

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号
特表2023-523375
(P2023-523375A)

(43)公表日 令和5年6月5日(2023.6.5)

(51)国際特許分類
G 0 6 F 3/0481(2022.01)

F I
G 0 6 F 3/0481

テーマコード (参考)
5 E 5 5 5

審査請求 未請求 予備審査請求 未請求 (全45頁)

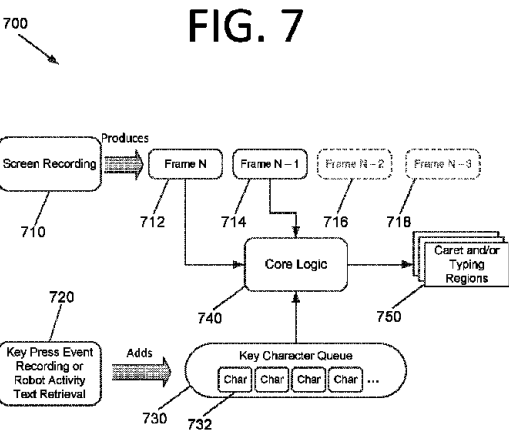
(21)出願番号	特願2020-553483(P2020-553483)	(71)出願人	520262319
(86)(22)出願日	令和2年9月21日(2020.9.21)		ユーアイパス、インコーポレイテッド
(85)翻訳文提出日	令和3年1月19日(2021.1.19)		UiPath, Inc.
(86)国際出願番号	PCT/US2020/051878		アメリカ合衆国、ニューヨーク 100
(87)国際公開番号	WO2021/221713		17, ニューヨーク、ヴァンダービルト
(87)国際公開日	令和3年11月4日(2021.11.4)		アベニュー 1, シックスティース フロア
(31)優先権主張番号	16/864,708		1 Vanderbilt Avenue
(32)優先日	令和2年5月1日(2020.5.1)		, 60th Floor, New Yo
(33)優先権主張国・地域又は機関	米国(US)		rk, NY 10017, United
			States of America
(81)指定国・地域	AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA	(74)代理人	100180781
	,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(		弁理士 安達 友和
	AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,A	(74)代理人	100182903
	T,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR		弁理士 福田 武慶
	,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,	(72)発明者	ダニエル ダインズ
	最終頁に続く		最終頁に続く

(54)【発明の名称】 ロボティックプロセスオートメーションのためのロボット実行の画面応答確認

(57)【要約】

ロボティックプロセスオートメーション（RPA）のためのロボット実行の画面応答確認が開示されている。RPAロボットが対話しているコンピュ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において、テキスト、画面変更、画像、及び／又は他の予測される視覚的アクションが発生するかが認識されてもよい。ロボットがタイピングしている位置が判断されて、一又は複数の文字、画像、ウィンドウなどが表示されている位置の現在の解像度に基づいて画面上の物理的位置が提供されてもよい。このような要素の物理的位置又はその欠如により、アプリケーション及びコンピューティングシステムが意図したとおりに応答していることを確認する目的で、ロボットがタイプしているフィールドと、関連するアプリケーションが何であるかを判断することが可能であってもよい。予測される画面変更が生じない場合に、ロボットは停止して例外をスローし、戻って意図される対話を再度試み、ワークフローを再開し、又は別の適切なアクションを実行してもよい。

【選択図】 図7



【特許請求の範囲】**【請求項 1】**

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た、ロボティックプロセスオートメーション（RPA）のための画面応答確認ロジックを含むコンピュータプログラムであって、

現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に差異が存在するか否かを判断することと、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームの間に差異が存在し、前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の差異がRPAロボットのワークフローのアクティビティに基づいて予測される場合に、

前記RPAロボットの前記ワークフローの前記アクティビティに基づいて一又は複数の予測される画面変更に対して前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の一又は複数の変更の確認を行うことと、

前記確認が失敗した場合、例外をスローする又は修正アクションを開始することと、

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、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含むことを特徴とするコンピュータプログラム。

【請求項 2】

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異がなく、前記RPAロボットの前記ワークフローの前記アクティビティに基づいて、前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異が生じないと予測される場合に、前記コンピュータプログラムは、

次のスクリーンショットフレームをフェッチすることと、

前記現在のスクリーンショットフレームを以前のスクリーンショットフレームとして使用し、前記次のスクリーンショットフレームを現在のスクリーンショットフレームとして使用して、プロセスを繰り返すことと、

を少なくとも1つのプロセッサにさらに実行させるように構成されていることを特徴とする、請求項1に記載のコンピュータプログラム。

【請求項 3】

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異がなく、前記RPAロボットの前記ワークフローの前記アクティビティに基づいて、前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異が生じると予測される場合に、前記コンピュータプログラムは、

例外をスローする又は修正アクションを開始すること

を少なくとも1つのプロセッサにさらに実行させるように構成されていることを特徴とする、請求項1に記載のコンピュータプログラム。

【請求項 4】

前記修正アクションには、前記RPAロボットが、前記RPAロボットが対話することを試みているアプリケーションのウィンドウがアクティブウィンドウであるか否かをドライバレベルでチェックすること、変更が生じることが予測されるグラフィック要素に対するマウスポインタ又はキャレットの位置をチェックすること、正しいグラフィック要素がアクティブ要素であるか否かをチェックすること、又はこれらのうちの任意の組み合わせが含まれることを特徴とする、請求項1に記載のコンピュータプログラム。

【請求項 5】

前記修正アクションには、前記RPAロボットが、正しいウィンドウをアクティブでフォーカスされたウィンドウとして設定すること、フォーカスをアクティブ要素に設定すること、マウスを動かしてマウスクリックイベントを生じさせること、又はこれらのうちの任意の組み合わせが含まれることを特徴とする、請求項1に記載のコンピュータプログラム。

10

20

30

40

50

【請求項 6】

前記例外のスローには、前記 R P A ロボットの動作の停止、画面へのメッセージの表示、前記 R P A ロボットが前記一又は複数の予測される画面変更を達成できなかったことを示すメッセージの送信、失敗した一又は複数のアクションをリストすること、又はこれらのうちの任意の組み合わせが含まれることを特徴とする、請求項 1 に記載のコンピュータプログラム。

【請求項 7】

前記コンピュータプログラムは、

時間窓の間に発生した前記 R P A ロボットによって作成されたキー押下イベントを含むキー文字キューを生成すること

を少なくとも 1 つのプロセッサにさらに実行させるように構成されていることを特徴とする、請求項 1 に記載のコンピュータプログラム。

【請求項 8】

前記 R P A ロボットはキー押下イベントとキー押下イベントの間一定時間待機するように構成されていることを特徴とする、請求項 7 に記載のコンピュータプログラム。

【請求項 9】

前記 R P A ロボットの前記ワークフローの前記アクティビティに基づいて、前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異が生じると予測され、前記 R P A ロボットの前記ワークフローの前記アクティビティに基づいて予測される画面変更が一又は複数の文字を含む場合に、前記コンピュータプログラムは、

前記現在のスクリーンショットフレームが前記以前のスクリーンショットフレームとは異なる一又は複数の位置で光学式文字認識（OCR）を実行すること

を少なくとも 1 つのプロセッサにさらに実行させるように構成され、

前記確認は、前記 OCR によって認識された文字をキー文字キュー内の文字に適合させることを含むことを特徴とする、請求項 7 に記載のコンピュータプログラム。

【請求項 10】

前記コンピュータプログラムは、

前記キー文字キューから適合する文字を削除すること

を少なくとも 1 つのプロセッサにさらに実行させるように構成されていることを特徴とする、請求項 9 に記載のコンピュータプログラム。

【請求項 11】

前記キー文字キューは、前記時間窓の間のキー押下イベントの各々について、キー押下イベントの文字と該キー押下イベントが発生した時とを含む先入れ先出し（FIFO）キューであることを特徴とする、請求項 7 に記載のコンピュータプログラム。

【請求項 12】

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異があり、前記 R P A ロボットの前記ワークフローの前記アクティビティに基づいて、前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に差異が生じると予測される場合に、前記コンピュータプログラムは、

前記現在のスクリーンショットフレームが前記以前のスクリーンショットフレームと異なる一又は複数の領域を判断することと、

判断された前記一又は複数の領域における一又は複数の結合されたメンバを抽出し、抽出された前記一又は複数の結合されたメンバに対して光学式文字認識（OCR）を実行して、一又は複数の認識された文字、一又は複数の認識された文字シーケンス、又はこれら両方と、これらの各々の位置とを生成することと、

を少なくとも 1 つのプロセッサにさらに実行させるように構成され、

前記確認は、前記一又は複数の認識された文字、前記一又は複数の認識された文字シーケンス、又はこれら両方を、前記アクティビティに基づいて表示されると予測される、前記キー文字キューと、或いは、一又は複数の文字、一又は複数の文字シーケンス、又はこ

10

20

30

40

50

れら両方と比較することを含むことを特徴とする、請求項 1 に記載のコンピュータプログラム。

【請求項 1 3】

前記コンピュータプログラムは、

ターゲットのグラフィック要素及び前記ターゲットのグラフィック要素に入力されるコンテンツを、前記 R P A ロボットの前記ワークフローの前記アクティビティに基づいて判断することと、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間における前記一又は複数の変更が前記ターゲットのグラフィック要素の位置と適合するか否かと、前記一又は複数の変更が前記ターゲットのグラフィック要素に入力されるコンテンツと適合するか否かとを確認することと、

前記位置、前記一又は複数の変更、又はこれら両方が適合しない場合、前記例外をスローする又は前記修正アクションを開始することと、

を少なくとも 1 つのプロセッサにさらに実行させるように構成されていることを特徴とする、請求項 1 に記載のコンピュータプログラム。

【請求項 1 4】

前記アクティビティには、グラフィック要素へのテキストの入力、キャレットの前記グラフィック要素への移動、コンテンツの前記グラフィック要素へのペースト、又はこれらうちの組み合わせが含まれることを特徴とする、請求項 1 に記載のコンピュータプログラム。

【請求項 1 5】

前記 R P A ロボットの前記ワークフローの前記アクティビティに基づく前記一又は複数の予測される画面変更には、フォームの送信、新しいアプリケーションウィンドウを開くこと、又は現在のアプリケーションウィンドウの変更が含まれることを特徴とする、請求項 1 に記載のコンピュータプログラム。

【請求項 1 6】

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た、ロボティックプロセスオートメーション (R P A) のための画面応答確認ロジックを含むコンピュータプログラムであって、

ターゲットのグラフィック要素及び前記ターゲットのグラフィック要素に入力されるコンテンツを、 R P A ロボットのワークフローのアクティビティに基づいて判断することと、

現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に一又は複数の差異が存在するか否かを判断することと、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に一又は複数の差異がない場合、例外をスローする又は修正アクションを開始することと、

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、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含むことを特徴とするコンピュータプログラム。

【請求項 1 7】

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に一又は複数の差異が存在する場合に、前記コンピュータプログラムは、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の変更が前記ターゲットのグラフィック要素の位置と適合するか否かと、前記変更がターゲットのグラフィック要素に入力されるコンテンツと適合するか否かとを確認することと、

前記位置、前記変更、又はこれら両方が適合しない場合、例外をスローする又は修正アクションを開始することと、

を少なくとも 1 つのプロセッサにさらに実行させるように構成されていることを特徴と

10

20

30

40

50

する、請求項 16 に記載のコンピュータプログラム。

【請求項 18】

前記コンピュータプログラムは、

コンピュータビジョナルゴリズムを適用して、前記ユーザインタフェースにおけるグラフィック要素及び関連する境界矩形を判断すること

を少なくとも 1 つのプロセッサにさらに実行させるように構成され、

判断された前記グラフィック要素は、前記ターゲットのグラフィック要素であることを特徴とする、請求項 16 に記載のコンピュータプログラム。

【請求項 19】

前記コンピュータプログラムは、

時間窓の間に発生した前記 RPA ロボットによって作成されたキー押下イベントを含むキー文字キューを生成すること

を少なくとも 1 つのプロセッサにさらに実行させるように構成されていることを特徴とする、請求項 16 に記載のコンピュータプログラム。

【請求項 20】

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間に一又は複数の差異が存在する場合に、前記コンピュータプログラムは、

前記現在のスクリーンショットフレームが前記以前のスクリーンショットフレームとは異なる一又は複数の位置で光学式文字認識（OCR）を実行すること

を少なくとも 1 つのプロセッサにさらに実行させるように構成され、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の変更がターゲットのグラフィック要素に入力されるコンテンツと適合するか否かの確認は、前記 OCR によって認識された文字をキー文字キュー内の文字に適合させることを含むことを特徴とする、請求項 19 に記載のコンピュータプログラム。

【請求項 21】

前記アクティビティには、グラフィック要素へのテキストの入力、キャレットの前記グラフィック要素への移動、コンテンツの前記グラフィック要素へのペースト、又はこれらのうちの組み合わせが含まれることを特徴とする、請求項 16 に記載のコンピュータプログラム。

【請求項 22】

前記 RPA ロボットの前記ワークフローの前記アクティビティに基づく一又は複数の予測される画面変更には、フォームの送信、新しいアプリケーションウィンドウを開くこと、又は現在のアプリケーションウィンドウの変更が含まれることを特徴とする、請求項 16 に記載のコンピュータプログラム。

【請求項 23】

ロボティックプロセスオートメーション（RPA）のための画面応答確認を実行するコンピュータ実施方法であって、

RPA ロボットによって、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に差異が存在するか否かを判断することと、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームの間に差異がなく、前記 RPA ロボットのワークフローのアクティビティに基づいて、前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の差異が生じると予測される場合に、前記 RPA ロボットによって、例外をスローする又は修正アクションを開始することと、

を含み、

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含むことを特徴とするコンピュータ実施方法。

【請求項 24】

前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームの間に差異が存在し、前記 RPA ロボットのワークフローのアクティビティに基づいて、前記

10

20

30

40

50

現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の差異が予測される場合に、前記コンピュータ実施方法は、

前記 P R A ロボットによって、前記 R P A ロボットの前記ワークフローの前記アクティビティに基づいて一又は複数の予測される画面変更に対して前記現在のスクリーンショットフレームと前記以前のスクリーンショットフレームとの間の一又は複数の変更の確認を行うことと、

前記確認が失敗した場合、前記 P R A ロボットによって、例外をスローする又は修正アクションを開始することと、

をさらに含むことを特徴とする、請求項 2 3 に記載のコンピュータ実施方法。

【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5月1日に出願された米国非仮特許出願番号 16 / 864, 708 の利益を主張する。こ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 (R P A)、より詳細には、R P A のためのロボット実行の画面 (スクリーン) 応答確認 (バリデーション) に関する。

