。ロギングRESTエンドポイントが使用されて、例えばエラー、ロボットによって送られた明示的なメッセージ、その他の環境固有の情報など、様々な情報をログに記録してもよい。デプロイメントRESTエンドポイントがロボットによって使用されて、コンダクタ230でジョブ開始コマンドが使用される場合に実行する必要があるパッケージバージョンをクエリしてもよい。キューイングRESTエンドポイントは、例えばキューへのデータの追加、キューからのトランザクションの取得、トランザクションのステータスの設定など、キュー及びキューアイテ

ムの管理を担当してもよい。

【0062】

監視RESTエンドポイントは、ウェブアプリケーション232及びエージェント214を監視してもよい。通知・監視API236は、エージェント214の登録、エージェント214へのコンフィギュレーション設定の配信、並びにサーバ及びエージェント214からの通知の送受信に使用されるRESTエンドポイントであってもよい。幾つかの実施形態において、通知・監視API236はまた、WebSocket通信を使用してもよい。

【0063】

永続層は、この実施形態では1対のサーバ、つまり、データベースサーバ240（例えば、SQLサーバなど）及びインデクササーバ250を含む。この実施形態のデータベースサーバ240は、ロボット、ロボットグループ、関連プロセス、ユーザ、ロール（役割）、スケジュールなどのコンフィギュレーションを記憶する。このような情報は、幾つかの実施形態において、ウェブアプリケーション232を介して管理される。データベースサーバ240は、キュー及びキューアイテムを管理してもよい。幾つかの実施形態において、データベースサーバ240は、（インデクササーバ250に加えて又はその代わりに）ロボットによってログに記録されたメッセージを記憶してもよい。

【0064】

幾つかの実施形態において任意であるインデクササーバ250は、ロボットによってログに記録された情報を記憶し、インデックスを付ける。特定の実施形態において、インデクササーバ250は、コンフィギュレーション設定を通じて無効にされてもよい。幾つかの実施形態において、インデクササーバ250は、オープンソースプロジェクトの全文検索エンジンであるElasticSearch（登録商標）を使用する。ロボットによって（例えば、ログメッセージ、行書き込みなどのアクティビティを使用して）ログに記録されたメッセージは、ロギングRESTエンドポイントを介してインデクササーバ250に送られてもよく、そこで将来の利用のためにインデックスが付けられてもよい。

【0065】

図3は、本発明の一実施形態による、デザイナ310とアクティビティ320、330とドライバ340との間の関係300を示すアーキテクチャ図である。上記のとおり、開発者は、デザイナ310を使用して、ロボットによって実行されるワークフローを開発する。ワークフローは、ユーザ定義のアクティビティ320とUI自動化アクティビティ330を含んでもよい。幾つかの実施形態は、本明細書においてコンピュータビジョン（CV）という、画像における非テキストのビジュアルコンポーネントを識別することができる。そのようなコンポーネントに関連する一部のCVアクティビティには、クリック（click）、タイプ（type）、テキストを取得（get text）、ホバー（hover）、要素の有無を検出（element exists）、スコープの更新（refresh scope）、ハイライト（highlight）などが含まれてもよいが、これらに限定されない。幾つかの実施形態において、クリック（click）は、例えば、CV、光学式文字認識（OCR）、ファジーテキストマッチング、及び複数のアンカーを使用する要素を識別し、クリックする。タイプ（type）は、上記を使用して要素を識別してもよく、要素内でタイプする。テキストを取得（get text）は、特定のテキストの位置を識別し、OCRを使用してスキャンしてもよい。ホバー（hover）は、要素を識別し、その上にホバーしてもよい。要素の有無を検出（element exists）は、上記の技術を使用して、要素が画面上に存在するか否かを確認してもよい。幾つかの実施形態において、デザイナ310で実装可能な数百又は数千ものアクティビティがあってもよい。しかし、本発明の範囲から逸脱することなく、任意の数及び／又は種類のアクティビティが利用可能であってもよい。

【0066】

UI自動化アクティビティ330は、低レベルのコードで書かれた特別な低レベルのアクティビティ（例えば、CVアクティビティなど）のサブセットであり、画面との対話を

10

20

30

40

50

容易にする。UI自動化アクティビティ330は、ロボットが所望のソフトウェアと対話することを可能にするドライバ340を介して、このような対話を容易にする。例えば、ドライバ340は、OSドライバ342、ブラウザドライバ344、VMドライバ346、エンタープライズアプリケーションドライバ348などを含んでもよい。

【0067】

ドライバ340は、低レベルでOSと対話して、フックを探したりキーを監視したりするなどしてもよい。ドライバ340は、Chrome（登録商標）、IE（登録商標）、Citrix（登録商標）、SAP（登録商標）などとの統合を容易にしてもよい。例えば、「クリック」アクティビティは、ドライバ340を介してそのような異なるアプリケーションで同じ役割を果たす。

【0068】

図4は、本発明の一実施形態によるRPAシステム400を示すアーキテクチャ図である。幾つかの実施形態において、RPAシステム400は、図1及び／又は図2のRPAシステム100及び／又は200であってもよいし、それを含んでもよい。RPAシステム400は、ロボットを実行する複数のクライアントコンピューティングシステム410を含む。コンピューティングシステム410は、そこで実行されるウェブアプリケーションを介してコンダクタコンピューティングシステム420と通信可能である。次に、コンダクタコンピューティングシステム420は、データベースサーバ430及び任意のインデクササーバ440と通信可能である。

【0069】

図1及び図3に関して、これらの実施形態においてウェブアプリケーションが使用されているが、本発明の範囲から逸脱することなく、任意の適切なクライアント／サーバソフトウェアが使用されてもよいことに留意されたい。例えば、コンダクタは、クライアントコンピューティングシステム上の非ウェブベースのクライアントソフトウェアアプリケーションと通信するサーバ側アプリケーションを実行してもよい。

【0070】

図5は、本発明の一実施形態による、RPAロボットのセッション間の自動化を容易にするように構成されたコンピューティングシステム500を示すアーキテクチャ図である。幾つかの実施形態において、コンピューティングシステム500は、本出願において図示及び／又は説明される一又は複数のコンピューティングシステムであってもよい。コンピューティングシステム500は、情報を通信するためのバス505又は他の通信メカニズムと、情報を処理するためにバス505に接続されたプロセッサ510とを含む。プロセッサ510は、中央処理装置（CPU）、特定用途向け集積回路（ASIC）、フィールドプログラマブルゲートアレイ（FPGA）、グラフィックスプロセッシングユニット（GPU）、それらの複数の例、及び／又はそれらのうちの任意の組み合わせを含む、任意の種類の汎用又は特定用途のプロセッサであってもよい。プロセッサ510はまた、複数の処理コアを有してもよく、コアの少なくとも一部が、特定の機能を実行するように構成されてもよい。幾つかの実施形態において、複数並列処理を使用されてもよい。特定の実施形態において、少なくとも1つのプロセッサ510が、生体ニューロンを模倣する処理要素を含むニューロモーフィック回路であってもよい。幾つかの実施形態において、ニューロモーフィック回路は、フォンノイマンコンピューティングアーキテクチャの典型的なコンポーネントを必要としなくてもよい。

【0071】

コンピューティングシステム500は、プロセッサ510によって実行される情報及び命令を記憶するためのメモリ515をさらに含む。メモリ515は、ランダムアクセスメモリ（RAM）、リードオンリメモリ（ROM）、フラッシュメモリ、キャッシュ、例えば磁気若しくは光ディスクなどの静的記憶装置、又は任意の他の種類の非一時的なコンピュータ読み取り可能な媒体、又はこれらのうちの組み合わせのうちの任意の組み合わせから構成されてもよい。非一時的なコンピュータ読み取り可能な媒体は、プロセッサ510によってアクセス可能な任意の利用可能な媒体であってもよく、揮発性媒体、不揮発性媒

10

20

30

40

50

体、又はその両方を含んでもよい。媒体は、取り外し可能、取り外し不可能、又はその両方であってもよい。

【0072】

さらに、コンピューティングシステム500は、無線及び／又は有線接続を介して通信ネットワークへのアクセスを提供するために、例えばトランシーバなどの通信デバイス520を含む。幾つかの実施形態において、通信デバイス520は、本発明の範囲から逸脱することなく、周波数分割多元接続(FDMA)、シングルキャリアFDMA(SC-FDMA)、時分割多元接続(TDMA)、符号分割多元接続(CDMA)、直交周波数分割多重方式(OFDM)、直交周波数分割多元接続(OFDMA)、移動体用グローバルシステム(GSM)通信、汎用パケット無線サービス(GPRS)、ユニバーサル移動体通信システム(UMTS)、cdma2000、広帯域CDMA(W-CDMA)、高速ダウンリンクパケットアクセス(HSDPA)、高速アップリンクパケットアクセス(HSUPA)、高速パケットアクセス(HSPA)、Long Term Evolution(LTE)、LTEアドバンスド(LTE-A)、802.11x、Wi-Fi、Zigbee、超広帯域無線(UWB)、802.16x、802.15、Home Node-B(HnB)、Bluetooth、Radio Frequency Identification(RFID)、Infrared Data Association(IrDA)、Near-Field Communications(NFC)、第5世代(5G)、New Radio(NR)、これらのうちの任意の組み合わせ、及び／又は任意の他の現在存在する又は将来実施される通信規格及び／又はプロトコルを使用するように構成されてもよい。幾つかの実施形態において、通信デバイス520は、本発明の範囲から逸脱することなく、単一、アレイ、フェーズド、スイッチド、ビームフォーミング、ビームステアリング、これらのうちの組み合わせ、及び／又は任意の他のアンテナ構成である一又は複数のアンテナを含んでもよい。

