(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6723989号
(P6723989)
(45)発行日　令和2年7月15日(2020.7.15)　(24)登録日　令和2年6月26日(2020.6.26)

(51)Int.Cl.　F I
G06F 11/36 (2006.01)　G06F 11/36 184

請求項の数 59 (全 44 頁)

(21)出願番号　特願2017-513040(P2017-513040)
(86)(22)出願日　平成27年9月4日(2015.9.4)
(65)公表番号　特表2017-531855(P2017-531855A)
(43)公表日　平成29年10月26日(2017.10.26)
(86)国際出願番号　PCT/US2015/048528
(87)国際公開番号　WO2016/040154
(87)国際公開日　平成28年3月17日(2016.3.17)
審査請求日　平成30年7月20日(2018.7.20)
(31)優先権主張番号　62/047,256
(32)優先日　平成26年9月8日(2014.9.8)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　509123208
アビニシオ　テクノロジー　エルエルシー
アメリカ合衆国　０２４２１　マサチューセッツ州　レキシントン　スプリング　ストリート　２０１
(74)代理人　100107984
弁理士　廣田　雅紀
(74)代理人　100102255
弁理士　小澤　誠次
(74)代理人　100096482
弁理士　東海　裕作
(74)代理人　100188352
弁理士　松田　一弘
(74)代理人　100131093
弁理士　堀内　真

最終頁に続く

(54)【発明の名称】データ駆動型検査用フレームワーク

(57)【特許請求の範囲】
【請求項１】
アプリケーションを検査するための装置であって、メモリ、及び前記メモリに動作可能に接続されるプロセッサを有するデータ処理機械を含み、前記データ処理機械が、データエンジニアリングモジュールと、計算環境マネージャと、結果分析モジュールとを含むデータ駆動型検査用フレームワークを実装するように構成され、
前記データエンジニアリングモジュールが、検査を受けるアプリケーションに少なくとも部分的に基づいて工学設計された検査データを作り出すように構成され、
検査を受ける前記アプリケーションの論理を通して、少なくとも１つの導出変数の導出過程を追跡するため、データ系列を使用するステップ、
前記導出変数が導出される少なくとも１つの入力変数を識別するステップ、及び
前記工学設計された検査データ中に、前記導出変数についての第２の値に対応する前記入力変数についての第１の値を含め、少なくとも前記入力変数及び前記導出変数を検査するために選択された実際のプロダクションデータに、前記実際のプロダクションデータとは異なる、補強した新しく製造されたデータを追加するステップであって、前記補強した新しく製造されたデータが、前記実際のプロダクションデータに対応する第１のデータセット中のポインタが第２のデータセット中の任意のレコードを指さないという判定に応答して前記工学設計された検査データの第２のデータセットに追加される、少なくとも１つの修復レコードを含み、前記追加された少なくとも１つの修復レコードが前記第１のデータセット中のポインタによって指され、前記補強した新しく製造されたデータが、検査

を受ける前記アプリケーションのコードカバレッジの拡張をさらにもたらし、前記補強した新しく製造されたデータが、（ｉ）少なくとも１つの既存のプロダクションデータレコードの１若しくは２以上の新しいフィールドに追加される生成されるデータ、（ii）生成される１若しくは２以上の新しいデータレコード、及び／又は（iii）１若しくは２以上の実際のプロダクションデータレコードの既存のフィールド中の変更したデータ、のうちの１又は２以上を含む、ステップ
を含み、
前記計算環境マネージャが、前記補強した新しく製造されたデータを含む前記工学設計された検査データを前記アプリケーションが操作するべき計算環境を制御するように構成され、前記入力変数及び前記導出変数が前記計算環境内でアクセスされ、
前記結果分析モジュールが、前記アプリケーションによる操作を受けている、前記補強した新しく製造されたデータを含む前記工学設計された検査データから得られる出力を予想される出力と比較するように構成される、前記装置。
【請求項２】
前記データエンジニアリングモジュールが、前記実際のプロダクションデータのサブセットを抽出するように構成され、
前記サブセットが、指定されるコードカバレッジを達成するように選択され、
前記工学設計された検査データが、前記プロダクションデータの前記サブセットを含む、
請求項１に記載の装置。
【請求項３】
前記データエンジニアリングモジュールが、前記プロダクションデータから精製されたデータを生成するためのデータ精製器を含む、請求項１に記載の装置。
【請求項４】
前記データエンジニアリングモジュールが、既存データのサブセットを抽出するように構成され、前記データエンジニアリングモジュールが、前記サブセットを補強し、それによって、前記補強した新しく製造されたデータを生成するようにさらに構成され、前記工学設計された検査データが、前記補強した新しく製造されたデータを含む、請求項１に記載の装置。
【請求項５】
前記補強した新しく製造されたデータが、前記サブセットの１又は２以上のレコードに追加される１又は２以上のフィールドを含む、請求項４に記載の装置。
【請求項６】
前記補強した新しく製造されたデータが、指定されたコードカバレッジを達成するように選択される、請求項４に記載の装置。
【請求項７】
前記データエンジニアリングモジュールが、データ精製器からの精製されたデータを受け取り前記精製されたデータを拡張するためのデータ拡張器を含む、請求項１に記載の装置。
【請求項８】
前記データエンジニアリングモジュールが、検査を受ける前記アプリケーションに少なくとも部分的に基づいてデータを生成するように構成され、前記生成されるデータが、指定されるコードカバレッジを達成するように選択され、前記工学設計された検査データが、前記生成されるデータを含む、請求項１に記載の装置。
【請求項９】
前記データエンジニアリングモジュールが肯定データを生成するための肯定データ製造器をさらに含む、請求項１に記載の装置。
【請求項１０】
前記データエンジニアリングモジュールが検査を受ける前記アプリケーションに少なくとも部分的に基づいてデータを生成するように構成され、前記データがプロダクションデ

ータにない、請求項１に記載の装置。
【請求項１１】
前記データエンジニアリングモジュールが否定データを生成するための否定データ製造器をさらに含む、請求項１に記載の装置。
【請求項１２】
前記データエンジニアリングモジュールが、前記工学設計された検査データを生成するための手段を含む、請求項１に記載の装置。
【請求項１３】
前記データエンジニアリングモジュールが、前記工学設計された検査データの参照整合性を決定するための整合性チェッカを含む、請求項１に記載の装置。
【請求項１４】
前記データエンジニアリングモジュールが、参照整合性中のエラーを検出するようにさらに構成される、請求項１に記載の装置。
【請求項１５】
前記データエンジニアリングモジュールが、データ中の参照整合性の損失を、前記工学設計された検査データとして前記データを出力する前に訂正するための再参照器を含む、請求項１に記載の装置。
【請求項１６】
前記データエンジニアリングモジュールがデータ中の参照整合性の損失を訂正するようにさらに構成される、請求項１に記載の装置。
【請求項１７】
前記データエンジニアリングモジュールが、前記工学設計された検査データを受け取り、前記工学設計された検査データを閲覧すること及び前記工学設計された検査データのプロファイリングを行うことのうちの少なくとも一方をユーザに可能にする点検ユニットを含む、請求項１に記載の装置。
【請求項１８】
前記データエンジニアリングモジュールが、前記工学設計された検査データを受け取り、前記工学設計された検査データを閲覧することをユーザに可能にするデータ点検ユニットを含む、請求項１に記載の装置。
【請求項１９】
前記データエンジニアリングモジュールが、前記工学設計された検査データを受け取り、前記工学設計された検査データのプロファイリングを行うことをユーザに可能にするプロファイラを含む、請求項１に記載の装置。
【請求項２０】
前記データエンジニアリングモジュールが、前記工学設計された検査データのプロファイリングを行うことをユーザに可能にさせるようにさらに構成される、請求項１に記載の装置。
【請求項２１】
前記データエンジニアリングモジュールが、前記工学設計された検査データを閲覧することをユーザに可能にさせるようにさらに構成される、請求項１に記載の装置。
【請求項２２】
前記データエンジニアリングモジュールが、前記工学設計された検査データを生成するための複数の手段を含み、前記工学設計された検査データを生成するための特定の手段が、検査を受ける前記アプリケーションに関する情報に少なくとも部分的に基づいて生成される、請求項１に記載の装置。
【請求項２３】
前記データエンジニアリングモジュールが、データ拡張器、データ精製器、否定データ製造器、及び肯定データ製造器を含み、それらの各々が、前記工学設計された検査データにとっての基礎をなすデータを提供するように構成される、請求項１に記載の装置。
【請求項２４】

前記データエンジニアリングモジュールが、検査を受ける前記アプリケーション内の検査を受ける論理機能を識別するように構成され、それらの論理機能をデータ精製器に提供する論理抽出器を含む、請求項１に記載の装置。

【請求項２５】

前記データエンジニアリングモジュールが検査を受ける前記アプリケーション内の論理機能を識別し、前記実際のプロダクションデータのサブセットを獲得するための基礎として使用されることになるそれらの論理機能を提供するようにさらに構成される、請求項１に記載の装置。

【請求項２６】

前記計算環境マネージャが、前記アプリケーションの検査が行われる計算環境を自動的にセットアップし削除するための手段を含む、請求項１に記載の装置。

【請求項２７】

前記計算環境マネージャが環境移行機械を含み、前記環境移行機械が、前記工学設計された検査データのソースを識別するように構成され、前記環境移行機械が、検査を受ける前記アプリケーションによる前記工学設計された検査データの処理で生じるデータを入れるターゲットを識別するようにさらに構成される、請求項１に記載の装置。

【請求項２８】

前記環境移行機械が、第１のリポジトリからソースに前記工学設計された検査データをコピーするようにさらに構成される、請求項２７に記載の装置。

【請求項２９】

前記環境移行機械がターゲットから第２のリポジトリに前記工学設計された検査データをコピーするようにさらに構成される、請求項２８に記載の装置。

【請求項３０】

前記計算環境マネージャが、環境バックアップ機械及び復元機械を含み、前記環境バックアップ機械が、第２の環境へと第１の環境を変換する前に、前記第１の環境をバックアップするために構成され、前記復元機械が、前記第２の環境を前記第１の環境で置き換えるために構成され、前記第２の環境が、検査を受ける前記アプリケーションの検査が行われることになる環境である、請求項１に記載の装置。

【請求項３１】

前記計算環境マネージャが実行器を含み、前記実行器が検査を受ける前記アプリケーションの実行を行わせるように構成される、請求項１に記載の装置。

【請求項３２】

アプリケーションの実行を行わせるときに、自動的にスクリプトを実行するように前記実行器が構成される、請求項３１に記載の装置。

【請求項３３】

前記計算環境マネージャが、環境移行機械、環境バックアップ機械、復元機械、及び実行器を含み、前記環境移行機械が、前記工学設計された検査データのソースを識別するように構成され、前記環境移行機械が、検査を受ける前記アプリケーションによる前記工学設計された検査データの処理で生じるデータを入れるターゲットを識別するようにさらに構成され、前記環境バックアップ機械が、第２の環境へと第１の環境を変換する前に、前記第１の環境をバックアップするために構成され、前記復元機械が、前記第２の環境を前記第１の環境で置き換えるために構成され、前記第２の環境が、検査を受ける前記アプリケーションの検査が行われることになる環境であり、前記実行器が検査を受ける前記アプリケーションの実行を行わせるように構成される、請求項１に記載の装置。

【請求項３４】

コンピューティングシステム中でデータを処理するための方法であって、

アプリケーションを検査するステップを含み、アプリケーションを検査するステップが、検査を受けるアプリケーションを表す情報を、入力デバイス及びデータ処理システムのポートのうちの１つを介して受け取るステップ、及び前記受け取った情報を処理するステップを含み、前記受け取った情報を処理するステップが、

前記情報に少なくとも部分的に基づいて工学設計された検査データを作り出すステップであって、

検査を受ける前記アプリケーションの論理を通して、少なくとも１つの導出変数の導出過程をたどるため、データ系列を使用するステップ、

前記導出変数が導出される少なくとも１つの入力変数を識別するステップ、及び

前記工学設計された検査データ中に、前記導出変数についての第２の値に対応する前記入力変数についての第１の値を含め、少なくとも前記入力変数及び前記導出変数を検査するために選択された実際のプロダクションデータに、前記実際のプロダクションデータとは異なる、補強した新しく製造されたデータを追加するステップであって、前記補強した新しく製造されたデータが、前記実際のプロダクションデータに対応する第１のデータセット中のポインタが第２のデータセット中の任意のレコードを指さないという判定に応答して前記工学設計された検査データの第２のデータセットに追加される、少なくとも１つの修復レコードを含み、前記追加された少なくとも１つの修復レコードが前記第１のデータセット中のポインタによって指され、前記補強した新しく製造されたデータが、検査を受ける前記アプリケーションのコードカバレッジの拡張をさらにもたらし、前記補強した新しく製造されたデータが、（ｉ）少なくとも１つの既存のプロダクションデータレコードの１若しくは２以上の新しいフィールドに追加される生成されるデータ、（ii）生成される１若しくは２以上の新しいデータレコード、及び／又は（iii）１若しくは２以上の実際のプロダクションデータレコードの既存のフィールド中の変更したデータ、のうちの１又は２以上を含む、ステップ

を含む、前記作り出すステップと、

前記補強した新しく製造されたデータを含む前記工学設計された検査データを前記アプリケーションが操作するべき計算環境を制御するステップであって、前記入力変数及び前記導出変数が前記計算環境内でアクセスされる、前記制御するステップと、

前記アプリケーションによる操作を受けている、前記補強した新しく製造されたデータを含む前記工学設計された検査データから得られる出力を予想される出力と比較するステップと、

前記比較を示す結果を出力するステップと

を含む、前記方法。

【請求項３５】

アプリケーションの検査を管理するため、コンピュータ可読媒体上に非一時的な形態で記憶されるソフトウェアであって、

検査を受けるアプリケーションに少なくとも部分的に基づいて工学設計された検査データを作り出すステップであって、

検査を受ける前記アプリケーションの論理を通して、少なくとも１つの導出変数の導出過程をたどるため、データ系列を使用するステップ、

前記導出変数が導出される少なくとも１つの入力変数を識別するステップ、及び

前記工学設計された検査データ中に、前記導出変数についての第２の値に対応する前記入力変数についての第１の値を含め、少なくとも前記入力変数及び前記導出変数を検査するために選択された実際のプロダクションデータに、前記実際のプロダクションデータとは異なる、補強した新しく製造されたデータを追加するステップであって、前記補強した新しく製造されたデータが、前記実際のプロダクションデータに対応する第１のデータセット中のポインタが第２のデータセット中の任意のレコードを指さないという判定に応答して前記工学設計された検査データの第２のデータセットに追加される、少なくとも１つの修復レコードを含み、前記追加された少なくとも１つの修復レコードが前記第１のデータセット中のポインタによって指され、前記補強した新しく製造されたデータが、検査を受ける前記アプリケーションのコードカバレッジの拡張をさらにもたらし、前記補強した新しく製造されたデータが、（ｉ）少なくとも１つの既存のプロダクションデータレコードの１若しくは２以上の新しいフィールドに追加される生成されるデータ、（ii）生成される１若しくは２以上の新しいデータレコード、及び／又は（iii）１若しくは２以上

の実際のプロダクションデータレコードの既存の
フィールド中の変更したデータ、のうちの１又は２以上を含む、ステップ
を含む、前記作り出すステップと、
前記補強した新しく製造されたデータを含む前記工学設計された検査データを前記アプリケーションが操作するべき計算環境を制御するステップであって、前記入力変数及び前記導出変数が前記計算環境内でアクセスされる、前記制御するステップと、
前記アプリケーションによる操作を受けている、前記補強した新しく製造されたデータを含む前記工学設計された検査データから得られる出力を予想される出力と比較するステップと、
前記比較の分析を出力するステップと
を含む処理ステップをコンピューティングシステムに実行させるための命令を含む、前記ソフトウェア。
【請求項３６】
前記データエンジニアリングモジュールが、１又は２以上の供給されるルールに基づいて追加された１又は２以上のフィールドを満たすためのデータを生成するようにさらに構成される、請求項５に記載の装置。
【請求項３７】
前記データエンジニアリングモジュールが、既存のデータの精製によって前記工学設計された検査データを作り出すように構成され、前記工学設計された検査データが、前記既存データよりも高い論理濃度を有する、請求項１に記載の装置。
【請求項３８】
前記工学設計された検査データを作り出すステップが、前記実際のプロダクションデータのサブセットを抽出するステップを含み、
前記サブセットが、指定されるコードカバレッジを達成するように選択され、
前記工学設計された検査データが、前記プロダクションデータの前記サブセットを含む、請求項３４に記載の方法。
【請求項３９】
前記工学設計された検査データを作り出すステップが、前記工学設計された検査データの異なるテーブル間で参照整合性を決定するステップを含み、前記工学設計された検査データが複数のテーブルを含み、各テーブルが複数のレコードを含み、各レコードが１又は２以上のフィールドを含む、請求項３４に記載の方法。
【請求項４０】
前記工学設計された検査データを作り出すステップが、検査を受ける前記アプリケーション内の検査を受ける論理機能を識別するステップと、前記実際のプロダクションデータのサブセットを獲得するための基礎として使用されることになるそれらの論理機能を提供するステップとを含む、請求項３４に記載の方法。
【請求項４１】
前記工学設計された検査データを作り出すステップが、検査を受ける前記アプリケーションの論理機能を識別するために、既存のデータのサブセットを抽出するステップと、既存のデータの前記サブセットを補強し、それによって前記補強した新しく製造されたデータを生成するステップとを含み、前記工学設計された検査データが前記補強した新しく製造されたデータを含み、前記補強した新しく製造されたデータが、検査を受ける前記アプリケーションの前記識別された論理機能に基づいて、既存のデータの前記サブセットの１又は２以上のレコードに追加される１又は２以上のフィールドを含む、請求項３４に記載の方法。
【請求項４２】
前記工学設計された検査データを作り出すステップが、１又は２以上の供給されるルールに基づいて追加された１又は２以上のフィールドを満たすためのデータを生成するステップをさらに含む、請求項４１に記載の方法。
【請求項４３】