【背景技術】

【0003】

R P A ロボットは、ユーザと同様の方法でコンピューティングシステムと対話する場合がある。例えば、ロボットはマウスを動かしたり、テキストを入力したり、ボタンをクリックしたりする場合がある。しかし、人間のユーザとは異なり、R P A ロボットは、その対話に応じてコンピューティングシステムが何をしているかを「見る」ことができず、コンピューティングシステムが意図したとおりに応答していないにもかかわらず R P A ロボットがロジックを続行する場合がある。したがって、R P A ロボットとコンピューティングシステムとの対話を監視し検証するための向上したアプローチが有益であり得る。

【発明の概要】

【0004】

本発明の特定の実施形態は、現在の R P A 技術によってまだ十分に識別、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、R P A のためのロボット実行の画面応答確認に関する。

【0005】

一実施形態において、R P A のための画面応答確認ロジックを含むコンピュータプログラムが、非一時的なコンピュータ読み取り可能な媒体に具現化される。該プログラムは、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に差異が存在するか否かを判断することを少なくとも 1 つのプロセッサに実行させるように構成されている。現在のスクリーンショットフレームと以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含む。現在のスクリーンショットフレームと以前のスクリーンショットフレームの間に差異が存在し、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間の差異が R P A ロボットのワークフローのアクティビティに基づいて予測される場合に、該プログラムは、R P A ロボットのワークフローのアクティビティに基づいて一又は複数の予測される画面変更に対して現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間の一又は複数の変更の確認を行うことと、確認が失敗した場合、例外をスローする又は修正アクションを開始することと、を少なくとも 1 つのプロセッサにさらに実行させるように構成されている。

【0006】

別の一実施形態において、RPAのための画面応答確認ロジックを含むコンピュータプログラムが、非一時的なコンピュータ読み取り可能な媒体に具現化される。該プログラムは、ターゲットのグラフィック要素及びターゲットのグラフィック要素に入力されるコンテンツを、RPAロボットのワークフローのアクティビティに基づいて判断することと、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に一又は複数の差異が存在するか否かを判断することと、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。現在のスクリーンショットフレームと以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含む。現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に一又は複数の差異がない場合、該プログラムは、例外をスローする又は修正アクションを開始することを少なくとも1つのプロセッサにさらに実行させるように構成されている。

10

【0007】

さらに別の一実施形態において、RPAのための画面応答確認を実行するコンピュータ実施方法は、RPAロボットによって、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に差異が存在するか否かを判断することを含む。現在のスクリーンショットフレームと以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含む。該コンピュータ実施方法は、現在のスクリーンショットフレームと以前のスクリーンショットフレームの間に差異がなく、RPAロボットのワークフローのアクティビティに基づいて、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間の差異が生じると予測される場合に、RPAロボットによって、例外をスローする又は修正アクションを開始することを含む。

20

【図面の簡単な説明】

【0008】

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。

【0009】

【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。

30

【0010】

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

【0011】

【図3】本発明の一実施形態による、デザイナー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。

【0012】

【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。

【0013】

【図5】本発明の一実施形態による、RPAのためのロボット実行の画面応答確認を実行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。

40

【0014】

【図6】図6A及び図6Bは、本発明の一実施形態による、RPAのためのロボット実行の画面応答確認のためのプロセスを示すフローチャートである。

【0015】

【図7】本発明の一実施形態による、RPAのためのロボット実行の画面応答確認のためのプロセスを示すアーキテクチャ図である。

【0016】

【図8A】本発明の一実施形態による、それぞれフレームN-1及びフレームNの同窓生

50

寄付ウィンドウを含むスクリーンショットの一部を示す。

【0017】

【図8B】本発明の一実施形態による、四角形グリッドに分割されたスクリーンショットN-1を示す。

【0018】

【図8C】本発明の一実施形態による、四角形グリッドに分割されたスクリーンショットNを示す。

【0019】

【図8D】本発明の一実施形態による、スクリーンショットN-1からの変更を含む強調表示された行を有するスクリーンショットNを示す。

【0020】

【図8E】本発明の一実施形態による、スクリーンショットN-1からの変更を含む強調表示された行における強調表示されたセルを有するスクリーンショットNを示す。

【0021】

【図8F】本発明の一実施形態による、スクリーンショットN-1からの変更を含む強調表示された領域を有するスクリーンショットNを示す。

【0022】

【図9】本発明の一実施形態による、変更に対応するために、解像度の変更を確認し、接続されているディスプレイの範囲の変更を確認し、キャレット追跡ビデオロジックを設定する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、ビデオレコーダのプロセスを示すフローチャートである。

【0024】

【図11】本発明の一実施形態による、キャレット追跡を実行するプロセスを示すフローチャートである。

【0025】

【図12】図12A～図12Dは、本発明の一実施形態による、RPAのためのロボット実行の画面応答確認を実行するプロセスを示すフローチャートである。

【0026】

【図13】図13A～図13Cは、本発明の一実施形態による、ペーストされたテキストの追跡及び確認を実行するプロセスを示すフローチャートである。

【0027】

【図14A】本発明の一実施形態による、 32×32 領域に結合された4つの 16×16 ブロックに含まれる文字「E」のバイナリ行列の一例を示す。

【0028】

【図14B】本発明の一実施形態による、図14Aの領域のバイナリ行列に含まれるメンバ（文字「E」）のトリミングされた 7×9 のバイナリ行列を示す。

【0029】

【図15】本発明の一実施形態による、CV及びキーキューを使用してアクティブ要素を判断しロボットアクションを確認するプロセスを示すフローチャートである。

【0030】

【図16】本発明の一実施形態による、RPAのためのロボット実行の画面応答確認を実行するように構成されたシステムを示すアーキテクチャ図である。

【0031】

【図17】本発明の一実施形態による、RPAのためのロボット実行の画面応答確認を実行するプロセスを示すフローチャートである。

【0032】

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。

10

20

30

40

50

【発明を実施するための形態】

【0033】

幾つかの実施形態は、RPAのためのロボット実行の画面応答確認に関する。幾つかの実施形態では、RPAロボットが対話しているコンピュ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において、テキスト、画面変更、画像、及び／又は他の予測される視覚的アクションが発生するか否かを認識する。そのような実施形態は、コンピューティングシステムに関連する画面上のどこにロボットがタイプしているかを探し、一又は複数の文字（キャラクタ）、画像、ウィンドウなどが表示されている位置の（例えば、座標などにおける）現在の解像度に基づいて画面上の物理的位置を提供してもよい。これらの要素の物理的な位置により、アプリケーション及びコンピューティングシステムが意図したとおりに応答していることを確認する目的で、ロボットが入力しているフィールドと、関連するアプリケーションが何であるかとを判断することが可能であってもよい。この分析から、文字、画像、ウィンドウなどが表示されていないこと、又はこれらの要素が間違った位置に表示されていること、又は間違ったタイプを含むことが判断される可能性があり、ロボットは停止して例外をスローし、戻って意図される対話を再度試み、ワークフローを再開し、又は、コンピューティングシステム上のアプリケーションがワークフローの機能を実装していない場合には、本発明の範囲から逸脱することなく任意の他の適切なアクションを実行してもよい。

10

【0034】

幾つかの実施形態は、継続的又は定期的に、現在のスクリーンショットを以前のスクリーンショットと比較する、或いは、現在のスクリーンショットの一部を以前のスクリーンショットの一部（例えば、可視アプリケーションウィンドウに関連するスクリーンショットの一部）と比較するフィードバックループプロセスで実装されて、変更が発生したか否かと、発生した場合はその変更内容とを特定する。ウィンドウに関連するスクリーンショットの一部を比較する特定の実施形態において、ロボットがユーザと一緒に動作するアテンディッドロボットである場合、アルゴリズムは、アルゴリズムがスクリーンショットの現在の部分がスクリーンショットの以前の部分にマッピングされていることをアルゴリズムが認識するように、ウィンドウの位置、ウィンドウのサイズ、ウィンドウの形状、ウィンドウのズーム変数、又はこれらのうちの組み合わせなどの変更に対応してもよい。

20

【0035】

タイプされたテキストの確認の場合、画面上で視覚的な変更が発生した位置を特定し、変更が発生した位置で光学式文字認識（OCR）を実行してもよい。次に、OCRの結果をキーキューの内容と比較して（例えば、RPAロボットによって生成されたキー押下イベントによって判断されるとして）、適合が存在するか否かを判断してもよい。変更が発生した位置は、現在のスクリーンショットからのピクセルのボックスを以前のスクリーンショットの同じ位置におけるピクセルのボックスと比較することで判断されてもよい。

30

【0036】

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してもよい。

40

【0037】

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11

50

0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。

【0038】

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。

10

【0039】

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（モニタリング）、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。

20

【0040】

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（操作要）ロボット132、アンアテンディッド（操作不要）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、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。

30

【0041】

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で実行してもよい。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）、Sales Force（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

40

50

ーションを自動化してもよい。

【0042】

コンダクタ120は、プロビジョニング、デプロイメント、バージョニング、コンフィギュレーション、キューイング、監視、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）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。バージョニングには、幾つかの実施形態において、ある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、Elastic Search（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。

10

【0043】

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。

20

【0044】

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。

30

【0045】

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）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。

40

【0046】

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。

50

【0047】

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。

10

【0048】

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。

20

【0049】

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。

30

【0050】

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。

40

【0051】

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を送り）、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の間の通信は、常にエージェント214によっ

50

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 チャンネルを開き、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。

【0052】

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）（OData）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、OData 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。

【0053】

ウェブアプリケーション 232 に加えて、コンダクタ 230 には、OData 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。

【0054】

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。

【0055】

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 API 236 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 21

4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。

【0056】

永続層は、この実施形態では一対のサーバ、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。

【0057】

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。

【0058】

図3は、本発明の一実施形態による、デザイナー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、デザイナー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、複数のアンカーを使用して要素を識別し、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、デザイナー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。

【0059】

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。

【0060】

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。

【００６１】

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。

10

【００６２】

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。

20

【００６３】

図５は、本発明の一実施形態による、ＲＰＡのためのロボット実行の画面応答確認を実行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。

30

【００６４】

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。

40

【００６５】

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５

50

20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。

10

20

30

40

50

【0066】

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。

【0067】

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対話してもよい。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。

【0068】

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RPAロボット確認モジュール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。

【0069】

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、組込みコンピュータ、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。

10

【0070】

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実装されてもよい。

【0071】

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よい。

20

30

【0072】

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。

【0073】

図6A及び図6Bは、本発明の一実施形態による、RPAのためのロボット実行の画面応答確認のためのプロセス600を示すフローチャートである。キー押下記録602及びスクリーン記録604（例えば、スクリーンショットの記録、サーバによる行われる可能性がある後の処理のために長い一連のスクリーンショットのスクリーンビデオ記録のリアルタイムでの比較など）が実行されて、ロボットによって「押された」キー（例えば、ロボットによってコンピューティングシステムで発生したキー押下イベント、アプリケーションレベルのAPI呼び出しなど）と、グラフィック変更が発生した画面上の位置とを判断する。しかし、幾つかの実施形態において、ロボットが入力しようとしていたテキストがロボットのアクティビティから判断され得るので、キー押下記録は実行されない。本明細書では、ロボットによって入力された文字を「キー押下」といい、ステップ602を「

40

50

キー押下記録」というが、R P A ロボットは実際には物理キーボードでタイプしない。むしろ、R P A ロボットは、ユーザがキーボードで物理的にタイプすることに起因してアプリケーション、オペレーティングシステム、又はその両方によって生成されるであろうイベントと同じ又は類似のシステム及び／又はアプリケーションレベルのイベントを生成する。幾つかのリアルタイム又はほぼリアルタイムの実施形態において、フレーム間差分がオンザフライで計算され得るので、ビデオデータは記録、記憶されなくてもよい。

【0074】

キー押下記録602は、オペレーティングシステムからのキー押下イベントの監視によって実行されてもよい。しかし、本発明の範囲から逸脱することなく、任意のプログラミング言語及び任意のオペレーティングシステム（例えば、モバイル、P C、M a c など）のキー押下情報を提供する任意のキー押下イベント又は機能が使用されてもよい。

10

【0075】

キー押下イベントは、どの文字が押されたキーに関連付けられているかに関する情報（例えば、文字「a」、数字「2」、「%」記号など）、キー押下イベントが発生した時などを含んでもよい。キーの文字のキュー（例えば、先入れ先出し（F I F O）キューなど）が、ロボットによってキーが押された時と対応する文字が画面に表示される時との間の遅延を考慮して、時間窓（例えば、500ミリ秒（m s）、1秒など）の間格納されてもよい。時間窓は通常、ロボットがキーを押す時とキーが画面に表示される時の間の通常の時間遅延よりも長い（例えば、50ミリ秒の文字表示遅延と500ミリ秒のバッファウィンドウなど）。

20

【0076】

近頃の高速コンピューティングハードウェアによるロボットの動作速度に起因してよく起こることであるが、キューは、画面に一度に表示される複数の文字をキャプチャする目的にも役立つ。例えば、ロボットが「a b c」を素早く連続して（例えば、1ミリ秒以内に）押しても、1秒あたり25フレームのみキャプチャされる場合（即ち、40ミリ秒毎に1フレーム）、次のスクリーンショットで、テキスト「a b c」が一度に表示されてもよい。キューに「a」、「b」、「c」を含めることにより、テキスト認識がこれらの文字及び／又はシーケンスを検出したときに、アルゴリズムがこれらの文字及び／又はこれらのシーケンスをそれぞれ検索してもよい。例えば、幾つかの実施形態において、ロボットによるタイプ「a b c」、「a b」が次のフレームに現れる場合、キー押下キュー内の順序は、画面上に表示されるものと同じであると想定してもよい。

30

【0077】

特定の実施形態において、ロボットは、キーを押す間に遅延を提供してもよい。例えば、ロボットはあるキー押下イベントを作成し、5ミリ秒待機し、別のキー押下イベントを作成するなどしてもよい。これにより、あるスクリーンショットから別のスクリーンショットへの変更がそれ程極端でなく人間の対話速度に近くなり得るので、ロボットはコンピューティングシステムとよりユーザのような対話を行うことができ、変更後の確認においても役立つ。追加的に又は代替的に、ロボットは、テキストが入力されてから送信ボタン、電子メール送信ボタンなどを「押す」までの十分な時間を待って、変更が送信後に消える前に確認のためにスクリーンショットに表示されるようにする。

40

【0078】

スクリーン記録604は、本発明の範囲から逸脱することなく、任意の適切なハードウェア、ソフトウェア、又はそれらの任意の組み合わせによって実行されてもよい。例えば、ビデオ記録は、外部ビデオカメラ、内部グラフィックカード、（接続されたコンピューティングシステムを介して又は介さずに）ディスプレイからのビデオストリームを監視するリモートコンピューティングシステムなどによって実行されてもよい。記録されたスクリーンショットは、ピクセルが完全であるか否かに関係なく、任意の所望の形式（J P E G、B M P、P N Gなど）で任意の場所に記憶されてもよい。

