

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4216372号
(P4216372)

(45) 発行日 平成21年1月28日 (2009. 1. 28)

(24) 登録日 平成20年11月14日 (2008. 11. 14)

(51) Int. Cl.

F I

G 0 6 F 9/06 (2006. 01)

G 0 6 F 9/06 4 1 0 C

G 0 6 F 11/22 (2006. 01)

G 0 6 F 11/22 3 6 0 K

請求項の数 15 (全 20 頁)

(21) 出願番号	特願平10-201001	(73) 特許権者	597001637
(22) 出願日	平成10年7月15日 (1998. 7. 15)		デル・ユーエスエイ・エルピー
(65) 公開番号	特開平11-161476		DELL USA, L. P.
(43) 公開日	平成11年6月18日 (1999. 6. 18)		アメリカ合衆国テキサス州78682-2
審査請求日	平成17年5月31日 (2005. 5. 31)		244, ラウンド・ロック, ワン・デル・
(31) 優先権主張番号	921438		ウェイ (番地なし)
(32) 優先日	平成9年8月29日 (1997. 8. 29)		One Dell Way, Round
(33) 優先権主張国	米国 (US)		Rock, TX 78682-2244,
前置審査			United States of Am
			erica
		(74) 代理人	100058479
			弁理士 鈴江 武彦
		(74) 代理人	100091351
			弁理士 河野 哲
			最終頁に続く

(54) 【発明の名称】 オーダーに対して組立てられたコンピュータシステムのためのソフトウェア設置およびテストを容易にするデータベース

(57) 【特許請求の範囲】

【請求項 1】

ターゲットコンピュータシステムの構成要素のリストを受信する手段と、
ステップテーブルと構成要素テーブルとを備えているデータベースと、

データベースを調べ、データベースから、ターゲットコンピュータシステムの構成要素
の前記リストに記載された構成要素に、対応したステップを取り出すことにより、実行可
能な命令ラインを含んでいるステップシーケンスを生成するように構成されているステッ
プメーカーコンピュータシステムと、

実行可能なファイルを記憶するように構成された不揮発性記憶媒体、および / またはファ
イルサーバとを備えており、

ステップテーブルは、考えられるコンピュータシステムの異なった構成要素に対する 1 組
のソフトウェア設置ステップ、および / またはテストステップを含んでおり、ステップテ
ーブルは、ソフトウェア設置ステップ、および / またはテストステップを実行するための
実行可能な命令ラインを備えており、

構成要素テーブルは、考えられるコンピュータシステム内に含まれる 1 組の考えられる構
成要素を含んでおり、

データベースは、構成要素テーブルの構成要素とステップテーブルのステップとの間の
関係を含んでおり、

ステップシーケンスは、ターゲットコンピュータシステムの構成要素の設置、および / ま
たはテストをするための実行可能な命令ラインを含んでおり、

10

20

ステップメカコンピュータは、前記ステップシーケンスを不揮発性記憶媒体、および／またはファイルサーバに出力するように構成され、
実行可能なファイルは、ターゲットコンピュータシステム上でステップシーケンスを実行することが可能である、

ターゲットコンピュータシステムにソフトウェアを設置し、および／またはソフトウェアをテストする装置。

【請求項 2】

ステップテーブルは、各ソフトウェア設置ステップまたはテストステップに対する識別を行うステップ識別属性を含んでいる請求項 1 記載の装置。

【請求項 3】

ステップテーブルは、ステップが実行される製造のフェーズを指定するフェーズ属性を含んでいる請求項 1 または 2 記載の装置。

【請求項 4】

ステップテーブルは、対応するステップが実行された後、一時停止または再ブートが必要か否かを識別するアフターコードを含んでいる請求項 1 または 2 記載の装置。

【請求項 5】

ステップテーブルは、対応するステップが動作する回数の最大数を示す最大インスタンス属性を含んでいる請求項 1 または 2 記載の装置。

【請求項 6】

ステップテーブルは、ソフトウェア設置ステップまたはテストステップと関連する構成要素を識別するクラス識別属性を含んでいる請求項 1 または 2 記載の装置。

【請求項 7】

構成要素テーブルは、構成要素テーブル内の各構成要素を識別する構成要素識別属性を含んでいる請求項 1 または 2 記載の装置。

【請求項 8】

構成要素テーブルは、構成要素のクラスを識別するクラス識別属性を含んでいる請求項 1 または 2 記載の装置。

【請求項 9】

製造されるターゲットコンピュータシステムはコンピュータシステムのファミリーに対応し、データベースは

さらに、1組のソフトウェア設置ステップと製造されているコンピュータシステムの各ファミリーとの間の関係を識別するファミリー関係型テーブルと、

コンピュータシステムのファミリーと1組の構成要素との間の関係を識別するファミリー構成要素関係型テーブルと、

構成要素と対応する構成要素に対して適切な1組のソフトウェア設置ステップとの間の関係を識別するファミリーステップ関係型テーブルとを有する請求項 1 または 2 記載の装置。

【請求項 10】

ファミリー関係型テーブルは、コンピュータシステムの特定の1つのファミリーを識別するファミリー識別属性を含んでいる請求項 9 記載の装置。

【請求項 11】

ファミリー関係型テーブルは、特定のファミリーに対して適切な特定の1組のステップを識別するステップ識別属性を含んでいる請求項 9 記載の装置。

【請求項 12】

ファミリー関係型テーブルは、ファミリーと関係するステップが実行される予め定められた順序を表している各ファミリーに対応するシーケンス番号を含んでいる請求項 9 記載の装置。

【請求項 13】

ファミリー構成要素関係型テーブルは、コンピュータファミリーを識別する属性と、構成要素を識別する属性との間の関係を定めている請求項 9 記載の装置。

10

20

30

40

50

【請求項 1 4】

ファミリー構成要素関係型テーブルは、構成要素識別属性とステップ識別属性との間の関係を定めている請求項 9 記載の装置。

【請求項 1 5】

さらに、名称によりコンピュータシステムのファミリーを識別するファミリーテーブルを含んでいる請求項 1 または 2 記載の装置。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明はコンピュータシステム診断、特にコンピュータシステムのソフトウェア設置および/またはテストのステップの順序付けの方法に関する。

【0002】**【従来の技術】**

汎用パーソナルコンピュータシステムとIBMの両立式パーソナルコンピュータシステムは特に多数の階層の社会に計算力を提供するために広く使用されている。パーソナルコンピュータシステムはデスクトップ型、直立型、またはポータブルマイクロコンピュータとして通常限定されることができ、これはシステムプロセッサ、関連する揮発性および非揮発性メモリを有するシステム装置と、ディスプレイモニタ、キーボード、1以上のディスク駆動装置、固定したディスク記憶装置、およびオプションな装置としてプリンタを含んでいる。

【0003】

ビジネスまたは個人の顧客へ出荷される前に、コンピュータシステムにソフトウェアをインストールしテストを行うことが知られている。ソフトウェアの設置とテストをする目的は、エラーがなく、動作する準備が整った便利で信頼性のあるコンピュータシステムが効率的に製造されビジネスまたは個人の顧客へ出荷されることができるようになるためである。通常、テストはコンピュータシステムのハードウェア部分およびソフトウェア部分の両方で生じたエラーを検出し解析する。コンピュータシステムハードウェアテストの部分的なリストはプロセッサ、メモリ、ディスク記憶装置、オーディオ装置、グラフィック装置、キーボード、マウス、プリンタなどのハードウェア構成要素についての診断を含んでいる。ソフトウェア設置は所望のパッケージのソフトウェアをコンピュータシステムへ負荷し、コンピュータに適切な環境変数を準備し、負荷されたソフトウェアウェアに適切な初期化ファイルを準備することを含んでいる。ソフトウェアテストはしばしばソフトウェアの所望なバージョンがコンピュータシステムへインストールされ、適切なドライバがコンピュータシステム上に存在することを確認することを含んでいる。

【0004】

顧客の元へ出荷される前に固定した手順を実行することにより製造中に、コンピュータシステムにソフトウェアをインストールしテストすることが産業上知られている。例えば、あるタイプのコンピュータシステム用の診断テストを含んだディスクセットが製造される。ディスクセットは長くて、多くは複雑なバッチファイルを含んでおり、これはソフトウェア設置と診断処理を行う。ディスクセットはさらに購入されたコンピュータシステムについてテストを行うための実行可能なファイルを全て含んでいる。

