

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5417264号  
(P5417264)

(45) 発行日 平成26年2月12日(2014.2.12)

(24) 登録日 平成25年11月22日(2013.11.22)

(51) Int. Cl.		F I			
<b>G06F 11/34</b>	<b>(2006.01)</b>	G06F 11/34		S	
<b>G06F 11/30</b>	<b>(2006.01)</b>	G06F 11/30		A	
<b>G06F 9/445</b>	<b>(2006.01)</b>	G06F 9/06	610A		

請求項の数 7 (全 40 頁)

(21) 出願番号	特願2010-140104 (P2010-140104)	(73) 特許権者	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(22) 出願日	平成22年6月21日(2010.6.21)	(74) 代理人	100100310 弁理士 井上 学
(65) 公開番号	特開2012-3647 (P2012-3647A)	(74) 代理人	100098660 弁理士 戸田 裕二
(43) 公開日	平成24年1月5日(2012.1.5)	(72) 発明者	小林 恵美子 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所 システム開発研究 所内
審査請求日	平成24年3月23日(2012.3.23)	(72) 発明者	工藤 裕 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所 システム開発研究 所内

最終頁に続く

(54) 【発明の名称】 分析情報提供方法

(57) 【特許請求の範囲】

【請求項1】

複数の第1コンピュータと分析コンピュータとを有する計算機システムにおける分析情報提供方法であって、

前記複数の第1コンピュータに含まれ、所定のアプリケーションをインストール済みかつ起動済みである、一つ以上の第1コンピュータは、前記所定のアプリケーション起動以前に行った複数の構成変更の情報及び前記所定のアプリケーションの起動結果の履歴を含むログを、前記分析コンピュータに送信し、

前記分析コンピュータは、前記ログを受信し、前記ログに基づいて、各構成変更後の前記所定のアプリケーションの起動結果を特定し、構成変更の種別毎に、各種別の構成変更後の前記所定のアプリケーション起動のうち、失敗した回数の割合である起動失敗率を計算し、

前記複数の第1コンピュータに含まれる、前記一つ以上の第1コンピュータ以外の他の第1コンピュータが：

(1) 前記分析コンピュータから前記所定のアプリケーションに関する構成変更の種別毎の起動失敗率を含む第1の情報を受信し、

(2) 前記起動失敗率に基づいて、前記所定のアプリケーションの起動が失敗する原因となる構成変更の種別を表示する、

ことを特徴とした分析情報提供方法。

【請求項2】

請求項 1 記載の分析情報提供方法であって、  
前記第 1 の情報に含まれる起動失敗率は、前記分析コンピュータが前記ログに基づいて検知した全ての構成変更の種別の一部の種別の各々についての起動失敗率であって、前記一部の種別は、前記起動失敗率に基づいて前記分析コンピュータが選択した、ことを特徴とした分析情報提供方法。

【請求項 3】

請求項 2 記載の分析情報提供方法であって、  
前記第 1 の情報には前記起動失敗率を計算したときを示す集計時間を含み、  
前記他の第 1 コンピュータは、前記集計時間を表示する、  
ことを特徴とした分析情報提供方法。

10

【請求項 4】

請求項 3 記載の分析情報提供方法であって、  
前記分析コンピュータは、前記ログに基づいて前記構成変更の種別毎に、当該構成変更後に前記所定のアプリケーションの起動が成功した割合である起動成功率を計算し、  
前記他の第 1 コンピュータは、  
（3）前記分析コンピュータから前記所定のアプリケーションに関する構成変更の種別毎の起動成功率を含む第 2 の情報を受信し、  
（4）前記起動成功率に基づいて、前記所定のアプリケーションの起動が成功する原因となる構成変更の種別を表示する、  
ことを特徴とした分析情報提供方法。

20

【請求項 5】

請求項 4 記載の分析情報提供方法であって、  
前記他の第 1 コンピュータは、前記所定のアプリケーション以外の複数のアプリケーションをインストール済みであり、  
前記他の第 1 コンピュータは、前記全ての構成変更の種別に含まれる所定の種別について、起動が失敗する可能性のあるアプリケーションを前記所定のアプリケーション及び前記複数のアプリケーションから選択し、選択したアプリケーションの識別子を表示する、  
ことを特徴とした分析情報提供方法。

【請求項 6】

請求項 5 記載の分析情報提供方法であって、  
前記（1）の処理は、前記所定のアプリケーションのインストールに関連して実行する、  
ことを特徴とした分析情報提供方法。

30

【請求項 7】

請求項 6 記載の分析情報提供方法であって、  
前記種別とは、構成項目又は / 及び変更タイプである、  
ことを特徴とした分析情報提供方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、概してコンピュータ内の原因分析に関し、より詳細には、構成の変更を分析することによって知識データベースを使用することなくアプリケーションの失敗の解決策を見つけるための構成の変更を伴う原因分析に関する。

40

【0002】

デスクトップ・コンピューティング環境の管理者にとってもっともストレスの大きい仕事の 1 つは、トラブルが発生した事例の原因分析（トラブルシューティング）である。原因分析はまた、呼者に対して解決策を提供する必要があるヘルプデスク要員にとっても決定的なものとなる。エンド・ユーザは、多くの種類のソフトウェアをインストールし、または OS の設定を変更する傾向にあり、それが問題を生じさせることがある。さらに、多くの自動アップ・グレード・プログラムがエンド・ユーザのコンピュータ内で日常的に走

50

っていることから、エンド・ユーザの自覚なしにコンピュータの構成が変更され得る。したがって、エンド・ユーザは、いつ構成が障害のあるものとなったかがわからず、いつから問題が生じていたかを思い出せないことがある。その種のデスクトップ・コンピューティング環境の管理者またはヘルプデスク要員は、エンド・ユーザに解決策を提供するために、自分たちの専門知識を用いてトラブルの背景を深く知る必要がある。

【0003】

現在の解決策は、たとえば、コンピュータ内のイベント・ログを遠隔調査するテクノロジー、構成項目およびそれらの変更履歴を収集し、ストアするテクノロジー、アプリケーションの起動を検出し、それらの起動履歴をストアするテクノロジー、過去の解決策における知識をストアするテクノロジー、および上に述べた情報を組み合わせることによって根本原因を演繹するテクノロジーを含む。

10

【0004】

特許文献1は、イベント・ログ、構成の変更、および知識データベースによって根本原因を演繹する例である。段落0134は、目標監視コンピュータからのエラー・ログ、イベント情報、および構成の変更の時系列データの収集を開示している。段落0137および図16、17、および18は、目標コンピュータ上におけるエラー状況と過去のデータにおけるその比較を開示している。

【0005】

イベント・ログを遠隔収集する例は、特許文献2および特許文献3を含む。特許文献4は知識ベースを使用する障害分析の例である。特許文献5はエラー・ログを使用する障害分析の例である。特許文献6は構成変更を遠隔取得する例である。

20

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2007 293393号

【特許文献2】米国特許第6,289,379号

【特許文献3】米国特許第5,857,190号

【特許文献4】米国特許第6,012,152号

【特許文献5】米国特許第6,598,179号

【特許文献6】米国特許出願公開第2007/0214193 A1号

30

【発明の概要】

【発明が解決しようとする課題】

【0007】

管理者またはヘルプデスク要員は、イベント・ログ、構成変更履歴、アプリケーション起動履歴、またはこれらの類を精査することによって解決策を提供する種類の知識を必要とする。知識は、他者によって記述された『原因』および『解決策』を与える知識データベースから獲得可能である。知識データベースを誰かが最新に維持する必要があることから、これは維持費用を必要とする。