【0079】

一実装において、スクリーンショットは、F o r m a t 1 6 b p p R g b 5 5 5形式の

50

16ビット色深度でBitmapDataとして記憶される。スクリーンショットを非可逆形式に圧縮することにより、一又は複数のピクセルを変更すると、（例えば、JPEG形式などにおける）圧縮アーティファクトの存在により、一部の形式で画像全体に伝播されるカラーマップが変更されたり、色深度に影響したり、全体の詳細が減少／シフトしたり、画像のピクセルに影響したりするため、精度が低下する場合がある。これに対応するために、許容範囲（トレランス）が使用されてもよい。スクリーン記録604は、特定のフレームレートでフレームを生成し、フレームレートは、コンピューティングシステムの現在の処理負荷に応じて変えてもよい（例えば、毎秒30フレームなど）。

【0080】

次に、606で、現在のスクリーンショットフレームと直前のスクリーンショットフレームを互いに比較して、それらの間に差分が存在する否か（例えば、それらにおける対応するピクセルの少なくとも一部が互いに異なるなど）を判断する。ロボットがワークフローで実行している現在のアクションに基づいて、対応する画面の変更が予測される場合もあるし、予測されない場合もある。608で画面の変更が発生せず、610でワークフローの実行に基づいて画面の変更が予測された場合、612で、プロセスは次のスクリーンショットに進み、606で、該次のスクリーンショットと以前の現在のスクリーンショットとの比較に戻る。

【0081】

608で画面の変更が発生せず610で変更が予測された場合、又は、608で画面の変更が発生したが614で変更が予測されなかった場合、616で、システムが例外をスローする、或いは、ロボットが修正措置をとるように指示される。例外がスローされた場合、ロボットの動作が停止されてもよく、ロボットが所望のアクションを達成できなかったことを示すメッセージが、画面に表示されるか、或いは、コンピューティングシステムのユーザに（例えば、テキスト又は電子メールを介して）送られる。特定の実施形態において、失敗したアクションがユーザに提供されてもよく、そうすると、ユーザは、問題のトラブルシューティングを試みることができ、又は、ユーザは、RPA開発者によって修正されるべき問題を報告できる。

【0082】

別のオプションは、例外のスローよりもむしろ、ロボットに修正措置をとらせることである。ロボットは、ワークフローのどのステップが失敗したかを通知され、何らかの修正措置をとって、再度該ステップを試みてもよい。これには、ロボットが、ロボットが対話を試みているアプリケーションのウィンドウがアクティブウィンドウであるか否かをドライバレベルでチェックすること、関連するグラフィック要素に対するマウスポインタ又はキャレットの位置をチェックすること、正しいグラフィック要素がアクティブ要素であるか否かをチェックすること、これらのうちの任意の組み合わせなどが含まれてもよいが、これらに限定されない。ロボットは、正しいウィンドウをアクティブでフォーカスされたウィンドウに設定する、フォーカスを所望のアクティブ要素に設定する、マウスを動かしてクリックするなどして、その後、再度該ステップを試みてもよい。修正措置が成功した場合、プロセスはステップ612に戻ってもよい。しかし、修正措置が失敗した場合、例外がスローされてもよい。

【0083】

608で画面の変更が発生し614でワークフローに基づいて変更が予測された場合、プロセスは図6Bに進む。618で、画面が変更された領域が判断され、（もしあれば）領域内の文字が判断される。特定の実施形態において、これには、所望のウィンドウが表示された又は消えたことを判断すること、画像が表示された又は変更されたことを判断すること、グラフィックコンポーネントが表示された又は変更されたことを判断すること（例えば、テキストボックス、テキストフィールド、テーブルなど）、又はこれらのうちの組み合わせなどが含まれてもよい。このような変更は、例えば、CVを使用して判断されてもよい。

【0084】

特定の実施形態において、精度をさらに高めるために、複数のフレームが使用されてもよい。例えば、文字がタイプされる時と文字が画面に表示される時の間に様々な非同期がある場合（例えば、30ミリ秒から、42ミリ秒まで、24ミリ秒まで変化するなど）、複数のフレームを使用すると、タイプされたテキストの識別に役立つ場合がある。次に、視覚的な変更の位置が分離され、変更が発生した位置に対してアルゴリズムが実行されて文字を認識してもよい。このアルゴリズムは、OCR、特定のフォントの文字のブル配列マップに対するピクセル領域の比較などを使用してもよい。幾つかの実施形態において、文字認識は、発生した変更が分離され分析された比較的小さな領域に対してのみ実行され、残りの領域は破棄される。これは、画面全体（例えば、3840×2160ピクセルの解像度など）に対してOCRを実行すると、文字が表示される速度にコンピューティングシステムが遅れずについていくには計算コストが高過ぎる可能性があるコンピューティングシステムで、アルゴリズムをリアルタイムで実行できるようにするのに役立つ。しかし、十分な処理能力を有するコンピューティングシステムの場合、最初に変更が発生しなかった領域を解析せずに、画面全体を分析してもよい。

10

【0085】

上記のとおり、特定の実施形態において、ビデオフレーム比較計算は、ピクセル完全ではなく、許容範囲を使用する。色の強度、明るさ、及び／又は他の要因が、それらが特定の許容範囲内にある場合、同じとみなされてもよい。例えば、ピクセルは、それらの値の一又は複数の変更が特定の数、特定のパーセンテージなどに満たない場合、ピクセルは同じとみなされてもよい。赤、緑、青、及び明るさの変更が5未満、1%未満などは、同じとみなされてもよい。特定の実施形態において、これらの変数の一又は複数の異なる許容範囲を有してもよい。例えば、真のピクセルの変更を示すために、明るさの変更を色の変更よりも大きくしたり小さくしたりする必要があってもよい。ファジー画像マッチングが、類似点／相違点を識別するために特定の実施形態で実行されてもよい。

20

【0086】

幾つかの実施形態において、ファジー画像マッチングは、明るさ、画像テンプレート、エッジ比較、二値化、ダウンスケール及びビット削減、膨張を考慮して、カーネルブラー、それらの組み合わせなどを適用して、適合をより正確に識別する。RGB値に許容範囲を適用するピクセル間RGBマッチングが使用されて、完全に同じではない近い値が適合として識別されてもよい。ビット深度及び／又はカラースケールが低減されてもよく、ピクセル間のRGB又はグレースケールマッチングが適用されてもよい。画像からエッジが検出されて比較されてもよい。二値化は画像に適用されてもよく（例えば、二値閾値、大津閾値、適応閾値など）、ピクセル間のマッチングが二値画像に適用されてもよい。画像のスケールが縮小され、ピクセル間のマッチングが実行されてもよい。画像の膨張が実行され、次に、ピクセル間のマッチングが適用されてもよい。キーポイントが画像から抽出されて（例えば、最大安定極値領域（MSER）記述子など）、抽出されたキーポイントが、特徴マッチャー（例えば、ブルートフォースマッチング、k最近傍（kNN）マッチングなど）を使用して比較されてもよい。

30

【0087】

許容範囲ベースの計算が有益である可能性がある様々な理由がある。例えば、フレームがキャプチャされた後に画像が圧縮される場合、非可逆圧縮はピクセル値に影響を与える可能性があるため、許容範囲を計算に含める必要がある。また、元のビジュアルソースが、非可逆圧縮を使用してキャプチャする前に圧縮される場合がある（例えば、仮想コンピューティングシステムがエミュレータを介して起動され、エミュレータが仮想コンピュータの画面コンテンツを圧縮する場合）。これは、画像がリモートマシン（例えば、サーバなど）からローカルコンピューティングシステムにブロードキャストされるために発生する場合がある。

40

【0088】

618で、変更が発生した且つ／又は他のグラフィック変更が発生した画面領域の文字が識別されると、文字は、キー押下イベントに対応する格納された文字のキューと比較さ

50

れる。適合が発見された場合、620で、適合位置の画面座標が抽出され、検出された他のグラフィック変更の画面座標も抽出される。場合によっては、文字認識アルゴリズムが、画面上の文字を実際の文字のように認識できないことがある。例えば、OCRアルゴリズムは、画面上の文字「O」を数字「0」として認識する場合がある。その場合、幾つかの実施形態において、アルゴリズムは、画面上のキャレットの位置を追跡する。これは、画像検出（例えば、CVなど）などを使用して、様々なキャレットの形状の画像パターンを画面と比較することで判断されてもよい。幾つかの実施形態において、アルゴリズムは、キャレットが点滅する場合、キャレットの点滅を考慮してもよい。

【0089】

特定の実施形態において、ファジーマッチングを使用して、OCRの結果をキュー内の文字と比較してもよい。ファジーマッチングロジックは、文字「O」が数字「0」に類似して見えることを認識して、これらの文字を適合として識別してもよい。キューに他の類似する文字がない場合は、適合が確認されてもよい。

【0090】

特定の実施形態において、キャレット追跡が実行されてもよい。画像の変更された領域の分析が、キャレットに対応する可能性のある候補のリストを作成するために、実行されてもよい（例えば、候補は、細い縦線又は同様のものとして表示される）。確認（バリデーション）が、候補が時間をかけて点滅していることを識別するために実行されてもよく、その後、真のキャレットが識別されてもよい。テキスト入力可能なグラフィック要素（例えば、テキストボックス、ワードプロセッサ文書、テキストフィールドなど）内にキャレットが表示されることを検証するために、さらなる確認が実行されてもよい。

【0091】

画面の他の位置で変更が発生しなかった場合、又は、他の変更が欠落しているもの以外のキュー内の文字と適合する場合、アルゴリズムは、これが唯一の未識別の変更であるため、欠落している文字である可能性があると推測する。次に、アルゴリズムは、認識された文字「O」が実際には文字キュー内の識別されていない「0」であると推測し、620で、適合位置の画面座標を抽出してもよい。これにより、アルゴリズムの精度が向上する可能性がある。

【0092】

幾つかの実施形態において、文字又は文字シーケンスが画面上で発見されて一意に識別された場合、文字はキューから削除されてもよく、所定の時間が経過した後（例えば、500ミリ秒、1秒など）、所定のサイズ（例えば、20文字など）のキューに基づいてキューの最後で文字を取り出してもよい。時間窓外にある文字を削除するために、キューは、タイプされた文字及びタイムスタンプを有する変数を格納してもよい。アルゴリズムは、（「先入れ」の終わりから始まる可能性がある）キュー内のキー押下変数のタイムスタンプを現在の時刻と定期的に比較してもよい。時間窓より古いキューでキー押下変数が発見された場合、その変数は削除されてもよい。特定の実施形態において、時間窓内にあるキー押下変数が発見されると、キュー内の全ての他の変数が時間窓内にあると想定されてもよく、その反復のために処理が停止されてもよい。

【0093】

620で、最近タイプされた文字及び／又は識別されたグラフィック要素を含む画面領域の座標が抽出された後、622で、座標が実行中のアプリケーションと比較され、抽出された座標が該当する要素に基づいて、アクティブ要素が判断される。座標がグラフィカルアプリケーションの任意の可能性のあるアクティブ要素に対応していない、又は、座標が間違ったアクティブ要素に対応している可能性がある。入力されたテキストが不完全又は間違っている、或いは、所望の画面変更が発生しなかった可能性もある。したがって、624で、ワークフローにおける対応するロボットアクティビティをチェックし、アクティブ要素、入力されたテキスト、及び／又は他の所望の画面変更が発生したことを確実にすることで、確認が実行される。これには、対応するアクティビティ又は一連のアクティビティでロボットがとるべきアクション（例えば、フィールドの選択、フィールドへのテ

キストの入力、ボタンのクリックなど）の判断、アクティビティ又は一連のアクティビティに基づいて画面の変更が発生するべきであること（例えば、新しいアプリケーションウィンドウが開くはずである、アプリケーションウィンドウが新しい形式に変更されるはずである、特定のテキストがアプリケーションによって表示されるはずであるなど）の判断、又は本発明の範囲から逸脱することのなく他の適切な確認アクションが含まれてもよいが、これらに限定されない。確認は、ロボット自体によって、ロボットの実行を監視する別のアプリケーション又はロボットによって、或いはこれらの両方によって実行されてもよい。626で確認が成功した場合、図6Aのステップ612で、プロセスは次のスクリーンショットに進む。しかし、確認が失敗した場合（例えば、ロボットが請求書のフィールドに「\$1, 234.50」を入力するはずであったが、その代わりに、これが「会社名」フィールドに表示された、又は、文字が欠落しているので入力された値が間違っていた場合（例えば、「\$1, 34.50」）など）、図6Aのステップ616で、プロセスは、修正措置をとる例外のフローに進む。

10

【0094】

幾つかの実施形態において、スクリーンショット記録、キー押下記録、及び／又はそれらの処理は、RPAロボットによって実行される。特定の実施形態において、レコーダアプリケーションが、スクリーンショット又はビデオとしてロボットアクティビティを記録し、一連のキー押下を記録し、後の処理のためにこの情報を記憶する、或いは、リアルタイム又はほぼリアルタイムの処理のためにこの情報をコンピューティングシステム又は別のコンピューティングシステムで実行されている別のアプリケーションに渡す。CVは、ビデオ及びキー押下の処理の前に適用され、ビデオ及びキー押下の処理の直後に適用されて、或いはもっと後に適用されて、境界矩形を含む認識されたグラフィック要素のセットを提供してもよい。次に、グラフィック要素である境界矩形とキャレット／テキストの座標との間に交差が発見された場合、その特定の要素が現在アクティブである又は「フォーカス」されている（即ち、「アクティブ要素」である）とみなされてもよい。

20

【0095】

場合によっては、あるフレームから次のフレームへの画面の変更が大きいことがある。例えば、ロボットがウィンドウを閉じると、画面の大部分が変わる場合がある。したがって、幾つかの実施形態において、変更閾値が決定され適用されて、時間隣接するスクリーンショットをとにかく比較するか否かを判断する（例えば、画面の2%を超える変更、5%、10%、30%、50%、70%を超える、など）。このような閾値に達する又は超えると、2つの時間隣接するフレーム間の差分が閾値を下回るまで、フレーム比較プロセスがスキップされてもよい。

30

【0096】

幾つかの実施形態において、ロボットは、確認が受け取られるまで、コンピューティングシステムの画面に変更を生じさせるワークフローステップの後に一時停止してもよい。これにより、前のステップが失敗した場合に、ロボットが次のステップ、その次のステップ、その次のステップへと進むことを防止する。したがって、誤った操作が迅速に特定され中止されてもよい。

