

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)公開番号
特開2023-75577
(P2023-75577A)

(43)公開日 令和5年5月31日(2023.5.31)

(51)国際特許分類

G 0 6 F 8/34 (2018.01)

F I

G 0 6 F

8/34

テーマコード(参考)

5 B 3 7 6

審査請求 未請求 請求項の数 14 O L (全27頁)

(21)出願番号 特願2021-188564(P2021-188564)
(22)出願日 令和3年11月19日(2021.11.19)(71)出願人 390002761
キヤノンマーケティングジャパン株式会社
東京都港区港南2丁目16番6号
592135203
(71)出願人 キヤノンITソリューションズ株式会社
東京都港区港南2丁目16番6号
(74)代理人 100189751
弁理士 木村 友輔
(72)発明者 岩永 尚己
東京都港区港南2丁目16番6号 キヤノンITソリューションズ株式会社内
(72)発明者 柴本 文洋
東京都港区港南2丁目16番6号 キヤノンITソリューションズ株式会社内
最終頁に続く

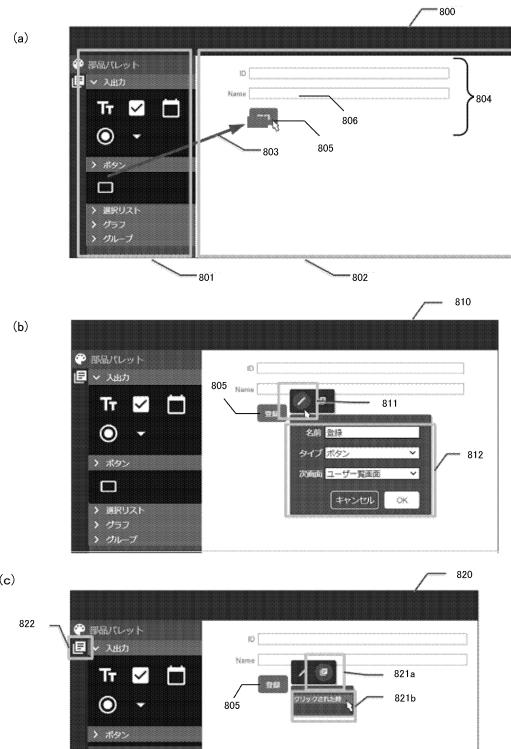
(54)【発明の名称】 情報処理装置、情報処理方法およびプログラム

(57)【要約】 (修正有)

【課題】本番用のアプリケーションソフトウェアにおける複数のイベントの実行を仮想的に見ることのできるコンテンツを容易に生成する情報処理装置、情報処理方法及びプログラムを提供する。

【解決手段】方法は、アプリケーションソフトウェアで表示される複数の画面のうちいずれかを選択する第1の選択手段と、第1の選択手段で選択された画面に対応づけて記憶された動作のうちいずれかを選択する第2の選択手段と、第1の選択手段で選択された画面を表示する第1のイベントと、第2の選択手段で選択された動作に対応する第2のイベントとを含む複数のイベントを順次実行することを指示する指示情報を生成するように制御する制御手段と、を有する。

【選択図】図8



【特許請求の範囲】**【請求項 1】**

アプリケーションソフトウェアで表示される複数の画面のうちいずれかを選択する第1の選択手段と、

前記第1の選択手段で選択された画面に対応づけて記憶された動作のうちいずれかを選択する第2の選択手段と、

前記第1の選択手段で選択された画面を表示する第1のイベントと、前記第2の選択手段で選択された動作に対応する第2のイベントとを含む複数のイベントを順次実行することを指示する指示情報を生成するように制御する制御手段と
を有することを特徴とする情報処理装置。

10

【請求項 2】

前記第1の選択手段は、前記アプリケーションソフトウェアを作成する際に定義された複数の画面にそれぞれ対応する複数の選択肢のうちいずれかを選択する操作に応じて選択することを特徴とする請求項1に記載の情報処理装置。

【請求項 3】

前記制御手段は、前記第1の選択手段によって前記複数の画面のうちいずれかが選択されると、選択された画面に対応づけて記憶された少なくとも1つの動作がある場合に、前記少なくとも1つの動作を前記第2の選択手段での選択肢として表示するように制御することを特徴とする請求項1または2に記載の情報処理装置。

20

【請求項 4】

生成される前記指示情報において前記複数のイベントを実行する順序を指定する操作を受け付ける第1の受付手段を更に有することを特徴とする請求項1乃至3のいずれか1項に記載の情報処理装置。

【請求項 5】

前記第2の選択手段で選択された動作が、特定の処理を実行した結果に応じた表示を行うことを含む動作である場合にも、前記指示情報に基づいて実行される前記第2のイベントは、前記特定の処理を実行することなく予め定められた表示を行うイベントであることを特徴とする請求項1乃至4のいずれか1項に記載の情報処理装置。

【請求項 6】

前記第1の選択手段で選択された画面に含まれる表示要素に対応づけた付加情報の入力を受け付ける第2の受付手段を更に有し、

30

前記複数のイベントは、前記第2の受付手段で入力を受け付けた付加情報を表示する第3のイベントをさらに含むことを特徴とする請求項1乃至3のいずれか1項に記載の情報処理装置。

【請求項 7】

前記制御手段は、前記第1の選択手段によって前記複数の画面のうちいずれかが選択されると、選択された画面に対応づけて生成された前記指示情報がある場合に、当該指示情報に基づいて、前記選択された画面と、前記第3のイベントで表示される付加情報を表示するように制御することを特徴とする請求項6に記載の情報処理装置。

40

【請求項 8】

前記付加情報は文字列であることを特徴とする請求項6または7に記載の情報処理装置。

【請求項 9】

前記付加情報は、対応する表示要素に対する吹き出しに表示する情報であることを特徴とする請求項6乃至8のいずれか1項に記載の情報処理装置。

【請求項 10】

前記第1の選択手段で選択された画面に含まれる入力領域であって、前記指示情報に対応する本番用の画面においてはユーザからのデータ入力を受け付ける入力領域に対して自動的に表示するデータの入力を受け付ける第3の受付手段を更に有し、

50

前記複数のイベントは、前記第3の受付手段で入力を受け付けたデータを表示する第4

のイベントを更に含むことを特徴とする請求項 1 乃至 9 のいずれか 1 項に記載の情報処理装置。

【請求項 1 1】

前記指示情報は、前記指示情報に含まれる最初のイベントの実行の後は、前記複数のイベントをユーザからの操作が無くとも自動的に順次実行することを指示する情報であることを特徴とする請求項 1 乃至 1 2 のいずれか 1 項に記載の情報処理装置。

【請求項 1 2】

前記指示情報を反映するプログラムのソースコードを生成する生成手段を更に有することを特徴とする請求項 1 乃至 1 1 のいずれか 1 項に記載の情報処理装置。

【請求項 1 3】

アプリケーションソフトウェアで表示される複数の画面のうちいずれかを選択する第 1 の選択ステップと、

前記第 1 の選択ステップで選択された画面に対応づけて記憶された動作のうちいずれかを選択する第 2 の選択ステップと、

前記第 1 の選択ステップで選択された画面を表示する第 1 のイベントと、前記第 2 の選択ステップで選択された動作に対応する第 2 のイベントとを含む複数のイベントを順次実行することを指示する指示情報を生成するように制御する制御ステップとを有することを特徴とする情報処理装置の制御方法。

【請求項 1 4】

コンピュータを、請求項 1 乃至 1 2 のいずれか 1 項に記載された情報処理装置の各手段として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、開発中のアプリケーションソフトウェアの動作を仮想的に実行する情報処理装置、情報処理方法およびプログラムに関する。

【背景技術】

【0 0 0 2】

顧客の為のアプリケーションソフトウェアを作成する場合、顧客の要望を汲み取るために、顧客とのミーティングを重ねてアプリケーションソフトウェアを作成することになる。

【0 0 0 3】

特に、アプリケーションソフトウェアの画面や操作手順は、顧客と開発者とで適宜情報共有され、完成されるアプリケーションソフトウェアのイメージを一致させるべく、プレビューを繰り返すことがある。なお、以下、アプリケーションソフトウェアを省略してアプリケーションと称することもあるものとする。

【0 0 0 4】

アプリケーションソフトウェアのプレビューの際に有効な手段が、プロトタイプ開発されたアプリケーション（プロトタイプアプリケーション）である。

【0 0 0 5】

プロトタイプアプリケーションとは、開発段階の本番用アプリケーションソフトウェアを顧客に開示するためのアプリケーションであり、あたかも本番用アプリケーションソフトウェアが動作しているかのように画面遷移させることができるアプリケーションである。プロトタイプアプリケーションは、顧客が操作する部品操作（テキスト入力や G U I ボタン押下などの画面部品に対する操作）を受け付け、本番用アプリケーションソフトウェアと同じように画面遷移を行う。但し、部品操作に対応するデータの検索や結果の出力などは行わず、予め設定された出力結果などを表示する。画面部品に対する操作に対応するデータの検索や結果の出力などは行わないため、開発者はサーバサイドや本番用アプリケーションソフトウェア内でのビジネスロジックを生成する前に、顧客の望むアプリケーションソフトウェアのイメージを確認することができる。顧客と開発者が本番用アプリケー

10

20

30

40

50

ションソフトウェアの完成イメージを共有できるため、両者での完成イメージの齟齬を少なくさせ、開発の手戻りを減らすことができる。また、MVP (Minimum Viable Product : 実用最小限の製品) による開発時には、プロトタイプアプリケーションは、顧客からのフィードバックを早いタイミングで得られるため、不必要的機能の開発に工数を使わず、開発工数の削減に寄与できる。

【0006】

本番用アプリケーションの提供元の人材（開発者や営業担当など）と顧客とが直接会って行う対面でプロトタイプアプリケーションを用いた説明を行う状況のほかに、非対面で、提供元の人材からの直接の説明無しに顧客にプロトタイプアプリケーションを披露する状況もある。非対面で説明なしとして効率的に顧客にプロトタイプアプリケーションを披露できる一方、プロトタイプアプリケーションを顧客に頒布するだけでは、提供元の人材からの説明がない分、顧客はプロトタイプアプリケーションの操作方法がわからないといったことや、十分な理解ができないことも考えられる。そこでプロトタイプアプリケーションを顧客に頒布する際に説明書類を添えたり、わかりやすい動画サイトや動画アプリケーションを提供することが考えられる。

