

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5528640号
(P5528640)

(45) 発行日 平成26年6月25日(2014.6.25)

(24) 登録日 平成26年4月25日(2014.4.25)

(51) Int.Cl.

F I

G 0 6 F 11/28 (2006.01)

G 0 6 F 11/28 3 4 0 A

請求項の数 12 (全 9 頁)

(21) 出願番号	特願2013-538617 (P2013-538617)	(73) 特許権者	512297583
(86) (22) 出願日	平成23年5月9日(2011.5.9)		ヒョンダイ モーター カンパニー
(65) 公表番号	特表2013-546077 (P2013-546077A)		大韓民国, ソウル 1 3 7 - 9 3 8, ソチ
(43) 公表日	平成25年12月26日(2013.12.26)		ョーグ, ヤンゼードン 2 3 1
(86) 国際出願番号	PCT/KR2011/003436	(73) 特許権者	512297594
(87) 国際公開番号	W02012/153879		キアモータース コーポレーション
(87) 国際公開日	平成24年11月15日(2012.11.15)		大韓民国, ソウル 1 3 7 - 9 3 8, ソチ
審査請求日	平成25年5月9日(2013.5.9)		ョーグ, ヤンゼードン 2 3 1
		(73) 特許権者	510054603
			イファ ユニバーシティーインダストリ
			ー コラボレーション ファウンダーショ
			ン
			大韓民国, ソウル 1 2 0 - 7 5 0, ソデ
			ムング, デヒョンードン, 1 1 - 1, イ
			ファ ユニバーシティ
			最終頁に続く

(54) 【発明の名称】 例外処理テスト装置及び方法

(57) 【特許請求の範囲】

【請求項 1】

デバイスドライバー、デバイスマネージャー及びアプリケーションを含むシステムに対する例外処理テスト装置において、

欠陥モデル及びデバイスマネージャーから獲得した情報に基づき変形されたデバイスドライバーを生成する生成モジュール、

前記変形されたデバイスドライバーを利用して前記デバイスドライバーをフッキングするフッキングモジュール、

前記アプリケーションが動作する間、フッキングされ前記変形されたデバイスドライバーから前記アプリケーションに返還されるテスト情報を収集するスキャンングモジュール、及び

前記スキャンングモジュールにより収集されたテスト情報を分析する分析モジュールを含むことを特徴とする例外処理テスト装置。

【請求項 2】

前記欠陥モデルは、アプリケーションとデバイスとの間の連結、デバイスオープン、連結解除、デバイス閉鎖、ヌル(NULL)データ、データ誤謬、デバイス電源、及び処理遅延のうちの少なくとも一つであることを特徴とする請求項 1 に記載の例外処理テスト装置。

【請求項 3】

前記テスト情報は、前記変形されたデバイスドライバーのオペレーション返還値、前記

10

20

デバイスマネージャーで発生する例外コード値、前記アプリケーションの返還値、及びシステム異常コードのうちの少なくとも一つであることを特徴とする請求項 1 に記載の例外処理テスト装置。

【請求項 4】

前記テスト情報が欠陥の発生を意味する場合、前記アプリケーションは前記発生した欠陥に従い例外処理を行うことを特徴とする請求項 1 に記載の例外処理テスト装置。

【請求項 5】

前記例外処理が正常に行われない場合、前記分析モジュールは前記テスト情報を全体システムダウン、一部プロセスダウン、一部プロセスの非正常的な終了、一部プロセスの非正常的な動作、非正常リターンしてエラーコードなし、及び正常リターンして不正確なエラーコード返還のうちの少なくとも一つを分析することを特徴とする請求項 4 に記載の例外処理テスト装置。