【0097】

幾つかの実施形態において、監視及びロボット確認プロセスは、コンピューティングシステムを制御することを試みている同じロボットによって実行されてもよい。これは、ロボットが自身の動作を確認するためのメカニズムを提供する。特定の実施形態において、同じコンピューティングシステムで動作する、又はビデオ及びキー押下情報をリモートで受け取る異なるコンピューティングシステム（例えば、サーバ）で動作する別のロボットが、監視、確認を実行してもよい。幾つかの実施形態において、ローカル又はリモートで動作する非ロボットアプリケーションが監視、確認を実行する。特定の実施形態において、ビデオ及びキー押下の分析を実行するアプリケーション又はロボットと、確認を実行するロボット又はアプリケーションとは異なる。

40

【0098】

50

図 7 は、本発明の一実施形態による、R P A のためのロボット実行の画面応答確認を実行するためのプロセス 7 0 0 を示すアーキテクチャ図である。スクリーン記録 7 1 0 は、フレーム N (7 1 2)、N - 1 (7 1 4)、N - 2 (7 1 6)、N - 3 (7 1 8) を生成するために実行される。フレームは、画面全体、実行中のアプリケーションに関連付けられるウィンドウなどであってもよい。特定の実施形態において、複数の実行中のウィンドウアプリケーションのフレームセットが記憶されて比較されてもよく、或いは、これらのフレームがスクリーンショットから取得されてもよい。キー押下イベントの記録又はロボットアクティビティのテキスト検索 7 2 0 はまた、ロボットによって試みられたキー押下に関連付けられる文字 7 3 2 を含む時間ウィンドウキー文字キュー 7 3 0 を生成するために実行される。幾つかの実施形態において、文字は、それに関連付けられるキー押下が所定の期間を超えると、文字キューから削除される。特定の実施形態において、文字は、一意である場合、画面上で発見された後、キューから削除される。しかし、幾つかの実施形態において、一意でない場合（例えば、ユーザが「a a a a a」を連続してすばやく押す）、「a」の最も古いインスタンスがキューから削除されてもよい。

10

20

30

40

50

【 0 0 9 9 】

コアロジック 7 4 0 は、現在のフレーム及び直前のフレーム（この場合、フレーム 7 1 2、7 1 4）、並びにキー文字キュー 7 3 0 を受け取る。幾つかの実施形態において、コアロジック 7 4 0 は、図 6 A 及び図 6 B に関して上述のロジックの一部又は全てを実行してもよい。例えば、コアロジック 7 4 0 は、フレーム 7 1 2、7 1 4 を互いに比較し、視覚的な変更が発生したフレーム内の領域を判断してもよい。これは、例えば、それぞれのピクセルの赤／緑／青（R G B）値が相互に対して閾値を超えているか否かを比較することで判断されてもよい。

【 0 1 0 0 】

例えば、コアロジック 7 4 0 は、視覚的な変更の位置を分離し、その位置における文字を認識するために文字認識を実行し、キー文字キュー 7 3 0 の内容に基づいて、その文字が表示されると予測される文字に対応するか否かを判断してもよい。コアロジック 7 4 0 はまた、認識された文字をキー文字キュー 7 3 0 内の文字 7 3 2 に適合（照合）させてもよい。適合が発見された場合、適合位置の画面座標が抽出され、キャレット及び／又は文字タイプ領域 7 5 0 として提供されてもよい。

【 0 1 0 1 】

文字認識を使用することに加えて又はその代わりに、幾つかの実施形態において、画像認識（例えば、C V など）を使用して、新たに表示される又は非表示にする画像又はアイコンを検出してもよい。このプロセスは、領域差分分析段階でこの置換又は補足のロジックを伴うことを除いて、上記のプロセスと同じであってもよい。これは、例えば、ロボットが画像又はテキストをドキュメント又はフィールドにカットアンドペーストしている位置を判断するのに役立つ。

【 0 1 0 2 】

テキストのカットアンドペーストの場合、クリップボードからテキストがフラッシュされると、個々の文字はキー押下イベントとしてキャプチャされない。このような場合、クリップボードの内容とユーザが C T R L + V を押している時がキャプチャされてもよい。その後、クリップボードの内容は、個々の文字及び文字シーケンスが識別される方法と同様に、O C R の結果と比較されてもよい。しかし、アルゴリズムは、タイプされた文字キューをレビューするのではなく、C T R L + V が発生する前にクリップボードに記憶されていた文字列を処理するであろう。

【 0 1 0 3 】

幾つかの実施形態において、現在のアクティブ要素が、画面の変更を生じさせる特定のキーをロボットが押した場合のために、記憶されてもよい。例えば、ロボットが E n t e r キーを押すと、フォームが送られてウィンドウが閉じられてもよい。別の例において、ロボットがウェブブラウザの U R L 入力フィールドで E n t e r キーを押すと、これにより、ウェブブラウザがウェブサイトへのナビゲートを開始してもよい。アルゴリズムはこ

れを認識し、特定の時間待機してから、このようなシナリオにおいてアクティブ要素がある位置を確認してもよい。何故なら、大幅な画面の変更が発生する可能性があるためである。画面が比較的静的になると（例えば、画面の比較的小さな部分だけがあるスクリーンショットから次のスクリーンショットに変わるなど）、タイプされたテキストの検出とキャラクタ追跡が再開され、新たなアクティブ要素を発見してもよい。したがって、ロボットがEnterキー、Escapeキーなどを押したりした場合に、どの要素がフォーカスされたかを知ることが役立つ場合がある。また、現在のオペレーティングシステムが、現在のフォアグラウンドウィンドウのウィンドウ境界矩形をネイティブに取得する信頼できる方法を提供している場合（例えば、Windows（登録商標）のuser32.dllで利用可能な「GetForegroundWindow」など）、フォアグラウンドウィンドウの境界矩形を使用して、スクリーンキャプチャとフレーム処理に使用される領域（エリア）を制限してもよい。

10

【0104】

幾つかの実施形態において、フォーカスされた要素又はアクティブ要素が判断された場合、テキストが表示させないキーを押すときの振る舞い（例えば、Enter、Escape、Control、Control及び／又はAltと一又は複数の文字との組み合わせなど）は、アクティブ要素の性質に基づいて判断されてもよい。例えば、このような「ホットキー」（非表示のキー又はキーの組み合わせ）が押された場合、「ホットキー」がトリガしたアクションが分類されてもよい。フォーカスされた要素に「パスワード」というラベルが付され、ロボットが「Enter」を押した場合、「Enter」の押下は「ログイン」アクションを表すと推測されてもよい。これは、ロボットによる「Enter」の押下をただ知ることよりもより説明的であり得る。

20

【0105】

図8Aは、本発明の一実施形態による、それぞれフレームN-1及びフレームNの同窓生寄付ウィンドウを含むスクリーンショット800、810の一部802、812を示す。図8Aに示すように、フレームN-1の全てのフィールドは空であるが、フレームNでは、ロボットが名（first name）フィールドに文字「E」をタイプしている。この変更が発生したはずであるか否かを判断するために、幾つかの実施形態のアルゴリズムは、スクリーンショットをスケーリングし、それらを四角形に正規化する。この場合、スクリーンショット800（図8Bを参照）及びスクリーンショット810（図8Cを参照）は、64個の四角形×48個の四角形のグリッドに正規化され、そのうち27×24の部分が図8B～図8Fに示されている。これらの四角形又は領域は、64×48のバイナリ行列として表されてもよい。

30

【0106】

次に、各スクリーンショットの行を相互に比較して、そこに含まれるピクセルが変更されているか否かを確認し、変更を伴う各行について行列の値が「1」に設定されてもよい。これは、各行にゼロ以外の要素が存在するか否かを確認することで行われてもよい。図8Dに示すように、アルゴリズムが、ここで814で示す行列の行4を確認すると、そこに変更が検出される。次に、図8Eに示すように、アルゴリズムは、行4にステップインし、列7で816で示す四角形が、新たに入力された文字「E」の一部を含む。アルゴリズムは、変更を含む四角形を識別し続け、隣接する四角形818を、この部分でOCRを実行するために提出される領域としてつなぎ合わせる。図8Fを参照されたい。これを、ここで「領域」という。幾つかの実施形態において、領域を構成する一又は複数の四角形に変更がない場合、これらの四角形は、例えば、OCR処理をより速く実行するために、完全に1つの色に設定されてもよい。特定の実施形態において、十分な処理リソースが利用可能である場合、スクリーンショット間の差異領域を識別し、差異領域に対してOCRを実行してテキストを認識し、このテキストをキー文字キューと比較して、適合するものを検索してもよい。

40

【0107】

「E」が名フィールドに表示されると予測された場合、処理及び確認は次のフレームに

50

進んでもよい。しかし、文字「E」が正しい入力ではなかった場合（例えば、「E」のみの代わりに「Ed」が予測された場合など）、「E」が間違っただけのフィールドに表示された場合、又はその両方の場合、これは確認ロジックによって認識されてもよい。次に、ロボットは入力を試みて、誤ったテキストを削除し、同じフィールド又は別のフィールドに正しいテキストを入力する、或いは、その他の修正アクションを実行してもよい。修正措置が失敗した場合、ロボットはワークフローの実行を一時停止又は停止する、ユーザ又は開発者にエラーを通知するなどしてもよい。

【0108】

時には、ディスプレイ解像度の変更されたり、追加のモニタが接続されたりすることがある。幾つかの実施形態では、このような変更を検出して対応し、検出及び確認プロセスを依然として正確なままにする。図9は、本発明の一実施形態による、変更に対応するために、解像度の変更を確認し、接続されているディスプレイの範囲の変更を確認し、フレーム取得ロジックを設定するプロセス900を示すフローチャートである。該プロセスは、910で、コンピューティングシステムのための一又は複数の接続されたディスプレイを確認し、920で、接続されているディスプレイを以前に接続されていたディスプレイと比較することから始まる。これは、別の表示デバイスが存在するか否かの確認、表示デバイスの解像度の変更されたか否かの確認などを含んでもよい。幾つかの実施形態において、「接続されている」ディスプレイは、コンピューティングシステムと統合されているディスプレイであってもよい（例えば、スマートフォン、ラップトップコンピュータなどを用いる通常の場合のように）。

【0109】

930で、接続されている表示デバイス及び／又は解像度の変更されている場合、940で、解像度及びスケール（縮尺）が、接続された各ディスプレイについて取得される。950で、キャプチャされるスクリーンショットのスクリーンショット領域は、全画面表示寸法にスケールを掛けて所望の大きさ（例えば、8、16など）の倍数に調整された値に設定される。この倍数により、本明細書において後でさらに詳細に説明するように、スクリーンショットを四角形に分割することを容易にしてもよい。次に、960で、フレームキャプチャロジックが設定される（例えば、再起動、再初期化、新しい表示設定の提供など）。

【0110】

図10は、本発明の一実施形態による、ビデオレコーダのプロセス1000を示すフローチャートである。該プロセスは、1010で、スクリーンショットを撮ることから始まる。幾つかの実施形態において、これは、Windows（登録商標）のGraphics Device Interface (GDI) CopyFromScreen() 命令を使用して、C#で実行されてもよい。次に、1020で、スクリーンショットがフレームとしてバッファに追加される。これは、例えば、C#でビットマップオブジェクトとしてスクリーンショットをバッファに追加することで、実現されてもよい。1030で、該プロセスが依然として実行されている場合（例えば、アプリケーション、画面解像度の変更などを閉じてプロセスが停止されていない、など）、次のスクリーンショットについてスクリーンショットのキャプチャが繰り返されてもよい。C#の例が提供されているが、プロセス1000及び本明細書に開示されている他のプロセスについては、本発明の範囲から逸脱することなく、任意の適切なプログラミング言語、オペレーティングシステム、API、及び関数が使用されてもよいことに留意されたい。

【0111】

幾つかの実施形態では、キャラクタ追跡を実行して、ロボットが画面のどの要素にフォーカスしているかをより正確に識別する。例えば、テキストが表示されているグラフィック要素にキャラクタが表示されている場合、キー文字キューに新たに追加されたテキストが、このグラフィック要素に表示されているものである可能性がある。図11は、本発明の一実施形態による、キャラクタ追跡を実行するプロセス1100を示すフローチャートである。通常、キャラクタは、表示され、ユーザが直近にクリックした位置で又はその近

くで点滅を開始する。したがって、幾つかの実施形態では、直近のマウスクリックの座標を記憶し、この位置に近接するキャレットを検索する。これにより、キャレットを見つけるために処理されるスクリーンショットの量が減り、精度がさらに向上する可能性がある。幾つかの実施形態において、マウスクリックの履歴バッファ又は単一の直近のクリック位置が使用される。特定の実施形態において、例えば、ロボットがタブキーを「押す」と、システムは、キャレットが画面上の次のグラフィック要素に移動した可能性があるとして想定し、既知の場合はその位置に検索を再フォーカスし、又は、スクリーンショット全体を検索してもよい。

【0112】

1105で、ピクセルの変更が、変更が発生したスクリーンショットの各領域について計算され、この領域がバイナリ行列に投影される。バイナリ行列は、領域のピクセルが変更されたか否かの表現であり、スクリーンショット間で変更がないピクセルについての「0」と、変更されたピクセルについての「1」とを含んでもよい。幾つかの実施形態において、「領域」は、スクリーンショットからの複数の四角形を含み得る、変更が発生した四角形である。しかし、本発明の範囲から逸脱することなく、任意の他の適切な形状（例えば、長方形、六角形など）が使用されてもよい。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の処理能力に応じて、固定数の領域が分析のためにサポートされる。例えば、幾つかの実施形態では、2つの領域、3つの領域、10の領域などの抽出及びOCRをサポートする。幾つかの実施形態では、キャレットとタイプ又はペーストされたテキストとの両方を探索してもよい。現在のスクリーンショットと以前のスクリーンショットの間にL個を超える変更領域が発見された場合、最初に発見されたL個の領域が処理されてもよいし、又は、スクリーンショットが完全に無視されてもよい。これは、ユーザが別のウィンドウを起動したり画面の十分な部分が変更されたりして、次のスクリーンショットがキャプチャされる前にOCRが時間内に完了しない可能性がある画面を無視するのに役立つ。

【0113】

各バイナリ行列について、1110で、点滅するキャレット領域の候補が抽出され、1115で、バイナリ行列のメンバが結合される。「メンバ」は、本明細書で使用される場合、例えば文字、カーソルなどを表す形状など、バイナリ行列に存在する接続された形状である。行列のメンバの結合は、例えば、成分が8連結される連結成分（Connected Components）アルゴリズムを使用して実行されてもよい。連結成分は、各ピクセルが全ての他のピクセルに連結されるピクセルのセットである。

