

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5504212号  
(P5504212)

(45) 発行日 平成26年5月28日(2014.5.28)

(24) 登録日 平成26年3月20日(2014.3.20)

(51) Int.Cl.

F I

G06F 11/28 (2006.01)

G06F 11/28 340A

請求項の数 4 (全 25 頁)

<p>(21) 出願番号 特願2011-132517 (P2011-132517)</p> <p>(22) 出願日 平成23年6月14日 (2011.6.14)</p> <p>(65) 公開番号 特開2013-3729 (P2013-3729A)</p> <p>(43) 公開日 平成25年1月7日 (2013.1.7)</p> <p>審査請求日 平成25年5月2日 (2013.5.2)</p>	<p>(73) 特許権者 000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号</p> <p>(74) 代理人 110000176 一色国際特許業務法人</p> <p>(72) 発明者 松下 直樹 東京都品川区南大井六丁目26番2号 株式会社日立製作所 アプリケーション開発事業部内</p> <p>(72) 発明者 西脇 康人 東京都品川区南大井六丁目26番2号 株式会社日立製作所 アプリケーション開発事業部内</p> <p>審査官 相崎 裕恒</p> <p style="text-align: right;">最終頁に続く</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(54) 【発明の名称】 テストケース自動生成システム、テストケース自動生成方法、およびテストケース自動生成プログラム

(57) 【特許請求の範囲】

【請求項1】

システムの設計書のデータに基づいて前記システムのテストケースを自動生成するコンピュータであって、

各種設計書間に共通する項目について予め定義したテーブルを記憶した記憶部と、

テストケース別の、前記項目が記述に含まれたテスト内容の定義および出力条件について、入力部を介しユーザから指定を受けて、テスト情報データベースとして記憶部に格納する処理と、

処理対象となる設計書のデータを前記テーブルに照合し、該テーブルで定義された各項目に該当する情報を前記設計書のデータから抽出し、該抽出データを前記テーブルの項目に合わせて、設計情報データベースとして記憶部に記憶する処理と、

記憶部から前記設計情報データベースのデータを読み出し、該読み出したデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する処理と、

前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベースのデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する処理を実行する演算部と、

を備えるテストケース自動生成システム。

**【請求項 2】**

前記演算部は、

記憶部から読み出した前記設計情報データベースのデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行うに際し、前記出力条件が示す情報の組み合わせを正規表現とし、これを前記設計情報データベースのデータに対しパターンマッチングさせることを特徴とする請求項 1 に記載のテストケース自動生成システム。

**【請求項 3】**

システムの設計書のデータに基づいて前記システムのテストケースを自動生成すべく、各種設計書間に共通する項目について予め定義したテーブルを記憶した記憶部を備えたコンピュータが、

10

テストケース別の、前記項目が記述に含まれたテスト内容の定義および出力条件について、入力部を介しユーザから指定を受けて、テスト情報データベースとして記憶部に格納する処理と、

処理対象となる設計書のデータを前記テーブルに照合し、該テーブルで定義された各項目に該当する情報を前記設計書のデータから抽出し、該抽出データを前記テーブルの項目に合わせて、設計情報データベースとして記憶部に記憶する処理と、

記憶部から前記設計情報データベースのデータを読み出し、該読み出したデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する処理と、

20

前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベースのデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する処理と、

を実行することを特徴とするテストケース自動生成方法。

**【請求項 4】**

システムの設計書のデータに基づいて前記システムのテストケースを自動生成すべく、各種設計書間に共通する項目について予め定義したテーブルを記憶した記憶部を備えたコンピュータに、

30

テストケース別の、前記項目が記述に含まれたテスト内容の定義および出力条件について、入力部を介しユーザから指定を受けて、テスト情報データベースとして記憶部に格納する処理と、

処理対象となる設計書のデータを前記テーブルに照合し、該テーブルで定義された各項目に該当する情報を前記設計書のデータから抽出し、該抽出データを前記テーブルの項目に合わせて、設計情報データベースとして記憶部に記憶する処理と、

記憶部から前記設計情報データベースのデータを読み出し、該読み出したデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する処理と、

40

前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベースのデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する処理と、

を実行させることを特徴とするテストケース自動生成プログラム。

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、テストケース自動生成システム、テストケース自動生成方法、およびテストケース自動生成プログラムに関するものであり、具体的には、設計書におけるフォーマッ

50

トによらずテストケースを自動生成すると共に、ユーザによるテストケース編集も可能として、システム検証の効率化、品質低下の防止を図る技術に関する。

【背景技術】

【0002】

システム開発の高品質化、短納期化が求められる中で、高品質なシステムを短期間で開発するため、ソースの自動生成やテスト作業の自動化が検討されている。テスト作業の自動化に関しては、さらにテストケース、テストデータの自動生成や、テストの自動実行、テスト結果の自動検証なども検討されている。特にテストケースの自動生成に関しては、テスト対象となるプログラムに関する設計書をもとにテストケースを自動生成するシステムが知られている。

10

【0003】

そうした従来技術としては、例えば、テストケース及びテストデータを生成するためにコンピュータを、画面遷移定義情報からテストケース生成の対象画面を取得する対象画面取得手段、前記対象画面を構成する画面構成情報からテスト対象カラムを取得する画面内カラム取得手段、データ定義情報から、前記テスト対象カラムに対する制約を取得するカラムチェック情報取得手段、前記テスト対象カラムに対する前記制約とテストケーステンプレートをマッチングさせ、テストケースとテストデータを生成するテストケースデータ生成手段、として機能させることを特徴とするテストケース生成プログラム（特許文献1参照）などが提案されている。

【0004】

また、テストデータを構成する複数の項目情報と、該項目情報が漢字又は数字又は文字か否かの属性を示す属性情報と、前記項目情報のレコード長とを含むテストデータ仕様情報を複数格納する仕様書ファイル部と、テストデータの属性情報及びレコード長に対応した変数を含む複数の項目属性パラメータ情報を格納する項目属性パラメータファイル部と、前記仕様書ファイル部に格納したテストデータ仕様情報及び前記項目属性パラメータファイル部に格納した項目属性パラメータ情報を参照してコンピュータシステムのテストデータを内部テーブル部に生成する制御部とを備えるテストデータ作成システムであって、前記制御部が、前記仕様書ファイル部からテストデータ仕様情報を抽出する第1工程と、該第1工程により抽出したテストデータ仕様情報の属性情報及びレコード長に対応した複数の項目属性パラメータ情報を項目属性パラメータファイル部から抽出する第2工程と、前記第1工程により抽出したテストデータ仕様情報の複数の項目情報を該項目情報のレコード長に従って前記内部テーブル部に設定する第3工程と、該第3工程により設定した内部テーブル部の複数の項目情報に、前記該第2工程により抽出した項目属性パラメータ情報の変数に従ったデータを割り当てることによりテストデータを生成する第4工程とを実行するテストデータ作成システム（特許文献2参照）なども提案されている。

20

30

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特開2006-260390号公報

【特許文献2】特開2009-289083号公報

40

【発明の概要】

【発明が解決しようとする課題】

【0006】

従来技術を利用することでテストケースの自動生成は可能である。しかしながら、テストケースの自動生成に利用する設計書については、そのフォーマット等が固定化されているのが前提であって、例えばフォーマットや記載項目が固定基準と異なる設計書について適用することが出来ないという課題が残されている。

【0007】

また、従来技術によって出力されるテストケースの形式が固定であることから、設計書によっては不要な項目についてもテストケースが出力され、更にはその不要なテストケー

50

の実行によって、システム検証の作業量自体が増大するという問題も生じる。一方で、必要なテストケースが出力されないことも起こりうるため、テスト漏れによるシステムの品質低下の事態が生じる恐れもある。

【0008】

そこで本発明の目的は、設計書におけるフォーマットによらずテストケースを自動生成すると共に、ユーザによるテストケース編集も可能として、システム検証の効率化、品質低下の防止を図る技術を提供することにある。

【課題を解決するための手段】

【0009】

上記課題を解決する本発明のテストケース自動生成システムは、システムの設計書のデータに基づいて前記システムのテストケースを自動生成するコンピュータであって、各種設計書間に共通する項目について予め定義したテーブルを記憶した記憶部と、テストケース別の、前記項目が記述に含まれたテスト内容の定義および出力条件について、入力部を介しユーザから指定を受けて、テスト情報データベースとして記憶部に格納する処理と、処理対象となる設計書のデータを前記テーブルに照合し、該テーブルで定義された各項目に該当する情報を前記設計書のデータから抽出し、該抽出データを前記テーブルの項目に合わせて、設計情報データベースとして記憶部に記憶する処理と、記憶部から前記設計情報データベースのデータを読み出し、該読み出したデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する処理と、前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベースのデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する処理を実行する演算部と、を備える。

【0010】

また、本発明のテストケース自動生成方法は、システムの設計書のデータに基づいて前記システムのテストケースを自動生成すべく、各種設計書間に共通する項目について予め定義したテーブルを記憶した記憶部を備えたコンピュータが、テストケース別の、前記項目が記述に含まれたテスト内容の定義および出力条件について、入力部を介しユーザから指定を受けて、テスト情報データベースとして記憶部に格納する処理と、処理対象となる設計書のデータを前記テーブルに照合し、該テーブルで定義された各項目に該当する情報を前記設計書のデータから抽出し、該抽出データを前記テーブルの項目に合わせて、設計情報データベースとして記憶部に記憶する処理と、記憶部から前記設計情報データベースのデータを読み出し、該読み出したデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する処理と、前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベースのデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する処理と、を実行することを特徴とする。