【0005】

構築された各コンピュータシステムにはこのディスクセットのコピーがそれぞれ設けられている。これらのディスクセットは製造プロセス中に工場床面で構築されたコンピュータシステムに付随され、テストがバッチファイルの固有の順番にしたがってそれぞれのコンピュータシステムで行われる。プロセスに変形を行うことが必要であるならば、バッチファイルはバッチコードに付加またはバッチコードから削除することによってそれに対応して変更される。バッチファイルに対する変更によって、製造される次の各コンピュータシステムのテストパラメータ（テストが行われる順序を含む）に対応する変化をもたらし、各コンピュータシステムは同一のバッチファイル診断手順を共有している。

【 0 0 0 6 】

【 発明が解決しようとする課題 】

この種類の診断装置は出荷前にコンピュータシステムの信頼性を増加させるのにある程度有効であるが、改良の余地が残されている。例えば、テストがより複雑で完全になるように続けるとき、診断テストのバッチファイルと実行可能なファイルはしばしばディスクットの記憶容量を超過する。さらに、多くの場合、1つの注文組込みコンピュータシステムまたはあるコンピュータシステムのファミリにおけるテストおよびソフトウェア設置処理のカスタム化は、他のシステムまたはファミリのテストを変更せずに行うことは困難または不可能である。さらに、他のシステムまたはファミリに対する順序を変更せずに単一の注文組込みコンピュータシステムまたはあるコンピュータシステムのファミリに対するソフトウェア設置またはテストの順序を変更することは困難あるいは不可能である。多くは複雑であるという現在のバッチファイル構造の特性は、製造業者がテストおよびソフトウェア設置手順を迅速に効果的に故障点検またはテストを維持することを困難にする。したがって、顧客へ出荷する前に、ソフトウェアをインストールし、コンピュータシステムをテストする改良された方法を発明することが望まれている。

10

【 0 0 0 7 】

【 課題を解決するための手段 】

コンピュータシステム上にソフトウェアをインストールするための装置は、ステップテーブルおよび構成要素テーブルを含んでいる。ステップテーブルは、製造されるほぼ全てのコンピュータシステムの異なる構成要素間で共用される1組のソフトウェア設置ステップを含んでいる。好ましい実施形態において、コンピュータシステムはコンピュータシステムのファミリーに対応し、装置はさらにファミリー関係型テーブルと、ファミリー構成要素関係型テーブルを含んでいる。ファミリー関係型テーブルは、1組のソフトウェア設置ステップと、製造されるコンピュータシステムの各ファミリーとの間の関係を識別する。ファミリー構成要素関係型テーブルは、コンピュータシステムのファミリーと1組の構成要素との間の関係を識別する。ファミリーステップ関係型テーブルは、構成要素と、対応する構成要素に適した1組のソフトウェア設置ステップとの間の関係を識別する。

20

【 0 0 0 8 】

別の好ましい実施形態において、本発明は、ステップがそれぞれの構成要素記述子と関連付けられ、それぞれのシーケンス番号を含んでいる複数のステップを提供する装置に関する。構成要素記述子は、コンピュータシステムのそれぞれの構成要素を記述する。装置は、ステップテーブルおよび構成要素テーブルを含んでいる。ステップテーブルは、製造されるほぼ全てのコンピュータシステムの異なる構成要素間で共用される1組のソフトウェア設置ステップを含んでいる。構成要素テーブルは、製造されるコンピュータシステム内に含まれたほぼ全ての可能な1組の構成要素を含んでいる。

30

【 0 0 0 9 】

従って、前述の装置は有効なソフトウェア設置とコンピュータテストを行い、それによって注文に応じて組立てられたコンピュータシステムにおける簡単なトラベルシューティングとカスタム化を可能にする。装置のモジュール設計はテストシステムの基本的なメンテナンスと、新しいコンピュータシステムおよびファミリのステップを迅速に生成することを有効に可能にする。

40

【 0 0 1 0 】

本発明のこれらおよび他の目的、特徴、利点をさらに説明し、好ましい実施形態の詳細な説明からより明白になるであろう。

【 0 0 1 1 】

【 発明の実施の形態 】

以下、本発明を実行するために考えられた最良のモードについて詳細に説明する。説明は例示のためであり、権利を限定するものではない。図面では類似または同一の素子は同一の参照符号で示されている。説明上、モジュールは1つの命令または命令のセットとして限定されている。

50

【 0 0 1 2 】

図 1 はソフトウェア設置およびテストシステム90の概略図である。動作において、注文組込みターゲットコンピュータシステム160 を購入するように注文92が行われる。ターゲットシステム160 は複数のハードウェアとソフトウェア構成要素を含むように製造されている。例えば、ターゲットシステム160 は、あるブランドのハードウェアドライブと、特定のタイプのモニタと、あるブランドのプロセッサと、特定のバージョンの動作システムとを含んでいる。ターゲットシステム160 が顧客の元へ出荷される前に、複数の構成要素がインストールされテストされる。このようなソフトウェア設置とテストは、入手してすぐに動作できる信頼性のある動作コンピュータシステムを確実にするために有効である。

【 0 0 1 3 】

異なったファミリのコンピュータシステムおよび異なった個人のコンピュータ構成要素が異なったソフトウェア設置およびテストステップを必要とするので、ターゲットシステム160 で行う必要があるテストと、これらのテストを実行する順序を決定する必要がある、それによって実効的なソフトウェア設置およびテストプロセスを実現する。ステップメーカ140 はターゲットシステム160 で行われるソフトウェア設置およびテストステップの順序を定めて配列するために構成されたコンピュータシステムである。ソフトウェア設置および/またはテストステップを配列するため、ステップメーカ140 、特にそこに存在する配列プログラム204 は最初に記述子ファイル96から複数の構成要素記述子を読取る。記述子ファイル96は所望の構成要素を有する所望のコンピュータシステムに対応する注文92を変換モジュール94によりコンピュータの読取り可能なフォーマットに変換することによって与えられる。

【 0 0 1 4 】

構成要素記述子はターゲットシステム160 の構成要素におけるコンピュータが読取り可能な記述であり、その構成要素は注文92により限定される。好ましい実施形態では、構成要素記述子はシステム記述子記録と呼ばれる記述子ファイルに含まれており、この記述子ファイルは構成要素、ハードウェアおよび/またはソフトウェア構成要素のリストを含んでいるコンピュータの読取り可能なファイルであり、ターゲットシステム160 へインストールされる。複数の構成要素記述子を読取った後、配列プログラム204 は、ネットワーク接続110 によってデータベース100 から構成要素記述子に対応する複数のソフトウェア設置および/またはテストステップを検索する。ネットワーク接続110 は構内通信網、イントラネット、インターネット等の技術でよく知られた任意のネットワーク接続でよい。データベース100 に含まれている情報は矢印130 で示されている変更によって更新される。

【 0 0 1 5 】

ターゲットシステム160 に適切なソフトウェア設置および/またはテストステップを検索し、配列プログラム204 は各ステップに対応するシーケンス番号にしたがって予め定められた順序でステップの順序を定めて配列する。ターゲットシステム160 に必要なステップを配列した後、配列プログラム204 は一連の出力ファイルをステップディスク150 へ書込む。図 1 で説明した実施形態では、出力ファイルは適切なソフトウェア設置および/またはテストステップをターゲットシステム160 で実行するために適切な命令ラインを含んだテキストファイルを含んでいる。各ステップに対応するシーケンス番号にしたがって予め定められた順序で実行される。ステップディスク150 は、工場床面上のターゲットシステム160 に伴い、そこではステップディスク150 または代わりにネットワーク接続180 を経てターゲットシステム160 に接続されているファイルサーバ190 から直接テストが行われる。好ましくはネットワーク接続180 はターゲットコンピュータシステムの対応するネットワークポートへプラグ接続された一般的なネットワーク装置である。ソフトウェア設置およびテストステップの実行に続いて、ソフトウェア設置とテストの結果がネットワーク接続180 によってファイルサーバ190 へ戻り記録される。

【 0 0 1 6 】

図 2 は本発明の別の実施形態にしたがったソフトウェア設置およびテストシステム192 の概略図である。顧客は注文組込みターゲットコンピュータシステム160 を購入する注文92

10

20

30

40

50

を行う。ターゲットシステム160 は複数の構成要素を含むように製造され、それらの構成要素はハードウェアおよび/またはソフトウェア構成要素の両者を含んでもよい。ターゲットシステム160 が顧客へ出荷される前に、複数の構成要素がインストールされテストされる。このようなソフトウェア設置とテストは、顧客が入手してすぐに作動できる信頼性のある動作コンピュータシステムを確実にするために有効である。