【0114】

1120で、形状が、行列のメンバの連結結果から抽出され、1125で、その形状が確認される。形状は、通常、例えば線を含み得る完全な矩形である必要がある。1130で、確認された形状の候補が記憶され、確認キューと比較される。キャレットの候補の位置、サイズ、及び形状が、タイムスタンプとともに記憶されてもよい。キャレットが点滅する頻度は、許容範囲内で一貫している必要がある（例えば、5%など）。キャレットが点滅するため、候補は分析のために記憶されて、キャレットの期待されるプロパティ（即ち、位置、サイズ、頻度）と適合するか否かを確認する必要がある。これは、複数のスクリーンショット（例えば、20など）で比較したときに、キャレットの候補が特定の頻度で点滅しているか否かを判断するのに役立つ。この情報は、ユーザが新しいフィールドをマウスでクリックしたり、タブキーを押したりした後キャレットが他の位置に再び表示される場合に、キャレットを識別するのに役立つ。当然、幾つかの実施形態において、プロセスの開始時に、確認キューは空である。

【0115】

1135で、所与のキャレットの候補が、キャレットの候補の出現／消失、サイズ、及び位置に基づいて点滅していることが確認された場合、1140で、点滅しているキャレットについて、キャレット追跡データが生成される。これは、画面上のキャレットが存在するグラフィカル要素（即ち、アクティブ要素）などの位置を含んでもよい。1145で

、確認された候補の領域と対応するメンバのバイナリ行列のデータとが、例えば、後で確認するために、確認キューに保存される。幾つかの実施形態において、図 11 のプロセスは、新しいスクリーンショット毎に繰り返されてもよい。

【0116】

図 12 A ~ 図 12 D は、本発明の一実施形態による、RPA のためのロボット実行の画面応答確認を実行するプロセス 1200 を示すフローチャートである。プロセス 1200 の前に、解像度の変更について確認が行われて、変更に対応するようにキャレット追跡ビデオロジックが設定されてもよい。例えば、図 9 を参照されたい。特定の実施形態において、プロセス 1200 は、ビデオレコーダと並行して実行されてもよい。例えば、図 10 を参照されたい。プロセス 1200 は、C # 及び Windows (登録商標) オペレーティングシステムを使用した例である。しかし、本発明の範囲から逸脱することなく、任意の適切なプログラミング言語、オペレーティングシステム、関連する API、フォーマット、及び関数が使用されてもよい。

10

【0117】

このプロセスは、1202 で、フォーマット `Format16bppRgb555` を使用してアイテム N (例えば、スクリーンショット、画面の一部、アプリケーションウィンドウなど) に対して `LockBits` を実行して N についての `BitmapData` オブジェクトを作成することから始まる。`LockBits` は、ビットマップの矩形部分をロックし、指定された形式でピクセルデータを読み書きするために使用できる一時バッファを提供する。`BitmapData` は、ビットマップの属性を記憶する。

20

【0118】

次に、1204 で、`BitmapData N` 及び `BitmapData N-1` (即ち、以前のアイテムの `BitmapData` オブジェクト) が、16 ピクセルの高さを有する横の行に分割される。しかし、本発明の範囲から逸脱することなく、プロセス 1200 のこのステップ及び他のステップについて、任意の高さ (例えば、8 ピクセル、32 ピクセルなど) が使用されてもよい。1206 で、同じ縦位置 (即ち、同じ「行」内。図 8 A ~ 図 8 F を参照) の `BitmapData N` 及び `BitmapData N-1` の各横の行について、`MPCMP` 命令が実行され、バイト配列の高速比較が実行される。`MEMCMP` は、行が同じであるか否かを示すものを提供する。

【0119】

30

1208 で、全ての行が `BitmapData N` と `BitmapData N-1` の間で同じであり (即ち、少なくとも 1 つの対応する行に差異があり)、1210 で、ロボットによって実行されている現在のアクティビティに基づいて差異が予測されなかった場合、プロセスは、図 12 D のステップ 1252、次の画面のキャプチャに進む。しかし、1210 で、差異が予測される場合、プロセスは、図 12 D のステップ 1256 に進み、エラーの修正を試みる。1208 で、全ての行が `BitmapData N` と `BitmapData N-1` の間で同じではなく (即ち、少なくとも 1 つの対応する行に差異があり)、1212 で、差異が予測されなかった場合、プロセスは、図 12 D のステップ 1256 に進み、エラーの修正を試みる。しかし、幾つかの実施形態において、`BitmapData N` と `BitmapData N-1` の比較結果が予測されるものでない場合 (即ち、変更が予測されなかった場合に予測されない変更が発生した場合、又は、変更が予測された場合に変更が発生しなかった場合)、ステップ 1214 ~ 1250 のプロセス又はそのサブセットが依然として実行されてもよいことに留意されたい。

40

【0120】

1208 で、全ての行が `BitmapData N` と `BitmapData N-1` の間で同じではなく (即ち、少なくとも 1 つの対応する行に差異があり)、1212 で、差異が予測された場合、1214 で、`MEMCMP` の結果が 0 に等しくない同じ行内の `BitmapData N` 及び `BitmapData N-1` についての横の行が抽出され、1216 で、抽出された横の行が、 16×16 ピクセルのサイズに分割される。例えば、図 8 C を参照されたい。1218 で、`BitmapData N` 及び `BitmapData`

50

a N - 1 の 16×16 ピクセルのブロック毎に、それらの間に差異があるブロックが抽出される。例えば、図 8 F を参照されたい。これは、Intel Intrinsics (登録商標) 命令をループする long XOR 関数の組み合わせ又は他の適切な機能を使用して実行されてもよい。

【0121】

上記のとおり、幾つかの実施形態において、処理できる領域の数は、所定の数 L に制限される。特定の実施形態において、各領域に含めることができるブロックの数を制限してもよい。例えば、次のスクリーンショットが取得される前に各領域で OCR を実行できることを確実にするために、4 個の四角形、9 個の四角形、16 個の四角形などの制限が課されてもよい。これは「最適化閾値」であってもよく、最適化閾値は、変更された領域の数の制限、所与の変更された領域に含まれる四角形の数の制限、又はその両方を含んでもよい。

10

【0122】

1220 で、各領域における抽出された 16×16 ピクセルのブロックの総数、領域の数、又はその両方が、最適化閾値と比較される。1220 で、最適化閾値が満たされると、プロセスは図 12 D のステップ 1252、次の画面のキャプチャに進む。ステップ 1220 で、最適化閾値を超えない場合、1222 で、近接する 16×16 ピクセルのブロックが、連結成分アルゴリズムを使用して結合される。連結成分アルゴリズムは、幾つかの実施形態において、8 連結の連結成分アルゴリズムであってもよい。これにより、どのブロックが隣接するブロックであるかが判断される。

20

【0123】

接続された隣接するブロックが判断されると、1224 で、近接するブロックの各セクタの境界矩形が判断され、領域が形成される。これは、極値アルゴリズムによって判断されてもよく、この場合、最高の x 値と最低の x 値を有するブロック (即ち、左端と右端のブロック) と最高の y 値と最低の y 値を有するブロック (即ち、上端と下端のブロック) が含まれる。そのような例が、図 8 F に示される。

【0124】

1226 で、領域の境界矩形毎に、ピクセルの変更が計算されて、バイナリ行列に投影される。 32×32 の領域に結合された 4 つの 16×16 のブロックに含まれる文字「E」のバイナリ行列 1400 の例を図 14 A に示す。

30

【0125】

殆どの場合、領域はそこに含まれるメンバ (例えば、文字、キャラット、ピクセルが変更された他の形状など) よりも大きくなる。OCR アルゴリズムの速度を上げるために、1228 で、各バイナリ行列について、各領域に含まれるメンバが (例えば、連結成分アルゴリズムを使用して) 判断され、バイナリ行列が各メンバについてトリミングされる。これにより、各領域における各メンバのトリミングされた行列が生成される。バイナリ行列 1400 から生成された文字「E」の例示的なトリミングされたメンバの行列 1410 を図 14 B に示す。幾つかの実施形態において、トリミングは、極値アルゴリズムを使用して実行されてもよい。

【0126】

40

次に、1230 で、点滅するキャラット領域の候補が、メンバの行列から抽出される。例えば、候補は矩形形状であってもよく、矩形形状は、幾つかの実施形態において、単一ピクセルの幅を有する縦線を含む可能性がある。次に、1232 で、抽出された点滅するキャラット領域の候補及び対応するメンバの行列データが、確認キューと比較され、点滅のサイズ、位置、及び頻度が分析されてもよい。1234 で点滅している場合、1236 で、点滅するキャラットについてキャラット追跡データが生成される。次に、1238 で、点滅するキャラット領域及びそれらに対応するメンバの行列データが、確認キューに保存される。幾つかの実施形態において、プロセス 1200 のこの部分が、図 11 のプロセス 1100 と同じ又は同様であってもよい。

【0127】

50

メンバのバイナリ行列は、所与のピクセルがスクリーンキャプチャN-1からスクリーンキャプチャNに変更されたか否かのみを示す。したがって、1240で、変更された各ピクセルについて、ピクセルデータがBitmapData Nから取得される。次に、1242で、メンバの矩形が生成され、OCRのため準備される。これは、変更された各ピクセルについてのピクセルデータの取り込み、キャレットのピクセルの削除、背景の処理（例えば、変更されていないピクセルをnull又は非常に縮小した値に設定するなど）などを含む。キャレットのピクセルが削除された場合は、この時点で、キャレット自体が特定の位置、形状、及びバイナリ行列のメンバのセットで検出されたと想定してもよい。この情報は、キャレット追跡の目的で記憶されてもよい。次に、1244で、準備されたメンバの矩形のピクセルデータについてOCRが実行される。

10

【0128】

1246で、RPAワークフローにおける現在のロボットアクティビティの予測される変更が、判断、確認される。予測される変更は、キー文字キューにおけるキー押下イベント、RPAワークフローにおけるロボットアクティビティに基づいて表示されるはずである変更などを分析することで、判断されてもよい。

【0129】

1248で確認が成功し、キー文字キューが使用される場合、1250で、OCR領域に適合したキー文字キューアイテム（例えば、キー押下イベントなど）がキー文字キューから削除されてもよい。文字の複数のインスタンスが存在する場合、例えば、キー文字キューにおけるその文字の最も古いインスタンスが削除されてもよい。次に、1252で、UnlockBitsがBitmapData N-1について実行され、システムメモリからこのビットマップのロックを解除し、1254で、BitmapData Nが位置N-1に移動される。プロセス1200は、次のキャプチャされたアイテムについて開始に戻ってもよい。

20

【0130】

しかし、1248で確認が失敗した場合（例えば、予測された変更が発生しなかった、又は、部分的にのみ発生した場合）、1256で修正措置が試みられる。例えば、ロボットは、ワークフローのどのステップが失敗したかを通知され、何らかの修正措置をとって、再度該ステップを試みてもよい。これには、ロボットが、ロボットが対話を試みているアプリケーションのウィンドウがアクティブウィンドウであるか否かをドライバレベルでチェックすること、関連するグラフィック要素に対するマウスポインタ又はキャレットの位置をチェックすること、正しいグラフィック要素がアクティブ要素であるか否かをチェックすること、これらのうちの任意の組み合わせなどが含まれてもよいが、これらに限定されない。ロボットは、正しいウィンドウをアクティブでフォーカスされたウィンドウに設定する、フォーカスを所望のアクティブ要素に設定する、マウスを動かしてクリックするなどして、その後、再度該ステップを試みてもよい。幾つかの実施形態において、修正措置をとっている間、フレーム比較プロセスは一時停止されてもよい。1258で修正措置が成功した場合、プロセスはステップ1252に進んでもよい。

30

【0131】

しかし、1258で修正措置が失敗した場合、1260で、エラーロジックが実行されてもよい。エラーロジックには、例外のスロー、ロボットの実行の停止及び画面へのメッセージの表示又はロボットが所望のアクションを達成できなかったことを示す（例えば、テキスト又は電子メールを介した）人間へのメッセージの送信が含まれてもよいが、これらに限定されない。特定の実施形態において、失敗したアクションがユーザに提供されてもよく、そうすると、ユーザは、問題のトラブルシューティングを試みることができ、又は、ユーザは、RPA開発者によって修正されるべき問題を報告できる。

40

【0132】

図13A～図13Cは、本発明の一実施形態による、ペーストされたテキストの追跡及び確認を実行するプロセス1300を示すフローチャートである。該プロセスは、任意に1305でキー押下記録を実行し、1310でスクリーンキャプチャ（例えば、スクリー

50

ン記録、画面の全て又は一部のスクリーンショットのキャプチャなど)を実行して、押されたキーとグラフィックの変更が発生した画面上の位置とをそれぞれ判断することから始まる。次に、1315でロボットアクションが解釈される。ロボットアクションには、マウスクリック、CTRL+Vの押下、右クリックとメニューからのペーストの選択、ホームボタンのクリックとアプリケーションにおけるペーストなどが含まれるが、これらに限定されない。ロボットがある位置をクリックしてクリップボードから素早くペーストすると、キャラットが見落とされ、ロボットのアクションを別の方法で再構築する必要がある場合がある。1320でクリップボードからのペーストが行われていない場合、1325で、タイプされたテキストの検出及びキャラット追跡ロジックが実行され、幾つかの実施形態において、図12A～図12Dのプロセス1200のラインに沿って実行される可能性がある。

10

【0133】

しかし、1320で、クリップボードにロボットから最近ペーストされたデータが含まれている場合(例えば、直近の500ミリ秒、直近の1秒以内にペーストされた場合など)、1330で、フレームNとフレームN-1の間のピクセルの差分が計算される。1335で、フレームNとフレームN-1の間で変更が発生した、予測される同じ位置の領域が、抽出される。1340で、各領域について、ピクセルの変更が計算され、変更がバイナリ行列に投影される。

【0134】

1345で、各バイナリ行列について、例えば、連結成分アルゴリズムを使用してメンバが判断され、各メンバについて、メンバの行列が判断される。1350で、メンバの行列において変更されたピクセル毎にピクセルデータが抽出され、メンバの矩形が生成される。1355で、各メンバの矩形はOCRのために準備され、1360で、OCRは準備された各メンバの矩形について実行される。1370でクリップボードモニタによって提供されるクリップボードコンテンツを使用して、1365で、OCRの結果とクリップボードコンテンツとのファジーマッチング比較が、各OCRの結果について実行される。幾つかの実施形態において、クリップボードのテキストコンテンツは、Clipboard.GetText()を使用してSystem.Windows.Forms.dllから取得される。1375で、適合が発見されると、1380で、(例えば、矩形の形式で)ペーストされたテキストの座標が生成され、クリップボードがフラッシュされ、プロセスは、次のフレームのためにステップ1305及び1310に戻る。