10

【0007】

特許文献1は、アプリケーションのリプレイ動画像を作成するための機能を汎用化し、ユーザの所望の部分のリプレイ動画像を作成でき、他のユーザが閲覧できるようにすることが記載されている。

20

【先行技術文献】

【特許文献】

【0008】

【特許文献1】特開2014-106648号公報

【発明の概要】

【発明が解決しようとする課題】

【0009】

特許文献1のように動画として本番用アプリケーションの動作を見れるようすれば、顧客（本番用アプリケーションを提供される側のユーザ）は、本願用のアプリケーションにおける複数のイベントの実行がどのように行われるのかを分かりやすく理解することができる。例えば、画面の初期表示という第1のイベントの後に、GUIボタンの操作という第2のイベントが発生して画面遷移する様子などを、動画を見ることで確認することができる。しかしながら、特許文献1の場合、アプリケーションの任意の部分のリプレイ動画像を作成するにあたって、本番用アプリケーションソフトウェアが動作可能な状態まで完成しなければリプレイ動画像を作成できない。すなわち、本番用アプリケーションソフトウェア開発中は、サーバサイドのプログラムや本番用アプリケーションソフトウェア内のビジネスロジックが完成されていないことがあり、リプレイ動画像は確実に作成できるとは限らない。すなわち、本番用のアプリケーションソフトウェアにおける複数のイベントの実行を仮想的に見ることのできるコンテンツを容易に生成することができないという課題があった。

30

【0010】

そこで本発明は、本番用のアプリケーションソフトウェアにおける複数のイベントの実行を仮想的に見ることのできるコンテンツを容易に生成できるようにすることを目的とする。

40

【課題を解決するための手段】

【0011】

本発明の情報処理装置は、

アプリケーションソフトウェアで表示される複数の画面のうちいずれかを選択する第1の選択手段と、

前記選択手段で選択された画面に対応づけて記憶された動作のうちいずれかを選択する第2の選択手段と、

50

前記第1の選択手段で選択された画面を表示する第1のイベントと、

前記第2の選択手段で選択された動作が示す特定の処理に対応するイベントであって、前記特定の処理を実行することなく行われる第2のイベントと

を含む複数のイベントを順次実行することを指示する指示情報を生成するように制御する制御手段と

を有することを特徴とする。

【発明の効果】

【0012】

本発明によれば、本番用のアプリケーションソフトウェアにおける複数のイベントの実行を仮想的に見ることのできるコンテンツを容易に生成することができる。

10

【画面の簡単な説明】

【0013】

【図1】本実施形態に係るシステム構成図である。

【図2】本実施形態に係るプログラム開発装置、実行サーバ、データベースサーバ、クライアント装置として適用可能なハードウェア構成の一例を示すブロック図である。

20

【図3】プロトタイプアプリケーション生成処理のフローチャートである。

【図4】プロトタイプアプリケーション実行処理のフローチャートである。

【図5】再生モジュール生成処理のフローチャートである。

【図6】再生情報定義受付処理のフローチャートである。

【図7】再生モジュール実行処理のフローチャートである。

20

【図8】プロトタイプアプリケーション生成処理に係る表示例である。

【図9】プロトタイプアプリケーション生成処理に係る表示例である。

【図10】プロトタイプアプリケーション生成処理に係る表示例である。

【図11】プロトタイプアプリケーション実行処理に係る表示例である。

【図12】プロトタイプアプリケーションの一部の例である。

【図13】プロトタイプアプリケーションの一部の例である。

【図14】再生情報定義受付処理に係る表示例である。

【図15】再生情報定義受付処理で生成される再生情報定義の例である。

【図16】再生モジュール実行処理に係る表示例である。

【図17】再生モジュール実行処理に係る表示例である。

30

【発明を実施するための形態】

【0014】

以下、図面を参照して本発明の実施形態を詳細に説明する。

<第一実施形態>

図1は、本実施形態のシステム構成図である。本システムには、プログラム開発装置101(開発者がWebアプリケーション生成のために操作する情報処理装置)、実行サーバ102、データベースサーバ103、クライアント装置104を含む。

【0015】

プログラム開発装置101は、開発者の操作に従って画面レイアウト及びデータベース検索指示などを定義する。プログラム開発装置101は、プログラム生成、アプリケーション生成を行う。

【0016】

なお、本実施形態においては、プログラム開発装置101で生成するアプリケーションソフトウェアはWebアプリケーションとしたが、これに限定するものではなく、携帯電話・スマートフォン・タブレットなどの情報処理装置で動作するアプリケーションソフトウェアや組込みソフトウェアなど、Web技術による通信を利用したアプリケーションソフトウェアでなくてもよい。

【0017】

実行サーバ102は、プログラム開発装置101で開発されたアプリケーションソフトウェアを実行する。また、データベースサーバ103と接続して動作することが可能であ

40

50

る。

【0018】

データベースサーバ103は、開発されたアプリケーションソフトウェアが使用するデータベースであり、また本実施形態におけるアプリケーションソフトウェアの開発時にも動作確認などのために利用してもよい。例えば、開発者が利用するためにデータベースサーバ103は、プログラム開発装置101や、実行サーバ102と同一の装置で構成されてもよいし、LANなどのネットワーク105内に配置されてもよい。

【0019】

クライアント装置104（情報処理装置）は、実行サーバ102と協調してプログラム開発装置101で開発したアプリケーションプログラムを動作させる、エンドユーザの入力端末である。この、クライアント装置104は、携帯端末などの情報処理装置であってもよいこととする。

【0020】

なお、プログラム開発装置101、実行サーバ102、データベースサーバ103、および、クライアント装置104の何れかを、クラウドなどのインターネット上に配置してもよいし、いくつかの情報処理装置を一つの筐体としてもよい。

【0021】

図2は、本発明に係わるプログラム開発装置101、実行サーバ102、データベースサーバ103、クライアント装置104として適用可能な各ハードウェア構成の一例を示すプロック図である。

【0022】

図2において、CPU201は、システムバス204に接続される各デバイスを統括的に制御する。

【0023】

また、ROM203あるいは外部メモリ211には、CPU201の制御プログラムであるオペレーティングシステム(OS)や、各サーバ、クライアント、装置など情報処理装置の後述する各種機能を実現するためのプログラムが記憶されている。

【0024】

RAM202は、CPU201の主メモリ、ワークエリア、一時待避領域等として機能する。

【0025】

入力コントローラ205は、入力部209からの入力を制御する。この入力部209としては、情報処理装置では、キーボード、マウス等のポインティングデバイス、タッチパネルが挙げられる。

【0026】

なお、入力部209がタッチパネルの場合、ユーザがタッチパネルに表示されたアイコンやカーソルやボタンに合わせて押下（指等でタッチ）することにより、各種の指示を行うことができることする。

【0027】

また、タッチパネルは、マルチタッチスクリーン等の、複数の指でタッチされた位置を検出することができるタッチパネルであってもよい。

【0028】

出力コントローラ206は、出力部210の表示を制御する。この出力部210としては、例えば、CRTや液晶ディスプレイ等が挙げられる。尚、本体と一体になったノート型パソコンのディスプレイも含まれるものとする。また、プロジェクタであってもよいことする。

【0029】

外部メモリコントローラ207は、ポートプログラム、各種のアプリケーションソフトウェア、フォントデータ、ユーザーファイル、編集ファイル、プリンタドライバ等を記憶する外部メモリ211へのアクセスを制御する。外部メモリ211には、各サーバ、クラ

10

20

30

40

50

イアント、装置等の各種機能を実現するための各種テーブル、パラメータが記憶されている。この外部メモリ 211 としては、ハードディスク (HD) やフレキシブルディスク (FD)、PCMCIA カードスロットにアダプタを介して接続されるコンパクトフラッシュ (登録商標)、スマートメディア等が挙げられる。

【0030】

なお、CPU201 は、例えば RAM202 内の表示情報用領域へアウトラインフォント展開 (ラスタライズ) 処理を実行することにより、出力部 210 上での表示を可能としている。また、CPU201 は、出力部 210 上の不図示のマウスカーソル等でのユーザ指示を可能とする。

【0031】

通信 I/F コントローラ 208 は、ネットワークを介して外部機器との通信制御処理を実行する。例えば、TCP/IP を用いた通信等が可能である。

【0032】

本発明を実現するためのプログラム 212 は外部メモリ 211 に記録されており、必要に応じて RAM202 にロードされることにより CPU201 によって実行されるものである。

【0033】

次の図 3 から図 4 を参照して、Web アプリケーションソフトウェア (本番用アプリケーション) のプロトタイプアプリケーションを生成する手順を説明する。図 3 ~ 図 4 の処理は、プログラム開発装置 101 の CPU201 が、ROM203 に記録され、RAM202 に展開されたプログラムを実行することにより各処理を実行する。

【0034】

図 3 は、Web アプリケーションソフトウェア (以下、Web アプリケーションと略す) の生成処理のうち、プロトタイプアプリケーション生成処理に関する部分のフローチャートの一例を示す図である。

【0035】

S301 では、プログラム開発装置 101 の CPU201 は、開発者により Web アプリケーションの画面定義の入力を受け付ける。この処理は、生成される Web アプリケーションの画面のどの位置にどのような画面部品を配置 (レイアウト) するか、各画面部品にどのような機能を持たせるかの指示をユーザから受け付ける処理である。図 8 を参照して、S301 の処理を説明する。

【0036】

図 8 (a) の画面 800 の左側にある部品パレット部 801 は画面に配置する部品 (GUI 部品) の一覧であり、ユーザはこの中から WEB アプリケーションの画面に配置する GUI 部品を選択する。右側の画面定義エディタ部 802 は Web アプリケーションの画面に表示される表示画面を編集する領域である。図 8 は、ユーザ ID と名前をデータベースなどに登録させる処理を実行する Web アプリケーションを生成している例である。

【0037】

たとえば、開発者が Web アプリケーションの表示画面にボタンアイコン 805 を配置する場合、部品パレット部 801 内のボタン部品を矢印 803 のようにドラッグ & ドロップして配置する。同様にテキスト入力が可能なテキストボックス 806 を図示の位置に配置することにより、画面部品配置例 804 が生成される。なお、画面部品の配置方法は、部品パレット部 801 から画面定義エディタ部 802 へのドラッグ & ドロップに限らず、既に配置した部品を移動する方法や、既に配置した部品をコピー & ペーストにより複製する方法であってもよい。また、一度配置した部品を削除できてもよい。