10

【請求項 6】

前記欠陥モデルは、テスト対象となるデバイス及びデバイスドライバに従い変更されることを特徴とする請求項 1 に記載の例外処理テスト装置。

【請求項 7】

デバイスドライバ、デバイスマネージャー及びアプリケーションを含むシステムに対する例外処理テスト方法において、

欠陥モデル及び前記デバイスマネージャーから獲得した情報に基づき変形されたデバイスドライバを生成する生成段階、

20

前記変形されたデバイスドライバを利用して前記デバイスドライバをフッキングするフッキング段階、

前記アプリケーションが動作する間、フッキングされ前記変形されたデバイスドライバから前記アプリケーションに返還されるテスト情報を収集するスキニング段階、及び収集されたテスト情報を分析する分析段階を含むことを特徴とする例外処理テスト方法。

【請求項 8】

前記欠陥モデルは、アプリケーションとデバイスとの間の連結、デバイスオープン、連結解除、デバイス閉鎖、ヌル (NULL) データ、データ誤謬、デバイス電源、及び処理遅延のうち少なくとも一つであることを特徴とする請求項 7 に記載の例外処理テスト方法。

30

【請求項 9】

前記テスト情報は、前記変形されたデバイスドライバのオペレーション返還値、前記デバイスマネージャーで発生する例外コード値、前記アプリケーションの返還値、及びシステム異常コードのうちの少なくとも一つであることを特徴とする請求項 7 に記載の例外処理テスト方法。

【請求項 10】

前記テスト情報が欠陥の発生を意味する場合、前記アプリケーションは前記発生した欠陥に伴い例外処理を行うことを特徴とする請求項 7 に記載の例外処理テスト方法。

【請求項 11】

40

前記例外処理が正常に行われない場合、前記分析段階は前記テスト情報を全体システムダウン、一部プロセスダウン、一部プロセスの非正常的な終了、一部プロセスの非正常的な動作、非正常リターンしてエラーコードなし、及び正常リターンして不正確なエラーコード返還のうちの少なくとも一つを分析することを特徴とする請求項 10 に記載の例外処理テスト方法。

【請求項 12】

前記欠陥モデルは、テスト対象となるデバイス及びデバイスドライバに従い変更されることを特徴とする請求項 7 に記載の例外処理テスト方法。

【発明の詳細な説明】

【技術分野】

50

【 0 0 0 1 】

本発明は例外処理テスト装置及び方法に関し、より詳しくは変形されたデバイスドライバを利用して欠陥を発生させ、前記欠陥に対するシステムの例外処理の良否を確認する例外処理テスト装置及び方法に関する。

【 背景技術 】

【 0 0 0 2 】

多様なソフトウェア及びハードウェアを含むエンベデッドシステムにおいて、システムの運営環境で相互作用 (i n t e r a c t i o n) する動作が重要である。したがって、ソフトウェアのみで構成されたテストのための仮想データでない、システムが実運用されながら得る実際データ (l i v e e n v i r o n m e n t w i t h r e a l d a t a) を利用する動作が重要であり、システムをテストしようとする場合にも、実運用データを利用してシステムテストを行って始めて潜在的欠陥、又は実運用上発生する欠陥を効果的に捜し出すことができる方案が求められる。

10

【 0 0 0 3 】

さらに、システムを構成するソフトウェアそれぞれを個別にテストするためには、連結する相対コンポーネントを代替するための装備を備えなければならず、加えて多くのテスト資源と費用を要するため、システムに含まれる全てのプロセスをリアルタイムでテストすることができ、メモリ空間のオーバーヘッド又は当該コード遂行による性能オーバーヘッドを最小化するテスト装置が求められる。

20

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 4 】

本発明の目的は、フッキング技術を利用して変形されたデバイスドライバをシステムに挿入し、前記変形されたデバイスドライバにより発生する欠陥を確認し、確認された欠陥に伴う例外処理の良否を確認することにより、実運用されるシステムに対するテストが可能であり、テスト資源のオーバーヘッドを最小化することができる例外処理テスト装置及び方法を提供することにある。

【 課題を解決するための手段 】

【 0 0 0 5 】

本発明は例外処理テスト装置に関し、欠陥モデル及びデバيسマネージャーから獲得した情報に基づき変形されたデバイスドライバを生成する生成モジュール、前記変形されたデバイスドライバを利用して前記デバイスドライバをフッキングするフッキングモジュール、アプリケーションが動作する間、フッキングされた前記変形されたデバイスドライバから前記アプリケーションに返還されるテスト情報を収集するスキニングモジュール、及びスキニングモジュールにより収集されたテスト情報を分析する分析モジュールを含む。