【0011】

また、本発明のテストケース自動生成プログラムは、システムの設計書のデータに基づいて前記システムのテストケースを自動生成すべく、各種設計書間に共通する項目について予め定義したテーブルを記憶した記憶部を備えたコンピュータに、テストケース別の、前記項目が記述に含まれたテスト内容の定義および出力条件について、入力部を介しユーザから指定を受けて、テスト情報データベースとして記憶部に格納する処理と、処理対象となる設計書のデータを前記テーブルに照合し、該テーブルで定義された各項目に該当する情報を前記設計書のデータから抽出し、該抽出データを前記テーブルの項目に合わせて、設計情報データベースとして記憶部に記憶する処理と、記憶部から前記設計情報データベースのデータを読み出し、該読み出したデータと、前記テスト情報データベースにお

10

20

30

40

50

ける各テストケースの出力条件とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する処理と、前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベースのデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する処理と、を実行させることを特徴とする。

【発明の効果】

【0012】

本発明によれば、設計書におけるフォーマットによらずテストケースを自動生成すると共に、ユーザによるテストケース編集も可能として、システム検証の効率化、品質低下の防止を図ることができる。

10

【図面の簡単な説明】

【0013】

【図1】本実施形態のテストケース自動生成システムの構成例を示す図である。

【図2】本実施形態における設計書のフォーマット例を示す図である。

【図3】本実施形態における設計項目間の関係例を示す図である。

【図4】本実施形態における設計書の例を示す図である。

【図5】本実施形態における設計項目データの関係例を示す図である。

【図6】設計情報データベースとテスト情報データベースのテーブル構成例を示す図である。

20

【図7】本実施形態における設計書テーブルの例を示す図である。

【図8】本実施形態における設計項目テーブルの例を示す図である。

【図9】本実施形態における設計書詳細テーブルの例を示す図である。

【図10】本実施形態におけるテストケース自動生成方法の処理手順例1を示すフロー図である。

【図11】本実施形態における設計項目データテーブルの例を示す図である。

【図12】本実施形態における雛形ファイルの例を示す図である。

【図13】本実施形態における環境設定ファイルの例を示す図である。

【図14】本実施形態におけるテストケースの例を示す図である。

【図15】本実施形態におけるテストケーステーブルの例を示す図である。

30

【図16】本実施形態におけるテストケース詳細テーブルの例を示す図である。

【図17】本実施形態における出力条件テーブルの例を示す図である。

【図18】本実施形態におけるテストケース自動生成方法の処理手順例2を示すフロー図である。

【図19】本実施形態におけるテストケース自動生成方法の処理手順例3を示すフロー図である。

【発明を実施するための形態】

【0014】

以下に本発明の実施形態について図面を用いて詳細に説明する。図1は、本実施形態のテストケース自動生成システム100の構成例を示す図である。図1に示すテストケース自動生成システム100は、設計書におけるフォーマットによらずテストケースを自動生成すると共に、ユーザによるテストケース編集も可能として、システム検証の効率化、品質低下の防止を図るコンピュータシステムである。

40

【0015】

テストケース自動生成システム100のハードウェア構成は例えば以下の如くとなる。前記テストケース自動生成システム100は、ハードディスクドライブなど適宜な不揮発性記憶装置で構成される記憶部101、RAMなど揮発性記憶装置で構成されるメモリ103、前記記憶部101に保持されるプログラム102をメモリ103に読み出すなどして実行し装置自体の統括制御を行なうとともに各種判定、演算及び制御処理を行なうCPUなどの演算部104、ユーザからのキー入力等を受け付けるキーボードやマウス等の入

50

力部 105、処理データの表示を行うディスプレイ等の表示部 106 を備える。なお、記憶部 101 内には、本実施形態のテストケース自動生成システムとして必要な機能を実装する為のプログラム 102 と、各種テーブル、データ類が記憶されている。

【0016】

記憶部 101 に格納されているテーブルやデータの類としては、各種設計書間に共通する項目について予め定義したテーブル 125 (設計書テーブル 601、設計項目テーブル 602、設計書詳細テーブル 603、設計項目データテーブル 604 を含む) と、予めユーザから指定を受けているテスト情報データベース 126 (テストケーステーブル 605、テストケース詳細テーブル 606、出力条件テーブル 607 を含む) がある。また、その他にも、雛形ファイル 127、環境設定ファイル 128 など記憶部 101 に格納されている。これらデータベースやファイルに関しては後述する。

10

【0017】

こうしたテストケース自動生成システム 100 では、前記各テーブル 601 ~ 604 が示す項目に応じて、画面遷移定義情報 5 やデータ仕様情報 6 など構成される設計書 10 を設計情報データベース 125 に登録する。またテストケース自動生成システム 100 は、設計情報データベース 125、テスト情報データベース 126、および環境設定ファイル 128 を元に、雛形ファイル 127 を利用してテストケース 20 を自動生成することとなる。

【0018】

続いて、本実施形態のテストケース自動生成システム 100 が備える機能について説明する。上述したように、以下に説明する機能は、例えばテストケース自動生成システム 100 の演算部 104 がプログラム 102 をそれぞれ実行することで実装される機能と言える。

20

【0019】

前記テストケース自動生成システム 100 は、テストケース別の、前記項目が記述に含まれたテスト内容の定義 (テストケーステーブル 605、テストケース詳細テーブル 606) および出力条件 (出力条件テーブル 607) について、入力部 105 を介しユーザから指定を受けて、テスト情報データベース 126 として記憶部 101 に格納する機能を有している。

【0020】

また、前記テストケース自動生成システム 100 は、処理対象となる設計書 10 のデータを前記各テーブル 601 ~ 604 に照合し、該テーブルで定義された各項目に該当する情報を前記設計書 10 のデータから抽出し、該抽出データを前記テーブル 601 ~ 604 の項目に合わせて、設計情報データベース 125 として記憶部 101 にて記憶する機能を有している。

30

【0021】

設計書 10 はプログラム等を開発する上で必要となる設計項目を記載するものであり、同様のシステムを開発する場合、そのフォーマットはプロジェクト毎に異なるが、設計項目についてはプロジェクト間でほぼ共通となりやすい。そこで本実施形態のテストケース自動生成システム 100 は、設計書 10 のデータを、前記テーブル 601 ~ 604 からのデータ項目に合わせて抽出し設計情報データベース 125 へ導入する処理を行うことで、テストケース 20 の生成に利用する設計書 10 の間のフォーマットの違いを吸収することとなる。

40

【0022】

また、前記テストケース自動生成システム 100 は、記憶部 101 から前記設計情報データベース 125 のデータを読み出し、該読み出したデータと、前記テスト情報データベース 126 における各テストケースの出力条件 (出力条件テーブル 607 が示す情報) とのパターンマッチングを行って、前記出力条件が満たされたテストケース、ないし、出力条件が元々設定されていなかったテストケースについて特定する機能を有している。

【0023】

50

また、前記テストケース自動生成システム100は、前記特定したテストケースについて、前記テスト情報データベースの該当テストケースの定義が含む各項目に対応するデータを、前記読み出した設計情報データベース125のデータから特定し、該特定したデータを前記定義に設定することで該当テストケースを生成する機能を有している。

【0024】

なお、前記テストケース自動生成システム100は、記憶部101から読み出した前記設計情報データベース125のデータと、前記テスト情報データベース126における各テストケースの出力条件とのパターンマッチングを行うに際し、前記出力条件が示す情報の組み合わせを正規表現とし、これを前記設計情報データベース125のデータに対しパターンマッチングさせるとすれば好適である。

【0025】

こうした本実施形態のテストケース自動生成システム100によれば、テスト情報データベース126におけるテーブル605～607についてユーザがメンテナンス可能であり、テストケースに含ませるチェック条件や確認内容等、出力するテストケースの追加、変更、削除が可能となる。また、テストケースの出力条件の指定では正規表現を導入することで、パターンマッチングによるテストケースの出力制御を確実化かつ効率化できる。

【0026】

続いて、本実施形態で処理対象となる設計書10にデータ構成例等について説明しておく。図2は、本実施形態のテストケース自動生成システム100における処理対象となる設計書10のフォーマット例を示した図である。前記テストケース自動生成システム100（以下、システム100）が扱う設計書10の一例としては、1つの表計算ファイルに存在する1つのシート上において、画面構成の定義等が記述されたデータを想定している（勿論、画面構成に関する設計書はあくまで一例である）。一般的に、表計算ファイルにおけるシートは、値の入力可能な矩形セルが縦横に行列配置されて構成されたものであり、各セルは行列中での配置を示す座標値（例えば、“A列の3行目”であれば、「A3」）でシート上の絶対位置が特定できる。1つの表計算ファイルに含まれるシート数に制限は無いことが普通であり、複数の各設計書10を記述した複数のシートを、1つの表計算ファイルに含めておくことも可能である。

【0027】

図2においては、画面一覧11、画面項目定義書12、画面ボタン定義書13という3つの設計書フォーマットの例を示している。画面一覧11は、システム開発で設計される全画面の情報を一覧化して表現した設計書フォーマットである。図2の例では、1つの表計算ファイル中の1つのシート（“画面一覧”のタグがシート左下に存在する）に記述したものとなっている。

【0028】

一方、画面項目定義書12と画面ボタン定義書13は、前記画面一覧11にて定義された各画面上に存在するすべての画面項目とボタン項目の情報をそれぞれ一覧化して表現した設計書フォーマットである。図2の例では、1つの表計算ファイル中のそれぞれシート（計2シート）に記述したものとなっている。よって、シート左下に“画面項目定義書”、“画面ボタン定義書”の各タグが存在している。

【0029】

ここで、各シートの名、すなわちシート名は、処理対象の設計書10が、後述する設計書テーブル601のどのレコードに該当するものか識別するため、設計書テーブル601で定義された「設計書名称」と同一名称である必要がある。したがって、図2の例に示す設計書フォーマットを考えた場合、設計書テーブル601には「設計書名称」として「画面一覧」、「画面項目定義書」、「画面ボタン定義書」という3つのレコードが存在することになる。