【0017】

ソフトウェア設置とテストステップを配列するため、配列プログラム204 は記述子ファイル96から複数の構成要素記述子を読取る。注文92は変換モジュール94により記述子ファイル96へ変換される。構成要素記述子はターゲットシステム160 の構成要素のコンピュータの読取り可能な記述である。好ましい実施形態では、構成要素記述子はシステム記述子記録と呼ばれる記述子ファイル、即ちコンピュータの読取り可能なファイルに含まれており、これは各構成要素、即ちハードウェアおよび/またはソフトウェアの両者の構成要素のリストを含んでおり、ターゲットシステム160 へインストールされる。システム記述子記録はファイルサーバ202 上に直接記憶されてもよい。配列プログラム204 は、データベース100 から構成要素記述子に対応する複数のソフトウェア設置および/またはテストステップを検索する。ターゲットシステム160 の適切なソフトウェア設置および/またはテストステップを検索した後、配列プログラム204 は各ステップに対応するシーケンス番号にしたがって予め定められた順序でステップを配列する。ターゲットシステム160 に必要なステップを配列した後、配列プログラム204 はネットワーク接続 195、180 を経て予め定められた順序でターゲットシステム160 上でソフトウェア設置およびテストステップを実行する。ネットワーク接続200 はターゲットシステム160 の対応するポートへプラグ接続された一般的なネットワーク装置であることが望ましい。ネットワーク195 は技術でよく知られた任意の通信接続であってもよい。ソフトウェア設置および/またはテストステップの実行に続いて、設置およびテストの結果がネットワーク接続200 にわたってファイルサーバ202 へ戻され記録されるか、または適切なデータベース内に記憶される。図から明らかなように、図1のような分離したステップメカコンピュータシステム140 は必要ない。付加的に、ステップディスク150 も必要ではない。むしろブートターゲットシステム160 へ構成されているブートディスク220 だけが工場床面上のターゲットシステム160 に伴うことを必要とする。

【0018】

ソフトウェア設置およびテストシステムについて説明したが、以下、図1、2で説明されたシステム動作をさらに詳細に説明する。

【0019】

図3のaはコンピュータシステムに対する注文がコンピュータの読取り可能なシステムの記述子記録へ変換される好ましいプロセスを示している。特に、アイテム300でターゲットコンピュータシステムに対する注文が受信される。この注文は無数の形態の任意の1つでよい。例えば異なった注文フォーマットと、異なった注文配送機構が可能である。例えば、ターゲットコンピュータシステムの注文は電話、郵便によって、またはコンピュータネットワーク(例えばインターネット)によって行われる。注文を取る手段または注文の形態に関係なく、注文は、顧客が購入を希望しているターゲットコンピュータシステムのタイプおよび、顧客がターゲットコンピュータシステムが含んでいることを期待する特定の構成要素の明白なリストを含んでいる。注文の受信後、制御は送信モジュール310へ移行し、その期間にターゲットコンピュータシステムの注文がコンピュータネットワークによって、ターゲットコンピュータシステムを製造する製造システム(図示せず)へ送信される。ターゲットコンピュータシステムの注文はソフトウェア設置およびテストシステムへも与えられ、そこで注文はモジュール320の変換プログラムへ送られる。モジュール310で使用されるコンピュータネットワークは技術で知られた任意のタイプのものでよい。

【0020】

変換プログラムはターゲットコンピュータシステム注文を、製造プロセスで有効な記録へ変換する。特に、変換プログラムはモジュール330でコンピュータ注文を最初にしてB A

10

20

30

40

50

Rファイルと呼ばれる記録へ変換する。好ましくはB A Rファイルは製造された特定のターゲットコンピュータシステムを識別する特有の識別子を含んでいる。B A Rファイルはまた構成要素の詳細なリストを含んでおり、これはターゲットシステムに含まれているハードウェアおよびソフトウェアの両者を含んでいてもよい。さらにB A Rファイルは各構成要素に対する製造業者の特定の部品番号または他の有効な識別子を含んでいることが望ましい。最後に、B A Rファイルは氏名、住所、電話番号等の顧客特定情報を含んでもよい。

【 0 0 2 1 】

モジュール330 でB A Rファイルを生成した後、システム記述子記録がモジュール340 で生成される。好ましい実施形態ではシステム記述子記録はコンピュータの読取り可能なファイルであり、これはターゲットコンピュータシステムに含まれるハードウェアおよびソフトウェア構成要素を記述したものである。好ましい実施形態では、システム記述子記録はターゲットシステムの構成要素のリストをフォーマットで含んでおり、このリストはハードウェアタグ、ソフトウェアタグ、情報タグ、コメントを含んでいる。ハードウェアタグは配列プログラム204 に対して、タグに続く情報がハードウェア構成要素に関するものを識別する。同様に、ソフトウェアタグはタグに続く情報がソフトウェア構成要素に関するものと識別する。情報タグは一般的な情報が後続することを示している。コメントは種々のステートメントが配列プログラム204 で無視されるシステム記述子記録へ含まれることを可能にする。システム識別子記録は人による読取りが可能で理解しやすいテキストファイルであることが望ましい。このようなファイルはソフトウェア設置およびテストプロセスの故障点検とメンテナンスを容易に可能にする。システム記述子記録は特有のセットのトークンに対応する特有の識別子のリスト、例えば簡単な例ではシステム記述子記録は部品番号のリストであってもよいことが認識されよう。

【 0 0 2 2 】

図3のbはターゲットコンピュータシステム注文350 と、対応するB A Rファイル360 と、対応するシステム記述子記録370 の1例を示している。ターゲットコンピュータシステム注文350 はコンピュータファミリの名称、この図ではファミリ“X”を含んでいる。またターゲットコンピュータシステム注文350 にはPentium（商標名）プロセッサ、ハードドライブ、モニタを含む3つの例示的なハードウェア構成要素が含まれている。B A Rファイル360 は図3のaのモジュール320 で示されているような変換プログラムを経てターゲットコンピュータシステム注文350 を行うことにより生じる。B A Rファイル360 はファミリX内の特定のターゲットコンピュータシステムの特有の識別子を含んでいる。B A Rファイル360 はまたターゲットコンピュータシステム注文中にリストされた各構成要素の製造業者特定部品番号を含んでいる。さらにB A Rファイル360 はターゲットコンピュータシステムに含まれる各構成要素の所望な量と、各構成要素のテキスト記述を示す識別子を含んでいる。システム90はシステム記述子記録370 を生成するためB A Rファイル360 を使用する。

【 0 0 2 3 】

示されているように、システム記述子記録370 はまたファミリX内の特定のターゲットコンピュータシステムに対する特定の識別子を含んでいる。さらに、システム記述子記録370 は適切なタグを含んでおり、ここではプロセッサ、ハードドライブ、モニタはソフトウェアではなく全てハードウェア構成要素であることを示している。システム記述子記録370 はこれらの構成要素をテキスト記述で示している。さらに例示的なシステム記述子記録370 はソフトウェアタグを含んでおり、これはあるソフトウェアがファミリXに属するターゲットコンピュータシステムにインストールされ、あるいはテストされることを示している。例えばソフトウェアタグは、Pentium（商標名）プロセッサに適切なある動作システムが常にファミリXに属するターゲットコンピュータシステムのハードドライブにインストールされることを示している。

【 0 0 2 4 】

図4ではソフトウェア設置およびテストステップの順序を定めて配列する一般的な好まし

10

20

30

40

50

い方法が説明されている。モジュール400 では、ターゲットコンピュータシステムの特有の識別子がターゲットコンピュータシステム160 に対して生成される。図1で示されている実施形態では、ステップメカコンピュータシステム140 の前に着席している使用者は特有の識別子(例えばトラッキングコードとして機能するBAR識別子)をステップメカ140 の配列プログラム204 へ提供する。その代わりに、図2の実施形態ではターゲットコンピュータシステムの注文が受信された後に、特有の識別子が自動的に配列プログラム204 へ読込まれる。

【0025】

モジュール410 では、BAR識別子に対応するシステム記述子記録が位置を定められる。図1の実施形態では、ネットワーク接続110 またはネットワーク接続195 がシステム記述子記録の位置を定める。図2の実施形態ではネットワーク接続195 はシステム記述子記録の位置を定める。モジュール420 では、位置を定められたシステム記述子記録は配列プログラム204 へ与えられる。図1の実施形態では、配列プログラムはステップメカコンピュータシステム140 に存在し、一方、図2の実施形態では配列プログラムはファイルサーバ202 に存在する。配列プログラム204 は(図1および2の)データベース100 と共に動作し、それによってターゲットコンピュータシステム160 に対するソフトウェア設置およびテストステップを配列する。特定のターゲットコンピュータシステムに対して適切なソフトウェア設置およびテストステップが一度配列されると、配列プログラム204 はモジュール430 で示されているように出力ファイルを生成する。