前記補強した新しく製造されたデータが指定されたコードカバレッジを達成するように選択される、請求項４１に記載の方法。

【請求項４４】

前記工学設計された検査データを作り出すステップが、既存のデータの精製によって前記工学設計された検査データを作り出すステップを含み、前記工学設計された検査データが、前記既存のデータよりも高い論理濃度を有する、請求項３４に記載の方法。

【請求項４５】

前記工学設計された検査データを作り出すステップが、すべてがデータエンジニアリングモジュールの部分として実装される、データ拡張器、精製データ精製器、否定データ製造器、及び肯定データ製造器から、前記工学設計された検査データのための基礎をなすデータを提供するステップを含む、請求項３４に記載の方法。

【請求項４６】

前記工学設計された検査データを作り出すステップが、検査を受ける前記アプリケーション内の論理機能を識別するステップと、それらの論理機能を精製データ精製器に提供するステップとを含む、請求項３４に記載の方法。

【請求項４７】

前記計算環境を制御するステップが、前記工学設計された検査データのソースを識別するステップと、検査を受ける前記アプリケーションによる前記工学設計された検査データの処理で生じるデータを入れるターゲットを識別するステップとを含む、請求項３４に記載の方法。

【請求項４８】

前記計算環境を制御するステップが、第１の環境を第２の環境へと変換する前に、前記第１の環境をバックアップするステップと、前記第２の環境を前記第１の環境で置き換えるステップとを含み、前記第２の環境が、検査を受ける前記アプリケーションの検査が行われることになる環境である、請求項３４に記載の方法。

【請求項４９】

前記工学設計された検査データを作り出すステップが、前記実際のプロダクションデータのサブセットを抽出するステップを含み、

前記サブセットが、指定されるコードカバレッジを達成するように選択され、

前記工学設計された検査データが、前記プロダクションデータの前記サブセットを含む、請求項３５に記載のソフトウェア。

【請求項５０】

前記工学設計された検査データを作り出すステップが、前記工学設計された検査データの異なるテーブル間で参照整合性を決定するステップを含み、前記工学設計された検査データが複数のテーブルを含み、各テーブルが複数のレコードを含み、各レコードが１又は２以上のフィールドを含む、請求項３５に記載のソフトウェア。

【請求項５１】

前記工学設計された検査データを作り出すステップが、検査を受ける前記アプリケーション内の検査を受ける論理機能を識別するステップと、プロダクションデータのサブセットを獲得するための基礎として使用されることになるそれらの論理機能を提供するステップとを含む、請求項３５に記載のソフトウェア。

【請求項５２】

前記工学設計された検査データを作り出すステップが、検査を受ける前記アプリケーションの論理機能を識別するために、既存のデータのサブセットを抽出するステップと、既存のデータの前記サブセットを補強し、それによって補強した新しく製造されたデータを生成するステップとを含み、前記工学設計された検査データが前記補強した新しく製造されたデータを含み、前記補強した新しく製造されたデータが、検査を受ける前記アプリケーションの前記識別された論理機能に基づいて、既存のデータの前記サブセットの１又は２以上のレコードに追加される１又は２以上のフィールドを含む、請求項３５に記載のソフトウェア。

【請求項５３】
前記工学設計された検査データを作り出すステップが、１又は２以上の供給されるルールに基づいて追加された１又は２以上のフィールドを満たすためのデータを生成するステップをさらに含む、請求項５２に記載のソフトウェア。
【請求項５４】
前記補強した新しく製造されたデータが指定されたコードカバレッジを達成するように選択される、請求項５２に記載のソフトウェア。
【請求項５５】
前記工学設計された検査データを作り出すステップが、既存のデータの精製によって工学設計された検査データを作り出すステップを含み、前記工学設計された検査データが、前記既存データよりも高い論理濃度を有する、請求項３５に記載のソフトウェア。
【請求項５６】
前記工学設計された検査データを作り出すステップが、すべてがデータエンジニアリンググモジュールの部分として実装される、データ拡張器、精製データ精製器、否定データ製造器、及び肯定データ製造器から、前記工学設計された検査データのための基礎をなすデータを提供するステップを含む、請求項３５に記載のソフトウェア。
【請求項５７】
前記工学設計された検査データを作り出すステップが、検査を受ける前記アプリケーション内の論理機能を識別するステップと、それらの論理機能を精製データ精製器に提供するステップとを含む、請求項３５に記載のソフトウェア。
【請求項５８】
前記計算環境を制御するステップが、前記工学設計された検査データのソースを識別するステップと、検査を受ける前記アプリケーションによる前記工学設計された検査データの処理で生じるデータを入れるターゲットを識別するステップとを含む、請求項３５に記載のソフトウェア。
【請求項５９】
前記計算環境を制御するステップが、第１の環境を第２の環境へと変換する前に、前記第１の環境をバックアップするステップと、前記第２の環境を前記第１の環境で置き換えるステップとを含み、前記第２の環境が、検査を受ける前記アプリケーションの検査が行われることになる環境である、請求項３５に記載のソフトウェア。

【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照
本出願は、米国出願第６２／０４７，２５６号明細書の優先日２０１４年９月８日の利益を主張するものである。
【０００２】
本開示は、品質管理に関し、詳細には、ソフトウェアアプリケーション中の欠陥又は欠点を識別するために使用されるデバイス及び方法に関する。
【背景技術】
【０００３】
データ処理機械は、それを汎用コンピュータから特定のタスクを実行する専用機械へと変換する再設定を必要とする。結果として得られる再設定は、汎用コンピュータが以前は実行できなかった物事を実行する能力を汎用コンピュータにもたらすことで、汎用コンピュータをこうして改善する。この再設定は、典型的には、汎用コンピュータにある種の特殊なソフトウェアを実行させることによって実行される。この特殊なソフトウェアは、「アプリケーション」又は「アプリ」と呼ばれることが多い。
【０００４】

大きいプロジェクトでは、検査を受けるアプリケーションは、エンジニアのチームによって設計されて実装される。このアプリケーションは、次いで品質保証チームに提供される。品質保証チームは、典型的には、設計チームとは別個である。品質保証チームは、このアプリケーション中の欠陥又は欠点の調査を進める。

【発明の概要】

【発明が解決しようとする課題】

【０００５】

アプリケーションを検査するための手順は、非常に難しい可能性がある。この難しさは、多くの理由で生じる。そのような理由の１つは、品質保証チームが、本質的に否定的見解、すなわち、検査されているソフトウェア中に欠陥又は欠点が存在しないことを証明しようと試みているのであるということである。一般的に、あらゆる可能な事例をカバーするために大量の検査を行うのは、コスト効果的でない。したがって、検査データを賢明に選択することが必要である。

【０００６】

アプリケーションを検査するための手順における別の難しさは、検査が行われる環境が違いを起こす可能性があることである。環境とは、一般的に、実行中であるソフトウェア、及びアプリケーションが操作することが意図されるデータの両方を含む。検査されているアプリケーションと他のソフトウェアの間の相互作用の場合、どの他のソフトウェアが実行中であるのかを知ることが重要である。正しいデータを存在させることが重要である。というのは、検査されているアプリケーションの特徴は、アプリケーションに提供されるデータに非常に依存するためである。例えば、アプリケーションは、データベースからのある種のデータを要求する場合がある。そのような場合では、アプリケーションの検査は、データベースが正しいデータを有するかどうかを知っていることが必要である。したがって、品質保証チームは、一般的に、環境を制御するステップを行う。

【０００７】

アプリケーションの検査において生じるさらに別の難しさは、結果の完全性を確立することである。いくつかの場合では、特定の環境中で処理される入力データの所与の入力の組について、どんな結果が「正しい」又は「誤り」と考えるべきかを知ることが困難な場合がある。

【０００８】

検査は、ソフトウェア開発のライフサイクルの主要な部分であるので、検査をより効率的に実行する方法を提供するのは有用である。

【課題を解決するための手段】

【０００９】

一態様では、本発明は、アプリケーションを検査するための装置を特徴とする。そのような装置は、メモリとメモリに動作可能に接続されるプロセッサとを有するデータ処理機械を含む。データ処理機械は、データエンジニアリングモジュールと、計算環境マネージャと、結果分析モジュールとを含むデータ駆動型検査用フレームワークを実装するように構成される。データエンジニアリングモジュールは、検査を受けるアプリケーションに少なくとも部分的に基づいて工学設計された検査データを作り出すように構成される。一方、計算環境マネージャは、工学設計された検査データをアプリケーションが操作するべき計算環境を制御するように構成される。最後に、結果分析モジュールは、アプリケーションによる操作を受けた工学設計された検査データを予想される出力と比較するように構成される。

【００１０】

いくつかの実施形態では、データエンジニアリングモジュールは、プロダクションデータのサブセットを抽出するように構成される。このサブセットは、指定されるコードカバレッジを達成するように選択される。工学設計された検査データは、次いで、プロダクションデータのこのサブセットを含むことになる。

【００１１】

他の実施形態では、データエンジニアリングモジュールは、プロダクションデータから精製されたデータを生成するためのデータ精製器を含む。

【００１２】

本発明の範囲内にやはり含まれるのは、データエンジニアリングモジュールが、プロダクションデータのサブセットを抽出し、サブセットを追加データで補強し、それによって補強データを生成するように構成される実施形態である。追加データは、指定されたコードカバレッジを達成するように選択され、工学設計された検査データは補強データを含む。

【００１３】

いくつかの実施形態では、データエンジニアリングモジュールは、データ精製器と、データ精製器からの精製されたデータを受け取り精製されたデータを拡張するためのデータ拡張器とを含む。

【００１４】

さらなる実施形態としては、データエンジニアリングモジュールが、検査を受けるアプリケーションに少なくとも部分的に基づいてデータを生成するように構成される実施形態が挙げられる。生成されるデータは、指定されるコードカバレッジを達成するように選択され、工学設計される検査データは、生成されるデータを含む。

【００１５】

他の実施形態としては、データエンジニアリングモジュールが肯定データを生成するための肯定データ製造器をさらに含む実施形態、データエンジニアリングモジュールが検査を受けるアプリケーションに少なくとも部分的に基づいてデータを生成するように構成され、そのデータがプロダクションデータにない実施形態、及びデータエンジニアリングモジュールが否定データを生成するための否定データ製造器をさらに含む実施形態が挙げられる。

【００１６】

いくつかの実施形態では、データエンジニアリングモジュールは、工学設計された検査データを生成するための手段を含む。

【００１７】

さらなる実施形態としては、データエンジニアリングモジュールが工学設計された検査データの参照整合性を決定するための整合性チェッカを含む実施形態、並びにデータエンジニアリングモジュールが参照整合性中のエラーを検出するようにさらに構成される実施形態が挙げられる。

【００１８】

データ中の参照整合性の損失を、工学設計された検査データとしてデータを出力する前に訂正するための再参照器をデータエンジニアリングモジュールが含む実施形態、及びデータエンジニアリングモジュールがデータ中の参照整合性の損失を訂正するようにさらに構成される実施形態がやはり挙げられる。

【００１９】

さらなる実施形態としては、データエンジニアリングモジュールが、工学設計された検査データを受け取り、工学設計された検査データを閲覧すること又は工学設計された検査データのプロファイリングを行うことのいずれかをユーザに可能にする点検ユニットを含む実施形態、データエンジニアリングモジュールが、工学設計された検査データを受け取り、工学設計された検査データを閲覧することをユーザに可能にするデータ点検ユニットを含む実施形態、データエンジニアリングモジュールが、工学設計された検査データを受け取り、工学設計された検査データのプロファイリングを行うことをユーザに可能にするプロファイラを含む実施形態、データエンジニアリングモジュールが、工学設計された検査データのプロファイリングを行うことをユーザに可能にさせるようにさらに構成される実施形態、及びデータエンジニアリングモジュールが、工学設計された検査データを閲覧することをユーザに可能にさせるようにさらに構成される実施形態が挙げられる。

【００２０】

いくつかの実施形態では、データエンジニアリングモジュールは、工学設計された検査データを生成するためのいくつかの方法を含む。これらの実施形態では、どのようにして工学設計された検査データを生成するかの選択は、検査を受けるアプリケーションに関する情報に少なくとも部分的に依存する。他の実施形態では、データエンジニアリングモジュールは、データ拡張器、データ精製器、否定データ製造器、及び肯定データ製造器を含み、それらの各々が、工学設計された検査データにとっての基礎をなすデータを提供するように構成される。

【００２１】

検査を受けるアプリケーション内のそれらの論理機能を識別するように構成され、それらの論理機能をデータ精製器に提供する論理抽出器をデータエンジニアリングモジュールが含む実施形態、及びデータエンジニアリングモジュールが検査を受けるアプリケーション内のそれらの論理機能を識別するようにさらに構成されて、プロダクションデータのサブセットを獲得するための基礎として使用されることになるそれらの論理機能を提供する実施形態がやはり挙げられる。

【００２２】

さらなる実施形態では、計算環境マネージャが、アプリケーションの検査が行われることになる計算環境を自動的にセットアップし削除するための手段を含む。

【００２３】

計算環境マネージャが環境移行機械を含む実施形態も、本発明の実施形態の中にある。環境移行機械は、工学設計された検査データのソースを識別するように構成され、検査を受けるアプリケーションによる工学設計された検査データの処理で生じるデータを入れるターゲットを識別するようにさらに構成される。

【００２４】

いくつかの実施形態では、環境移行機械は、第１のリポジトリからソースに工学設計された検査データをコピーするようにさらに構成される。環境移行機械がターゲットから第２のリポジトリに工学設計された検査データをコピーするようにさらに構成される実施形態が、これらの実施形態の中にある。

【００２５】

本発明の実施形態としては、計算環境マネージャが、環境バックアップ機械並びに復元機械を含む実施形態が挙げられる。そのような実施形態では、環境バックアップ機械は、検査を受けるアプリケーションの検査が行われることになる第２の環境へと第１の環境を変換する前に、第１の環境をバックアップするために構成される。復元機械は、第２の環境を第１の環境で置き換えるために構成される。

【００２６】

いくつかの実施形態では、計算環境マネージャは、検査を受けるアプリケーションの実行を行わせるように構成される実行器を含む。アプリケーションの実行を行わせるときに、自動的にスクリプトを実行するように実行器が構成される実施形態が、これらの実施形態の中にある。

【００２７】

さらに他の実施形態は、環境移行機械、環境バックアップ機械、復元機械、及び実行器を有する計算環境マネージャを含む。これらの実施形態では、環境移行機械は、工学設計された検査データのソースを識別するように構成され、環境移行機械は、検査を受けるアプリケーションによる工学設計された検査データの処理で生じるデータを入れるターゲットを識別するようにさらに構成され、環境バックアップ機械は、検査を受けるアプリケーションの検査が行われることになる第２の環境へと第１の環境を変換する前に、第１の環境をバックアップするために構成される。復元機械は、第２の環境を第１の環境で置き換えるために構成される。また、実行器は検査を受けるアプリケーションの実行を行わせるように構成される。

【００２８】

別の態様では、本発明は、コンピューティングシステム中でデータを処理するための方

法を特徴とする。そのような方法は、アプリケーションを検査するステップを含む。この場合の、アプリケーションを検査するステップは、検査を受けるアプリケーションを表す情報を、入力デバイス又はデータ処理システムのポートを介して受け取るステップと、受け取った情報を処理するステップとを含む。この受け取った情報を処理するステップは、この情報に少なくとも部分的に基づいて工学設計された検査データを作り出すステップと、工学設計された検査データをアプリケーションが操作するべき計算環境を制御するステップと、アプリケーションによる操作を受けた工学設計された検査データを予想される出力と比較するステップと、比較を示す結果を出力するステップとを含む。

【００２９】

別の態様では、本発明は、アプリケーションを検査するためのコンピューティングシステムを特徴とする。そのようなコンピューティングシステムは、情報を記憶するための手段と、情報を処理するための手段とを含む。情報を処理するための手段は、データ駆動型検査のための手段を含む。このデータ駆動型検査のための手段は、入力デバイス及びデータ処理システムのポートのいずれか又は両方を介して情報を受け取るための手段を含む。この情報は、検査を受けるアプリケーションを表す。データ駆動型検査のための手段は、検査を受けるアプリケーションに少なくとも部分的に基づいて工学設計された検査データの集合を生成するための手段、並びに検査を受けるアプリケーションに少なくとも部分的に基づいて工学設計された検査データの集合を生成するための手段によって生成される工学設計された検査データをアプリケーションが操作するべき計算環境を管理するための手段、及びアプリケーションによる操作を受けた工学設計された検査データを予想される出力と互いに比較するための手段をさらに含む。コンピューティングシステムは、結果の分析を出力するための手段をさらに含む。

【００３０】

別の態様では、本発明は、アプリケーションの検査を管理するため、コンピュータ可読媒体上に非一時的な形態で記憶されるソフトウェアを特徴とする。そのようなソフトウェアは、コンピューティングシステムにある種の処理ステップを実行させるための命令を含む。これらの処理するステップは、検査を受けるアプリケーションに少なくとも部分的に基づいて工学設計された検査データを作り出すステップと、工学設計された検査データをアプリケーションが操作するべき計算環境を制御するステップと、アプリケーションによる操作を受けた工学設計された検査データを予想される出力と比較するステップと、比較の分析を出力するステップとを含む。