【0030】

また、設計書10に記載される設計項目は、1つの設計書において複数存在することを想定している。図2の例の場合、画面一覧11においては、「画面ID」と「画面名」の

10

20

30

40

50

2つの設計項目がある。また、画面項目定義書12においては「画面ID」、「画面項目ID」、「画面項目名」、「コントロール」、「桁」、「必須」の6つの設計項目がある。また、画面ボタン定義書13においては、「画面ID」、「画面ボタンID」、「画面ボタン名」の3つの設計項目がある。

【0031】

なお、同一の設計項目が複数の設計書、すなわち図2の例の場合の画面一覧11、画面項目定義書12、画面ボタン定義書13らに存在することも想定できる。複数の設計書に存在する設計項目は、設計書間の紐付けを行う設計項目であり、同一の設計項目のデータ値が互いに同じである設計書同士は関連する設計書となる。図2の例では、「画面ID」が複数の設計書に記述されており、この「画面ID」が一致する設計書同士が関連する設計書となる。

10

【0032】

こうした設計書らのフォーマットに基づく設計項目間の関係例を図3にて示す。前記システム100が扱う各設計項目の間は、1:Nの親子関係があることを想定している。親となる設計項目は子となる設計項目を一意に決定する設計項目である。親となる設計項目は複数の子となる設計項目を持つことができるが、子となる設計項目は親となる設計項目を1つしか持てない。したがって、親の設計項目に相当する値が決まると、子となる設計項目に相当する値が自動的に決まる。

【0033】

図3に示す例では、図2に示した設計書フォーマット11~13らの設計項目の間の親子関係が表現されている。この図において、「画面ID」は画面を一意に識別する設計項目であり、この画面IDが決まると、「画面名」が自動的に決まるので、「画面ID」と「画面名」との間には「画面ID」が親かつ「画面名」が子という関係ができる。

20

【0034】

また、「画面ID」に対して「画面名」は1つなので、「画面ID」と「画面名」は1:1の親子関係となる。「画面項目ID」は画面上の項目を一意に識別するためのIDである。画面上の項目は画面が決まると自動的に決まるので、「画面ID」と「画面項目ID」の間には「画面ID」が親かつ「画面項目ID」が子という関係ができる。また画面には複数の画面項目が存在するため、「画面ID」と「画面項目ID」の間は1:Nの親子関係となりうる。

30

【0035】

また、「画面項目名」、「コントロール」、「桁」は「画面項目ID」が決まると自動的に決まる項目であり、「画面項目ID」と「画面項目名」、「コントロール」、「桁」には「画面項目ID」が親かつ「画面項目名」、「コントロール」、「桁」が子という関係ができる。また、「画面項目ID」に対して「画面項目名」、「コントロール」、「桁」はそれぞれ1つなので、「画面項目ID」と「画面項目名」、「コントロール」、「桁」の各間は1:1の親子関係となる。

【0036】

また、「画面ボタンID」は「画面項目ID」と同様の理由から、「画面ID」が親かつ「画面ボタンID」が子という1:Nの親子関係となる。さらに、「画面ボタン名」は「画面項目名」、「コントロール」、「桁」と同様の理由から、「画面ボタンID」が親かつ「画面ボタン名」が子という1:1の親子関係となる。

40

【0037】

このようなフォーマットに基づいてユーザが作成した設計書10としては、画面一覧14、画面項目定義書15、および画面ボタン定義書16の各例を図4に示す。図4に示す例では、開発対象となる画面数は2つあり、画面一覧14には2つの画面の情報が一覧形式で記述されている。また、画面項目定義書15と画面ボタン定義書16は、1画面についてそれぞれ作成した例であり、画面項目定義書15では画面上に存在する2つの画面項目の情報が一覧形式で記述されており、画面ボタン定義書16では画面上に存在する1つのボタンの情報が一覧形式で記述されている。

50



## 【 0 0 3 8 】

こうした設計書について図3同様に設計項目間の関係を示したのが図5となる。図4で例示した設計例では、画面には2つの画面項目と1つのボタン項目が存在している。そこで、画面ID: SCREEN01には、1つの画面名: ユーザ登録画面と、2つの画面項目ID: user Id、user Nameと、1つのボタン項目ID: insertが子の設計項目データとして存在する。さらに画面項目ID: user Idには画面項目名: ユーザIDと、コントロール: テキストと、桁: 10と、必須: 、という子となる4つの設計項目データが存在し、画面項目ID: user Nameには、画面項目名: ユーザ名と、コントロール: テキストと、桁: 5と、必須: -、という子となる4つの設計項目データが存在する。また、画面ボタンID: insertには、画面ボタン名: 登録という子となる1つの設計項目データが存在する。

10

## 【 0 0 3 9 】

続いて、前記システム100が利用する設計情報データベース125とテスト情報データベース126のテーブル構成例について図6~9に基づき説明する。

## 【 0 0 4 0 】

設計情報データベース125は、設計書テーブル601、設計項目テーブル602、設計書詳細テーブル603、および設計項目データテーブル604の4つのテーブルから構成される。

## 【 0 0 4 1 】

設計書テーブル601は、前記システム100の入力となる設計書10に関する情報を保持するためのテーブルである。この設計書テーブル601は、図7にも示すように、設計書ID701と設計書名称702の2つの項目からなる。設計書ID701は設計書テーブル601の主キーとなる項目であり、入力となる設計書を一意に識別するためのIDである。設計書名称702は入力となる設計書の日本語名称を表すものである。設計書名称702は、図2で説明したとおり、ユーザが指定した設計書がどの設計書なのかを識別するために、入力となる設計書のシート名と同一である必要がある。

20

## 【 0 0 4 2 】

上記図2の例では、設計書には画面一覧14、画面項目定義書15、および画面ボタン定義書16の3つの設計書があるため、図7に示す設計書テーブル601の例では、各設計書3件の情報が登録されている。設計書ID701には一意となるIDを設定する必要があり、画面一覧14、画面項目定義書15、画面ボタン定義書16に対する各設計書IDには、それぞれ重複しない値として、例えば、「ScreenList」、「ScreenItem」、「ScreenButton」を設定している。設計書名称702には、図2で説明したとおり表計算ファイルのシート名と一致する必要があるため、設計書名称702にはそれぞれ、「画面一覧」、「画面項目定義書」、「画面ボタン定義書」を設定している。

30

## 【 0 0 4 3 】

また、設計項目テーブル602はシステム開発する上で必要となる、設計書に記載された設計項目の情報を保持するためのテーブルである。この設計項目テーブル602は、図8にて示すように、設計項目ID801、設計項目名称802、親設計項目ID803の3つの項目からなる。設計項目ID801は設計項目テーブル602の主キーとなる項目であり、設計項目を一意に識別するためのIDである。設計項目名称803は入力となる設計書に記載された設計項目の日本語名称を表すものである。親設計項目ID802は、図3で説明したとおり、設計項目間に1:Nの親子関係があることを踏まえ、子となる設計項目に対して親となる設計項目の設計項目IDを設定するための項目である。

40

## 【 0 0 4 4 】

図8に示す設計項目テーブル602の例においては、設計項目として、「画面一覧」について「画面ID」と「画面名」の2つの設計項目が、「画面項目定義書」について「画面ID」、「画面項目ID」、「画面項目名」、「コントロール」、「桁」、「必須」の6つの設計項目が、「画面ボタン定義書」について「画面ID」、「画面ボタンID」、

50

「画面ボタン名」の3つの設計項目が存在する。また「画面ID」は「画面一覧」、「画面項目定義書」、「画面ボタン定義書」のすべての設計書に存在しており、設計項目としては重複を除き「画面ID」、「画面名」、「画面項目ID」、「画面項目名」、「コントロール」、「桁」、「必須」、「画面ボタンID」、「画面ボタン名」の9つの設計項目が存在する。したがって、設計項目テーブル602には、9件の情報が登録されている。設計項目ID801には一意となるIDを設定する必要があり、「画面ID」、「画面名」、「画面項目ID」、「画面項目名」、「コントロール」、「桁」、「必須」、「画面ボタンID」、「画面ボタン名」に対する設計項目IDにはそれぞれ重複しない値として、「ScreenId」、「ScreenName」、「ScreenItemId」、「ScreenItemName」、「ScreenItemCtl」、「ScreenItemSize」、「ScreenItemNecessary」、「ScreenButtonId」、「ScreenButtonName」を設定している。

10

## 【0045】

また、設計項目名称803は、設計項目の日本語名称であるため、「画面ID」、「画面名」、「画面項目ID」、「画面項目名」、「コントロール」、「桁」、「必須」、「画面ボタンID」、「画面ボタン名」といった名称を設定している。さらに親項目の設計項目ID802には、設計項目間に図3に示す親子関係があるため、「画面ID」に関しては未設定を示す「-」が設定され、「画面名」、「画面項目ID」、「画面ボタンID」に関しては「ScreenId」が設定され、「画面項目名」、「コントロール」、「桁」、「必須」に関しては「ScreenItemId」が設定され、「画面ボタン名」

20

## 【0046】

また、設計書詳細テーブル603は、設計書10と設計項目とを紐付けるためのテーブルであり、設計書10のどの位置にどの設計項目が存在するかという情報を保持するためのテーブルである。設計書詳細テーブル603は、図9に例示するように、設計書ID901、設計項目ID902、絶対位置903、繰返し有無904、データ読込方向905、登録有無906の6つの項目からなる。