【0026】

図1で示されている実施形態では、出力ファイルは6つの別々のファイルでステップディスク150 (図1参照)へ書込まれることが好ましい。これらのファイルは(1)ステップファイル、(2)Setenv.bat ファイル、(3)Qt.txt ファイル、(4)Et.txt ファイル、(5)Etlast.txt、(6)Ft.txt ファイルを含んでいる。ステップファイルはターゲットコンピュータシステムが注文されているソフトウェア設置およびテストステップを実行するための適切な命令ラインのリストを含んだASCIIテキストファイルであることが好ましい。好ましい実施形態では、ステップファイルはまたループされてもよい命令を含んでいる。特に、ステップファイルは命令が、限定された数または反復数、或いは限定された長さの時間に対して繰返されることを可能にする。このようなフォーマットはソフトウェア設置またはテストステップが、計算された予め定められた方法で繰返されることを有効に可能にする。Setenv.bat ファイルはターゲットコンピュータシステムに環境変数を設定することが好ましい。動作モードで、ステップファイルとSetenv.bat ファイルのみが装着およびテストに必要であることが認識されよう。ステップファイルとSetenv.bat ファイルは、ターゲットコンピュータシステムのソフトウェア設置およびテストステップを実行するための適切な命令ラインのリストを含んだASCIIテキストスクリプトファイルである。Qt.txt、Et.txt、Etlast.txt、Ft.txt ファイルは全て、ターゲットコンピュータシステムの製造業者のクイックテスト(Qt)、拡張テスト1(Et)、拡張テスト2(Etlast)、最終テスト(Ft)のフェーズで、ターゲットコンピュータシステムに対する設置およびテストステップを実行するための適切な命令ラインのリストを含んだASCIIテキストファイルであることが好ましい。

【0027】

一方、図2の実施形態では、出力ファイルは図1で示されているステップディスクへ書込まれない。代わりに、出力ファイルはファイルサーバ202 またはファイルサーバ190 に存在し、ここではこれらはターゲットコンピュータシステム160 でのソフトウェア設置および/またはテストステップの実行を導くことに使用される。

【0028】

図5は図1、2で示されている配列プログラム204 の動作についてのより詳細な概略図である。モジュール500 では、ターゲットコンピュータシステム160 に対応するシステム記述子記録が配列プログラム204 に与えられる。モジュール500 では、構成要素記述子はシステム記述子記録から読取られる。各構成要素記述子はターゲットコンピュータシステム

のそれぞれの構成要素、即ちハードウェアまたはソフトウェアを記述している。

【 0 0 2 9 】

図 3 の b を参照すると、モジュール 370 で Pentium (商標名) プロセッサを含んでいるシステム記述子記録のラインは 1 例の構成要素記述子である。モジュール 520 では、配列プログラム 204 はターゲットコンピュータシステム 160 の複数の構成要素に対応する複数の得られたオブジェクトを例示する。好ましい実施形態ではこれらの得られたオブジェクトは、ターゲットコンピュータシステム 160 で動作される必要があるソフトウェア設置およびテストステップについての (データベース 100 から得られた) 情報を記憶するために使用される。モジュール 500 では、ターゲットコンピュータシステム 160 のそれぞれの構成要素に関連するソフトウェア設置およびテストステップがデータベース 100 から検索され、適切な得られたオブジェクト中に記憶される。図 1 の実施形態では、ステップはネットワーク接続 110 を経て検索され、図 2 の実施形態ではステップはファイルサーバ 202 から直接検索されてもよい。好ましい実施形態でデータベース 100 からステップが検索される方法を記述するためデータベースの好ましい構造の記述が必要である。

10

【 0 0 3 0 】

図 6 はデータベース 100 の設計を示している。データベース 100 は予め定められた順序によるソフトウェア設置および / またはテストステップのシーケンスをコンピュータシステムのファミリに関連付けている。さらにデータベース 100 はコンピュータシステムの構成要素をコンピュータシステムのファミリに関連づけるように構成されている。さらにデータベース 100 はソフトウェア設置および / またはテストステップをコンピュータシステムの構成要素に関連付ける。

20

【 0 0 3 1 】

データベース 100 は、関係型データベースであることが好ましい。データベース 100 は幾つかのテーブルを含んでおり、そのそれぞれは、上述のアソシエーションを生成するのに適切な属性を含んでいる。

【 0 0 3 2 】

データベース 100 は、ステップテーブル 102 、 Sys Family テーブル 104 、 Sys __ Step __ Seq テーブル 106 、構成要素テーブル 108 、 Sys __ Comp テーブル 112 、 Comp __ Step テーブル 114 を含んでいる。好ましい実施形態において、各テーブルは、属性および基本キーとして機能する下線を引かれた属性のリストを含んでいる。

30

【 0 0 3 3 】

ステップテーブル 102 は、一組のソフトウェア設置を含み、テストステップは全てのコンピュータファミリの異なる素子間で共用されている。好ましい構成において、ステップテーブル 102 は、Step I D 、 Phase 、 Name 、 Cmd 、 Cmd Type 、 After Code および Max Instance を含む属性を有している。Step I D は、各ソフトウェア設置あるいはテストステップに特有の識別番号である。Phase は、ステップが実行される製造の位相を示す。例えば Phase は、コンピュータシステム製造の 4 つの位相、すなわち、(1) クイックテスト、(2) 拡張テスト 1 、(3) 拡張テスト 2 、(4) 最終テストに対応するように選択された整数である。Name はステップを記述する名前を割当てるストリングである。Cmd は、ターゲットシステム 160 (図 1 および図 2 に示されている) においてソフトウェア設置あるいはテストステップを実行するための実行可能な命令ラインを割当てるストリングである。After Code は、ソフトウェア設置あるいはテストステップが実行された後に休止あるいはリブートが必要とされるかどうかを決定する識別子である。Max Instance は、ステップがランすることを許可された最大回数を示す識別子である。

40

【 0 0 3 4 】

Sys Family テーブル 104 は、属性 Sys I D において特定された識別整数でコンピュータシステムの各ファミリを識別する。Sys Family テーブル 104 にはまた、ファミリの名前を識別するストリングも含まれている。

【 0 0 3 5 】

Sys __ Step __ Seq テーブル 106 は、ステップテーブル 102 と Sys Family テーブル 104

50

との間の関係を含む関係型テーブルである。Sys__Step__Seqテーブル106 は、(Sys Family テーブル104 からの) コンピュータシステムの特定のファミリーに対する属性 Sys ID において特定されたファミリー識別整数と、そのファミリーに適切な特定のステップの組を識別する (ステップテーブル102 からの) 属性 Step ID において特定されたステップ識別整数と、シーケンス番号とを含んでいる。シーケンス番号は、特定のファミリーに関連したステップが行われる予め定められた順序を表わす属性 SeqNum 内に含まれていることが好ましい。テストエンジニアは、特定のターゲットシステムに最も有効となるように選択された順序で、各製造位相内で特有なシーケンス番号を割当てる。シーケンス番号を割当てる別の方法が使用されてもよいことが認識されるであろう。

【 0 0 3 6 】

構成要素テーブル108 は、製造されるコンピュータシステム内に含まれる全ての可能な構成要素を含んでいる。このテーブルの属性は、各構成要素に識別子を割当てる Comp ID 、各構成要素にストリングの名前を割当てる Name Desc 、および構成要素のタイプを示す (例えば、ハードドライブ、C D - R O M ドライブ等、) Class Id であることが好ましい。

【 0 0 3 7 】

Sys__Comp テーブル112 は、コンピュータシステムのファミリーとそのファミリーに含まれることのできる 1 組の構成要素との間の関係を含む関係型テーブルである。Sys__Comp テーブル112 の属性は、(Sys Family テーブル104 からの) 属性 Sys ID において特性されたコンピュータファミリー識別整数と、(構成要素テーブル108 からの) 属性 Comp ID において特定された構成要素識別整数とを含んでいる。

【 0 0 3 8 】

Comp__Step テーブル114 は、構成要素と、その構成要素に適切な 1 組のソフトウェア設置およびテストステップとの間の関係を含む関係型テーブルである。Comp__Step テーブル114 の属性は、(構成要素テーブル108 からの) 属性 Comp ID において特定された構成要素識別整数と、(ステップテーブル102 からの) 属性 Step ID において特定された構成要素識別整数とを含んでいる。

