

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-79342

(P2010-79342A)

(43) 公開日 平成22年4月8日(2010.4.8)

(51) Int. Cl.		F I			テーマコード (参考)	
G 0 6 F	9/44	(2006.01)	G 0 6 F	9/06	6 2 0 L	5 B 1 0 9
G 0 6 F	17/21	(2006.01)	G 0 6 F	17/21	5 8 6 A	5 B 1 7 6
G 0 6 F	3/048	(2006.01)	G 0 6 F	3/048	6 5 1 B	5 E 5 0 1
			G 0 6 F	17/21	5 3 8 A	

審査請求 未請求 請求項の数 1 O L (全 16 頁)

(21) 出願番号	特願2008-243600 (P2008-243600)	(71) 出願人	000233055
(22) 出願日	平成20年9月24日 (2008. 9. 24)		日立ソフトウェアエンジニアリング株式会社
			東京都品川区東品川四丁目12番7号
		(74) 代理人	100095267
			弁理士 小島 高城郎
		(74) 代理人	100124176
			弁理士 河合 典子
		(74) 代理人	100108051
			弁理士 小林 生央
		(72) 発明者	松前 健太郎
			東京都品川区東品川四丁目12番7号 日
			立ソフトウェアエンジニアリング株式会社
			内
		Fターム(参考)	5B109 NH11 SA03 TB03
			最終頁に続く

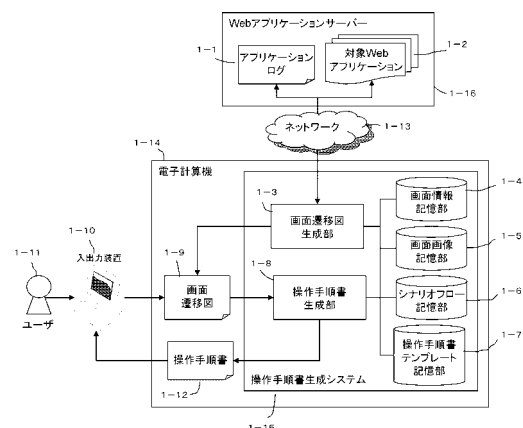
(54) 【発明の名称】 Webアプリケーションの操作手順書生成システム

(57) 【要約】

【課題】完成したWebアプリケーションから操作手順書の雛形を自動生成することで、開発者の文書作成のコストを抑えることができるWebアプリケーションの操作手順書生成システムを提供する。

【解決手段】Webアプリケーションの操作手順書生成システム1-15は、ユーザが、ルートURLを指定すると、ルートURLが示すWebアプリケーションサーバー1-16のWebアプリケーションの画面にアクセスする手段と、アクセスした画面を引数として画面解析処理を呼び出す手段と、画面解析処理によって、解析対象の画面のHTMLを解析し、画面中に存在する全てのリンク、フォームの情報を取得し、画面情報記憶部に登録する手段と、解析対象の画面をキャプチャし、それを画像ファイルに変換し、画面画像記憶部に保存する手段と、取得したすべてのリンク、フォームについて、アクセス、又は、フォーム送信を行い、画面遷移を行う手段等を備えている。

【選択図】図1



【特許請求の範囲】

【請求項 1】

操作手順書の生成の対象となるWebアプリケーションが稼動しているWebアプリケーションサーバーにネットワークを介して通信自在に接続される電子計算機に搭載されるWebアプリケーションの操作手順書生成システムであって、

前記Webアプリケーションの操作手順書生成システムは、画面遷移、画面の構成などの画面情報を記憶する画面情報記憶部と、

画像ファイルを記憶する画面画像記憶部と、

操作手順書を作成するためのシナリオフローの情報を記憶するシナリオフロー記憶部と

10

、
操作手順書テンプレートを記憶する操作手順書テンプレート記憶部と、

ユーザが、WebアプリケーションのルートURLを指定すると、ルートURLが示す前記WebアプリケーションサーバーのWebアプリケーションの画面にアクセスする手段と、

アクセスした画面を引数として画面解析処理を呼び出す手段と、

画面解析処理によって、解析対象の画面のHTMLを解析し、画面中に存在する全てのリンク、フォームの情報を取得し、前記画面情報記憶部に登録する手段と、

解析対象の画面をキャプチャし、それを画像ファイルに変換し、前記画面画像記憶部に保存する手段と、

取得したすべてのリンク、フォームについて、リンクであればリンク先にアクセスし、フォームであればフォーム送信を行い、画面遷移を行う手段と、

20

画面遷移を行った結果、HTTPリダイレクト応答を受け取ったかどうか判定する手段と、

HTTPリダイレクト応答を受け取った場合は、リダイレクト先のパスを前記画面情報記憶部に登録し、さらにリダイレクト先のパスに遷移する手段と、

遷移先の画面についての画面情報が、前記画面情報記憶部に登録されているかどうかをチェックする手段と、

遷移先の画面についての画面情報が、前記画面情報記憶部に登録されていなければ、遷移先の画面を解析対象の画面として、前記画面解析処理を再帰的に呼び出し、画面中に含まれるリンクを辿り、Webアプリケーションを構成する全ての画面情報及び画面画像を取得する手段と、

前記画面情報記憶部に登録されたリンクやフォームの情報について、それらの画面要素を強調するような画像を作成し、前記画面画像記憶部に保存する手段と、

30

画面解析処理によって画面情報記憶部、画面画像記憶部に蓄積された情報を使って画面遷移図を作成する手段と、

前記画面遷移図を参照してユーザ操作によって予め作成されたシナリオフロー情報を前記シナリオフロー記憶部から1レコードずつ取得する手段と、

前記操作手順書テンプレート記憶部から取得した操作手順書テンプレートに、シナリオフロー情報のシナリオ名を埋め込む手段と、

前記操作手順書テンプレートに、シナリオフロー情報のレコードの項番を埋め込む手段と、

前記操作手順書テンプレートに、操作手順メッセージを埋め込む手段と、

40

取得したレコードが、シナリオフロー情報の最後のレコードかどうか判定する手段と、

最後のレコードでない場合は、シナリオフローのレコードの「遷移実行画像」が示す画面ショット画像を、前記操作手順書テンプレートに埋め込む手段と、

最後のレコードである場合は、シナリオフローのレコードの「リンク元画像」が示す画面ショット画像を、前記操作手順書テンプレートに埋め込む手段と、

前記操作手順書テンプレートを操作手順書として出力する手段とを備えることを特徴とするWebアプリケーションの操作手順書生成システム。

【発明の詳細な説明】

【技術分野】

【0001】

50

本発明はWebアプリケーションの使用方法をユーザに説明するための操作手順書を生成するためのWebアプリケーションの操作手順書生成システムに関するものである。

【背景技術】

【0002】

企業がビジネスを遂行していく上でWebアプリケーションシステムの重要性がますます高まってきている。近年ではライバル企業に差をつけるために、より短期間・低コストで品質の高いシステムの開発が求められるようになってきた。短期間・低コストでシステムを開発するには、その開発プロセスが非常に重要になってくる。