【００３１】

本発明のこれら及び他の特徴は、以下の詳細な説明及び添付図面から明らかとなろう。

【図面の簡単な説明】

【００３２】

【図１】アプリケーション検査機械のためのデータ駆動型検査用フレームワークの構成要素間の構造上の関係の図である。

【図２】ユーザインターフェースからのスクリーンを示す図である。

【図３】いくつかのボックスが展開された、図２のスクリーンを示す図である。

【図４】図３中で指定された入力データファイル及び出力データファイルを使用して検査されているグラフを示す図である。

【図５】入力データファイルを設定するためのオプションを示す図である。

【図６】ベースラインを設定するための情報を指定するためのボックスを示す図である。

【図７】レコード単位の比較のためのオプションを示す図である。

【図８】検査が実際に正しく実施したかどうかに関する情報を示す図である。

【図９】ベースラインに対する、アプリケーションを検査した結果の概要を示す図である。

【図１０】他のボックスが展開された、図２のスクリーンを示す図である。

【図１１】ソースレベルのコードカバレッジについての例示的な報告を示す図である。

【図１２】図１のデータ駆動型検査用フレームワーク中に示されるデータサブセット器の

構成要素間の構造上の関係の図である。
【図１３】図１のデータ駆動型検査用フレームワーク中に示されるデータ製造器の構成要素間の構造上の関係の図である。
【図１４】図１のデータ駆動型検査用フレームワーク中に示されるデータ補強器の構成要素間の構造上の関係の図である。
【図１５】図１のデータ駆動型検査用フレームワークの環境管理機械の構成要素間の構造上の関係の図である。
【図１６】効率的な検査手順の概要を示す図である。
【発明を実施するための形態】
【００３３】
検査のために良好なデータを確実に利用可能にすること、知られている環境中でアプリケーションの反復可能な検査を自動的に実施するための方法を提供すること、正確さを測定する、さもなければ検査下のアプリケーションの性能を評価するために使用できる結果を収集すること、及びそれらの結果を評価するための方法を有することによって、より効率的な検査を実現することができる。
【００３４】
図１は、検査コンピュータ１２の中にインストールされて、その検査コンピュータ１２上でアプリケーション１４の秩序だった効率的な検査を容易にするデータ駆動型検査用フレームワーク１０を示す。本明細書で使用する「検査コンピュータ」とは、協調してアプリケーション検査手順を実行する１又は２以上の処理システムを含むことを意図する。
【００３５】
図２は、データ駆動型検査用フレームワーク１０が、アプリケーション１４を検査することに関連して使用するために提供するユーザインターフェースの第１のスクリーンを示す。第１のスクリーンは、１０個のボックスを有する。クリックされると、図３に示されるようにこれらのボックスの各々が展開し、ユーザに多数の選択肢を提供するさらなるボックスが現れる。図１及び図２の両方のボックスは、アプリケーション１４を検査する期間に一般的に実行されるタスクの順序と一致するように、左から右に列状に配置される。
【００３６】
図２の第１の列は、「単一検査」ボックス、「入力データセット」ボックス、及び「出力データセット」ボックスを示す。
【００３７】
図３においてその展開した形で示されるように、「単一検査」ボックスによって、ユーザが、特定の検査を設定すること、検査データセットがどこに保存されることになるかを指定すること、及び検査環境をセットアップするため又はティアダウンするためのいずれかのためのカスタム論理を実装するために使用されることになる任意のグラフ、プラン、若しくはスクリプトを識別すること、又は検査結果の分析を実行することが可能になる。
【００３８】
「入力データセット」及び「出力データセット」ボックスによって、ユーザが、入力データセット及び出力データセットの位置を指定することが可能になる。一般的に、出力データセットは、アプリケーション１４が変更するものであり、一方、入力データセットは、どのようにして出力データセットを変更するのかを決定するためにアプリケーション１４が使用するものである。例えば、アプリケーション１４は、複数の自動車販売店の各々から収入についての日報を受け取ることができ、蓄積された収入のデータベースを更新することができる。その場合、更新されるべきデータベースは、「出力」データセットとなり、日々の収入報告は、「入力」データセットとなる。
【００３９】
図３に示される特定の例は、図４に示されるグラフを検査することに関連する。このグラフは、５つの入力データセット及び２つの出力データセットを特徴とする。図３において、これらのデータセットの名前は、適宜、「入力データセット」及び「出力データセット」ボックス中に列挙される。

【００４０】
図５は、図３中の「Ａ－カスタマ」データベースに検査用フレームワークの注意を引く際に表示する、入力設定ボックスを示す。入力設定ボックスによって、ユーザが、データセットの名前及びタイプを識別することが可能になる。データセットタイプの例としては、入力ファイル及び入力データベーステーブルが挙げられる。入力設定ボックスによって、ユーザが、入力データセットの状態を指定することがやはり可能になる。データセットの状態の例は、データセットが圧縮されているのかいないのかである。入力設定ボックスによって、ユーザが、入力データセットへのパスを指定すること、及びデータセットの記録フォーマットを示すことがやはり可能になる。検査用フレームワーク１０は、指定される入力データセット及び出力データセットの各々について、同様のボックスを示す。
【００４１】
アプリケーションがデータを操作すると、アプリケーションは、典型的には、データをいくつかの方法で変更する。アプリケーション１４がデータを正しく変更するかどうかが、アプリケーション１４が正しく動作しているかどうかに対する重要な手がかりを提供する。しかし、変更されたデータを簡単に点検し、変更されたデータが正しい又は誤っていると断言するのは一般的に可能ではない。一般的に、変更されたデータを、正しいことが判っている他のデータと比較することが必要である。正しいことが判っているデータは、「ベースライン」と呼ばれる。
【００４２】
第１のスクリーンの第２の列は、アプリケーション１４がその機能を正しく実行したかどうかについての確認に関係するボックスを含む。この第２の列は、「ベースライン比較」ボックス及び「メトリクス」ボックスを特徴とする。
【００４３】
「メトリクス」ボックスは、ユーザが、アプリケーションの実行に関するどの統計値が提示されるべきかを指定することを可能にするためのオプションを提供する。これは、例えば、経過時間、ＣＰＵ時間、及びコードカバレッジを含む。
【００４４】
「ベースライン比較」ボックスによって、ユーザが、ベースラインデータを識別し、ベースラインとしてベースラインデータを使用するための準備で、ベースラインデータに対してある種の操作を実行することが可能になる。例えば、ベースラインデータが出力データ中に存在しないある種のフィールドを有するもの、又はベースラインデータ中のある種のフィールドが出力データ中の対応するフィールドと本質的に整合しないことになるものであってよい。一例は、致し方ないが両方の場合で異なる可能性がある、日付／時間スタンプである。
【００４５】
図６は、図３中の「ベースライン比較」ボックス中の「ベースラインを設定する」オプションに検査用フレームワークの注意を引く際に表示する、ベースライン設定ボックスを示す。ベースライン設定ボックスは、比較のタイプを選択する機会をユーザに提供する。比較タイプの例としては、検査データセットリポジトリ中の、シリアルファイル又はＭＦＳファイルの間の比較がある。ベースライン設定ボックスは、ベースラインがどこに配置されるのか、ベースラインが圧縮されているのかどうか、ベースラインの記録フォーマット、及び比較前に削除するいずれかのベースラインフィールド若しくは出力フィールドを指定するための変更をユーザにやはり提供する。
【００４６】
図３に示されるように、ベースラインとアプリケーション１４の出力の間の比較を実行する２つの方法がある。１つの方法は、レコード単位の比較を実行することである。これは、図３中で、「レコード単位の比較を設定する」というオプションによって示される。別の方法は、レコード単位の比較をせずに、補強データを点検することである。これは、図３中で、「統計的比較を設定する」というオプションによって示される。これの例としては、データセット中のレコードの数が予想されるレコードの数に対応するかどうかを判

定することがある。

【００４７】

図６．５は、図３の「ベースライン比較」ボックス中の「レコード単位の比較を設定する」上でクリックした際に利用可能なオプションを示す。利用可能なオプションとしては、比較されるべきキーを指定すること、及びどのフィールドを比較において除外するのかを指定することが挙げられる。これは、例えば、フィールドが本質的に整合しない日付／時間スタンプを含む場合に有用である。というのは、同じ時間は、１回以上発生することができないからである。

【００４８】

第３の列は、検査の実際の実行を制御するための単一検査実施ボックスを含む。単一検査実施ボックスによって、履歴結果を保存する、並びにベースライン分析のみを実施するというオプションが可能になる。

【００４９】

第４で最後の列は、結果の分析のためのオプションを含む。様々な報告を生成することができる。しかし、検査の結果を実際に点検する前に、検査が実際に正しく実施されたのかを判定することが有用である。特に、全ての入力ファイル及び出力ファイルが正しく指定されたこと、並びに検査をセットアップするステップ、検査を実際に実施するステップ、及び結果を分析するステップが全て首尾よく完了したことを確認するのは、有用である。これは、第４の列中の「単一検査結果」ボックス中の「実施についてのイベント詳細を閲覧する」を選択することによって実行することができる。これは、図８に示されるような報告をもたらすことになる。図８に図示される報告によれば、特定の分析ステップを除いて全てがうまくいった。何が悪かったのかについての詳細は、さらに報告中をクリックすることによって識別することができる。

【００５０】

検査がユーザの満足をもたらしたかどうかを決定した後、検査の結果をベースライン結果と比較する報告を点検することが可能である。図９に示される１つのそのような報告は、ベースラインと、アプリケーション１４を検査することによりもたらされる結果の間の比較の概要である。この報告は、図３中の「ベースライン比較結果」ボックス中の「概要を閲覧する」上でクリックすることによって得られる。報告はベースラインレコードの数及び違いのあるレコードの数を示す。明らかであるように、図９の検査結果は、検査を受けたアプリケーションが多数のエラーを起こしたことが示唆される。

【００５１】

どれだけ多くのエラーをアプリケーションが起こしたのか、及びどこでエラーが発生したのかを見ることに加えて、コードカバレッジについての報告を閲覧することも可能である。コードカバレッジは、グラフレベル、構成要素レベル、及び種類レベルのカバレッジメトリクスを含む、様々な方法で表すことができる。利用可能な選択は、図３中の「コードカバレッジ結果」ボックス上でクリックすることによって見ることができる。これがボックスを展開し、図１０に示される選択が現れる。

【００５２】

図１１は、ソースレベルカバレッジメトリクスについての報告の例を示す。この報告は、図１０中の「コードカバレッジ結果」ボックス中の「ソースレベルカバレッジメトリクスを閲覧する」上でクリックすることによって得られる。

【００５３】

説明されるデータ駆動型検査用フレームワーク１０は、データ駆動型検査用フレームワーク１０のインストール前に検査コンピュータ１２に存在しなかった機能性を、検査コンピュータ１２に提供する。この方法では、説明されるデータ駆動型検査用フレームワーク１０は、それがインストールされている検査コンピュータ１２の動作に、著しい技術的な改善をもたらす。

【００５４】

検査を受けるアプリケーション１４は、ソースコードのコンパイルを通して得られたオ

ブジェクトコードを含むことができる。ある種の実施形態では、このソースコードは、指向型非周期グラフを表す。他の実施形態では、ソースコードは、プランを表す。

【００５５】

いくつかの実施形態では、ソースコードは、グラフを表す。これらのグラフのノードは、構成要素間のデータの流れを可能にするため、指向性リンクにより接続されるポートを有する処理構成要素を規定する。そのようなグラフでは、構成要素は、入力ポート上で入力データを受け取るステップ、そのデータを処理するステップ、及び出力ポート上に結果となる出力を提供するステップによって、計算を実施する。

【００５６】

いくつかの実施形態では、ソースコードは、プランを表す。プランは、ノードがタスクを表す指向型非周期グラフであり、指向型リンクは、上流タスクが終了するまで下流タスクは開始できないように、タスク間の依存関係を規定する。いくつかの実施形態では、タスクは、グラフを実行するために使用される。

【００５７】

アプリケーション１４に関連するコンパイルされたソースコードは、「pset」又はパラメータセットを表す情報を含むこともできる。パラメータセットは、パラメータ及びそれらのパラメータの各々に対応する値のリストを提供する。いくつかの実施形態では、パラメータセットは、グラフをカスタマイズするためのパラメータを提供するために使用される。

【００５８】

アプリケーション１４は、アプリケーション１４が導出されるソースコードがデータの流れ図、制御の流れ図、及びプランを表すものに限定されない。実施形態は、Ｃのコード又はＪａｖａのコードなどの任意のコンピュータ言語で書かれたソースコードの、好適なコンパイル又は解釈によって得られるオブジェクトコードを、アプリケーション１４が含む実施形態をやはり含む。そのようなアプリケーションの実行のさらなる記載は、その内容が参照により本明細書に組み込まれる、２０１４年８月７日公開の、Isman, et al.の「DATA RECORDS SELECTION」という名称の米国特許公開第２０１４－０２２２７５２号明細書に提供される。

【００５９】

アプリケーション１４は、その実行が、１又は２以上の変数の値によってトリガされるルールを実装することが多い。これらの変数は、入力データに対応する入力変数であってよい。あるいは、これらの変数は、入力データ中の１又は２以上の入力変数に依存する導出変数であってよい。アプリケーションの効果的な検査のため、アプリケーション中の完全なコードカバレッジが達成されるように、アプリケーション１４中のあらゆる論理ルールの実行を行わせるのに十分な検査データを提供するのが望ましいことがある。少なくとも対応する最小回数、論理ルールが実行させられること、又は逆に、対応する最大回数以下、論理ルールが実行させられることが望ましいこともある。

【００６０】

効率的な検査への第１の障害は、アプリケーション１４による操作を受けるときに、上述の要件を満足する好適な検査データを獲得することである。本明細書で企図される具体的な検査データは、その各々が１又は２以上のフィールドからなる一連のレコードとして構成されるデータである。

【００６１】

検査データを取得する１つの方法は、プロダクションシステムから引かれてきた全データ量を使用することである。原理的に、この方法は、非常に大きいのでコードの何らかの特徴の検査を漏らす可能性が漸近的にゼロに近づくことになる量のデータを検査することに依存する。

【００６２】

これらのデータ量は非常に大きいことが多かった。結果として、各検査サイクルは、法外な長時間がかかることになる。

【００６３】
上述の障害を克服するために、説明されるデータ駆動型検査用フレームワーク１０は、アプリケーション１４の検査に使用するための工学設計された検査データを生成するデータエンジニアリングモジュール１６を含む。どのようにして工学設計された検査データを生成するのかについての例は、Isman, et al.の、２０１３年１２月１８日に出願された「DATA GENERATION」という名称の米国仮出願第６１／９１７，７２７号明細書、及びIsman, et al.の、２０１３年３月１４日に出願された、米国特許公開第２０１４／０２２２７５２号明細書と「DATA RECORDS SELECTION」という名称の米国出願第１３／８２７，５５８号明細書の両方に記載される。上述の出願の両方の内容は、本明細書に参照によって組み込まれる。

【００６４】
本明細書に記載されるデータ駆動型検査用フレームワーク１０では、全データ量はコードカバレッジが依存する唯一のものではないという発見を利用することが意図される。実際に、コードカバレッジは、データ自体の性質にも依存する。特に、コードカバレッジは、そのデータの論理濃度又は論理分散に依存する。実際には、検査に実際に使用されるデータがより高い論理濃度を有するように工学設計されることを条件として、劇的に小さい量のデータを使用して、所望のコードカバレッジを達成することが通常できる。

【００６５】
本明細書で使用する、「コードカバレッジ」という用語は、ソースコードが検査手順によって検査を受けた範囲の広さである。これは、第１の値対第２の値の比率として表すこと、しばしば百分率として表すことができ、第２の値が検査を受けるコードの全量の定量的測度を表し、第１の量が検査を受ける実際の量の定量的測度を表す。いくつかの場合では、第１と第２の変数は、検査を受ける特徴対実装される特徴を表す。他の場合では、第１と第２の変数は、検査を受けるソースコードの列とソースコードの全列を表す。定量的測度の正確な性質は、本発明の理解に対し明らかに重要ではない。

【００６６】
データ駆動型検査用フレームワーク１０は、１００％コードカバレッジは言うまでもなく、何らかの特定のコードカバレッジを達成する必要はない。コードカバレッジは、工学設計上の判断に基づいて、ユーザによって設定されるパラメータである。しかし、ユーザが選択するどんなコード検査カバレッジでも、本明細書に記載される方法及び装置は、そのコードカバレッジを達成するために必要な検査データの量を減少させることになり、プロダクションデータの全量の単純な操作によって達成することが場合によって可能なものよりも、より信頼性が高く確定的な方法で、そのターゲットのコード検査カバレッジを達成することになる。

【００６７】
特に、検査データの組が与えられれば、コードのある部分を働かせることになる。異なる検査データセットは、一般的に、コードの異なる部分を働かせることになる。例えば、検査データがデータレコードを何度も何度も単に繰り返す場合、コードの非常に限られたサブセットだけを働かせることになる。対照的に、全ての種類の値の組合せを有する多様なレコードを含む検査データは、コードのより大きいサブセットを働かせる可能性が高いことになる。

【００６８】
データエンジニアリングモジュール１６は、構成要素セットから選択される１又は２以上の構成要素を含む。各構成要素は、特定の方法を使用して工学設計された検査データを生成する。どの方法を使用するのか、したがってどの構成要素が必要であるのかについての選択は、さしあたっての特定の状況に依存する。

【００６９】
データエンジニアリングモジュール１６の構成要素は、データサブセット器１８、データ補強器２０、肯定データ製造器２２、及び否定データ製造器２４のうちの１又は２以上を含む。データサブセット器１８は、その論理濃度を増加させるように、既存のデータの

精製を通して、工学設計された検査データを生成する。データ補強器２０は、既存のデータを補強することによって、工学設計された検査データを生成する。肯定データ製造器２２及び否定データ製造器２４の両方は、検査要件に基づいて、工学設計された検査データを作り出す。

【００７０】

アプリケーション１４中のある論理を検査するのに必要な種類のデータが既存のデータ中に存在しない場合がある。これは、しかし、この論理が検査を受けないはずであるという意味ではない。