30

【 0 0 0 6 】

本発明は例外処理テスト方法に関し、欠陥モデル及びデバيسマネージャーから獲得した情報に基づき変形されたデバイスドライバを生成する生成段階、前記変形されたデバイスドライバを利用して前記デバイスドライバをフッキングするフッキング段階、前記アプリケーションが動作する間、フッキングされた前記変形されたデバイスドライバから前記アプリケーションに返還されるテスト情報を収集するスキニング段階、及び収集されたテスト情報を分析する分析段階を含む。

40

【 発明の効果 】

【 0 0 0 7 】

本発明によれば、ソースコードでない実運用されるコードを対象にするので、システムが実際に運用される間にデバيسマネージャーを利用した例外処理テストを行うことができる。

【 0 0 0 8 】

さらに、それぞれのプロセスに対する個別的なコード挿入ではない所定のメモリ空間の

50

プロセスコンテキスト情報を活用したフッキング方法を利用するので、システムのオーバーヘッドを最小化することができる。

【 0 0 0 9 】

さらに、プロセス間の共有資源が制限された場合にも、リアルタイムでテスト情報を収集してテスト結果を確認することができる。

【図面の簡単な説明】

【 0 0 1 0 】

【図 1】本発明の一実施例に係わる例外処理テスト装置が利用される全体システムを示す図である。

【図 2】本発明の一実施例に係わる例外処理テスト装置の構成を示す図である。

10

【図 3】本発明の一実施例に係わる例外処理テスト方法を示すフローチャートである。

【発明を実施するための形態】

【 0 0 1 1 】

本発明は、システムが実運用される環境でのテストに係り、システムの始動から終了までの動的状況の中でシステムをテストすることができる装置及び方法に関する。本発明は、コード挿入技術の一種であるフッキングを利用し、システム内で動作する全てのプロセスに対するテストを行うことができる。より詳しくは、システム内で動作するデバイスから発生可能な欠陥類型及び前記欠陥に対する例外処理テストを行うことができる。

【 0 0 1 2 】

図 1 は、本発明の一実施例に係わる例外処理テスト装置が利用される全体システムを示す図である。

20

【 0 0 1 3 】

図 1 に示したように、全体システムは例外処理テスト装置 1 0 0、システム 2 0 0、入力装置 3 0 0、格納装置 4 0 0 及び出力装置 5 0 0 を含む。

【 0 0 1 4 】

例外処理テスト装置 1 0 0 は、システム 2 0 0 に含まれたアプリケーションとデバイスとの間の通信過程で発生する欠陥類型、及び前記欠陥に対する例外処理をテストするための装置である。例外処理テスト装置 1 0 0 は、システムに含まれたデバイスを管理するデバيسマネージャーを利用して変形されたデバイスドライバを生成し、前記変形されたデバイスドライバを利用して欠陥を生成することによりシステムをテストすることができる。前記例外処理テスト装置 1 0 0 は、図 1 で示したようにシステム 2 0 0 と独立に備えられてもよく、システム 2 0 0 の内部に含まれる形態で具現されてもよい。

30

【 0 0 1 5 】

前記システム 2 0 0 は、マルチプロセスが通信する全てのエンベデッドシステム及びコンピュータシステムであって、例えば、インフォテインメントシステム、マルチメディアシステム、有無線ネットワークシステムなどである。例えば、システム 2 0 0 が車両インフォテインメントシステムの場合、システム 2 0 0 は AVN (Audio Video Navigation) 機能を中心に TDMB (Terrestrial Digital Multimedia Broadcasting)、TPEG (Transport Protocol Expert Group)、IPOD、USB、ブルートゥース (Bluetooth (登録商標))、VR (Voice Recognition)、PGS (Parking Guidance System)、RDS (Radio Data System)、VDC (Vehicle CDC)、DIS (Driver Information System)、RSE (Rear Seat Entertainment)、HEV (Hybrid Electric Vehicle) display、AHU (Air Handling Unit) display などを含む。