20

30

【0135】

しかし、1375で適合が発見されなかった場合(例えば、予測されるペーストされたコンテンツが表示されなかった、又は、部分的にのみ表示された場合など)、1385で修正アクションが試行されます。例えば、ロボットは、ワークフローのどのステップが失敗したかを通知され、何らかの修正措置をとり、再度該ステップを試みてもよい。これには、ロボットが、ロボットが対話することを試みているアプリケーションのウィンドウがアクティブウィンドウであるか否かをドライバレベルでチェックすること、関連するグラフィック要素に対するマウスポインタ又はキャラットの位置をチェックすることが含まれるが、正しいグラフィック要素がアクティブ要素であるか否かをチェックすること、これらのうちの任意の組み合わせなどが含まれてもよいが、これらに限定されない。ロボットは、正しいウィンドウをアクティブでフォーカスされたウィンドウに設定する、フォーカスを所望のアクティブ要素に設定する、マウスを動かしてクリックするなどして、その後、再度該ステップを試みてもよい。幾つかの実施形態において、修正措置をとっている間、フレーム比較プロセスは一時停止されてもよい。1390で修正措置が成功した場合、プロセスはステップ1380に進んでもよい。

40

【0136】

しかし、1390で修正措置が失敗した場合、1395で、エラーロジックが実行されてもよい。エラーロジックには、例外のスロー、ロボットの実行の停止及び画面へのメッセージの表示又はロボットが所望のアクションを達成できなかったことを示す(例えば、

50

テキスト又は電子メールを介した) 人間へのメッセージの送信が含まれてもよいが、これらに限定されない。特定の実施形態において、失敗したアクションがユーザに提供されてもよく、そうすると、ユーザは、問題のトラブルシューティングを試みることができ、又は、ユーザは、RPA開発者によって修正されるべき問題を報告できる。

【0137】

図15は、本発明の一実施形態による、CV及びロボットアクティビティの予測される結果を使用してアクティブ要素を判断し、ロボットアクションを確認するプロセス1500を示すフローチャートである。該プロセスは、1510で、フレームが所定の閾値を超えて以前のフレームから変更されたか否かを判断することから始まる。これは、フレームの特定の部分を超えて(例えば、2%を超えて)変更されたか否か、所定のピクセル数を

10

【0138】

1520で、閾値を超えた場合、画面上のグラフィック要素の少なくとも一部も変更されている可能性がある。1530で、CV前処理を実行して、グラフィック要素の種類及び位置を識別する。グラフィック要素の種類及び位置は、メモリに記憶されてもよい。幾つかの実施形態において、CV前処理が実行されている間、フレーム比較プロセスが一時停止されてもよい。

【0139】

1520で閾値を超えなかった場合、又は、1530でCV前処理が完了した後、1540で、以前のフレームと現在のフレームの間に追加されたキー文字キューに新たに出現した要素が判断される。キー文字キューに新たに出現した要素がある場合、これらは、画面上の適切なグラフィック要素に表示されたと考えられる。次に、1550で、現在のフレームで画面が変更された位置が判断され、1560で、グラフィック要素の位置へ変更を適合させることを試みる。変更がグラフィック要素の1つのみで発生した場合、1570で、適合したグラフィック要素が、アクティブ要素として設定される。しかし、複数のグラフィック要素内で変更が発生した場合、又は、グラフィック要素で変更が発見されなかった場合、1580で、修正アクションが実行される(例えば、コンテンツが表示されるべきではなかったフィールドからコンテンツを削除し、そのコンテンツを正しいフィールドに挿入することを試みるなど)。1590で、次のフレームがフェッチされ、プロセスが繰り返される。

20

30

【0140】

図16は、本発明の一実施形態による、RPAのためのロボット実行の画面応答確認を実行するように構成されたシステム1600を示すアーキテクチャ図である。システム1600は、例えばデスクトップコンピュータ1602、タブレット1604、スマートフォン1606などのユーザコンピューティングシステムを備える。しかし、本発明の範囲を逸脱することなく、スマートウォッチ、ラップトップコンピュータ、モノのインターネット(IoT)デバイス、車両コンピューティングシステムなどを含むがこれらに限定されない、任意の所望のコンピューティングシステムが使用されてもよい。

【0141】

各コンピューティングシステム1602、1604、1606は、それ自体の動作又は別個のロボットの動作に関するスクリーンショット、キーストローク、実行中のアプリケーション、アプリケーション視覚要素、視覚要素位置、アプリケーション位置、ロボットワークフロー、実行されている現在のステップなどを記録する、そこで実行されるデジタルプロセス1610を有する。実際、本発明の範囲から逸脱することなく、画面グラフィック、ロボット入力、表示要素などに関する任意の所望の情報が記録されてもよい。特定の実施形態において、ビデオ及びキーストロークの記録のみが最初にキャプチャされ、その後、他の情報がCVを使用して判断される。しかし、追加情報は、CVプロセスにフォーカスして向上させるのに役立ってもよい。デジタルプロセス1610は、本発明の範囲から逸脱することなく、RPAデザイナーアプリケーション、オペレーティングシステムの

40

50

一部、パーソナルコンピュータ（ＰＣ）又はスマートフォン用のダウンロード可能なアプリケーションを介して生成されるロボットであってもよいし、或いは、任意の他のソフトウェア及び／又はハードウェアであってもよい。実際、幾つかの実施形態において、一又は複数のデジタルプロセス１６１０のロジックが、物理ハードウェアを介して部分的又は完全に実装される。

【０１４２】

デジタルプロセス１６１０は、記録されているスクリーンショット、キーストローク、実行中のアプリケーション、アプリケーション要素及び位置、これらのうちの組み合わせなどを、ネットワーク１６２０（例えば、ローカルエリアネットワーク（ＬＡＮ）、移動通信ネットワーク、衛星通信ネットワーク、インターネット、これらのうちの任意の組み合わせなど）を介してサーバ１６３０に送る。幾つかの実施形態において、サーバ１６３０はコンダクタアプリケーションを実行させてもよく、データがハートビートメッセージの一部として定期的に送られてもよい。サーバ１６３０は、データベース１６４０にデジタルプロセス１６１０からの情報を記憶してもよい。

10

【０１４３】

サーバ１６３０は、サーバ１６３０がデータを受け取るコンピューティングシステム（即ち、コンピューティングシステム１６０２、１６０４、１６０６）のための画面応答確認ロジックのインスタンス１６３２を実行する。サーバは、インスタンス１６３２からの結果を分析して、ロボットがコンピューティングシステムと対話するとき何をしているかを判断してもよい。ロボットが何をすることを試みているかをインスタンス１６３２が確実に判断できるように、ワークフローにおけるステップをマッピングし、ビデオ及びキー押下に時間を同期させてもよい。確認が成功したか失敗したか、失敗が生じた場合の失敗の性質、ロボットが行っているステップなどに関する情報が、デジタルプロセス１６１０によって監視されたロボットの実行を確認するために、各々のユーザコンピューティングシステム１６０２、１６０４、１６０６に送り返されてもよい。幾つかの実施形態において、サーバ１６３０による確認は、リアルタイム又はほぼリアルタイムで行われてもよく、そうすると、プロセス１６１０は、ＲＰＡロボットが各コンピューティングシステムで実行されるときに、確認を実行できる。

20

【０１４４】

図１７は、本発明の一実施形態による、ＲＰＡのためのロボット実行の画面応答確認を実行するプロセス１７００を示すフローチャートである。プロセスは、１７０５で、ＣＶアルゴリズムを適用して、ユーザインタフェースにおけるグラフィック要素及び関連する境界矩形を判断することから始まる。幾つかの実施形態において、判断されたグラフィック要素は、ＲＰＡアクティビティのターゲットグラフィック要素を含んでもよい。１７１０で、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間に差異が存在するか否かが判断される。現在のスクリーンショットフレームと以前のスクリーンショットフレームは、ユーザインタフェースの全体又は一部を含んでもよい。１７１５で、現在のスクリーンショットフレームと以前のスクリーンショットフレームの間に差異が存在し、１７２０で、ＲＰＡロボットのワークフローのアクティビティに基づいて、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間の差異が予測される場合、１７２５で、ＲＰＡロボットのワークフローのアクティビティに基づいて一又は複数の予測される画面変更に対して現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間の一又は複数の変更が確認される。幾つかの実施形態において、確認には、現在のスクリーンショットフレームが以前のスクリーンショットフレームとは異なる一又は複数の位置でＯＣＲを実行することと、ＯＣＲによって認識された文字をキー文字キュー内の文字に適合させることが含まれる。

30

40

【０１４５】

１７３０で確認が成功した場合、プロセスは次のフレーム（即ち、現在のスクリーンショットフレームとしての次のフレームと以前のスクリーンショットフレームとしての現在のスクリーンショットフレーム）に進み、ワークフローにおける次のアクティビティ又は

50

同じアクティビティによって影響を受ける次の変更に進む。しかし、確認が失敗した場合、1740で例外がスローされる、又は、修正アクションがとられる。1740で修正アクションが取られ、それが成功した場合、プロセスは、ステップ1740に進んでもよい。

【0146】

ステップ1710に戻ると、1715で差異が生じず、1745で差異が予測されなかった場合、プロセスはステップ1735に進む。しかし、1745で差異が予測された場合、プロセスはステップ1740に進む。

【0147】

幾つかの実施形態において、修正アクションには、RPAロボットが、該ロボットが対話することを試みているアプリケーションのウィンドウがアクティブウィンドウであるかをドライバレベルでチェックすることと、変更が生じることが予測されるグラフィック要素に対するマウスポインタ又はキャレットの位置をチェックすること、正しいグラフィック要素がアクティブ要素であるか否かをチェックすること、又はこれらのうちの任意の組み合わせが含まれる。特定の実施形態において、修正アクションには、RPAロボットが、正しいウィンドウをアクティブでフォーカスされたウィンドウとして設定すること、フォーカスをアクティブ要素に設定すること、マウスを動かしてマウスクリックイベントを生じさせること、又はこれらのうちの任意の組み合わせが含まれる。幾つかの実施形態において、例外のスローには、RPAロボットの動作の停止、画面へのメッセージの表示、RPAロボットが一又は複数の予測される画面変更を達成できなかったことを示すメッセージの送信、失敗した一又は複数のアクションをリストすること、又はこれらのうちの任意の組み合わせが含まれる。

【0148】

幾つかの実施形態において、1750で、時間窓の間に発生したRPAロボットによって作成されたキー押下イベントを含むキー文字キューが生成される。特定の実施形態において、適合する文字は、1725で、確認の間にキー文字キューから削除される。幾つかの実施形態において、キー文字キューは、時間窓の間のキー押下イベントの各々について、キー押下イベントの文字とキー押下イベントが発生した時とを含むFIFOキューである。

【0149】

幾つかの実施形態において、確認は、現在のスクリーンショットフレームが以前のスクリーンショットフレームと異なる一又は複数の領域を判断することと、判断された一又は複数の領域における一又は複数の結合されたメンバを抽出することと、抽出された一又は複数の結合されたメンバに対してOCRを実行することと、一又は複数の認識された文字、一又は複数の認識された文字シーケンス、又はこれら両方と、これらの各々の位置とを生成することとを含む。確認は、一又は複数の認識された文字、一又は複数の認識された文字シーケンス、又はこれら両方を、アクティビティに基づいて表示されると予測される、キー文字キューと、或いは、一又は複数の文字、一又は複数の文字シーケンス、又はこれら両方と比較することを含む。特定の実施形態において、ターゲット（対象）のグラフィック要素及びターゲットのグラフィック要素に入力されるコンテンツは、RPAロボットのワークフローのアクティビティに基づいて判断され、確認は、現在のスクリーンショットフレームと以前のスクリーンショットフレームとの間における一又は複数の変更がターゲットのグラフィック要素の位置と適合するか否かと、一又は複数の変更がターゲットのグラフィック要素に入力されるコンテンツと適合するか否かとを確認することを含む。幾つかの実施形態において、アクティビティには、グラフィック要素へのテキストの入力、キャレットのグラフィック要素への移動、コンテンツのグラフィック要素へのペースト、又はこれらのうちの組み合わせが含まれる。特定の実施形態において、RPAロボットのワークフローのアクティビティに基づく一又は複数の予測される画面変更には、フォームの送信、新しいアプリケーションウィンドウを開くこと、又は現在のアプリケーションウィンドウの変更が含まれる。

10

20

30

40

50

【0150】

図6A、図6B、図7、図9～図13B、図15、図17で実行されるプロセスステップは、本発明の実施形態に従って、コンピュータプログラムによって実行されて、図6A、図6B、図7、図9～図13B、図15、図17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で具現化されてもよい。コンピュータ読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任意の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6A、図6B、図7、図9～図13B、図15、図17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。

10

【0151】

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。

20

【0152】

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。

【0153】

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。

30

【0154】

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。

40

【0155】

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。

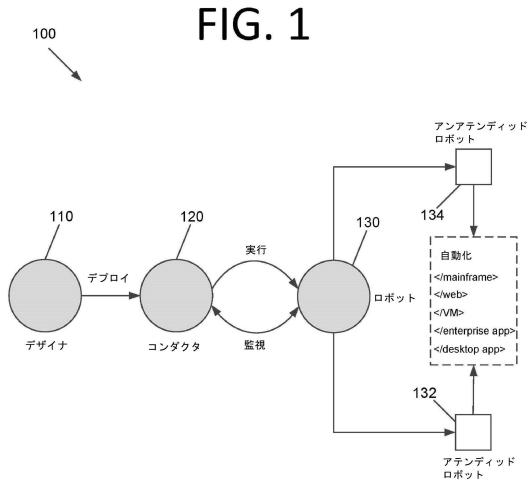
【0156】

50

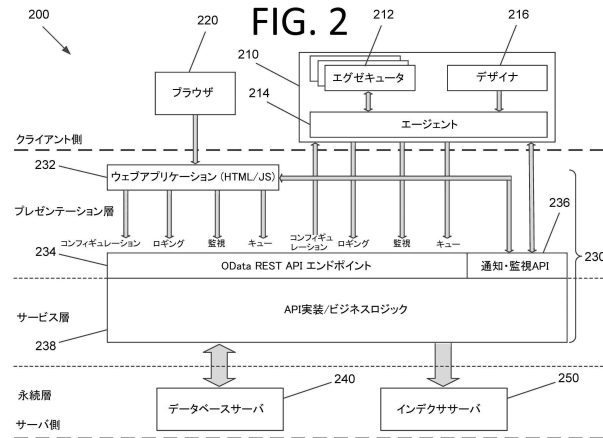
当業者は、上記の本発明が異なる順序のステップで、及び／又は開示されているものとは異なる構成のハードウェア要素で実施されてもよいことを容易に理解するであろう。したがって、本発明をこのような好ましい実施形態に基づいて説明してきたが、本発明の主旨及び範囲内にありながら、特定の修正、変形、及び代替構造が明らかであることは当業者には明らかであろう。したがって、本発明の境界及び範囲を決定するために、添付の特許請求の範囲を参照されたい。