【００７１】

この論理を働かせるため検査データにのみ依存する場合、論理は、全く検査を受けないことになる。これは、最大限の既存のデータの精製でも、その論理を検査するために使用できるデータをもたらすことが補償されないことになるためである。この状況に対応するため、データエンジニアリングモジュール１６のある種の実施形態は、否定データ製造器２４を含む。

【００７２】

否定データ製造器２４は、通常は存在しないデータを提供する。これは、通常ならば検査を受ける機会を決して持たないコードの実行を可能にすることによって、検査のコードカバレッジを拡げる。否定データ製造器２４は肯定データ製造器２２とは異なる。というのは、否定データ製造器２４は、本明細書では「否定データ」と呼ばれる、典型的なデータセット中に（又は、典型的なデータセットのサンプル中に）通常存在しないデータを提供するからである。対照的に、肯定データ製造器２２は、本明細書では「肯定データ」と呼ばれる、典型的なデータセット中に（又は、典型的なデータセットのサンプル中に）通常存在するデータを生成するからである。否定データの例としては、そのフィールドにとって予め規定された文字の組の中にない文字を含むフィールドエントリ、又はそのフィールドにとって予め規定された値の範囲外にある値を有するフィールドエントリ、又はフィールドエントリの１若しくは２以上の部分に正しくない数の文字を含むフィールドエントリなどといった、フィールドのフォーマットに適さないフィールドエントリが挙げられる。一例としては、文字を含む社会保障番号、又はゼロの値を有する誕生月がある。否定データの他の例は、フィールドフォーマットとは一致するが、しかし参照整合性を混乱させるものが挙げられる。一例としては、既存のカスタマを何ら特定しない、正しくフォーマットされたカスタマ番号がある。そのような否定検査事例の使用が、コードカバレッジを拡張する。しかし、そのような否定データは、プロダクションデータセット中に存在しない可能性があり、したがって、製造する必要が一般的にあることになる。

【００７３】

工学設計された検査データを生成した結果として、アプリケーション１４が開発されている間に、アプリケーション１４のインタラクティブなデバッグを容易に実行することが可能になる。これは、実施するのに何分も、又は数時間さえかかる可能性がある大きいデータセットを処理するよりも、はるかに生産的である。例えば、工学設計された検査データが局在化された環境で使用されると、ビジネスルール環境中のルールを変えることの効果を各レコード上で見ることが可能になる。

【００７４】

データサブセット器１８は、アプリケーション１４の開発者がアプリケーション１４になされた変更の効果を迅速に見ることができるように、十分に小さい工学設計された検査データの組を生み出す。しかし、工学設計された検査データの組は、ただ小さい以上のものである。工学設計された検査データの組は、高い検査論理濃度も有する。その高い検査論理濃度の結果として、工学設計された検査データは、全データセットを必要とすることなく、アプリケーション１４中の全てのコードを働かせる。これによって、同じ計算リソースの消費で高いコードカバレッジを達成させる結果となる。

【００７５】

図１２は、データサブセット器１８の詳細を示す。データサブセット器１８は、実際の

プロダクションデータ２６（又は、サブセットを行うための任意の入力データセット）、論理仕様２８、及び制御変数３０を受け取る。論理抽出器３１は、検査を受ける論理機能を識別し、それらをデータ精製器３２に提供する。論理抽出器３１とデータ精製器３２の両方は、データサブセット器１８の構成部分である。データ精製器３２は、次いで、プロダクションデータ２６を処理して、データ精製物３３を生成する。データ精製器３２は、制御変数３０によって指定されるような抽出手順を使用し、論理抽出器３１によって指定される論理を検査するのに関係するそれらの部分を抽出することによって、上記を行う。こうして、本明細書に使用される「データ精製器」という用語は、指定された抽出手順を使用して、入力データセットからデータの一部を抽出し、「データ精製物」と呼ばれる抽出データを生み出す処理モジュールのことを言うために使用される。

【００７６】

データ精製物３３は、サブセットルールに基づいてプロダクションデータ２６から選択される。これらのサブセットルールは、いくつかのソースから来る可能性がある。一例では、ユーザがサブセットルールを指定する。別の例では、サブセットルールは、アプリケーションの実行からのフィードバックに基づいて策定される。さらに別の例では、アプリケーション１４中のコードの一部又は全部を実行させることになるデータレコードを、データ精製物３３が含む。

【００７７】

一例として、プロダクションデータ２６はデータレコードを含むことができ、データレコードの各々がフィールドを含み、いくつかのフィールドがある種の許容される値を有し、そのいくつかは他のものよりも発生する可能性が高い。異なる許容される値は、コードの異なる部分を働かせる。こうして、コードを網羅的に検査するために、全ての値の全ての組合せが生じなければならない。いくつかの実施形態では、工学設計された検査データは、許容される値の全ての組合せを得るためさほど多くないレコードが必要となるように、それらの可能性の低い値について、発生する可能性をより高くさせることによって導出される。

【００７８】

この場合、工学設計された検査データは、レコードの値の確率分布がより均一になされているデータと見なすことができる。言い換えれば、特定の許容される値が、プロダクションデータ２６中で比較的低い確率で発生する場合、その値は、工学設計された検査データ中ではより高い確率で発生することになる。逆に、特定の許容される値が、プロダクションデータ２６中で比較的高い確率で発生する場合、その値は、工学設計された検査データ中ではより低い確率で発生することになる。これは、最も起こる可能性のあるイベントの確率が減らされ、最も起こる可能性のないイベントの可能性が増やされる工学設計された検査データを有することの正味の効果を有する。これは、確率値の拡がりを減らす。確率値の拡がりがゼロである、これの限定した事例は、均一な分散を規定することによる。確率値の全体の拡がりの減少は、こうして、分散を均一な分散にする結果となる。これは、検査のためのより効率的なデータセットをもたらす結果となる。というのは、より可能性の高い値により引き起こされる冗長性が減らされると同時に、最も可能性の低い値を確実に得るために必要な量が減らされるからである。この効率の程度は、工学設計された検査データの検査論理濃度に対応する。

【００７９】

多くの場合に、プロダクションデータ２６は、データベースからの複数のテーブルからなることになる。これらのテーブルは、第１のテーブルポイント中にポインタを有することによって、第２のテーブル中のレコードに結合すること、又は第２のテーブル中のレコードを「参照」することができる。

【００８０】

ポインタが何かを指すときはいつも、２つの可能性がある。（１）ポインタが何か有効なものを指す。（２）ポインタが何か有効なものを指さない。

【００８１】

第１の可能性では、第１のテーブル中の各ポインタは、第２のテーブル中の有効なレコードを指す。この第１の可能性では、２つのテーブルが「参照整合性」を有すると言われる。そのため、本明細書で使用する「参照整合性」という用語は、データセットの別の部分の中の値へのデータセットの１つの部分の中の各参照が有効である、１又は２以上のデータセットを記載するために使用される。

【００８２】

上に記載された第２の可能性では、第１のテーブル中の少なくとも１つのポインタは、第２のテーブル中の有効なレコードを指さない。この第２の可能性では、２つのテーブルが参照整合性を欠如していると言われる。

【００８３】

適切な検査のため、プロダクションデータ２６が参照整合性を有する場合、工学設計された検査データも参照整合性を有さなければならないことが好ましい。したがって、データ精製器３２は、参照整合性を保持するデータ精製物３３を提供しなければならない。

【００８４】

そのような参照整合性が保持されているかどうかを決定するために、データ精製器３２は、データ精製物３３を整合性チェッカ３４に提供する。データ精製物３３が参照整合性を有すると整合性チェッカ３４が決定する場合、データ精製物３３は、データサブセット器１８の出力データサブセット３５として提供される。さもなければ、データ精製物３３は、修復のために再参照器３６に提供され、その後出力データサブセット３５として提供される。

【００８５】

いくつかの実施形態では、再参照器３６は、データ補強器２０と同じ機能性を実装する。例えば、１つのデータセット中のポインタが別のデータセット中のレコードを指さないために参照整合性の欠如が生じる場合、再参照器３６は、データ補強器２０により使用されるのと同じ方法を使用して、第２のデータセットを好適なレコードで補強することができる。再参照器３６は、こうして、データエンジニアリングモジュール１６の任意選択の構成部分と見なすことができる。

【００８６】

示される特定の実施形態では、データサブセット器１８は、出力データサブセット３５を閲覧すること及び／又は出力データサブセット３５のプロファイリングを行うことを可能にするデータ点検ユニット３７も含む。しかし、他の実施形態では、データ点検ユニット３７が存在しない。

【００８７】

データ点検ユニット３７がビューアである実施形態及びデータ点検ユニット３７がプロファイラである実施形態は、データ点検ユニット３７を有する実施形態の中にある。データ点検ユニット３７が、ユーザが何をしたいのかに基づいて閲覧及びプロファイリングの両方を行うことが可能な構造である実施形態は、データ点検ユニット３７を含む実施形態の中にやはり含まれる。

【００８８】

本明細書で使用するデータサブセットを「プロファイリングすること」は、例えば、メタデータ又はそのサブセットについての補強データを取得することが挙げられ、プロファイリングの結果が「プロファイル」と呼ばれる。補強データとしては、レコードの数、それらのレコード中の値の範囲、及び確率分布のｎ次モーメント、ここでｎは正の整数などのデータ内の値の統計的又は確率的記述などの特徴が挙げられる。

【００８９】

例えば新しいシステムを開発するとき、精製するのに利用可能なプロダクションデータが存在しないことがある。他の場合には、プロダクションデータは、獲得するのが非常に困難となる。これらの状況に対処するため、データエンジニアリングモジュール１６の肯定データ製造器２２を活性化する。

【００９０】

図１３を参照して、肯定データ製造器２２は、論理仕様２８、制御変数３０、及び重要な関係情報３８を受け取る。論理抽出器３１は、検査を受ける論理機能を識別し、それらをデータ生成器４０に提供する。データ生成器４２は、次いで、制御変数３０により指定されるような抽出手順を使用して、好適な検査データを生成する。どのようにしてデータを生成するのかについての例は、Isman, et al.の、２０１３年１２月１８日に出願された「DATA GENERATION」という名称の米国仮出願第６１／９１７，７２７号明細書、及びIsman, et al.の、２０１４年８月７日公開の、「DATA RECORDS SELECTION」という名称の米国特許公開第２０１４／０２２２７５２号明細書に記載される。

【００９１】

好ましくは、結果として得られる製造された検査データ３９は、適切な検査のため、参照整合性を有する。したがって、製造された検査データ３９は、整合性チェッカ３４に提供されて、参照整合性が確立されているかどうかを決定する。製造されたデータが参照整合性を有すると整合性チェッカ３４が決定する場合、製造された検査データ３９は、肯定データ製造器出力４１として提供される。製造された検査データが参照整合性を有さない場合、製造された検査データ３９は、修復のために再参照器３６に提供され、次いで肯定データ製造器２２の出力として提供される。

【００９２】

いくつかの実施形態では、肯定データ製造器２２は、データ駆動型検査用フレームワーク１０内で、製造された検査データ３９を閲覧すること及び製造された検査データ３９のプロファイリングを行うことを可能にするデータ点検ユニット３７も含む。他の実施形態では、データ点検ユニットが存在しない。

【００９３】

いくつかの場合では、プロダクションデータ２６が存在するが、全く要求される形式ではない。そのような場合では、データエンジニアリングモジュール１６のデータ補強器２０を活性化することにより、プロダクションデータを補強することが有用である。

【００９４】

データ補強器２０は、例えば、既存のプロダクションデータ２６に１又は２以上のフィールドを追加するため、及び供給されるルールに基づいてそれらのフィールドを満たすようデータを生成するために使用することができる。

【００９５】

図１４は、データ補強器２０の詳細を示す。データ補強器２０は、実際のプロダクションデータ２６（又は、補強されるべき任意の入力データセット）、論理仕様２８、及び制御変数３０を受け取る。論理抽出器３１は、検査を受ける論理機能を識別し、それらをデータ精製器３２及びデータ変更器４８に提供する。データ精製器３２は、次いで、制御変数３０によって指定されるような抽出手順を使用し、論理抽出器３１によって指定される論理を検査するのに関係するそれらの部分を抽出するように、プロダクションデータ２６を処理する。論理抽出器３１によって提供される情報に基づいて、データ変更器４８は、適したフィールドを追加し、それらのフィールドの中に好適な値を入力し、こうして補強データ４９を生成する。

【００９６】

好ましくは、データ変更器４８によって提供される補強データ４９は、適切な検査のため、参照整合性を有する。したがって、データ変更器４８によって提供される補強データ４９は、整合性チェッカ３４に提供されて、参照整合性が確立されているかどうかを決定する。補強データ４９が参照整合性を有すると整合性チェッカ３４が決定する場合、補強データ４９は、データ補強器２０の補強したデータ出力５１として提供される。さもなければ、補強データ４９は、修復のために再参照器３６に提供され、次いでデータ補強器２０の補強したデータ出力５１として提供される。

【００９７】

いくつかの実施形態では、データ補強器２０は、データ駆動型検査用フレームワーク１０内で、補強したデータ出力５１を閲覧すること及び補強したデータ出力５１のプロファ

イリングを行うことを可能にするデータ点検ユニット３７も含む。他の実施形態では、データ補強器２０は、データ点検ユニットを持たない。

【００９８】

いくつかの場合では、通常、プロダクションデータ中に現れる任意のデータにより働かされることのないコードセグメントを働かせたいと望む場合がある。これを行うために、データエンジニアリングモジュールは、否定データ製造器２４を含み、その機能は、そのような否定的な検査事例を作り出すことである。

【００９９】

効率的な検査への第２の障害は、検査環境をセットアップし、制御し、次いでティアダウン(tear down)する必要から生じる。

【０１００】

一般的に、検査は、検査群中の複数の検査を実施すること、及び多くの外部データセットと相互作用する１又は２以上のグラフ及びプラン上でそうすることを含む。これらのデータセットは、ファイル、テーブル、待ち行列、マルチファイル、及びウェブサービスから来る場合がある。アプリケーション１４に検査群を実行させるタスクを達成させるために、データ駆動型検査用フレームワーク１０は、計算環境マネージャ４４を提供する。

【０１０１】

計算環境マネージャ４４は、知られている環境中の知られている入力で制御されているようにアプリケーション１４を実施するタスクを実行する。これは、検査を受ける特定のアプリケーション１４を指定するのに柔軟性を提供する。計算環境マネージャ４４は、アプリケーション１４により処理されるべき入力データに対応する補強データ、データフラグ、出力ディレクトリ、並びにセットアップ、ティアダウン、及び報告のためのカスタマイズ可能論理を含むリポジトリフォルダ(repository folder)を保持する。

【０１０２】

計算環境マネージャ４４は、データセットをファイル又はテーブルとして自動的にセットアップする。これらのデータセットは、データのソース、すなわちアプリケーション１４が操作することになるデータ、及びターゲット、すなわちアプリケーション１４による処理の結果が最終的に置かれることになる場所を含む。計算環境マネージャ４４は、次いで、自動的に、ソース及びターゲットを正しい初期状態へと設定し、好適な検査群を使用してアプリケーション１４を実施し、結果をターゲット中に置き、環境をそのプリセット条件に復元する。いくつかの場合では、環境マネージャ４４は、以前の環境をバックアップし、検査が完了した後に以前の環境を復元する。環境の自動的なセットアップ及びティアダウンによって、最小の手作業で検査を繰り返すことが容易になる。

【０１０３】

コンピュータシステムは、常に増加する抽象性の、ネストされた(nested)層の組と見なすことができる。各層は、より高いレベルの抽象化の層が使用することができる論理構造を作り出す。論理構造は、メモリ状態及び環境変数の値を含む。

【０１０４】

アプリケーションを実行するとき、アプリケーションは、これらの層上で実行中であると見なすことができる。より下位の層が作り出す論理構造の組は、アプリケーションを実行する環境と見なすことができる。物理的な構造の適切な検査が一定の物理的な環境を維持することに依存することが多いのとほとんど同じ形で、アプリケーションの適切な検査のために、同じ環境を維持することが好ましい。

【０１０５】

ここで図１５を参照して、一実施形態では、計算環境マネージャ４４は、一方はセットアップフェーズの期間のもの、他方はティアダウンフェーズの期間のものという２つの環境の移行を行わせる環境移行機械４６を含む。

【０１０６】

環境移行機械４６は、入力仕様５３及び出力仕様５０を受け取る。入力仕様５３は、入力検査データが来るべきソース５２を識別する。この入力は、ファイル、マルチファイル

、待ち行列、ウェブサービス、又はそれらの任意の組合せであってよい。出力仕様５０は、検査の出力が置かれることになっているターゲット５４を識別する。環境移行機械４６は、入力の初期状態、出力、及び任意の環境変数についての情報を含む初期化信号５６も受け取る。最後に、環境移行機械４６は、検査の開始を示すための検査信号５８を受け取る。

【０１０７】

いくつかの実施形態では、セットアップフェーズ期間に、環境移行機械４６は、第１のデータリポジトリからソース５２へ検査データ及び／又はベースラインデータをコピーする。検査データ及び／又はベースラインデータはソース５２に実際の検査手順の期間記憶される。検査手順が完了した後、ティアダウンフェーズが始まる。このティアダウンフェーズ期間に、環境移行機械４６は、ターゲット５４から検査データを削除する。

【０１０８】

検査信号５８を受け取ったら、環境移行機械４６は、環境バックアップ機械６０と通信して、環境のバックアップ６２を作り出す。これの後に、入力ソーススイッチ６４に好適なソース５２を指し示させ、出力ソーススイッチ６６に好適なターゲット５４を指し示させることが続く。

【０１０９】

これらのタスクが完了すると、環境移行機械４６は、実行器６８に信号伝達して、アプリケーション１４に１又は２以上の検査８０を含む検査群７９を実行させる。いくつかの実践では、検査群の実行は、１又は２以上のスクリプトの自動実行を含む。実行が完了すると、実行器６８は環境復元機械７０に信号伝達し、環境復元機械７０は、次いでバックアップ６２を取り出して環境をその初期状態に復元する。