【 0 0 3 9 】

図 3 の b に示された例示的なターゲットコンピュータシステムは、ソフトウェア設置およびテストステップを検索するために先に概要を説明されたデータベース設計がどのように使用されるかを説明するために使用される。システム記述子記録識別ファミリー X におけるコンピュータファミリー識別子は、Sys Family テーブル104 におけるファミリー X に対応する Sys ID に関連している。構成要素テーブル108 は、ターゲットコンピュータシステムのオーダーに記載されたターゲットコンピュータシステムの構成要素がリーガルであるかどうかを検査するために使用される。換言すると、順序付けプログラムおよびデータベースは、図 3 の b のシステム記述子記録中に含まれたプロセッサ、ハードドライブ、モニタおよびソフトウェアが、構成要素テーブル108 中の Comp ID によって特定された対応するエントリおよび対応する整数を有しているかどうかを決定する。構成要素がリーガルでない場合 (例えば、システム記述子記録中の構成要素が構成要素テーブル108 中に含まれていない場合等) 、エラーフラグが生じる。Sys__Comp テーブル112 は、構成要素テーブル108 および Sys Family テーブル104 からのマッピングを含む関係型テーブルである。Sys__Comp テーブル112 は、ファミリー X に属するターゲットコンピュータシステム上に含まれている全てのリーガルな構成要素を含んでいる。従って、Sys__Comp テーブル112 は、ターゲットシステムの全ての構成要素がリーガルであるかどうかを検査するために使用される。言い換えると、順序付けプログラムおよびデータベースは、図 3 の b のシステム記述子記録中に含まれたプロセッサ、ハードドライブ、モニタおよびソフトウェアが、Sys__Comp テーブル112 において対応する関係を有しているかどうかを決定する。構成要素がリーガルでない場合 (すなわち、システム記述子記録中の構成要素がファミリー X に属するターゲットシステム上に含まれている場合等、) エラーフラグが生じる。

10

20

30

40

50

【 0 0 4 0 】

関係型 Sys__Step__Seqテーブル106 において、ステップテーブル102 および Sys Family テーブル104 からのマッピングが存在している。Sys__Step__Seqテーブル106 は、ファミリー X に属するターゲットコンピュータシステム上で合法的にランする全てのソフトウェア設置およびテストステップを含んでいる。さらに、シーケンスおよび位相番号が各ソフトウェア設置およびテストステップに関連しているのは、この Sys__Step__Seqテーブル106 においてである。これらのシーケンスおよび位相番号は、コンピュータシステムの特定のファミリーに対してステップがランされるべき適切な順序を示す。それ故、Sys__Step__Seqテーブル106 は、ファミリー X のターゲットコンピュータシステム上でランされるステップのリストと、ステップが実行されるべき予め定められた順序を示すシーケンスおよび位相番号とを含んでいる。

10

【 0 0 4 1 】

Comp__Step テーブル114 は、構成要素テーブル108 およびステップテーブル102 からのマッピングを含む関係型テーブルである。Comp__Step テーブル114 は、ターゲットコンピュータシステムのプロセッサ、ハードドライブ、モニタおよびソフトウェアに対してランされるソフトウェア設置およびテストステップを含んでいる。

【 0 0 4 2 】

ターゲットシステムに含まれたそれぞれの構成要素に関連したソフトウェア設置およびテストステップの検索は、ターゲットコンピュータシステム160 の構成要素上でランされる中間セトリスティングステップを得るために Sys__Comp テーブル112 および Comp__Step テーブル114 上での接合動作の実行を含む。

20

【 0 0 4 3 】

接合動作の結果、図 3 の b に示されたシステム記述子記録中に記入されたプロセッサ、ハードドライブ、モニタおよびソフトウェア上でステップのリストがランされる。その後、Sys__Comp テーブル112 および Comp__Step テーブル114 の接合の結果は、ファミリー X に対する全てのステップを含む Sys__Step__Seqテーブル106 と接合される。この接合動作の結果はシーケンス番号および位相番号の形態のシーケンス情報を含み、シーケンス番号は特定の位相内の特有のものである。従って、Sys__Comp テーブル112 、 Comp__Step テーブル114 および Sys__Step__Seqテーブル106 の 3 つのテーブルの接合は、適切なソフトウェア設置およびテストステップ、ならびにシーケンスおよび位相番号の形態のシーケンス情報を生じ、それによってターゲットコンピュータシステム160 におけるソフトウェアのインストールおよび / またはテストをする。

30

【 0 0 4 4 】

第 1 の接合動作 (Sys__Comp テーブル112 と Comp__Step テーブル114 の接合) の結果が空セットである場合には、エラー状態が生じ、ターゲットシステムに含まれる構成要素はシステム記述子記録上に記入されたファミリーには属していないという信号を送る。この例は 1 つの例示である。システム記述子記録はファミリー Y に属するターゲットコンピュータシステムを正確に示しているとする。しかしながら、ファミリー X 中のターゲットシステムにのみ属するハードドライブ (ハードドライブ Z) がファミリー Y 中のターゲットシステム上に含まれていることをシステム記述子記録が不正確に示していると仮定する。そのような場合、Comp__Step テーブル114 は、ハードドライブ Z に関連したステップを含んでいる。Sys__Comp テーブル112 は、ファミリー Y に関連した構成要素を含んでいる。従って、Comp__Step テーブル114 を Sys__Comp テーブル112 に接合すると空のセットが生成され、それはハードドライブ Z がファミリー Z に関連した構成要素ではないためである (代りに、ファミリー X だけには関連している) 。上述の例から明らかであるように、データベースの好ましい設計によって、あるファミリーのターゲットシステムがそのファミリーに適切な構成要素だけを含んでいることが有効に確実にされる。

40

【 0 0 4 5 】

再び図 5 を参照すると、ターゲットシステム中に含まれる構成要素に関連したステップが検索された後、順序付けプログラム204 は、システム記述子記録を読取り、ターゲットシ

50

ステムに含まれる構成要素に対応する環境ファイルを生成することによってモジュール560中のターゲットコンピュータシステムに対する環境変数を処理する。例えば、図3のbに示されたシステム記述子記録が読取られ、システム記述子記録のプロセッサハードウェア構成要素に対応する“セットcpu=pentium”のような環境変数が用意される。

【0046】

図5のモジュール570において、上述の3つのテーブルの接合によって検索された、複数の検索されたソフトウェア設置およびテストステップが予め定められた順序で順序付けされる。この順序付けはそれぞれのシーケンス番号および位相番号にしたがい、ステップシーケンスを提供する。シーケンスそれ自体は、技術においてよく知られた多くの分類アルゴリズムの任意の1つを使用して達成される。

10

【0047】

モジュール580において、順序付けプログラム204はファイルを出力する。先に言及したように、出力ファイルは、図1に示された実施形態の6個の別々のファイル中のステップディスク150（図1参照）に書き込まれることが好ましい。それらのファイルは、（1）ステップファイル、（2）Setenv.batファイル、（3）Qt.txtファイル、（4）Et.txtファイル、（5）Etlast.txt、（6）Ft.txtファイルを含んでいる。ステップファイルはASCIIテキストファイルであることが望ましい。好ましい実施形態において、ステップファイルはループされる命令も含んでいる。特に、命令はステップファイルによって定められた数または反復回数あるいは定められた長さの時間だけ反復される。Setenv.batファイルは、ターゲットコンピュータシステムに環境変数を設定する。ステップファイルは、ターゲットコンピュータシステムの製造のクイックテスト（Qt）、拡張テスト1（Et）、拡張テスト2（Etlast）および最終テスト（Ft）の位相の期間中にそれぞれ実行されるステップを含んでいる。反対に、図2の実施形態において、出力ファイルは図1に示されているようなステップディスクには書き込まれない。その代りに、出力ファイルはファイルサーバ202あるいはファイルサーバ190上にあり、そこにおいてそれらはターゲットコンピュータシステム160上でのソフトウェア設置およびテストステップの実行を指導するために使用できる。

