

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5330302号
(P5330302)

(45) 発行日 平成25年10月30日(2013.10.30)

(24) 登録日 平成25年8月2日(2013.8.2)

(51) Int. Cl. F 1
G 0 6 F 3 / 0 2 (2 0 0 6 . 0 1) G 0 6 F 3 / 0 2 3 6 0 G

請求項の数 3 (全 23 頁)

<p>(21) 出願番号 特願2010-72741 (P2010-72741) (22) 出願日 平成22年3月26日 (2010. 3. 26) (65) 公開番号 特開2011-204139 (P2011-204139A) (43) 公開日 平成23年10月13日 (2011.10.13) 審査請求日 平成23年9月27日 (2011. 9. 27)</p>	<p>(73) 特許権者 598057291 株式会社富士通エフサス 東京都港区浜松町一丁目5番1号 (74) 代理人 100103528 弁理士 原田 一男 (72) 発明者 吉田 恵介 東京都港区芝公園四丁目1番4号 株式会 社富士通エフサス内 審査官 篠原 功一</p>
--	--

最終頁に続く

(54) 【発明の名称】 操作記録プログラム及び装置

(57) 【特許請求の範囲】

【請求項1】

操作記録開始の指示に応じて又は起動後に、所定のキー以外のキー入力を禁止するステップと、

クリックに応じて、当該クリックの操作を再現させるためのスクリプトデータを生成し、データ格納部に格納するステップと、

前記所定のキーに含まれるテキスト入力用キーの押下又は別途用意されているテキスト入力用ボタンのクリックに応じて、テキスト入力画面を表示して、テキスト入力を受け付け、当該テキスト入力を再現させるためのスクリプトデータを生成し、前記データ格納部に格納するステップと、

前記所定のキーに含まれる定義参照入力キーの押下又は別途用意されている定義参照入力ボタンのクリックに応じて、文字列及び当該文字列に対応付けられている値の入力又は選択を受け付け、定義済みの文字列及び当該定義済みの文字列に対応付けられている値が格納されている定義ファイルから当該入力又は選択された文字列に対応付けられている値を読み出し且つ当該値を入力させるためのスクリプトデータを生成し、前記データ格納部に格納する定義参照スクリプト生成ステップと、

操作記録終了後に、前記データ格納部に格納されている前記スクリプトデータを含むファイルを出力するステップと、

を、コンピュータに実行させるための操作記録プログラム。

【請求項2】

前記定義参照スクリプト生成ステップが、
特定の文字列及び当該特定の文字列に対応付けられている値の入力に応じて、前記定義ファイルに、前記特定の文字列及び当該特定の文字列に対応付けられている値を格納するステップ

を含む請求項 1 記載の操作記録プログラム。

【請求項 3】

操作記録開始の指示に応じて又は起動後に、所定のキー以外のキー入力を禁止する手段と、

クリックに応じて、当該クリックの操作を再現させるためのスクリプトデータを生成し、データ格納部に格納する手段と、

前記所定のキーに含まれるテキスト入力用キーの押下又は別途用意されているテキスト入力用ボタンのクリックに応じて、テキスト入力画面を表示して、テキスト入力を受け付け、当該テキスト入力を再現させるためのスクリプトデータを生成し、前記データ格納部に格納する手段と、

前記所定のキーに含まれる定義参照入力キーの押下又は別途用意されている定義参照入力ボタンのクリックに応じて、文字列及び当該文字列に対応付けられている値の入力又は選択を受け付け、定義済みの文字列及び当該定義済みの文字列に対応付けられている値が格納されている定義ファイルから当該入力又は選択された文字列に対応付けられている値を読み出し且つ当該値を入力させるためのスクリプトデータを生成し、前記データ格納部に格納する定義参照スクリプト生成手段と、

操作記録終了後に、前記データ格納部に格納されている前記スクリプトデータを含むファイルを出力する手段と、

を有する操作記録装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータの操作記録を行うための技術に関する。

【背景技術】

【0002】

例えば、特開 2007 - 4734 号公報には、ユーザが過去に実行した操作を再現したり、過去のある時点から実施した一連の操作を繰り返し実行したりするための技術が開示されている。具体的には、本情報処理装置は、操作履歴を時間系列順に自動記録する機能と、時間系列上でユーザが指定した箇所へ遡って操作の自動再現・記憶する機能を有する。そして、操作履歴の内容を、ユーザがアプリケーション上から実施できる操作を機能毎に分類する。この機能分類では、再現時に内容変更をユーザに許可するものとし、変更を許可するものは、再現時にユーザの必要に応じて変更して再生するものである。しかしながら、異なる情報処理装置で、一部分だけ変更しつつ同様の操作を繰り返し実施するような場合を想定した工夫は見えない。

【0003】

また、特開 2004 - 78268 号公報には、同一あるいは類似した操作手順を繰り返すことが要求されるシステムにおいて、操作上のミス防止し、操作に必要な手間や時間を削減するための技術が開示されている。具体的には、スクリプトファイルと、スクリプトファイルに対するパラメータの値を定義する設定ファイルと、スクリプトファイルおよび設定ファイルを読み出し、スクリプトに指定された処理を実行するスクリプトエンジンとを用意する。そして、スクリプトファイルには、スクリプトが格納され、スクリプトは、スクリプトエンジンにより解析される指令とスクリプトの解釈を制御する制御情報とを含む情報の集合である。設定ファイルは、スクリプトによって使用されるパラメータの値を定義したファイルである。スクリプトが変更できない固定の情報であるのに対して、設定ファイルの情報は変更可能な情報である。スクリプトエンジンは、スクリプトファイル及び設定ファイルに従って、スクリプトエンジン以外のプログラムを制御する。しかしな

10

20

30

40

50

がら、スクリプトファイル及び設定ファイルをどのように生成するかについては考察されていない。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2007-4734号公報

【特許文献2】特開2004-78268号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

以上のように、異なるコンピュータにおいて、部分的に異なるがほぼ同様の操作を繰り返し実施するような場合、例えば、同一企業の複数のサーバのネットワーク設定を行うような場合に、より簡単に且つ適切に、使用するデータを生成する方法は開示されていない。

【0006】

従って、本発明の目的は、異なるコンピュータにおいて、部分的に異なるがほぼ同様の操作を繰り返し実施するためのデータを、より簡単に且つ適切に生成するための技術を提供することである。

【課題を解決するための手段】

【0007】

本発明の一側面に係る操作記録プログラムは、コンピュータに、(A)操作記録開始の指示に応じて又は起動後に、所定のキー以外のキー入力を禁止するステップと、(B)クリックに応じて、当該クリックの操作を再現させるためのスクリプトデータを生成し、データ格納部に格納するステップと、(C)所定のキーに含まれるテキスト入力用キーの押下又は別途用意されているテキスト入力用ボタンのクリックに応じて、テキスト入力画面を表示して、テキスト入力を受け付け、当該テキスト入力を再現させるためのスクリプトデータを生成し、データ格納部に格納するステップと、(D)所定のキーに含まれる定義参照入力キーの押下又は別途用意されている定義参照入力ボタンのクリックに応じて、文字列及び当該文字列に対応付けられている値の入力又は選択を受け付け、定義済みの文字列及び当該定義済みの文字列に対応付けられている値が格納されている定義ファイルから当該入力又は選択された文字列に対応付けられている値を読み出し且つ当該値を入力させるためのスクリプトデータを生成し、データ格納部に格納する定義参照スクリプト生成ステップと、(E)操作記録終了後に、データ格納部に格納されているスクリプトデータを含むファイルを出力するステップとをコンピュータに実行させるものである。

【0008】

このように、ユーザに対して操作記録中キー入力について特別に意識させた上で、装置毎に異なるキー入力部分については、定義ファイルを用いて入力を行うようなスクリプトデータを生成するようにしている。このようにすれば、クリックするだけの部分はユーザにとって簡単にスクリプトデータが生成されると共に、キー入力については通常のキー入力と装置毎に異なるキー入力部分とを分離しつつ、適切にスクリプトデータを生成することができるようになる。

【0009】

なお、上で述べた定義参照スクリプト生成ステップが、特定の文字列及び当該特定の文字列に対応付けられている値の入力に応じて、定義ファイルに、特定の文字列及び当該特定の文字列に対応付けられている値を格納するステップを含むようにしてもよい。

【0010】

このように、定義ファイルを生成しつつ、スクリプトデータを生成するようにしても良い。定義ファイルについては、ユーザがエディットして、スクリプトデータを含むファイルを実行させる装置毎にカスタマイズするようにしても良い。

【0011】

10

20

30

40

50

なお、上で述べたようなプログラムは、例えばフレキシブル・ディスク、CD-ROM、光磁気ディスク、半導体メモリ、ハードディスク等のコンピュータ読み取り可能な記憶媒体又は記憶装置に格納される。なお、処理途中のデータについては、コンピュータのメモリ等の記憶装置に一時保管される。