【0038】

画面定義エディタ部 802 に配置された部品の中でアクションイベントを含む部品 (たとえば、ボタンアイコン 805 など) がある場合は、ボタンに対するアクションを登録する。具体的には、ボタンアイコン 805 において、マウスの右クリックもしくはマウスオーバーで所定時間経過すると、アイコン 811 を表示する。表示されたアイコン 811 をユ

10

20

30

40

50

ーザがクリックすると登録画面 812（登録ダイアログ）を表示する。図 8（b）は、表示された登録画面 812 に、ユーザが「名前」項目として「登録」と入力し、「タイプ」項目としてドロップダウンメニューから「ボタン」、「次画面」項目として、ドロップダウンメニューから「ユーザー覧画面」を選択している例である。すなわち、図 8 の登録画面 812 は、「登録」と言う名前のボタンアイコン 805 が押下されると、「ユーザー覧画面」に遷移するという内容を登録した例である。この画面で生成される画面の JSON データの例を図 12 を参照して説明する。

【0039】

図 12 は、図 8 の画面の JSON データの抜粋した例である。図 12 の JSON コードの内、枠 1204 で囲まれた “instance003” が、ボタンアイコン 805 を示す 10 データである。枠 1201 内には、“code” というキーに対して、登録を意味する “register” と言う値を持ち、“name” というキーには、図 8（b）の登録画面 812 で入力した “登録” と言う文字、“nextUI” というキーには、次のユーザー覧画面を指す “sampleUi002”（登録画面 812 では、ユーザー覧画面）が設定されている。枠 1202 内の “actions” というキーに対応する値には、本番用 Web アプリケーションでボタンアイコン 805 を押下すると実行される処理が記述される。枠 1203 内の “examples” というキーに対応する値には、プロトタイプアプリケーションでボタンアイコン 805 を押下すると実行される処理が記述される。どちらの場合も、図示の例では Web アプリケーションの作成途中であるため、ボタンアイコン 805 の押下時は画面遷移しか設定されていない。そのため、枠 1205 内に示すように “instance003” がキーの “onClickListener” 時の処理は何もしない “null” と設定されている。図 3 のフローチャートの説明に戻る。

【0040】

図 3 の S302 において、プログラム開発装置 101 の CPU201 は、プロトタイプアプリケーション実行時に画面に表示するプロトタイプデータの入力要求があったか否かを判定する。プロトタイプデータとは、プロトタイプアプリケーションを実行する際に Web アプリケーション画面に表示されるデータであり、開発者により、例として予め設定されている値や図である。つまり、プロトタイプデータとは、本番環境のようにデータベースからデータを検索したり、算出したデータ結果から作図をしたりするものではない。プロトタイプデータは、アプリケーション開発ツールを介して、開発者によって予め入力設定または配置される。S302 において、プロトタイプデータを入力する要求があった場合には S303 へと遷移し、プロトタイプデータを入力する要求がない場合は、S304 へと処理を遷移する。図 8（c）を参照して、ユーザからプロトタイプデータの入力要求がある場合の画面表示例を説明する。

【0041】

図 8（c）は、プロトタイプデータ入力要求時の画面表示例である。画面 820 のボタンアイコン 805 にマウスオーバーした状態で、マウスの右クリックもしくはマウスオーバーで所定時間経過すると、アイコン 821a を表示する。表示されたアイコン 821a をユーザがクリックすると選択肢 821b を表示する。選択肢 821b は「クリックされた時」と表記された選択肢である。選択肢 821b を選択して押下（マウスクリック）することによりプロトタイプデータの入力要求がなされる。プロトタイプデータが入力されると、本番用アプリケーションの提供側人材（開発者や営業担当など）がデモなどでプロトタイプアプリケーションを実行した際に、ボタンアイコン 805 を押下すると、同じ画面にプロトタイプデータを表示したり、遷移先画面にプロトタイプデータを表示した画面を表示することができる。本実施形態では、遷移先画面（図 9）にプロトタイプデータが表示される例を説明する。なお、アイコン 821a と同じ機能はアイコン 822 を押下しても可能である。アイコン 822 を押下した場合は、図 9（a）の領域 906 に選択肢 821b と同じ選択肢を表示する。図 3 のフローチャートの説明に戻る。

【0042】

図 3 の S303 において、プログラム開発装置 101 の CPU201 は、S302 で指

定されたアクション（選択された選択肢 8 2 1 b、すなわち、「クリックされた時」が示すアクションであるボタンアイコンの押下）により表示されるプロトタイプデータの入力を受け付ける。プロトタイプデータの入力を受け付ける処理を、図 9、図 10 を参照して説明する。

【 0 0 4 3 】

図 9、図 10 は、プロトタイプデータを入力する処理を説明する画面例である。ボタンアイコン 8 0 5 の次の遷移画面が、図 8 (a) の画面定義エディタ部 8 0 2 に示したようなユーザ登録画面ではなく、ユーザー一覧画面であるため、選択肢 8 2 1 b が押下されると、次の遷移画面である登録者リスト 9 0 2（ユーザー一覧画面のこと）を含む図 9 (a) の画面 9 0 0 に切り替わる。画面 9 0 0 は、プロトタイプデータを入力する画面であるため、画面のタイトルは部品パレット部 8 0 1 に示した「部品パレット」から図 9 (a) のプロトタイプ設定部 9 0 1 に示すように「プロトタイプ設定」に変わっている。登録者リスト 9 0 2 が、ボタンアイコン 8 0 5 が押下された際に遷移する遷移先画面（ユーザー一覧画面）であり、ユーザにより予め生成されている画面である。登録者リスト 9 0 2 は、図 8 (a) の画面部品配置例 8 0 4 に含まれる各テキストボックスに入力された ID や名前がリスト形式で表示されるものである。登録者リスト 9 0 2 には、ID 表示領域 9 0 3 、名前表示領域 9 0 4 、登録日時表示領域 9 0 5 が含まれる。複数のユーザが登録されている場合にはこれらがリスト形式で表示される。登録者リスト 9 0 2 は、前述した図 12 の JSON コードでは、枠 1 2 0 1 内の “nextUi” というキーに対応する “sampleUi002” に該当する。

【 0 0 4 4 】

図 9 (b)において、ユーザがプロトタイプデータを入力したい場合、入力する項目（画面の場合は名前表示領域 9 0 4 ）において、マウスの右クリックもしくはマウスオーバーで所定時間経過させる。それに応じて、アイコン 9 1 1 とアイコン 9 1 2 を表示する。表示されたアイコン 9 1 1 をユーザがクリックすると図 9 (c)の画面 9 2 0 に表示が切り替わり、プロトタイプデータ入力ダイアログ 9 2 1 が表示される。例えば図示のように、プロトタイプデータ入力ダイアログ 9 2 1 に「谷川 則之」という文字列（値）を入力して OK ボタンを押下することで、プロトタイプデータとして登録される。登録されたプロトタイプデータは、プロトタイプアプリケーションの実行時に、登録対象のアクション（ボタンアイコン 8 0 5 の押下）に応答して表示される遷移先画面において表示される。この結果、例えば、ボタンアイコン 8 0 5 の押下に応じて図 10 (a)の登録者リスト 9 0 2 の名前表示領域 9 0 4 に「谷川 則之」という文字列のように表示される。別の例として、プロトタイプデータを複数まとめて入力させたい場合の方法を説明する。図 9 (b)のアイコン 9 1 2 をユーザがクリックすると、図 10 (b)の表示に切り替わる。枠 1 0 2 1 で示すプロトタイプデータのリスト入力ダイアログに含まれるインポート欄 1 0 2 2 に、例えば CSV 形式でデータを入力して OK ボタン 1 0 2 4 を押下すると、それぞれ対応する値をプロトタイプデータとして登録できる。同様にファイルによる指定も可能であり、アイコン 1 0 2 3 を押下後、適切なファイルを選択して、データを入力できる。これらのプロトタイプデータを設定した JSON コードの例を図 13 を参照して説明する。

【 0 0 4 5 】

図 13 は、図 10 (b)に図示した入力例の入力が行われた後の JSON データの抜粋した例である。図 13 の JSON コードの内、枠 1 3 0 2 内の “actions” というキーに対応する値には、本番用 Web アプリケーションで動作、表示する処理が記述されている。枠 1 3 0 1 内の “examples” というキーに対応する値には、プロトタイプアプリケーション実施時に表示されるデータが登録されており、インポート欄 1 0 2 2 に登録されているデータが枠 1 3 0 1 内に登録されている。

【 0 0 4 6 】

図 11 (a)、(b)に、生成されたプロトタイプアプリケーションを実行した場合の表示例を示す。図 11 (a)の画面 1 1 0 0 、前述した画面定義エディタ部 8 0 2 に部品を配置する操作によって作成した画面である。図 11 (b)の画面 1 1 1 0 は、図 11 (

10

20

30

40

50

a) の登録ボタン 1101 (前述したボタンアイコン 805 に対応するボタン) が押下されたことに応じて表示される遷移先画面であり、前述した登録者リスト 902 に対応するものである。プロトタイプアプリケーション動作時は、遷移前の画面 1100 において ID 入力欄 1102 や名前入力欄 1103 にデータを入力してもしなくても、登録ボタン 1101 を押下すると、図 9 (b) の画面 1110 において、予め登録されているプロトタイプデータ 1111 が表示される。プロトタイプデータ 1111 は、前述した図 10 (b) のインポート欄 1022 で登録された値を反映した表示である。図 3 のフローチャートの説明に戻る。

【0047】

図 3 の S304 において、プログラム開発装置 101 の CPU201 は、画面定義 (部品の配置など) やプロトタイプデータを保存する要求 (図示しない保存ボタンの押下) があったかどうかを判断する。保存要求があった場合は S305 へと処理を遷移し、保存要求がない場合は S306 へと処理を進める。

【0048】

S305 において、プログラム開発装置 101 の CPU201 は、画面定義情報やプロトタイプデータを保存する。

【0049】

S306 において、プログラム開発装置 101 の CPU201 は、プロトタイプアプリケーション生成要求があったかを判定する。プロトタイプアプリケーション生成要求があったと判定した場合には、S307 に遷移する。一方、プロトタイプアプリケーション生成要求がなかった場合には、S301 に遷移する。

【0050】

S307 において、プログラム開発装置 101 の CPU201 は、プロトタイプアプリケーションのソースコードを生成する。ソースコード生成の際に、S305 で保存した画面定義や画面遷移定義、プロトタイプデータなどを読み込んでソースコードを生成する。