【0073】

プロセッサ510は、バス505を介して、例えばプラズマディスプレイ、液晶ディスプレイ(LCD)、発光ダイオード(LED)ディスプレイ、電界放出ディスプレイ(FED)、有機発光ダイオード(OLED)ディスプレイ、フレキシブルOLEDディスプレイ、フレキシブル基板ディスプレイ、プロジェクションディスプレイ、4Kディスプレイ、高精細ディスプレイ、Retina(登録商標)ディスプレイ、インプレーンスイッチング(IPS)ディスプレイ、又はユーザに情報を表示するための任意の他の適切なディスプレイなどのディスプレイ525にさらに接続される。ディスプレイ525は、抵抗性、容量性、表面弾性波(SAW)容量性、赤外線、光学イメージング、分散信号技術、音響パルス認識、フラストレート全内部反射などを使用して、タッチ(触覚)ディスプレイ、3次元(3D)タッチディスプレイ、マルチ入力タッチディスプレイ、マルチタッチディスプレイなどとして構成されてもよい。任意の適切な表示デバイス及び触覚I/Oが、本発明の範囲から逸脱することなく、使用されてもよい。

【0074】

キーボード530と、例えばコンピュータマウス、タッチパッドなどのようなカーソル制御デバイス535とが、さらにバス505に接続されて、ユーザがコンピューティングシステム500とインタフェースをとることを可能にする。しかし、特定の実施形態において、物理的なキーボード及びマウスが存在しなくてもよく、ユーザは、ディスプレイ525及び／又はタッチパッド(図示略)を介してのみデバイスとインタフェースをとってもよい。入力デバイスの任意の種類及び組み合わせが、設計上の選択事項として使用されてもよい。特定の実施形態において、物理的な入力デバイス及び／又はディスプレイが存在しない。例えば、ユーザは、コンピューティングシステム500と通信する別のコンピューティングシステムを介してリモートでコンピューティングシステム500と対話してもよく、或いは、コンピューティングシステム500は自律的に動作してもよい。

【0075】

メモリ515は、プロセッサ510によって実行されると機能を提供するソフトウェア

モジュールを記憶する。該モジュールは、コンピューティングシステム500用のオペレーティングシステム540を含む。モジュールは、本明細書に記載されているプロセス又はその派生のプロセスの全て又は一部を実行するように構成されるセッション間自動化モジュール545をさらに含む。コンピューティングシステム500は、追加の機能を含む一又は複数の追加の機能モジュール550を含んでもよい。

【0076】

当業者は、「システム」が、本発明の範囲から逸脱することなく、サーバ、組込みコンピューティングシステム、パーソナルコンピュータ、コンソール、パーソナルデジタルアシスタント(PDA)、携帯電話、タブレットコンピューティングデバイス、量子コンピューティングシステム、任意の他の適切なコンピューティングデバイス、又はデバイスの組み合わせとして具現化され得ることを理解するであろう。上記の機能を「システム」によって実行されるものとして示すことは、決して本発明の範囲を限定することを意図するものではなく、本発明の多くの実施形態の一例を示すことを意図する。実際、本明細書において開示される方法、システム、及び装置は、クラウドコンピューティングシステムを含むコンピューティング技術と整合するローカライズされ分散された形式で実装されてもよい。

【0077】

本明細書に記載されているシステム機能の一部は、実装の独立性をより強調するため、モジュールとして示されていることに留意されたい。例えば、モジュールは、カスタムの超大規模集積(VLSI)回路又はゲートアレイを含むハードウェア回路、ロジックチップ、トランジスタ、又は他のディスクリートコンポーネントなどの既製の半導体として実装されてもよい。モジュールは、例えばフィールドプログラマブルゲートアレイ、プログラマブルアレイロジック、プログラマブルロジックデバイス、グラフィックスプロセッシングユニットなどのプログラマブルハードウェアデバイスにおいて実装されてもよい。

【0078】

モジュールは、様々な種類のプロセッサによる実行のため、ソフトウェアで少なくとも部分的に実装されてもよい。例えば、実行可能コードの識別されたユニットは、例えばオブジェクト、手順、又は機能として構成され得るコンピュータ命令の一又は複数の物理ブロック又は論理ブロックを含んでもよい。これにも関わらず、識別されたモジュールの実行可能ファイルは物理的に一緒に配置される必要はないが、論理的に結合されるとモジュールを含んでモジュールの上記目的を達成するような様々な場所に記憶された異種の命令を含んでもよい。さらに、モジュールは、本発明の範囲から逸脱することなく、コンピュータ読み取り可能な媒体に記憶されてもよく、コンピュータ読み取り可能な媒体は、例えば、ハードディスクドライブ、フラッシュデバイス、RAM、テープ、及び／又はデータを記憶するために使用される他のそのような非一時的なコンピュータ読み取り可能な媒体であってもよい。

【0079】

実際、実行可能コードのモジュールは、単一の命令であっても多数の命令であってもよく、異なるプログラム間で複数の異なるコードセグメントにわたり、複数のメモリデバイスにわたって分散されてもよい。同様に、動作データが、識別されて、本明細書においてモジュール内に示されてもよく、任意の適切な形式で具体化され、任意の適切な種類のデータ構造内で構成されてもよい。動作データは、単一のデータセットとしてまとめられてもよく、或いは、異なるストレージデバイスを含む異なる場所に分散されてもよく、少なくとも部分的に、単にシステム又はネットワーク上の電子信号として存在してもよい。

【0080】

従来、アテンディッド自動化では、同じセッションで実行されているRPAロボットがそのタスクを完了するまで、人間のユーザは待機する。しかし、幾つかの実施形態では、一又は複数のロボットセッションを作成し、そのセッションにおいてRPAロボットをホストし、実行する。従来のRPAシステムと異なり、ロボットがロボットセッションで自動化を実行している間、ユーザは、ユーザのコンピューティングシステムと対話する機能

10

20

30

40

50

の恩恵を受けることができる。幾つかの実施形態において、ユーザは、ロボットセッションのためのホスト自動化ウィンドウを介して、ロボットが何をしているかを監視し、ロボットと対話してもよい。幾つかの実施形態において、ユーザ及びロボットセッションは、ユーザのコンピューティングシステムによって制御されるリモートマシンで実行されていてもよい。

【0081】

しかし、幾つかの実施形態において、RPAロボットは、ユーザがIPCファシリテータを介して使用しているアプリケーションと対話しない。代わりに、RPAロボットは、RPAファシリテータ及びRPAドライバを介して、ユーザに見えない又はユーザによって使用されていないアプリケーション又は他のプロセスと対話してもよい。このようなロボットは通常、ユーザとアプリケーションとの対話を直接強化することではなく、アンアテンディッドロボットとみなされてもよい。

【0082】

特定の実施形態において、RPAロボットは、ユーザのコンピューティングシステムで実行されていて、リモートランタイムを介して（例えば、UiPath Remote Runtime（商標）を介して）リモートコンピューティングシステムを駆動していてもよい。UiPath Remote Runtime（商標）は、リモートアプリケーション又はデスクトップ（例えば、Citrix Virtual Apps and Desktops（商標）など）と、専用のUiPath（登録商標）拡張機能（例えば、Citrix（商標）用のUiPath（登録商標）拡張機能、Windows（登録商標）リモートデスクトップ用のUiPath（登録商標）拡張機能など）との間の通信を容易にするコンポーネントである。UiPath Remote Runtime（商標）は、リモートアプリケーションの対象となるUI要素に関連する情報を収集し、この情報を対応する拡張機能に送り、UI Explorer（商標）でセレクトタがネイティブに生成されるようにする。

【0083】

図6A～図6Gは、本発明の一実施形態による、ロボットセッションで実行されるRPAロボット、IPCファシリテータ、及びRPAドライバを使用して、ユーザセッションでフォームを完成させる一例を示す。図6Aでは、ユーザセッションウィンドウ600が示され、ここで、ユーザは、UI内のアプリケーションと対話でき、ロボットは、現在実行されていない。ロボットトレイアイコン610が、ユーザセッションウィンドウ600の右下部分に見える。この実施形態において、RPAドライバ640は、サブプロセスとしてIPCファシリテータ642を含む。しかし、特定の実施形態において、IPCファシリテータ642は、本発明の範囲から逸脱することなく、RPAドライバ640と通信する別個のアプリケーション又はプロセスであってもよい。そのような実施形態において、IPCファシリテータ642は、IPC、API呼び出しなどを介して、RPAドライバ640と通信してもよい。実際、特定の実施形態において、IPCは使用されなくてもよい。