【発明の効果】

【0012】

本発明によれば、異なるコンピュータにおいて、部分的に異なるがほぼ同様の操作を繰り返し実施するためのデータを、より簡単に且つ適切に生成できるようになる。

【図面の簡単な説明】

【0013】

【図1】図1は、実施の形態におけるシステム概要図である。

【図2】図2は、操作記録プログラムのウィンドウ例を示す図である。

【図3】図3は、実施の形態におけるメイン処理フローを示す図である。

【図4】図4は、操作記録プログラムのウィンドウ例を示す図である。

【図5】図5は、キー入力関連処理の処理フローを示す図である。

【図6】図6は、プログラム起動処理の処理フローを示す図である。

【図7】図7は、操作対象のウィンドウ例を示す図である。

【図8】図8は、スクリプトデータの一例を示す図である。

【図9】図9は、クリック処理の処理フローを示す図である。

【図10】図10は、スクリプトデータの一例を示す図である。

【図11】図11は、クラスとスクリプトとの対応関係を表す図である。

【図12】図12は、クラスとスクリプトとの対応関係を表す図である。

【図13】図13は、操作対象のウィンドウに対する操作例を示す図である。

【図14】図14は、スクリプトデータの一例を示す図である。

【図15】図15は、操作対象のウィンドウに対する操作例を示す図である。

【図16】図16は、スクリプトデータの一例を示す図である。

【図17】図17は、操作対象のウィンドウに対する操作例を示す図である。

【図18】図18は、スクリプトデータの一例を示す図である。

【図19】図19は、操作対象のウィンドウに対する操作例を示す図である。

【図20】図20は、スクリプトデータの一例を示す図である。

【図21】図21は、操作対象のウィンドウに対する操作例を示す図である。

【図22】図22は、スクリプトデータの一例を示す図である。

【図23】図23は、操作対象のウィンドウに対する操作例を示す図である。

【図24】図24は、スクリプトデータの一例を示す図である。

【図25】図25は、操作対象のウィンドウに対する操作例を示す図である。

【図26】図26は、スクリプトデータの一例を示す図である。

【図27】図27は、操作対象のウィンドウに対する操作例を示す図である。

【図28】図28は、スクリプトデータの一例を示す図である。

【図29】図29は、操作対象のウィンドウに対する操作例を示す図である。

【図30】図30は、スクリプトデータの一例を示す図である。

【図31】図31は、操作対象のウィンドウに対する操作例を示す図である。

【図32】図32は、スクリプトデータの一例を示す図である。

【図33】図33は、操作対象のウィンドウに対する操作例を示す図である。

【図34】図34は、スクリプトデータの一例を示す図である。

【図35】図35は、テキスト入力ウィンドウの一例を示す図である。

【図36】図36は、スクリプトデータの一例を示す図である。

【図37A】図37Aは、操作対象のウィンドウに対する操作例を示す図である。

【図37B】図37Bは、定義名及び定義値の入力ウィンドウを示す図である。

【図38】図38は、定義ファイルの一例を示す図である。

【図39】図39は、定義名及び定義値選択ウィンドウの一例を示す図である。

10

20

30

40

50

【図 4 0】図 4 0 は、スクリプトデータの一例を示す図である。

【図 4 1】図 4 1 は、操作対象のウィンドウに対する操作例を示す図である。

【図 4 2】図 4 2 は、操作対象のウィンドウに対する操作例を示す図である。

【図 4 3】図 4 3 は、スクリプトデータの一例を示す図である。

【図 4 4】図 4 4 は、制御入力処理の処理フローを示す図である。

【図 4 5】図 4 5 は、記録停止後のウィンドウ構成例を示す図である。

【図 4 6】図 4 6 は、定義ファイルの他の例を示す図である。

【図 4 7】図 4 7 は、スクリプトファイル及び定義ファイルを使用する際の機能ブロック図である。

【図 4 8】図 4 8 は、コンピュータの機能ブロック図である。

10

【発明を実施するための形態】

【0014】

図 1 に、本発明の一実施の形態に係るシステムの概要を示す。操作記録を行う対象のコンピュータ 3 には、液晶表示装置や CRT (Cathode Ray Tube) ディスプレイなどの表示装置 1 と、キーボード 5 と、マウス 7 とが接続されている。なお、LAN (Local Area Network) などに接続している場合もあるが、以下の説明にはあまり関係のないため、説明を省略する。

【0015】

コンピュータ 3 は、オペレーティングシステム (OS : Operating System) や業務アプリケーション・プログラムなどを含む各種プログラム (以下、一般プログラム 3 9 と呼ぶ) 及び本実施の形態に係る操作記録プログラム 3 1 がインストールされており、必要に応じて実行される。また、コンピュータ 3 は、ハードディスクドライブなどのデータ格納部 3 3 を有しており、一般プログラム 3 9 及び操作記録プログラム 3 1 のデータを格納すると共に、操作記録プログラム 3 1 が生成するスクリプトファイル 3 3 1 及び定義ファイル 3 3 3 とが格納される。

20

【0016】

本実施の形態に係る操作記録プログラム 3 1 は、コンピュータ 3 に、クリック処理部 3 1 1 と、テキスト入力処理部 3 1 2 と、定義参照処理部 3 1 3 と、キー入力制限部 3 1 4 と、拡張機能処理部 3 1 5 とを実現させるものである。

【0017】

次に、図 2 乃至図 4 7 を用いて操作記録プログラム 3 1 による処理内容について説明する。まず、ユーザは、操作記録プログラム 3 1 の起動をコンピュータ 3 に指示する。操作記録プログラム 3 1 は、起動すると、例えばスクリプトファイル 3 3 1 の保存先ディレクトリ及びファイル名の指定をユーザに促す。操作記録プログラム 3 1 は、ユーザからスクリプトファイル 3 3 1 の保存先ディレクトリ及びファイル名の指定を受け付け、指定の保存先ディレクトリにおいて指定のファイル名のスクリプトファイル 3 3 1 をデータ格納部 3 3 にオープンしておく。そして、表示装置 1 に例えば図 2 に示すようなウィンドウを表示する。図 2 の例では、記録開始ボタンと、スクリプト生成ボタンと、設定ボタンとが含まれる。スクリプト生成ボタンは、操作記録が完了しないとクリックできないようになっている。設定ボタンは各種設定を行うためのボタンであり、本実施の形態にはあまり関係しないので、詳細な説明は省略する。ここで、ユーザは、記録開始ボタンをクリックするものとする。

30

40

【0018】

そうすると、操作記録プログラム 3 1 は、ユーザから記録開始の指示を受け付け (図 3 : ステップ S 1)、例えば図 4 に示すようなウィンドウを表示装置 1 に表示する。図 2 に示した記録開始ボタンをクリックしたので、記録開始ボタンは記録停止ボタンに変更される。スクリプト生成ボタン及び設定ボタンについては変化しない。そして、記録開始の指示に応じて、テキスト入力ボタンと、仮想キー入力ボタンと、定義選択ボタンと、プログラム起動ボタンと、部分画像採取ボタンと、スナップショットボタンと、リポートボタンと、TAG ボタンと、対象ウィンドウ指定ボタンと、コメント入力ボタンと、e c h o 入

50

カボタンと、log入力ボタンと、フリー入力ボタンと、Sleep入力ボタンと、最大化ボタンとが表示される。部分画像採取ボタン以下のボタンについては、本実施の形態の主旨とは関係しないので、説明を省略する。また、仮想キー入力ボタンについては、Alt、Ctrl、Shift、カーソル移動といった特別なキーの入力を行うためのボタンである。この仮想キー入力ボタンについても、本実施の形態の主旨とはあまり関係しないのでこれ以上説明しない。

【0019】

なお、各ボタンの左側に示されている記号は、ホットキーの表示であって、ボタンをクリックするのではなく、キーボード5のキーを押下げることによっても指示できるようになっている。

10

【0020】

また、操作記録プログラム31のキー入力制限部314は、記録開始指示の後に、ホットキー以外のキーボード入力を禁止する(ステップS3)。このようにすることによって、どのようなキー入力が必要なのかについてユーザに意識させることができるようになる。具体的には、通常のテキスト入力なのか、スクリプトを実行させるコンピュータ毎に設定を変更しなければならない定義選択なのか、さらに仮想キー入力ボタンで入力しなければならないようなキー入力なのかを意識させる。

【0021】

そして、操作記録プログラム31は、ユーザからの入力を受け付け始め、何らかの入力を受け付けると、以下の処理を実施する(ステップS5)。

20

【0022】

この入力がホットキー入力又はホットキーに対応するボタンのクリックである場合(ステップS7:Yesルート)、キー入力関連処理を実施する(ステップS9)。キー入力関連処理が終了するとステップS19に移行する。キー入力関連処理については、図5乃至図8を用いて説明する。

【0023】

ステップS5で受け付けた入力が、プログラム起動ボタンのクリック又はホットキー「X」の押下である場合には(図5:ステップS21)、プログラム起動処理を実施する(ステップS23)。このプログラム起動処理が終了すると、端子Cを介して元の処理に戻る。プログラム起動処理については、図6乃至図8を用いて説明する。