【0008】

本発明のもう一つの課題は、エンドユーザが構成変更を行う際に、知識データベースの使用を伴うことなく、構成変更に関連するソフトウェアの問題分析情報を提供することである。

40

【課題を解決するための手段】

【0009】

本発明の例示的な実施態様は、知識データベースの必要性を伴うことなく、どの構成変更が問題を生じさせたかを決定するテクニックを提供する。したがって本発明は、根本原因を提供しないが、取り除かれる必要がある『容認限界』（すなわち、最終回答）を提供する。本発明は、『根本原因』指向と言うよりは、むしろ解決策指向の問題を取り扱う直接的な方法を提供する。

【0010】

50

エンド・ユーザが問い合わせを行う。ヘルプデスクが分析を開始する。最初のステップで目標期間を検出する。目標期間の検出のために、原因分析プログラムが、アプリケーション起動の最後の成功およびアプリケーション起動の最初の失敗を、イベント・ログおよびアプリケーション起動履歴両方に基づいて検出する。構成変更の検出については、構成変更が、構成変更履歴と目標期間検出の結果の組み合わせによって決定される。これらの構成変更が、アプリケーション起動に影響を与えたおそれがある。これらの構成変更の1つが容認限界であることになる。次のステップは、ほかのコンピュータのチェックである。原因である可能性がもっとも高い構成変更を決定するために、原因分析プログラムは、同じ構成変更を経験したほかのコンピュータをチェックする。原因分析プログラムは、各構成変更が行われる前および後のアプリケーション起動の結果をチェックし、カウントする。当該ほかのコンピュータ内において同じ構成変更が見つかったら、プログラムは、そのコンピュータ内において各構成変更が同じ問題を生じさせたか、または修復したかをチェックする。プログラムは、すべてのコンピュータについて類似の事例をカウントする。その後続いて、プログラムは、各構成変更についてのすべてのインスタンスに対する成功から失敗への変化を伴うインスタンスの比および失敗から成功への変化を伴うインスタンスの比を計算する。原因分析プログラムは、その後その結果を表示する。分析の結果は、図表を用いてランク付けの形式で示される。ヘルプデスクは、いずれの構成変更がもっとも影響を受けやすいかについて質問に答えることができる。

10

## 【0011】

本発明は、人間によって記述される知識データベースをまったく使用しない。ユーザが根本原因を知った場合であっても、どのようにしてそれを容易に修復するかをユーザが知らないこともあるため、根本原因を見つけることを求めない。それに代えて本発明は容認限界を識別する。ユーザが行うべきは、その容認限界を取り除くことである。ユーザにとっては、問題を修復する方が、根本原因が何かを告げられるより効果的である。

20

## 【0012】

本発明の態様は、複数のコンピュータのうち目標コンピュータのための原因分析の方法に指向されており、それにおいて目標コンピュータは、第1の失敗時においてコンピュータ・アプリケーションのアプリケーション起動失敗を、第1の失敗時に先行する第1の成功時においてコンピュータ・アプリケーションのアプリケーション起動成功を、第1の成功時と第1の失敗時の間の第1の時間期間内にコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している（たとえば図14参照）。この方法は、（1）コンピュータ・アプリケーションの第1の時間期間の間に生じた1つまたは複数の第1の構成変更を識別するステップ、および（2）原因構成変更分析（A）または修復構成変更分析（B）のうち少なくとも1つを実行するステップを包含する。分析（A）は、それらの複数のコンピュータのうち目標コンピュータを除くほかのすべてのコンピュータの各々について、前記ほかのすべてのコンピュータの各々が、第2の成功時において同一のコンピュータ・アプリケーションのアプリケーション起動成功を、第2の成功時より後の第2の失敗時において同一のコンピュータ・アプリケーションのアプリケーション起動失敗を、第2の成功時と第2の失敗時の間の第2の時間期間内に同一のコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している、ほかのアプリケーション起動失敗のインスタンスを識別すること、および第2の時間期間の間に生じた1つまたは複数の第2の構成変更を識別すること、および当該1つまたは複数の第2の構成変更のそれぞれについての合計の原因構成変更数を、目標コンピュータを除いた複数のコンピュータのすべてについて合計することによって獲得することを含む（たとえば図17参照）。分析（B）は、それらの複数のコンピュータのうち目標コンピュータを除くほかのすべてのコンピュータの各々について、前記ほかのすべてのコンピュータの各々が、第3の失敗時において同一のコンピュータ・アプリケーションのアプリケーション起動失敗を、第3の失敗時より後の第3の成功時において同一のコンピュータ・アプリケーションのアプリケーション

30

40

50

起動成功を、第3の失敗時と第3の成功時の間の第3の時間期間内に同一のコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している、アプリケーション起動成功のインスタンスを識別すること、および第3の時間期間の間に生じた1つまたは複数の第3の構成変更を識別すること、および当該1つまたは複数の第3の構成変更のそれぞれについての合計の修復構成変更数を、目標コンピュータを除いた複数のコンピュータのすべてについて合計することによって獲得することを含む(たとえば図19参照)。

【0013】

本発明の別の態様は、原因分析のシステムに指向されており、当該システムは、目標コンピュータを含む複数のコンピュータ、およびそれら複数のコンピュータに接続された分析コンピュータを有する。分析コンピュータは、上記のステップ(1)および(2)を実行するべくプログラムされる。いくつかの実施態様においては、分析コンピュータが上記複数のコンピュータのうちの1つになる(たとえば図25参照)。

10

【0014】

本発明の別の態様は、上記のステップ(1)および(2)を実行するべくデータ・プロセッサをコントロールするための複数のインストラクションをストアしているコンピュータ可読ストレージ媒体に指向されている。

【0015】

いくつかの実施態様においては、方法がさらに、原因構成変更結果(A1)または修復構成変更結果(B1)のうちの少なくとも1つについての結果を呈示するステップを包含する。ステップ(A1)においては方法が、1つまたは複数の第2の構成変更のそれぞれについて、前記ほかのすべてのコンピュータの各々について識別されたアプリケーション起動失敗のインスタンスの数、および前記ほかのすべてのコンピュータの各々のための1つまたは複数の第2の構成変更のそれぞれを伴うすべてのインスタンスの数をリストすることを包含する(たとえば図10参照)。ステップ(B1)においては方法が、1つまたは複数の第3の構成変更のそれぞれについて、前記ほかのすべてのコンピュータの各々について識別されたアプリケーション起動成功のインスタンスの数、および前記ほかのすべてのコンピュータの各々のための1つまたは複数の第3の構成変更のそれぞれを伴うすべてのインスタンスの数をリストすることを包含する(たとえば図12参照)。

20

【0016】

いくつかの実施態様においては、ステップ(1)および(2)が、複数のコンピュータ・アプリケーションについて実行される。さらに方法は、複数のコンピュータ・アプリケーションのそれぞれについて、原因構成変更結果(A2)または修復構成変更結果(B2)のうちの少なくとも1つについての結果を呈示するステップを包含する。ステップ(A2)においては方法が、1つまたは複数の第2の構成変更のそれぞれについて、前記ほかのすべてのコンピュータの各々について識別されたアプリケーション起動失敗のインスタンスの数、および前記ほかのすべてのコンピュータの各々のための1つまたは複数の第2の構成変更のそれぞれを伴うすべてのインスタンスの数をリストすること、および複数のコンピュータ・アプリケーションのそれぞれについて分析した日時をリストすることを包含する(たとえば図21参照)。ステップ(B2)においては方法が、1つまたは複数の第3の構成変更のそれぞれについて、前記ほかのすべてのコンピュータの各々について識別されたアプリケーション起動成功のインスタンスの数、および前記ほかのすべてのコンピュータの各々のための1つまたは複数の第3の構成変更のそれぞれを伴うすべてのインスタンスの数をリストすること、および複数のコンピュータ・アプリケーションのそれぞれについて分析した日時をリストすることを包含する。