40

【 0 0 1 6 】

さらに、前記システム 2 0 0 はデバイス、前記デバイスとアプリケーションとの間のデータ送受信のためのデバイスドライバ、前記デバイスを管理するためのデバيسマネー

50

ジャー及び前記デバイスマネージャーと連結されデバイスを利用して所定のプログラムを実行するアプリケーションを含む。したがって、例外処理テスト装置１００はデバイスマネージャーを利用して現在活性化されたデバイスドライバーを確認し、前記デバイスドライバーに対応する変形されたデバイスドライバーを生成し、前記変形されたデバイスドライバーを利用してデバイスドライバーをフッキングすることにより、例外処理テストを行う。ここで、デバイスはＵＳＢ、マルチメディアプレーヤーなどのハードウェアデバイスであって、アプリケーションが動作する装置と物理的に分離された装置であってもよい。本発明のシステム２００は、物理的に分離されたデバイス及びアプリケーションが動作する装置を含む概念である。

【００１７】

10

入力装置３００は、例外処理テスト装置１００及びシステム２００を駆動させるための使用者入力装置である。入力装置３００はキーボード又はタッチスクリーンに具現されてもよく、入力装置３００を介して入力される信号に応じてシステム２００に含まれるアプリケーションが実行され、例外処理テスト装置１００により変形されたデバイスドライバーがフッキングされたシステム２００が動作する。

【００１８】

格納装置４００は、例外処理テスト装置１００により収集された情報を格納する。前記例外処理テスト装置１００は格納装置４００と一つの装置に具現され、テスト情報を収集及び分析するとともに収集されたテスト情報を格納する。

【００１９】

20

出力装置５００は、例外処理テスト装置１００により収集されたテスト情報又は分析されたテスト結果を、イメージ又は音声で出力する装置である。

【００２０】

図２は、本発明の一実施例に係わる例外処理テスト装置の構成を示す図である。

【００２１】

図２に示したように、例外処理テスト装置１００は生成モジュール１０１、フッキングモジュール１０３、スキャニングモジュール１０５及び分析モジュール１０７を含む。例外処理テスト装置１００と連結されたシステム２００はデバイス（図示省略）、前記デバイスとアプリケーションとの間のデータ送受信のためのデバイスドライバー（図示省略）、前記デバイスを管理するデバイスマネージャー２０１及び前記デバイスマネージャー２０１と連結され、前記デバイスを介して所定の機能を具現するためのアプリケーション２０２を含む。例外処理テスト装置１００は二つ以上のデバイス（図示省略）、及びアプリケーション２０２に対する例外処理テストを行う。

30

【００２２】

デバイスマネージャー２０１は、システム２００に含まれた全てのデバイスドライバーに対するランタイム実行情報を有する。ここで、ランタイム実行情報とは、システム２００が動作するあいだ全てのデバイスとプロセスとの間の実行情報を意味する。即ち、デバイスマネージャー２０１はシステム２００内で活性化されたデバイス名及びデバイスドライバー、各デバイスドライバーで利用されるオペレーション及びそのオペレーションのスタートアドレス、デバイスドライバーがローディングされたメモリアドレス、メモリ空間などに対する情報を管理する。したがって、全てのシステム内で前記言及した情報を含むデータ構造又は装置は、本発明で意味するデバイスマネージャー２０１に該当する。

40

【００２３】

生成モジュール１０１は、デバイスマネージャー２０１を利用してシステム２００で現在動作しているデバイスドライバーに対する情報を収集し、欠陥モデル及び収集したデバイスドライバーに対する情報に従い変形されたデバイスドライバーを生成する。前記欠陥モデルは下記の図３に関連して詳しく説明する。