30

【0024】

例えばテキスト入力処理部312は、実行ファイル名等の入力ウィンドウを表示装置1に表示させる(ステップS61)。例えば図7に示すようなウィンドウが表示される。このウィンドウに対するキー入力は許可される。図7の例では、実行すべきプログラムのリスト選択表示をさらに行ってプログラムを選択させるためのラジオボタンとスタートメニューを表示させてプログラムを選択させるためのラジオボタンとそれらのラジオボタンのいずれかの選択を確定させるためのボタンとを含むプログラム選択部と、実行ファイル名を入力するための入力欄と、実行ファイル名をディレクトリ表示から指定するための参照ボタンと、実行すべきプログラムに渡すべき引数を入力するための入力欄と、実行すべきプログラムの作業ディレクトリ名を入力欄と当該作業ディレクトリ名をディレクトリ表示から指定するための参照ボタンと、設定入力を登録するためのOKボタンと、設定入力をキャンセルするためのキャンセルボタンとが含まれる。プログラム選択欄の操作の選択の結果は、実行ファイル名入力欄に反映される。ユーザは、このようなウィンドウに従って、入力を行ってOKボタンをクリックする。

40

【0025】

そして、例えばテキスト入力処理部312は、ユーザから実行ファイル名等の入力を受け付け(ステップS63)、実行ファイル名等を含むスクリプトを生成して、スクリプトファイル331に格納する(ステップS65)。図8に示すようなスクリプトが、スクリプトファイル331に格納される。図8に示したスクリプトは、指定された実行ファイル名のプログラムを実行させるためのスクリプトである。そして、例えばテキスト入力処理

50

部 3 1 2 は、OS に指示して、一般プログラム 3 9 のうち、実際に指定されたプログラムを起動させる (ステップ S 6 7)。

【 0 0 2 6 】

なお、図 5 においてプログラム起動ではない場合 (ステップ S 2 1 : N o ルート) については、以下で必要な場面において説明することとする。

【 0 0 2 7 】

図 3 の処理の説明に戻って、ステップ S 9 が終了すると、操作記録プログラム 3 1 は、処理終了であるか判断し (ステップ S 1 9)、処理終了でなければ端子 B を介してステップ S 5 に戻る。一方、処理終了が指示された場合には、処理を終了する。なお、記録終了ボタンがクリックされないで終了が指示された場合やスクリプト生成ボタンがクリックされないで終了が指示された場合には、ユーザに警告の上、処理を終了する。

10

【 0 0 2 8 】

一方、ステップ S 5 における入力が、ホットキー又はホットキーに対応するボタンのクリックではなく (ステップ S 7 : N o ルート)、マウス 7 によるクリックであれば (ステップ S 1 1 : Y e s ルート)、クリック処理部 3 1 1 は、クリック処理を実施する (ステップ S 1 3)。このクリック処理については、図 9 乃至図 3 4 を用いて説明する。クリック処理の後には、ステップ S 1 9 に移行する。

【 0 0 2 9 】

図 9 を用いてクリック処理を説明する。なお、ユーザは、最初に、操作の対象となるプログラムのウィンドウを選択するため、そのウィンドウをクリックしたものとする。

20

【 0 0 3 0 】

すなわち、クリック処理部 3 1 1 は、ウィンドウの選択、すなわち起動したプログラムについてのウィンドウにおける特定のクラス (例えば Static) についてクリックを行ったかを判断し (図 9 : ステップ S 7 1)、ウィンドウの選択を行った場合には、選択ウィンドウ名を含むスクリプトを生成し、スクリプトファイル 3 3 1 に格納する (ステップ S 7 3)。例えば、この段階で図 1 0 に示すようなスクリプトデータがスクリプトファイル 3 3 1 に格納される。図 1 0 の点線で囲われた部分が、ステップ S 7 3 で追加されたものである。ここで、ウィンドウ名は「S w i t c h B t n T e s t」である。ステップ S 7 3 の後に元の処理に戻る。

【 0 0 3 1 】

30

なお、以下で述べる図 1 1 及び図 1 2 に示すクラス及び上記特定のクラス以外のクラス「ComboLbox」のクリックである場合には、スクリプト「SelectItem ("テキスト")」が生成され、格納される。ここで「テキスト」は、コンボボックスのリストボックスで選択されたテキストである。

【 0 0 3 2 】

一方、ウィンドウの選択ではない場合、クリック処理部 3 1 1 は、ボタン等のクリックであるか判断する (ステップ S 7 5)。より具体的には、クリック対象のウィンドウにおける、図 1 1 に示すようなクラスについてのクリックであるか判断する。すなわち、ここで列挙されているリストボックス及びボタンのクラスに該当するか否かを判断する。図 1 1 に示すテーブルは、クラス名と、対応して生成すべきスクリプトの例を示している。図 1 1 に列挙されているクラスの場合には、クリック処理部 3 1 1 は、図 1 1 のテーブルにおいて対応付けて示されているように、選択したボタン等についての情報 (選択したテキスト、クリックしたボタン等の名前、クリックした結果のオン又はオフなど) を取得して、当該情報を含むスクリプトを生成し、スクリプトファイル 3 3 1 に格納する (ステップ S 7 7)。そして元の処理に戻る。

40

【 0 0 3 3 】

一方、図 1 1 に示すようなクラスではなく図 1 2 に示すようなクラスである場合には、クリック処理部 3 1 1 は、図 1 2 のテーブルにおいて対応付けて示されているように、クリックした座標値及び必要な情報を取得して、当該クリック座標値及び取得情報を含むスクリプトを生成して、スクリプトファイル 3 3 1 に格納する (ステップ S 7 9)。そして

50

元の処理に戻る。

【 0 0 3 4 】

例えば図 1 3 に示すようなウィンドウが操作対象のウィンドウであるとする。図 1 3 のウィンドウのタイトルは、上でも述べた「 S w i t c h B t n T e s t 」である。ここでは、ユーザはマウス 7 を操作して、左上のチェックボックス「 C h e c k 1 」をクリックして、未チェック状態からチェック状態に遷移させたものとする。このような場合には、ステップ S 7 7 に移行して、図 1 4 の点線内に示すようなスクリプトを生成する。図 1 1 の 2 行目のクラスということで、ウィンドウタイトル及びクリックしたチェックボックスの情報を含むスクリプトが生成される。ここでスタイルは「 C h e c k 1 」であり、キャプションは「 1 」であり、状態は「 t r u e 」である。状態「 t r u e 」は、押されたボタンの処理を待ち、状態「 f a l s e 」は、ボタンを閉じたままにするという意味である。

10

【 0 0 3 5 】

次に、例えば図 1 5 に示すようなマウス操作を行ったものとする。すなわち、チェックボックス「 C h e c k 2 」をクリックして、チェック状態を未チェック状態に遷移させたものとする。このような場合には、ステップ S 7 7 に移行して、図 1 6 の点線内に示すようなスクリプトを生成する。図 1 1 の 2 行目のクラスということで、ウィンドウタイトル及びクリックしたチェックボックスの情報を含むスクリプトが生成される。ここでスタイルは「 C h e c k 2 」であり、キャプションは「 0 」であり、状態は「 t r u e 」である。

20

【 0 0 3 6 】

次に、例えば図 1 7 に示すようなマウス操作を行ったものとする。すなわち、チェックボックス「 ¥ ” 」をクリックして、未チェック状態をチェック状態に遷移させたものとする。このような場合には、ステップ S 7 7 に移行して、図 1 8 の点線内に示すようなスクリプトを生成する。図 1 1 の 2 行目のクラスということで、ウィンドウタイトル及びクリックしたチェックボックスの情報を含むスクリプトが生成される。ここでスタイルは ¥ ” を表すための「 ¥ ¥ ¥ ” 」であり、キャプションは「 1 」であり、状態は「 t r u e 」である。

【 0 0 3 7 】

さらに、例えば図 1 9 に示すようなマウス操作を行ったものとする。すなわち、左下の 3 つのラジオボタンの中央の選択肢「 R a d i o 2 」をクリックしたものとする。選択肢「 R a d i o 1 」についての選択は解除される。このような場合には、ステップ S 7 7 に移行して、図 2 0 の点線内に示すようなスクリプトを生成する。図 1 1 の 2 行目のクラスということで、ウィンドウタイトル及びクリックしたラジオボタンの情報を含むスクリプトが生成される。ここでスタイルは「 R a d i o 2 」であり、キャプションは「 1 」であり、状態は「 f a l s e 」である。

30

【 0 0 3 8 】

さらに、例えば図 2 1 に示すようなマウス操作を行ったものとする。すなわち、左下の 3 つのラジオボタンの下の選択肢「 R a d i o 3 」をクリックしたものとする。選択肢「 R a d i o 2 」についての選択は解除される。このような場合には、ステップ S 7 7 に移行して、図 2 2 の点線内に示すようなスクリプトを生成する。図 1 1 の 2 行目のクラスということで、ウィンドウタイトル及びクリックしたラジオボタンの情報を含むスクリプトが生成される。ここでスタイルは「 R a d i o 3 」であり、キャプションは「 1 」であり、状態は「 f a l s e 」である。