【0003】

これまで、システムの開発現場では「ウォーターフォール型開発手法」がよく用いられていた。これは、最初に要件定義を行い、外部仕様書、詳細仕様書を定義し、すべての仕様書が完成した時点で実際のプログラムを開発、テストする、という開発手法である。プログラムを開発する時点で既にこれから開発するべきプログラムの詳細な仕様が決定しているため、仕様漏れやミスを防ぐことができ、効率よく開発ができる。

【0004】

しかしウォーターフォール型開発手法は、仕様書がすべて完成しなければプログラムの実装を行うことができないという特徴がある。仕様書の作成には多大な時間と労力がかかり、これでは短期間・低コストでシステムを開発することは困難になってくる。そこで近年注目されているのが「アジャイルソフトウェア開発」である。これは、より高速にソフトウェアを開発するための開発手法の総称である。たとえばアジャイルソフトウェア開発の具体的な手法のひとつである「エクストリーム・プログラミング」では、2~4週間という短い期間で設計、実装、テスト、リリースを行い、それを何度も繰り返すことによって開発を進める。これにより早い段階で実際に動作するソフトウェアを利用者に提供することができ、また利用者からの声を開発に即座にフィードバックすることで、より品質の高いソフトウェアを開発することが可能となる。またアジャイル開発では、仕様書をたくさん書くよりも、実際に動くソフトウェアを作り、フィードバックを受けて繰り返し改善して行く、という基本的な考え方がある。そのため、ウォーターフォール型開発に比べて作成する仕様書などの文書の数大幅に減る。

【0005】

しかし、作成する文書が少ないために、システムの保守が困難になってしまうという問題も指摘されている。たとえば設計文書が少ないために、システムに障害が発生した際に保守担当者が問題の原因を特定することが困難になってしまう場合がある。また、システムそのものの使い方を記述した利用手順書（マニュアル）が無いために、システムの利用者や管理者がシステムを使用できない、という状態にもなりえる。

【0006】

このような問題を解決するには、いかに開発者の文書作成の負荷を減らすかという点が重要になってくる。その解決方法のひとつとして、既に完成したWebアプリケーションから操作手順書を自動生成して、文書作成にかかる手間を省く方法が提案されている。特許文献1では、あらかじめ手作業でWebアプリケーションの画面に対応するURLを登録しておくことで、自動的にWebアプリケーションの画面ショットを画像ファイルとして保存する機能を有するシステムについて述べたものである。これにより、操作手順書などを作成するときに必要となるWebアプリケーションの画面ショット画像を簡単に用意することが可能となる。

【特許文献1】特開2008-9674号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

しかし特許文献1で言及している方法は画面ショット画像を作成することだけに特化しており、操作手順書自体は、依然として手作業で作成する必要がある。また、あらかじめ

10

20

30

40

50

画面ショットを撮影したい画面のURLを指定しておく必要があるため、Webアプリケーションの画面数が膨大になった場合にはURLの指定作業に多大なコストがかかってしまう。

【 0 0 0 8 】

以上の現状に鑑み、本発明は、完成したWebアプリケーションから操作手順書の雛形を自動生成することで、開発者の文書作成のコストを抑えることができるWebアプリケーションの操作手順書生成システムを提供することを目的とする。

【課題を解決するための手段】

【 0 0 0 9 】

上記の課題を解決すべく、本発明は以下の構成を提供する。

請求項 1 に係る発明は、操作手順書の生成の対象となるWebアプリケーションが稼動しているWebアプリケーションサーバーにネットワークを介して通信自在に接続される電子計算機に搭載されるWebアプリケーションの操作手順書生成システムであって、

前記Webアプリケーションの操作手順書生成システムは、画面遷移、画面の構成などの画面情報を記憶する画面情報記憶部と、

画像ファイルを記憶する画面画像記憶部と、

操作手順書を作成するためのシナリオフローの情報を記憶するシナリオフロー記憶部と、

操作手順書テンプレートを記憶する操作手順書テンプレート記憶部と、

ユーザが、WebアプリケーションのルートURLを指定すると、ルートURLが示す前記WebアプリケーションサーバーのWebアプリケーションの画面にアクセスする手段と、

アクセスした画面を引数として画面解析処理を呼び出す手段と、

画面解析処理によって、解析対象の画面のHTMLを解析し、画面中に存在する全てのリンク、フォームの情報を取得し、前記画面情報記憶部に登録する手段と、

解析対象の画面をキャプチャし、それを画像ファイルに変換し、前記画面画像記憶部に保存する手段と、

取得したすべてのリンク、フォームについて、リンクであればリンク先にアクセスし、フォームであればフォーム送信を行い、画面遷移を行う手段と、

画面遷移を行った結果、HTTPリダイレクト応答を受け取ったかどうか判定する手段と、

HTTPリダイレクト応答を受け取った場合は、リダイレクト先のパスを前記画面情報記憶部に登録し、さらにリダイレクト先のパスに遷移する手段と、

遷移先の画面についての画面情報が、前記画面情報記憶部に登録されているかどうかをチェックする手段と、

遷移先の画面についての画面情報が、前記画面情報記憶部に登録されていなければ、遷移先の画面を解析対象の画面として、前記画面解析処理を再帰的に呼び出し、画面中に含まれるリンクを辿り、Webアプリケーションを構成する全ての画面情報及び画面画像を取得する手段と、

前記画面情報記憶部に登録されたリンクやフォームの情報について、それらの画面要素を強調するような画像を作成し、前記画面画像記憶部に保存する手段と、

画面解析処理によって画面情報記憶部、画面画像記憶部に蓄積された情報を使って画面遷移図を作成する手段と、

前記画面遷移図を参照してユーザ操作によって予め作成されたシナリオフロー情報を前記シナリオフロー記憶部から 1 レコードずつ取得する手段と、

前記操作手順書テンプレート記憶部から取得した操作手順書テンプレートに、シナリオフロー情報のシナリオ名を埋め込む手段と、

前記操作手順書テンプレートに、シナリオフロー情報のレコードの項番を埋め込む手段と、

前記操作手順書テンプレートに、操作手順メッセージを埋め込む手段と、

取得したレコードが、シナリオフロー情報の最後のレコードかどうか判定する手段と、

最後のレコードでない場合は、シナリオフローのレコードの「遷移実行画像」が示す画面ショット画像を、前記操作手順書テンプレートに埋め込む手段と、

10

20

30

40

50

最後のレコードである場合は、シナリオフローのレコードの「リンク元画像」が示す画面ショット画像を、前記操作手順書テンプレートに埋め込む手段と、

前記操作手順書テンプレートを操作手順書として出力する手段とを備えることを特徴とするWebアプリケーションの操作手順書生成システムを提供するものである。

【発明の効果】

【0010】

本発明のWebアプリケーションの操作手順書生成システムによれば、次のような効果がある。

(1)

本システムによって自動生成される画面遷移図上で、操作手順書を作成したい範囲を選択するだけで、操作手順書の雛形が自動生成されるため、操作手順書の作成コストを大幅に減らすことができる。

(2)

生成される操作手順書に挿入される画面ショット画像には、操作手順を示すためのマークが追加描画されるので、開発者が画像を編集する手間を省くことができ、かつ利用者にとってより分かりやすい操作手順書を提供できる。