【００２４】

前記変形されたデバイスドライバーは、デバイスと送受信するデータを変形させてシステム動作の欠陥を発生させる。即ち、変形されたデバイスドライバーは各オペレーション

50

で利用されるプロトコルデータをデバイスに従い分析し、デバイスを通過するデータを変形規則に従い変形させることにより、変形したデータをアプリケーションとデバイスとの間の通信過程で利用し、システム動作の欠陥を発生させる。変形規則の一実施例として、デバイスと送受信するデータが損失されるようにする変形規則、デバイスと送受信するデータに有効範囲の境界に該当する値を追加するようにする変形規則、データ送信障害によりデータ値が入りまじるようにする変形規則などがある。

【 0 0 2 5 】

入力装置 3 0 0 によりテスト対象が先ず選択された後、テスト対象となるデバイスドライバに対する情報を収集してもよく、現在動作しているデバイスドライバに対する情報を収集した後、テスト対象となるデバイスドライバを選択してもよい。

10

【 0 0 2 6 】

フッキングモジュール 1 0 3 は、原本デバイスドライバを前記生成モジュール 1 0 1 により生成され変形されたデバイスドライバでフッキングする。即ち、フッキングモジュール 1 0 3 は生成モジュール 1 0 1 により収集された情報を利用して原本デバイスドライバをフッキングする。例えば、デバيسマネージャ 2 0 1 により管理される情報のうちテスト対象となるデバイスドライバのオペレーションスタートアドレスを変形されたデバイスドライバのオペレーションスタートアドレスに変更することにより、原本デバイスドライバをフッキングする。したがって、システムが動作するあいだ原本デバイスドライバが呼び出される代わりに、変形されたデバイスドライバが呼び出される。

【 0 0 2 7 】

20

スキャンングモジュール 1 0 5 は、デバイスとアプリケーションが動作する間、変形されたデバイスドライバにより実行されアプリケーションに返還されるテスト情報を収集する。例えば、デバイスドライバのオペレーション返還値、デバيسマネージャ 2 0 1 で発生する例外コード値、アプリケーションの返還値又はシステム異常コードなどを収集する。さらに、スキャンングモジュール 1 0 5 はアプリケーション 2 0 2 が動作する間、欠陥判定に必要なシステム動的状態情報、例えば、関数パラメーター及び返還値、性能測定値などを収集する。

【 0 0 2 8 】

即ち、スキャンングモジュール 1 0 5 は変形されたデバイスドライバにより変形されたデータを利用することにより発生するシステム動作の欠陥類型を収集し、前記欠陥に対するシステムの例外処理結果を確認する。

30

【 0 0 2 9 】

分析モジュール 1 0 7 は、前記スキャンングモジュール 1 0 5 により収集された例外及び例外処理結果を分析する。即ち、収集された情報に基づきアプリケーションが例外処理を正常に動作したのかしないのかが判断される。例えば、例外処理結果が P A S S の場合、システムが発生した例外に対し好適な例外コードを出力し、適切な処理を行ったと分析され、例外処理結果が F A I L の場合、システムが発生した例外に対して適宜処理できなかったと分析される。

【 0 0 3 0 】

前記例外処理結果が F A I L の場合は、全体システムダウン (C a t a s t r o p h i c)、一部プロセスダウン (R e s t a r t)、一部プロセスの非正常的な終了 (A b o r t)、一部プロセスの非正常的な動作 (A b n o r m a l)、非正常リターンしてエラーコードなし (S i l e n t)、及び正常リターンして不正確なエラーコード返還 (H i n d e r i n g) に分類される。

40

【 0 0 3 1 】

図 3 は、本発明の一実施例に係わる例外処理テスト方法を示すフローチャートである。

【 0 0 3 2 】

図 3 は、アプリケーションとデバイスとの間の通信過程で発生する欠陥、及び前記欠陥に対する例外処理テストを行うための例外処理テスト方法を示す。先ず、欠陥モデルが定義される (S 1 1 0)。ここで、欠陥モデルとは、欠陥類型に従いテストされなければな

50

らないデバイス及びデバイスドライバの欠陥発見／復旧のための情報を意味する。欠陥モデルは、テスト対象となるデバイス及びデバイスドライバに従い変更される。定義された欠陥モデルに従い、システム200が正常に動作する間にテスト情報を収集できる、変形されたデバイスドライバが生成される。