30

40

【0017】

特定の実施態様においては方法が、さらに、指定されたコンピュータ・アプリケーションについて、整合原因構成変更分析(A3)または整合修復構成変更分析(B3)のうちの少なくとも1つを実行するステップを包含する。ステップ(A3)においては方法が、指定されたコンピュータ・アプリケーションと整合するコンピュータ・アプリケーション

50

についての (A2) の結果を整合する原因構成変更結果として検索し、当該整合する原因構成変更結果を分析のために取り出すことを含む (たとえば図22参照)。ステップ (B3) においては方法が、指定されたコンピュータ・アプリケーションと整合するコンピュータ・アプリケーションについての (B2) の結果を整合する修復構成変更結果として検索し、当該整合する修復構成変更結果を分析のために取り出すことを含む。

【0018】

いくつかの実施態様においては方法が、さらに、組み合わせ原因構成変更分析 (C) または組み合わせ修復構成変更分析 (D) のうちの少なくとも1つを実行するステップを包含する。ステップ (C) においては方法が、複数のコンピュータのうちの目標コンピュータを除くほかのすべてのコンピュータの各々について、前記ほかのすべてのコンピュータの各々が、第4の成功時において同一のコンピュータ・アプリケーションのアプリケーション起動成功を、第4の成功時より後の第4の失敗時において同一のコンピュータ・アプリケーションのアプリケーション起動失敗を、第4の成功時と第4の失敗時の間の第4の時間期間内に同一のコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している、ほかのアプリケーション起動失敗のインスタンスを識別すること、および第4の時間期間の間に生じた第4の構成変更の1つまたは複数の組み合わせを識別すること、および第4の構成変更の組み合わせのそれぞれについての合計の原因構成変更数を、目標コンピュータを除いた複数のコンピュータのすべてについて合計することによって獲得することを包含する (たとえば図24参照)。ステップ (D) においては方法が、複数のコンピュータのうちの目標コンピュータを除くほかのすべてのコンピュータの各々について、前記ほかのすべてのコンピュータの各々が、第5の失敗時において同一のコンピュータ・アプリケーションのアプリケーション起動失敗を、第5の失敗時より後の第5の成功時において同一のコンピュータ・アプリケーションのアプリケーション起動成功を、第5の失敗時と第5の成功時の間の第5の時間期間内に同一のコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している、アプリケーション起動成功のインスタンスを識別すること、および第5の時間期間の間に生じた第5の構成変更の1つまたは複数の組み合わせを識別すること、および1つまたは複数の第5の構成変更のそれぞれについての合計の修復構成変更数を、目標コンピュータを除いた複数のコンピュータのすべてについて合計することによって獲得することを包含する。

【0019】

いくつかの実施態様においては方法が、さらに、組み合わせ原因構成変更結果 (C1) または組み合わせ修復構成変更結果 (D1) のうちの少なくとも1つについての結果を呈示するステップを包含する。ステップ (C1) においては方法が、第4の構成変更の1つまたは複数の組み合わせのそれぞれについて、前記ほかのすべてのコンピュータの各々について識別されたアプリケーション起動失敗のインスタンスの数、および前記ほかのすべてのコンピュータの各々のための第4の構成変更の1つまたは複数の組み合わせのそれぞれを伴うすべてのインスタンスの数をリストすることを含む。ステップ (D1) においては方法が、第5の構成変更の1つまたは複数の組み合わせのそれぞれについて、前記ほかのすべてのコンピュータの各々について識別されたアプリケーション起動成功のインスタンスの数、および前記ほかのすべてのコンピュータの各々のための第5の構成変更の1つまたは複数の組み合わせのそれぞれを伴うすべてのインスタンスの数をリストすることを含む。

【0020】

本発明の別の態様は、複数のコンピュータのうちの目標コンピュータについての原因分析を実行するためのコンピュータ・システム内における方法に指向されており、それにおいて目標コンピュータは、第1の失敗時においてコンピュータ・アプリケーションのアプリケーション起動失敗を、第1の失敗時に先行する第1の成功時においてそのコンピュータ・アプリケーションのアプリケーション起動成功を、第1の成功時と第1の失敗時の間

10

20

30

40

50

の第1の時間期間内にそのコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している。この方法は、コンピュータ・アプリケーションの第1の時間期間の間に生じた1つまたは複数の第1の構成変更、および当該1つまたは複数の第1の構成変更のそれぞれの第1の構成変更に対応する絵図表であって、失敗率の領域および成功率の領域を有する絵図表をリストする原因構成変更テーブルを呈示するステップを包含する。失敗率の領域は、複数のコンピュータのうちの目標コンピュータを除くほかのすべてのコンピュータの各々について、前記ほかのすべてのコンピュータの各々が、第2の成功時において同一のコンピュータ・アプリケーションのアプリケーション起動成功を、第2の成功時より後の第2の失敗時において同一のコンピュータ・アプリケーションのアプリケーション起動失敗を、第2の成功時と第2の失敗時の間の第2の時間期間内に同一のコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している、ほかのアプリケーション起動失敗のインスタンスを識別する失敗事例の表現を示す。第2の時間期間の間には、テーブル上にリストされた対応する第1の構成変更等に等しい第2の構成変更が生じる。成功率の領域は、複数のコンピュータのうちの目標コンピュータを除くほかのすべてのコンピュータの各々について、前記ほかのすべてのコンピュータの各々が、第3の成功時において同一のコンピュータ・アプリケーションのアプリケーション起動成功を、第3の成功時より後の第3の失敗時において同一のコンピュータ・アプリケーションのアプリケーション起動失敗を、第3の成功時と第3の失敗時の間の第3の時間期間内に同一のコンピュータ・アプリケーションのそのほかのアプリケーション起動成功を伴うことなく、かつそのほかのアプリケーション起動失敗を伴うことなしに経験している、ほかのアプリケーション起動失敗のインスタンス以外を識別する成功事例の表現を示す。第3の時間期間の間には、テーブル上にリストされた対応する第1の構成変更等に等しい第3の構成変更が生じる。

10

20

#### 【0021】

いくつかの実施態様においては、上記の絵図表が棒グラフを包含し、失敗率の領域は、失敗事例の数または失敗事例および成功事例の両方の合計数と比較したときの失敗事例のパーセンテージのうちの少なくとも1つを示し、失敗成功領域は、成功事例の数または失敗事例および成功事例の両方の合計数と比較したときの成功事例のパーセンテージのうちの少なくとも1つを示す。原因構成変更テーブルは、1つまたは複数の第1の構成変更の構成項目および変更タイプ、1つまたは複数の第1の構成変更に対応する変更日時、および失敗率の領域および成功率の領域を示す絵図表をリストする。さらにこの方法は、構成項目、変更タイプ、変更日時、および絵図表の失敗率のうちの任意の1つに従って原因構成変更テーブルを並べ替えるための並べ替えキー指標をユーザに対して呈示するステップ、およびユーザにより入力された並べ替えキー指標の選択にตอบสนองしてユーザにより入力された選択に従って並べ替えられた原因構成変更テーブルを呈示するステップを包含する。