【発明を実施するための最良の形態】

【0011】

以下、実施例を示した図面を参照しつつ本発明の実施の形態を説明する。

尚、上記各手段は、電子計算機のCPUが必要なコンピュータプログラムを読み込んで実行することにより実現される手段であり、そのフローチャートが図3、図4、図14である。

図1は本発明で提案する操作手順書生成システムと、操作手順書生成システムを動作させるための関連システムとから成る全体構成を示したものである。操作手順書生成システム(1-15)は、一般的な電子計算機(1-14)上に搭載され、動作するプログラム群である。電子計算機(1-14)にはディスプレイやキーボードなどの入出力装置(1-10)が接続されており、さらにLAN・インターネット等のネットワーク(1-13)に接続されている。電子計算機(1-14)はネットワーク(1-13)を通じてWebアプリケーションサーバー(1-16)に接続されている。Webアプリケーションサーバー(1-16)上では、操作手順書の生成の対象となるWebアプリケーション(1-2)が稼動している。アプリケーションログ(1-1)は、Webアプリケーション(1-2)の動作状況が記録されたファイルである。例えばWebアプリケーション(1-2)に対するリクエストがあると、リクエストがあった時刻、リクエストされたURL、送信されたリクエストパラメータ、等がアプリケーションログ(1-1)に記録される。本発明が前提としているアジャイルソフトウェア開発では、前に述べたように短い期間の間に頻りにリリース作業を行う。リリースされたWebアプリケーション(1-2)は実際に顧客に使用される。そのためアプリケーションログ(1-1)には十分な量のログが出力されていることを前提とする。

電子計算機(1-14)はネットワーク(1-13)を通じてWebアプリケーションサーバー(1-16)に接続されているため、Webアプリケーション(1-2)へのアクセスや、アプリケーションログ(1-1)の読み込みが可能となる。

また、電子計算機(1-14)は記憶装置を備え、記憶装置はデータベースから成る後述する記憶部1-4～1-7を格納する。

【0012】

本システムを用いて操作手順書を生成するには、まず画面遷移図生成部(1-3)がWebアプリケーション(1-2)にアクセスし、その画面遷移、画面の構成などの情報を画面情報として取得し、それを画面情報記憶部(1-4)に記憶する。画面遷移図生成部(1-3)は、画面上のリンクを次から次へと辿り、Webアプリケーション(1-2)のすべての画面の情報を取得する。画面の情報を取得する際、同時にWebアプリケーション(1-2)の画面をキャプチャし、画像ファイルとして画面画像記憶部(1-5)に保持し

10

20

30

40

50

ておく。Webアプリケーション（１－２）の画面内に、ユーザからの入力を受け付ける画面要素（HTMLのinputタグなど）がある場合は、アプリケーションログ（１－１）を読み込み、そこから過去に入力された情報を得て、画面情報を取得する際に利用する。このようにして収集した画面情報、画面画像を組み合わせ、画面遷移図（１－９）を作成し、入出力装置（１－１０）上に表示する。

【００１３】

次にユーザ（１－１１）は、画面遷移図を見て、操作手順書を作成したいシナリオフローを入力する。入力したシナリオフローの情報は、シナリオフロー記憶部（１－６）に記憶される。すると、操作手順書生成部（１－８）はシナリオフロー記憶部（１－６）からシナリオフローの情報を、操作手順書テンプレート記憶部（１－７）から操作手順書テンプレートを取得し、操作手順書（１－１２）を生成する。生成した手順書は入出力装置（１－１０）に表示される。

【００１４】

ここからは具体的な例として、図２に示す画面遷移をもったWebアプリケーションの操作手順書を生成する手順を説明する。

図２は日記を書いたり、表示したりするWebアプリケーションの画面遷移を示したものである。「トップページ」画面（２－１）がこのWebアプリケーションを利用する際に最初に表示する画面である。このページには「日記を書く」「日記を見る」という２つのリンクが設けられている。「日記を書く」リンクをクリックすると「日記を書く」画面（２－２）が表示される。この画面中の「タイトル」テキストフィールドと「本文」テキストエリアに、日記のタイトルと本文を入力し「確認」ボタンを押す。すると「確認」画面（２－３）が表示され、「日記を書く」画面で入力した日記の内容が再度表示される。ユーザはこの内容が正しいことを確認し「登録」ボタンを押す。すると日記の内容が保存され、日記表示画面（２－５）が表示される。また「トップページ」画面の「日記を見る」リンクをクリックした場合は、「日記一覧」画面（２－４）が表示される。この画面にはいままでに登録した日記が一覧形式で表示される。一覧から任意の日記のタイトルをクリックすると、日記表示画面（２－５）が表示される。以降では、このWebアプリケーションを「日記アプリケーション」と呼ぶ。

【００１５】

なお、確認画面（２－３）から日記表示画面（２－５）への画面遷移はHTTPリダイレクト応答による遷移を含んでいる。HTTPリダイレクト応答とは、HTTPにおけるサーバからの応答の一種であり、URLが変更されたことを通知するためのものである。一般的なWebブラウザでは、HTTPリダイレクト応答を受け取ると、HTTPリダイレクト応答に含まれる「リダイレクト先」のURLに自動的に画面遷移するように動作する。日記アプリケーションにおいては、確認画面中の「登録」ボタンを押すと、「/submit」というパスに対するHTTPリクエストを発行する。すると、リダイレクト先が「/view」であるHTTPリダイレクト応答が返ってくる。Webブラウザはこれを認識し、今度は「/view」に対してHTTPリクエストを発行する。すると今度は日記表示画面のHTMLデータがHTTPレスポンス情報として返ってくる。これにより、日記表示画面がWebブラウザに表示される。

【００１６】

図３は画面遷移図生成部の処理内容を示したフローチャートである。

（３－１）まず本システムの利用者は画面遷移図生成部に対してWebアプリケーションのルートURLを指定する。ルートURLとは、Webアプリケーションの開始画面、もしくは基点となる画面のURLのことである。日記アプリケーションの「トップページ」画面のURLがこれに該当する。

（３－２）次に、ルートURLが示すWebアプリケーションの画面にアクセスする。画面遷移図生成部はHTTPクライアントとしての機能を備えている。HTTPクライアントとは、HTTP通信によってインターネットやLAN上のWebページにアクセスし、HTML文書や、文書が参照している画像ファイルなどをやり取りするプログラムのことである。通常我々が使用しているWebブラウザもHTTPクライアントの一種である。

(3 - 3) 上記処理でアクセスした画面を引数として、画面解析処理を呼び出す。画面解析処理については図 4 でその詳細を説明する。

(3 - 4) 画面解析処理によって画面情報記憶部、画面画像記憶部に情報が蓄積されるので、それらの情報を使って画面遷移図を作成する。作成手順については図 8 でその詳細を説明する。

【 0 0 1 7 】

図 4 は、画面解析処理の処理内容を示したフローチャートである。この処理は引数として、解析対象の画面を取る。たとえば図 3 の処理では、ルート URL の画面を解析対象の画面としてこの処理を実行している。