## 【0047】

設計書ID901と設計項目ID902は設計書詳細テーブル603の主キーとなる項目であり、どの設計書にどの設計項目が記載されるかを示すものである。設計書ID901は設計書テーブル601に存在する設計書ID701である必要があり、設計項目ID902は設計項目テーブル602に存在する設計項目ID801である必要がある。絶対位置903は設計書のどの位置に設計項目が存在するかを示す情報である。図2をもとに例を示すと、設計書：画面一覧にある設計項目：画面IDに対するデータは「A2」の位置から開始されるため、絶対位置903には「A2」が設定される。また、繰返し有無904は、設計項目のデータが設計書間で1つなのか複数なのかを示す情報であり、1つの場合には「無」が、複数の場合には「有」が設定される。図2の例をもとに説明すると、設計書：画面一覧にある設計項目：画面IDに対するデータは複数記述されるため、繰返し有無904には「有」が設定される。一方、設計書：画面項目定義書にある設計項目：画面IDに対するデータは1つしか記述されないため、繰返し有無904には「無」が設定される。また、データ読込方向905は設計項目のデータが複数の場合に利用され、データの繰返し方向を示すものであり、該当設計項目が上方向に繰返される場合には「」が、下方向に繰返される場合には「」が、左方向に繰返される場合には「」が、右方向に繰返される場合には「」が設定される。また設計項目のデータが1つの場合には「-」が設定される。図2をもとに例を示すと、設計書：画面一覧にある設計項目：画面IDに対するデータは下方向に複数記述されるため、データ読込方向905には「」が設定される。一方、設計書：画面項目定義書にある設計項目：画面IDに対するデータは1つしか記述されないため、データ読込方向905には「-」が設定される。また、登録有無906は複数の設計書に記載される設計項目に関して、どの設計書から設計項目データを登録するかを示す情報である。複数の設計書から同一の設計項目データを登

30

40

50

録しても無駄な情報が増えてしまうだけなので、設計項目データは1つの設計書からしか登録できないようにしている。したがって、ある設計項目について、ある設計書で登録有無906が「有」になっている場合、その設計項目について他の設計書に関して「無」とする必要がある。図2をもとに例を示すと、設計項目：画面IDは3つの設計書に記載されるため、このうちの1つの設計書でしか登録有無906を「有」に設定できない。したがって、設計書：画面一覧にて登録有無906を「有」に設定した場合、設計書：画面項目定義書と設計書：画面ボタン定義書では登録有無906を「無」にしか設定できない。

#### 【0048】

図9に例示した設計詳細テーブルでは、設計書ID901が「ScreenList」かつ設計項目ID902が「ScreenId」と、設計書ID901が「ScreenList」かつ設計項目ID902が「ScreenName」という2つのレコードが登録されている。図2の例であれば、「画面ID」は「A2」の位置から下方向に複数記述されるため、設計書ID901が「ScreenList」かつ設計項目ID902が「ScreenId」のレコードの絶対位置903に「A2」が、繰返し有無904には「有」が、データ読込方向905には「」が設定される。また、「画面ID」は複数の設計書に記述されており、この例では「画面一覧」から登録するため、登録有無906を「有」にしている。したがって、「画面ID」が記述されている画面項目定義書15と画面ボタン定義書16では登録有無906を「無」にしている。同様に「画面名」は「B2」の位置から下方向に複数記述されているため、設計書ID901が「ScreenList」かつ設計項目ID902が「ScreenName」のレコードの絶対位置903に「B2」が、繰返し有無904には「有」が、データ読込方向905には「」が設定される。また「画面名」は「画面一覧」にのみ記述されているため、登録有無906は「有」に設定されている。他の設計書についても同様に設定することができる。こうした、設計書のデータをテーブル11～13の項目に基づいて抽出・設定し、設計情報データベース125を作成する処理は、前記システム100が実行する。表計算ファイルのシートは上述のように絶対位置が判明しているセルによって構成されているから、前記システム100は、設計書10からのデータ抽出等の際には、シート中の規定セル(例：“A1”など最上行の最左列のセル)から順次セルの値を読み込んで、該当設計項目のIDや名称が設定されたセルを特定し、それと連結ないし隣接したセルの値すなわち設計項目の値を読み取るといった処理を実行していけばよい。他のテーブル類に基づく設計書10からのデータ抽出についても同様である。

#### 【0049】

また、設計項目データテーブル604は、設計書10から読込んだ設計項目のデータを保持するためのテーブルである。設計項目データテーブル604はデータ番号1101、設計項目ID1102、値1103、親項目のデータ番号1104という4つの項目からなる。データ番号1101は設計項目のデータを一意に識別するための情報であり、設計項目のデータを設計項目データテーブル604に格納する際にシステム100が自動採番して作成する。設計項目ID1102は設計項目のデータがどの設計項目に関するものかを示すものであり、設計項目テーブル602に存在する設計項目ID801である必要がある。値1103は設計書10から読込んだ設計項目のデータを格納する。親項目のデータ番号1104は、図5で説明したとおり、設計項目間における1:Nの親子関係に基づいて、子となる設計項目データに対して親となる設計項目データのデータ番号を設定する項目である。親項目のデータ番号1104は設計項目データテーブル604に存在するデータ番号1101である必要がある。

#### 【0050】

一方、テスト情報データベース126は、テストケーステーブル605、テストケース詳細テーブル606、出力条件テーブル607の3つのテーブルから構成される。テストケーステーブル605は生成するテストケース1件の情報を保持するテーブルである。テストケーステーブル605は、図15に示すようにテストケースID1501、テストケース名1502、出力条件ID1503の3つの項目からなる。テストケースID150

10

20

30

40

50

1 はテストケーステーブル 605 の主キーとなる項目であり、テストケースを一意に識別するための ID である。テストケース名 1502 はテストケースの日本語名称を表すものである。出力条件 ID 1503 はテストケースを出力するための条件を表すものであり、ここに値を設定することで、テストケースの出力制御が可能となる。出力条件 ID 1503 は出力条件テーブル 607 に存在する出力条件 ID 1701 である必要がある。

#### 【0051】

また、テストケース詳細テーブル 606 は、テストケースを実行する手順を操作単位で格納するテーブルである。図 16 に示すように、テストケース詳細テーブル 606 はテストケース ID 1601、操作番号 1602、出力区分 1603、操作内容 1604 の 4 つの項目からなる。テストケース ID 1601 と操作番号 1602 はテストケース詳細テーブル 606 の主キーとなる項目であり、テストケースを実現するための手順を構成する 1 つの操作を示すものである。テストケース ID 1601 はテストケーステーブル 605 に存在するテストケース ID 1501 である必要がある。出力区分 1603 は操作の区分を示すものであり、テストケースのチェック条件となる操作に対しては「条件」を、テストケースの確認内容となる操作に対しては「確認」を設定する。操作内容 1604 はテストする際の具体的な操作を示すものである。チェック条件となる操作としては「画面項目に値を入力する」、「ボタンを押す」などの操作があり、確認内容となる操作としては「ボタン押下後の画面状態を確認する」などの操作がある。また、操作内容 1604 には、設計情報データベース 125 の設計項目テーブル 602 における設計項目 ID 801 が埋め込み可能であり、設計項目 ID 801 を「{設計項目 ID}」という形式で設定することで対応する設計項目のデータを出力可能にしている。ここで設定する設計項目 ID は、設計項目テーブル 602 に存在する設計項目 ID 801 である必要がある。さらに、出力するファイルで利用可能な関数を記述可能にすることで、設計項目を利用した関数による演算結果も出力することが可能である。

#### 【0052】

また、出力条件テーブル 607 はテストケースの出力を制御するための条件情報を格納するテーブルである。出力条件テーブル 607 は出力条件 ID 1701、設計項目 ID 1702、出力条件内容 1703 の 3 つの項目からなる。出力条件 ID 1701 は出力条件テーブル 607 の主キーとなる項目であり、出力条件を一意に識別する ID である。設計項目 ID 1702 と出力条件内容 1703 はテストケースを出力するための条件を表すものであり、設計項目 ID 1702 に対する設計項目データが出力条件内容に指定されたパターンを満たす場合にテストケースを出力する。設計項目 ID 1702 は設計項目テーブル 602 に存在する設計項目 ID 801 である必要がある。

#### 【0053】

続いて、本実施形態における設計書 10 のデータを設計情報データベース 125 に登録する際の処理を図 10 に基づき説明する。なお、上述したように、前記システム 100 が、設計書 10 のデータから、設計書テーブル 601、設計項目テーブル 602、および設計書詳細テーブル 603 の各設計項目について値を抽出し、該当テーブルに対応した形式で記憶部 101 にデータを保持しているものとする。

#### 【0054】

この場合、前記システム 100 は、例えば、入力部 105 にてユーザより入力されたフォルダ名の情報を取得する（ステップ 110）。システム 100 は、例えば記憶部 101 に格納されていた該当フォルダにアクセスし、設計書 10 を格納した表計算ファイルを全て取得する（ステップ 120）。

#### 【0055】

次に前記システム 100 は、前記ステップ 120 で取得した表計算ファイルを 1 つずつ読んでいく。読込む表計算ファイルがある場合（ステップ 130：有）、前記システム 100 はステップ 140 に処理を進める。一方、読込む表計算ファイルが無い場合（ステップ 130：無）、前記システム 100 は処理を終了する。こうしたステップ 130 の処理は、前記ステップ 120 で得た全ての表計算ファイルについて、読込む表計算ファイル

10

20

30

40

50

が無くなるまで繰り返し実行される。

【 0 0 5 6 】

続いて前記システム 1 0 0 は、前記ステップ 1 2 0 で取得した表計算ファイルから、各設計書 1 0 のデータが記述されたシートを取得する（ステップ 1 4 0）。前記システム 1 0 0 は、取得した表計算ファイルの全シートを 1 シートずつ読んでいく（ステップ 1 5 0）。読むシートがある場合（ステップ 1 5 0：有）、前記システム 1 0 0 は処理をステップ 1 6 0 に進める。他方、読むシートが無い場合（ステップ 1 5 0：無）、前記システム 1 0 0 は、処理をステップ 1 3 0 に戻す。こうした処理は、前記ステップ 1 4 0 で取得した全シートについて、読むシートが無くなるまで繰り返し実行する。

【 0 0 5 7 】

続いて前記システム 1 0 0 は、読んだシートからシート名を取得する（ステップ 1 6 0）。また前記システム 1 0 0 は、設計情報データベース 1 2 5 の設計書テーブル 6 0 1 にアクセスし、設計書テーブル 6 0 1 の設計書名称 7 0 2 が前記取得したシート名と一致するレコードの設計書 ID 7 0 1 を取得する（ステップ 1 7 0）。