【0051】

S308 において、プログラム開発装置 101 の CPU201 は、S507 において生成したソースコードのコンパイルを行う。

【0052】

S309 において、プログラム開発装置 101 の CPU201 は、S508 においてコンパイルしたプロトタイプアプリケーションを実行サーバ 102 にデプロイする。こうして、プロトタイプアプリケーションが生成される。なお、生成されたプログラム (プロトタイプアプリケーション) は、クライアントで動作する SPA (Single Page Application) としてデプロイしても良い。図 4 以降では SPA としてデプロイされた例として説明する。以上で図 3 のフローチャートの説明を終了する。

【0053】

次に、図 4 を参照して、クライアント装置 104 がプロトタイプアプリケーション実行する際の処理の流れを説明する。

【0054】

図 4 のフローチャートは、前述した図 3 の処理によって生成されて実行サーバにデプロイされたプロトタイプアプリケーションに、クライアント装置 104 の Web ブラウザからユーザのアクセスがあった際に開始される処理の流れである。なお、以下で SPA と記載するものは、図 3 で生成されたプロトタイプアプリケーションであるものとする。

【0055】

S401 において、クライアント装置 104 の CPU201 は、実行サーバにデプロイされた SPA を取得し、SPA の実行画面をクライアント装置 104 の出力部 210 (ディスプレイ) に表示する。これによって前述した図 11 (a) のような表示が行われる。

【0056】

S402 において、クライアント装置 104 の CPU201 は、ユーザによるアクション実行要求を受付ける。具体的には、図 11 (a) の登録ボタン 1101 の押下などを受

10

20

30

40

50

け付ける。このとき、ユーザは本番用のWebアプリケーションを使用した際のイメージを再現するための動作イメージを確認することが目的であるため、先にも述べたようにID入力欄1102や名前入力欄1103にはユーザによる入力は必ずしも必要ではない。

【0057】

S403において、クライアント装置104のCPU201は、ユーザによるアクション実行要求があったかを判定する。具体的には、図11(a)の登録ボタン1101が押下されたかなどを判定する。アクション実行要求があった(登録ボタン1101が押下された)と判定した場合は、S404に遷移し、ユーザによるアクション実行要求がなかった場合は、S402へと処理を戻す。

【0058】

S404において、クライアント装置104のCPU201は、S402で受け付けたアクション実行要求が画面遷移を伴うかを判定する。アクション実行要求が画面遷移を伴うと判定した場合は、S406に遷移し、アクションが画面遷移を伴わない場合は、S407に遷移する。

【0059】

S405において、クライアント装置104のCPU201は、アクション実行要求に伴う遷移先の画面を表示する。

【0060】

S406において、クライアント装置104のCPU201は、取得したSPAに登録されているプロトタイプデータを表示する。具体的なプロトタイプデータの表示例が、図11(b)で前述した遷移先画面である画面1110である。以上で、図4のフローチャートの説明を終了する。

【0061】

次に、図5を参照して、プログラム開発装置101が本実施形態のプロトタイプアプリケーションを順序立てて表示させる(再生させる)ためのモジュール(以下、再生モジュールと称する)を生成する処理の流れについて説明する。生成される再生モジュールは、複数のイベントを仮想的に順次自動的に実行するモジュールである。すなわち、再生モジュールを実行すると複数のイベントの仮想的な実行により、動画のような自動的な画面遷移が行われる。仮想的なイベント実行は、本番環境とは異なり、本番同様のイベントを実行することなく、イベントに対応するプロトタイプデータの出力を実行する。この処理は、図3でプロトタイプアプリケーションが生成された後で実行されるフローチャートである。

【0062】

S501において、プログラム開発装置101のCPU201は、再生モジュールを生成するための再生情報定義受付処理(再生情報定義の設定・編集受け付ける処理)を実行する。詳細は図6において後述する。再生情報定義とは、再生情報定義受付処理でユーザから指示された複数のイベントを順次実行することを指示する指示情報である。

【0063】

S502において、プログラム開発装置101のCPU201は、S501で設定または編集された再生情報定義を保存する要求があったか否かを判定する。保存要求があれば、S503へと処理を遷移し、保存要求がなければ、S504へと処理を遷移する。

【0064】

S503において、プログラム開発装置101のCPU201は、再生情報定義を実行サーバ102に保存する。

【0065】

S504において、プログラム開発装置101のCPU201は、保存した再生情報定義に基づく再生モジュールを生成する要求があったか否かを判定する。再生モジュールの生成要求があれば、S505へと処理を遷移し、再生モジュール生成要求がない場合は、S501へと処理を戻す。

【0066】

10

20

30

40

50

S 5 0 5において、プログラム開発装置101のCPU201は、保存された再生情報定義に対応する再生モジュールのソースコードを生成し、S 5 0 6でコンパイルし、S 5 0 7で実行サーバ102にSPAとしてデプロイする。生成された再生モジュールは、実行サーバ102に記録された再生情報定義を参照して動作するように生成される。

【0067】

次に、図6を参照して、前述した図5のS 5 0 1の再生情報定義受付の処理の詳細を説明する。なお、後述する処理において、「イベント」は生成される再生モジュールの実行によって画面に変化を及ぼすものの総称であり、画面アクセス、アクション、吹き出しの表示、プロトタイプデータの表示を含むものとする。

【0068】

S 6 0 1において、プログラム開発装置101のCPU201は、図3のS 3 0 7で生成されたプロトタイプアプリケーションのソースコード（画面定義情報や部品情報、画面遷移情報、プロトタイプデータを含む）を読み込み、出力部210（ディスプレイ）に再生情報の編集画面を表示する。

【0069】

図14(a)に再生情報の編集画面（プロトタイプ動作記録の編集画面）の表示例を示す。

【0070】

UI定義選択エリア1410はプロトタイプアプリケーションまたは対応する本願用アプリケーションに定義されているUI定義（画面）を選択肢として表示する領域である。プロトタイプアプリケーションのソースコードから取得された少なくとも1つのUI定義が、UI定義選択エリアに一覧で表示される。図示の例では、UI定義選択エリア1410に、選択項目1411～1414が表示されている。選択項目1411は“MENU”であり、本番用アプリケーションにおける複数のメニュー項目が定義されたメニュー画面のUI定義を選択する項目である。選択項目1412は“ML_QUERY”であり、メーリングリストアドレスをデータベースなどから検索する画面のUI定義を選択する項目である。選択項目1413は“MEMBER_LIST”であり、参加しているメーリングリストのメンバを一覧で表示する画面のUI定義を選択する項目である。選択項目1414は“MEMBER_EDIT”であり、参加しているメーリングリストのメンバを編集する画面のUI定義の選択項目である。UI定義選択エリア1410に表示される選択肢は、本番用アプリケーションソフトウェアを作成する際に定義された複数の画面にそれぞれ対応する複数の選択肢である。

【0071】

アクション選択エリア1420は、UI定義選択エリア1410で選択されたUI定義に含まれるアクション（動作）を選択肢（選択項目）として表示する領域である。このように、プロトタイプアプリケーションのUI定義を取得することで、プロトタイプアプリケーションで実行するアクションを一覧で表示でき、後述するアクション選択が容易に可能となる。各アクションはそれぞれ、本番用アプリケーションで行われる特定の処理を示すものである。

プレビューエリア1430は、選択されたUI定義に定義されたGUI部品（表示要素）を用いて、選択されたUI定義に定義された画面が表示される領域である。また、再生順序指定エリア1440に吹き出しと値セットが登録されている場合には、これらもUI定義に定義された画面上に表示し、どのように吹き出しや値が表示されるかを確認できるようになっている。

【0072】

再生順序指定エリア1440は、再生モジュールを実行した場合に順次行われるイベントを上から下の順番で並べて表示した領域である。ユーザは、再生モジュールで実行させたいイベントを実行させたい順に再生順序指定エリア1440に上から下に並べて登録していくことで、所望の順序でイベントを自動実行させる再生モジュールを生成することができる。再生順序指定エリア1440に登録された各イベントは、プログラム開発装置1

10

20

30

40

50

01のRAM202に保持され、保存指示に応じて実行サーバ102に保存される。図15の再生情報定義1500と再生順序指定エリア1440に表示された各イベントはそれぞれ対応しており、画面アクセス1441は再生情報定義1500の枠1501内に、イベント1442～1445は、再生情報定義1500の枠1502～1505内にそれぞれ対応している。再生情報定義1500は、画面アクセス1441、イベント1442～1445を順次実行することを指示指示する指示情報である。

【0073】

図6のS602において、プログラム開発装置101のCPU201は、UI定義選択エリアに表示されたいずれかの選択肢が選択されたか否かを判定する。UI定義の選択肢がユーザにより選択された場合はS603へと処理を遷移し、そうでない場合はS602へと処理を戻し、UI定義が選択されるのを待つ。

【0074】

S603において、プログラム開発装置101のCPU201は、S602で選択されたUI定義に含まれるアクションを実行サーバ102から取得し、アクション選択エリア1420に表示する。より詳しくは、本番用アプリケーションの各UI定義のうち、S602で選択されたUI定義に含まれるアクションを取得する。図14(a)の例では、UI定義として選択されている“ML_QUERY”に、検索ボタンが含まれており、検索ボタンのクリックに応じた検索というアクション(動作)が定義づけられている。そのため、アクション選択エリア1420には、“ML_QUERY”的画面を初期表示するアクションの選択項目1421と、検索のアクションの選択項目1422とが表示される。

【0075】

S604において、プログラム開発装置101のCPU201は、S602のUI定義選択で選択されたUI定義(図14(a)の場合、選択項目1412の“ML_QUERY”)を画面アクセスとして設定してRAMに保持する。再生順序指定エリア1440に表示された画面アクセス1441に対応するイベントが、選択されたUI定義の画面を表示するイベントとなる。

【0076】

S605において、プログラム開発装置101のCPU201は、実行サーバ102から、選択されたUI定義に関連付けて記録された再生情報定義(選択されたUI定義が画面アクセスとして登録された再生情報定義)を取得し、取得した再生情報定義に従って、再生順序指定エリア1440の画面アクセス1441に続いて実行されるイベントとして表示する。選択されたUI定義に関連付けて、以前に再生順序の定義を行っていた場合は。これによって以前保存した再生順序が読み込まれ、それに対して編集を加えることができる。実行サーバ102に、選択されたUI定義に関連付く再生情報定義が記録されていない場合は、新規作成となり、このタイミングでは再生順序指定エリア1440には、画面アクセス1441だけが表示される。また、プレビューエリア1430には、選択されたUI定義に定義されたGUI部品と、再生順序指定エリア1440に表示された吹き出しと値セットがプレビュー表示される。