【図面】

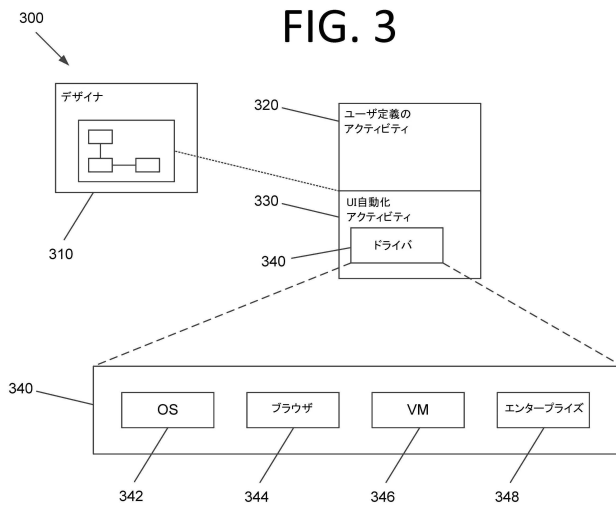
【図 1】



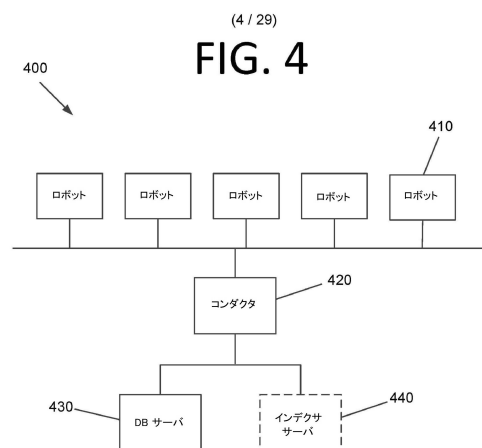
【図 2】



【図 3】



【図 4】



10

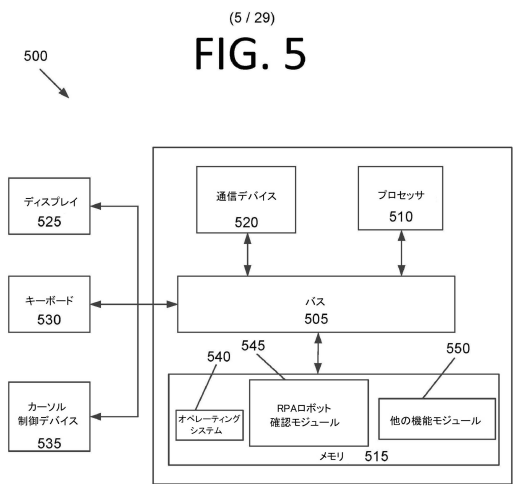
20

30

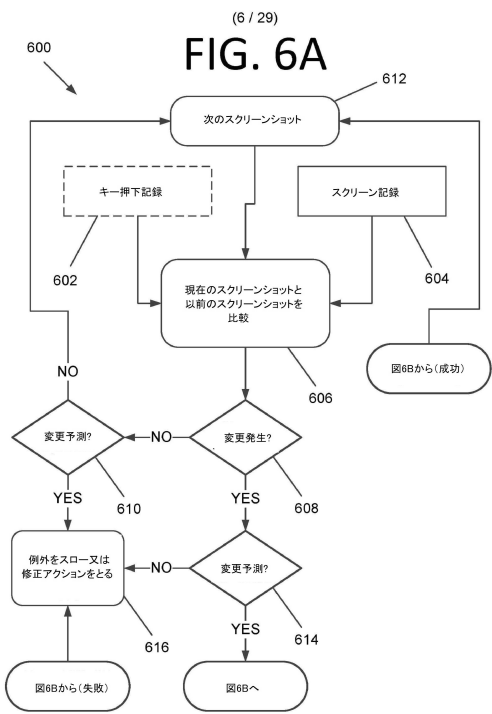
40

50

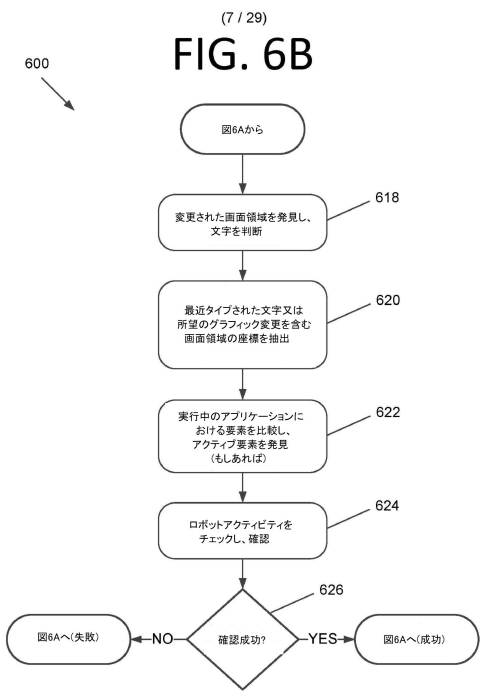
【図 5】



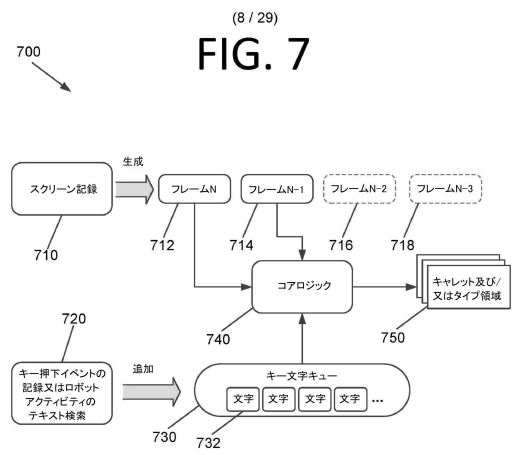
【図 6 A】



【図 6 B】



【図 7】



10

20

30

40

50

【図 8 A】

(9 / 29)

FIG. 8A

802

フレームN-1

812

フレームN

Alumni Donation

Name*

First Name Last Name

Email*

Phone*

Amount you would like to Donate

\$

Would you like to make this donation in someone else's name?

☐ Yes ☐ No

Donate

Alumni Donation

Name*

E First Name Last Name

Email*

Phone*

Amount you would like to Donate

\$

Would you like to make this donation in someone else's name?

☐ Yes ☐ No

Donate

【図 8 B】

FIG. 8B

800

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Alumni Donation

Name*

First Name Last Name

Email*

Phone*

Amount you would like to Donate

\$

Would you like to make this donation in someone else's name?

☐ Yes ☐ No

Donate

【図 8 C】

FIG. 8C

810

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Alumni Donation

Name*

E First Name Last Name

Email*

Phone*

Amount you would like to Donate

\$

Would you like to make this donation in someone else's name?

☐ Yes ☐ No

Donate

【図 8 D】

FIG. 8D

810

814

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Alumni Donation

Name*

E First Name Last Name

Email*

Phone*

Amount you would like to Donate

\$

Would you like to make this donation in someone else's name?