【０１１０】

実行の最中に、アプリケーション１４は、１又は２以上のルールを実施する。いくつかの実施形態では、ルールは、少なくとも条件表現及び実行表現を含む仕様によって指定される。条件表現が「真」と評価されると、アプリケーション１４は、実行表現を評価するように進行する。しかし条件表現が「真」と評価されるかどうかは、データ中の１又は２以上の変数の値に依存する場合がある。これらの変数は、入力データに対応する入力変数であってよい。あるいは、これらの変数は、１又は２以上の入力変数に依存する導出変数であってよい。アプリケーション１４が特定の検査の実施期間にルールを実行するかどうかは、こうして最終的に、ルールに対応する条件表現を「真」と評価させることになる変数を検査データの選択が有するかどうかに依存する。

【０１１１】

いくつかの例では、アプリケーション１４は、トリガされるルールの全てを実行する。他の例では、アプリケーション１４は、トリガされるルールの全てよりも少ないルールを実行する。ルールは、参照によりその内容が本明細書に組み込まれる、２００７年４月１０日出願の米国特許第８，０６９，１２９号明細書の５列６１行と６列１１行の間により詳細に記載される。

【０１１２】

実行器６８が検査群７９を完了すると、結果分析モジュール７２が引き継いで、検査結果の分析を開始する。結果分析モジュール７２の機能には、正しい結果のこれらの知られている組を作り出す機能、及び検査されているアプリケーション１４が最終的に正しい答えに到達することを確認するプロセスを自動化する機能がある。

【０１１３】

いくつかの場合では、検査されているアプリケーションの、より古いバージョンがある。この、検査されているアプリケーションのより古いバージョンは、典型的には、現在使用中のバージョンである。そのため、検査されているアプリケーションのより古いバージョンは、出力の正確さを確立するための最高の基準と見なすことができる。したがって、検査されているアプリケーションによって置き換えられることが意図されるアプリケーションの、このより古いバージョンは、「ゴールドスタンダードバージョン」(gold standa

rd version)と呼ばれることになる。

【０１１４】

同じ環境を使用して同じデータに対して実行したときに、検査されているアプリケーションのバージョンが、ゴールドスタンダードバージョンによって獲得される結果と一致する結果をもたらさない場合、検査されているアプリケーションのバージョンが正しくない結果を出力しているという推論を行うことができる。

【０１１５】

アプリケーション１４を検査することの中に生じる１つのステップは、アプリケーション１４が、事実上データを正しく処理したのかどうかを決定するステップである。このステップを実行するために、アプリケーション１４の機能仕様により規定される、データセットに対する操作の予想される結果と、実行器６８により獲得されるような、同じデータセットに対する操作の測定結果との間で、何らかの対応を確立する方法が存在しなければならない。言い換えれば、正しい答えのベースライン７４を獲得する必要がある。そのようなベースライン７４が入手できると、結果分析モジュール７２が、結果７８をベースライン７４と比較することによって結果７８を確認する。

【０１１６】

ベースライン７４を獲得する方法は、アプリケーション１４が置き換えている何らかのものと、アプリケーション１４がどれだけ異なっているのかに部分的に依存する。一般的に、違いが大きいほど、ベースラインを生成するのが困難になる。

【０１１７】

抽象的なレベルで、データセットＸ及び環境Ｅが与えられた場合、アプリケーションｆのバージョンｎは、出力Y=fn(X,E)を生成することになる。問題は、Ｙが正しいことをどのようにして決定するのかである。

【０１１８】

一般的に、３つの可能性がある。

【０１１９】

第１の可能性は、(X,E)を操作することができる、アプリケーションの異なるバージョン、すなわちバージョンｍが存在する可能性である。バージョンｍが信頼できると考えられる場合、fn(X,E)=fm(X,E)が成立するかどうかを問うことによって結果Ｙの正確さを確立する。

【０１２０】

第２の可能性は、完全に信頼できると考えられない、アプリケーションの別のバージョン、すなわちバージョンｍが存在する可能性である。その場合、fn(Z,E)=fm(Z,E)であって、Z⊂Xであり、fm(X,E)がＺについては信頼できるがＺ^{c}については信頼できないと考えられ、Ｚ^{c}がＺの補数であることが成立するのかを問わなければならない。fn(Z^{C},E)の正確さを確立するために、典型的には、正しい結果を手動で決定しなければならない。

【０１２１】

第３の可能性は、信頼できると判っているアプリケーションのバージョンがない可能性である。これは、単に、Z= である、第２の可能性の縮退した場合である。この場合、正しい結果を決定するための手順は、手動で実行される。

【０１２２】

ベースライン７４を獲得する１つの方法は、検査下のアプリケーション１４が既存のアプリケーションを、本質的に同じ機能性を有して置き換えることを意図するときに有用である。これは、上に規定された第１の可能性に対応する。この場合、ベースライン７４は、アプリケーションのゴールドスタンダードバージョンによって生成される結果から来てよい。

【０１２３】

いくつかの場合では、検査下であるアプリケーション１４は、既存のアプリケーションに対する拡張となっている。拡張とは、検査下であるアプリケーション１４が、異なる結果をもたらすことが予想されて、事実上意図されるようなものである。上の第２の可能性

に対応するこの状況は、例えば、ゴールドスタンダードバージョンが正しくない答えを引き起こすバグを有し、検査下のアプリケーション１４がそのバグを直すことが意図される場合に生じる可能性がある。

【０１２４】

これらの場合について、結果分析モジュール７２は、どのフィールドが変わったのか、及び／又は出力中のレコードの数が変わったのかどうかを報告する。結果分析モジュール７２は、何らかのフィールドが想定されていないときに意図せずに変化したことを直ちに認識できるように、任意の不一致を報告する。変化が予想されたフィールドについて、正しい答えを決定し、正しい答えをベースライン７４へと入力させるため、人間の介在が必要な場合がある。

【０１２５】

他の場合には、検査下のアプリケーション１４は、真新しいシステムである。これは、上に概説された第３の可能性に対応する。結果として、ベースライン７４を作り出すための基礎として使用できる既存の出力データが存在しない。

【０１２６】

この場合、ベースライン７４は、既存のプロダクションデータ２６で開始し、そのプロダクションデータ２６のサブセットについて正しい結果を（例えば、手動で）入力することによって形成される(built)。これは、検査を受けるアプリケーション１４の基礎をなす論理に目を向け、その論理に基づいて、アプリケーション１４を通る様々な論理パスによって最も影響を受ける可能性が高いソースデータ中のそれらのフィールドを識別することによって達成される。これらは、データのサブセットを選択するときに選ばれるべきフィールドである。

【０１２７】

いくつかの場合では、ベースライン７４を点検する必要なしで、ある種の簡単な検査を自動的に実行することができる。例えば、アプリケーション１４が入力の各レコードについて出力の１つのレコードを出すことが判っている場合、アプリケーション１４に知られている濃度のプロダクションデータ２６を操作させることができ、この場合、出力データの濃度は、アプリケーション１４が機能することについての何らかの情報を提供することになる。特に、プロダクションデータ２６の各々の濃度と、アプリケーション１４によるプロダクションデータ２６に対する操作により生み出される濃度との間に、ゼロでない違いが存在する限り、結果分析モジュール７２は、アプリケーション１４の実装における欠陥の可能性を自動的に信号伝達することができる。

【０１２８】

例えば、ある場合に、アプリケーション１４は、異なる濃度のいくつかの構成部分を含む出力を生成するように意図されており、ここで、それらの異なる濃度間には関係が存在する。一例では、アプリケーション１４は、ソース５２の中の入力を操作し、ターゲット５４の中に２つの別個のテーブルを生成する。それら２つのテーブルの濃度間に関係が存在する限り、結果分析モジュール７２は、自動的にそのような違いを検出して、アプリケーション１４の実装における欠陥を示す情報を出力する。

【０１２９】

別の例では、ソース５２の中の入力テーブルは、Ｎ個のレコードを有する場合がある。ターゲット５４の中の出力テーブルがＮ個のレコードをやはり有するはずであることが判っている場合、出力テーブルの中のレコードの数を確認するのは、ソフトウェアがどれだけ良好に働いたのかを確認する良好な方法である。例えば、入力の中にただＮ個のレコードが存在するときに出力の中にＮ＋１個のレコードが存在することが観測される場合、これはエラーを示唆することになる。

【０１３０】

上述の例の一般化である別の例では、アプリケーションが、確定的な方法で、レコードの数を変えることが知られている。こうして、一般的に、Ｎ個のレコードの入力テーブルについてのレコードの出力数が何らかの知られている関数ｆについてｆ（Ｎ）である場合

、アプリケーション中のエラーを識別する１つの方法は、入力テーブルがＮ個のレコードを有するときに出力テーブルが実際にｆ（Ｎ）個のレコードを有するのかを見ることである。

【０１３１】

実行後、アプリケーション１４の実行を示す情報、特に、アプリケーション１４に提供された検査データとアプリケーション１４の相互作用に関する情報を提供する報告を提供することが有用である。そのような情報の例としては、アプリケーション１４が実行した若しくは実行しなかったルール、アプリケーション１４中の各ルールが実行された回数、又はアプリケーション１４と検査データの間の相互作用に光を当てることになる任意の他の情報を挙げることができる。

【０１３２】

報告に基づいて、ユーザが、追加の検査データを識別することが可能である。この追加の検査データは、例えば、任意の未実行ルールを実行させたはずであったデータ、又は特定の論理ルールを指定の回数実行させたはずであったデータ、又は別の所望の実行結果をもたらしたはずであったデータであってよい。ユーザは、次いで、それらの追加のサブセットルールにしたがって、データレコードの更新サブセットを選択させる、新しいサブセットルールを策定することができる。データレコードの更新サブセットは、以前の未実行ルールの一部又は全部を実行させるのに十分なデータレコード、指定の回数ルールの一部又は全部を実行させるのに十分なデータレコード、又は別の所望の実行結果をもたらすのに十分なデータレコードを含むことができる。

【０１３３】

結果分析モジュール７２が提供できる情報の種類には、検査データがコードを働かせた範囲についての報告がある。この報告は、検査を受けたコードの列の百分率などの補強スコア、並びにコードのどの列が検査を受けなかったのかなどのより詳細な情報を含む。この情報によって、検査を受けるコードの百分率及び検査から省かれたコードの重要性の両方の観点から検査が適切であったかどうかを、ユーザが判断することが可能になる。

【０１３４】

図１６は、本明細書に記載される構成要素を使用する、効率的な検査手順の全体的な概要を提供する。検査手順は、一般的に、データ関連ステップ８２とアプリケーション関連ステップ８４とに分けられる。

【０１３５】

データ関連ステップ８２は、任意の既存のプロダクションデータにプロファイリングを実施することを含む。これは、図１６中の「プロダクションデータをプロファイリングする」というテキストによって識別される、ステップ８６として識別される。

【０１３６】

次のデータ関連ステップは、そのプロファイルから、プロダクションデータに関するある種の補強データを獲得することである。このステップは、「メタデータを得る」というテキストによって識別される、ステップ８８として図１６中で識別される。「メタデータ」とは、補強データのことを言うと理解される。そのような補強データの例としては、限定するものではないが、キーのリスト、フィールド濃度、及び値の範囲が挙げられる。

【０１３７】

このメタデータ又は「補強データ」は、「関係のある完全なサブセットを作る」というテキストによって識別されるステップ９０で図１６中に識別されるように、データの関係のある完全なサブセットを生成することに使用される。

【０１３８】

いくつかの実践は、否定検査データを作り出して含むことによって、関係のある完全なデータサブセットを補強することを含む。これは、「否定データを作り出す」というテキストによって識別されるステップ９２によって、図１６中に示される。

【０１３９】

他の実践は、合成データの製造によって、関係のある完全なデータサブセットを補強す

ることを含む。これは、「新しいデータを製造する」というテキストによって識別されるステップ９４によって、図１６中に示される。

【０１４０】

アプリケーション関連ステップ８４は、アプリケーションを形成すること、又は何らかの方法で既存のアプリケーションを直すこと若しくは拡張することによって、既存のアプリケーションを変更することのいずれかを含む。アプリケーションを形成するステップは、ステップ９６として図１６中に示され、「ＡＰＰを形成する」というテキストによって識別される。何らかの方法で既存のアプリケーションを直すこと若しくは拡張することによって、既存のアプリケーションを変更するステップは、ステップ９８として図１６中に示され、「ＡＰＰを変更する」というテキストによって識別される。図１６にわたる略語「ＡＰＰ」は、アプリケーション１４のことを言うと理解される。

【０１４１】

アプリケーション関連ステップ８４は、アプリケーションの計算モジュール及びアプリケーションによりアクセスされる又は産生されるデータセットがどのようにして互いに依存するのかを表す依存性分析と共に、アプリケーション１４のリポジトリに対する確認のステップをやはり含む。これはステップ１００として図１６中に示され、「ＡＰＰ、依存性分析を確認する」というテキストで標示される。

【０１４２】

アプリケーションは、次いで、「工学設計されたデータに対してＡＰＰを実施する」と標示されるステップ１０２で、図１６中に示されるように、工学設計された検査データを操作させられる。

【０１４３】

結果は、「コードカバレッジを報告する」というテキストで標示されるステップ１０４において図１６中に示されるように、コードカバレッジを決定するように点検される。

【０１４４】

これらのカバレージの報告に基づいて、データ駆動型検査用フレームワーク１０は、より良好なコードカバレッジを提供するために、検査データに行うことができる変更の示唆を提供する。これは、「コードカバレッジを増加させる方法を示唆する」というテキストで標示されるステップ１０６において、図１６中に示される。

【０１４５】

ステップ１０６の結果は、追加データを作り出すこと、又はデータのサブセットを既存のデータから抽出するような変更のいずれかによって、データエンジニアリング手順を変更することをもたらしていてもよい。このステップは、ステップ１０８として図１６中に識別され、「データエンジニアリングを変更する」と標示される。

【０１４６】

加えて、出力データの整合性は、出力データをベースライン７４と比較すること、すなわちステップ１１０として図１６に示され「ＡＰＰについての正しい結果を決定する」と標示されるステップによって評価される。

【０１４７】

結果が異なる限り、アプリケーション１４は、「アプリケーションを変更する」というテキストによりマーキングされるステップ９８によって図１６中に示されるように、違いを解消するように変更される。違いがあるかどうかの決定は、参照番号「１１２」によって図１６中に識別され、「結果を予想される結果と比較する」というテキストで標示されるステップ中で実行される。

【０１４８】

いくつかの実施形態では、データ精製器３２は、１又は２以上のサブセットルールにしたがって、プロダクションデータ２６を精製する。サブセットルールは、データ精製器３２に、データレコードのより大きい組から、選択されるべきデータレコードのサブセットを識別させるルールである。結果として得られるデータ精製物３３は、こうして、元のデータよりも少ない量であること、及び検査論理濃度がより高いことの両方となる。これは

、最終的に、より効率的な検査につながる。というのは、アプリケーション１４がデータ精製物３３を操作すると、より少ない量のデータで、より高いコードカバレッジを達成できるからである。

【０１４９】

データ精製器３２が依拠するサブセットルールは、内部的に、データエンジニアリングモジュール１６内から、データ駆動型検査用フレームワーク１０内の他の場所から、又は外部ソースから生じてよい。

【０１５０】

一例では、サブセットルールは、データレコードをプロファイリングし、結果として得られるプロファイルの分析に基づいてサブセットルールを策定するため、論理仕様２８を使用する、論理抽出器３１によって提供される。これらのサブセットルールは、次いでデータ精製器３２に提供され、データ精製器３２は、次いでサブセットルールを使用してデータ精製物３３を作り出す。

【０１５１】

別の例では、サブセットルールは、結果分析モジュール７２からもたらされ、結果分析モジュール７２は、特定の検査データに対してアプリケーション１４を実行したという結果を含む情報に依拠する。データサブセット器１８は、次いで、例えば結果分析モジュール７２からの報告に基づいた、これらの結果の分析に基づくサブセットルールを策定する。これらのルールは、最終的にデータ精製器３２によって実行されて、データ精製物３３を作り出す。

【０１５２】

さらに別の例では、サブセットルールを策定する代わりに、データサブセット器１８が外部ソースからサブセットルールを受け取る。いくつかの場合では、データサブセット器１８は、実際に検査コンピュータ１２に座って、ユーザインターフェースを介してサブセットルールを手動で指定しているユーザから直接サブセットルールを受け取る。他の場合では、データサブセット器１８は、検査コンピュータ１２に、ハードディスクなどの非一時的コンピュータ可読記憶媒体からサブセットルールを読み取らせること、又は検査コンピュータ１２に、インターネットなどのワイドエリアネットワークを含むネットワークなどの、非一時的コンピュータアクセス可能伝送媒体を介してサブセットルールを受け取らせることによって、サブセットルールを獲得する。

【０１５３】

外部から受け取る、又は内部で生成するに関係なく、サブセットルールは、原子的又は分子的のいずれかである。原子的サブセットルールは、さらなるサブセットルールへと分解することはできない。分子的サブセットルールは、２又は３以上の原子的若しくは分子的サブセットルールの組合せからなる。典型的には、ブール演算子が、原子的サブセットルールを連結して、分子的サブセットルールを形成する。

【０１５４】

サブセットルールは、確定的又は確率的のいずれかでもある。確定的サブセットルールの例は、全てのレコードの選択を特定の基準と一致させるルールである。確率的サブセットルールの例は、特定の基準と一致する全てのレコードのうち、それらのレコードの２つがランダムに選択されるものを指定するルールである。