【0084】

図6Bでは、ユーザは、ウェブブラウザを起動し、請求書作成ウェブページ620にアクセスする。図6Cでは、ユーザは、（例えば、ロボットトレイアイコン610をクリックすることにより）ロボットトレイ612を引き出し、クライアントセッションロボットオプション614を選択してユーザのコンピューティングシステムで実行する。実行するロボットを選択した後、図6Dに示すように、ロボットセッションのためのロボットセッションウィンドウ630が、画面上に子ウィンドウとして表示される。RPAロボットは、ロボットセッションで動作する。この実施形態において、クライアントセッションロボットのためのウィンドウ632が、ロボットセッションウィンドウ630内で自動的に起動し、ウィンドウ632は、ユーザセッションでウェブページ620のためのデータを検索するためのフォームフィラーワークフローを実行するためのボタン634を含む。

【0085】

幾つかの実施形態において、ロボットセッションウィンドウは表示されなくてもよく、ロボットは、ユーザに見えない状態で自動的に、起動し、動作し、終了してもよい。特定の実施形態において、ロボットは、そのワークフローを完了した後、そのセッションを終了してもよい。幾つかの実施形態において、ロボットトレイから起動するのではなく、ロボットトレイ612を使用せずに（例えば、ユーザがメインセッションのアプリケーションでボタンをクリックすることにより）、ロボットセッションが開始されてもよく、ロボットが起動し、動作してもよい。

【0086】

次に、図6Eでは、ユーザがボタン634をクリックした後、ロボットは、ウェブページ620におけるフォームフィールドに記入するためのデータの検索を開始する。ロボットがデータの一部を取得すると、データはIPCファシリテータ642に送られ、次に、IPCファシリテータ642は、RPAドライバ640に、データをウェブブラウザ620に入力させる。例えば、RPAドライバ640は、マウスを所与のフィールド上に動かし、データからフィールドにテキストを入力させてもよい。図6Eでは、RPAドライバ640は、Invoice Number（請求書番号）フィールドに既に記入しており、マウスをInvoice Amount（請求書金額）フィールドに移動し、そのフィールドをクリックして、数字1を入力してある。図6Eでは、キャレットは、数字1の後ろに表示されている。特定の実施形態において、RPAファシリテータ642は、RPAドライバ640にウェブページ620のフィールドに記入させる前に、ロボット632からデータの完全なセットを受け取ってもよい。

【0087】

上記のとおり、ロボットセッションで実行されるロボットとユーザセッションで実行されるIPCファシリテータ642との間の通信は、IPCプロトコルを使用して達成される。IPCプロトコルは、ネットワーク、パイプ、コンポーネントオブジェクトモデル（COM）、リモートプロシージャコール（RPC）、ソケットなどを介した通信を促進する可能性がある。適切なセッション作成メカニズム及びIPCプロトコルが、サポートされている他のオペレーティングシステムにも使用されてもよい。ロボットは、IPCプロトコルを介して、IPCファシリテータ642に、ステータス通知（例えば、ロボットが開始している、実行している、一時停止しているなどを示す）、取得したデータ、エラーメッセージ、コマンド、又は他の通信を送ってもよい。

【0088】

ロボットによって取得されたデータからの関連するテキストが、ウェブページ620が別のウィンドウによって覆われていない又は最小化されていない場合に、RPAドライバ640によって入力されるとユーザに見える。RPAドライバ640がフォームのためのデータ操作（インタラクション）を完了するまで、ユーザは、他のアプリケーションと対話して生産性を維持することができ、又は、幾つかの実施形態においてウェブページ620自体と対話することさえできるが、RPAドライバ640が、ユーザが情報を入力したのと同じフィールドについてデータを変更する場合、ユーザの入力は上書きされてもよい。

【0089】

次に、図6Fでは、ロボットがそのワークフローを完了した後、メッセージがウィンドウ632に表示される。背景に見えるように、ロボットはフォームフィールドのデータ検索を完了し、RPAドライバ640による入力後、フォームフィールドについての新しい入力がウェブページ620に見える。次に、ユーザは、ロボットセッションウィンドウ630を閉じてもよく、ロボットが、ロボットセッションウィンドウ630（及び、場合によってはロボットセッション自体）を自動的に閉じてもよく、又は、ロボットセッションウィンドウ630は開いたままであってもよい。その後、ユーザは完成したフォームを送信してもよい。図6Gを参照。

【0090】

図7A～図7Gは、本発明の一実施形態による、直接変数変更によりロボットセッションで実行されるRPAロボットを使用して、ユーザセッションでフォームを完成させる一

例を示す。図 7 A では、ユーザセッションウィンドウ 7 0 0 が示され、ここで、ユーザは、UI 内のアプリケーションと対話でき、ロボットは、現在実行されていない。ロボットトレイアイコン 7 1 0 が、ユーザセッションウィンドウ 7 0 0 の右下部分に見える

【0091】

図 7 B では、ユーザは、ウェブブラウザを起動し、請求書作成ウェブページ 7 2 0 にアクセスする。図 7 C では、ユーザは、（例えば、ロボットトレイアイコン 7 1 0 をクリックすることにより）ロボットトレイ 7 1 2 を引き出し、クライアントセッションロボットオプション 7 1 4 を選択してユーザのコンピューティングシステムで実行する。実行するロボットを選択した後、図 7 D に示すように、ロボットセッションのためのロボットセッションウィンドウ 7 3 0 が、画面上に子ウィンドウとして表示される。RPA ロボットは、ロボットセッションで動作する。この実施形態において、クライアントセッションロボットののためのウィンドウ 7 3 2 が、ロボットセッションウィンドウ 7 3 0 内で自動的に起動し、ウィンドウ 7 3 2 は、ユーザセッションでウェブページ 7 2 0 のためのフォームフィラーワークフローを実行するためのボタン 7 3 4 を含む。

【0092】

幾つかの実施形態において、ロボットセッションウィンドウは表示されなくてもよく、ロボットは、ユーザに見えない状態で自動的に、起動し、動作し、終了してもよい。特定の実施形態において、ロボットは、そのワークフローを完了した後、そのセッションを終了してもよい。幾つかの実施形態において、ロボットトレイから起動するのではなく、ロボットトレイ 7 1 2 を使用せずに（例えば、ユーザがメインセッションのアプリケーションでボタンをクリックすることにより）、ロボットセッションが開始されてもよく、ロボットが起動し、動作してもよい。

【0093】

次に、図 7 E では、ユーザがボタン 7 3 4 をクリックした後、ロボットは、ウェブページ 7 2 0 におけるフォームフィールドの変数にアクセスして対話（操作）することで、ウェブページ 7 2 0 におけるフォームフィールドへの記入を開始する。例えば、ロボットは、RAM、永久的なローカルストレージ、データベース、又は本発明の範囲から逸脱することなく任意の他の種類のストレージ又は場所に記憶されているフィールドのメモリ値を変更してもよい。特定の実施形態において、変数は、ウェブフォーム 7 2 0 に関連するウェブアプリケーションに割り当てられ、そのウェブアプリケーションによって使用されるメモリの一部として記憶されてもよい。

【0094】

ロボットによって変更された各々のメモリ値は、ウェブページ 7 2 0 がリフレッシュされるときに、ウェブページ 7 2 0 の各々のフィールドを更新させる。データ変更によりロボットによって変更された関連するテキストが、ウェブページ 7 2 0 が別のウィンドウによって覆われていない又は最小化されていない場合に、ユーザに見える。RPA ドライバ 6 4 0 がマウスを操作して A P レベルで及び／又はネイティブメカニズムによりテキストを入力する図 6 A ～図 6 G の例とは異なり、データは、そのような対話（操作）なしにウェブページ 7 2 0 のフォームフィールドに現れる。ロボットがフォームのためのデータ操作を完了するまで、ユーザは、他のアプリケーションと対話して生産性を維持することができ、又は、幾つかの実施形態においてウェブページ 7 2 0 自体と対話することさえできるが、ロボットが、ユーザが情報を入力したのと同じフィールドについてデータを変更する場合、ユーザの入力は上書きされてもよい。

【0095】

次に、図 7 F では、ロボットがそのワークフローを完了した後、メッセージがウィンドウ 7 3 2 に表示される。背景に見えるように、ロボットはフォームフィールドに関連する各々のメモリの変数についてデータ変更を完了し、ウェブページ 7 2 0 がリフレッシュした後、フォームフィールドについての新しい入力がウェブページ 7 2 0 に見える。次に、ユーザは、ロボットセッションウィンドウ 7 3 0 を閉じててもよく、ロボットが、ロボットセッションウィンドウ 7 3 0 （及び、場合によってはロボットセッション自体）を自動的

に閉じてもよく、又は、ロボットセッションウィンドウ 730 は開いたままであってもよい。その後、ユーザは完成したフォームを送信してもよい。図 7 G を参照。

【0096】

幾つかの実施形態において、ロボットセッションは子セッションであってもよく、子セッションは、オペレーティングシステムの子セッション API を介して作成されてもよい。幾つかの実施形態において、本発明の範囲から逸脱することなく、オペレーティングシステムによって提供される Windows（登録商標）ターミナルサービスの子セッション又は別の子セッション API を使用して、第 2 のセッションを作成してもよい。次に、ロボットトレイアプリケーション（例えば、UiPath（登録商標）Robot Agent Desktop など）又はロボットを起動するように構成された別のアプリケーションが、適切な引数を使用してオペレーティングシステムでプロセス作成 API を使用して、その子セッションでロボットプロセスを開始してもよい。次に、ロボットトレイアプリケーション又は他の適切なアプリケーションが、適切なプロトコル（例えば、名前付きパイプ上に構築されたプロトコルなど）を使用して、ロボットプロセスと通信してもよい。