(4 - 1) 解析対象の画面の HTML を解析し、画面中に存在する全てのリンク、フォームの情報を取得し、画面情報記憶部に登録する。画面の HTML を解析する手法としては、XML パーサーを用いて HTML を XML として読み込み、DOM (Document Object Model) として扱う方法等がある。DOM ツリーからリンク (A タグ) 要素やフォーム (FORM タグ) 要素を取得することで、画面中に存在する全てのリンク、フォームの情報を取得することが可能となる。取得した情報を画面情報記憶部に登録する処理の詳細については図 5 で述べる。

(4 - 2) 解析対象の画面をキャプチャし、それを画像ファイルに変換し、画面画像記憶部に保存する。例えばウィンドウズ (登録商標) を搭載したパーソナルコンピュータであれば、「PrintScreen」というキーを押すことで、現在ディスプレイに表示されている画像を、画像ファイルとして保存することが可能なので、これらの機能を用いて画面キャプチャを行えばよい。

(4 - 3) さらに上記手順でキャプチャした画像を加工する。(4 - 1) で画面中に存在するリンクやフォームの情報を全て取得したが、それらの画面要素を強調するような画像を作成する。たとえば画面中にリンクが存在する場合は、リンク部分を赤い枠 (例えば操作マーク) で囲み、強調した画面の画像を作成する。作成した画像は、前手順と同じく画面画像記憶部に保存する。これらの処理の詳細は図 7 で述べる。

【 0 0 1 8 】

(4 - 4) (4 - 1) で取得したすべてのリンク、フォームについて、リンクであればリンク先にアクセスし、フォームであればフォーム送信を行い、画面遷移を行う。これを画面中の全てのリンク、フォームについて行う。なお、この処理については図 6 で詳細な説明を行う。

(4 - 5) 画面遷移を行った結果、HTTP リダイレクト応答を受け取ったかどうか判定する。

(4 - 6) HTTP リダイレクト応答を受け取った場合は、リダイレクト先のパスを画面情報記憶部に登録し、さらにリダイレクト先のパスに遷移する。この処理の詳細は図 5 で説明する。

(4 - 7) 遷移先の画面についての画面情報が、画面情報記憶部に登録されているかどうかをチェックする。

(4 - 8) 遷移先の画面についての画面情報が、画面情報記憶部に登録されていなければ、遷移先の画面を解析対象の画面として、画面解析処理を再帰的に呼び出す。

以上のように、画面中に含まれるリンクを次々と辿り、画面解析処理を再帰的に呼び出すことによって、Web アプリケーションを構成する全ての画面情報、画面画像を取得できる。

【 0 0 1 9 】

図 5 は、画面遷移図生成部が日記アプリケーションを解析し、完了した時点で、画面情報記憶部に登録される情報を示したものである。画面情報記憶部には、ある画面から画面に遷移する情報と、その遷移を行うときの操作内容についての情報と、画面キャプチャ画像のポインタ (後述) を格納する。「リンク元」列 (5 - 1) には各画面のパスを格納する。「リンク先」列 (5 - 2) には各画面から別画面に遷移する際の遷移先のパスを格納する。「リダイレクト先」列 (5 - 3) には画面遷移時に HTTP リダイレクト応答がある場合に、そのリダイレクト先のパスを格納する。「操作」列 (5 - 10) には、画面遷移時

10

20

30

40

50

に操作する内容を格納する。

例として、図2のトップページ画面(2-1)を解析する際の処理を説明する。まずトップページ画面のパスは“/”であるため、(5-4)に“/”を登録する。また図4の(4-1)で説明したようにトップページ画面を解析すると「日記を書く」「日記を見る」という二つのリンク要素(Aタグ)が見つかる。そこでリンク先のパスであるAタグのhref属性の値である“/add”、“/list”を(5-5)(5-6)に登録する。なお、図4の(4-5)の判定処理の結果は偽であるため、「リダイレクト先」には何も登録しない。さらに画面遷移時には「日記を書く」「日記を見る」リンクをクリックすればいいので、(5-11)(5-12)にそれぞれ、“「日記を書く」リンクをクリック”、“「日記を見る」リンクをクリック”を登録する。

10

もう一つの例として、図2の確認画面(2-3)を解析する際の処理を説明する。図2の説明でも述べたように、確認画面から日記表示画面(2-5)への画面遷移はリダイレクトによるものである。まず、確認画面のパスである“/confirm”を(5-7)に登録する。また確認画面に含まれるフォームのリクエスト先のパスは“/submit”であるため、それを(5-8)に登録する。次に、図4の(4-5)の判定処理により、HTTPリダイレクト応答に含まれるリダイレクト先パスである“/view”を(5-9)に登録する。また、画面遷移を行うには「確認」ボタンをクリックする必要があるので、(5-13)に“「確認」ボタンをクリック”を登録する。

このようにして、画面情報記憶部にすべての画面の情報が登録される。

【0020】

20

図6は、図4の(4-4)の処理の詳細について説明したものである。(4-4)では画面中の全てのリンク先にアクセス、もしくはフォーム送信して画面遷移を行っている。リンクの場合は、単純にリンク先URLに対してHTTPのGETメソッドを用いてアクセスすればよい。しかしフォームの場合はテキストフィールドやテキストエリアの値を入力した上でHTTPのPOST送信を行う必要がある。テキストフィールドやテキストエリアには適当な値を入力すればよいわけではなく、たとえば日記アプリケーションの日記登録処理であれば、日記のタイトルや日記の本文といった、意味のあるデータをセットする必要がある。そこで、それらの情報をWebアプリケーションが出力するアプリケーションログから取得する。

例えば(6-1)は「日記を書く」画面であるが、この画面には「タイトル」テキストフィールド、「本文」テキストエリア、「確認」ボタンが表示されている。この画面を構成するHTMLの一部を示したものが(6-2)である。

30

画面遷移図生成部は、(6-2)のHTMLを解析し、「タイトル」テキストフィールドに対応する

【0021】

次に画面遷移図生成部は、Webアプリケーションが出力したアプリケーションログを読み込む。(6-6)がアプリケーションログの例である。このログについて、HTTPメソッド名がPOST、かつリクエスト先のパスが「/confirm」、POST送信時のパラメータに「title」「body」を含む、ログのレコードを検索する。

40

すると(6-7)に示すログのレコードが見つかる。このログのレコードによるとPOST送信時のパラメータのうち「title」パラメータの値は“日記1”、「body」パラメータの値は“これは日記1です・・・”という文字列であることが分かる。

以上により画面遷移図生成部は、「日記を書く」画面において、どのような文字列をテキストフィールド、テキストエリアに設定するべきかを決定することができる。

【0022】

また、以上の処理によりフォーム送信時にはテキストエリアなどに入力を行う必要があることがわかるので、それを画面情報に追加登録する必要がある。例えば図6におけるHT

50

MLの解析により、「タイトル」テキストフィールド、「本文」テキストエリアに入力を行う必要があることが分かる。次にこれらのタグを識別する識別子を取得する。今回の例では、(6 - 4) (6 - 5) より各タグのname属性が識別子として使用できるので、「title」「body」を各タグの識別子とする。これにより、(5 - 1 4) (5 - 1 5) に、“titleテキストフィールドに入力”、“bodyテキストエリアに入力”という情報がセットされる。