【0033】

ここで、アプリケーションとデバイスとの間の通信過程で発生する欠陥類型は、アプリケーションとデバイスとの間の連結(CONNECTION)、デバイスオープン(OPEN)、連結解除(DISCONNECTION)、デバイス閉鎖(CLOSE)、データ入出力と関連してヌルデータ(NULLDATA)とデータ誤謬(ILLEGAL DATA)と処理遅延(TIMING)、又はデバイス電源(POWER)と関連した欠陥

10

【0034】

次に、S110段階で定義された欠陥モデル、及びシステム200で現在動作しているデバイスドライバに対する情報に基づき、変形されたデバイスドライバが生成される(S120)。

【0035】

次に、原本デバイスドライバが前記変形されたデバイスドライバにフッキングされる(S130)。即ち、原本デバイスドライバが呼び出される時点で、原本デバイスドライバの代わりに変形されたデバイスドライバが呼び出される。

【0036】

20

次に、前記デバイスマネージャ201と連結されたアプリケーション202が駆動され、前記アプリケーション202が動作するに伴い、デバイスマネージャ201及びデバイス(図示省略)が共に動作する(S140)。

【0037】

アプリケーション202が動作する間、変形されたデバイスドライバにより実行された結果、即ち、システム動作で発生した欠陥及び前記欠陥に対するシステムの例外処理結果が収集される(S150)。S150段階で収集されるテスト情報は現在コールスタック情報、デバイスドライバ情報及びプロセス情報を含む。コールスタック情報を利用し、処理フロー中に如何なる問題が発生したのかが確認される。デバイスドライバ情報はリターン値、誤謬IDなどを含み、プロセス情報はプロセスID及び名前を含む。

30

【0038】

次に、S150段階で収集されたテスト情報が分析される(S160)。S160段階の分析結果に従い、アプリケーションが例外処理を正常に動作させたか否かを確認することができる。

【0039】

本発明に係わる欠陥モデルは、アプリケーションとデバイスとの間の連結、デバイスオープン、連結解除、デバイス閉鎖、データ入出力と関連してヌルデータ(NULL)とデータ誤謬及び処理遅延、又はデバイス電源に係わる欠陥などを含む。

【0040】

連結に係わる欠陥は、デバイス連結初期化失敗、デバイスオープン失敗に分類される。

40

【0041】

連結解除に係わる欠陥は、デバイス終了化失敗、デバイス連結解除失敗に分類される。

【0042】

デバイス閉鎖に係わる欠陥は、デバイス閉鎖失敗に定義される。

【0043】

ヌル(NULL)データに係わる欠陥は、デバイスからのデータ読み取り失敗、デバイスへのデータ書き込み失敗、デバイスへのデータ探索失敗、デバイスからのI/OCTLデータ伝達失敗に分類される。

【0044】

データ誤謬に係わる欠陥は、デバイスからの誤ったデータ読み取り、デバイスへの誤っ

50

たデータ読み取り、デバイスへの誤ったデータ探索、デバイスへの誤った値の I O C T L 送信、デバイスからの誤った値の I O C T L 送信に分類され得る。

【 0 0 4 5 】

デバイス電源に係わる欠陥は、デバイスパワーアップ失敗、デバイスパワーダウン失敗に分類される。

【 0 0 4 6 】

本発明は、プロセッサが読み取り可能な記録媒体に、プロセッサが読み取り可能なコードとして具現できる。プロセッサが読み取り可能な記録媒体は、プロセッサが読み取り可能なデータが格納される全ての種類の記録装置を含む。プロセッサが読み取り可能な記録媒体の例には R O M、R A M、C D - R O M、磁気テープ、フロッピー(登録商標)ディスク、光データ格納装置などがある。さらに、プロセッサが読み取り可能な記録媒体はネットワークで連結したコンピュータシステムに分散され、プロセッサが読み取り可能なコードが分散方式で格納され実行される。