☐ Yes ☐ No

Donate

10

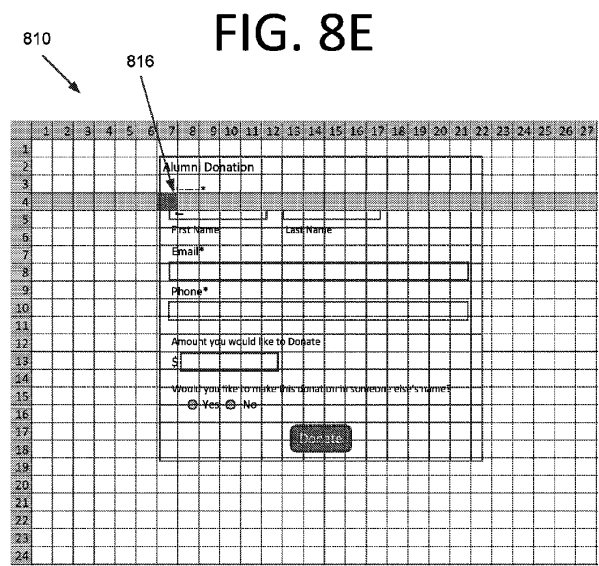
20

30

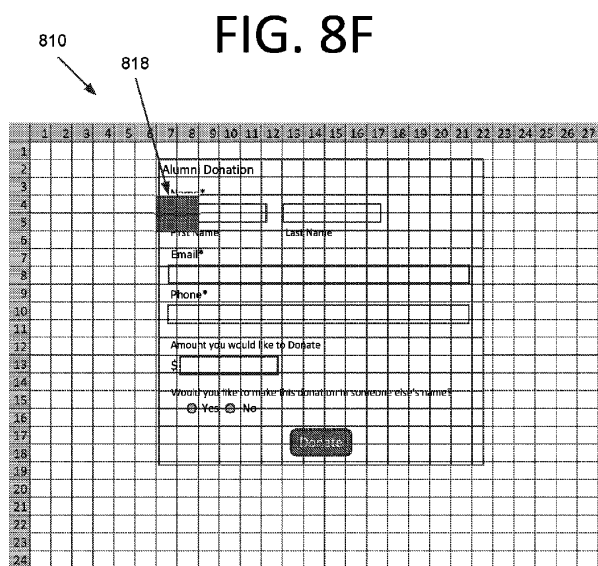
40

50

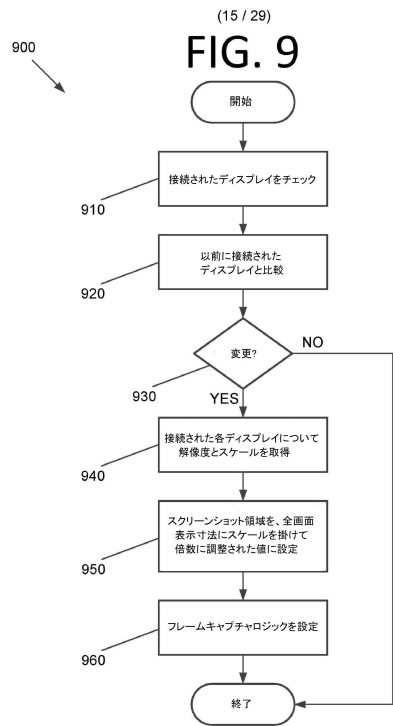
【図 8 E】



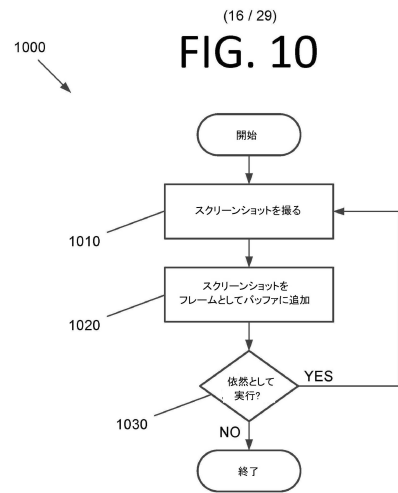
【図 8 F】



【図 9】



【図 10】



10

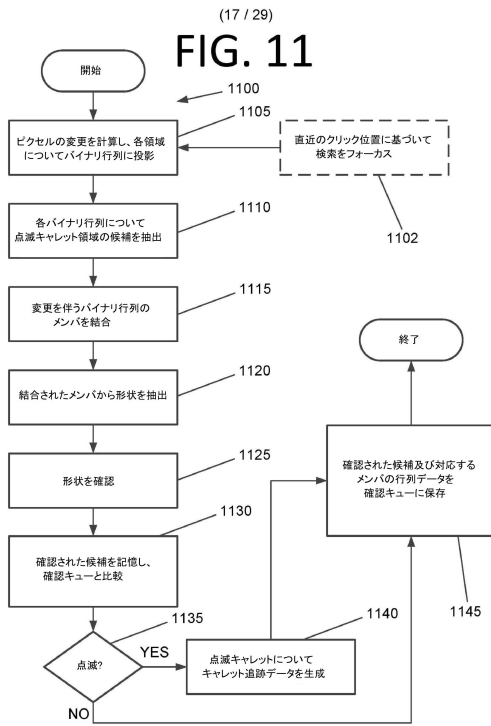
20

30

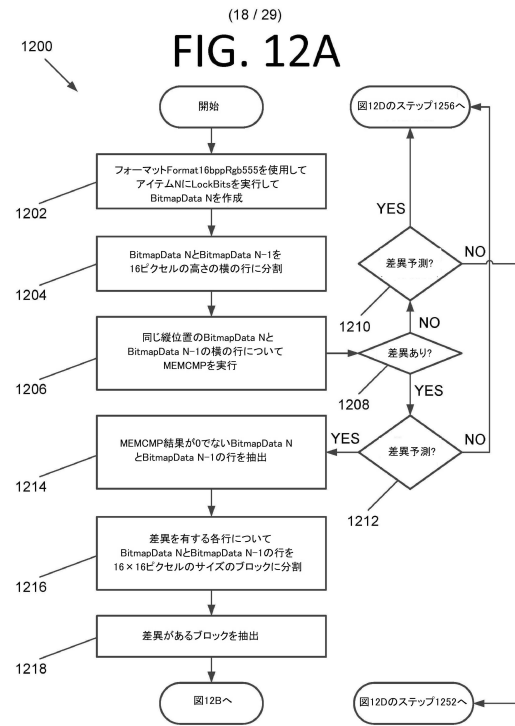
40

50

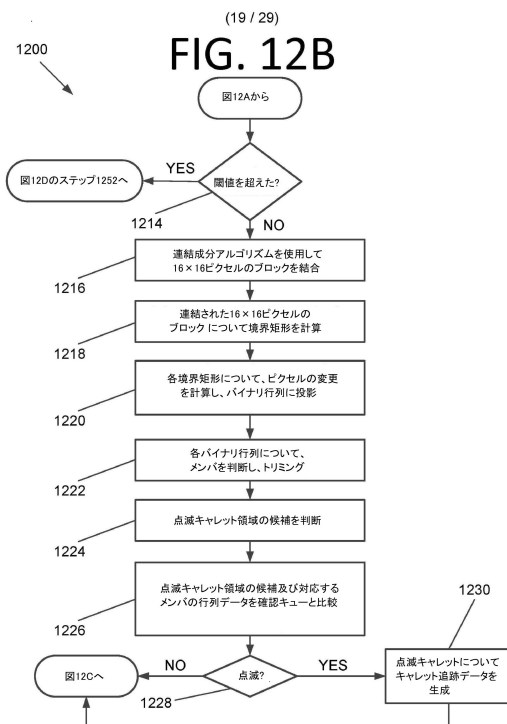
【図 1 1】



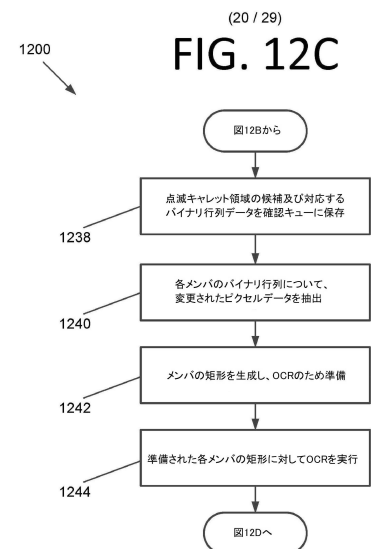
【図 1 2 A】



【図 1 2 B】



【図 1 2 C】



10

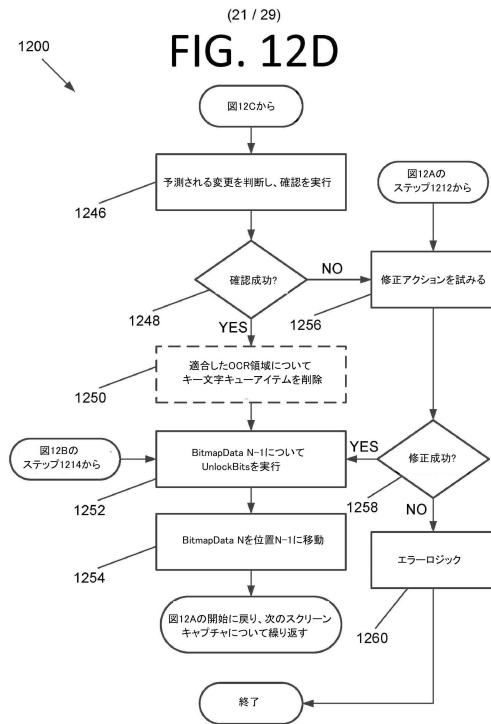
20

30

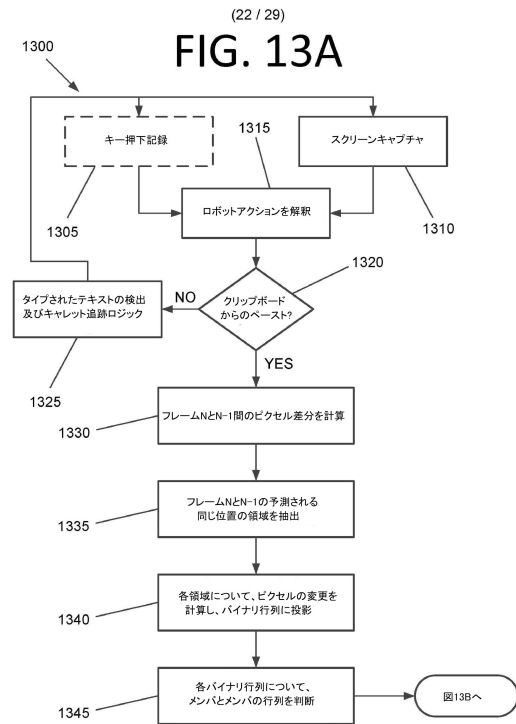
40

50

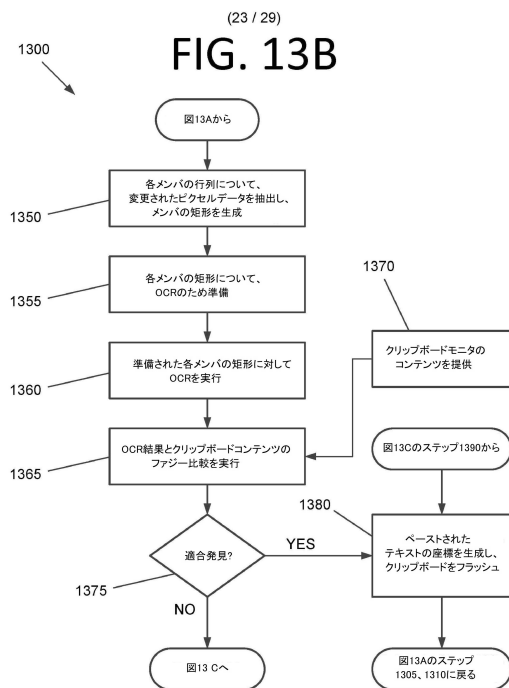
【図 12 D】



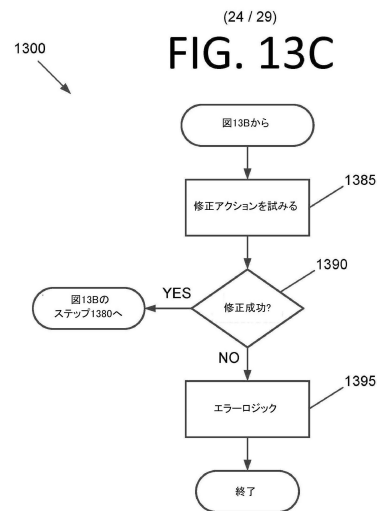
【図 13 A】



【図 13 B】



【図 13 C】



10

20

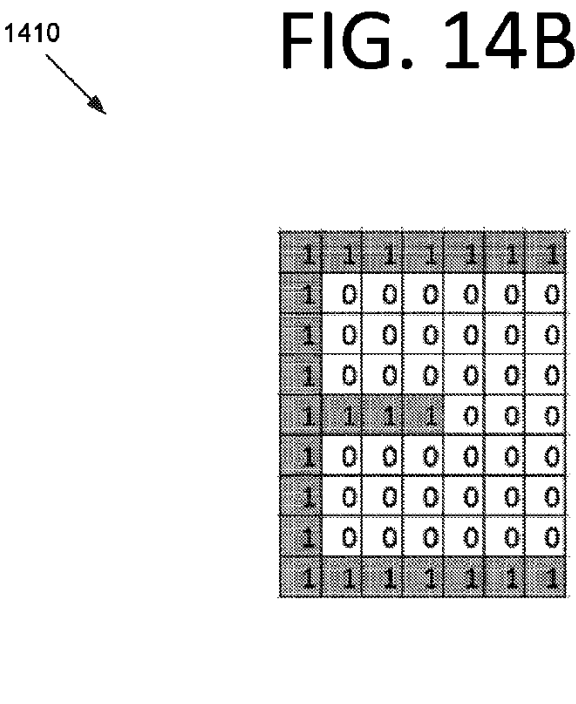
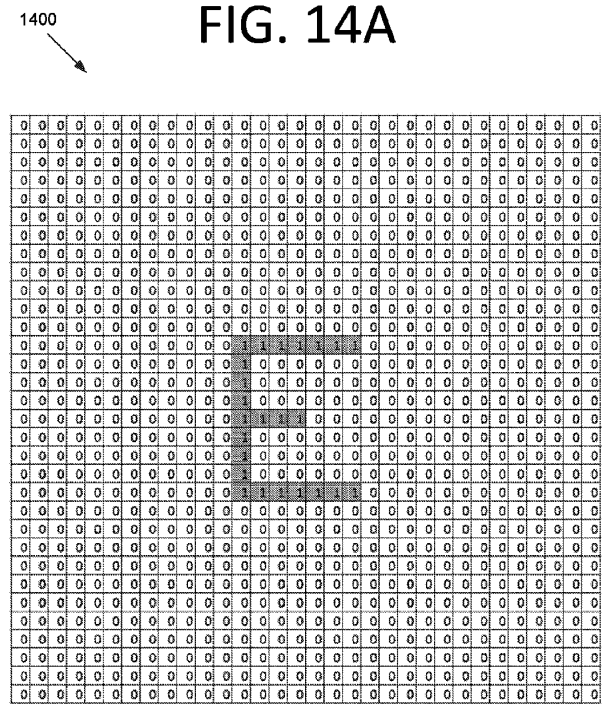
30

40

50

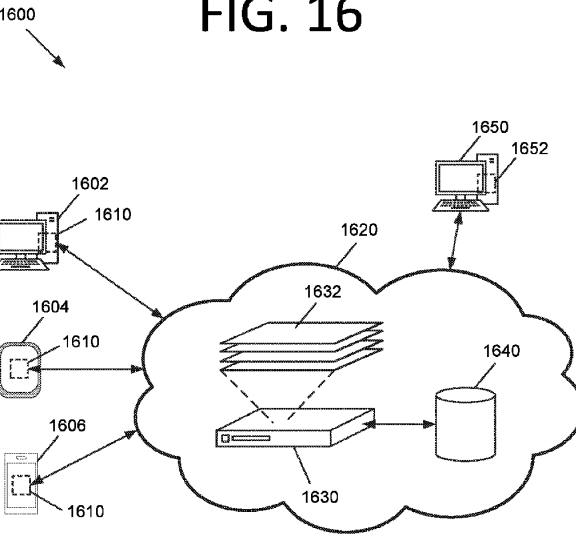
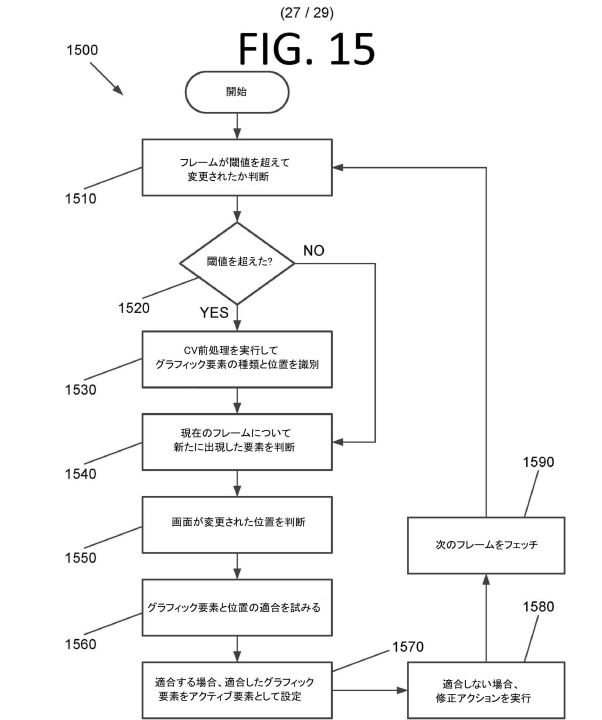
【図 1 4 A】

【図 1 4 B】



【図 1 5】

【図 1 6】



10

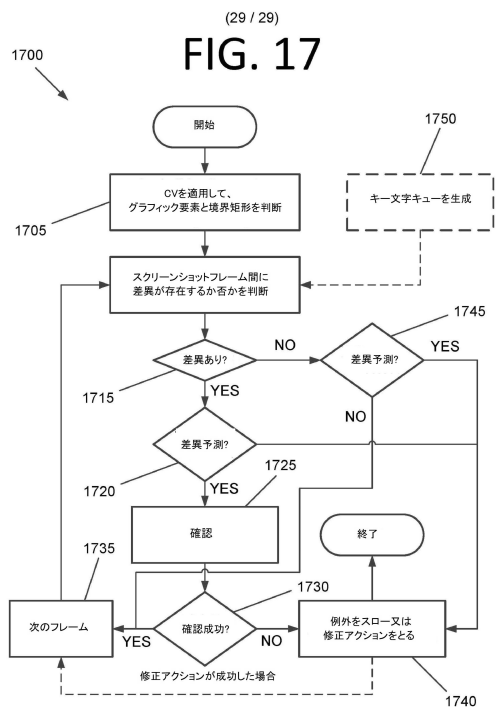
20

30

40

50

【図 17】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2020/051878

A. CLASSIFICATION OF SUBJECT MATTER G06K 9/00(2006.01)i; G06K 9/62(2006.01)i; G06F 40/279(2020.01)i; G06F 9/54(2006.01)i; G06F 3/023(2006.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06K 9/00(2006.01); B64F 5/40(2017.01); G06F 17/30(2006.01); G06F 9/445(2006.01); G06F 9/455(2006.01); G06K 15/02(2006.01); G06N 3/08(2006.01); G06N 99/00(2010.01); G06Q 10/06(2012.01) Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: screen, response, difference, activity, validate		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2019-0324781 A1 (EPLANCE SOFTWARE PVT. LTD.) 24 October 2019 (2019-10-24) paragraphs [0027]-[0071] and figures 1-11	1-24
A	KR 10-2019-0095651 A (SAMSUNG SDS CO., LTD.) 16 August 2019 (2019-08-16) claim 1 and figures 1-14	1-24
A	US 2018-0197123 A1 (SUTHERLAND GLOBAL SERVICES INC.) 12 July 2018 (2018-07-12) claim 1 and figures 1-14	1-24
A	US 2017-0242676 A1 (AIRBUS OPERATIONS (S.A.S.)) 24 August 2017 (2017-08-24) paragraph [0094], claims 1-6 and figures 1-7B	1-24
A	US 2019-0156165 A1 (CANON KABUSHIKI KAISHA) 23 May 2019 (2019-05-23) claim 1 and figures 1-17	1-24
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "D" document cited by the applicant in the international application "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 29 January 2021		Date of mailing of the international search report 29 January 2021
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon 35208, Republic of Korea Facsimile No. +82-42-481-8578		Authorized officer KANG MIN JEONG Telephone No. +82-42-481-8131

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/US2020/051878

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2019-0324781	A1	24 October 2019	None			
KR	10-2019-0095651	A	16 August 2019	None			
US	2018-0197123	A1	12 July 2018	GB	2558676	A	18 July 2018
US	2017-0242676	A1	24 August 2017	FR	3048099	A1	25 August 2017
				FR	3048099	B1	23 March 2018
US	2019-0156165	A1	23 May 2019	JP	2019-097021	A	20 June 2019
				US	10546220	B2	28 January 2020

Form PCT/ISA/210 (patent family annex) (July 2019)

フロントページの続き

MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,N
E,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,
CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,IT,JO,JP,K
E,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,N
G,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,
TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

(特許庁注：以下のものは登録商標)

1. J A V A S C R I P T

2. Z I G B E E

3. B L U E T O O T H

4. W C D M A

アメリカ合衆国，１００１６ ニューヨーク，ニューヨーク，パーク アベニュー ９０，トウエン
ティース フロア

F ターム (参考) 5E555 AA05 AA09 BA02 BB02 BC01 CA02 CA17 CB05 CB20 CB72
CC01 DA01 DB03 DB41 DC13 DC63 DC84 DD06 EA03 EA24 FA00