【 0 0 2 3 】

図 7 は、(4 - 3) の処理の詳細について説明したものである。一般的に、Webアプリケーションの操作手順書において、例えば「トップページ画面中の“日記を書く”リンクをクリックしてください」という説明を行うときに、当該する画面ショット画像を合わせて掲載し、さらにどこをクリックすればよいのかを分かりやすくするために、画面ショット画像内のクリックすべき箇所を赤色の枠で囲むなどして強調して表示することがある。この処理を行うのが(4 - 3) の処理である。例えば(7 - 1) に示す「トップページ」画面の場合、「日記を書く」リンクや「日記を見る」リンクをクリックすることを説明するには(7 - 2) (7 - 3) に示すような画像を作成して操作手順書中に掲載すればよい。

10

図 4 の(4 - 1) の処理によって、画面中のリンクやフォームを構成するHTMLの要素の位置はわかっているので、画面遷移図生成部は、そのHTML要素のstyle属性にスタイル情報を付加するなどしてクリックすべき箇所を赤色の枠で強調表示させる。(7 - 4) はその例である。このようにして強調表示を追加した画面に対して、図 4 の(4 - 2) と同様に画面キャプチャ処理を行い、画面ショット画像ファイルとして画面画像記憶部に保存する。

20

【 0 0 2 4 】

また、(7 - 5) は、画面情報記憶部に格納されている画面情報の一部を示したものである。ここで画面情報と画面ショット画像の関連付けを行う。画面情報の「リンク元画像」「遷移操作画像」に、(4 - 2) (4 - 3) の処理で作成した画像ファイルのポインタを登録することで関連付けを行う。

例えば、(7 - 1) の画像は、(7 - 6) (7 - 7) に関連付けられる。また(7 - 2) の画像は(7 - 8) に、(7 - 3) の画像は(7 - 9) に関連付けられる。

上記の関連付け処理を全ての画像ファイルについて行う。

30

【 0 0 2 5 】

図 8 は、画面遷移図生成部によって蓄積された画面情報と画面画像から、画面遷移図を生成する様子を示したものである。画面遷移図生成部は、まず画面情報記憶部(8 - 1) に蓄積された情報から、Webアプリケーションの全体の画面遷移を再構成する。

たとえば画面情報記憶部の(8 - 3) (8 - 4) (8 - 5) の関係から、「トップページ」画面(8 - 2) から、「日記を書く」画面(8 - 6) と「日記一覧」画面(8 - 7) にリンクが張られていることが分かる。そこで、(8 - 1 8) (8 - 1 9) のポインタが示す画像ファイルを画面画像記憶部(8 - 8) から取得し、それぞれ画面遷移図上に描画する。次に、「トップページ」画面から「日記を書く」画面、「日記一覧」画面に向けて、(8 - 9) (8 - 1 0) に示すような矢印線を画面遷移図上に描画する。

40

また、(8 - 1 3) (8 - 1 4) (8 - 1 5) の関係から、リダイレクト処理による画面遷移が発生していることがわかるため、(8 - 2 1) (8 - 2 0) のポインタが示す、「確認」画面(8 - 1 1) と、日記表示画面(8 - 1 2) 画面の画像と、それらを結ぶ矢印線(8 - 1 6) を、それぞれ画面遷移図上に描画する。

このような処理を画面情報記憶部に登録されている情報すべてに対して行った結果、(8 - 1 7) に示すような図が完成する。

ここまでが、画面遷移図生成部の処理の詳細である。

【 0 0 2 6 】

図 9 は、画面遷移図が入出力装置に表示される様子を示したものである。ユーザはこの画面をみて、Webアプリケーションの画面遷移を確認することができる。ユーザは入出力

50

装置を使って、画面上に表示されたポインタ（ 9 - 1 ）を操作し、画面遷移図上の各画面を選択し、操作手順書を生成したい画面遷移の順序、範囲を指定する。

【 0 0 2 7 】

図 1 0 は、ユーザがポインタを使用して、操作手順書を生成したい画面遷移の順序、範囲を指定した様子を示したものである。この例では、ユーザは「日記を書くための手順書を生成したい」と考え、「トップページ」画面（ 1 0 - 1 ）、「日記を書く」画面（ 1 0 - 2 ）、「確認」画面（ 1 0 - 3 ）、日記表示画面（ 1 0 - 4 ）、の順番に選択した様子を示している。すると図に示すとおり、選択した画面とそれぞれの画面の間の矢印が強調表示される。

【 0 0 2 8 】

なお、ここで指定した画面遷移に対応する画面情報は、シナリオフロー記憶部にコピーされる。たとえば、図 1 0 のように画面遷移を指定すると、（ 5 - 1 6 ）（ 5 - 1 7 ）（ 5 - 1 8 ）（ 5 - 1 9 ）のレコードが、シナリオフロー記憶部に登録される。シナリオフロー記憶部に登録される情報の詳細については図 1 2 で説明する。

ユーザは画面の選択が完了したら「生成」ボタン（ 8 - 5 ）を押す。すると「シナリオ名の指定」ダイアログボックスが表示される。

【 0 0 2 9 】

図 1 1 は、「シナリオ名の指定」ダイアログボックス（ 1 1 - 1 ）が表示される様子を示したものである。ユーザは指定した画面遷移に対してシナリオ名をつけて、それをダイアログボックスの入力欄（ 1 1 - 2 ）に入力する。ここで指定したシナリオ名は、生成される操作手順書の見出し文字列として使用される。図では「日記の作成・登録」というシナリオ名を入力している。このシナリオ名は、シナリオフロー記憶部に保存される。

最後に「OK」ボタン（ 1 1 - 3 ）を押すことで、操作手順書が生成される。

操作手順書は、操作手順書テンプレート記憶部に格納されているテンプレートに、シナリオフロー記憶部に保存された情報を埋め込むことで生成する。

【 0 0 3 0 】

図 1 2 は、図 1 0 、図 1 1 における操作によって、シナリオフロー記憶部に登録された情報を示している。各レコードは画面情報記憶部に登録されたレコードを、項番を付加してコピーしたものである。この情報に沿って、操作手順書が生成されることになる。

【 0 0 3 1 】

図 1 3 は操作手順書テンプレートの一例を示したものである。テンプレートの生成方法は本発明では特に限定していない。図 1 3 で例示しているテンプレートは、{ } で囲まれた範囲がプログラムによって任意の文字列に置き換えられる、というものである。

【 0 0 3 2 】

図 1 4 が操作手順書テンプレートの生成処理の手順である。

（ 1 4 - 1 ）ユーザ操作によって作成されたシナリオフロー情報を 1 レコードずつ取得する。

（ 1 4 - 2 ）テンプレートに、シナリオフローのシナリオ名を埋め込む。例えば（ 1 2 - 1 ）が示す文字列を、（ 1 3 - 1 ）の箇所に埋め込む。

（ 1 4 - 3 ）テンプレートに、シナリオフローのレコードの項番を埋め込む。例えば（ 1 2 - 2 ）列に登録された番号を、（ 1 3 - 2 ）の箇所に埋め込む。