30

#### 【0022】

これらの、およびこのほかの本発明の特徴および利点は、以下の特定の実施態様の詳細な説明を検討することにより当業者に明らかなものとなるであろう。

40

#### 【図面の簡単な説明】

#### 【0023】

【図1】本発明の方法および装置の適用が考えられるクライアント サーバ・アーキテクチャのためのハードウェア構成の例を図解したブロック図である。

【図2】図1のアーキテクチャに適用される本発明の機能ブロックの例を図解したブロック図である。

【図3】本発明の基本概念を図解したアプリケーション起動結果と構成変更の間の関係の例を示す説明図である。

【図4】原因分析プログラムのユーザ・インターフェースの例を示した説明図である。

【図5】原因分析プログラムの結果スクリーンの例を示した説明図である。

50

【図6】分析コンピュータ内に常駐するイベント・ログ・テーブルの例を示した説明図である。

【図7】分析コンピュータ内に常駐する構成変更履歴テーブルの例を示した説明図である。

【図8】分析コンピュータ内に常駐するアプリケーション起動履歴テーブルの例を示した説明図である。

【図9】本発明の第1の実施態様に従って分析コンピュータ内に常駐する原因構成変更一時テーブルの例を示した説明図である。

【図10】分析コンピュータ内に常駐する原因構成変更テーブルの例を示した説明図である。

10

【図11】分析コンピュータ内に常駐する修復構成変更一時テーブルの例を示した説明図である。

【図12】分析コンピュータ内に常駐する修復構成変更テーブルの例を示した説明図である。

【図13】分析コンピュータ内に常駐するログ収集プログラムによって実行される場所のログ収集を一例で図解したフローチャートである。

【図14】本発明の第1の実施態様に従って分析コンピュータ内に常駐する原因分析プログラムによって実行される場所の原因分析を一例で図解したフローチャートである。

【図15】分析コンピュータ内に常駐する目標期間検出によって実行される場所の目標期間検出プロセスを一例で図解したフローチャートである。

20

【図16】分析コンピュータ内に常駐する起動結果チェックによって実行される場所のアプリケーション起動結果をチェックするプロセスを一例で図解したフローチャートである。

【図17】本発明の第1の実施態様に従って分析コンピュータ内に常駐する原因構成変更分析によって実行される場所の原因構成変更分析プロセスを一例で図解したフローチャートである。

【図18】図17の原因構成変更分析プロセスのサブルーチンを一例で図解したフローチャートである。

【図19】分析コンピュータ内に常駐する修復構成変更分析によって実行される場所の修復構成変更分析プロセスを一例で図解したフローチャートである。

30

【図20】図19の修復構成変更分析プロセスのサブルーチンを一例で図解したフローチャートである。

【図21】本発明の第2の実施態様に従って原因構成変更テーブルの例を示した説明図である。

【図22】本発明の第2の実施態様に従って原因分析プログラムによって実行される場所の原因分析を一例で図解したフローチャートである。

【図23】本発明の第3の実施態様に従って原因構成変更テーブルの例を示した説明図である。

【図24】本発明の第3の実施態様に従って原因構成変更分析によって実行される場所の原因構成変更分析プロセスを一例で図解したフローチャートである。

40

【図25】本発明の第4の実施態様に従って全体のシステムのハードウェア・アーキテクチャの構成、ソフトウェア・モジュール、およびテーブルの例を図解した説明図である。

【図26】第5の実施態様における分析コンピュータと目標コンピュータ内エージェントの機能ブロックの例を図解したブロック図である。

【図27】第5の実施態様における集計プログラムの処理の一例を図解したフローチャートである。

【図28】第5の実施態様におけるエージェントの問題情報を提供するユーザインタフェースの例を示した説明図である。

【図29】第5の実施態様におけるエージェントの対策情報を提供するユーザインタフェースの例を示した説明図である。

50



【図30】第5の実施態様におけるエージェントの問題情報を提供するユーザインタフェースの別の例を示した説明図である。

【図31】第6の実施態様における集計プログラムの処理の一例を図解したフローチャートである。

【図32】第6の実施態様における原因分析プログラムの処理の一例を図解したフローチャートである。

【発明を実施するための形態】

【0024】

以下の本発明の詳細な説明においては、開示の一部を形成し、かつ限定ではなく例証の手段として本発明を実施できる例示的な実施態様が示された添付図面を参照する。図面において、類似の番号は、いくつかの図を通じて実質的に類似の構成要素を記述する。さらに注意される必要があるが、この詳細な説明は、以下に述べられており、かつ図面内に図解されているとおりに多様な例示的な実施態様を提供するが、ここに述べられ、図解されている実施態様に本発明が限定されることはなく、当業者が知るとおり、または知ることになるとおり、ほかの実施態様への拡張が可能である。この明細書において『1つの実施態様』、『この実施態様』、または『これらの実施態様』と言及するときは、実施態様に関して述べられている特定の特徴、構造、または特性が本発明の少なくとも1つの実施態様に含まれている意味であり、この明細書の多様な箇所でのこれらの成句の出現は、必ずしもすべてが同一の実施態様を参照していない。それに加えて以下の詳細な説明においては、本発明の完全な理解を提供するために多くの特定の詳細が示されている。しかしながら当業者には明らかとなるが、これらの特定の詳細がすべて本発明の実施に必要とならないこともある。そのほかの状況においては、不必要に本発明を不明瞭にすることがないように周知の構造、材料、回路、プロセス、およびインターフェースが詳細に説明されていないか、かつ/またはブロック図形式で図解されていることがある。

【0025】

さらに、以下の詳細な説明のいくつかの部分は、コンピュータ内の動作のアルゴリズムおよび記号表現の形で与えられている。これらのアルゴリズム記述および記号表現は、データ処理分野の当業者によって自分たちの新機軸の本質をほかの当業者にもっとも効果的に伝えるために使用される手段である。アルゴリズムは、望ましい結末の状態または結果を導く一連の定義済みステップである。本発明においては、実行されるステップが、知覚可能な結果を達成するために知覚可能な量の物理的な操作を必要とする。必ずではないが、通常はこれらの量が、ストア、転送、結合、比較、およびその他の操作が可能な電気的または磁気的信号またはインストラクションの形式を取る。主として共通的慣習上の理由から、これらの信号をビット、値、要素、記号、文字、項、数、インストラクション、またはこれらの類として言及するとしばしば好都合であることが立証されている。しかしながらこれらの、および類似の用語のすべては適切な物理量に関連付けされるものであり、それらの量に適用される便宜上のラベルに過ぎないことを念頭に置く必要がある。特にほかに言及していない限り、以下の考察から明らかとなっており、この説明を通じて、『処理』、『コンピューティング』、『計算』、『決定』、『表示』、またはこれらの類といった用語を使用した考察が、コンピュータ・システムのレジスタおよびメモリ内の物理的な（電子的な）量として表されるデータを操作し、コンピュータ・システムのメモリまたはレジスタまたはその他の情報ストレージ、送信、または表示デバイス内の物理量として類似に表されるほかのデータに変換するコンピュータ・システムまたはその他の情報処理デバイスの動作およびプロセスを含むことが可能であると認識される。

【0026】

本発明は、これにおける動作を実行するための装置にも関係する。この装置は、求められている目的のために特別に構成されたもの、または1つまたは複数のコンピュータ・プログラムによって選択的に起動されるかまたは再構成される1つまたは複数の汎用コンピュータを含むものとすることができる。その種のコンピュータ・プログラムは、限定ではないが光ディスク、磁気ディスク、読み出し専用メモリ、ランダム・アクセス・メモリ、