10

【0097】

図 8 は、本発明の一実施形態による、ロボットセッションで実行されている RPA ロボット、IPC ファシリテータ、及び RPA ドライバを介してユーザセッションで実行されるプロセスの自動化のためのプロセス 800 を示すフローチャートである。該プロセスは、810 でユーザセッションウィンドウを起動することから始まる。これは、例えば、ユーザコンピューティングシステムで実行されているオペレーティングシステムに関連するメインウィンドウであってもよい。次に、820 で、ロボットセッションウィンドウが、ユーザセッションウィンドウの子ウィンドウとして起動される。幾つかの実施形態において、ロボットセッションウィンドウは、例えば、ロボットが開始される又は起動されることに応じて、起動されてもよい。次に、830 で、ロボットはロボットセッションで開始される。

20

【0098】

840 で、ロボットが実行され、IPC を介して IPC ファシリテータと通信する。IPC ファシリテータは、IPC を介してロボットから、ステータス通知、取得されたデータ、エラーメッセージ、コマンド、又は他の通信を受け付けてもよい。IPC ファシリテータは、ロボットにメッセージ（例えば、コマンド、ステータスの更新、IPC ファシリテータ又は RPA ドライバからのエラーメッセージなど）を送ってもよい。

30

【0099】

850 で、ロボットから取得されたデータを使用して、RPA ファシリテータは、RPA ドライバに、取得されたデータ及び／又はロボットからの他の通信を使用してアプリケーション操作を行うように制御する又は実行させる。例えば、RPA ドライバに、マウスの移動、フィールドのクリック、テキストの入力、ボタンのクリック、メニューのナビゲート、アプリケーション又はウィンドウの開閉などを行わせてもよい。ロボットの実行が完了した後、860 で、幾つかの実施形態では、自動的に、ロボットセッションが終了されてもよく、ロボットセッションウィンドウが閉じられてもよい。

40

【0100】

図 9 は、本発明の一実施形態による、直接変数変更によりロボットセッションで実行される RPA ロボットを介してユーザセッションで実行されるプロセスの自動化のためのプロセス 900 を示すフローチャートである。該プロセスは、910 でユーザセッションウィンドウを起動することから始まる。これは、例えば、ユーザコンピューティングシステムで実行されているオペレーティングシステムに関連するメインウィンドウであってもよい。次に、920 で、ロボットセッションウィンドウが、ユーザセッションウィンドウの子ウィンドウとして起動される。幾つかの実施形態において、ロボットセッションウィンドウは、例えば、ロボットが開始される又は起動されることに応じて、起動されてもよい。次に、930 で、ロボットはロボットセッションで開始され、940 で、ロボットは、

50

ロボットセッションからのユーザセッションで実行されているアプリケーション（例えば、ウェブページ、スプレッドシート、ERPアプリケーション、セールスアプリケーションなど）に関連するデータを操作する。例えば、ロボットは、コンピューティングシステムのメモリに記憶されている情報を作成、変更、又は削除して、ユーザセッションでアプリケーションによってアクセス、使用される共通の記憶されているデータに変更を生じさせてもよい。ロボットは、その実行を完了した後、950で、幾つかの実施形態では、自動的に、ロボットセッションが終了されてもよく、ロボットセッションウィンドウが閉じられてもよい。

【0101】

幾つかの実施形態において、複数の実行中の自動化プロセス（例えば、UI自動化プロセスなど）が、クライアントセッションにおいて分離されてもよい。これにより、例えば、これらのプロセスがメインセッションからIPCを介してオンデマンドで呼び出されることが可能であってもよい。このようにして、複数のRPAロボットプロセスが、オンデマンドで利用可能で、メインセッションアプリケーション、メインセッションRPAロボット、別のクライアントセッションRPAロボットなどから呼び出されてもよい。クライアントセッションで実行されているRPAロボットが、メインセッションからデータを収集可能であってもよい。特定の実施形態において、クライアントセッションで実行されているRPAロボットが、その自動化（自動動作）又は自動化の一部を実行する前に、メインセッションからの特定のトリガを待ってもよい。

【0102】

図10は、本発明の一実施形態による、メインセッションアプリケーションと対話する複数のクライアントセッションRPAロボットの一例を示すアーキテクチャ図1000である。図10では、メインセッション1010及びクライアントセッション1020が実行されている。クライアントセッション1020は、クライアントセッションRPAロボット1（1030）とクライアントセッションRPAロボット2（1032）を含む。

【0103】

クライアントセッションRPAロボット1030、1032は、別個のIPCファシリテータアプリケーション1040又はドライバ1050の一部である組込みRPAファシリテータプロセス若しくはサブルーチン1052と通信する。幾つかの実施形態において、ドライバ1050は、例えば、図3のドライバ340であってもよい。IPCファシリテータ1040又は1052はドライバ1050に、クライアントRPAロボット1030又は1032の各自のワークフローロジックによって指示された方法で、メインセッションアプリケーション及び／又は関連するアプリケーションオブジェクト1060（例えば、Microsoft Windows（登録商標）アプリケーションのコンポーネントオブジェクトモデル（COM）オブジェクト）と対話させる。例えば、ドライバ1050は、メインセッションアプリケーション1060において、マウスの移動、フォームデータへの記入、一又は複数のボタンのクリック、メニューの操作、これらのうちの組み合わせなどを行ってもよい。特定の実施形態において、RPAファシリテータアプリケーション1040は、メインセッションアプリケーション及び／又は関連するアプリケーションオブジェクト1060と直接通信してもよい。

【0104】

特定の実施形態において、クライアントセッションで実行されているRPAロボットは、アテンディッド自動化ロボットであってもよい。例えば、クライアントセッションで実行されているRPAロボットは、メインセッションでユーザからの入力を要求してもよい。これは、メインセッションロボット、ドライバなどを使用してIPCを介して実行されてもよい。

【0105】

図11は、本発明の一実施形態による、クライアントセッションでアテンディッド自動化RPAロボットを実行するプロセスを示すフロー図1100である。この実施形態において、1120で、ユーザは、クライアントセッションRPAロボットプロセスをIPC

10

20

30

40

50

を介して開始させるメインセッションアプリケーション1110を利用する。幾つかの実施形態において、メインセッションアプリケーション1110は、メインセッションにおけるユーザがクライアントセッションにおいてアテンディッドRPAロボットと対話することを可能にするRPAアシスタントアプリケーションであってもよい。例えば、RPAアシスタントアプリケーションのダッシュボードは、ロボットによってサポートされている現在のRPAプロセスを表示してもよく、ユーザは実行するプロセスを選択してもよい。特定の実施形態において、RPAアシスタントアプリケーションは、クライアントセッションRPAロボットが実行されている間、クライアントセッションワークフローのステップ及び／又は実行ステータスを表示してもよい。

【0106】

1120でクライアントセッションRPAロボットが開始した後、1130で、クライアントセッションRPAロボットはそのロジックの第1の部分を実行する。クライアントセッションRPAロボットは、1130でロジックの第1の部分を実行した後、1140でユーザ入力を要求する。これは、メインセッションアプリケーションと対話するIPCファシリテータ及びドライバを介して表示される、メインセッションで実行されているアプリケーションにおけるメッセージ、メインセッションで実行されているRPAロボットからのメッセージなどにより行われてもよい。次に、ユーザは要求された入力を提供し、この入力は、IPCを介してクライアントセッションRPAロボットに送られ、1150で、この入力を使用して、ロジックの第2の部分が実行される。1160でクライアントセッションRPAロボットプロセスが終了した後、メインセッションアプリケーション1110は、RPAロボットプロセスが終了した旨の通知を受け取ってもよい。

【0107】

特定の実施形態において、自動化の一部はメインセッションで実行されてもよく、自動化の他の部分はクライアントセッションで実行されてもよい。例えば、メインセッションアプリケーション又はRPAロボットが、メインセッションで特定の動作を実行し、次に、メインセッションアプリケーション又はロボットがその実行の特定のポイントに到達する、トリガを受け取る、論理条件が満たされるなどすると、クライアントセッションで実行されているRPAロボットに自動化の別の部分を実行させてもよい。クライアントセッションRPAロボットは、自動化の一部をメインセッションアプリケーション又はロボットと順次又は並行して実行してもよい。クライアントセッションRPAロボットは、メインセッションアプリケーション又はロボットに指示、要求されたデータ、実行結果などを提供してもよい。

【0108】

図12は、本発明の一実施形態による、メインセッションロボットM1と、クライアントセッションアプリケーションA1と、一対のクライアントセッションロボットC1、C2との間のマルチロボット共同ワークフローの実行を示すフロー図1200である。A1、C1、及び／又はC2は、同じクライアントセッションにあっても、異なるクライアントセッションにあってもよい。M1は、そのワークフローの実行を開始し、C1がワークフローを完了するようコール（要求）するアクティビティに到達する。これは、幾つかの実施形態において、クライアントセッションでRPAプロセスを実行するように設定されたフラグを有する「ワークフローの呼び出し」アクティビティによって実行されてもよい。M1は、C1にIPCを介してそのワークフローを実行させ、コールアクティビティは待機する。この間、幾つかの実施形態において、M1は他のタスクを実行してもよい。