【０１５５】

いくつかの例では、サブセットルールは、１又は２以上のターゲットデータフィールドを指示し、ターゲットデータフィールドについて、各々固有の値又は値の分類がデータ精製物３３中に含まれることを指定する。この例を実装するため、データ精製器３２は、データレコード中のターゲットデータフィールドについて各固有の値を識別し、サブセットルールを満足するデータレコードをのみ有するデータ精製物３３を作り出す。

【０１５６】

例えば、５０の州の各々について固有の値を有する「州」データフィールド、及び２つの固有の値を有する「性別」データフィールドは、ターゲットデータフィールドとして識

別することができる。この場合では、「州」についての５０の値の各々及び「性別」についての２つの値の各々が、データ精製物３３中の少なくとも１つのデータレコードに含まれるように、データ精製器３２は、データ精製物３３のためにデータレコードを選択する。

【０１５７】

いくつかの例では、データサブセット器１８は、データレコードの同じ組内、又はデータレコードの異なる組間のデータレコードの間の関係のタイプを指定するサブセットルールを実装する。これらの例では、データ精製器３２は、サブセットのために選択された他のデータレコードとの、データレコードの関係に基づいてデータレコードを選択する。例えば、データ精製器３２は、データ精製物３３中に含むために、カスタマ識別子データフィールドについて共通の値を共有するデータレコードを選択することができる。

【０１５８】

データサブセット器１８は、フィルタ処理に依拠するサブセットルールを実装することもできる。これらの場合では、データ精製器３２は、データ精製物３３中内に、ある種のターゲットフィールド中に特定の値を有するレコードを含む。例えば、データ精製器３２は、「州」の各値が少なくとも一度は表されるようにレコードを選択することができる。又は、データ精製器３２は、「人口」というフィールドの値を考えて、「州」の値を有するレコードの数がその州に関連する「人口」の値に依存するようにデータレコードを選択することによって、配分方式を適用することができる。

【０１５９】

いくつかの実施形態では、データ分析者又はアプリケーション開発者などのユーザがサブセットルールを提供する。例えば、ユーザは、ターゲットフィールドを識別し、又はデータレコード間の関係を指定し、及びそのような指定をデータサブセット器１８に提供することができる。

【０１６０】

他の例では、データサブセット器１８は、データレコードをプロファイリングし、プロファイルの分析を実行して、好適なデータサブセットルールを識別又は策定する。プロファイリングを実行するため、データサブセット器１８は、関連するデータレコードにアクセスし、データレコードのある種の特徴を分析して、データレコードのプロファイルを生成する。これらの特徴は、以下、すなわち、単一のデータセットの個別のデータレコード、データレコードの組内のデータフィールド間の関係、及びデータレコードの異なる組にわたるデータフィールド間の関係のうちの１又は２以上を含む。

【０１６１】

データレコードの組のプロファイルは、データレコードの組中のデータの概要である。この概要は、フィールド単位で提供することができる。プロファイルは、データレコードの組中のデータを特徴付ける情報を含むことができる。そのような情報の例としては、データレコード中のデータフィールドのうちの１又は２以上の濃度、データフィールドのうちの１又は２以上の中の値の分類、個別のデータレコード中のデータフィールド間の関係、及びデータレコード間の関係が挙げられる。データレコードの組のプロファイルは、「疑似フィールド」を特徴付ける情報も含むことができる。疑似フィールドは、関係するデータレコード中の１又は２以上のデータフィールドから取られる値を操作することにより決定される値でポピュレートされている合成データフィールドである。

【０１６２】

データレコードの生成されたプロファイルに基づいて、データ精製器３１は、アプリケーション１４について良好なコードカバレッジを達成するデータレコードのサブセットの選択に関係するデータレコードの特徴を識別する。例えば、データレコードのプロファイルに基づいて、データ精製器３１は、アプリケーションの入力変数及び導出変数に関する可能性がある、１又は２以上のデータフィールド又はデータフィールドの組合せを識別することができる。いくつかの場合では、サブセットルールは、ユーザから若しくはコンピュータ記憶媒体から受け取った入力に基づいて、及び／又はアプリケーション１４の実行

の結果に基づいて、例えば結果分析モジュール７２から受け取った入力に基づいて策定することもできる。

【０１６３】

データサブセット器１８は、異なる解析法に基づいてサブセットルールを指定することができる。いくつかの実施形態では、データサブセット器１８は、個別のデータレコード内のデータフィールドの分析に基づいて、サブセットルールを指定する。一例では、これは、どのデータフィールドがアプリケーション１４中の変数に関係する可能性があるのかを判定することを含む。別の例では、データサブセット器１８は、フィールドの許容される値の数に基づいてターゲットデータフィールドを識別する。例えば、「性別」データフィールドは、ただ２つの許容される値を有し、ターゲットデータフィールドとして識別することができる。他方、「電話番号」データフィールドは、ターゲットデータフィールドとして識別される可能性がない。

【０１６４】

さらに他の例では、データサブセット器１８は、ターゲットデータフィールドとして、１又は２以上のデータフィールド中のデータの操作の結果得られるデータでポピュレートされた疑似フィールドを識別する。例えば、「所得」データフィールドは、（例えば、高、中、又は低といった）カテゴリへと分類することができ、「所得」データフィールドの分類でポピュレートされた疑似フィールドは、ターゲットデータフィールドとして識別することができる。

【０１６５】

他の例では、データサブセット器１８は、ターゲットデータフィールドと、プロファイル中に示されるような同じレコード内の１又は２以上の他のデータフィールドとの間の関係に基づいて、ターゲットデータフィールドを識別する。例えば、プロファイルは、データフィールド「州」及び「郵便番号」が無関係ではないことを示すことができる。この依存性に基づいて、データサブセット器１８は、それらのデータフィールドのうちのただ１つを可能なターゲットデータフィールドと考えることができる。

【０１６６】

データサブセット器１８は、プロファイル中に示されるような、データレコードの組内の異なるデータレコード間、及び／又はデータレコードの異なる組にわたる関係の分析に基づいて、１又は２以上のサブセットルールを指定することもできる。例えば、プロファイルは、データレコードがデータフィールドの共通の値を介してリンクできることを示すことができる。リンクする値の例は、カスタマＩＤデータフィールドの値となる。

【０１６７】

データサブセット器１８がデータレコードのサブセットを選択し、データ点検ユニット３７がそれらの有効性を確認したら、データエンジニアリングモジュール１６は、データレコードのサブセットを計算環境マネージャ４４に提供し、計算環境マネージャ４４は、最終的に、データレコードのサブセットが、検査されているアプリケーション１４による操作を受けるための準備をする。データエンジニアリングモジュール１６は、データ精製物３３を含むデータレコード又はそれらのデータレコードを示すデータのいずれかを提供する。例えば、データエンジニアリングモジュール１６は、計算環境マネージャ４４に、データ精製物３３を含むデータレコードについての識別子、又はそれらのデータレコードについてのアドレスを提供することができる。データエンジニアリングモジュール１６は、計算環境マネージャ４４に、データレコードの選択されたサブセットを含むファイルを提供することもできる。

【０１６８】

実行後、結果分析モジュール７２は、データ精製物３３に対してアプリケーション１４を実行した結果を示すデータを含むカバレッジ分析報告を生成する。いくつかの実践では、結果分析モジュール７２は、アプリケーション１４がコンパイルされたソースコードの部分が実行したのか実行しなかったのかを識別する情報、又はアプリケーション１４がコンパイルされたソースコードの各部分が何回実行したのかを識別する情報を含むカバレッ

ジ分析報告を生成する。ある種の実践では、結果分析モジュール７２は、アプリケーション１４が実行した又は実行しなかったルールを識別する情報、及びアプリケーション１４が各ルールを実行した回数を識別する情報を含むカバレッジ分析報告を生成する。他の実践では、結果分析モジュール７２は、アプリケーション１４がコンパイルされ、実行した又は実行しなかったソースコードの部分、並びにアプリケーション１４がコンパイルされたソースコードの選択された部分が実行した回数を識別する情報を含むカバレッジ分析報告を生成する。他の実践では、結果分析モジュール７２は、アプリケーション１４がコンパイルされたソースコードの特定の部分を実行する試みに関連して生じるエラーを識別する情報を含むカバレッジ分析報告を生成する。さらに他の実践では、結果分析モジュール７２は、アプリケーション１４がある種のルールを実行することを試みたときに生じるエラーを識別する情報、並びに実行したときにエラーという結果となったルールの識別情報を含むカバレッジ分析報告を生成する。

【０１６９】

いくつかの実践では、結果分析モジュール７２は、実行した又は実行しなかったルールを直接識別するカバレッジ分析報告を生成する。他の実践では、結果分析モジュール７２は、各論理ルールが実行した回数、実行期間のアプリケーションの各変数の値、又は他の情報などの、アプリケーション１４の実行についての追加情報を含むカバレッジ分析報告を生成する。

【０１７０】

他の実践では、実行しなかったアプリケーション中の各論理ルールについて、結果分析モジュール７２は、その論理ルールに関するアプリケーション１４の１又は２以上の変数を識別する。他の実践では、結果分析モジュール７２は、アプリケーション１４を通したデータの流れを示すデータなどの報告に含まれるデータに基づく、又はアプリケーションについてプリロードされた情報に基づく変数をやはり識別する。いくつかの場合では、結果分析モジュール７２は、論理ルールを実行させた各変数についての値又は値の範囲も識別する。識別されると、データエンジニアリングモジュール１６は、入力データフィールド及び変数に対応する値又は値の範囲を使用して、データレコードの更新サブセットの以降の選択に、追加のサブセットルールを指定する。

【０１７１】

例えば、識別された変数がデータレコードのデータフィールドのうちの１つに直接対応するアプリケーションの入力変数である場合、データエンジニアリングモジュール１６は、対応するデータフィールド及びデータフィールドについての値又は値の範囲を識別する。

【０１７２】

例えば、入力変数が何らかの閾値より大きいときにアプリケーション１４中の論理ルールを実行する場合、入力変数が閾値よりも大きい値を有する少なくとも１つのデータレコードを、任意の製造された又は精製されたデータが含むはずであると、データエンジニアリングモジュール１６は決定する。この情報に基づいて、そのルールに対する入力変数が閾値を上回るときにだけ実行する論理ルールの実行を引き起こすのに十分なデータをアプリケーション１４に提供される後続のデータレコードが含むことになるように、データエンジニアリングモジュール１６は、追加のサブセットルールを指定する。

【０１７３】

別の例では、識別された変数は、データレコードのデータフィールドのうちの１つに直接対応しない。そのような変数は、「導出変数」と呼ばれる。導出変数の場合には、データエンジニアリングモジュール１６は、アプリケーション１４の論理を通して、導出変数の導出過程をたどるため、データ系列を分析する。このデータ系列分析によって、識別された変数が導出された特定の１又は２以上の入力変数を識別することが可能になる。データエンジニアリングモジュール１６は、次いで、データフィールドについての、対応する１又は２以上のデータフィールド及び値又は値の範囲を識別する。

【０１７４】

例えば、導出変数の値が特定の値に等しいときにアプリケーション１４中の論理ルールが実行する場合、データエンジニアリングモジュール１６は、データ系列分析のための命令を実行して、導出変数が３つの入力変数の論理的な組合せから導出されることを決定する。導出変数の論理的導出過程に従うことによって、データエンジニアリングモジュール１６は、これらの３つの入力変数の要求された値が、特定の導出変数を実現すると決定する。

【０１７５】

導出変数の所望の値を産出するのに必要な値の決定は、データサブセット器１８に提供されて、導出変数に所望の値を獲得させ、したがって関連する論理ルールの実行をトリガするのに十分なデータをデータ精製物３３が含むように、データサブセット器１８が追加のサブセットルールを指定する。

【０１７６】

いくつかの例では、カバレッジ分析の結果は、ユーザにやはり提供される。これに応じて、ユーザは、データサブセット器１８に追加のサブセットルールを提供することができ、又は以前に提供されたサブセットルールを変更することができる。

【０１７７】

いくつかの論理ルールは、非常にまれにトリガされるので、データレコードの完全な組でさえも、その論理ルールを実装するコードをアプリケーション１４に実行させるのに十分なデータを、単に偶然に含む可能性はない。完全なデータセット中のそのような不備を識別するために、アプリケーション１４を、データレコードの全てを入力として使用して、１回又は２回以上実行することができる。結果として得られる報告は、入力のために選択されるデータレコードのサブセットにかかわらずカバーすることができないルールを識別する。この不備に対処するために、データ駆動型検査用フレームワーク１０は、肯定データ製造器２２及び／又は否定データ製造器２４を使用して必要なデータを製造する。

【０１７８】

いくつかの実施形態では、データエンジニアリングモジュール１６は、フィルタ処理によってデータサブセット化を実行する。フィルタ処理は、肯定的又は否定的であってよい。肯定的フィルタ処理では、空の組で開始して、いくつかの条件を満足するデータレコードだけを追加する。否定的フィルタ処理では、不足のないデータセットで開始して、いくつかの条件を満足するデータレコードを削除することによって、データセットを少しずつ削り落とす。

【０１７９】

他の実施形態では、データエンジニアリングモジュール１６は、各ターゲットデータフィールドについて、各々の許容される値が少なくとも１回現れる、又は指定の回数現れるように、ターゲットデータフィールドを識別すること、各々のそのようなフィールドの肯定的な値を決定すること、及びデータレコードを選択することによって、データのサブセット化を実行する。

【０１８０】

さらに他の実施形態では、データエンジニアリングモジュール１６は、データ分類によってデータのサブセット化を実行する。これは、ターゲットデータフィールドを識別する方法と同様であるが、値の範囲が実際のターゲット値を置き換える。こうして、ターゲットデータフィールドがリスク評価に使用されるコレステロールレベルを表す場合、範囲を使用して、低、中、及び高収入を表す値域を規定することができる。その場合、データレコードは、各値域又は分類が同じ所定の数のレコードを有するように選択されることになる。

【０１８１】

さらなる実施形態では、データエンジニアリングモジュール１６は、値の組合せに依拠することによってデータのサブセット化を実行する。これは、２つの許容される値（例えば、性別）を有する第１のフィールドと１２の許容される値（例えば、誕生月）を有する第２のフィールドといった、２つのターゲットデータフィールドを考えることによって理

解することができる。各々の可能な値が少なくとも１回確実に存在することをのみ望む場合、この要件は、単に１２のレコードで満足することができる。しかし、これら２つのフィールドの全ての可能な組合せを有することを希望する可能性があることが考えられる。その場合、少なくとも２４のレコードを選択する必要があることになる。

【０１８２】

上の方法、並びにデータサブセット器１４によって実装できるさらなる方法のさらなる詳細は、既に参照により組み込まれている「DATA RECORDS SELECTION」という題名の特許公開中に見いだすことができる。

【０１８３】

データエンジニアリングモジュール１６は、肯定データ製造器２２、否定データ製造器２４、及びデータ補強器２０を使用して、既に参照により組み込まれている「DATA GENERATION」という題名の出願中に記載された原理にしたがって動作する。

【０１８４】

データエンジニアリングモジュール１６は、ユーザが指定することができる、特定のタイプのデータを生成する。例示的なデータタイプとしては、文字列、１０進整数、日付、及び時間が挙げられる。データエンジニアリングモジュール１６は、製造される１０進又は整数データについて許容される値の範囲、製造される文字列データについて平均文字列長、製造されるデータ中で使用することができる値又は文字の組、及び他の特性などといった、製造されるデータへの制限を課する。データエンジニアリングモジュール１６は、既存のソースレコードの１又は２以上のフィールド中の値を変更すること、レコード中に新しいフィールドを作り出すこと及びポピュレートすることによってソースレコードを補強すること、又は完全に新しいレコードを作り出すことによってデータを製造することができる。いくつかの例では、ユーザが、ユーザインターフェースを通して、設定可能なオプションを指定する。

【０１８５】

データエンジニアリングモジュール１６は、肯定データ製造器２２を使用してアプリケーション１４によって処理するためのデータを製造する。データエンジニアリングモジュール１６は、データ補強器２０を使用して、プロダクションデータ２６などの既存のデータを変更又は補強することもできる。例えば、データ補強器２０は、プロダクションデータ２６から取られた１若しくは２以上のフィールドについて値を変更することができ、又は１若しくは２以上の新しいフィールドを作り出してポピュレートし、それらをプロダクションデータ２６中の既存のデータレコードに追加することができる。肯定データ製造器２２を使用して、データエンジニアリングモジュール１６は、完全に新しいデータレコードを製造することもできる。いくつかの実施形態では、これらの新しいレコードのフォーマットは、プロダクションデータ２６に基づき、一方他の実施形態では、ユーザなどの外部エージェントが、サブセットルールを指定することに関連して上で議論したのと同じ方法を使用して、フォーマットを指定することになる。

【０１８６】

データエンジニアリングモジュール１６は、ターゲット中に記憶されるデータを製造する。いくつかの例では、データエンジニアリングモジュール１６は、プロダクションデータ２６に基づいてデータを製造する。他の例では、データエンジニアリングモジュール１６は、最初からデータを製造する。本明細書で使用する、「最初から（from scratch）」製造するとは、指定される特性にしたがうが、既存のデータに基づかずに製造することを意味する。

【０１８７】

プロダクションデータは、ファイル、データベース、パラメータセット、又は別のデータのソースであってよい。プロダクションデータ２６は、各々がデータの１又は２以上のフィールドを有する、１又は２以上のレコードを含むことができる。例えば、プロダクションデータ２６は、小売店のカスタマについてのカスタマレコードを記憶するカスタマデータベースであってよい。そのようなデータベース中の各レコードは、個別のカスタマを

表す。各レコードは、複数のフィールドを有することができる。プロダクションデータ２６は、フィールドの数、各フィールド中のデータのタイプ、及び値の許容範囲、最大許容値、又は許容される文字のリストなどの各フィールド中のデータの特性などのレコードのフォーマットを指定するレコードフォーマットを有することができる。いくつかの例では、データエンジニアリングモジュール１６は、最初からデータを生成する。そのような場合、データソースは提供されない。