10

20

30

40

50

ソリッド・ステート・デバイスおよびドライブ、または電子情報のストアに適したそのほかの任意タイプの媒体等のコンピュータ可読ストレージ媒体内にストアすることができる。ここで与えられているアルゴリズムおよび表示は、本質的に任意の特定のコンピュータまたはそのほかの装置に関係しない。多様な汎用システムを、この中にある教示に従ってプログラムおよびモジュールとともに使用してもよく、または望ましい方法ステップを実行する、より特化された装置を構成すると好都合なことも立証できる。それに加えて本発明は、いずれかの特定のプログラミング言語を参照して述べられていない。認識されることになろうが、多様なプログラミング言語を使用してここに述べられているとおりの本発明の教示を実装することができる。プログラミング言語（１つまたは複数）のインストラクションは、１つまたは複数の処理デバイス、たとえば中央処理ユニット（CPU）、プロセッサ、またはコントローラによって実行され得る。

10

## 【0027】

本発明の例示的な実施態様は、より詳細を以下に述べるとおり、知識データベースの使用を伴うことなしに構成変更を分析することによってアプリケーションの失敗に対する解決策を見つけるための装置、方法、およびコンピュータ・プログラムを提供する。

## 【0028】

## A. 第1の実施態様

## 1. システム構成

図1は、本発明の方法および装置の適用が考えられるクライアントサーバ・アーキテクチャのためのハードウェア構成の例を図解する。分析コンピュータ101および複数の目標コンピュータ102が、LAN 103を通じて接続されている。分析コンピュータ101は、CPU 111、メモリ112、ディスク113、ビデオ・インターフェース114、およびネットワーク・インターフェース115を含む汎用コンピュータである。各要素は、システム・バス116を通じて接続される。分析コンピュータ101は、そのメモリ112内に原因分析プログラム121およびログ収集プログラム122を有する。原因分析プログラム121は、CPU 111によって実行される目標期間検出131、原因構成変更分析132、修復構成変更分析133、および起動結果チェック134を含む。分析コンピュータ101は、そのディスク113内に原因構成変更一時テーブル144、修復構成変更一時テーブル145、原因構成変更テーブル146、修復構成変更テーブル147、およびログ情報123を有する。ログ情報123は、イベント・ログ・テーブル141、アプリケーション起動履歴テーブル142、および構成変更履歴テーブル143を含む。分析コンピュータ101は、LAN 103に接続されて複数の目標コンピュータ102からのログ情報171の収集に使用されるネットワーク・インターフェース115を有する。表示器117がビデオ・インターフェース114に接続され、原因分析プログラム121のユーザ・インターフェースおよび原因分析プログラム121による原因構成変更分析の結果の表示に使用される。

20

30

## 【0029】

目標コンピュータ102は、CPU 151、メモリ152、ディスク153、ビデオ・インターフェース154、およびネットワーク・インターフェース155を含む汎用コンピュータである。各要素は、システム・バス156を通じて接続される。目標コンピュータ102は、LAN 103を介して分析コンピュータ101にログ情報171を送信するエージェント161を有する。目標コンピュータ102は、そのディスク153内にログ情報171を有する。表示器157がビデオ・インターフェース154に接続されている。

40

## 【0030】

図2は、図1のアーキテクチャに適用される本発明の機能ブロック図の例を図解する。ログ収集プログラム122は、分析コンピュータ101内に常駐し、目標コンピュータ102内に常駐する各エージェント161と通信することによってログ情報171を収集し、その情報を、分析コンピュータ101内のログ情報123のイベント・ログ・テーブル141、アプリケーション起動履歴テーブル142、および構成変更履歴テーブル143

50

にストアする。

【0031】

原因分析プログラム121は、ログ情報123を読み出し、以下に示すとおり、原因構成変更分析を実行する。目標期間検出131は、イベント・ログ・テーブル141、アプリケーション起動履歴テーブル142、および構成変更履歴テーブル143を読み出し、特定のアプリケーションが問題なく起動できた時点と、そのアプリケーションが問題なく起動できなかった(=失敗した)時点の間における期間を検出する。その後目標期間検出131は、構成変更履歴テーブル143を参照することによって当該期間内の目標コンピュータ上の構成変更を決定する。原因構成変更分析132は、そのほかのコンピュータのログ情報123をチェックして結果を原因構成変更テーブル146にストアする。原因構成変更一時テーブル144は、原因構成変更分析132が原因構成変更を分析するときの一時データとして使用される。

10

【0032】

修復構成変更分析133は、修復構成変更を検出して結果を修復構成変更テーブル147にストアする。修復構成変更一時テーブル145は、修復構成変更分析133が修復構成変更を分析するときの一時データとして使用される。修復構成変更は、アプリケーション起動失敗等の問題のある状況を修復する構成変更である。起動結果チェッカ134は、イベント・ログ・テーブル141およびアプリケーション起動履歴テーブル142両方を参照することによって特定のアプリケーションが問題なく起動できたか否かを検出するサブルーチンである。

20

【0033】

図3は、本発明の基本概念を図解したアプリケーション起動結果と構成変更の関係の例301~304を示す。

【0034】

図式301は、この原因構成変更分析の目標コンピュータ102の状況を示す。図式301によれば、成功したアプリケーション起動と失敗した起動の間に4つの構成変更が生じている。これらの構成変更の間に、このほかの起動はない。したがってアプリケーション起動失敗は、これら4つの構成変更のうちの一つによって生じた可能性がある。図式302、303、および304は、そのほかのコンピュータの状況を示しており、それらは詳細な分析のために使用されることになる。

30

【0035】

図式302によれば、別のコンピュータAに同じ構成変更が生じているが、『VPN CLIENT v1.8』の削除および『VPN CLIENT v2.0』の追加のいずれもそのアプリケーションの起動に影響を与えていない。したがって、これら2つの構成変更がそのアプリケーションの起動に影響を与えた確信度はより低くなる。これに対して、『PRINTER DRIVER A』および『PATCH 2322』の追加が成功と失敗の間における結果をもたらしている。したがって、これら2つの構成変更がそのアプリケーションの起動に影響を与えた確信度はより高くなる。

【0036】

図式303によれば、『PRINTER DRIVER A』の追加が成功と失敗の間における結果をもたらしている。したがって、この構成変更がアプリケーションの起動に影響を与えた確信度がさらに高くなる。さらに失敗の後の『PRINTER DRIVER A』の削除が失敗と成功の間における結果をもたらしている。したがって『PRINTER DRIVER A』の削除が問題を修復したと見られる。この構成変更がアプリケーションの起動を修復したであろうという確信度がより高くなる。それに加えて『PATCH 2322』の追加は、成功と成功の間における結果をもたらしていることから、この構成変更がアプリケーションの起動に影響を与えた確信度はより低くなる。

40

【0037】

図式304によれば、『PATCH 1234』の追加が失敗と成功の間にあり、したがって『PATCH 1234』の追加が問題を修復したと見られる。この種類の観察は

50

、2つの種類の結果を導くことが可能である。一方は、どの構成変更がアプリケーションの起動に影響を与えたかについての確信度である。他方は、どの構成変更がアプリケーションの起動の問題（失敗）を修復可能であるかについての確信度である。この例におけるほかのコンピュータの数は3であるが、分析の正確度は、ほかのコンピュータの数を増加することによってより高くすることが可能である。