【0109】

C1がワークフローを完了した後、C1はIPCを介してM1に通知し、M1は、クライアントセッションのプロセスであるA1に対して実行を開始する又はタスクを完了するようコールするアクティビティに到達するまで、実行を再開する。M1は、A1にIPCを介してタスクを遂行及び／又は実行させ、コールアクティビティ（例えば、「プロセスの呼び出し」アクティビティなど）は待機する。幾つかの実施形態において、プロセス呼び出しアクティビティは、クライアントセッションでA1を実行するように設定されたフ

10

20

30

40

50

ラグを有してもよい。この場合も、幾つかの実施形態において、M 1 は、A 1 を待機している間に他のタスクを実行してもよい。

【0110】

A 1 がそのプロセス又はタスクを完了した後、A 1 はIPCを介してM 1 に通知し、M 1 は、C 2 がワークフローを完了するようコールするアクティビティに到達するまで、実行を再開する。M 1 は、C 2 にIPCを介してそのワークフローを実行させ、コールアクティビティは待機する。この場合も、幾つかの実施形態において、M 1 は、C 2 を待機している間に他のタスクを実行してもよい。

【0111】

C 2 がそのワークフローを完了すると、M 1 は、IPCを介して通知され、C 1 にワークフローを完了するよう再度コールするアクティビティに到達するまで、実行を再開する。これは、以前に実行されたものと同じワークフローアクティビティ又は異なるワークフロー又は一連のアクティビティであってもよい。M 1 は、C 1 にIPCを介してそのワークフローを実行させ、コールアクティビティは待機する。さらにまた、幾つかの実施形態において、M 1 は、C 1 を待機している間に他のタスクを実行してもよい。C 1 がそのワークフローを完了すると、M 1 は、IPCを介して通知され、M 1 のワークフローが完了するまで実行を再開する。特定の実施形態において、C 1 は、最初に呼び出された後も実行されたままであってもよく、M 1 によって再度コールされたときに依然として実行されていてもよい。

【0112】

上記のとおり、幾つかの実施形態において、M 1 は、一又は複数の「ワークフローの呼び出し」アクティビティ、及び／又は、クライアントセッションにおけるRPAロボット（例えば、C 1 及び／又はC 2）及び／又は他のプロセス（例えば、A 1）を呼び出し得る「プロセスの呼び出し」アクティビティを含んでもよい。このようなアクティビティは、各々のRPAロボット及び／又はプロセスを開始したり、既に実行されているRPAロボット及び／又はプロセスと通信するなどしてもよい。このようなアクティビティは、RPA開発者がM 1 についてのワークフローを開発するときに作成されてもよい。特定の実施形態において、RPAロボットは、RPAファシリテータ又は他の適切な機能を介して、クライアントセッションからメインセッションにRPAロボット及び／又はプロセスを呼び出してよい。

【0113】

M 1 は、実行中、IPCを介してA 1、C 1、C 2 と通信してもよく、その逆も可能である。例えば、M 1 は、A 1、C 1、C 2 にコメントと要求を送り、M 1 はA 1、C 1、C 2 などからステータスメッセージ、結果、及びエラーメッセージを受け取るなどしてもよい。このようにして、M 1 は、クライアントセッションにおけるクライアントセッションRPAロボット及び／又は他のプロセス（例えば、アプリケーションなど）を制御するメインRPAロボットとして機能してもよい。特定の実施形態において、M 1、A 1、C 1、及び／又はC 2 は、並行して動作してもよい。

【0114】

幾つかの実施形態において、M 1 は、メインセッションで実行されているアプリケーションとの全ての対話を実行してもよく、これにより、IPCファシリテータの必要性が排除されてもよい。したがって、C 1、C 2 は、様々なデータ検索及び処理タスクを実行してもよく、M 1 は、UI対話（インタラクション）タスクを実行してもよい。特定の実施形態において、M 1、C 1、及び／又はC 2 の少なくとも一部のワークフローアクティビティが、並行して実行されてもよい。そのような一例が、図13のフロー図1300に示され、ここで、C 1 はM 1 のロジックに基づいて実行され、実行は並行して行われる。C 1 のセクションの間、M 1 とC 1 の間で通信が送られてもよい。しかし、特定の実施形態において、C 1 は、その実行を独立して完了してもよく、M 1 が終了した後にそのワークフローの実行を完了してもよい。

【0115】

幾つかの実施形態において、全てのRPAロボットアクティビティが、クライアントセッションで実行される。RPAロボットは、例えば、UiPath（登録商標）ロボットレイ、IPCファシリテータ、別のアプリケーションなどのロボットエグゼキュータによってメインセッションからコールされてもよい。RPAロボットの実行の結果及び他の通信が、IPCを介してロボットエグゼキュータ又は他のアプリケーションに提供されてもよい。

【0116】

特定の実施形態において、標準化された通信フォーマットが、クライアントセッションとメインセッションとの間のIPC通信に使用されてもよい。例えば、メインセッションプロセスとクライアントセッションプロセスがXMLデータをやり取りしてもよい。これは、通信が各々のプロセスが認識可能なフォーマットであることを確実にするのに役立つ。

10

【0117】

幾つかの実施形態において、同期の目的で、同じセッションで実行されているRPAプロセス間でIPCが使用されてもよい。例えば、メインセッションにフォアグラウンドプロセスがある状態で、複数のRPAプロセスがバックグラウンドで並行して実行されてもよい。IPCは、バックグラウンドプロセスとフォアグラウンド又はメインプロセスとの間で情報をやり取りするための同期メカニズムを提供してもよい。

【0118】

特定の実施形態において、ドライバ（例えば、図3のドライバ340）は、メインRPAワークフロープロセスにおいてロードされるであろう。子RPAプロセスとメインRPAワークフロープロセスとの間の通信が、IPCを介して行われてもよい。これにより、メインRPAワークフロープロセスがドライバを利用して、IPC情報に基づいてクライアントセッションロボットの操作を実行してもよい。

20

【0119】

幾つかの実施形態において、クライアントセッションRPAロボットは、メインセッションで開かれているアプリケーションを扱ってもよい。例えば、クライアントセッションRPAロボットは、メインセッションで開いているExcel（登録商標）スプレッドシートを扱ってもよい。これは、例えば、クライアントセッションRPAロボットとRPAファシリテータとの間のIPCを介して行われてもよい。RPAファシリテータは、クライアントセッションRPAロボットによって提供された情報に基づいて、ドライバにExcel（登録商標）スプレッドシートとの対話を実行させてもよい（例えば、スプレッドシートへのデータの入力、テーブル値の変更など）。特定の実施形態において、アクティビティは、両方のセッション間でアプリケーションのデータを自動的に取得するRPAワークフローに含まれてもよい（例えば、Microsoft Office（登録商標）アクティビティ、ウェブブラウザアクティビティなど）。このように、クライアントセッションRPAロボットは、基本的に、メインセッションで実行されているアプリケーションがクライアントセッションRPAロボット自身のセッションで実行されているかのように、メインセッションで実行されているアプリケーションを扱ってもよい。実際、幾つかの実施形態において、RPA開発者は、RPA開発者が開発しているワークフロー又はその一部がクライアントセッションで実行されることに気付かない場合もある。

30

40

【0120】

特定の実施形態において、例えば、デザイナアプリケーションを使用してRPAロボットを設計する場合、RPA開発者は、「ワークフローの呼び出し」アクティビティ又は「プロセスの呼び出し」アクティビティにクライアントセッションフラグを設定してもよい。ワークフロー呼び出しアクティビティは、RPAワークフローを呼び出す。プロセス呼び出しアクティビティは、ローカルマシンで使用可能なプロセスを実行する。このようなフラグを設定することにより、RPAワークフロー又はプロセスがメインセッションよりはむしろクライアントセッションで呼び出されてもよい。

【0121】

50

幾つかの実施形態において、クライアントセッションRPAロボットの実行は、メインセッションアプリケーション及び／又はRPAロボットからのトリガに基づいてもよい。図14は、本発明の一実施形態による、メインセッションアプリケーションのトリガに基づくクライアントセッションRPAロボットの実行を示すフロー図1400である。メインセッションアプリケーションA1（幾つかの実施形態においてRPAロボットであってもよい）は、実行を開始し、トリガを待つ。トリガが受け取られた後、A1は、IPCを介してクライアントセッションRPAロボットC1の実行を開始する。C1のセクションの間、A1とC1の間で通信が送られてもよい。次に、A1は別のトリガを待ち、C1はそのワークフローを実行する。

【0122】

図15は、本発明の一実施形態による、セッション間の自動化を実行するためのプロセス1500を示すフローチャートである。該プロセスは、1510でコンピューティングシステムのメインセッションでRPAロボットを実行することから始まる。1520で、RPAロボット又はアプリケーションが、メインセッションRPAロボットによってIPCを介して、クライアントセッションにおいてコールされる又は起動される。1530で、メッセージが、IPCを介して、クライアントセッションRPAロボット又はアプリケーションとメインセッションRPAロボットとの間で通信される。幾つかの実施形態において、メインセッションRPAロボットは、1540で、クライアントセッションRPAロボット又はアプリケーションの実行が終了するまで、その実行を一時停止する。