【０１８８】

データエンジニアリングモジュール１６は、データベース、ファイル、又は他のデータ構造中に記憶することができる、設定データに基づいてデータを製造する。設定データは、使用されるデータ生成手法、内容生成モード、製造されるデータのデータタイプ、製造されるデータについての内容基準、及び製造されるデータについての他の設定情報を指定することができる。

【０１８９】

いくつかの場合では、データエンジニアリングモジュール１６がデータを製造するために使用する設定データの一部又は全部を、検査コンピュータ１２上で利用可能なユーザインターフェースを通してユーザが指定する。他の例では、データエンジニアリングモジュール１６が設定データの一部又は全部を決定する。これらの場合では、データエンジニアリングモジュール１６は、プロダクションデータの分析に基づいて、又はターゲットの所望の性質についての情報に基づいて、決定する。

【０１９０】

いくつかの例では、データエンジニアリングモジュール１６は、データ補強器２０を使用して、設定データにしたがってプロダクションデータ２６中の既存のソースレコードのフィールドの１又は２以上についての値を変更すること、及びターゲット中に変更したレコードを記憶することによって、ターゲットについてのデータを製造する。他の例では、データエンジニアリングモジュール１６は、データ補強器２０を使用して、所与のフィールドについて値の全てを変更する。例えば、レコードの全てにわたる所与のフィールド中の値の分散が、設定データにより示されるようなターゲット分散と一致するように、各レコードについて、所与のフィールドに対して値を割り当てることができる。ユーザ又は設定データのいずれかが、このターゲット分散を指定する、又はこのターゲット分散を指定するための情報を提供する。

【０１９１】

いくつかの場合では、データエンジニアリングモジュール１６は、所与のフィールドについての値の全部ではなく一部を変更する。これらの場合の中には、データエンジニアリングモジュール１６が、設定データにより示されるような指定された基準に一致しない値をのみ変更する場合がある。そのような場合の例は、データエンジニアリングモジュール１６が、所与のフィールドについて許容される値の特定の範囲外にある、所与のフィールドについての任意の値を変更する場合である。

【０１９２】

いくつかの例では、データエンジニアリングモジュール１６は、プロダクションデータ２６の既存のソースレコードを、設定データにしたがって、１又は２以上の新しいフィールドで補強するためにデータ補強器２０を使用すること、及びターゲット中にこれらの補強されたレコードを記憶することによってデータを製造する。設定データは、新しいフィールドの数、新しいフィールドについてのデータタイプ及び値、並びに新しいフィールドの他の特性を決定するための命令を提供する。

【０１９３】

他の例では、データエンジニアリングモジュール１６は、設定データにより提供される情報を使用してデータを製造する。情報は、新しいフィールドについての値が、プロダクションデータ中の既存のフィールドについてのデータに基づいて製造されることを指定する。あるいは、この情報は、新しいフィールドについての値が、任意の既存のソースデータには基づかないが、代わりに設定データによって指定されるある種の特性にしたがって

製造されることを指定する。

【０１９４】

いくつかの例では、データエンジニアリングモジュール１６は、プロダクションデータ２６の既存のソースレコードを、設定データにしたがって、１又は２以上の新しいレコードで補強するためにデータ補強器２０を使用すること、及びターゲット中に補強されたレコード（すなわち、既存のソースレコード及び新しいレコードの両方）を記憶することによってデータを製造する。いくつかの実施形態では、新しいレコードは、ソースレコードと同じレコードフォーマットを有する。

【０１９５】

他の例では、設定データは、以下、すなわち、新しいレコードの数、新しいレコードのフィールドについての値、及び新しいレコードの他の特性のうちの１又は２以上の任意の組合せを決定するための命令を提供する。これらの例の中には、新しいレコード中の１又は２以上のフィールドについての値が最初から製造されることを設定データが指定する例がある。

【０１９６】

いくつかの他の例では、設定データは、プロファイルを指定し、新しいレコード中の１又は２以上のフィールドについての値が、そのプロファイルを満足するように製造されることを要求する。１つのそのような例では、プロファイルは、レコードの全ての中の特定のフィールドについての値が、指定された特性を集合的に満足することを指定する。特性の例は、値が特定の平均値又は特定の分散値を有することである。例えば、カスタマデータベースソースにおいて、レコードの全てにわたる「年齢」フィールドについての値が特定の平均を有するポアソン分布を満足するようにレコードが製造されることを、設定データが要求する場合がある。

【０１９７】

いくつかの例では、設定データは、データエンジニアリングモジュール１６が、データ生成に対して２以上の手法を施すことを要求する。１つのそのような例では、データエンジニアリングモジュール１６は、以下の手法、すなわち、１又は２以上のフィールドについての値を変更すること、ソースレコードを１又は２以上の新しいフィールドで補強すること、及びソースレコードを１又は２以上の新しいレコードで補強することの任意の組合せを施す。

【０１９８】

いくつかの例では、ターゲットは、製造されたレコードだけを記憶する。他の例では、ユーザがソースを指定し、データエンジニアリングモジュール１６が、特性に基づいてレコードを製造する。好適な特性の例は、ソースのレコードフォーマット、又はソースの１若しくは２以上のフィールドのプロファイルである。

【０１９９】

他の例では、ソースが指定されない。そのような例では、データエンジニアリングモジュール１６は、設定データにしたがって、自動的に最初からレコードを製造する。

【０２００】

いくつかの例では、ソースのレコードフォーマットがターゲットにマッピングされる。１つのそのような例では、設定データは、ソースのレコードフォーマットがターゲットによって採用されることを示す。別のそのような例では、設定データは、ソースのレコードフォーマットがターゲットに適用され、ソースのレコードフォーマットにしたがって、新しいレコードがデータエンジニアリングモジュール１６によって最初から製造されることを必要とする。他のそのような例では、データエンジニアリングモジュール１６が複数のソースに依拠し、各ソースのレコードフォーマットが、部分的又は完全にターゲットにマッピングされる。少なくとも１つのそのような例では、各ソースからの対象のフィールドのフォーマットが、ターゲットにマッピングされる。

【０２０１】

いくつかの例では、データエンジニアリングモジュール１６は、ソースのレコードフォ

ーマットをターゲットにマッピングし、それを変更する。これらの例の中には、設定データがデータエンジニアリングモジュール１６にフィールドの名前を変えさせる例、及び設定データがソースからフィールドを除去させる例がある。

【０２０２】

データエンジニアリングモジュール１６は、検査コンピュータ１２上で、ユーザがデータソースを識別することを可能にするソースウィンドウを有するユーザインターフェースを提供する。ソースウィンドウは、ファイル又はデータベースなどのソースタイプ、及びソースへ又はデータベースソースについての設定ファイルへのパスなどのソースの識別子をユーザが指定することを可能にするソースタイプメニューを含む。いくつかの例では、ソースがデータベースであるとき、ユーザは、データベースからソースデータを獲得するために使用されるクエリ（例えば、ＳＱＬクエリ）を指定する。ソースウィンドウは、データエンジニアリングモジュール１６が新しいレコードを製造するべきであるのか、またその場合、どれだけ多く製造するかをユーザに示すことができるオプションを提供する。ソースウィンドウによって、ユーザが、ソースについての他の情報を閲覧する又は指定することが可能になる。例えば、ユーザは、ソースのレコードフォーマットを閲覧すること、ソースのレコードフォーマットを規定するファイルを指定すること、ソースデータを閲覧すること、又はソースデータのプロファイルを閲覧することができる。

【０２０３】

いくつかの例では、ユーザインターフェースのソースウィンドウによって、ユーザが、ソースを指定することなく、データエンジニアリングモジュール１６にデータを製造させることが可能になる。特に、ソースウィンドウによって、ユーザが、ソースタイプメニュー中のソースタイプとして製造したデータを選択することが可能になる。ソースタイプとして製造データを選択することで、ユーザインターフェース中のデータ生成ウィンドウが表示される。データ生成ウィンドウによって、ユーザが、データを製造するために使用する方法を示すこと、及び製造される新しいレコードの数を示すことが可能になる。

【０２０４】

ユーザインターフェースは、ユーザがターゲットを識別することを可能にするターゲットウィンドウをやはり提供する。ターゲットウィンドウ中のターゲットタイプメニューによって、ユーザが、ターゲットのタイプを指定することが可能になる。ターゲットの例としては、ファイル又はデータベースが挙げられる。ターゲットウィンドウによって、ユーザが、ターゲットの識別子（例えば、ターゲットファイルへのパス、又はターゲットデータベースについての設定ファイルへのパス）を指定することがやはり可能になる。ターゲットウィンドウは、ソース及びターゲットが識別されると、データ生成のための様々な設定可能なオプションへのアクセスをユーザに提供するランボタンを提供する。

【０２０５】

データエンジニアリングモジュール１６は、データを製造するためのいくつかの手法を提供する。これらは、フィールド変更、フィールド作成、レコード作成、既存ソースの使用、親データセットの使用を含む。利用可能な手法にアクセスするために、ユーザは、ユーザインターフェースのデータ生成ウィンドウに依拠する。

【０２０６】

フィールド変更手法では、データエンジニアリングモジュール１６は、ソースレコードの１又は２以上のフィールドについての値を変更する。いくつかの場合では、データエンジニアリングモジュール１６は、所与のフィールドについての値の全てを変更する。いくつかの場合では、データエンジニアリングモジュール１６は、レコードの全てにわたる所与のフィールド中の値の分散がターゲット分散と一致するように、フィールドの値を変更する。別の例では、データエンジニアリングモジュール１６は、所与のフィールドについての値の全部ではなく一部を変更する。これらの例の中には、データエンジニアリングモジュール１６が指定された基準に一致しない値をのみ変更する例がある。例えば、特定のフィールドについて許容される値の特定の範囲の外にある任意の値を変更することができる。

【０２０７】
フィールド作成手法では、データエンジニアリングモジュール１６は、既存のレコードについて、１又は２以上の新しいフィールドを作り出す。いくつかの例では、データエンジニアリングモジュール１６は、ソースデータ中の既存のフィールドのためのデータに基づいて、新しいフィールドのための値を製造する。他の例では、データエンジニアリングモジュール１６は、新しいフィールドのための値を最初から製造する。
【０２０８】
レコード作成手法では、データエンジニアリングモジュール１６は、新しいレコードを製造する。ユーザは、新しいレコードの数及びそのフォーマットのうちの少なくとも１つを指定する。例えば、ターゲットが、既存のソースレコード及び新しく製造されたレコードの両方でポピュレートされる場合、新しいレコードのレコードフォーマットは、ソースレコードのレコードフォーマットと同じである。ターゲットが新しく製造されたレコードのみでポピュレートされる場合、製造されたレコードに適用されるレコードフォーマットをユーザが指定する。レコードフォーマットとしては、フィールドの数、各フィールドについてのデータのタイプ、例えば、最大値、最小値、許容される文字の組、及び他の特性といった各フィールドについてのデータの特性、並びにレコードフォーマットの他の特徴が挙げられる。
【０２０９】
既存データセット手法では、データエンジニアリングモジュール１６は、既存のソースレコード中の各々のキーとなる値について指定された数の新しいレコードを製造する。キーとなる値は、既存のソースレコード中の対象のフィールド中の値である。
【０２１０】
一例では、補助ソースが、ターゲットレコードのある種のフィールドをポピュレートするために使用されるデータを含む。しかし、補助ソースは、ソース又はターゲットのいずれかのレコードフォーマットと一致するレコードフォーマットを有さない。この場合には、データエンジニアリングモジュール１６は、補助ソースからターゲットレコードに、１又は２以上の対象のフィールドをマッピングする。親データセット手法では、ソースは、階層中の親データセットである。この場合には、データエンジニアリングモジュール１６は、親データセットに関係する子データセットを製造する。親データセット手法の一例では、ソースとして機能する親データセットは、カスタマレコードの組であり、ターゲットとして機能する子データセットは、各カスタマについての１又は２以上のトランザクションレコードの組である。キーフィールドは、子データセット中のレコードを親セット中の対応するレコードとリンクする。例えば、「カスタマＩＤ」フィールドが、カスタマレコードとトランザクションレコードをリンクするキーフィールドであってよい。いくつかの場合では、データエンジニアリングモジュール１６は、どれだけ多くの子レコードを製造するのかについての仕様を受け取る。他の場合では、データエンジニアリングモジュール１６は、子レコードを製造するのに使用されないことになる親レコードの百分率の仕様を受け取る。さらに他の場合では、データエンジニアリングモジュール１６は、子レコードのためのレコードフォーマットの仕様を受け取る。
【０２１１】
いくつかの例では、データエンジニアリングモジュール１６は、フォーマット仕様にしたがってデータを製造する。フォーマット仕様は、製造されることになるデータのフォーマットを指定する。一例では、フォーマット仕様は、製造されることになるデータのデータタイプを示す。
【０２１２】
他の例では、データエンジニアリングモジュール１６は、内容基準にしたがってデータを製造する。内容基準は、製造されることになるデータの特性を限定する。内容基準の例としては値の許容範囲、最大許容値、及び許容される文字のリストが挙げられる。
【０２１３】
いくつかの場合では、ターゲットレコードのレコードフォーマットが、フォーマット仕

様及び内容基準を指定する。他の例では、フォーマット仕様又はフィールドについての内容基準などの、フィールドの特性をユーザが指定することを可能にする、フィールドウィンドウをユーザインターフェースが提供する。

【０２１４】

ユーザインターフェースは、ユーザがターゲットレコードフォーマットを編集することを可能にするためのレコードフォーマットウィンドウをさらに含む。これは、ターゲットの１又は２以上のフィールドについて、データ特性を編集することを含むことになる。レコードフォーマットウィンドウは、ターゲットレコードフォーマット中にあるデータのリストを表示する。このフィールドリストは、各フィールドについてのデータタイプも示す。いくつかの例では、ターゲットレコードフォーマット中にあるフィールドは、ソースレコードフォーマット中にやはり現れる。ターゲットレコードフォーマット及びソースレコードフォーマットの両方に現れるこれらのフィールドは、フィールドリスト中にマーキングされていてもよい。いくつかの例では、マーキングされないフィールドは、ターゲットレコードフォーマット中にのみ現れる。他の例では、ソースレコードフォーマット中に現れるがターゲットレコードフォーマット中に現れないフィールドは、フィールドリストにはない。

【０２１５】

レコードフォーマットウィンドウによって、データエンジニアリングモジュール１６にデータ生成特性を通信するために、ユーザが、ターゲットレコードフォーマットの１又は２以上のフィールドを選択することが可能になる。何が選択されているのかをユーザが把握するのを助けるために、ユーザインターフェースは、ターゲットレコードフォーマットの選択されたフィールドの選択リストを含む。選択リスト中に列挙されるフィールドは、データ生成特性を指定することをユーザが意図するターゲットレコードフォーマットのフィールドである。

【０２１６】

いくつかの例では、選択リストは、ターゲットレコードフォーマット中のフィールドの全てのフィールドリストのサブセットである。これは、ターゲットレコードフォーマットのフィールドの単に一部について、データ生成特性を指定することをユーザが意図する場合に生じる。

【０２１７】

ユーザインターフェースによって、選択リスト中に表示される選択されたフィールドの各々について、レコードフォーマットをユーザが編集することが可能になる。例えば、選択されたフィールドの各々について、フィールドについてのデータタイプを指示すること、フィールドに対して内容生成モードを割り当てること、及びフィールドについてデータ特性を指定することの任意の組合せを、ユーザが実施することができる。ユーザインターフェースは、選択されたフィールドの各々について、データタイプウィンドウ、内容生成ウィンドウ、及びデータ特性ウィンドウのうちの１又は２以上を順に表示する。これらのウィンドウによって、選択されたフィールドの各々について様々な特徴をユーザが指定することが可能になる。

【０２１８】

上に記載されたデータ駆動型検査用フレームワーク１０は、例えば、好適なソフトウェア命令を実行するプログラム可能なコンピューティングシステムを使用して実装することができ、又はフィールドプログラマブルゲートアレイ（ＦＰＧＡ，field-programmable gate array）などの好適なハードウェア、又は何らかのハイブリッド形式で実装することができる。例えば、プログラム手法では、ソフトウェアは、各々が少なくとも１つのプロセッサ、（揮発性及び／若しくは不揮発性メモリ並びに／又は記憶素子を含む）少なくとも１つのデータ記憶システム、（少なくとも１つの入力デバイス又はポートを使用して入力を受け取るため、及び少なくとも１つの出力デバイス又はポートを使用して出力を提供するための）少なくとも１つのユーザインターフェースを含む、（分散型、クライアント／サーバ、又はグリッドなどの様々なアーキテクチャであってよい）１若しくは２以上の

プログラムされた、又はプログラム可能なコンピューティングシステム上で実行する、１若しくは２以上のコンピュータプログラム中の手順を含むことができる。ソフトウェアは、例えば、データフロー図の設計、設定、及び実行に関するサービスを提供する、より大きいプログラムの１又は２以上のモジュールを含むことができる。プログラムのモジュール（例えば、データフロー図の要素）は、データリポジトリ中に記憶されるデータモデルと一致する、データ構造又は他の組織化されたデータとして実装することができる。