（ 1 4 - 4 ）テンプレートに、操作手順メッセージを埋め込む。操作手順メッセージについては図 1 5 で説明する。例えば（ 1 3 - 3 ）の箇所に操作手順メッセージを埋め込む。

（ 1 4 - 5 ）（ 1 4 - 1 ）で取得したレコードが、シナリオフローのレコードが最後のレコードかどうか判定する。例えば（ 1 4 - 1 ）で取得したレコードが、（ 1 2 - 3 ）のレコードと等しいかどうか判定する。

（ 1 4 - 6 ）（ 1 4 - 5 ）の判定結果が偽の場合は、シナリオフローのレコードの「遷移実行画像」が示す画面ショット画像を、テンプレートに埋め込む。例えば（ 1 2 - 4 ）列に登録されているポインタが示す画像を、（ 1 3 - 4 ）の箇所に埋め込む。

10

20

30

40

50

(1 4 - 7) (1 4 - 5) の判定結果が真の場合は、シナリオフローのレコードの「リンク元画像」が示す画面ショット画像を、テンプレートに埋め込む。例えば (1 2 - 5) 列に登録されているポインタが示す画像を、(1 3 - 4) の箇所に埋め込む。

(1 4 - 8) 最後にテンプレートを、操作手順書として出力する。図 1 6 ~ 図 1 9 がその出力例である。

【 0 0 3 3 】

図 1 5 は、(1 4 - 4) の操作手順メッセージ埋め込み処理について説明する図である。操作手順メッセージは、シナリオフロー情報の「操作」列に記述されている内容に対応するメッセージを、(1 5 - 1) に示すテーブルから取得する。たとえば、(1 2 - 6) には、(1 5 - 2) のメッセージが対応する。「{アンカーテキスト}」の部分は「日記を書く」に置換される。その結果、(1 6 - 1) のように表示される。また (1 2 - 7) (1 2 - 8) (1 2 - 9) には (1 5 - 3) (1 5 - 4) (1 5 - 5) のメッセージがそれぞれ対応する。「{ボタン名}」「{ id }」は、それぞれ「確認」「title」「body」に置換される。その結果 (1 7 - 1) (1 7 - 2) (1 7 - 3) のように表示される。これらのメッセージを箇条書きにまとめたものが、操作手順メッセージとしてテンプレートに埋め込まれる。

【図面の簡単な説明】

【 0 0 3 4 】

【図 1】本発明による Web アプリケーションの操作手順書生成システムと、操作手順書生成システムを動作させるための関連システムとから成る全体構成図である。

【図 2】本発明の一例として用いる日記アプリケーションの画面遷移図である。

【図 3】本発明の画面遷移図生成部の処理の流れ図である。

【図 4】本発明の画面解析処理の流れを示す図である。

【図 5】本発明の画面情報の例を示す図である。

【図 6】本発明のアプリケーションログからフォーム送信データのサンプルを取得する手順を示す図である。

【図 7】本発明の画面キャプチャ画像の作成と、画面情報との関連付けの説明図である。

【図 8】本発明の画面遷移図の構築の説明図である。

【図 9】本発明の出力装置上に表示された画面遷移図である。

【図 1 0】本発明のユーザによりシナリオフローが指定された状態を表示する画面図である。

【図 1 1】本発明のユーザによりシナリオフロー名が指定された状態を表示する画面図である。

【図 1 2】本発明のシナリオフロー情報の例を示す図である。

【図 1 3】本発明の操作手順書のテンプレートの例を示す図である。

【図 1 4】本発明の操作手順書生成部の処理の流れを示す図である。

【図 1 5】本発明の操作手順メッセージの処理の説明図である。

【図 1 6】本発明の生成される操作手順書の例 (1) を示す図である。

【図 1 7】本発明の生成される操作手順書の例 (2) を示す図である。

【図 1 8】本発明の生成される操作手順書の例 (3) を示す図である。

【図 1 9】本発明の生成される操作手順書の例 (4) を示す図である。

【符号の説明】

【 0 0 3 5 】

1 - 2 Web アプリケーション

1 - 4 画面情報記憶部

1 - 5 画面画像記憶部

1 - 6 シナリオフロー記憶部

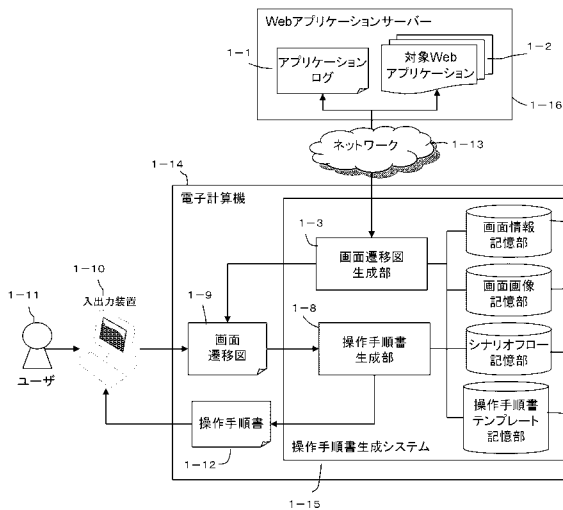
1 - 7 操作手順テンプレート記憶部

1 - 9 画面遷移図

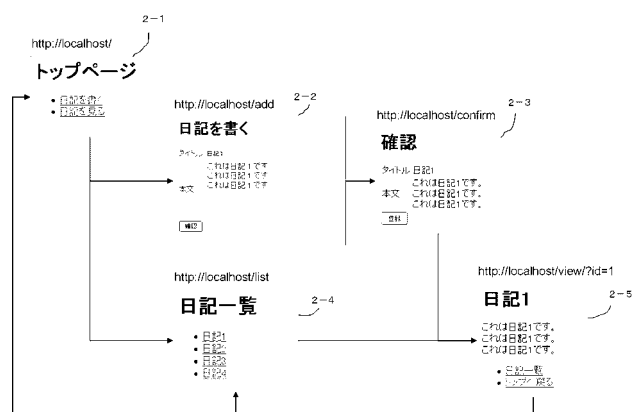
1 - 1 1 ユーザ

- 1 - 1 2 操作手順書
- 1 - 1 3 ネットワーク
- 1 - 1 4 電子計算機
- 1 - 1 5 操作手順書生成システム
- 1 - 1 6 Webアプリケーションサーバー