【0123】

1550で、クライアントセッションRPAロボット又はアプリケーションの実行の結果が、メインセッションRPAロボットによってIPCを介して受け取られる。1560で、メインセッションRPAロボットは、その結果を使用して、メインセッションRPAロボットのワークフローの少なくとも一部を完了するか、メインセッションにおいて実行されているアプリケーション若しくはアプリケーションオブジェクトと対話するか、又はその両方を行う。幾つかの実施形態において、1570で、該プロセスは、次のクライアントセッションRPAロボット又はクライアントセッションアプリケーションについて繰り返され、該プロセスは、ステップ1520に戻る。

【0124】

幾つかの実施形態において、クライアントセッションRPAロボット、クライアントセッションアプリケーション、又はその両方が、メインセッションRPAロボットのワークフローにおいて2回又はそれ以上の回数、コールされるか、起動される。特定の実施形態において、クライアントセッションアプリケーションは、メインセッションRPAロボットのワークフローのプロセス呼び出しアクティビティを介してコールされるか、起動され、或いは、クライアントセッションRPAロボットは、メインセッションRPAロボットのワークフローのワークフロー呼び出しアクティビティを介してコールされるか、起動され、或いはこれらの両方が行われる。幾つかの実施形態において、プロセス呼び出しアクティビティは、クライアントセッションアプリケーションがクライアントセッションで実行されることを示すフラグを含み、又は、ワークフロー呼び出しアクティビティは、クライアントセッションRPAロボットがクライアントセッションで実行されることを示すフラグを含み、又はこれらの両方である。特定の実施形態において、クライアントセッションアプリケーション、クライアントセッションRPAロボット、又はこれらの両方は、結果を提供した後、実行を継続する。幾つかの実施形態において、メインセッションRPAロボットとクライアントセッションRPAロボット及び／又はクライアントセッションアプリケーションとの間のIPC通信は、標準化された通信フォーマットである。

【0125】

図8～図15で実行されるプロセスステップは、本発明の一実施形態に従って、コンピュータプログラムによって実行されて、図8～図15に記載されているプロセスの少なくとも一部をプロセッサが実行するための命令を符号化してもよい。コンピュータプログラムは、非一時的なコンピュータ読み取り可能な媒体に格納されてもよい。コンピュータ読

10

20

30

40

50

み取り可能な媒体は、ハードディスクドライブ、フラッシュデバイス、RAM、テープ、及び／又はデータを記憶するために使用される他のそのような媒体又は媒体の組み合わせであってもよいが、これらに限定されない。コンピュータプログラムは、図8～図15に記載されたプロセスステップの全て又は一部を実施するようにコンピューティングシステムのプロセッサ（例えば、図5のコンピューティングシステム500のプロセッサ510など）を制御するための符号化された命令を含んでもよく、これもまた、コンピュータ読み取り可能な媒体に記憶されてもよい。

【0126】

コンピュータプログラムは、ハードウェア、ソフトウェア、又はハイブリッド実装で実装されてもよい。コンピュータプログラムは、互いに動作可能に通信し、表示する情報又は命令を渡すように設計されたモジュールで構成されてもよい。コンピュータプログラムは、汎用コンピュータ、ASIC、又は任意の他の適切なデバイスで動作するように構成されてもよい。

10

【0127】

本発明の様々な実施形態の構成要素は、本願の図面で一般的に記載され示されているように、多種多様な異なる構成で配置、設計されてもよいことは容易に理解されるであろう。したがって、添付の図面に表されている本発明の実施形態の詳細な説明は、クレームされている本発明の範囲を限定することを意図しておらず、単に本発明の選択された実施形態を表すものである。

【0128】

本明細書全体を通して説明される本発明の特徴、構造、又は特性は、一又は複数の実施形態において任意の適切な方法で組み合わせられてもよい。例えば、本明細書全体を通して「特定の実施形態」、「幾つかの実施形態」、又は類似の文言への言及は、実施形態に関連して説明される特定の特徵、構造、又は特性が本発明の少なくとも1つの実施形態に含まれることを意味する。したがって、本明細書全体を通して「特定の実施形態において」、「幾つかの実施形態において」、「他の実施形態において」という語句、又は同様の文言の出現は、必ずしも全て同じ実施形態のグループを指すとは限らず、説明された特徴、構造、又は特性は、一又は複数の実施形態において任意の適切な方法で組み合わせられてもよい。

20

【0129】

本明細書全体を通して特徴、利点、又は同様の文言への言及は、本発明で実現され得る特徴及び利点の全てが本発明の任意の単一の実施形態であるか或いはそれに含まれることを意味しないことに留意されたい。むしろ、特徴及び利点に言及する文言は、実施形態に関連して説明される特定の特徵、利点、又は特性が本発明の少なくとも1つの実施形態に含まれることを意味すると理解される。したがって、本明細書全体を通して特徴、利点、及び類似の文言の説明は、必ずしもそうではないが、同じ実施形態を指してもよい。

30

【0130】

さらに、本発明の説明された特徴、利点、及び特性は、一又は複数の実施形態において任意の適切な方法で組み合わせられてもよい。当業者は、特定の実施形態の特定の特徵又は利点のうちの一又は複数がなくても本発明を実施できることを認識するであろう。他の例において、本発明の全ての実施形態には存在しない可能性がある特定の実施形態において、追加の特徴及び利点が認識されてもよい。

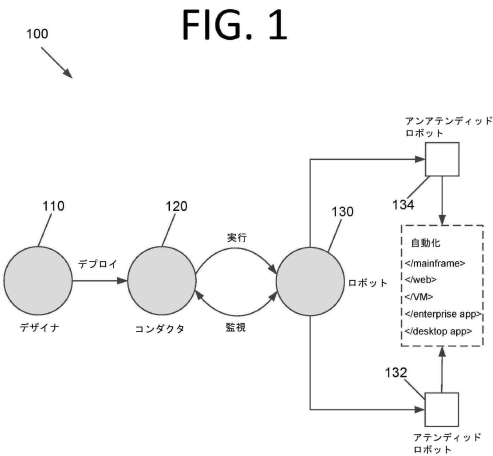
40

【0131】

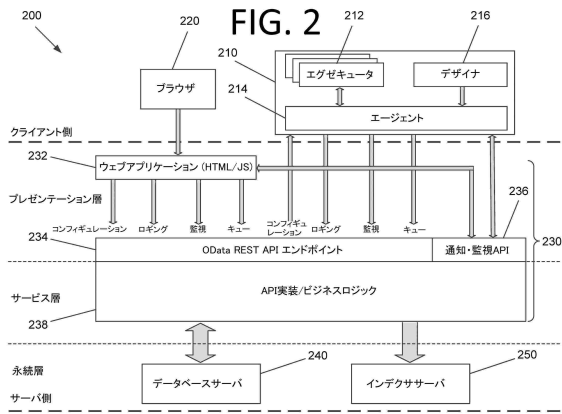
当業者は、上記の本発明が異なる順序のステップで、及び／又は開示されているものとは異なる構成のハードウェア要素で実施されてもよいことを容易に理解するであろう。したがって、本発明をこのような好ましい実施形態に基づいて説明してきたが、本発明の主旨及び範囲内にありながら、特定の修正、変形、及び代替構造が明らかであることは当業者には明らかであろう。したがって、本発明の境界及び範囲を決定するために、添付の特許請求の範囲を参照されたい。