【 0 0 5 8 】

次に前記システム 1 0 0 は、前記ステップ 1 7 0 で設計書 ID 7 0 1 を取得できた場合（ステップ 1 8 0：有）、処理をステップ 1 9 0 に進める。一方、前記ステップ 1 7 0 にて設計書 ID 7 0 1 を取得できなかった場合（ステップ 1 8 0：無）、前記システム 1 0 0 は処理をステップ 1 5 0 に戻す。

【 0 0 5 9 】

続いて前記システム 1 0 0 は、設計情報データベース 1 2 5 の設計書詳細テーブル 6 0 3 にアクセスし、設計詳細テーブルの設計書 ID 9 0 1 が前記ステップ 1 7 0 で取得した設計書 ID 7 0 1 と一致するレコード（設計書詳細レコード）を取得する（ステップ 1 9 0）。前記システム 1 0 0 は、取得した設計書詳細レコードを 1 つずつ読んでいく。読む設計書詳細レコードがある場合（ステップ 2 0 0：有）、前記システム 1 0 0 は、処理をステップ 2 1 0 に進める。他方、読む設計書詳細レコードが無い場合（ステップ 2 0 0：無）、前記システム 1 0 0 は、処理をステップ 1 5 0 に戻す。こうした処理は、前記ステップ 1 9 0 で取得した全設計書詳細レコードについて、読む設計書詳細レコードが無くなるまで繰り返し実行される。

【 0 0 6 0 】

続いて前記システム 1 0 0 は、前記読み込んだ設計書詳細レコードから設計項目 ID 9 0 2、絶対位置 9 0 3、繰り返し有無 9 0 4、データ読み方向 9 0 5、登録有無 9 0 6 の各値を取得する（ステップ 2 1 0）。そして前記システム 1 0 0 は、ここで取得した登録有無 9 0 6 が「有」の場合（ステップ 2 2 0：有）、処理をステップ 2 3 0 に進める。他方、登録有無 9 0 6 が「無」の場合（ステップ 2 2 0：無）、前記システム 1 0 0 は処理をステップ 2 0 0 に戻す。

【 0 0 6 1 】

次に前記システム 1 0 0 は、設計情報データベース 1 2 5 の設計項目テーブル 6 0 2 にアクセスし、設計項目テーブル 6 0 2 の設計項目 ID 8 0 1 が前記ステップ 2 1 0 で取得した設計項目 ID 9 0 2 と一致するレコードにおける、親項目の設計項目 ID 8 0 2 を取得する（ステップ 2 3 0）。

【 0 0 6 2 】

前記システム 1 0 0 は、親項目の設計項目 ID 8 0 2 を取得できた場合（ステップ 2 4 0：有）、処理をステップ 2 5 0 に進める。他方、親項目の設計項目 ID 8 0 2 を取得できなかった場合（ステップ 2 4 0：無）、前記システム 1 0 0 は処理をステップ 2 6 0 に進める。

【 0 0 6 3 】

続いて前記システム 1 0 0 は、記憶部 1 0 1 にて保持する所定のデータ番号（後述するステップ 2 8 0 で採番したもの）中から、前記ステップ 2 3 0 で得た親項目の設計項目 ID 8 0 2 に対応するデータ番号を、親項目のデータ番号として取得する（ステップ 2 5 0

10

20

30

40

50

)。勿論、前記ステップ170～ステップ240の処理は、設計項目テーブル602における各設計項目間の親子関係に沿って最上位の親から順次行っていくから、当初は未だデータ番号が採番されていない状況であり、親項目のデータ番号は取得できない。よって、前記親子関係において親をもつ設計項目に関する処理が行われるまで、当該ステップはスルーする。

#### 【0064】

次に前記システム100は、前記ステップ140より処理中となっている該当シートにおいて、絶対位置903にあるセルからデータを取得する(ステップ260)。前記ステップ210で得た繰返し有無904が「有」の場合、前記システム100はデータ読込方向905を元に全データを取得する。

10

#### 【0065】

続いて前記システム100は、前記ステップ260で取得した全データから1データずつ読んでいく。読込むデータがある場合(ステップ270:有)、前記システム100はステップ280に処理を進める。他方、読込むデータが無い場合(ステップ270:無)、前記システム100は処理をステップ200に戻す。こうした処理は、前記ステップ260で得た全データについて、読込むデータが無くなるまで繰返し実行される。

#### 【0066】

次に前記システム100は、前記ステップ260で得た各設計項目のデータを設計項目データテーブル604に登録すべく、該当データ(設計項目データ)用のデータ番号を自動採番する(ステップ280)。データ番号の自動採番については、登録順に1つずつ番号をインクリメントするといった一般的手法を採用すればよい。

20

#### 【0067】

続いて前記システム100は、設計項目データテーブル604におけるデータ番号1101の欄に、前記ステップ280で自動採番したデータ番号を、また同様に、設計項目ID1102の欄に前記ステップ210で取得した設計項目ID902を、また同様に、値1103の欄に前記ステップ260で取得したデータを、また同様に、親項目のデータ番号すなわち親データ番号1104の欄に前記ステップ250で取得したデータ番号を設定し、データ登録を行う(ステップ290)。前記システム100は、前記ステップ280で自動採番したデータ番号を記憶部101に保持し、ステップ200に戻る(ステップ300)。以降、前記システム100は、設計書詳細レコードがある限り(ステップ200:無 ステップ150)、またシートがある限り(ステップ150:無 ステップ130)、更に、表計算ファイルが無くなるまで(ステップ130:無 END)、上記処理を繰返し実行することとなる。

30

#### 【0068】

図11は、図4の設計書10を上述のフローを経て登録した設計項目データテーブル604の例を示す図である。ここでは一例として、設計書10のうち画面一覧14について説明する。システム100は画面一覧14のデータから「画面ID」と「画面名」を抽出し登録することになる。まず、「画面ID」は「SCREEN01」と「SCREEN02」の2つがあるため、システム100は、データ番号1101には自動採番された「1」と「2」を、設計項目ID1102には「画面ID」の設計項目ID:ScreenIdを、「値」には「SCREEN01」と「SCREEN02」をそれぞれ設定する。なお親データ番号1104は、「画面ID」に対する親項目の設計項目が無い場合未設定となる。また、「画面名」は「ユーザ登録画面」と「結果画面」の2つがあるから、前記システム100は、データ番号1101には自動採番した「3」と「4」を、設計項目ID1102には画面名の設計項目ID:ScreenNameを、値1103には「ユーザ登録画面」と「結果画面」をそれぞれ設定する。また「画面名」に対する親項目の設計項目が「画面ID」であるため、前記システム100は、「ユーザ登録画面」の親データ番号1104として「1」を、また「結果画面」の親データ番号1104として「2」をそれぞれ設定する。その他の項目も同様に設定されるものである。

40

#### 【0069】

50

続いて、テストケース自動生成の処理について説明する。まずは、テストケース自動生成に際し前記システム100が用いるファイル等について例示する。図12は、前記システム100が利用する雛形ファイル127の例を示す図である。前記システム100がテストケース自動生成時に利用する雛形ファイル127は、設計書10と同様に表計算ファイルが想定できる。前記システム100は設計情報データベース125に保持する設計項目のデータと、テスト情報データベース126に予め保持するテストケースの定義とその出力条件をもとに、出力すべきテストケースを作成し、これを雛形ファイル127に適用し出力することでテストケース(のファイル)を自動生成する。1つの表計算ファイルは、図13で説明する環境設定ファイル128で指定された設計項目IDに対する設計項目データ単位に作成し、1つのテストケースは1つのシートに出力する。

10

**【0070】**

図13は、前記システム100が利用する環境設定ファイル128の例を示す図である。前記システム100がテストケース自動生成時に利用する環境設定ファイル128はテキストファイルが想定できる。環境設定ファイル128には、テストケースの出力単位となる設計項目IDを指定する。前記システム100ではこの環境設定ファイル128で指定された設計項目IDに対する設計項目のデータと、その設計項目に紐づく子となるすべての設計項目のデータを元にテストケースを作成することになる。図13の例では、設計項目IDとして「画面ID」が指定されているため、この場合、前記システム100は図5に示す画面ID:SCREEN01にぶら下がるすべての設計項目のデータを元にテストケースを作成する。

20

**【0071】**

続いて、前記システム100が自動生成したテストケース20の例を示しておく。図14に示す例ではテストケース20が、「必須チェック(未入力)」と「入力可能桁数(桁数)」、「入力可能桁数(桁数+1)」の3つのテストケースから構成されている。

**【0072】**

そのうち、「必須チェック(未入力)」は、必須指定された項目に対して未入力の場合に入力エラーが発生するかを確認するテストケースである。また、「入力可能桁数(桁数)」、「入力可能桁数(桁数+1)」は、画面上の入力項目に対して桁数分しか入力できないことを確認するテストケースであり、入力項目への入力文字数のパターンとして桁数分と(桁数+1)分の2つのパターンを用意している。

30

**【0073】**

また、「必須チェック(未入力)」のテストケースは、設計項目のうち「必須」が「」になっている項目に対してのみ実施するテストケースであり、図4の例においては、「ユーザID」のみ「必須」が「」となっているため、「ユーザID」に対する1件のテストケース、すなわちこの図14で示す「必須チェック(未入力)」のテストケースが出力された結果となっている。また、「入力可能桁数(桁数)」と「入力可能桁数(桁数+1)」の各テストケースは、全画面項目に対して行うチェックであり、「ユーザID」と「ユーザ名」の両方に対してチェックしている。また、出力する表計算ファイルのシートはテストケース単位となっている。