【０２１９】

ソフトウェアは、揮発性若しくは不揮発性記憶媒体、又は任意の他の非一時的媒体中に、（例えば、表面のピットとランド、磁気領域、又は電荷といった）媒体の物理的性質を使用して具現化されるような非一時的な形態で、（例えば、ダイナミックＲＡＭなどの動的記憶デバイスのリフレッシュ期間の間の時間といった）期間の間、記憶することができる。命令をロードするための準備において、ソフトウェアは、ＣＤ－ＲＯＭ若しくは（例えば、汎用若しくは専用コンピューティングシステム若しくはデバイスによって可読である）他のコンピュータ可読媒体などの、有形の非一時的媒体上で提供することができ、又は命令が実行されるコンピューティングシステムの有形の非一時的媒体に、ネットワークの通信媒体を介して（例えば、伝播信号中にエンコードされて）送達することができる。処理の一部又は全部を、専用コンピュータ、又は、コプロセッサ若しくはフィールドプログラマブルゲートアレイ（ＦＰＧＡ）若しくは専用の特定用途向け集積回路（ＡＳＩＣ，application-specific integrated circuit）などの専用ハードウェアを使用して実施することができる。処理は、ソフトウェアにより指定される計算の異なる部分が異なる計算要素によって実施される、分散的に実装することができる。各々のそのようなコンピュータプログラムは、好ましくは、本明細書に記載された処理を実施するためにコンピュータが記憶デバイス媒体を読むときに、コンピュータを設定及び動作させるため、汎用若しくは専用プログラム可能コンピュータによりアクセス可能な記憶デバイスの、（例えば、固体メモリ若しくは媒体、又は磁気若しくは光媒体といった）コンピュータ可読記憶媒体に記憶又はダウンロードされる。本発明のシステムは、コンピュータプログラムで設定される、有形の非一時的媒体として実装されると考えることもでき、ここで、そのように設定される媒体は、コンピュータを、特定の予め規定したように動作させて、本明細書に記載された処理ステップのうちの１又は２以上を実施させる。

【０２２０】

本発明のいくつかの実施形態が記載されてきた。それにもかかわらず、上記の記載は、本発明の範囲を説明することを意図しており、本発明の範囲を限定することは意図しておらず、本発明の範囲は、以下の請求項の範囲によって規定されることを理解されたい。したがって、他の実施形態は、以下の請求項の範囲内にやはり入る。例えば、本発明の範囲から逸脱することなく、様々な変形形態を行うことができる。加えて、上に記載したステップのいくつかは、順序に依存しない場合があり、したがって、記載されたものと異なる順序で実施することができる。

【０２２１】

本発明、及びその好ましい実施形態を記載してきたが、新規であると請求され、特許証により保証されるものは、以下である。

【図１】

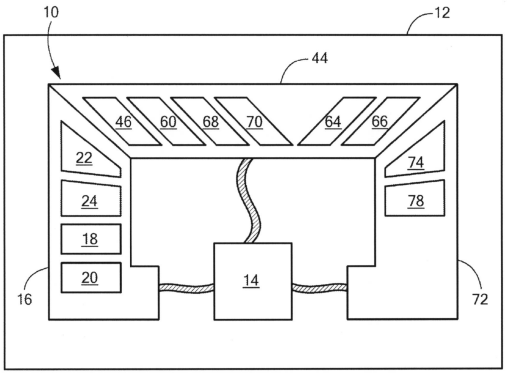


FIG. 1

【図２】

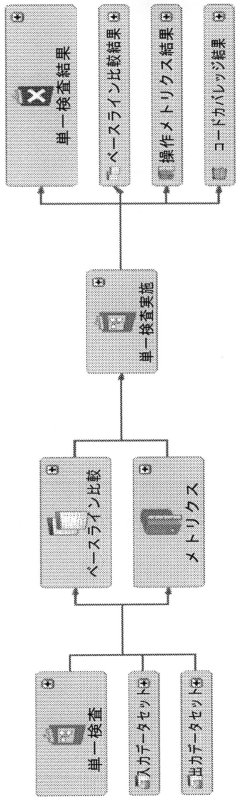


FIG. 2

【図３】

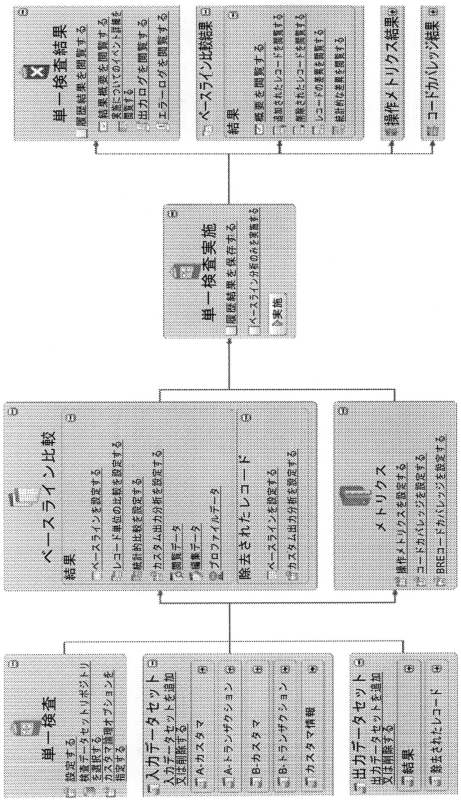


FIG. 3

【図４】

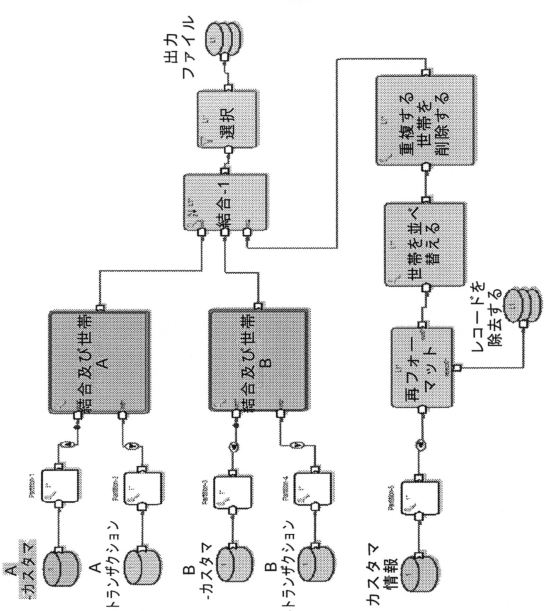


FIG. 4

【図５】



FIG. 5

【図６】

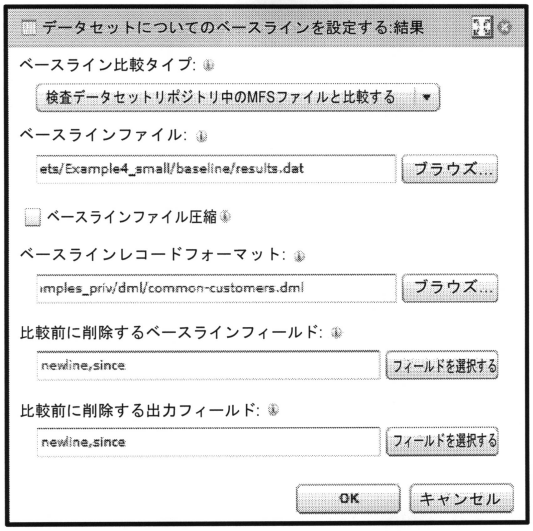


FIG. 6

【図７】

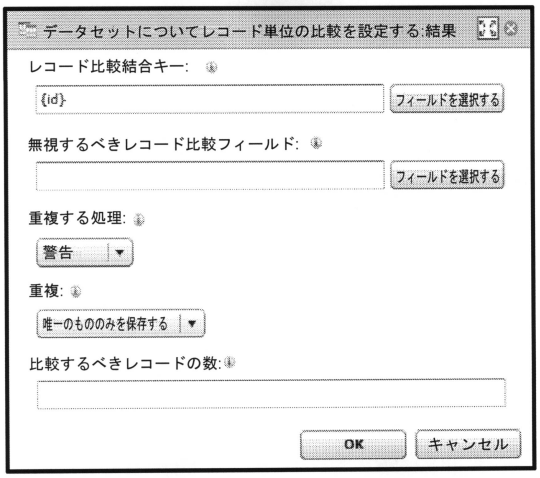


FIG. 7

【図８】

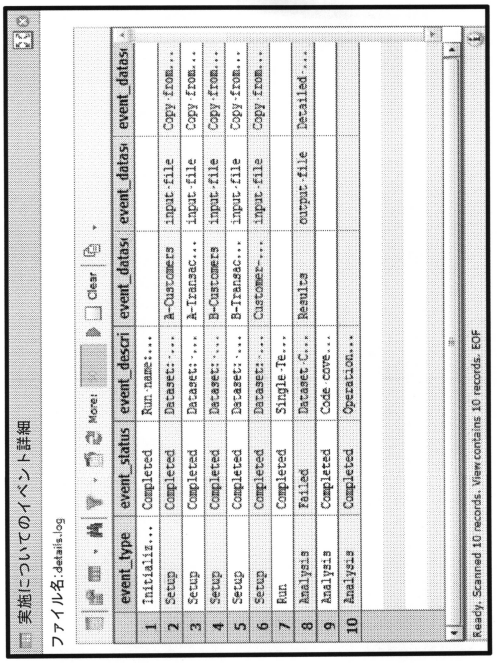


FIG. 8

【図９】

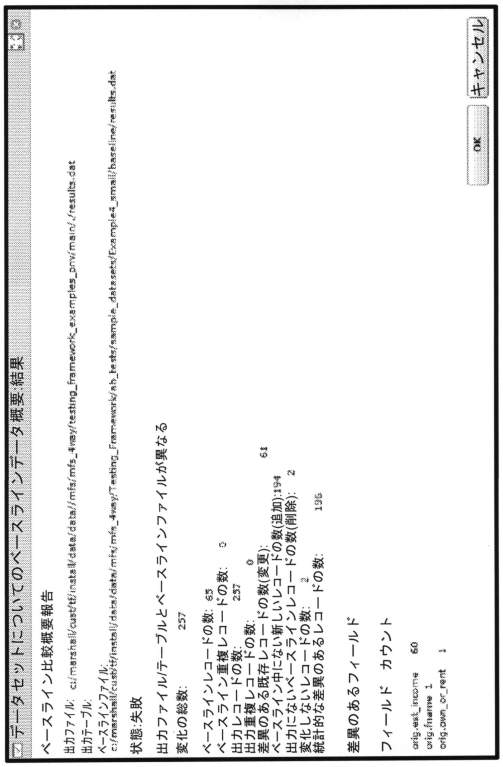


FIG. 9

【図１０】

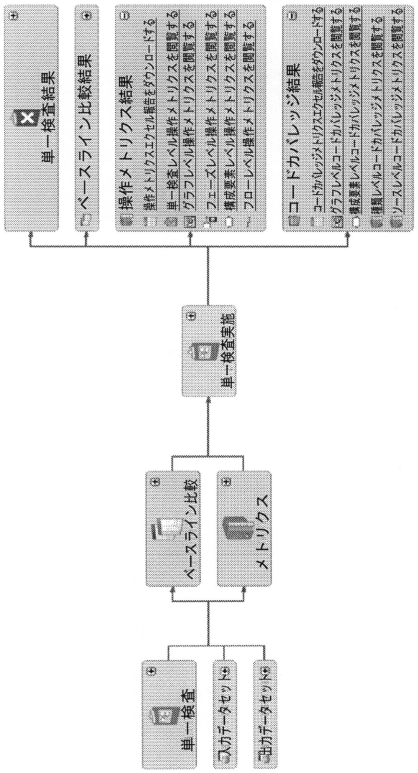


FIG. 10

【図１１】

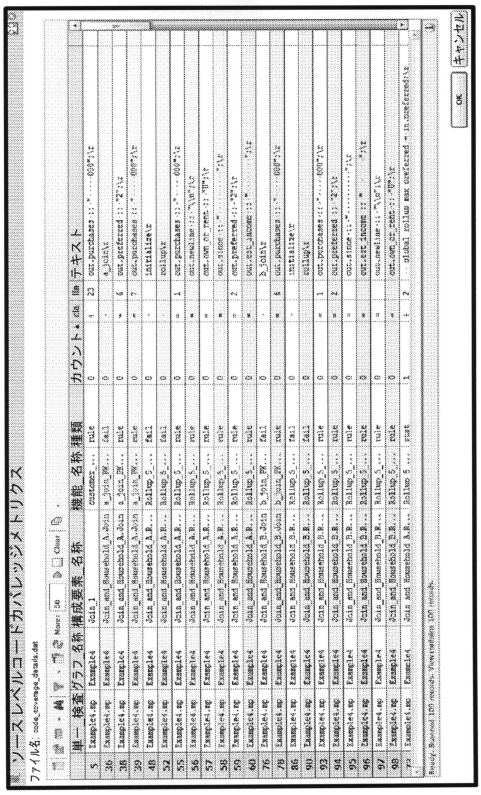


FIG. 11

【図１２】

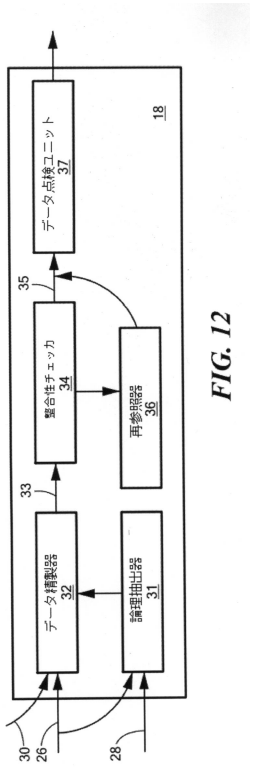


FIG. 12

【図１３】

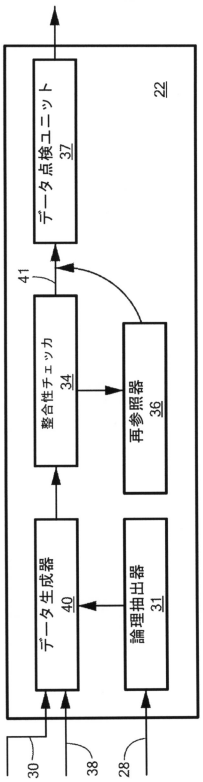


FIG. 13

【図１４】

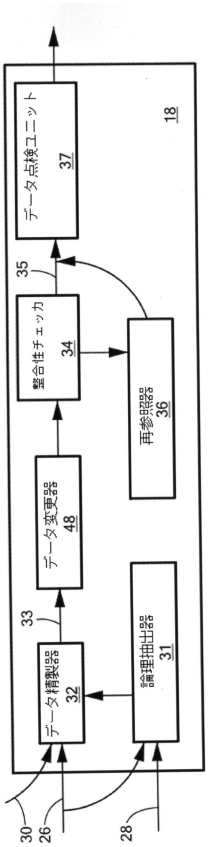


FIG. 14

【図１５】

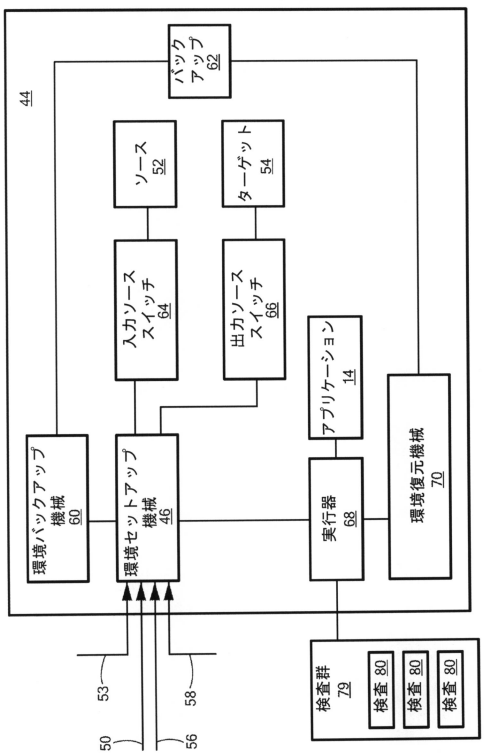


FIG. 15

【図１６】

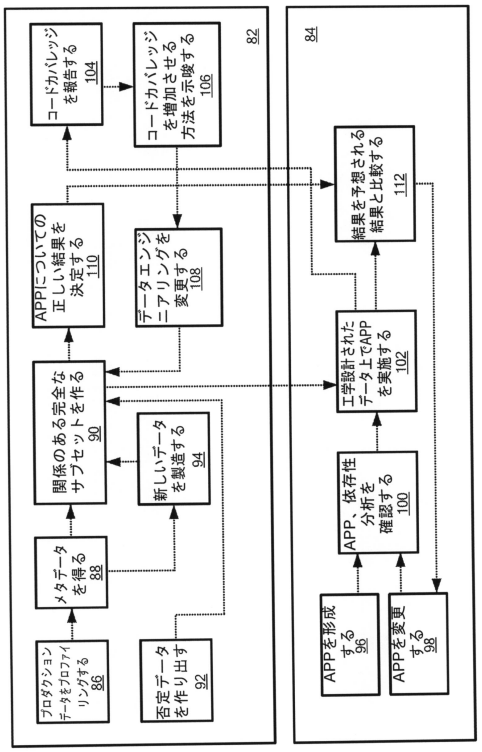


FIG. 16

フロントページの続き

(74)代理人 100150902
弁理士 山内 正子
(74)代理人 100141391
弁理士 園元 修一
(74)代理人 100198074
弁理士 山村 昭裕
(74)代理人 100145920
弁理士 森川 聡
(74)代理人 100096013
弁理士 富田 博行
(72)発明者 プリンツ フィリップ
アメリカ国 ０１８２１ マサチューセッツ ビレリカ コンコードロード３１３
(72)発明者 イスマン マーシャル アラン
アメリカ国 ０２４５８ マサチューセッツ ニュートン ヴァリースプリングロード１１

審査官 塚田 肇

(56)参考文献 米国特許出願公開第２０１４／０２２２７５２（ＵＳ，Ａ１）
特開２００６－２５２１５８（ＪＰ，Ａ）
特開２０１２－１９００６３（ＪＰ，Ａ）
特開２００９－２０５２３９（ＪＰ，Ａ）
特開２０１０－２５０６６９（ＪＰ，Ａ）
特開２０１３－１０５３５８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １１／３６