20

【0048】

再び図1および図2を参照すると、矢印130は、データベース100に変更が行われたことを示している。例えば、コンピュータシステムの新しいファミリーが生成された場合、それに応じてデータベース100を変更してもよい。特に、新しいファミリーはSysFamilyテーブル104のSysID中に新しいファミリー識別子を割当てられ、新しいファミリーの名前はSysFamilyテーブル104のネーム属性に割当てられる。ソフトウェア設置ステップおよびテストステップのリストはSys__Step__Seqテーブル106に加えられ、これらのステップは、新しいコンピュータシステムファミリーに関して、どのステップがランされる必要があるか、および予め定められたどの順序でランされる必要があるかを示す。コンピュータシステムの新しいファミリーが既存のファミリーと幾つかの類似点を共有している場合、Sys__Step__Seqテーブル106中の既存のファミリーに対するエントリは変更され、新しいファミリーに対するエントリが生成されることが可能である。任意の新しいステップがコンピュータシステムの新しいファミリーに対して生成される必要がある場合、これらのステップはステップテーブル102に加えられる。同様に、任意の新しい構成要素がコンピュータシステムの新しいファミリーを伴うと、これらの構成要素は構成要素テーブル108に加えられる。Comp__Stepテーブル114は、コンピュータシステムの新しいファミリーの各構成要素をそのソフトウェア設置およびテストに適切なステップに関連付けるために更新される。新しいファミリーがデータベース中に既に存在している構成要素だけを使用する場合、このテーブルは変更される必要はない。Sys__Compテーブル112は、新しいファミリー上に含まれていてもよい許容された構成要素のリストがデータベース中にあるように更新される。特に、新しいコンピュータシステムのSysIDを許容された各構成要素のCompIDと関連付ける必要がある。また、これはコンピュータシステム

30

40

50

の古いファミリーの既存のエントリをコピーし、その後変更することによって実行することができる。

【 0 0 4 9 】

好ましい実施形態に従ってデータベースを構成する際に、所定の顕著な利点を与えられることは理解される。特に、データベースのモジュラ設計によって、コンピュータシステムの新しいファミリーに対するソフトウェア設置およびテストステップを利点を有して簡単に設定することができる。さらに、コンピュータシステムの特定のファミリーあるいは特定の構成要素に対するソフトウェア設置およびテストステップは、別のソフトウェア設置およびテストステップに関係なく変更されることができる。

【 0 0 5 0 】

図 7 は、本発明によるソフトウェア設置およびテストプロセスのモジュラー修正を可能にするために、システム記述子記録およびステップシーケンスがどのようにしてパッチされるかを示す。モジュール 600 において、システム記述子記録が生成される。モジュール 610 において、システム記述子記録は、システム記述子記録パッチを使用して修正される。好ましい実施形態において、このパッチは、特定のターゲットコンピュータシステム、コンピュータシステムの特定のファミリー、または特定の構成要素に対してパッチが生成されることを可能にするモジュラーである。例えば、製造業者が、ある日のコンピュータシステムのあるファミリーに対するあるブランドのハードドライブを別のものと置換することを所望した場合、置換されるべきハードドライブを含む全てのシステム記述子記録を修正し、モジュール 610 において置換を行うパッチが形成されることができる。モジュール 620 において、ステップシーケンスは上述されたように決定される。モジュール 630 において、ステップシーケンスはステップシーケンスパッチを使用して修正される。好ましい実施形態において、このパッチは、特定のターゲットコンピュータシステム、コンピュータシステムの特定のファミリー、または特定の構成要素に対してパッチが生成されることを可能にするモジュラーである。例えば、製造業者が、ある日、ある構成要素に対して別のものの前に 1 つのテストステップを実行することを所望した場合、その順序が修正されるステップを含む全てのステップシーケンスを修正し、モジュール 640 における実行順序を対応的に変更するパッチが形成されてもよい。

【 0 0 5 1 】

以下、ターゲットシステム 160 におけるステップシーケンスの実行を説明する。ソフトウェア設置およびテストステップは、ターゲットコンピュータシステムに対応したステップシーケンスを読み出し、解釈し、実行するプログラムを使用してターゲットコンピュータシステム 160 で実行される。好ましい実施形態において、このプログラムをランステップと呼んでおり、それは図 1 の実施形態ではステップディスク 150 上に配置され、図 2 の実施形態ではファイルサーバ 202 上に配置されている。

【 0 0 5 2 】

図 8 は、任意のソフトウェア設置およびテストステップが実行される前にステップファイルに含まれるステップシーケンスの一部分を示す。前述のように、ステップシーケンスは、ソフトウェアをインストールし、および、または注文組込みターゲットコンピュータシステムをテストする命令を含んでいる。さらに、ステップファイル中のステップシーケンスによって、命令は定められた回数だけ、或は定められた期間だけ繰返されることが可能になる。さらに、ステップファイルは、ランステッププログラムによって無視される注釈 (r e m a r k) を含んでいる。ステップファイルにおいて、マーク 800 はステップシーケンスのフィールドを分離するために使用される。項目 810 は、ターゲットコンピュータシステム 160 をテストする命令である。この命令は、例えばメモリをテストし、かつ少数のコンピュータシステムインターフェース (S C S I) 装置をテストする命令を含んでいる。図面から認められるように、各命令は、特定のテスト環境に適した “ - o ” のようなスイッチを含んでいてもよい。項目 820 は、ランステッププログラムによって無視される注釈である。項目 810 c は、時間でループされる命令である。好ましい形態において、“ begin__time__loop ” 命令は、ループの開始点を示す。“ end__time__loop ” 命令は、ル

10

20

30

40

50

ープの終了点を示す。“begin__time__loop”命令は、ループを通して為される繰返しの時間長を示すフィールドと組合せられる。ここにおいて、例えば命令 810 c は、1 時間 30 分間実行される。項目 810 d は、反復数にしたがってループされる命令である。好ましい実施形態において、“begin__iterate__loop”命令は、反復的なループが実行されるようにランステッププログラムに命令する。“end__iterate__loop”命令は、ループ命令の終了を信号で通知する。ここでは、命令 810 d が 3 回実行されている。

【 0 0 5 3 】

ランステッププログラムは、ステップシーケンスを実行したときに、ソフトウェア設置およびテストプロセスの容易な故障点検および追跡を有効に可能にするタイムスタンプ情報をステップファイル中に配置する。

10

【 0 0 5 4 】

図 9 は、ステップが実行された後の図 8 のステップシーケンスの一部を示す。示されているように、ランステッププログラムは、ステップシーケンス中にタイムスタンプ情報を挿入する。項目 830 は、メモリテストが開始した時を示し、項目 832 は、そのテストが終了した時を示す。項目 834 は、テストの最後の反復が始まった時を示す。項目 836 および 838 は、scsi H D テストが始まった時、およびそれが終わった時をそれぞれ示す。項目 840 は、反復的なループが 3 回実行されたことを確認する。最後に、項目 842 および 844 は、scsi C D テストの最後の反復が始まった時、およびそれが終わった時をそれぞれ示す。実行された命令の隣りにタイムスタンプ情報を挿入することにより、ソフトウェア設置およびテストプロセスの効率的な故障点検および追跡が有効に可能になる。

20

【 0 0 5 5 】

図 1 0 は、ランステッププログラムの好ましい全体的フローチャートを示す。ランステッププログラム 860 は、ランステップバッチファイル 870 によりループ形態で実行される。このランステッププログラム 860 は、ステップシーケンス中のステップを読み出して解釈し、実行されるべき命令をステップシーケンスからバッチファイル 870 中に書込む。その後、バッチファイル 870 が実行され、ターゲットコンピュータシステム 160 においてステップを実行する。ステップが終了すると、制御はバッチファイルからランステッププログラム 860 に戻され、その後このランステッププログラム 860 がステップシーケンスの次のラインを読み出して解釈する。

【 0 0 5 6 】

30

図 1 1 は、ランステッププログラムのさらに詳細なフローチャートを示している。モジュール 900 に示されているように、ランステッププログラムは、最初にファイル名 Re __Run.bat が存在しているか否かを調べるためにチェックする。Re __Run.bat ファイルは、任意の命令がステップシーケンスから実行される前に生成され、その命令が成功的に終了した後に削除される。Re __Run.bat の存在は、最後の命令実行が成功せずに終わったことをモジュール 900 中のランステッププログラムに示す。このようにして、Re __Run.bat は、実行開始の指示として機能する。Re __Run.bat が存在する場合、モジュール 904 において、オペレータは、ソフトウェア設置およびテストプロセスを続けるか否か、或はその代わりにオペレータが故障点検の実行を希望しているかどうかを質問される。オペレータが継続を選択した場合、Runstep.bat ファイルが再実行されるモジュール 928 を実行するように制御が移行する。(この状態は、いずれのオプションも肯定的に選択されなかった場合、省略オプションとなる。)故障点検オプションが選択された場合、技術的によく知られているように故障点検が実行される。