【0077】

S606において、プログラム開発装置101のCPU201は、画面アクセスエディタ1450を表示し、S602で選択されたUI定義を表示する。

【0078】

S607において、プログラム開発装置101のCPU201は、UI定義が変更されたか否かを判定する。UI定義が変更された場合には、S603へ進み、変更後のUI定義に基づく処理を行う。UI定義選択エリア1410で別の選択項目を選択される、あるいは画面アクセスエディタ1450で別のUI定義が選択されると、UI定義が変更されたと判定する。これによって、別のUI定義に関する再生情報を定義(設定)することができる。S607でUI定義が変更されたと判定されなかった場合はS608へと処理を遷移する。

【0079】

10

20

30

40

50

S 6 0 8において、プログラム開発装置101のCPU201は、再生順序指定エリア1440にイベントを追加する操作が行われたか否かを判定する。イベントを追加する操作は、アクション選択エリア1420でいずれかの選択項目を選択することによりアクションを追加する操作（クリックあるいは再生順序指定エリア1440へのドラッグ＆ドロップ）か、追加ボタン1446の押下であるものとする。イベントを追加する操作があった場合にはS 6 0 9に進み、そうでない場合はS 6 2 9に進む。

【0080】

S 6 0 9において、プログラム開発装置101のCPU201は、S 6 0 8で判定されたイベントの追加操作がアクションを追加する操作だったかを判定する。アクションを追加する操作であった場合はS 6 1 0へと処理を遷移し、そうでない場合はS 6 1 1へと処理を遷移する。

【0081】

S 6 1 0において、プログラム開発装置101のCPU201は、再生順序指定エリア1440に表示済み（登録済み）のイベントの最後尾に、選択されたアクションを追加する。例えば、アクション選択エリア1420の選択項目1422を再生順序指定エリア1440にドラッグ＆ドロップすることにより、再生順序指定エリア1440に、選択項目1422が示す「検索」というアクションをイベント1445として追加して表示する。

【0082】

アクションではないイベントの追加操作、すなわち追加ボタン1446の押下であった場合には、S 6 1 1において、プログラム開発装置101のCPU201は、追加対象の選択肢を表示する。選択肢には、「値セット」と「吹き出し」を含むものとする。

【0083】

S 6 1 2において、プログラム開発装置101のCPU201は、S 6 1 1で表示した選択肢のうち、「値セット」と「吹き出し」のいずれが操作されたかを判定する。「値セット」が選択された場合にはS 6 1 3にしすすみ、そうでない場合、すなわち「吹き出し」が選択された場合はS 6 2 1に進む。

【0084】

S 6 1 3において、プログラム開発装置101のCPU201は、再生順序指定エリア1440に表示済み（登録済み）のイベントの最後尾に「値セット」のイベント追加し、追加した「値セット」のイベントを選択状態にして表示する。再生順序に「値セット」のイベントは複数登録することができるため。図示のイベント1442のように、それぞれの値セットを識別するために、通し番号を付して表示される。

【0085】

S 6 1 4において、プログラム開発装置101のCPU201は、編集画面の右下部分（図14（a）において画面アクセスエディタ1450が表示されている領域）に値セットエディタ1460を表示する。図14（b）に値セットエディタ1460を表示した場合の値セットエディタ1460と再生順序指定エリア1440の表示例を示す。図14（b）は、編集画面の右側の一部分の表示例である。再生順序指定エリア1440では値セットのイベント1442が選択されており、値セットエディタ1460には、選択されているイベント1442の値をセットする対象のG U I部品を選択する対象部品設定欄1461、値を入力する値入力欄1462、再生順序指定エリア1440から選択中の値セットのイベントを削除する指示を受け付ける削除ボタン1463が表示される。

【0086】

S 6 1 5において、プログラム開発装置101のCPU201は、値をセットする対象部品を選択する操作があったか否かを判断する。対象部品を選択する操作があった場合はS 6 1 6へと処理を遷移し、そうでない場合はS 6 1 7へ進む。対象部品を選択する操作は、例えば、対象部品設定欄1461を操作することで表示されるプルダウンメニューに表示された選択肢のいずれかを選択することで行う。なお、ユーザが対象部品を選択する操作を行う前は、デフォルト（初期設定）で、選択されているU I定義画面に含まれるG U I部品のいずれかが選択されるものとする。

【0087】

S616において、プログラム開発装置101のCPU201は、S615で判定された対象部品を選択する操作に応じて、値をセットする対象部品を変更する処理を行う。具体的には、値セットエディタ1460の対象部品設定欄1461に変更後の対象部品名を表示し、値入力欄1462に、変更後の部品に既に入力されていた値を表示する。この値は、図15に例示した再生情報定義に記録された値を取得して表示する。再生定義情報に値として表示するデータが記録されていない新規のGUI部品である場合は、値入力欄1462の初期値は空欄となる。

【0088】

S617において、プログラム開発装置101のCPU201は、値入力欄1462に値を入力する操作（対象部品に対する値を入力する操作）があったか否かを判定する。値を入力する操作があった場合にはS618へ進み、そうでない場合にはS619へと処理を遷移する。値の入力は例えば、値入力欄1462を選択してキーボードなどで文字列を入力することにより行う。

【0089】

S618において、プログラム開発装置101のCPU201は、値入力欄1462に入力された値を再生情報定義に記録（登録）する。また、入力された値をプレビューエリア1430に表示された対象部品に表示する。図14（b）の例では、対象部品として「ML_ADDRESS」が選択され、値入力欄1462に「dev」と入力されているので、図15の再生情報定義1500の枠1502内のように情報が登録される。また、プレビューエリア1430では、対象部品であるGUI部品1431（「MLアドレス」）に値入力欄1462に入力された文字列「dev」が表示されている。

【0090】

S619において、プログラム開発装置101のCPU201は、削除ボタン1463が押下されたか否かを判定する。削除ボタン1463が押下された場合にはS620へ進み、そうでない場合はS608へ進む。

【0091】

S620では、プログラム開発装置101のCPU201は、再生順序指定エリア1440から選択されている値セットを削除する。同時に、図15の再生情報定義1500からも対応する情報を削除する。これによって、再生モジュールを実行した場合に、削除されたイベントに対応する画面遷移は行われなくなる。

【0092】

S621において、プログラム開発装置101のCPU201は、再生順序指定エリア1440に表示済み（登録済み）のイベントの最後尾に「吹き出し」のイベント追加し、追加した「吹き出し」のイベントを選択状態にして表示する。再生順序に「吹き出し」のイベントは複数登録することができるため。図示のイベント1443やイベント1444のように、それぞれの吹き出しを識別するために、通し番号を付して表示される。

【0093】

S622において、プログラム開発装置101のCPU201は、編集画面の右下部分（図14（a）において画面アクセスエディタ1450が表示されている領域）に値セット吹き出しエディタ1470を表示する。図14（c）に吹き出しエディタ1470を表示した場合の吹き出しエディタ1470と再生順序指定エリア1440の表示例を示す。図14（c）は、編集画面の右側の一部分の表示例である。再生順序指定エリア1440では吹き出しのイベント1443が選択されており、吹き出しエディタ1470には、選択されているイベント1443が示す吹き出しを表示する対象のGUI部品を選択する対象部品設定欄1471が表示される。また、値を入力する値入力欄1472、再生順序指定エリア1440から選択中の吹き出しのイベントを削除する指示を受け付ける削除ボタン1473が表示される。なお、本実施形態では、GUI部品に対して吹き出しという形態で付加情報（値入力欄1472に入力された文字列）を表示するイベントを追加できる例を説明したが、付加情報の表示形式は吹き出しに限るものではない。引出線を付けた先

10

20

30

40

50

に付加情報を表示したり、注釈番号を付して注釈を記載するなどでもよい。

【0094】

S623において、プログラム開発装置101のCPU201は、吹き出しを表示する対象部品を選択する操作があったか否かを判断する。対象部品を選択する操作があった場合S624へと処理を遷移し、そうでない場合はS625へ進む。対象部品を選択する操作は、例えば、対象部品設定欄1471を操作することで表示されるプルダウンメニューに表示された選択肢のいずれかを選択することで行う。なお、ユーザが対象部品を選択する操作を行う前は、デフォルト(初期設定)で、選択されているUI定義画面に含まれるGUI部品のいずれかが選択されるものとする。

【0095】

S624において、プログラム開発装置101のCPU201は、S623で判定された対象部品を選択する操作に応じて、吹き出しを表示する対象部品を変更する処理を行う。具体的には、吹き出しエディタ1470の対象部品設定欄1471に変更後の対象部品名を表示し、値入力欄1472に、変更後の部品に既に入力されていた吹き出しの値を図15の再生情報定義から読みだして表示する。新規に吹き出しが追加された部品であれば、値入力欄1472の初期値は空欄となる。

【0096】

S625において、プログラム開発装置101のCPU201は、値入力欄1472に値を入力する操作があったか否かを判定する。値を入力する操作があった場合にはS626へ進み、そうでない場合にはS627へと処理を遷移する。値の入力は例えば、値入力欄1472を選択してキーボードなどで文字列を入力することにより行う。

【0097】

S626において、プログラム開発装置101のCPU201は、値入力欄1472に入力された値を再生モジュールの生成に用いる再生情報定義に記録もしくは上書き保存する。また、入力された値をプレビューエリア1430に表示された吹き出し(選択された対象部品からの吹き出し)に表示する。図14(c)の例では、対象部品として「ML_ADDRESS」が選択され、値入力欄1472に「MLアドレスを前方一致で入力します」と入力されているので、図15の再生情報定義1500の枠1503内のように情報が登録される。また、プレビューエリア1430では、対象部品であるGUI部品1431(「MLアドレス」)に対して表示された吹き出し1418に、値入力欄1472に入力された文字列が表示されている。

【0098】

S627において、プログラム開発装置101のCPU201は、削除ボタン1473が押下されたか否かを判定する。削除ボタン1473が押下された場合にはS628へ進み、そうでない場合はS608へ進む。

【0099】

S628では、プログラム開発装置101のCPU201は、再生順序指定エリア1440から選択されている吹き出しを削除する。また、プレビューエリア1430からも該当する吹き出しを削除する。同時に、図15の再生情報定義1500からも対応する情報を削除する。これによって、再生モジュールを実行した場合に、削除されたイベントに対応する吹き出しの表示は行われなくなる。