**【0074】**

40

続いて、前記システム100がテストケース自動生成時に利用するテスト情報データベース126を構成するテーブルについて説明する。図15は、テストケーステーブル605の例を示す図である。ここで例示するテストケーステーブル605は、テストケースとして「必須チェック(未入力)」、「入力可能桁数(桁数)」、「入力可能桁数(桁数+1)」の3つのテストケースに関する計3件のレコードが登録されている。こうしたレコードは、テスト担当のユーザが事前に設定しており、必要に応じてユーザによる更新作業が可能となっている(システム100が当該テーブルを出力部106に表示し、入力部105でユーザ指示を受ける)。以降、他のテーブルについても同様にシステム100が情報の表示を行って、ユーザによる閲覧、更新、削除等の指示を受け付けてデータ更新等を実行可能とする。従って、図15のテストケーステーブル605であれば、テストケース

50

の追加や削除、テストケース名1502や出力条件ID1502の変更等も、システム100が出力部106にて該当情報を表示してユーザ閲覧に供しつつ、入力部105にて変更等の指示を受けて、該当データの更新等を行うこととなる。

#### 【0075】

このテストケーステーブル605において、テストケースID1501は、テストケース毎に一意となる情報であるため、「必須チェック(未入力)」、「入力可能桁数(桁数)」、「入力可能桁数(桁数+1)」のそれぞれに関し、互いに重複しない値として「TC001」、「TC002」、「TC003」が設定されている。テストケース名1502はテストケースの日本語名称を表すものであるため、「必須チェック(未入力)」、「入力可能桁数(桁数)」、「入力可能桁数(桁数+1)」が設定されている。また、出力条件ID1503はテストケースの出力制御を行うものであり、「必須チェック(未入力)」のテストケースは「必須」が「」の場合にのみ出力するとの条件があるため、出力条件ID1503に例えば該当条件を示す「OP001」が設定されている。他方、「入力可能桁数(桁数)」と「入力可能桁数(桁数+1)」の各テストケースは、すべての画面項目に対して出力するテストケースであるため、出力条件ID1503は未設定となっている。

10

#### 【0076】

また、前記テスト情報データベース126を構成する、テストケース詳細テーブル606は、図16に例示するように、各テストケース(上記例では3つのテストケース)に対してそれぞれ実行手順が記述されたテーブルとなる。ここでは、テストケース：必須チェック(未入力)に関して説明する。テストケース：必須チェック(未入力)は、5つのチェック条件に関する操作と3つの確認に関する操作の計8つの操作からなる。

20

#### 【0077】

操作内容1604には{設計項目ID}が埋め込まれており、システム100が自動生成するテストケースにおいては、前記設計項目IDの記述が、対応する設計項目のデータ(設計書10から抽出したもの)で置換されて出力される。

#### 【0078】

また図16において、操作番号1602が「1」の操作内容1604は「{ScreenId}：{ScreenName}が表示されている。」となっており、設計項目IDとして「ScreenId」と「ScreenName」が埋め込まれているため、図4、図8の場合には、テストケースとしては{ScreenId}が「SCREEN01」に、{ScreenName}が「ユーザ登録画面」に置換されて、「SCREEN01：ユーザ登録画面が表示されている。」が出力されることになる。また、操作番号1602が「2」の操作内容1604は「以下の項目を未入力にする」となっており、設計項目IDは埋め込まれていないため、このまま出力される。また、操作番号1602が「3」の操作内容1604は「ScreenItemId：{ScreenItemName}」となっており、設計項目IDとして「ScreenItemId」、「ScreenItemName」が埋め込まれているため、図4、図8の場合には、画面項目が2つ存在しており、「userId：ユーザID」と「userName：ユーザ名」の2行が出力される。その他の操作番号のレコードについても同様に出力されるものである。

30

40

#### 【0079】

図17は、出力条件テーブル607の例を示す図である。上述までの例において、出力の制御をしたいテストケースとして「必須チェック(未入力)」がある。そこで図17の出力条件テーブル607では、「必須チェック(未入力)」のテストケース用の出力条件として、1件のレコードが登録されている。このレコードが示す条件において、「必須チェック(未入力)」のテストケースは、画面項目に対する「必須」が「」になっている場合にテストケースとして出力する。出力条件ID1701は出力条件を一意に識別するIDであり、ここでは「OP001」が設定されている。また、前記「必須チェック(未入力)」のテストケースに関する出力条件が、画面項目に対する「必須」が「」になっていることであることから、設計項目ID1702には、「必須」に対する設計項目ID：S

50



screenItemNecessaryが、また、出力条件内容1703に「 」が設定されている。

【0080】

続いて、前記システム100が、設計情報データベース125とテスト情報データベース126に基づいてテストケース20を自動生成する処理について、図18, 図19に基づき説明する。まず、前記システム100は、図13に例示した環境設定ファイル128から画面IDに相当する設計項目IDを取得する(ステップ510)。

【0081】

次に前記システム100は、設計情報データベース125の設計項目データテーブル604にアクセスし、設計項目データテーブル604の設計項目ID1102が、前記ステップ510で取得した設計項目IDと一致するレコード、すなわち設計項目データレコードを取得する(ステップ520)。

10

【0082】

また、前記システム100は、前記ステップ520で取得した設計項目データレコードを1つずつ読んでいく。設計項目データレコードがある場合(ステップ530:有)、前記システム100は処理をステップ540に進める。他方、設計項目データレコードが無い場合(ステップ530:無)、前記システム100は処理を終了する。こうした処理は、前記ステップ520で得た設計項目データレコードが無くなるまで繰り返し実行される。

【0083】

20

続いて前記システム100は、前記ステップ520で取得したレコードからデータ番号1101を取得する(ステップ540)。また前記システム100は、設計情報データベース125の設計項目データテーブル604にアクセスし、前記ステップ540で取得したデータ番号1101を頂点として連なる親子関係において、子となる設計項目のデータを、子が無くなるまですべて取得する(ステップ550)。つまり前記システム100は図3と図5に相当する情報を作成する。

【0084】

前記システム100は、テスト情報データベース126のテストケーステーブル605にアクセスし、全てのテストケースのレコードを取得する(ステップ560)。また前記システム100は、ここで取得したテストケースレコードを1つずつ読んでいく。テストケースレコードがある場合(ステップ570:有)、前記システム100は処理をステップ580に進める。他方、テストケースレコードが無い場合(ステップ570:無)、前記システム100は処理を終了する。こうした処理は、前記ステップ560で得たテストケースレコードが無くなるまで繰り返し実行される。

30

【0085】

続いて前記システム100は、前記ステップ570で取得したレコードから出力条件ID1503を取得する(ステップ580)。また前記システム100は、該ステップ580で出力条件ID1503を取得できた場合(ステップ590:有)、処理をステップ600に進める。他方、出力条件ID1503が無かった場合(ステップ590:無)、前記システム100は処理をステップ650に進める。

40

【0086】

前記ステップ590に続き、前記システム100は、テスト情報データベース126の出力条件テーブル607にアクセスし、出力条件テーブル607の出力条件ID1701が、前記ステップ580で取得した出力条件ID1503と一致する出力条件レコードを取得する(ステップ600)。

【0087】

そして前記システム100は、前記ステップ600で取得した出力条件レコードから設計項目ID1702と出力条件内容1703を取得する(ステップ610)。また前記システム100は、ここで取得した設計項目ID1702に対応する設計項目の値1103を、前記ステップ550で取得、作成した情報から取得する(ステップ620)。

50

## 【0088】

次に前記システム100は、前記ステップ610で取得した出力条件内容1703について正規表現を用いて、前記ステップ620で取得した設計項目の値1103とのマッチングを行う(ステップ630)。正規表現は、文字列のパターンを表現する表記法であり、文字列の検索・置換を行うときに利用されるものとなる。正規表現と呼ばれることもある。この正規表現においては、通常の文字と、メタキャラクタと呼ばれる特別な意味を持った記号を組み合わせた表記がなされる。正規表現を用いれば、文字列を直接指定せず、「特徴」(パターン)を指定することができるため、表記の揺れを吸収して検索を行ったり、複数の異なる文字列を一括して置換したりすることができる。例えば「^」という文字は「行頭」、「.」は「任意の1文字」、「+」という文字は「直前の要素の1回以上の繰り返り」を意味するなどの規定がある。従って前記システム100は、出力条件内容1703が示す内容について、こうした正規表現にて記述するために、正規表現の規定について辞書データと所定の変換アルゴリズムを予め記憶部101に備えており、出力条件内容1703を入力として正規表現を生成し、これをベースにして、前記ステップ620で取得した設計項目の値1103とのパターンマッチングを行うものとする。

10

## 【0089】

パターンマッチングの結果、前記出力条件内容1703と、前記ステップ620で取得した設計項目の値1103とがマッチした場合(ステップ640:可)、前記システム100は該当テストケースについて出力可と判定し、他方、マッチしなかった場合(ステップ640:否)には出力不可と判定する。

20

## 【0090】

続いて前記システム100は、前記ステップ640での出力可否の結果をもとに、出力可の場合(ステップ640:可)、処理をステップ650に進める。他方、出力不可の場合(ステップ640:否)、前記システム100は処理をステップ570に戻す。

## 【0091】

前記システム100は、前記ステップ560で取得したテストケースレコードからテストケースID1501とテストケース名1502を取得する(ステップ650)。この場合、前記システム100は、雛形ファイル127からシート名が「template」となっているシートをコピーして、前記取得したテストケース名1502をシート名に持つ新規シートを作成する(ステップ660)。

30

## 【0092】

次に前記システム100は、テスト情報データベース126のテストケース詳細テーブル606にアクセスし、テストケース詳細テーブル606のテストケースID1601が前記ステップ650で取得したテストケースID1501と一致するレコード、すなわちテストケース詳細レコードを取得する(ステップ670)。

## 【0093】

また前記システム100は、前記ステップ670で取得したテストケース詳細レコードを1つずつ読んでいく。テストケース詳細レコードがある場合(ステップ680:有)、システム100は処理をステップ690に進める。他方、テストケース詳細レコードが無い場合(ステップ680:無)、前記システム100は処理をステップ570に戻す。こうした処理は、前記ステップ670で得たテストケース詳細レコードが無くなるまで繰り返し実行される。