40

【 0 0 5 7 】

Re __Run.bat が存在しない場合、ランステッププログラムは、最後の命令が正しく終了したか否かを判断して決定し、モジュール 910 に制御が移行し、このモジュール 910 において、ステップファイルに含まれていることが好ましいステップシーケンスのラインが読出される。ランステッププログラムは、そのラインを読み出し、モジュール 912 に開始または終了タイムスタンプが存在しているか否かを判断して決定する。開始または終了タイムスタンプが存在している場合、ランステッププログラムは、このプログラムが読出して

50

るラインに対して開始タイムスタンプだけが存在しているか否かをモジュール914 において判断して決定する。開始タイムスタンプだけが存在している場合、ランステッププログラムは、モジュール916 においてソフトウェア設置またはテストステップが終わったばかりであると仮定し、モジュール918 において終了タイムスタンプを書入れる。終了タイムスタンプをファイルに書入れた後、制御はモジュール90に戻る。

【 0 0 5 8 】

ランステッププログラムが読出しているラインに対する開始タイムスタンプ以外のものが存在している場合、ランステッププログラムは、モジュール906 において開始および終了タイムスタンプの両方が存在しているか否かを判断して決定する。そうである場合、ランステッププログラムは、モジュール908 においてステップが実行されたと仮定し、制御がモジュール900 に戻る。ランステッププログラムがモジュール912 において開始または終了タイムスタンプのいずれにも遭遇しなかった場合、ランステッププログラムは、モジュール920 において開始タイムスタンプを書入れ、これが読出しているステップシーケンスのラインに関してステップを実行する準備をする。

【 0 0 5 9 】

モジュール922 において、ランステッププログラムは、実行されるべき命令がローカルドライブ（ステップファイルが、ローカルドライブがシステム中のどのドライブかを制御する）上に記憶されているか否かを判断して決定する。ローカルドライブは、例えばステップディスク、ターゲットシステムのハードドライブ、ターゲットシステムのRAMドライブ、またはネットワークドライブであってもよい。命令がローカルドライブ上に配置されていない場合、ランステッププログラムは、実行されるべきテストがネットワーク上のどこかのファイルサーバ上に含まれていると仮定する。ランステッププログラムは、モジュール932 においてこのプログラムが既にネットワークに接続されているか否かを判断して決定する。接続されていない場合、ランステッププログラムは、モジュール936 においてネットワークにログインする命令をRunstep.bat中にはめ込む。したがって、Runstep.batがネットワーク接続180 によってターゲットシステム160 においてステップを実行する前に、ネットワーク接続が形成される。

【 0 0 6 0 】

モジュール936 に続いて、制御はモジュール926 に移行する。ランステッププログラムは、それが既にネットワーク中に記録されている場合には、モジュール934 の期間中に、ネットワークにログインする命令をRunstep.batから除去し、ネットワーク接続が存在しているならば、付加的なログインステップは不要である。その後、制御はモジュール926 に移行する。実行されるべきステップがたまたまステップディスク150 上に位置している場合、ランステッププログラムは、ネットワーク中に記録する必要はない。したがって、ランステッププログラムは、モジュール924 においてネットワークにログインする命令をRunstep.batから除去する。その後、制御はモジュール926 に移行する。モジュール926 において、ランステッププログラムは、実行されるべき適切な命令をRunstep.batおよびRe __Run.bat中にはめ込む。そのようにはめ込まれた命令は、好ましくはステップファイルに含まれているステップシーケンスから採取される。モジュール928 において、ステップはRunstep.batを実行することによって実行され、それが成功的に実行された場合には、Re __Run.batは削除される。ステップが成功的に実行されなかった場合、Re __Run.batファイルは削除されず、制御が失敗状態929 に移行する。その後、ステップシーケンスから別のラインが読出されるように制御がモジュール900 に戻る。このプロセスは全てのソフトウェア設置およびテストステップが終了するまで続く。

【 0 0 6 1 】

ステップシーケンスの実行時に、ターゲットシステムはテストされ、ソフトウェアがインストールされる。図 1 の実施形態において、テストの選択番号はステップディスク150 から直接実行されてもよいが、テストの大部分はネットワーク接続180 によってファイルサーバ190 から実行される。ファイルサーバ190 からのテストを実行することによって、ステップディスク150 のようなフロッピーディスクの記憶容量によって課せられる限界がな

10

20

30

40

50

くなる利点を得られる。

【 0 0 6 2 】

図 2 の実施形態において、ステップは、ネットワーク接続180 によってファイルサーバ190 から実行される。ここではブートディスク220 であるフロッピーディスクは、ターゲットコンピュータシステム160 をブートするためだけに必要である。このようなシステムは、ソフトウェア設置およびテストプロセスを簡単化する利点がある。

【 0 0 6 3 】

図 1 および 2 を再度参照すると、矢印210 は、ソフトウェア設置およびテストからの結果が、ファイルサーバ190 またはファイルサーバ202 のいずれかに戻されて記録されることを示している。この結果には、全てのステップが成功的に終了したかどうか、およびどのタイプの故障（もし、あれば）が生じたかが含まれていることが好ましい。結果の記録には、ステップシーケンスの実行に続くステップファイルの修正されたバージョンの単なる保存または書き込みを含んでもよく、上述したように、ステップファイルはランステッププログラムによってタイムスタンプを付けられる。このようなシステムは、コンピュータシステム製造中に故障点検能力を高められる効果がある。

【 0 0 6 4 】

以上、本発明の特定の実施形態を図示および説明してきたが、当業者は本発明の技術的範囲を逸脱することなく変更および修正が為されてもよく、したがって添付された特許請求の範囲がこのような変更および修正の全てを本発明の技術的範囲内のものとして包含することを認識するであろう。

【図面の簡単な説明】

【図 1】ソフトウェア設置およびテストシステムを示す概略図。

【図 2】別の実施形態によるソフトウェア設置およびテストシステムの概略図。

【図 3】本発明にしたがってコンピュータ注文をシステムの記述子記録へ変換するためのフローチャート、並びに例示的なコンピュータシステム注文、ベース・アセンブリ・レコード（BAR）ファイルおよびシステム記述子記録の一部分。

【図 4】ステップシーケンスの生成および提供を示すフローチャート。

【図 5】ステップシーケンスの生成を示すさらに詳細なフローチャート。

【図 6】データベースの構成の概略図。

【図 7】システム記述子記録およびステップシーケンスを修正するためのフローチャート。

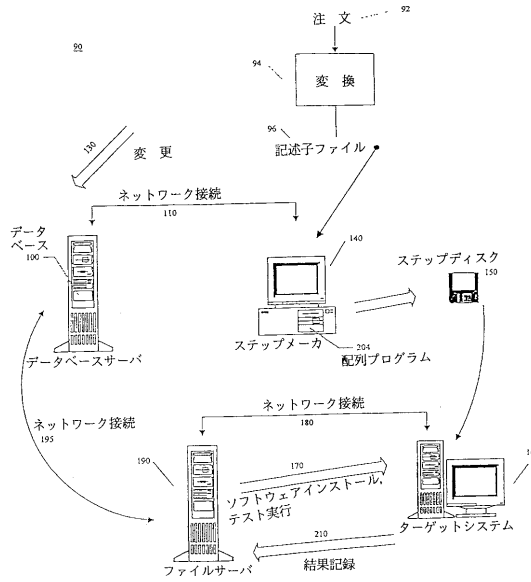
【図 8】実行される前のステップファイルの一例の概略図。

【図 9】実行された後のステップファイルの一例の概略図。

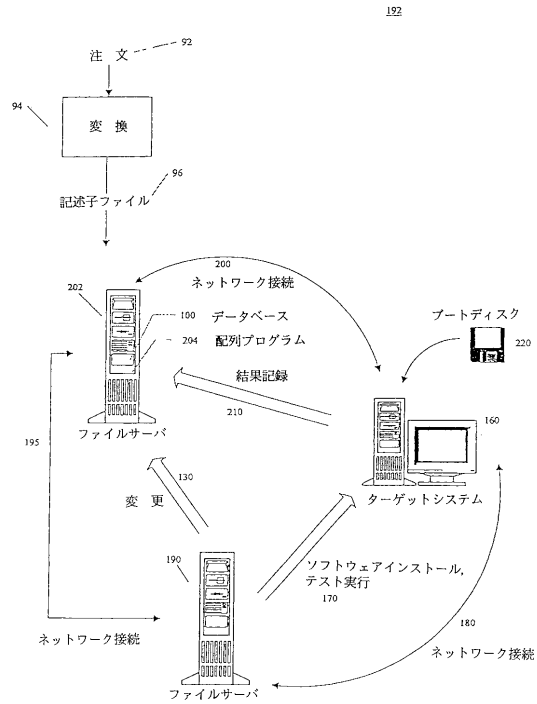
【図 10】ステップシーケンスを実行するためのプログラムの動作のフローチャート。