【0100】

S629へと処理を遷移すると、プログラム開発装置101のCPU201は、再生順序指定エリア1440の画面アクセス1441を選択する操作があったか否かを判定する。画面アクセス1441が選択された場合は、S630へと処理を遷移し、そうでない場合はS631へと処理を遷移する。

【0101】

S630において、プログラム開発装置101のCPU201は、再生順序指定エリア1440の画面アクセス1441を選択状態で表示して、S606に遷移する。そして、S606で画面アクセスエディタ1450を表示する。すなわち、値セットエディタ14

10

20

30

40

50

60または吹き出しエディタ1470が表示されている場合に画面アクセス1441が選択されると、値セットエディタ1460または吹き出しエディタ1470に代わって画面アクセスエディタ1450が表示される。

【0102】

S631において、プログラム開発装置101のCPU201は、再生順序指定エリア1440に表示されたイベントのうち、値セットのイベントを選択する操作と、プレビューエリア1430に表示された値セットが表示されたGUI部品を選択する操作のいずれかがあったか否かを判定する。いずれかの操作があった場合には、S632において、再生順序指定エリア1440における、S631での選択に対応する値セットのイベントを選択状態で表示して、S614に遷移する。そして、S614で値セットエディタ1460を表示する。すなわち、画面アクセスエディタ1450または吹き出しエディタ1470が表示されている場合に値セットに対応するイベントが選択されると、画面アクセスエディタ1450または吹き出しエディタ1470に代わって値セットエディタ1460が表示される。10

【0103】

S633において、プログラム開発装置101のCPU201は、再生順序指定エリア1440に表示されたイベントのうち、吹き出しのイベントを選択する操作と、プレビューエリア1430に表示された吹き出しを選択する操作のいずれかがあったか否かを判定する。いずれかの操作があった場合には、S634において、再生順序指定エリア1440における、S633での選択に対応する吹き出しのイベントを選択状態で表示して、S623に遷移する。そして、S623で吹き出しエディタ1470を表示する。すなわち、画面アクセスエディタ1450または値セットエディタ1460が表示されている場合に吹き出しに対応するイベントが選択されると、画面アクセスエディタ1450または値セットエディタ1460に代わって吹き出しエディタ1470が表示される。20

【0104】

S635において、プログラム開発装置101のCPU201は、再生順序指定エリア1440に表示されたイベントのうち、アクションを削除する操作があったか否かを判定する。アクションを削除する操作があった場合は、S636へと処理を遷移し、そうでない場合はS637へと処理を遷移する。アクションを削除する操作は、例えば、ユーザがマウスで再生順序指定エリア1440のアクションのイベントを選択し、キーボードの【DEL】(デリートキー)などの押下する操作である。30

【0105】

S636において、プログラム開発装置101のCPU201は、再生順序指定エリア1440ら対象のアクションのイベントを削除する。同時に、図15の再生情報定義1500からも対応するイベントの情報を削除する。例えば再生順序指定エリア1440において検索のアクションであるイベント1445が削除されると、再生情報定義1500のうち枠1505内の情報を削除する。その後、S608へ進む。

【0106】

S637において、プログラム開発装置101のCPU201は、プレビューエリア1430に表示されている吹き出しのサイズや位置を変更する操作を受け付けたかを判断する。吹き出しのサイズか位置を変更する操作を受け付けると、S638へと処理を遷移し、そうでない場合は、S639へと処理を遷移する。具体的には、たとえば、ユーザがマウスでプレビューエリア1430に表示された吹き出しを選択し、吹き出しをマウスでドラッグ・アンド・ドロップすることで吹き出しの表示位置を変更することができる。吹き出しの吹き出し元である対象のGUI部品はこの操作では変更されない。また、プレビューエリア1430に表示された吹き出しの四隅に表示されるマークをマウスでドラッグ・アンド・ドロップする操作によって、吹き出しを拡大または縮小してサイズを変更することができる。40

【0107】

S638において、プログラム開発装置101のCPU201は、S637で受け付け

50

た操作に応じて、選択された吹き出しのサイズや位置を変更する。変更されたサイズ・位置を再生情報定義 1500 に記録する。その後、S608 へ進む。

【0108】

S639において、プログラム開発装置 101 の CPU201 は、再生順序の並び替え操作（順序の指定操作）を受け付けたか否かを判定する。並び替え操作があった場合には S640 へと処理を遷移し、そうでない場合には、S640 へと処理を遷移する。並び替え操作は、たとえば、ユーザがマウスで再生順序指定エリア 1440 内のいずれかのイベントを再生順序指定エリア 1440 内でドラッグ・アンド・ドロップで移動し、他のイベントとの順序を入れ替える操作である。

【0109】

S640において、プログラム開発装置 101 の CPU201 は、S639 で受け付けた操作に応じて、再生順序指定エリア 1440 に表示されるイベントの表示位置を並べ替える。変更された順序を再生情報定義 1500 に記録する（「再生順」の情報を、変更後の再生順序通りの情報に更新する）。その後、S608 へ進む。なお、最初に行うイベント（再生順序が 1 番目のイベント）は、画面アクセス 1441 から変更できないものとする。

【0110】

S641において、プログラム開発装置 101 の CPU201 は、図 4 (a) に示した編集画面を閉じて再生情報定義受付処理を終了させる指示があったか否かを判定する。たとえば、図 14 (a) の終了ボタン 1401 などが押下されると終了させる指示があったと判定する。終了させる指示があった場合は図 6 のフローチャートの処理を終える。そうでない場合には S642 へ進む。

【0111】

S642 では、プログラム開発装置 101 の CPU201 は、値セットエディタ 1460 が表示されているか否かを判定する。値セットエディタ 1460 が表示されている場合には S615 に進み、そうでない場合には S643 へ進む。

【0112】

S642 では、プログラム開発装置 101 の CPU201 は、吹き出しエディタ 1470 が表示されているか否かを判定する。吹き出しエディタ 1470 が表示されている場合には S623 に進み、そうでない場合には S607 へ進む。

【0113】

図 6、図 14 を用いて説明した処理により、再生モジュールの再生情報定義を容易に設定、編集することができる。

【0114】

次に、図 7 を参照して、クライアント装置 104 が図 5 で生成された再生モジュールを実行することによる再生処理のフローチャートを示す。

【0115】

S701において、クライアント装置 104 の CPU201 は、S309 で実行サーバ 102 にデプロイされた再生モジュールを含むプロトタイプアプリケーションの SPA を取得する。

【0116】

S702において。取得した SPA に含まれる再生モジュールの再生要求を受け付ける画面を表示する。この時の表示例を図 16 (a) に示す。

【0117】

S703において、クライアント装置 104 の CPU201 は、プロトタイプアプリケーション再生要求を受け付けたか否かを判定する。図 16 (a) のボタン 1601 に対する操作があった場合には再生要求があったと判定して S704 に進み、そうでない場合には S702 に戻る。なお、再生要求を受け付けない場合、図 16 (a) の例にある「検索」ボタンの押下に応じて、対応するプロトタイプデータを表示するといったように、通常のプロトタイプアプリケーションと同様に機能させることもできる。

10

20

30

40

50

【0118】

S704において、クライアント装置104のCPU201は、再生モジュールに対応する再生情報定義を実行サーバ102から取得する。

【0119】

S705において、取得した再生情報定義に定義されたイベントを順番に1つずつ取得する。最初にS705に遷移した場合は1番目のイベント、すなわち再生アクセスを取得する。再生アクセスは対象となる画面定義UIの画面を表示する処理である。

【0120】

S706では、クライアント装置104のCPU201は、S705で取得したイベントが画面遷移を伴うイベントであるか否かを判定する。画面定義を伴うイベントであればS707に進み、そうでない場合はS708に進む。

【0121】

S707では、クライアント装置104のCPU201は、S705で取得したイベントを実行する。取得したイベントが画面アクセスであれば、対象となるUI定義画面に従って画面を表示する。取得したイベントが吹き出しであれば、再生情報定義に記録された定義に従って、対象となるGUI部品に吹き出しを表示する。取得したイベントが値入力(値セット)であれば、対象となるGUI部品にプロトタイプデータまたは再生情報定義に従った値を表示する。取得したイベントがアクション(本番環境ではユーザの操作をトリガーに行われるイベント)であればそのアクションに対応するプロトタイプデータの表示を行う。アクションが画面遷移を伴うものであれば、S707で遷移先の画面に切り替えたうえで、アクションに定義されたプロトタイプデータを表示する処理を行う。

【0122】

S709では、クライアント装置104のCPU201は、S705で取得したイベントが、再生情報定義における順番が最後のイベントであったか否かを判定する。最後でないと判定した場合にはS706に進んで、次のイベントを取得して実行する。なお、S709からS705への遷移は、予め設定された所定時間(例えば2秒)が経過してから遷移するものとする。これによって、再生情報定義に定義された複数のイベントが所定時間おき(例えば2秒ごと)に順次実行される。なお、遷移待機時間であるこの所定時間は固定である例を説明したが。これに限るものではなく、前出した図14の編集画面などでユーザがイベント毎に可変に設定できるようにしてもよい。S709で最後の処理であると判定した場合は図7の処理を終了する。

【0123】

図16(b)～図16(d)、図17(a)、図17(b)に、前述した図15の再生情報定義1500に従った再生モジュールの実行が行われた場合の画面遷移例を示す。再生モジュールの実行(再生)が行われると、再生情報定義1500に従って、図16(b)が表示され、所定時間後に図16(c)に表示が遷移し、さらにその後に図16(d)、図17(a)、図17(b)という順で表示が自動的に遷移する。

【0124】

図16(b)は、再生情報定義1500の再生順が1番目である、枠1501内に記録された情報に基づく表示であり、UI_QUERYのUI画面定義に従った画面表示である。この表示は、SPAに記録された画面定義情報を参照して、UI_QUERYに定義された情報に基づいて行われる。

【0125】

図16(b)は、再生情報定義1500の再生順が2番目である、枠1502内に記録された情報に基づく表示である。より詳しくは、入力ターゲットであるML_ADDRESSで示されるGUI部品に対して、再生情報定義1500に記録された入力値である「Dev」という文字列を、枠1621内に示すように表示したものである。ML_ADDRESSで示されるGUI部品は、本番用アプリケーションではユーザからのデータ入力を受け付ける入力領域である。しかし、再生モジュールを実行した場合には、再生モジュールの再生を見ているユーザ(顧客)がデータ入力をしなくても自動的に「Dev」とい