50

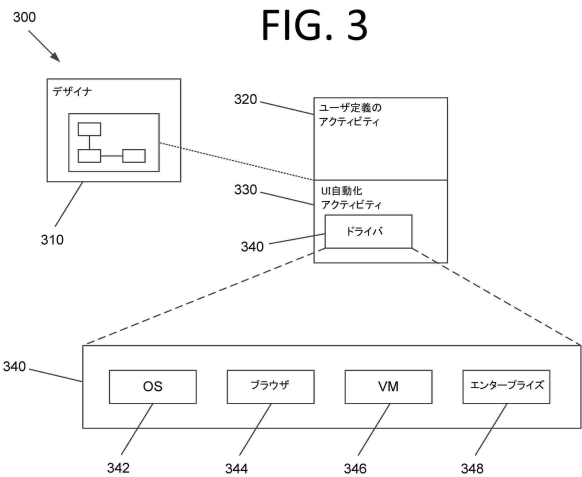
【図面】
【図 1】



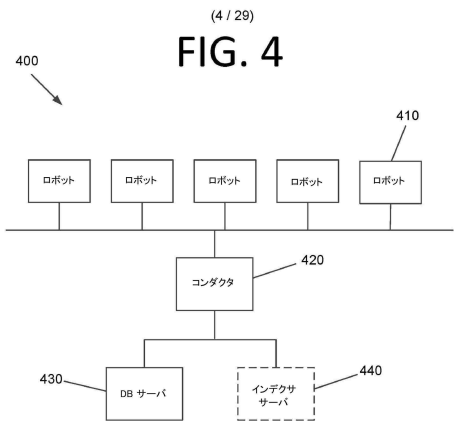
【図 2】



【図 3】



【図 4】



10

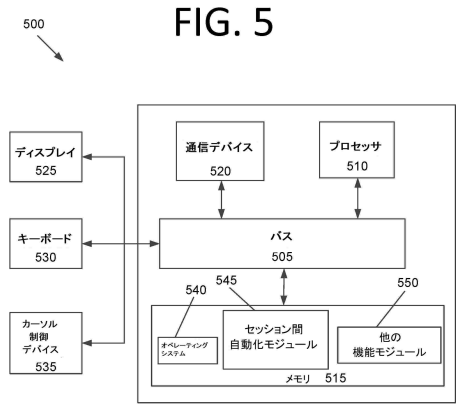
20

30

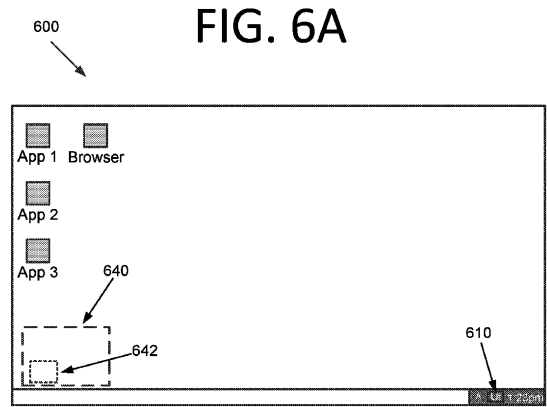
40

50

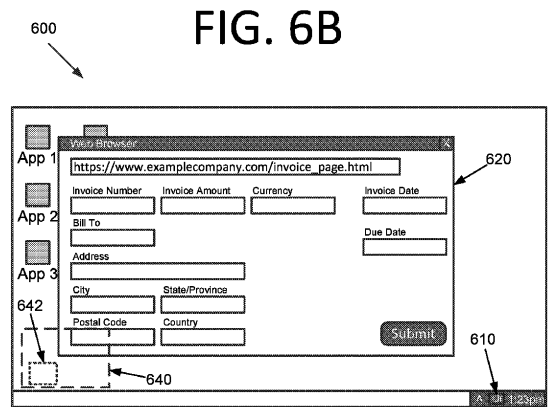
【図 5】



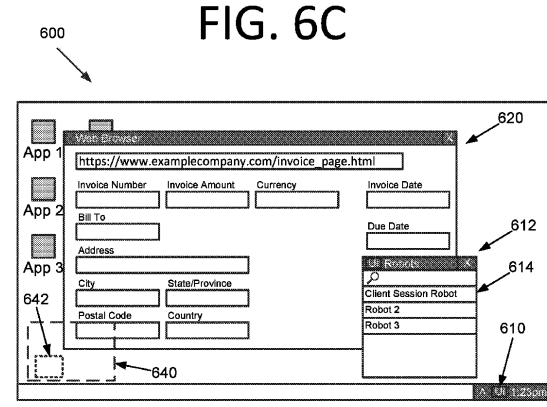
【図 6 A】



【図 6 B】



【図 6 C】



10

20

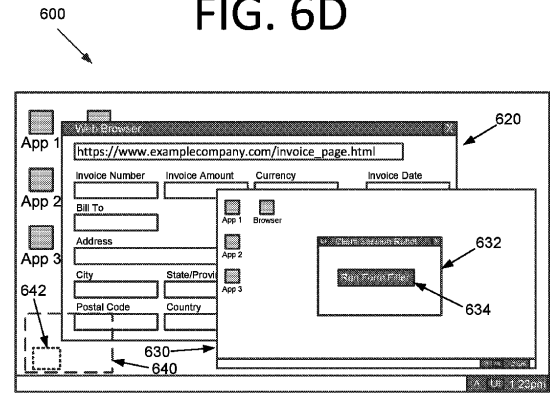
30

40

50

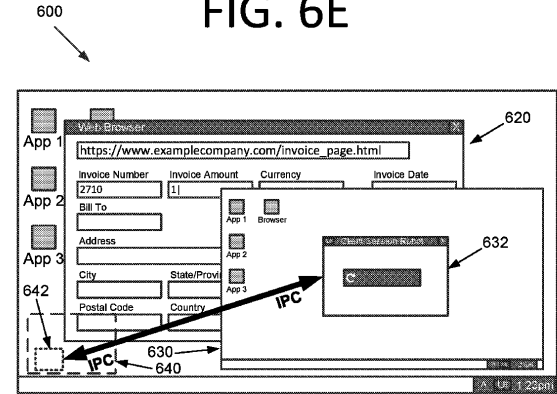
【図 6 D】

FIG. 6D



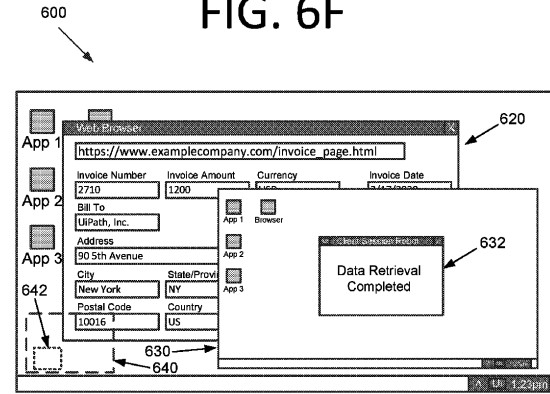
【図 6 E】

FIG. 6E



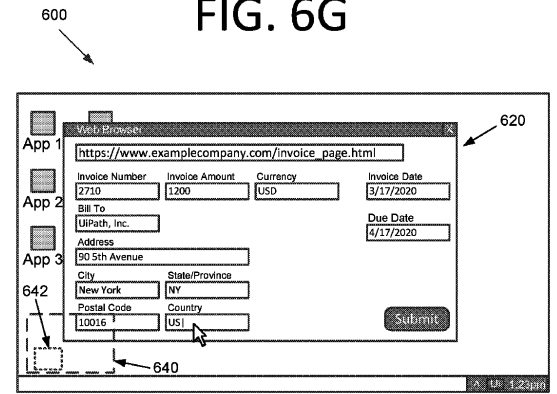
【図 6 F】

FIG. 6F



【図 6 G】

FIG. 6G



10

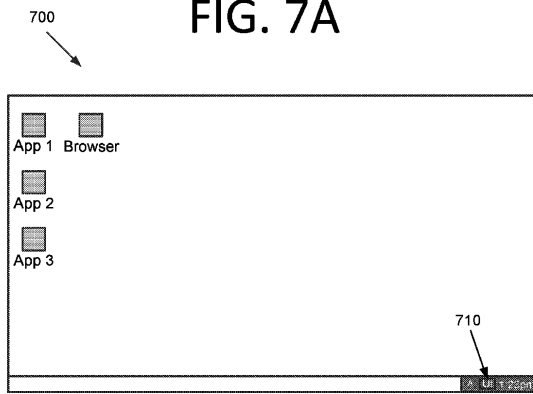
20

30

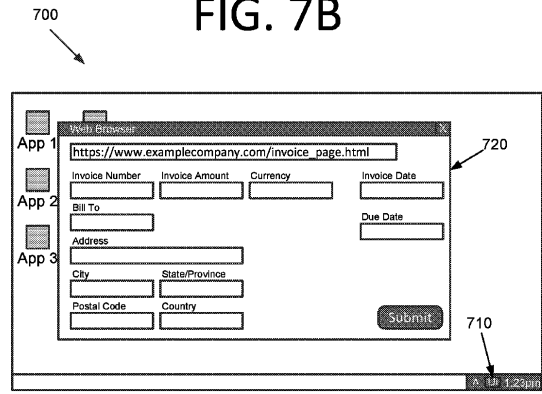
40

50

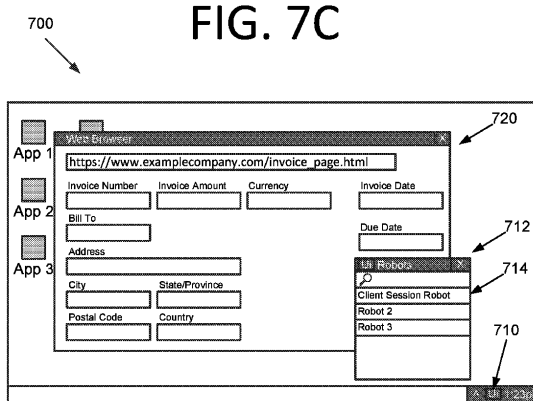
【図 7 A】



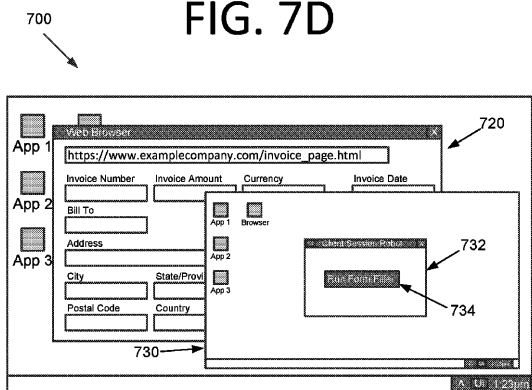
【図 7 B】



【図 7 C】



【図 7 D】



10

20

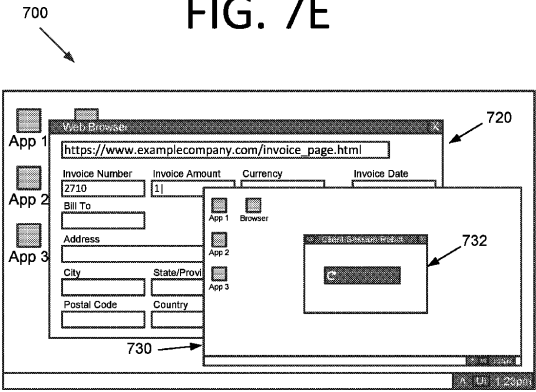
30

40