40

【 0 0 3 9 】

さらに、例えば図 2 3 に示すようなマウス操作を行ったものとする。すなわち、中央上段のボタンをクリックして、選択状態を非選択状態に遷移させたものとする。このような場合には、ステップ S 7 7 に移行して、図 2 4 の点線内に示すようなスクリプトを生成する。図 1 1 の 2 行目のクラスということで、ウィンドウタイトル及びクリックしたボタンの情報を含むスクリプトが生成される。ここでスタイルは「 P u s h B t n 」であり、

50

キャプションは「0」であり、状態は「true」である。

【0040】

さらに、例えば図25に示すようなマウス操作を行ったものとする。すなわち、中央上段のチェックボックスをクリックして、選択状態を非選択状態に遷移させたものとする。このような場合には、ステップS77に移行して、図26の点線内に示すようなスクリプトを生成する。図11の2行目のクラスということで、ウィンドウタイトル及びクリックしたチェックボックスの情報を含むスクリプトが生成される。ここでスタイルは「3-state」であり、キャプションは「0」であり、状態は「true」である。

【0041】

さらに、例えば図27に示すようなマウス操作を行ったものとする。すなわち、中央下段の排他的ボタンのボタン「Radio4」及び「Radio5」のうちボタン「Radio5」をクリックして、ボタン「Radio5」を選択状態に遷移させたものとする。このような場合には、ステップS77に移行して、図28の点線内に示すようなスクリプトを生成する。図11の2行目のクラスということで、ウィンドウタイトル及びクリックしたチェックボックスの情報を含むスクリプトが生成される。ここでスタイルは「Radio5」であり、キャプションは「1」であり、状態は「false」である。

【0042】

さらに、例えば図29に示すようなマウス操作を行ったものとする。すなわち、右上欄のリストビューのボックス「ListViewItem1」（チェックボックスではない）をクリックして、非選択状態を選択状態に遷移させたものとする。このような場合には、ステップS79に移行して、図30の点線内に示すようなスクリプトを生成する。図12の1行目のクラスということで、ウィンドウタイトル、クリック座標値、クリック対象の情報を含むスクリプトが生成される。スクリプト2行目中「295, 65」は、クリック座標値である。さらに、ctMouseEventRの最初の引数「0」は、左クリックを表し、「1」であれば右クリック、「2」であればダブルクリックを表す。また、スクリプト3行目中「ListViewItem1」は、クリック対象のテキストである。

【0043】

さらに、例えば図31に示すようなマウス操作を行ったものとする。すなわち、右上欄のリストビューのボックス「ListViewItem2」（チェックボックスではない）をクリックして、選択状態を非選択状態に遷移させたものとする。このような場合には、ステップS79に移行して、図32の点線内に示すようなスクリプトを生成する。図12の1行目のクラスということで、ウィンドウタイトル、クリック座標値、クリック対象の情報を含むスクリプトが生成される。スクリプト2行目中「295, 81」は、クリック座標値である。また、スクリプト3行目中「ListViewItem2」は、クリック対象のテキストである。

【0044】

さらに、例えば図33に示すようなマウス操作を行ったものとする。すなわち、右中段のテキスト入力欄3310をクリックしたものとする。次にテキストの入力を行うための準備である。このような場合には、ステップS79に移行して、図34の点線内に示すようなスクリプトを生成する。図12の6行目のクラスということで、ウィンドウタイトル及びクリック座標値を含むスクリプトが生成される。スクリプト2行目中「323, 180」は、クリック座標値である。

【0045】

その後、ユーザは、テキスト入力を行うわけであるが、ステップS3で示したように、ホットキー以外のキーボード入力は禁止されているので、操作記録プログラム31のテキスト入力ボタンをクリックするか、ホットキー「1」を押下げる。

【0046】

このような場合を、図3に戻って説明する。ステップS5で入力を受け付け、この入力がホットキー入力又は対応するボタンのクリックであると判断され、キー入力関連処理を実施する(ステップS9)。キー入力関連処理(図5)の処理の説明に移行して、ステッ

10

20

30

40

50

プ S 2 1 でプログラム起動の指示ではないと判断された上で (ステップ S 2 1 : N o ルート)、テキスト入力であると判断された場合 (ステップ S 2 5 : Y e s ルート)、テキスト入力処理部 3 1 2 は、テキスト入力ウィンドウを表示装置 1 に表示する (ステップ S 2 7)。例えば図 3 5 のようなウィンドウを表示する。

【 0 0 4 7 】

図 3 5 に示すように、文字列入力欄を含むウィンドウが表示され、キー入力制限部 3 1 4 はキー入力を許可する。図 3 5 の例では、「テスト」というテキストが入力されている。なお、改行が必要であれば、「改行を付加」のチェックボックスにチェックを付す。そして、送信ボタンをクリックする。なお、キャンセルする場合には、キャンセルボタンをクリックする。

10

【 0 0 4 8 】

テキスト入力処理部 3 1 2 は、入力テキストを受け付け (ステップ S 2 9)、入力テキストを含むスクリプトを生成し、スクリプトファイル 3 3 1 に格納する (ステップ S 3 1)。例えば図 3 6 の点線内のスクリプトが生成される。この例では、ウィンドウタイトルと、入力テキスト「テスト」が含まれる。そして、元の処理に戻る。なお、操作記録対象のプログラムにも、入力テキストは送信され、図 3 7 A に示すように、テキスト入力欄 3 3 1 0 に、「テスト」というテキストが入力された状態になる。

【 0 0 4 9 】

ここで、ユーザは、スクリプトファイル 3 3 1 を実行するコンピュータに応じて異なる内容のテキストを入力しなければならない場面になったと判断したものとす。このよう
なテキストは、以下の例とは異なるが、例えば装置毎に異なる IP アドレスや、その他の
パラメータである。このような場合には、図 4 のウィンドウにおける定義選択ボタンをク
リックするか、対応付けられているホットキー「3」を押下げる。

20

【 0 0 5 0 】

このような場合を、図 3 に戻って説明する。ステップ S 5 で入力を受け付け、この入力
がホットキー入力又は対応するボタンのクリックであると判断され、キー入力関連処理を
実施する (ステップ S 9)。キー入力関連処理 (図 5) の処理の説明に移行して、ステッ
プ S 2 1 でプログラム起動の指示ではなく (ステップ S 2 1 : N o ルート)、さらにステ
ップ S 2 5 でテキスト入力の指示ではないと判断された上で (ステップ S 2 5 : N o ルート)、
定義選択の指示であると判断された場合 (ステップ S 3 3 : Y e s ルート)、定義
参照処理部 3 1 3 は、データ格納部 3 3 に定義ファイル 3 3 3 が既に存在しているか確認
する (ステップ S 3 7)。

30

【 0 0 5 1 】

定義ファイル 3 3 3 が存在していない場合には、定義参照処理部 3 1 3 は、定義ファイ
ル 3 3 3 を生成するための入力ウィンドウを表示して、ファイル名等の入力を行わせて、
指定されたファイル名の定義ファイル 3 3 3 をデータ格納部 3 3 において生成する (ステ
ップ S 3 9)。そして、定義参照処理部 3 1 3 は、定義名及び定義値の設定ウィンドウを
表示装置 1 に表示させる (ステップ S 4 1)。例えば図 3 7 B に示すようなウィンドウを
表示させる。図 3 7 B の例では、定義名の入力欄と、定義値の入力欄と、OK ボタンと、
キャンセルボタンとが設けられている。本例では、定義名「T E S T D A T A」が入力され、
定義値「1 0」が入力され、OK ボタンがクリックされる。

40

【 0 0 5 2 】

定義参照処理部 3 1 3 は、定義名及び定義値の入力を受け付け、当該定義名及び定義値
を、定義ファイル 3 3 3 に格納する (ステップ S 4 3)。例えば、図 3 8 に示すようなデ
ータが、定義ファイル 3 3 3 に格納される。ここでは、定義名と定義値とが「=」で結ば
れて登録されている。

【 0 0 5 3 】

ステップ S 3 7 で定義ファイル 3 3 3 が存在すると判断された場合又はステップ S 4 3
の後に、定義参照処理部 3 1 3 は、定義名及び定義値選択ウィンドウを表示装置 1 に表示
させる (ステップ S 4 5)。例えば、図 3 9 に示すようなウィンドウが表示される。図 3

50

9の例では、定義名と定義値の対を複数選択可能に表示する欄と、追加ボタンと、編集ボタンと、削除ボタンと、選択ボタンと、キャンセルボタンとが設けられている。このようなウィンドウにおいて、ここでは、ステップS 4 1及びS 4 3で登録した定義名及び定義値を選択したものとす。すなわち、行3 9 1 0を選択した上で、選択ボタンをクリックする。