10

20

30

40

50

う文字列が表示される。

【0126】

図16(d)は、再生情報定義1500の再生順が3番目である枠1503に記録された情報に基づく表示を行い、さらにその後、再生順が4番目である枠1504に記録された情報に基づく表示を行った例である。より詳しくは、枠1503が示す情報に従って、吹き出しターゲットであるML_ADDRESSで示されるGUI部品に対して、再生情報定義1500に記録された吹き出し文字列が、吹き出し1631に示すように表示されている。また、枠1504が示す情報に従って、吹き出しターゲットであるSEARCN_BUTTONで示されるGUI部品(検索ボタン1616)に対して、再生情報定義1500に記録された吹き出し文字列が、吹き出し1632に示すように表示されている。

10

【0127】

図17(a)は、再生情報定義1500の再生順が5番目である枠1505に記録された情報に基づく表示を行い、さらにその後、再生順が6番目として記録された情報(不図示)に基づく表示を行った例である。より詳しくは、枠1505が示す情報に従って、実行ターゲットであるSEARCN_BUTTONが示すGUI部品(検索ボタン1616)に対して、実行処理に示す操作であるクリックが行われた場合のアクションに対応づいて記録されたプロトタイプデータの表示を行う。なお、検索ボタン1616のクリックというアクションには、画面遷移が定義されていないため、S706ではNoと判定され、画面遷移をすることなく、検索ボタン1616のクリックというアクションに対応付けて記録されたプロトタイプデータ1701が表示される。なお、本番環境のアプリケーションであれば検索ボタンのクリックに応じて、データベースにアクセスして、ML_ADDRESSで示されるGUI部品に対して入力された文字列を検索キーとして検索する処理を行う。これに対し、この再生モジュールの実行においては、データベースにアクセスして、ML_ADDRESSで示されるGUI部品に対して入力された文字列を検索キーとして検索する処理は行わず、対応するプロトタイプアプリケーションを表示するという処理が行われる。その後、再生順が6番目であるイベントに従って吹き出しが表示される。図17(b)は、再生情報定義1500の再生順が6番目以降である情報に基づく表示を行った例である。より詳しくは、図17(a)でリンク先を示すプロトタイプデータのクリックというイベントを実行し、その後、吹き出しを表示するイベントを2つ実行した後の表示例である。リンク先のクリックというアクションは画面遷移を伴うものとして定義されているため、図7のS706はYesと判定され、S707で遷移先の画面である画面1710を表示する。その後、画面1710に、リンク先のクリックに対応づけて記録されているプロトタイプデータ1711を表示する。なお、本番環境のアプリケーションであればリンク先のクリックに応じて、クリックしたリンク先にアクセスする処理を行う。これに対し、この再生モジュールの実行においてはリンク先にアクセスする処理は行わず、対応するプロトタイプアプリケーションを表示するという処理が行われる。その後、吹き出しを表示するイベントを2つ実行すると図17(b)の表示となる。

20

30

40

以上の処理により、本番用のアプリケーションソフトウェアにおける複数のイベントの実行を仮想的に見ることのできるコンテンツ(再生モジュール)を容易に生成することができる。このコンテンツを顧客(本番用アプリケーションを提供される側のユーザ)が見ることで、顧客は、本願用のアプリケーションにおける複数のイベントの実行がどのように行われるのかを分かりやすく理解することができる。従って本番用アプリケーションの提供元の人材(開発者や営業担当など)が顧客に対して直接説明する手間を省き、効率的に顧客に本番用アプリケーションの動きを理解してもらうことが可能となる。さらに、開発途中で画面の仕様が変わる場合などでも、再生モジュールは変更があったUI定義を読み込んで実行するため、修正にかかる工数を減らすことができる効果を有する。さらに、再生モジュールに更なる編集(吹き出し、値の削除や追加、アクションの追加など)を行いたい場合にも、既存の再生情報定義を読み込んで編集をすればよいので、一から作り直す必要がなく、編集に係る手間を削減することができる。

【0128】

50

以上のように、前述した実施形態の機能を実現するプログラムを記録した記録媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはC P UやM P U）が記録媒体に格納されたプログラムを読み出し、実行することによっても本発明の目的が達成されることは言うまでもない。

【0129】

この場合、記録媒体から読み出されたプログラム自体が本発明の新規な機能を実現することになり、そのプログラムを記録した記録媒体は本発明を構成することになる。

【0130】

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、C D - R O M、C D - R、D V D - R O M、磁気テープ、不揮発性のメモリカード、R O M、E E P R O M、シリコンディスク等を用いることが出来る。 10

【0131】

また、コンピュータが読み出したプログラムを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムの指示に基づき、コンピュータ上で稼働しているO S（オペレーティングシステム）等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0132】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるC P U等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。 20

【0133】

また、本発明は、複数の機器から構成されるシステムに適用しても、ひとつの機器から成る装置に適用しても良い。また、本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適応できることは言うまでもない。この場合、本発明を達成するためのプログラムを格納した記録媒体を該システムあるいは装置に読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。 30

【0134】

上記プログラムの形態は、オブジェクトコード、インタプリタにより実行されるプログラムコード、O S（オペレーティングシステム）に供給されるスクリプトデータ等の形態から成ってもよい。

【0135】

さらに、本発明を達成するためのプログラムをネットワーク上のサーバ、データベース等から通信プログラムによりダウンロードして読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。なお、上述した各実施形態およびその変形例を組み合わせた構成も全て本発明に含まれるものである。 40

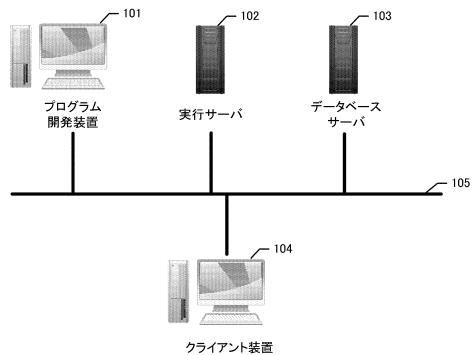
【符号の説明】

【0136】

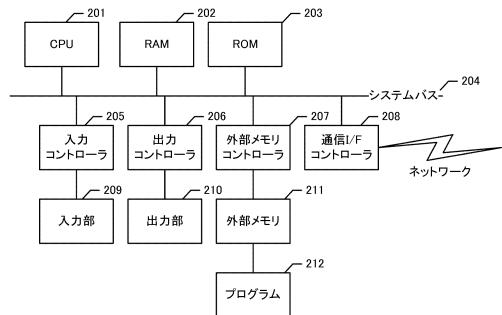
- 101 プログラム開発装置
- 102 実行サーバ
- 103 データベースサーバ
- 104 クライアント装置
- 105 ネットワーク