40

## 【0094】

続いて前記システム100は、前記ステップ670で取得したテストケース詳細レコードから出力区分1603と操作内容1604を取得する(ステップ690)。前記システム100は、ここで取得した操作内容1604に含まれる設計項目IDを取得する(ステップ700)。この場合、前記システム100は、操作内容1604のうち「{」と「}」に挟まれる値を設計項目IDとして扱う。

## 【0095】

前記システム100は、前記ステップ700で取得した設計項目IDに対応する設計項

50

目の値 1 1 0 3 を、前記ステップ 5 5 0 で取得、作成した情報から取得する（ステップ 7 1 0）。また前記システム 1 0 0 は、前記ステップ 7 1 0 で取得した設計項目の値 1 1 0 3 を、前記ステップ 6 9 0 で得た操作内容 1 6 0 4 に含まれる設計項目 ID と置換する（ステップ 7 2 0）。

【 0 0 9 6 】

前記システム 1 0 0 は、前記ステップ 6 9 0 で取得した、「条件」や「確認」といった出力区分 1 6 0 3 を元に、前記ステップ 7 2 0 で作成した操作内容をシート（前記ステップ 6 6 0 で雛形ファイル 1 2 7 から作成してあるシート。形式は図 1 2 参照）における該当出力区分のセルに出力していくことで、テストケースの生成・出力処理を行う（ステップ 7 3 0）。こうした処理を行うことで、図 1 4 で例示したテストケース 2 0（のうち出力可のもの）が生成されることになる。こうしたテストケース出力後、前記システム 1 0 0 は処理を前記ステップ 6 8 0 に戻す。以降、テストケース詳細レコードが無くなるまで（ステップ 6 8 0：無し ステップ 5 7 0）、テストケースレコードが無くなるまで（ステップ 5 7 0：無 ステップ 5 3 0）、設計項目データレコードが無くなるまで（ステップ 5 3 0：無 END）、前記システム 1 0 0 は処理を繰り返すこととなる。

10

【 0 0 9 7 】

以上、本発明を実施するための最良の形態などについて具体的に説明したが、本発明はこれに限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能である。

【 0 0 9 8 】

こうした本実施形態によれば、フォーマットが異なる各種設計書に対してテストケースを自動生成することが可能であり、テストケース生成作業の省略による作業量縮小を図ることができる。また、自動生成するテストケースに関して、ユーザによる追加、変更、削除も可能であり、不要なテストケースの消化による作業量増大や、必要なテストケースが出力されないことによるテスト漏れによる品質の低下といった事態の発生を防ぐことができる。

20

【 0 0 9 9 】

したがって、設計書におけるフォーマットによらずテストケースを自動生成すると共に、ユーザによるテストケース編集も可能として、システム検証の効率化、品質低下の防止を図ることができる。

【 0 1 0 0 】

本明細書の記載により、少なくとも次のことが明らかにされる。すなわち、前記テストケース自動生成システムにおいて、前記演算部は、記憶部から読み出した前記設計情報データベースのデータと、前記テスト情報データベースにおける各テストケースの出力条件とのパターンマッチングを行うに際し、前記出力条件が示す情報の組み合わせを正規表現とし、これを前記設計情報データベースのデータに対しパターンマッチングさせるとしてもよい。

30

【符号の説明】

【 0 1 0 1 】

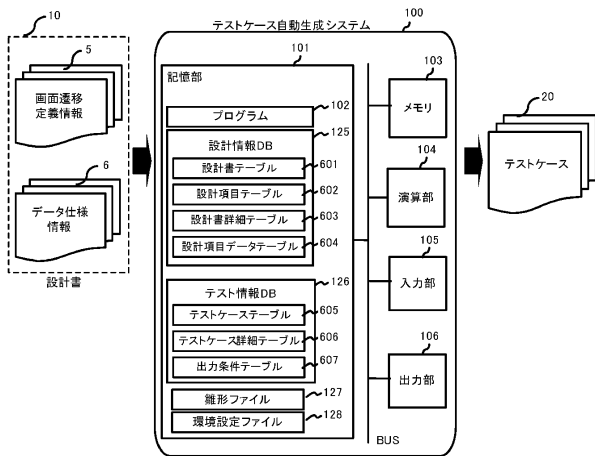
- 5 画面遷移定義情報
- 6 データ仕様情報
- 1 0 設計書
- 2 0 テストケース
- 1 0 0 テストケース自動生成システム
- 1 0 1 記憶部
- 1 0 2 プログラム
- 1 0 3 メモリ
- 1 0 4 演算部
- 1 0 5 入力部
- 1 0 6 出力部
- 1 2 5 設計情報データベース

40

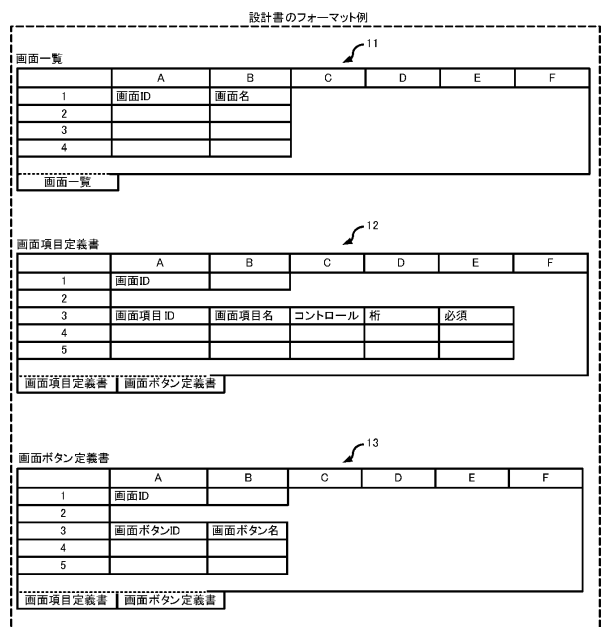
50

- 1 2 6 テスト情報データベース
- 1 2 7 雛形ファイル
- 1 2 8 環境設定ファイル
- 6 0 1 設計書テーブル
- 6 0 2 設計項目テーブル
- 6 0 3 設計書詳細テーブル
- 6 0 4 設計項目データテーブル
- 6 0 5 テストケーステーブル
- 6 0 6 テストケース詳細テーブル
- 6 0 7 出力条件テーブル

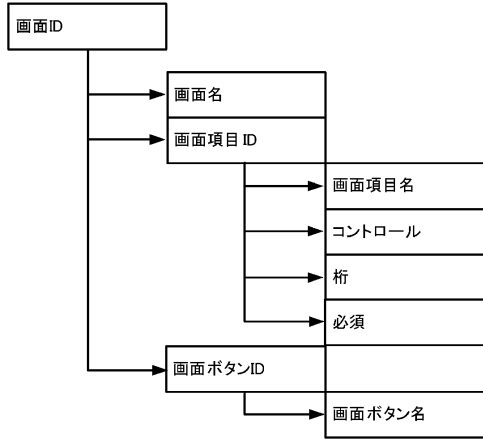
【図1】



【図2】



【図3】



【図4】

画面一覧						
	A	B	C	D	E	F
1	画面ID	画面名				
2	SCREEN01	ユーザ登録画面				
3	SCREEN02	結果画面				
4						
5						
6						

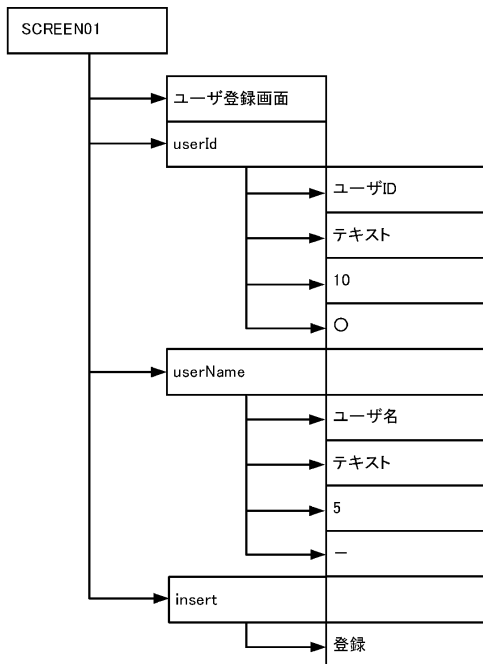
  

画面項目定義書						
	A	B	C	D	E	F
1	画面ID	SCREEN01				
2						
3	画面項目ID	画面項目名	コントロール	桁	必須	
4	userId	ユーザID	テキスト	10	○	
5	userName	ユーザ名	テキスト	5	-	
6						

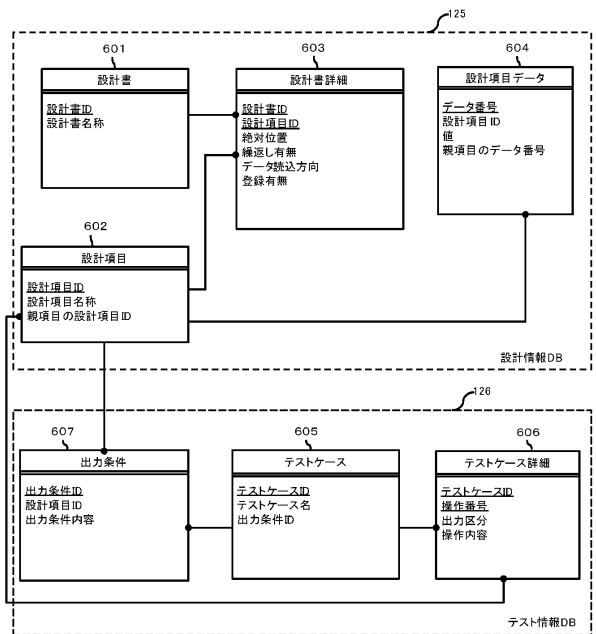
  

画面ボタン定義書						
	A	B	C	D	E	F
1	画面ID	SCREEN01				
2						
3	画面ボタンID	画面ボタン名				
4	insert	登録				
5						
6						