【0054】

このように、いずれかの行が選択され且つ選択ボタンをクリックされた場合には（ステップS 4 7：Y e s ルート）、定義参照処理部3 1 3は、ユーザから選択指示を受け付け、選択指示がなされたと認識して（ステップS 4 7：Y e s ルート）、選択された定義名を含み且つ対応する定義値を定義ファイル3 3 3から読み出すためのスクリプトを生成して、スクリプトファイル3 3 1に格納する（ステップS 4 9）。例えば図4 0の点線内のスクリプトが生成される。図4 0の点線においては、1行目は操作対象ウィンドウのタイトルを含み、2行目は選択された定義名「T E S T D A T A」を含む。これによって、スクリプトファイル3 3 1が実行された際には、対応する定義値「1 0」が実際には読み込まれることになる。ステップS 4 9の後に、今回選択された定義名に対応する定義値「1 0」を、操作対象ウィンドウに送って、図4 1に示すように、定義値「1 0」を入力欄3 3 1 0に表示させる。ステップS 4 9の後に、元の処理に戻る。

10

【0055】

なお、図3 9に示したウィンドウにおいて、追加ボタン又は特定の行を選択の上変更ボタンをクリックした場合には（ステップS 5 1）、ステップS 4 1に戻る。ここで、定義名及び定義値を追加するか、又は定義名と定義値とのうち少なくともいずれかを修正して登録する。一方、追加ボタン又は変更ボタンのクリックではなく、特定の行を選択した上での削除ボタンのクリックの場合には、定義参照処理部3 1 3は、定義ファイル3 3 3において、選択した行の定義名及び定義値の対を削除する（ステップS 5 3）。そして、端子Dを介してステップS 4 5に移行する。

20

【0056】

さらに、ステップS 3 3で定義選択のホットキーが押下げられていない場合には、拡張機能処理部3 1 5が、押下げられたホットキー又はクリックされたボタンに対応する処理を実施する（ステップS 3 5）。そして、端子Cを介して元の処理に戻る。

【0057】

以上のような処理を実施することによって、異なるコンピュータで異なる値を入力する必要がある場合にも、定義ファイル3 3 3だけを変更すればよいので、コンピュータ毎にスクリプトファイル3 3 1を生成するよりも誤りを低減して効率的に対処することができるようになる。

30

【0058】

次に、ユーザは、図4 2に示すように、操作対象のウィンドウにおけるOKボタンをクリックするものとする。このような場合には、ステップS 7 7に移行して、図4 3の点線内に示すようなスクリプトを生成する。図1 1の2行目のクラスということで、ウィンドウタイトル及びクリックしたボタンの情報を含むスクリプトが生成される。ここでスタイルは「OK」であり、キャプションは「1」であり、状態は「f a l s e」である。

40

【0059】

ここまでで操作記録が完了したものとす。そうすると、ユーザは、操作記録プログラム3 1のウィンドウにおける記録終了ボタンをクリックする。

【0060】

このための処理を図3に戻って説明する。ステップS 5で入力を受け付け、この入力ホットキー入力又は対応するボタンのクリックではなく（ステップS 7：N o ルート）、クリックでもない判断された上で（ステップS 1 1：N o ルート）、制御入力、すなわち操作記録プログラム3 1のウィンドウにおける記録停止などのボタンがクリックされた場合（ステップS 1 5：Y e s ルート）、制御入力処理を実施する（ステップS 1 7）。なお、制御入力でもない場合には端子Aを介してステップS 1 9に移行する。制御入力処

50

理については、図 4 4 及び図 4 5 を用いて説明する。記録停止ボタンがクリックされたと判断された場合には（ステップ S 8 1：Y e s ルート）、キー入力制限部 3 1 4 は、キーボード入力の禁止を解除する（ステップ S 8 3）。ここで、操作記録プログラム 3 1 のウィンドウは、例えば図 4 5 に示すような状態になる。ここで、記録開始ボタンと、スクリプト生成ボタンと、設定ボタンとがクリックできるようになっている。そして、この状態において、操作記録プログラム 3 1 は、さらにユーザからの入力を受け付ける（ステップ S 8 4）。

【 0 0 6 1 】

ステップ S 8 4 において、スクリプト出力ボタンがクリックされた場合には（ステップ S 8 5：Y e s ルート）、操作記録プログラム 3 1 は、データ格納部 3 3 内のスクリプトファイル 3 3 1 をクローズし（ステップ S 8 7）、元の処理に戻る。また、スクリプト出力ボタンがクリックされたわけではなく（ステップ S 8 5：N o ルート）、記録開始ボタンがクリックされた場合には（ステップ S 8 9：Y e s ルート）、端子 E を介して図 3 のステップ S 1 に戻る。記録再開ボタンがクリックされたわけではなく（ステップ S 8 9：N o ルート）、設定ボタンがクリックされると、操作記録プログラム 3 1 は、設定などの処理を実施する（ステップ S 9 1）。そして元の処理に戻る。なお、設定などの処理については、本実施の形態の主要部ではないので説明を省略する。

【 0 0 6 2 】

以上のような処理を実施することによって、ユーザは、簡単にスクリプトファイルを生成できる。さらに装置固有のテキスト入力については定義ファイルから読み出すようなスクリプトを生成させることによって定義ファイルに変更が必要な部分を集約させることで、誤りを低減させつつ簡易に個別装置への対応を可能にしている。

【 0 0 6 3 】

すなわち、ある装置には図 3 8 のような定義ファイル 3 3 3 を用いるが、他の装置には、図 4 6 に示すような定義ファイルに変更した上で、使用する。すなわち、T E S T D A T A = 1 0 であったものを、T E S T D A T A = 3 0 に変更している。このように、装置毎に変更を加えて、定義ファイル 3 3 3 を用意する。

【 0 0 6 4 】

なお、I P アドレスなどの装置固有のデータを保持する他のファイルから該当データ項目の値を読み出して、定義ファイル 3 3 3 において対応する定義名の定義値を変更するような機能を、操作記録プログラム 3 1 に設けるようにしても良い。このような機能を、操作記録プログラム 3 1 の他にツールとして用意するようにしても良い。

【 0 0 6 5 】

また、このように生成されたスクリプトファイル 3 3 1 及び定義ファイル 3 3 3 については、図 4 7 に示すように、スクリプトエンジン 4 1 0 0 によって解釈されて、操作記録対象のプログラム 4 2 0 0 に対して行った操作を再現するように、入力などの操作が行われる。この技術については例えば特開 2 0 0 4 - 7 8 2 6 8 号公報などに記載されている技術であるからこれ以上述べない。

【 0 0 6 6 】

以上本発明の一実施の形態を説明したが、本発明はこれに限定されるものではない。例えば、上で述べた機能ブロック図は一例であって、必ずしも実際のプログラムモジュール構成と一致しない場合もある。また、処理フローについても、処理結果が変わらない限りにおいて、処理順番を入れ替えたり、並列実行するようにしても良い。

【 0 0 6 7 】

さらに、操作記録プログラム 3 1 のウィンドウ構成については一例であって、他の構成であっても良い。また、ホットキーの割当を全てのボタンに対して行って、ウィンドウの表示自体を行わないようにしても良い。

【 0 0 6 8 】

また、操作記録を開始した後に、キーボード入力を制限していたが、操作記録プログラム 3 1 の起動に応じてキーボード入力を制限するようにしても良い。

10

20

30

40

50

【 0 0 6 9 】

なお、上で述べたコンピュータ3は、図48に示すように、メモリ2501とCPU2503とハードディスク・ドライブ(HDD)2505と表示装置2509に接続される表示制御部2507とリムーバブル・ディスク2511用のドライブ装置2513と入力装置2515とネットワークに接続するための通信制御部2517とがバス2519で接続されている。オペレーティング・システム(OS: Operating System)及び本実施例における処理を実施するためのアプリケーション・プログラムは、HDD2505に格納されており、CPU2503により実行される際にはHDD2505からメモリ2501に読み出される。必要に応じてCPU2503は、表示制御部2507、通信制御部2517、ドライブ装置2513を制御して、必要な動作を行わせる。また、処理途中のデータについては、メモリ2501に格納され、必要があればHDD2505に格納される。本技術の実施例では、上で述べた処理を実施するためのアプリケーション・プログラムはコンピュータ読み取り可能なリムーバブル・ディスク2511に格納されて頒布され、ドライブ装置2513からHDD2505にインストールされる。インターネットなどのネットワーク及び通信制御部2517を経由して、HDD2505にインストールされる場合もある。このようなコンピュータ装置は、上で述べたCPU2503、メモリ2501などのハードウェアとOS及び必要なアプリケーション・プログラムとが有機的に協働することにより、上で述べたような各種機能を実現する。