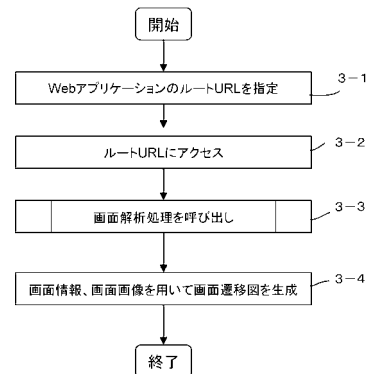
【図 1】



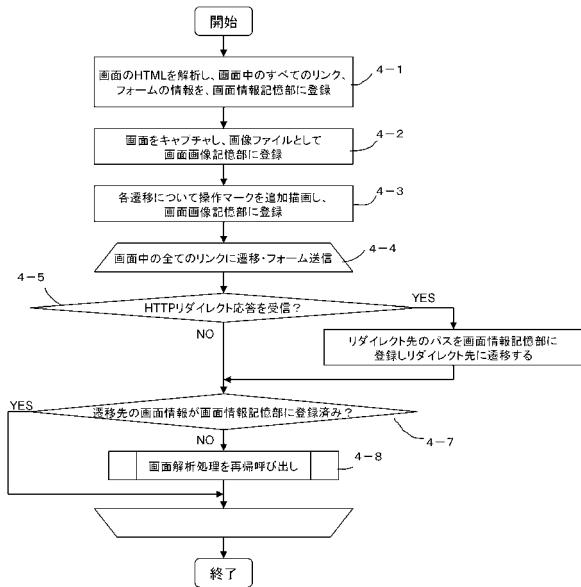
【図 2】



【図 3】



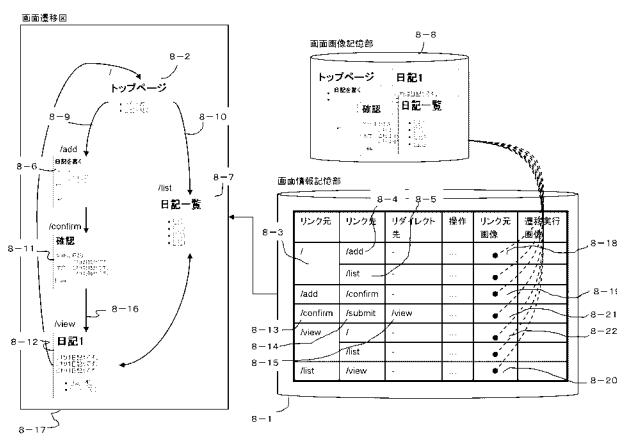
【図 4】



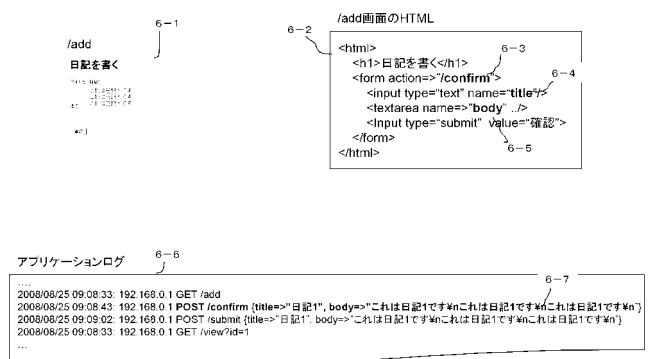
【図 5】

リンク元	リンク先	リダイレクト先	操作	リンク元画像	遷移実行画像
/	/add	-	「日記を書く」リンクをクリック	5-11	5-16
/	/list	-	「日記を見る」リンクをクリック	5-12	5-17
/add	/confirm	-	「title」テキストフィールドに入力 「body」テキストエリアに入力 「確認」ボタンをクリック	5-14 5-15	5-18
/confirm	/submit	/view	「登録」ボタンをクリック	5-13	5-19
/view	/	-	「トップへ戻る」リンクをクリック	-	-
/	/list	-	「一覧」リンクをクリック	-	-
/list	/view	-	日記リンクをクリック	-	-