【0038】

## 2. 分析プログラムのユーザ・インターフェース

図4は、原因分析プログラム121のユーザ・インターフェース401の例を示す。ユーザは、このユーザ・インターフェース401を使用することによって分析を開始することができる。原因分析プログラム・ユーザ・インターフェース401は、分析条件を入力する2つのテキスト・ボックスを有する。一方はコンピュータID 411であり、ユーザが分析目標コンピュータの識別子の指定に使用することができる。他方はアプリケーション名412であり、ユーザが、問題を有するアプリケーション名の指定に使用することができる。ユーザは、『分析開始』ボタン413を押すことによって分析を開始できる。

【0039】

図5は、原因分析プログラム121の結果スクリーンの例を示す。可能性のある原因構成変更が、上側のペイン内にランク付けの形式で表示される。列511は構成項目を示す。列512は変更タイプを示す。列513は、構成変更（すなわち、511および512）に対応する日付および時刻を示す。図5は、4つの構成変更レコード521～524を示す。列514は、構成変更に対応する確信度を示す棒グラフを示す。領域525は、すべてのコンピュータにおいて指定されたアプリケーションの起動に影響を与えたレコード521内の構成変更の事例（『PRINTER DRIVER A』 『追加』）の数を示す。領域526は、すべてのコンピュータにおいて指定されたアプリケーションの起動に影響を与えなかった構成変更の事例（『PRINTER DRIVER A』 『追加』）の数を示す。棒グラフの上のパーセンテージは、525と526の比を示す。記号515は、並べ替えキーの指標である。この例においては、数ではなくレート（確信度）の順でこれらの構成変更が示されている。この指標は、各列のリンク（アンダーライン）をクリックすることによってほかの列に移動させることができる。

【0040】

可能性のある修復構成変更が、下側のペイン内にランク付けの形式で表示されている。列531は構成項目を示す。列532は変更タイプを示す。列533は、構成変更に対応する確信度を示す棒グラフを示す。図5は、3つの構成変更レコード541～543を示す。領域544は、すべてのコンピュータにおいて指定されたアプリケーションの起動失敗を修復したレコード541内の構成変更の事例（『PRINTER DRIVER A』 『削除』）の数を示す。領域545は、すべてのコンピュータにおいて指定されたアプリケーションの起動失敗を修復しなかった構成変更の事例（『PRINTER DRIVER A』 『削除』）の数を示す。棒グラフの上のパーセンテージは、544と545の比を示す。記号534は、並べ替えキーの指標である。この例においては、数ではなくレート（確信度）の順でこれらの構成変更が示されている。

【0041】

## 3. データ構造

図6は、分析コンピュータ101内に常駐するイベント・ログ・テーブル141の例を示す。このテーブル内のイベント・ログ・データは、指定されたアプリケーションが問題なく起動できたか否かの決定に使用される。起動結果チェッカ134が、アプリケーションの起動時の直後にイベントの数をチェックする。起動の直後の特定の期間内にいくつかのイベントが見つかる場合に、起動結果チェッカ134は、そのアプリケーションの起動が失敗したと判断する。

【0042】

イベント・ログ・テーブル141は、コンピュータID（601）、日時（602）、およびイベント・タイプ（603）の3つの列を含む。コンピュータID 601、日時

10

20

30

40

50

602、およびイベント・タイプ603は、ログ収集プログラム122によって各目標コンピュータ102のエージェント161から収集されてこのテーブル内にストアされる。各目標コンピュータ102内のイベント・ログ・テーブルのテーブル概要は、この実施態様における分析コンピュータ101のイベント・ログ・テーブル141のそれと同じである。図6は、Comp 001のためのイベント・ログ・レコード415~417、Comp 002のためのレコード421~422、Comp 003のためのレコード431、Comp 006のためのレコード441、およびComp 007のためのレコード451~452等々を示している。各目標コンピュータ102内のイベント・ログ・テーブルは、それ独自のイベント・ログ・データを有する。分析コンピュータ101内のイベント・ログ・テーブル141は、各目標コンピュータ102から収集されるすべてのイベント・ログ・データを有する。

10

#### 【0043】

図7は、分析コンピュータ101内に常駐する構成変更履歴テーブル143の例を示す。このテーブル内の構成変更履歴データが、成功したアプリケーションの起動と失敗した起動の間においてどの種類の構成変更が行われたか、またどの種類の構成変更が失敗したアプリケーションの起動を修復したかの決定に使用される。

#### 【0044】

構成変更履歴テーブル143は、コンピュータID 701、変更日時702、構成項目703、および変更タイプ704の4つの列を含む。これら4つの列のデータは、ログ収集プログラム122によってそれぞれの目標コンピュータ102のエージェント161から収集されてこのテーブル内にストアされる。各目標コンピュータ102内の構成変更履歴テーブルのテーブル概要は、この実施態様における分析コンピュータ101の構成変更履歴テーブル143のそれと同じである。各目標コンピュータ102内の構成変更履歴テーブルは、それ独自の構成変更履歴データを有する。分析コンピュータ101内の構成変更履歴テーブル143は、各目標コンピュータ102から収集されるすべての構成変更履歴データを有する。

20

#### 【0045】

構成項目の例は、ソフトウェア、アプリケーション(追加/削除)、パッチ(追加/削除)、ドライバ(追加/削除)、OS設定、プロセッサ・スケジューリング(プログラム/バックグラウンド・サービス)、メモリ使用(プログラム/システム・キャッシュ)、任意のレジストリ項目、ハードウェア、メモリ容量、ハード・ドライブ容量、BIOS設定、ハイパー・スレッド(オン/オフ)、および仮想化テクノロジー(オン/オフ)を含む。図7は、Comp 001のための構成変更履歴レコード711~716、Comp 002のためのレコード721~726、Comp 003のためのレコード731~734、Comp 004のためのレコード741~743、Comp 005のためのレコード751~752、Comp 006のためのレコード761~762等々を示している。

30

#### 【0046】

図8は、分析コンピュータ101内に常駐するアプリケーション起動履歴テーブル142の例を示している。このテーブル内のアプリケーション起動履歴データが、構成変更前および後においていつアプリケーションが起動されたかの決定に使用される。このテーブルは、コンピュータID 801、起動日時802、およびアプリケーション名803の3つの列を含む。これら3つの列のデータは、ログ収集プログラム122によってそれぞれの目標コンピュータ102のエージェント161から収集されてこのテーブル内にストアされる。各目標コンピュータ102内のアプリケーション起動履歴テーブルのテーブル概要は、この実施態様における分析コンピュータ101のアプリケーション起動履歴テーブル142のそれと同じである。各目標コンピュータ102内のアプリケーション起動履歴テーブルは、それ独自のアプリケーション起動履歴データを有する。分析コンピュータ101内のアプリケーション起動履歴テーブル142は、各目標コンピュータ102から収集されるすべてのアプリケーション起動履歴データを有する。図8は、Comp 00

40

50

1のためのアプリケーション起動履歴レコード811~820、Comp 002のためのレコード821~822、Comp 003のためのレコード831~835、Comp 004のためのレコード841~844、Comp 005のためのレコード851等々を示している。

【0047】

図9は、本発明の第1の実施態様に従って分析コンピュータ101内に常駐する原因構成変更一時テーブル144の例を示している。このテーブルは、原因分析プログラム121が原因構成変更を決定するときの一時テーブルである。このテーブルは、構成変更の前および後におけるアプリケーションの起動の結果を示す。このテーブルは、コンピュータID 901、変更日時902、構成項目903、変更タイプ904、起動前905、および起動後906の6つの列を含む。起動前905は、構成変更前のアプリケーションの起動結果を示す。起動後906は、構成変更後のアプリケーションの起動結果を示す。