10

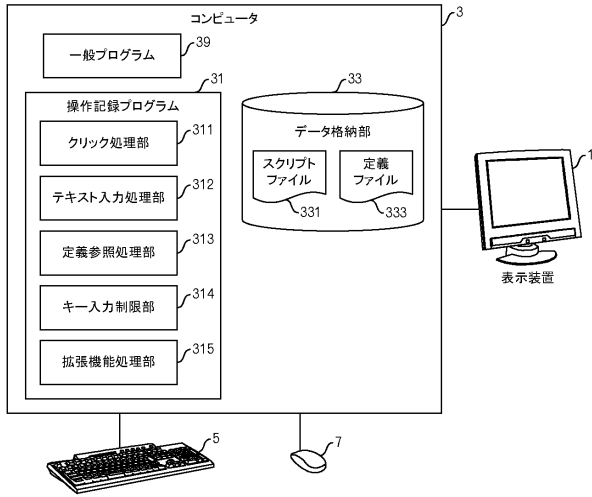
【 符号の説明 】

【 0 0 7 0 】

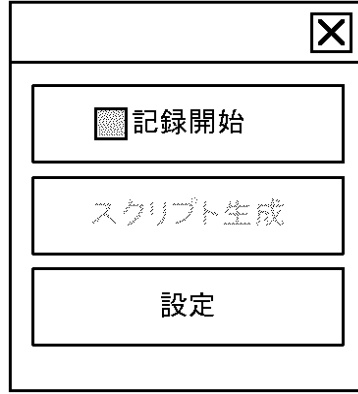
- | | | | |
|-------|-----------|-------|-----------|
| 1 | 表示装置 | 3 | コンピュータ |
| 5 | キーボード | 7 | マウス |
| 3 1 | 操作記録プログラム | 3 3 | データ格納部 |
| 3 9 | 一般プログラム | | |
| 3 1 1 | クリック処理部 | 3 1 2 | テキスト入力処理部 |
| 3 1 3 | 定義参照処理部 | 3 1 4 | キー入力制限部 |
| 3 1 5 | 拡張機能処理部 | | |

20

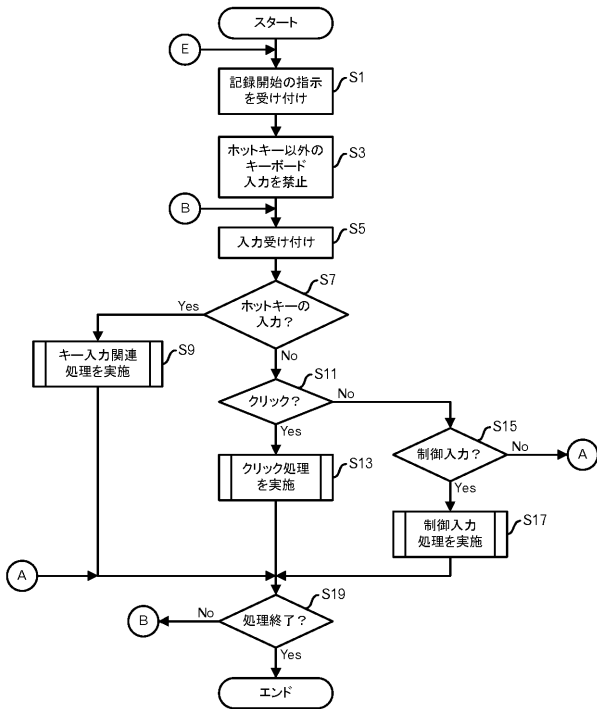
【図1】



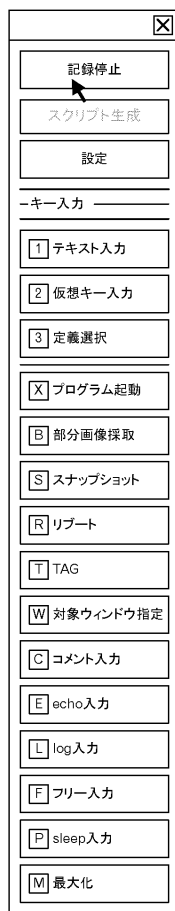
【図2】



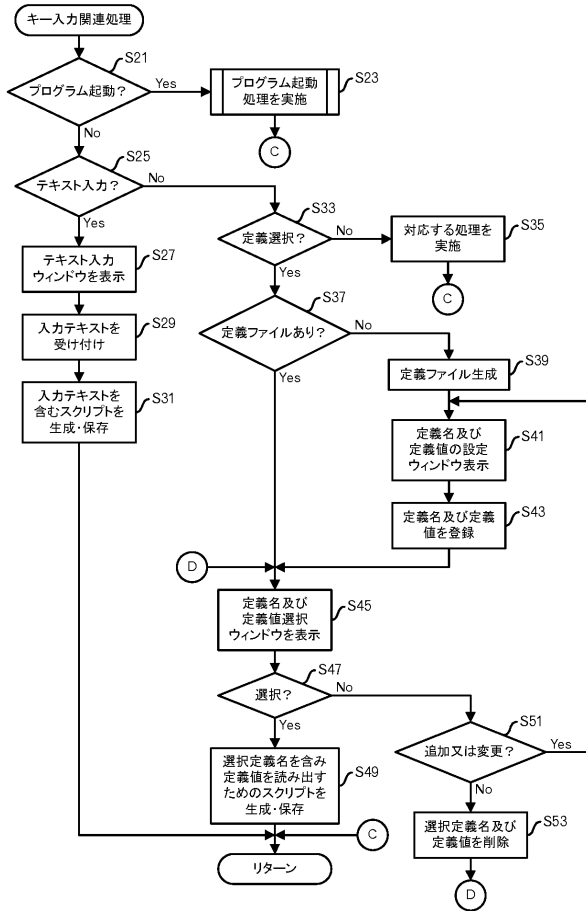
【図3】



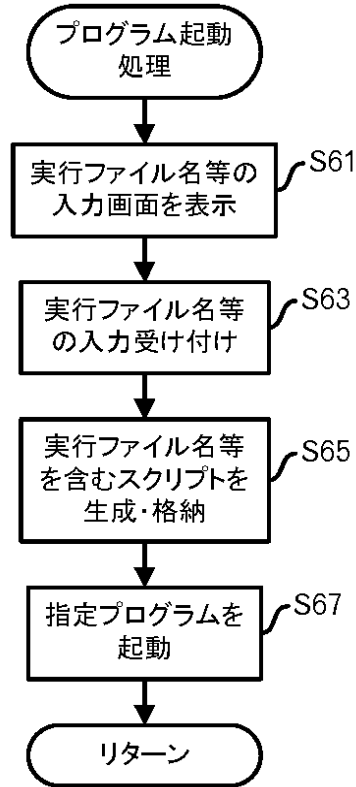
【図4】



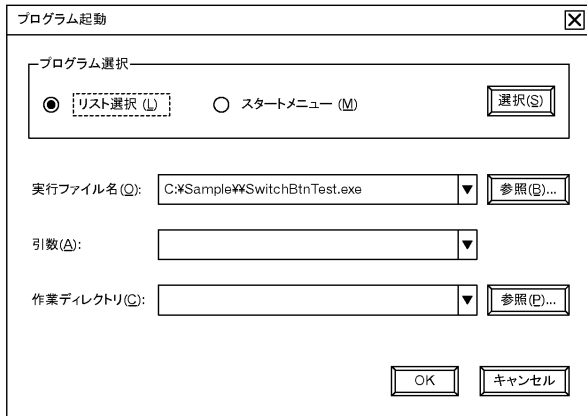
【図5】



【図6】



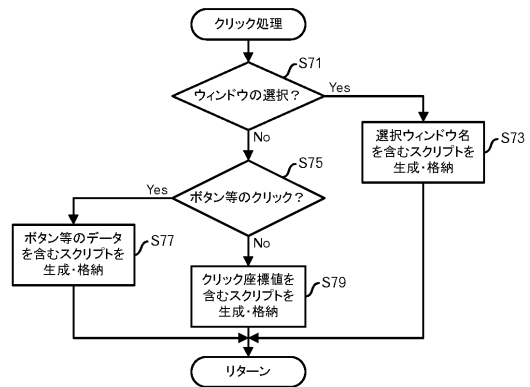
【図7】



【図8】

```
execProc("C:*Sample**SwitchBtnTest.exe", "", 0);
```

【図9】



【図10】

```
execProc("C:*Sample**SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
```

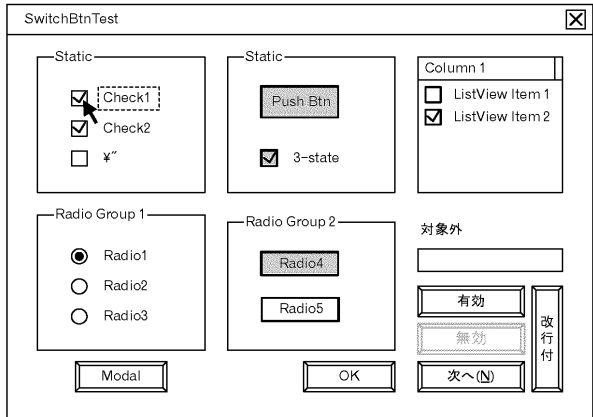
【図11】

クラス名	スクリプト
ListBox	CWF_retrySetActiveWindow("ウィンドウタイトル", "");
	SelectItem("テキスト");
Button	CWF_retrySetActiveWindow("ウィンドウタイトル", "");
	Button("スタイル", "キャプション", "状態");