10

【 0 0 4 7 】

本発明は、前述した一実施例に限定されず、特許請求範囲で記載された本発明の要旨を外れることなく、当該発明の属する技術分野の当業者により多様な変形実施が可能であり、このような変形実施等は本発明の技術的思想や展望から個別に理解されてはならない。

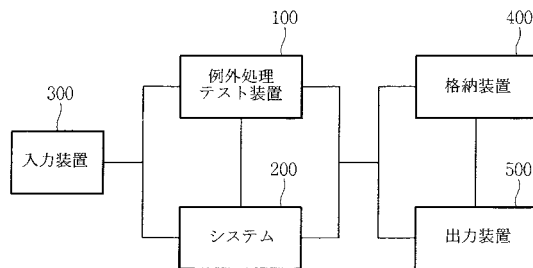
【産業上の利用可能性】

【 0 0 4 8 】

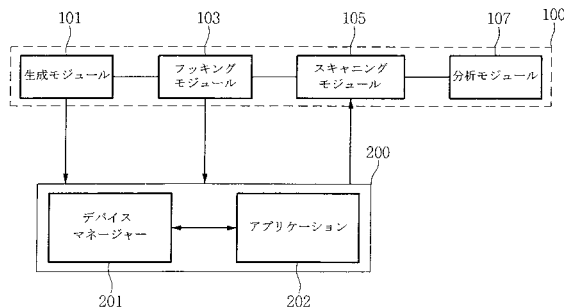
本発明によれば、システムが実際に運用される間にデバイスマネージャーを利用した例外処理テストを実行することが可能である。

20

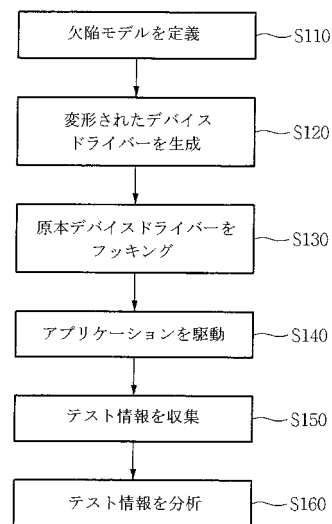
【 図 1 】



【 図 2 】



【 図 3 】



フロントページの続き

- (74)代理人 110000051
特許業務法人共生国際特許事務所
- (72)発明者 チォエ, ビョン - ジュ
大韓民国, ソウル 135 - 788, ガンナム - グ, アックジョン - ドン, ヒョンデ アパート#
74 - 1305
- (72)発明者 ソ, ジュ - ヨン
大韓民国, ソウル 138 - 200, ソンパ - グ, ムンジョン - ドン, オリンピック ファミリー
タウン アパート#105 - 304
- (72)発明者 ヤン, スン - ワン
大韓民国, ギョンギ - ド 435 - 704, グンボ - シ, グウンジョン - ドン, サムスン ジャン
ミ アパート#1137 - 2202
- (72)発明者 キム, ヨン - スウ
大韓民国, ギョンギ - ド 446 - 588, ヨンイン - シ, ギフン - グ, シンガル - ドン697,
ノクオオン マウル セチョンニヨン グリンビル アパート#509 - 503
- (72)発明者 オ, ジョン - ソク
大韓民国, ギョンギ - ド 423 - 801, グァンミョン - シ, グァンミョン1 - ドン70 - 13
, 2F
- (72)発明者 クォン, ヘ - ヨン
大韓民国, ギョンギ - ド 446 - 567, ヨンイン - シ, ギフン - グ, ボジョン - ドン, ヒョン
ダイ アイ - パーク 1 - チャ アパート#202 - 601
- (72)発明者 ジャン, スン - ヨン
大韓民国, ギョンギ - ド 412 - 741, ゴヤン - シ, ドクヤン - グ, ファジョン 2 - ドン,
ビョルビット マウル ゴンヨン アパート#1004 - 1101

審査官 多胡 滋

- (56)参考文献 特開平10 - 133914 (JP, A)
特開2009 - 104490 (JP, A)
特開2010 - 224795 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 11/28
G06F 11/36