【図5】



【図6】



【図7】

設計書テーブル

設計書ID	設計書名称
ScreenList	画面一覧
ScreenItem	画面項目定義書
ScreenButton	画面ボタン定義書

601

【図9】

設計書詳細テーブル

設計書ID	設計項目ID	絶対位置	繰返し有無	データ読み方向	登録有無
ScreenList	ScreenId	A2	有	↓	有
ScreenList	ScreenName	B2	有	↓	有
ScreenItem	ScreenId	B1	無	—	無
ScreenItem	ScreenItemId	A4	有	↓	有
ScreenItem	ScreenItemName	B4	有	↓	有
ScreenItem	ScreenItemCtl	C4	有	↓	有
ScreenItem	ScreenItemSize	D4	有	↓	有
ScreenItem	ScreenItemNecessary	E4	有	↓	有
ScreenButton	ScreenId	B2	無	—	無
ScreenButton	ScreenButtonId	A4	有	↓	有
ScreenButton	ScreenButtonName	B4	有	↓	有

603

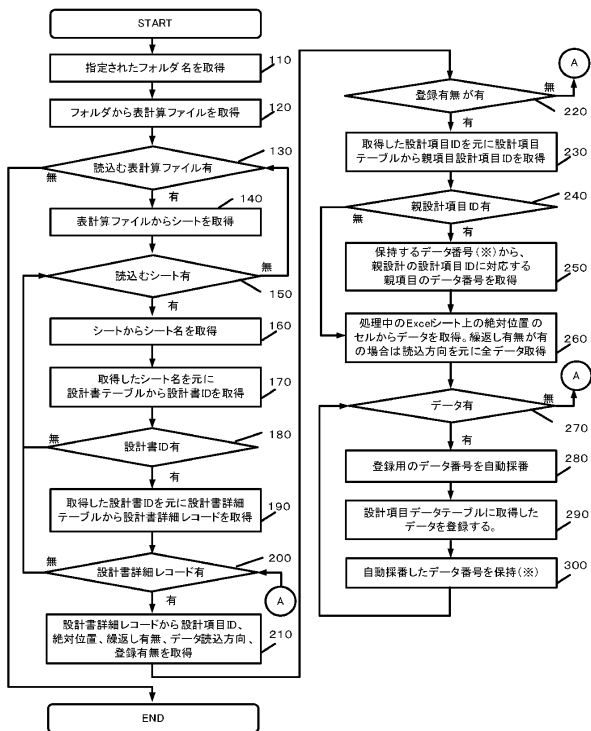
【図8】

設計項目テーブル

設計項目ID	親項目ID	設計項目名称
ScreenId	—	画面ID
ScreenName	ScreenId	画面名
ScreenItemId	ScreenId	画面項目ID
ScreenItemName	ScreenItemId	画面項目名
ScreenItemCtl	ScreenItemId	コントロール
ScreenItemSize	ScreenItemId	桁
ScreenItemNecessary	ScreenItemId	必須
ScreenButtonId	ScreenId	画面ボタンID
ScreenButtonName	ScreenButtonId	画面ボタン名

602

【図10】



【図11】

設計項目データテーブル

データ番号	設計項目ID	値	親データ番号
1	ScreenId	SCREEN01	—
2	ScreenId	SCREEN02	—
3	ScreenName	ユーザ登録画面	1
4	ScreenName	結果画面	2
5	ScreenItemId	userId	1
6	ScreenItemId	userName	1
7	ScreenItemName	ユーザID	5
8	ScreenItemCtl	テキスト	5
9	ScreenItemSize	10	5
10	ScreenItemNecessary	○	5
11	ScreenItemName	ユーザ名	6
12	ScreenItemCtl	テキスト	6
13	ScreenItemSize	5	6
14	ScreenItemNecessary	—	6
15	ScreenButtonId	insert	1
16	ScreenButtonName	登録	15

604

【図12】

雛形ファイル

	A	B	C	D	E
1	チェック条件/確認内容				
2	条件				
3	確認				
template					

127

【図13】

環境設定ファイル

#テストケース生成単位となる設計項目ID DesignItemId=ScreenId
-----------------------------------------------

128

【図14】

	A	B	C	D	E
1	チェック条件/確認内容				
2	条件	SCREEN01: ユーザ登録画面が表示されている。			○
3		以下の項目を未入力にする。			○
4		userId: ユーザID			○
5	確認	userName: ユーザ名			○
6		以下のボタンを押下する。			○
7		Insert: 登録			○
8	確認	SCREEN01: ユーザ登録画面が表示されている。			○
9		以下の項目に対して入力必須エラーが発生する			○
10		userId: ユーザID			○
11		userName: ユーザ名			○
必須チェック(未入力)		入力可能桁数(桁数)	入力可能桁数(桁数+1)		

	A	B	C	D	E	F	
1	チェック条件/確認内容						
2	条件	SCREEN01: ユーザ登録画面が表示されている。				○	○
3		以下の入力を行う。				○	○
4		userId: ユーザID				○	○
5	確認	userName: ユーザ名				○	○
6		SCREEN01: ユーザ登録画面が表示されている。				○	○
7		以下の確認を行う。				○	○
8	確認	userId: ユーザID 10桁かをチェック				○	○
9		userName: ユーザ名 5桁かをチェック				○	○
必須チェック(未入力)		入力可能桁数(桁数)	入力可能桁数(桁数+1)				

	A	B	C	D	E	F	
1	チェック条件/確認内容						
2	条件	SCREEN01: ユーザ登録画面が表示されている。				○	○
3		以下の入力を行う。				○	○
4		userId: ユーザID				○	○
5	確認	userName: ユーザ名				○	○
6		SCREEN01: ユーザ登録画面が表示されている。				○	○
7		以下の確認を行う。				○	○
8	確認	userId: ユーザID 10桁かをチェック				○	○
9		userName: ユーザ名 5桁かをチェック				○	○
必須チェック(未入力)		入力可能桁数(桁数)	入力可能桁数(桁数+1)				

【図15】

テストケーステーブル

テストケースID	テストケース名	出力条件ID
TC001	必須チェック(未入力)	OP001
TC002	入力可能桁数(桁数)	
TC003	入力可能桁数(桁数+1)	

【図16】

テストケース詳細テーブル

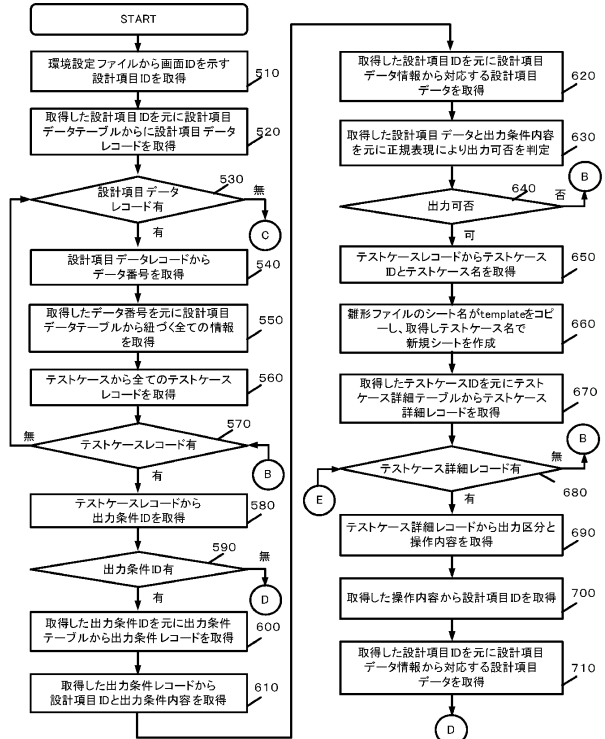
テストケースID	操作番号	出力区分	操作内容
TC001	1	条件	{ScreenId} {ScreenName}が表示されている。
	2	条件	以下の項目を未入力にする。
TC001	3	条件	{ScreenItemId} {ScreenItemName}
	4	条件	以下のボタンを押下する。
TC001	5	条件	{ScreenButtonId} {ScreenButtonName}
	6	確認	{ScreenId} {ScreenName}が表示されている。
TC001	7	確認	以下の項目に対して入力必須エラーが発生する。
	8	確認	{ScreenItemId} {ScreenItemName}
TC002	1	条件	{ScreenId} {ScreenName}が表示されている。
	2	条件	以下の入力を行う。
TC002	3	条件	{ScreenItemId} {ScreenItemName}
	4	確認	{ScreenId} {ScreenName}が表示されている。
TC002	5	確認	以下の確認を行う。
	6	確認	{ScreenItemId} {ScreenItemName} a={ItemSize}桁かをチェック
TC003	1	条件	{ScreenId} {ScreenName}が表示されている。
	2	条件	以下の入力を行う。
TC003	3	条件	{ScreenItemId} {ScreenItemName}
	4	確認	{ScreenId} {ScreenName}が表示されている。
TC003	5	確認	以下の確認を行う。
	6	確認	{ScreenItemId} {ScreenItemName} a={ItemSize}桁かをチェック

【図17】

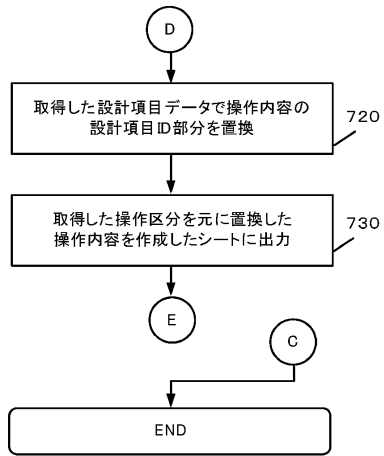
出力条件テーブル

出力条件ID	設計項目ID	出力条件内容
OP001	ScreenItemNecessary	○

【図18】



【図19】





---

フロントページの続き

(56)参考文献 特開2004-265326(JP,A)  
特開平11-282666(JP,A)  
特開2012-037938(JP,A)  
特開2006-309290(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G06F 9/06 - 11/36