【 図 1 2 】

クラス名	文
SysListView32	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y); SelectItem("テキスト");
SysTreeView32	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y); SelectItem("テキスト");
SysTabControl32	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y); SelectItem("テキスト");
ComboBoxEx32	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y);
SysIPAddress32	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(2,"ウインドウタイトル",マウス位置X,マウス位置Y);
Edit	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(2,"ウインドウタイトル",マウス位置X,マウス位置Y);
ComboBox	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y);
ScrollBar	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y);
その他	CWF_retrySetActiveWindow("ウインドウタイトル",""); ctlMouseEventR(0,"ウインドウタイトル",マウス位置X,マウス位置Y);

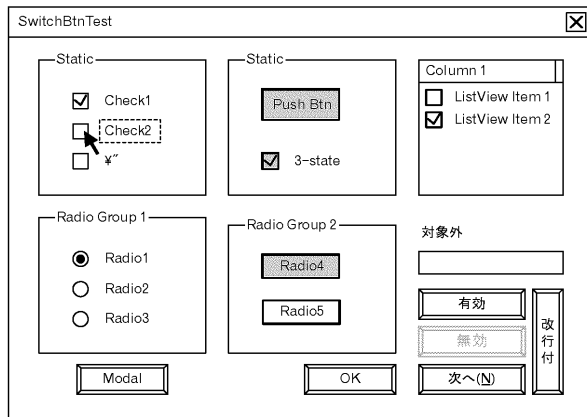
【 図 1 3 】



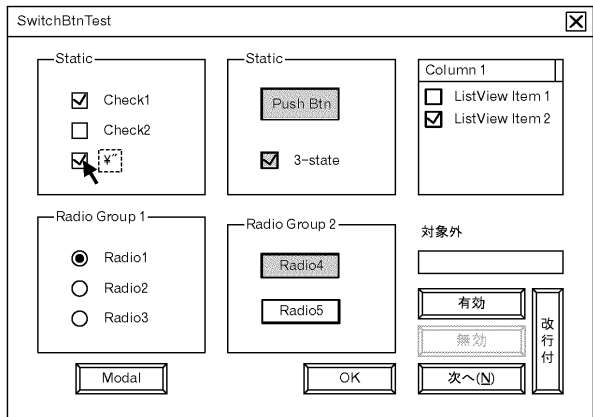
【 図 1 4 】

```
execProc("C:¥¥Sample¥¥SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);
```

【 図 1 5 】



【 図 1 7 】



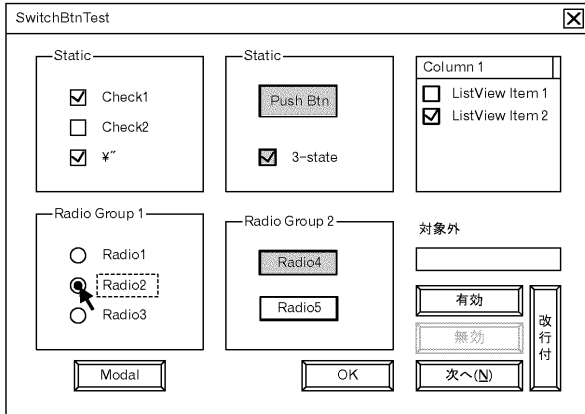
【 図 1 6 】

```
execProc("C:¥¥Sample¥¥SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);
```

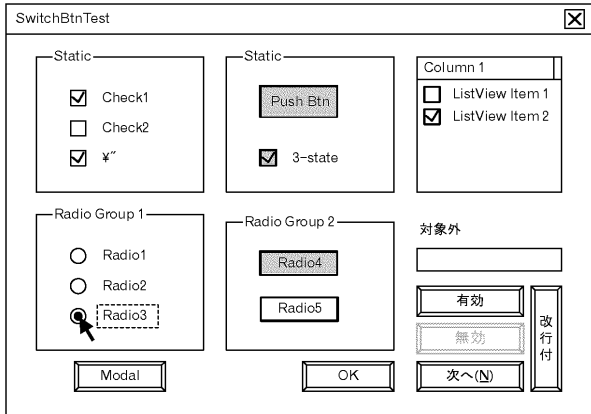
【 図 1 8 】

```
execProc("C:¥¥Sample¥¥SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥ ¥", 1,true);
```

【 図 19 】



【 図 21 】



【 図 20 】

```
execProc("C:\\Sample\\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥ ¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);
```

【 図 22 】

```
execProc("C:\\Sample\\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

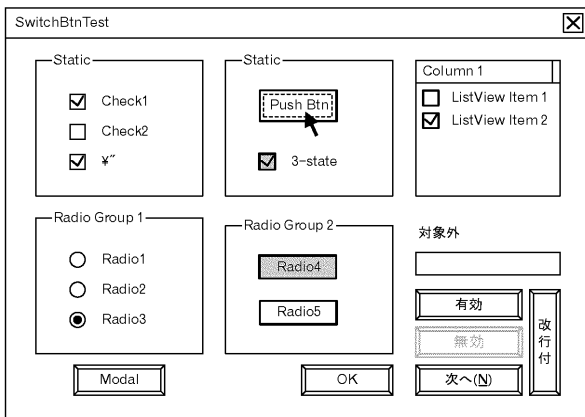
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥ ¥", 1,true);

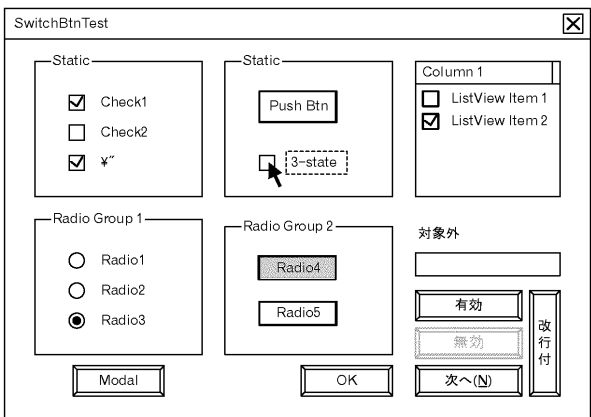
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);
```

【 図 23 】



【 図 25 】



【 図 24 】

```
execProc("C:\\Sample\\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥ ¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);
```

【 図 26 】

```
execProc("C:\\Sample\\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥ ¥", 1,true);

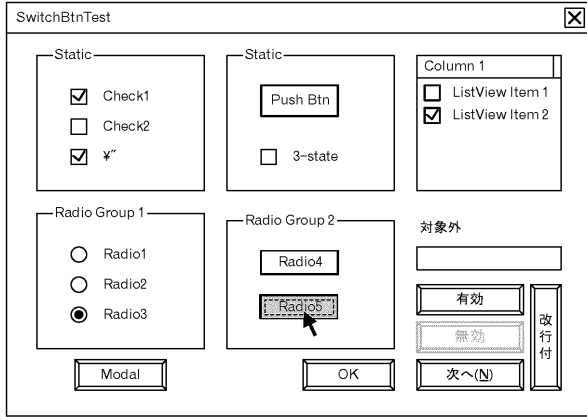
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);
```

【 27 】



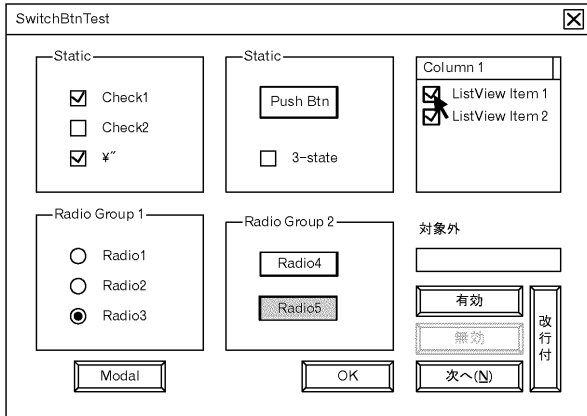
【 28 】

```

execProc("C:\Sample\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥¥", 1,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);

```

【 29 】



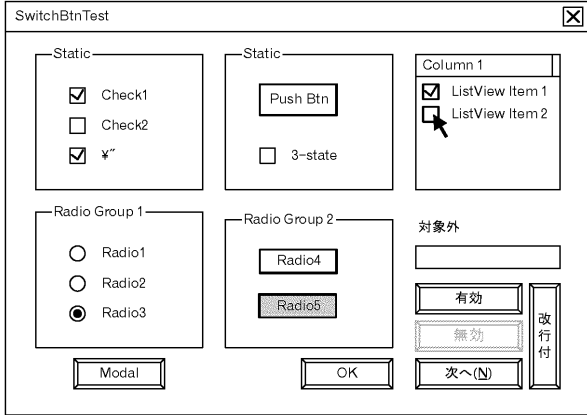
【 30 】

```

execProc("C:\Sample\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥¥", 1,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 65);
selectItem("ListView Item 1");