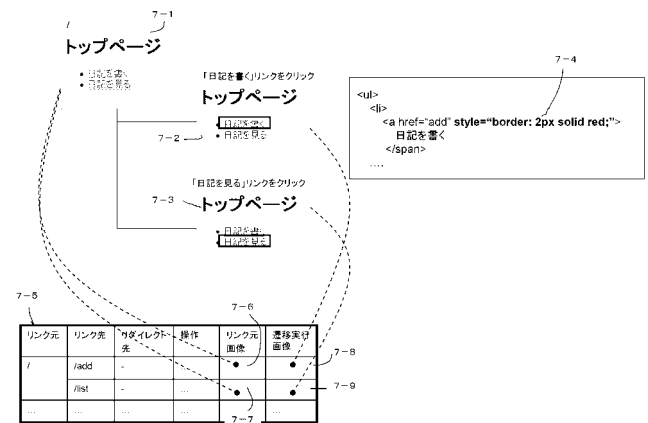
【図 8】



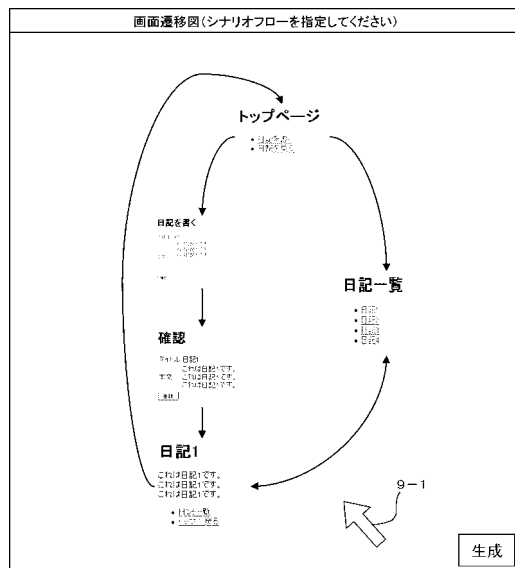
【図 6】



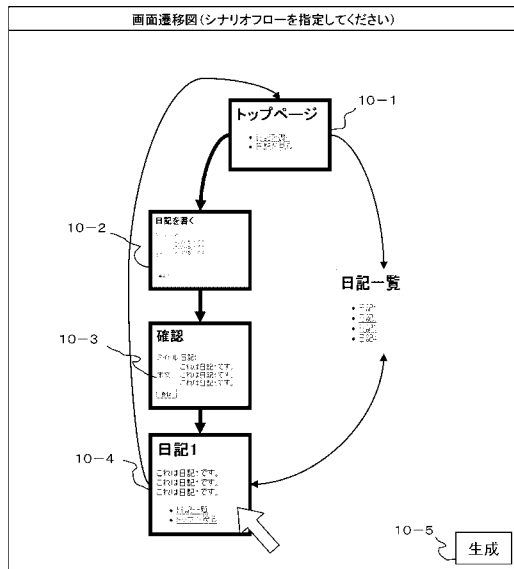
【図 7】



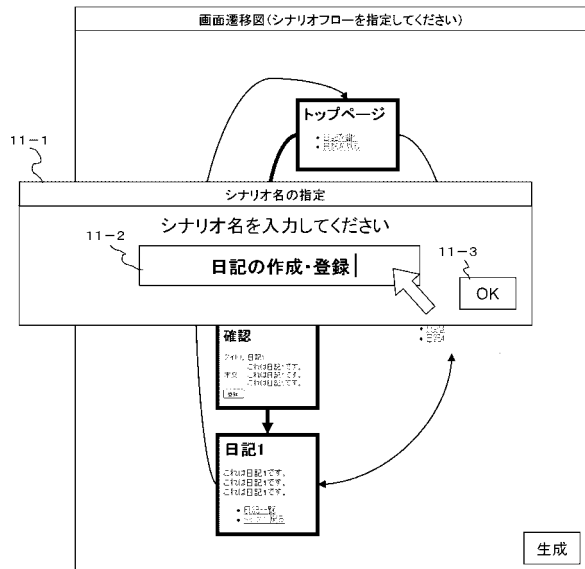
【図 9】



【図 10】



【図 11】

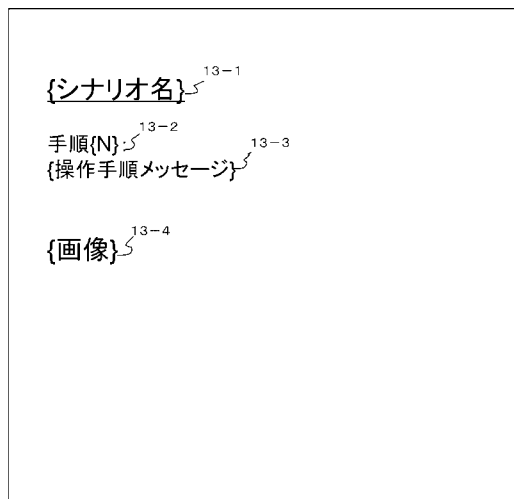


【図 12】

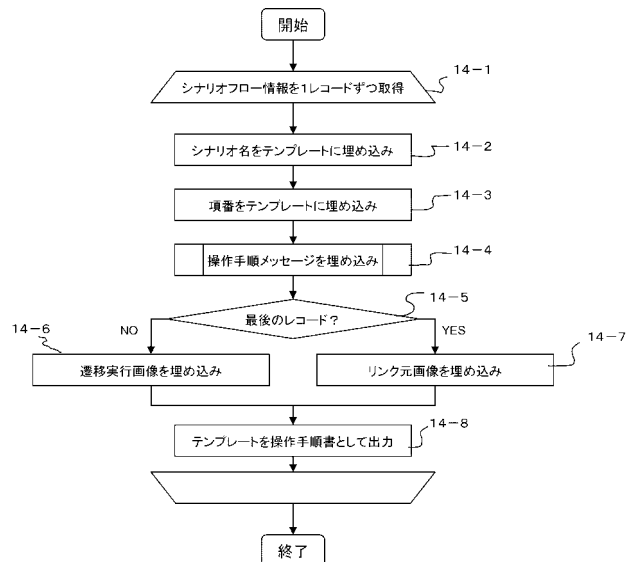
シナリオフロー(シナリオ名: 日記の作成・登録)

項番	リンク元	リンク先	リダイレクト先	操作	リンク元画像	遷移実行画像
1	/	/add	-	「日記を書く」リンクをクリック	12-6	...
2	/add	/confirm	-	「title」テキストフィールドに入力 「body」テキストエリアに入力 「確認」ボタンをクリック	12-7 12-8 12-9	...
3	/confirm	/submit	/view	「登録」ボタンをクリック	...	...
4	/view				...	...

【図 13】



【図 14】

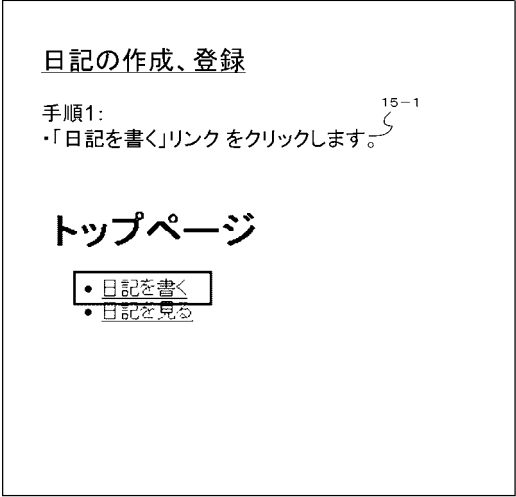


【図 15】

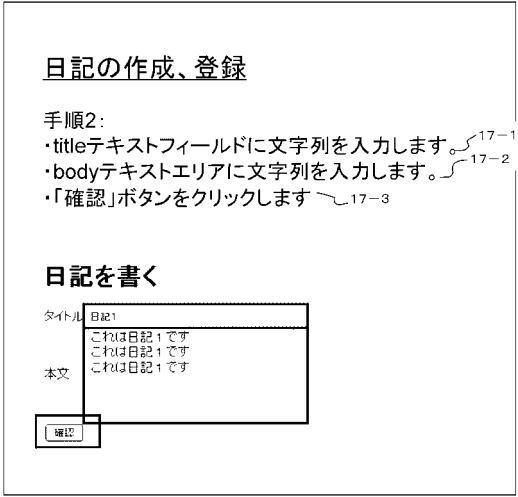
操作手順メッセージ

操作	メッセージ
リンクをクリック	{アンカーテキスト}リンクをクリックします。
ボタンをクリック	{ボタン名}ボタンをクリックします。
テキストボックスに入力	{id}テキストフィールドに文字列を入力します。
テキストエリアに入力	{id}テキストエリアに文字列を入力します。
チェックボックスをチェック	{id}チェックボックスをチェックします。
ラジオボタンをチェック	{id}ラジオボタンをチェックします。
...	...

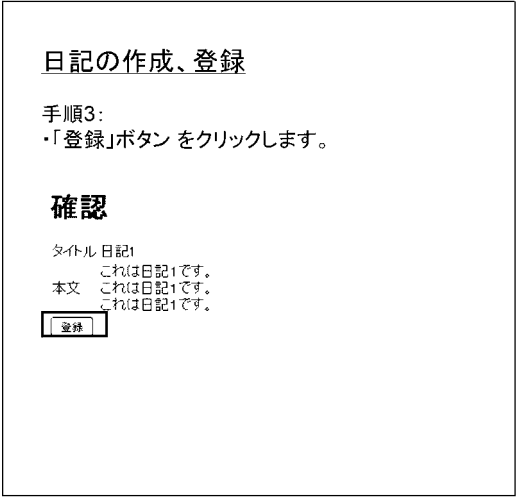
【 図 1 6 】



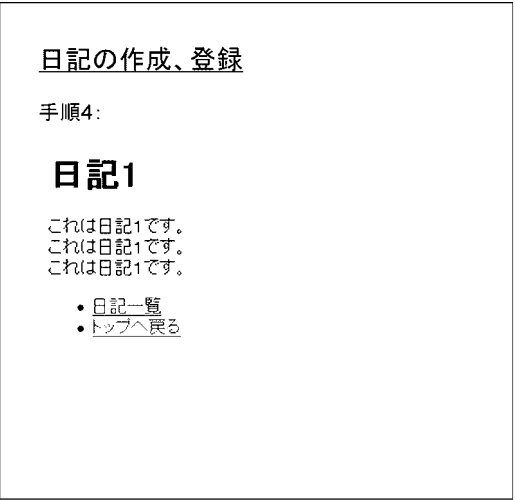
【 図 1 7 】



【 図 1 8 】



【 図 1 9 】



フロントページの続き

F ターム(参考) 5B176 DE03 DF00 DF03 DF07
5E501 AA01 AB15 AC19 AC22 AC34 BA05 CA02 CC17 DA16 EB11