10

【0048】

分析目標コンピュータ102ごとに、レコードが、最後に成功したアプリケーションの起動から最初に失敗したアプリケーションの起動までの期間を示す。図9内の分析目標コンピュータ102のコンピュータID 901の値が『Comp 001』であると仮定する。たとえば、レコード(911~914)の起動前905と起動後906のペアのすべての値は、それぞれ成功および失敗である。ほかのコンピュータについては、レコードが、分析目標コンピュータ102と同じ構成変更の前および後のアプリケーションの起動結果を示す。たとえば、構成項目903と変更タイプ904のペアの値は、『PRINTER DRIVER A - 追加』、『PATCH 2322 - 追加』、『VPN CLIENT v2.0 - 追加』、または『VPN CLIENT v1.8 - 削除』のうちの1つでなければならない。図9は、Comp 001のための原因構成変更レコード911~914、Comp 002のためのレコード921~924、Comp 003のためのレコード931~932、Comp 004のためのレコード941~942、Comp 005のためのレコード951~952、Comp 006のためのレコード961、Comp 007のためのレコード971等々を示している。

20

【0049】

図10は、分析コンピュータ101内に常駐する原因構成変更テーブル146の例を示している。このテーブルは、原因分析プログラム121によって作成される結果のテーブルである。このテーブルは、アプリケーションの起動を失敗させた構成変更および対応する確信度を示す。このテーブルは、構成項目1001、変更タイプ1002、変更日時1003、失敗事例の数1004、全事例の数1005の5つの列を含む。全事例の数1005は、構成項目1001と変更タイプ1002のペアについてのすべての事例の数を示す。失敗事例の数1004は、構成項目1001と変更タイプ1002のペアについての失敗した事例の数を示す。図10は、4つの原因構成変更レコード1011~1014を示す。たとえばレコード1011においては、構成変更が『PRINTER DRIVER A - 追加』であり、失敗事例の数1004が12であり、全事例の数1005が15である。

30

40

【0050】

図11は、分析コンピュータ101内に常駐する修復構成変更一時テーブル145の例を示している。このテーブルは、原因分析プログラム121が修復構成変更を決定するときの一時テーブルである。このテーブルは、指定のアプリケーションの起動失敗と起動成功の間に行われた構成変更の前および後のアプリケーションの起動の結果を示す。このテーブルは、コンピュータID 1101、変更日時1102、構成項目1103、変更タイプ1104、起動前1105、および起動後1106の6つの列を含む。それぞれの列の意味は、原因構成変更一時テーブル144と同じである。図11は、Comp 001のための修復構成変更レコード1111~1113、Comp 002のためのレコード1121、Comp 003のためのレコード1131、Comp 004のための

50

レコード 1 1 4 1 等々を示している。

【 0 0 5 1 】

図 1 2 は、分析コンピュータ 1 0 1 内に常駐する修復構成変更テーブル 1 4 7 の例を示している。このテーブルは、原因分析プログラム 1 2 1 によって作成される結果のテーブルである。このテーブルは、失敗したアプリケーションの起動を修復した構成変更および対応する確信度を示す。このテーブルは、構成項目 1 2 0 1、変更タイプ 1 2 0 2、修復事例の数 1 2 0 3、および全事例の数 1 2 0 4 の 4 つの列を含む。全事例の数 1 2 0 4 は、構成項目 1 2 0 1 と変更タイプ 1 2 0 2 のペアについてのすべての事例の数を示す。修復事例の数 1 2 0 3 は、構成項目 1 2 0 1 と変更タイプ 1 2 0 2 のペアについての修復した事例の数を示す。たとえばレコード 1 2 1 1 においては、構成変更が『 P R I N T E R D R I V E R A - 追加』であり、修復事例の数 1 2 0 3 が 2 9 であり、全事例の数 1 2 0 4 が 3 3 である。図 1 2 は、修復構成変更レコード 1 2 1 1 ~ 1 2 1 3 を示している。

10

【 0 0 5 2 】

4 . プロセス・フロー

図 1 3 は、分析コンピュータ 1 0 1 内に常駐するログ収集プログラム 1 2 2 によって実行されるところのログ収集を図解したフローチャートの例である。ログ収集プログラム 1 2 2 は、そのプロセスを特定の間隔で周期的に開始する。図 1 3 に図解されているとおり、ステップ 1 3 0 1 においてログ収集プログラム 1 2 2 は、ログ収集のための目標コンピュータを発見する。ステップ 1 3 0 2 においてはログ収集プログラム ( 1 2 2 ) が、すべての発見済みコンピュータが処理されたか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ 1 3 0 3 に進む。ステップ 1 3 0 3 においてログ収集プログラム 1 2 2 は、目標コンピュータ 1 0 2 内に常駐するエージェント 1 6 1 との通信によってイベント・ログを収集する。またそれは、分析コンピュータ 1 0 1 内のイベント・ログ・テーブル 1 4 1 の更新も行う。ステップ 1 3 0 4 においてはログ収集プログラム 1 2 2 が、目標コンピュータ 1 0 2 内に常駐するエージェント 1 6 1 との通信によってアプリケーション起動履歴を収集する。またそれは、分析コンピュータ 1 0 1 内のアプリケーション起動履歴テーブル 1 4 2 の更新も行う。ステップ 1 3 0 5 においてログ収集プログラム 1 2 2 は、目標コンピュータ 1 0 2 内に常駐するエージェント 1 6 1 との通信によって構成変更履歴を収集する。またそれは、分析コンピュータ 1 0 1 内の構成変更履歴テーブル 1 4 3 の更新も行う。すべての発見済みコンピュータからのログ情報を処理すると ( ステップ 1 3 0 2 においてチェックされる )、ログ収集プログラム 1 2 2 はプロセスを終了する。

20

30

【 0 0 5 3 】

図 1 4 は、本発明の第 1 の実施態様に従って分析コンピュータ 1 0 1 内に常駐する原因分析プログラム 1 2 1 によって実行されるところの原因分析を図解したフローチャートの例である。原因分析プログラム 1 2 1 は、原因分析プログラム・ユーザ・インターフェース 4 0 1 上におけるユーザの操作によってプロセスを開始する。図 1 4 に図解されているとおり、ステップ 1 4 0 1 においては原因分析プログラム 1 2 1 が、原因分析プログラム・ユーザ・インターフェース 4 0 1 からパラメータとしてコンピュータ ID 4 1 1 およびアプリケーション名 4 1 2 を受信する。この説明においてはコンピュータ ID を『 C o m p 0 0 1 』とし、アプリケーション名を『 D O C E D I T O R 』とする。ステップ 1 4 0 2 においては、原因分析プログラム 1 2 1 が一時テーブルおよび結果テーブル ( 1 4 4、1 4 5、1 4 6、および 1 4 7 ) を初期化する。

40

【 0 0 5 4 】

ステップ 1 4 0 3 において原因分析プログラム 1 2 1 は、コンピュータ ID 4 1 1 およびアプリケーション名 4 1 2 の値をパラメータとして目標期間検出 1 3 1 を呼び出す。結果は、原因構成変更一時テーブル 1 4 4 内にストアされることになる。このステップの時点で、構成変更一時テーブル 1 4 4 内にレコード ( 9 1 1 ~ 9 1 4 ) がストアされる。したがって、アプリケーションの起動の失敗を生じさせた構成変更は、それらの構成変更

50

( 9 1 1 ~ 9 1 4 ) のうちの 1 つであると見られる。

【 0 0 5 5 】

ステップ 1 4 0 4 において原因分析プログラム 1 2 1 は、コンピュータ ID 4 1 1 およびアプリケーション名 4 1 2 の値をパラメータとして原因構成変更分析 1 3 2 を呼び出す。結果は、原因構成変更テーブル 1 4 6 内にストアされることになる。このステップの時点で、原因構成変更テーブル 1 4 6 内にレコード ( 1 0 1 1 ~ 1 0 1 4 ) がストアされる。