```

【 図 3 1 】



【 図 3 2 】

```

execProc("C:\Sample\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);

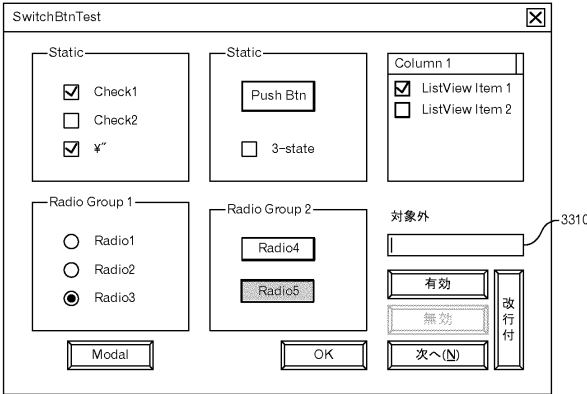
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 65);
selectItem("ListView Item 1");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 81);
selectItem("ListView Item 2");

```

【 図 3 3 】



【 図 3 4 】

```

execProc("C:\Sample\SwitchBtnTest.exe", "", 0);
CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);

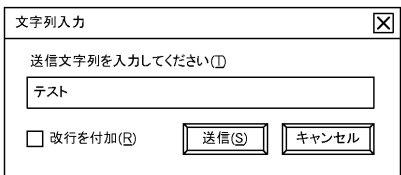
WF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 65);
selectItem("ListView Item 1");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 81);
selectItem("ListView Item 2");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(2, "SwitchBtnTest", 323, 180);

```

【 図 3 5 】



【 図 3 6 】

```

execProc("C:\\Sample\\SwitchBtnTest.exe", "", 0);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥ ¥ ¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);

WF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 65);
selectItem("ListView Item 1");

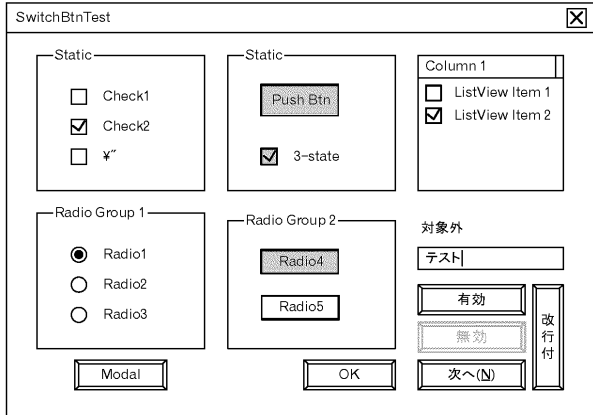
CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 81);
selectItem("ListView Item 2");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(2, "SwitchBtnTest", 323, 180);

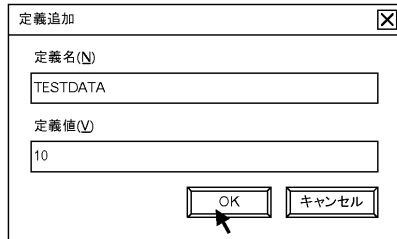
CWF_retrySetActiveWindow("SwitchBtnTest", "");
keyEvent("テスト");

```

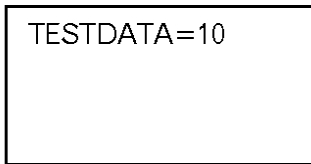
【 図 3 7 A 】



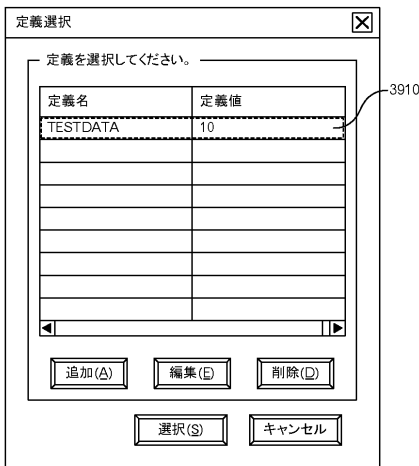
【 図 3 7 B 】



【 図 3 8 】



【 図 3 9 】



【 図 4 0 】

```

execProc("C:\\Sample\\SwitchBtnTest.exe", "", 0);

CWF_retrySetActiveWindow("SwitchBtnTest", "");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥ ¥ ¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);

WF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 65);
selectItem("ListView Item 1");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 81);
selectItem("ListView Item 2");

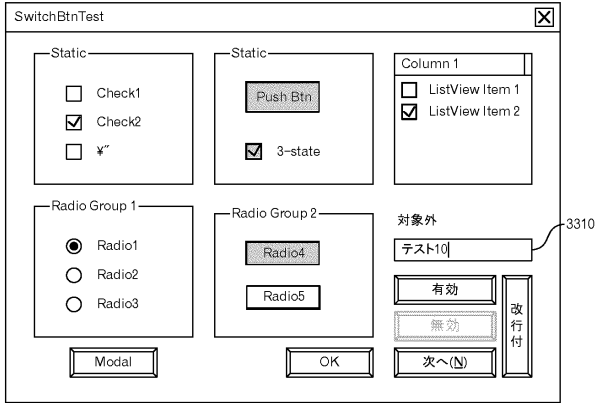
CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(2, "SwitchBtnTest", 323, 180);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
keyEvent("テスト");

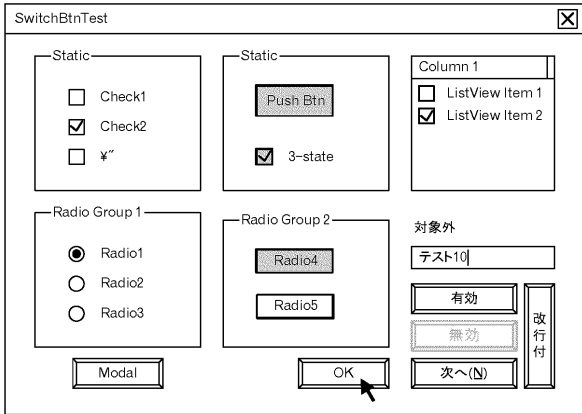
CWF_retrySetActiveWindow("SwitchBtnTest", "");
keyEvent(getDefineValue("TESTDATA"));

```

【 図 4 1 】



【 図 4 2 】



【 図 4 3 】

```

execProc("C:\Sample\SwitchBtnTest.exe", "", 0);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check1", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Check2", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("¥¥¥", 1,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio2", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio3", 1,false);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Push Btn", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("3-state", 0,true);

CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("Radio5", 1,false);

WF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 65);
selectItem("ListView Item 1");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(0, "SwitchBtnTest", 295, 81);
selectItem("ListView Item 2");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
ctlMouseEventR(2, "SwitchBtnTest", 323, 180);

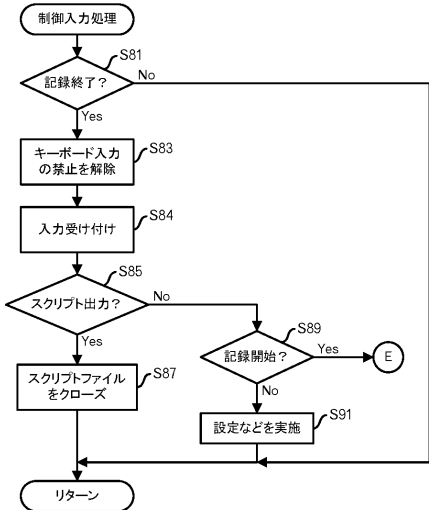
CWF_retrySetActiveWindow("SwitchBtnTest", "");
keyEvent("テスト");

CWF_retrySetActiveWindow("SwitchBtnTest", "");
keyEvent(getDefineValue("TESTDATA"));

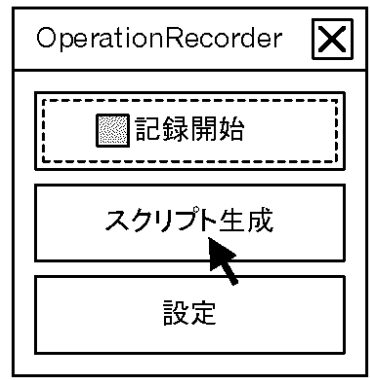
CWF_retrySetActiveWindow("SwitchBtnTest", "");
setSwitchButton("OK", 1,false);

```

【 図 4 4 】



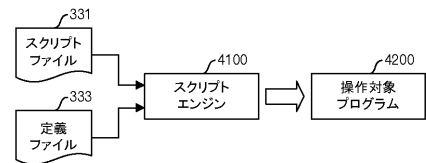
【 図 4 5 】



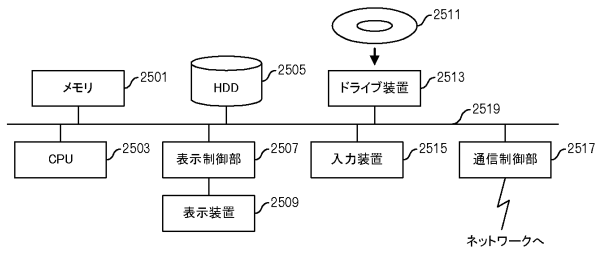
【 図 4 6 】

TESTDATA=30

【 図 4 7 】



【図48】



フロントページの続き

- (56)参考文献 特開2008-042706(JP,A)
特開2006-72806(JP,A)
特開2004-46445(JP,A)
特開2004-5437(JP,A)
特開2009-031916(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 3/02
G06F 9/445