【図 11】ステップシーケンスを実行する図 10 のプログラムの動作のさらに詳細なフローチャート。

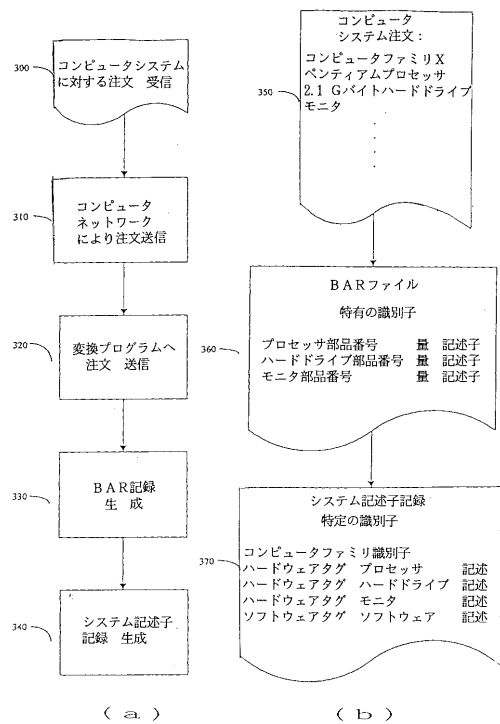
【図 1】



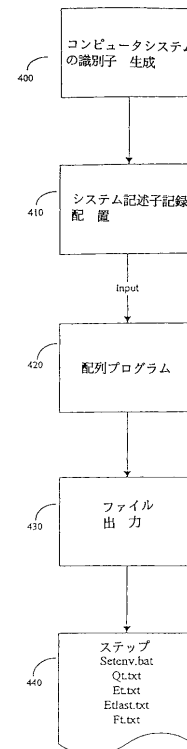
【図 2】



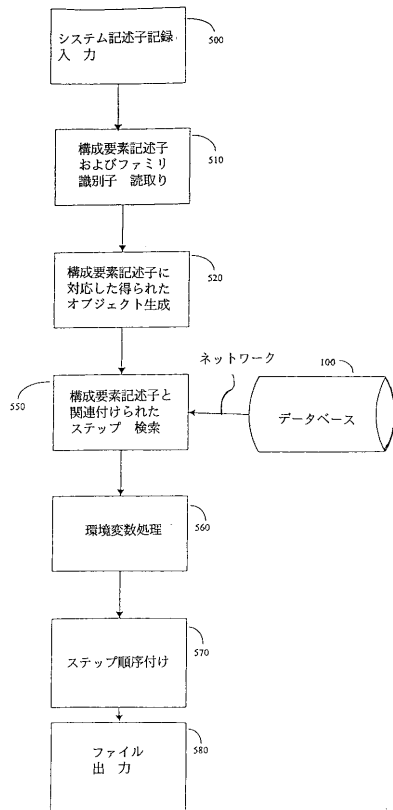
【図 3】



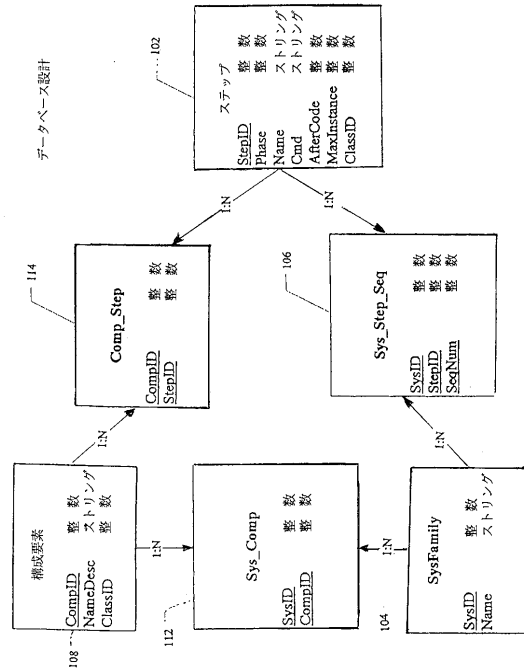
【図 4】



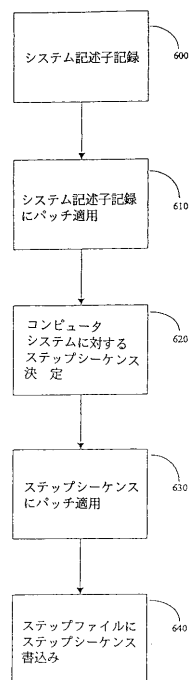
【図 5】



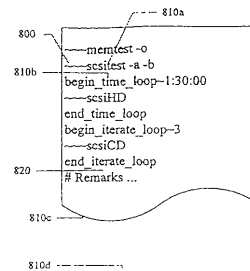
【図 6】



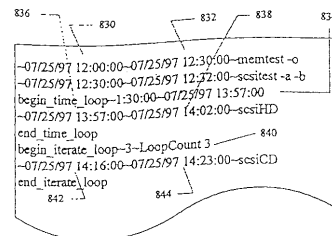
【図 7】



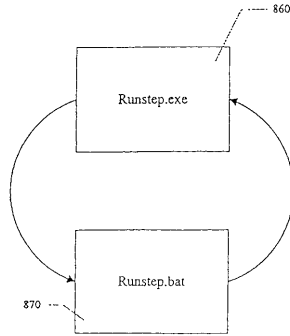
【図 8】



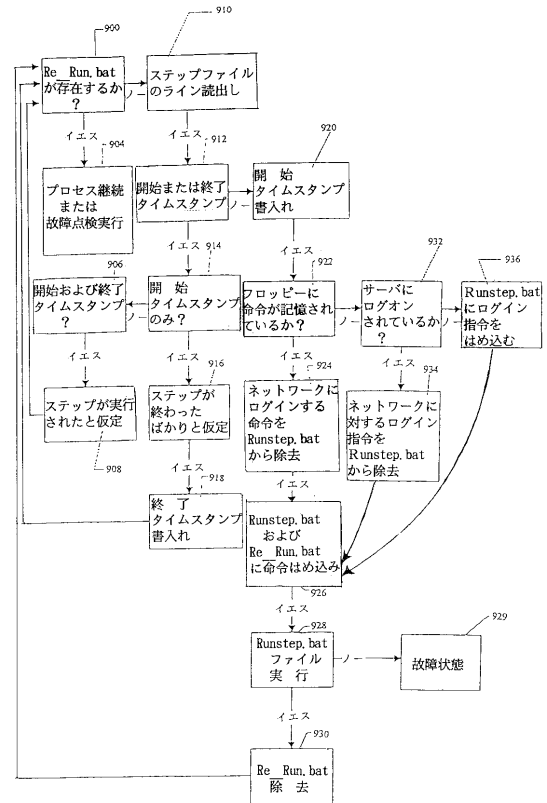
【図 9】



【図 10】



【図 11】



フロントページの続き

- (74)代理人 100088683
弁理士 中村 誠
- (74)代理人 100109830
弁理士 福原 淑弘
- (74)代理人 100095441
弁理士 白根 俊郎
- (74)代理人 100084618
弁理士 村松 貞男
- (74)代理人 100092196
弁理士 橋本 良郎
- (74)代理人 100100952
弁理士 風間 鉄也
- (72)発明者 リチャード・ディー・アムバーグ
アメリカ合衆国、テキサス州 7 8 7 5 3、オースティン、ゼラー・レーン 1 2 3 3 6
- (72)発明者 ロジャー・ダブリュ・ウォン
アメリカ合衆国、テキサス州 7 8 7 5 3、オースティン、ゼラー・レーン 1 2 4 2 7
- (72)発明者 マイケル・エー・ブランドリッジ
アメリカ合衆国、テキサス州 7 8 6 2 8、ジョージタウン、ドウ・ラン 4 2 8

審査官 須田 勝巳

- (56)参考文献 特開平 0 6 - 3 3 2 6 7 4 (J P , A)
特開平 0 7 - 0 6 4 7 7 1 (J P , A)
特開平 0 8 - 0 8 7 4 6 0 (J P , A)
特開平 8 - 2 3 4 9 6 7 (J P , A)
特開平 7 - 7 3 2 3 7 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F9/445

G06F11/22