【図面】

【図1】



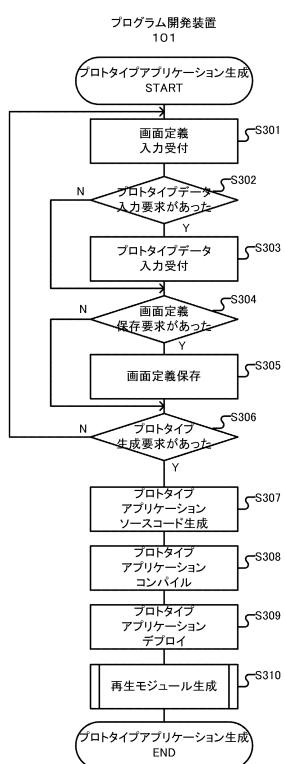
【図2】



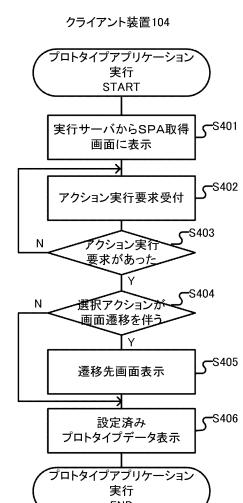
10

20

【図3】



【図4】

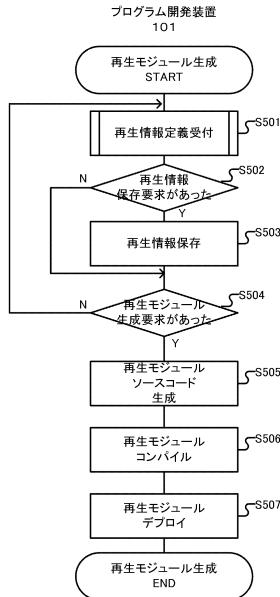


30

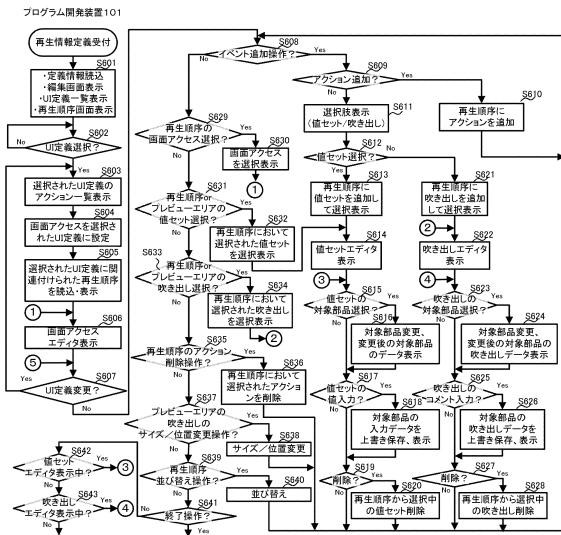
40

50

【図5】



【図6】



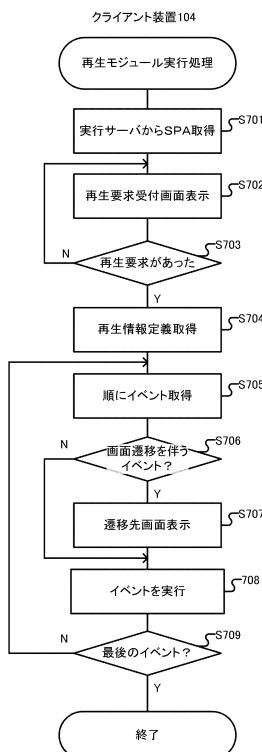
10

20

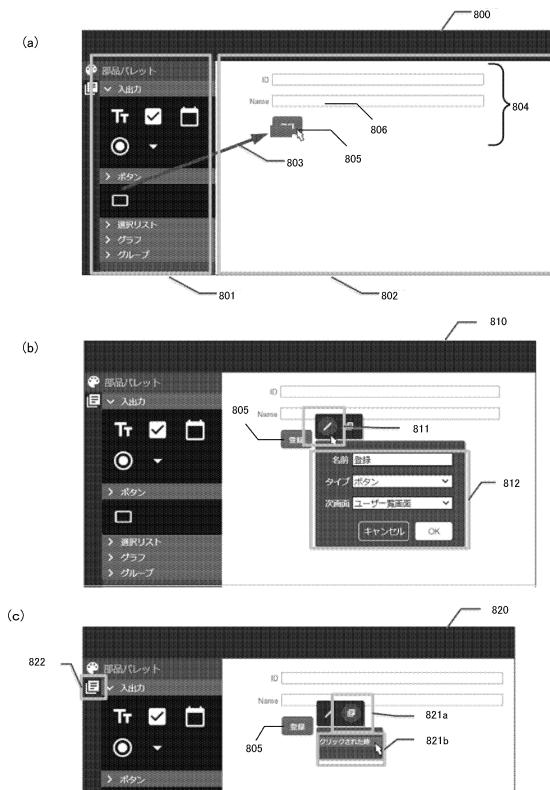
30

40

【図7】

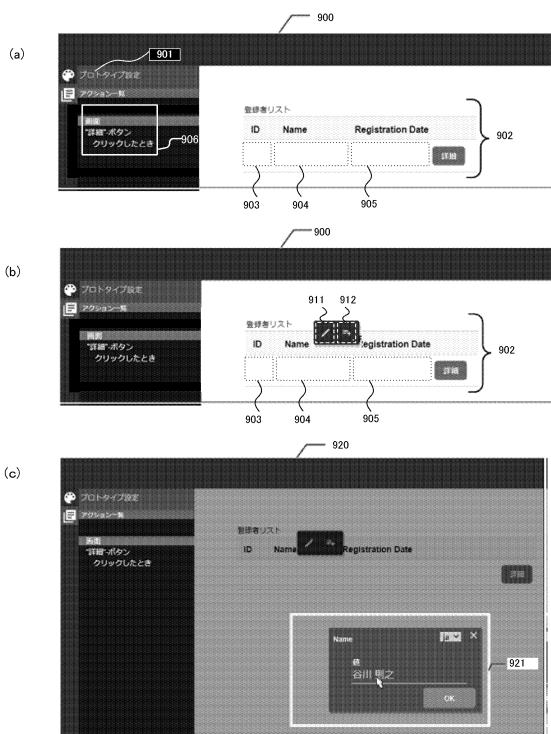


【図8】

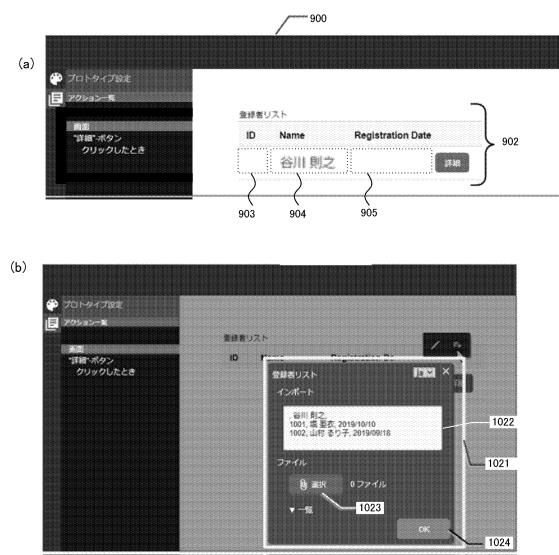


50

【図9】



【図10】



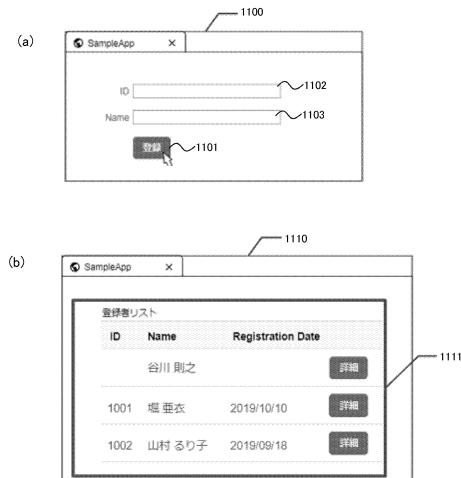
10

20

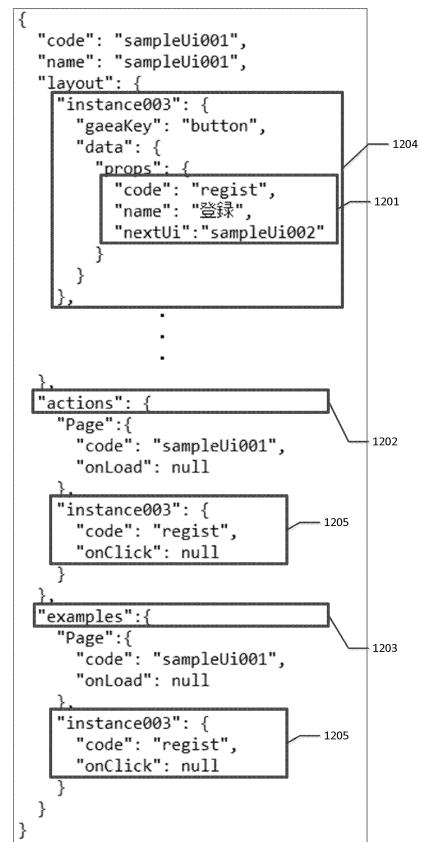
30

40

【図11】



【図12】



50

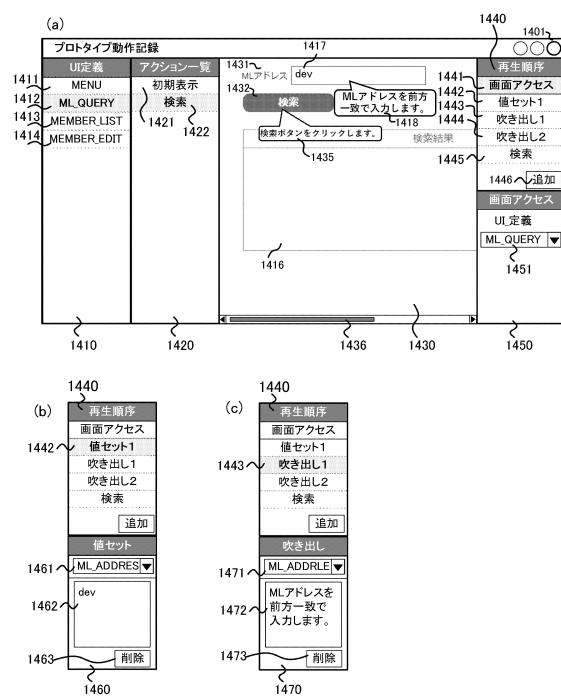
【図13】

```

{
  "code": "sampleUi002",
  "name": "sampleUi002",
  "layout": {
    "instance001": {
      "gakeKey": "list",
      "data": {
        "props": {
          "name": "userList",
          "name": "登録者リスト"
        }
      }
    },
    "children": [
      "instance002",
      "instance003",
      "instance004",
      "instance005"
    ]
  },
  "instance002": {
    "gakeKey": "output-text",
    "data": {
      "props": {
        "code": "id",
        "name": "ID"
      }
    }
  },
  .
  .
  .
  "actions": {
    "page": {
      "code": "sampleUi002",
      "onLoad": null
    },
    "instance003": {
      "code": "detail",
      "onClick": null
    }
  },
  "examples": {
    "Page": {
      "code": "sampleUi002",
      "onLoad": {
        "js": {
          "userList": [
            {"id": "", "name": "谷川 则之", "registrationDate": ""},
            {"id": "1001", "name": "堀 重次", "registrationDate": "2019/10/10"},
            {"id": "1002", "name": "山村 るい子", "registrationDate": "2019/09/18"}
          ]
        }
      }
    },
    "instance005": {
      "code": "detail",
      "onClick": null
    }
  }
}

```

【図14】



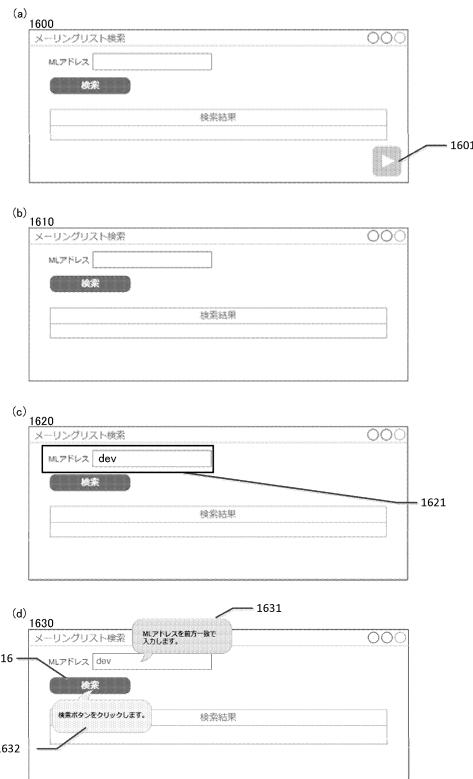
10

20

【図15】

1500 再生順 処理名 始点セッティング パラメータ	
1: 画面アクセス	アクセスターゲット ML_QUERY
2: 入力入力	入力ターゲット ML_ADDRESS
3: 入力値	入力値 dev
4: 検索出力文	検索出力ターゲット ML_ADDRESS
5: 検索出力文	検索出力文例 MLアドレスを前方一致で入力します。
6: 検索出力文	検索出力文例 検索ボタンをクリックします。
7: 実行結果	実行結果 クリック

【図16】



30

40

50

【図 1 7】



10



20

30

40

50

フロントページの続き

(72)発明者 高塚 剛

東京都港区港南2丁目16番6号 キヤノンITソリューションズ株式会社内

(72)発明者 辻 健太

東京都港区港南2丁目16番6号 キヤノンITソリューションズ株式会社内

F ターム(参考) 5B376 BC07 BC14 BC43