50

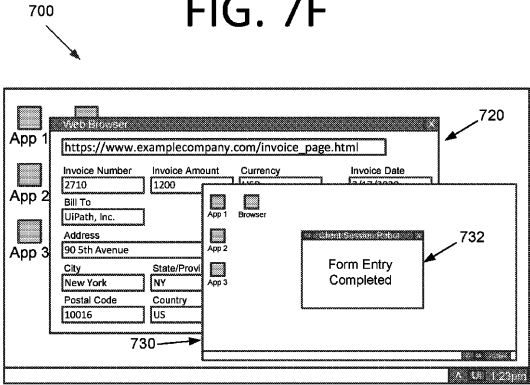
【図 7 E】

FIG. 7E



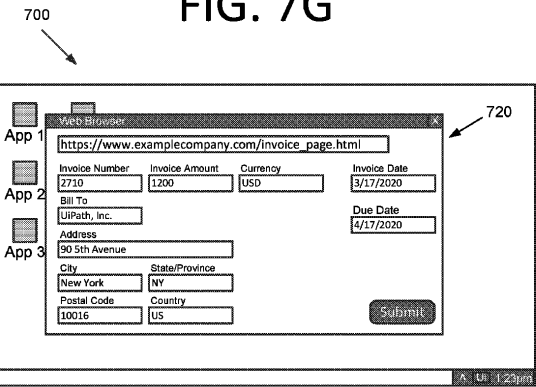
【図 7 F】

FIG. 7F



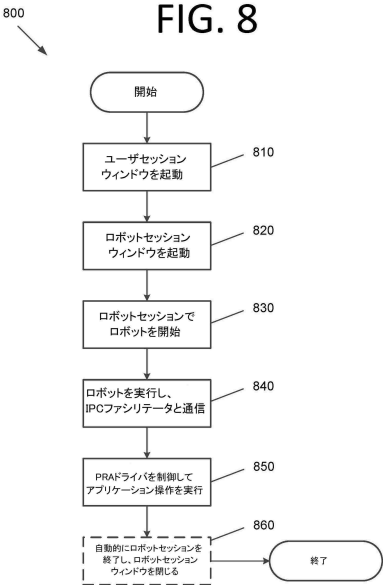
【図 7 G】

FIG. 7G



【図 8】

FIG. 8



10

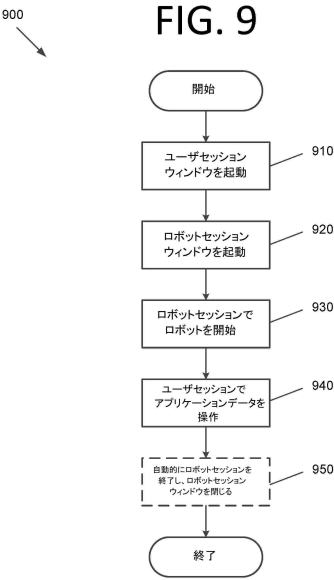
20

30

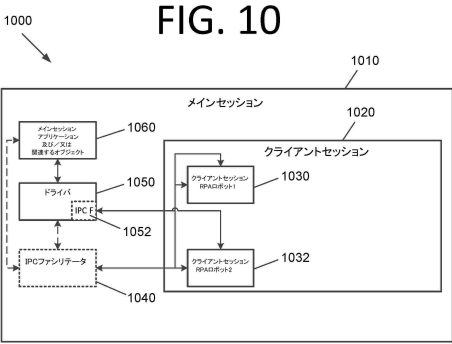
40

50

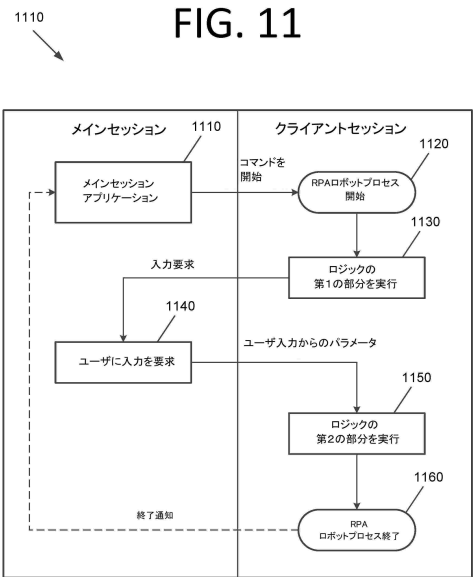
【図 9】



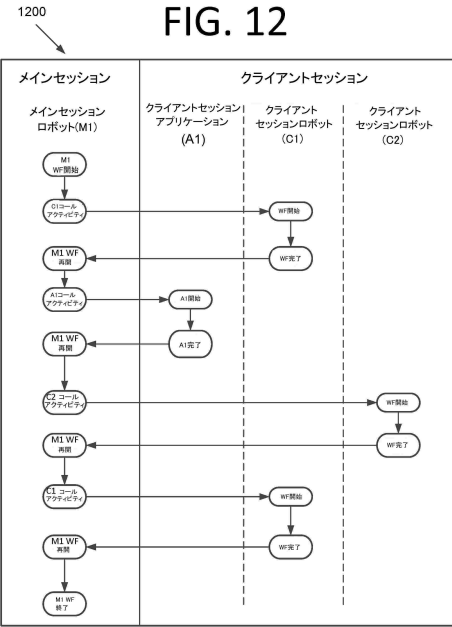
【図 10】



【図 11】



【図 12】



10

20

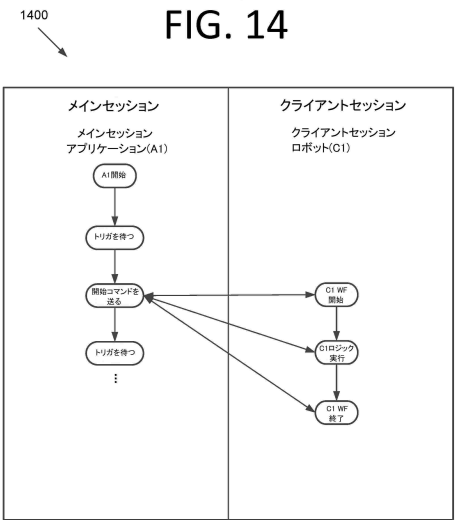
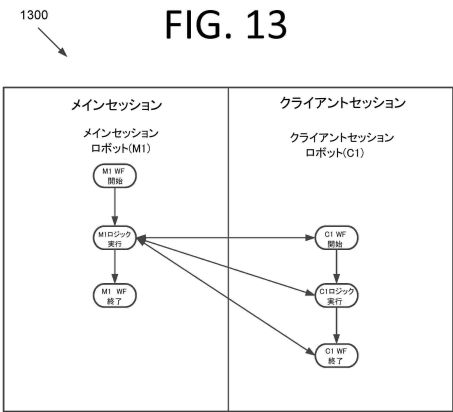
30

40

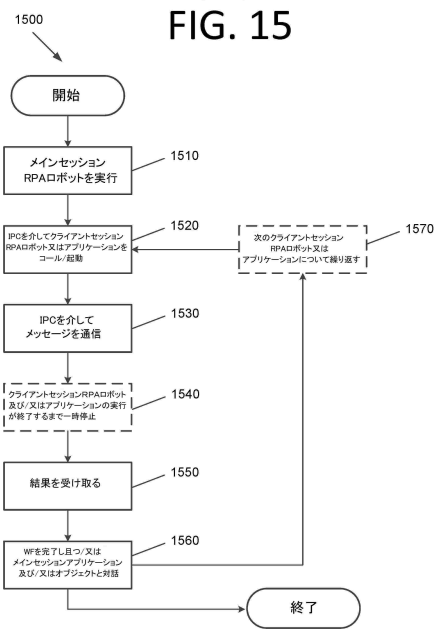
50

【図 1 3】

【図 1 4】



【図 1 5】



10

20

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)

アメリカ合衆国, 1 0 0 1 6 ニューヨーク, ニューヨーク, パーク アベニュー 9 0, トウェン
ティース フロア

審査官 円子 英紀

(56)参考文献

米国特許第1 0 6 5 4 1 6 6 (US, B 1)

米国特許出願公開第2 0 1 9 / 0 1 2 2 2 0 0 (US, A 1)

米国特許第0 9 8 1 7 9 6 7 (US, B 1)

(58)調査した分野 (Int.Cl., DB名)

G 0 6 F 9 / 4 4