【 0 0 5 6 】

ステップ 1 4 0 5 において原因分析プログラム 1 2 1 は、アプリケーション名 4 1 2 の値をパラメータとして修復構成変更分析 1 3 3 を呼び出す。結果は、修復構成変更テーブル 1 4 7 内にストアされることになる。このステップの時点で、修復構成変更テーブル 1 4 7 内にレコード ( 1 2 1 1 ~ 1 2 1 3 ) がストアされる。ステップ 1 4 0 6 においては、原因分析プログラム 1 2 1 が結果を表示器 1 1 7 上に表示する。

10

【 0 0 5 7 】

図 1 5 は、分析コンピュータ 1 0 1 内に常駐する目標期間検出 1 3 1 によって実行される目標期間検出プロセスを図解したフローチャートの例である。目標期間検出 1 3 1 は、原因分析プログラム 1 2 1 からの呼び出しによってプロセスを開始する。図 1 5 に図解されているとおり、ステップ 1 5 0 1 においては目標期間検出 1 3 1 が、パラメータとしてコンピュータ ID およびアプリケーション名を受領する。この説明においてはコンピュータ ID を『 C o m p 0 0 1 』とし、アプリケーション名を『 D O C E D I T O R 』とする。

20

【 0 0 5 8 】

ステップ 1 5 0 2 においては目標期間検出 1 3 1 が、ステップ 1 5 0 1 において受領したコンピュータ ID と同じコンピュータ ID のレコードを構成変更履歴テーブル 1 4 3 から抽出する。またそれは、変更日時 7 0 2 の降順によりレコードを並べ替える。このステップの時点で、コンピュータ ID 7 0 1 が『 C o m p 0 0 1 』のレコードが抽出される ( 構成変更履歴テーブル 1 4 3 上の 7 1 1 ~ 7 1 6 ) 。

【 0 0 5 9 】

ステップ 1 5 0 3 においては目標期間検出 1 3 1 が、ステップ 1 5 0 2 においてレコードが抽出されたか否かをチェックする。イエスであれば、プロセスがステップ 1 5 0 4 に進む。ノーであれば、プロセスが終了する。

30

【 0 0 6 0 】

ステップ 1 5 0 4 においては目標期間検出 1 3 1 が、先頭から 1 つのレコードを取り出し、変更日時 7 0 2、構成項目 7 0 3、および変更タイプ 7 0 4 の値を読み取る。このループの最初の実行においては変更日時 7 0 2 の値が『 0 6 / 0 4 / 2 0 0 8 0 8 : 2 0 : 1 1 』であり、構成項目 7 0 3 が『 P R I N T E R D R I V E R A 』、変更タイプ ( 7 0 4 ) が『 追加 』である。

【 0 0 6 1 】

ステップ 1 5 0 5 においては目標期間検出 1 3 1 が、コンピュータ ID ( ステップ 1 5 0 1 )、アプリケーション名ステップ ( 1 5 0 1 )、および変更日時 ( ステップ 1 5 0 4 ) の値をパラメータとして起動結果チェッカ 1 3 4 を呼び出す。このループの最初の実行においては、これらのパラメータが『 C o m p 0 0 1 』、『 D O C E D I T O R 』、および『 0 6 / 0 4 / 2 0 0 8 0 8 : 2 0 : 1 1 』である。

40

【 0 0 6 2 】

ステップ 1 5 0 6 においては目標期間検出 1 3 1 が、ステップ 1 5 0 5 の結果として起動前および起動後変数の値を受領する。結果は、アプリケーションがエラーをまったく伴うことなく構成変更の前および後に起動できたか否かを示す。このループの最初の実行においては、起動前の結果の値が『 成功 』であり、起動後の結果の値が『 失敗 』である。

【 0 0 6 3 】

50



ステップ1507においては目標期間検出131が、構成変更後の起動結果が成功であるか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ1508に進む。

**【0064】**

ステップ1508においては目標期間検出131が、コンピュータID 901、変更日時902、構成項目903、変更タイプ904、起動前905、および起動後906のレコードを作成する。またそれは、このレコードを原因構成変更一時テーブル144内に挿入する。最初のループのこのステップの時点で、原因構成変更一時テーブル144内にレコード911がストアされる。

**【0065】**

ステップ1509においては目標期間検出131が、ステップ1502において抽出されたすべてのレコードが処理されたか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ1504に戻る。この実施態様においては、目標期間検出131の実行後にレコード(911~914)が構成変更一時テーブル144内にストアされる。

**【0066】**

図16は、分析コンピュータ101内に常駐する起動結果チェッカ134によって実行されるところのアプリケーション起動結果をチェックするプロセスを図解したフローチャートの例である。起動結果チェッカ134は、目標期間検出131、原因構成変更分析132、および修復構成変更分析133からの呼び出しによってプロセスを開始する。図16に図解されているとおり、ステップ1601においては起動結果チェッカ134が、パラメータとしてコンピュータID、アプリケーション名、および変更日時を受領する。たとえば、コンピュータIDを『Comp 001』、アプリケーション名を『DOCUMENT EDITOR』、および変更日時を『06/04/2008 08:20:11』とする。

**【0067】**

ステップ1602においては起動結果チェッカ134が、コンピュータID(ステップ1601)およびアプリケーション名(ステップ1601)についてアプリケーション起動履歴テーブル142を参照することによって変更日時(ステップ1601)の直前のアプリケーション起動時刻を獲得する。受領した変更日時が『06/04/2008 08:20:11』であるとき、『06/04/2008 08:20:11』の直前の『DOCUMENT EDITOR』のアプリケーション起動時刻は、アプリケーション起動履歴テーブル142内に『06/02/2008 14:26:03』(818)として見つけることができる。

**【0068】**

ステップ1603においては起動結果チェッカ134が、コンピュータID(ステップ1601)についてイベント・ログ・テーブル141を参照することによってアプリケーション起動時刻(ステップ1602)の直後の特定時間内のイベント数をカウントする。起動時刻が『06/02/2008 14:26:03』であるとき、10秒内のイベントの数は0である。

**【0069】**

ステップ1604においては起動結果チェッカ134が、ステップ1603においてカウントされたイベント数をチェックする。それが0より大きい場合には、プロセスがステップ1606に進む。それ以外であればプロセスがステップ1605に進む。ステップ1605においては起動結果チェッカ134が、起動前変数を成功にセットする。イベントの数が0であることから、起動前変数は成功にセットされる。ステップ1606においては起動結果チェッカ134が、起動前変数を失敗にセットする。

**【0070】**

ステップ1607においては起動結果チェッカ134が、コンピュータID(ステップ1601)およびアプリケーション名(ステップ1601)についてアプリケーション起動履歴テーブル142を参照することによって変更日時(ステップ1601)の直後のア

10

20

30

40

50

アプリケーション起動時刻を獲得する。受領した変更日時が『06/04/2008 08:20:11』であるとき、『06/04/2008 08:20:11』の直後の『DOC EDITOR』のアプリケーション起動時刻は、アプリケーション起動履歴テーブル142内に『06/04/2008 08:29:23』（レコード417）として見つけることができる。

【0071】

ステップ1608においては起動結果チェッカ134が、コンピュータID（ステップ1601）についてイベント・ログ・テーブル141を参照することによってアプリケーション起動時刻（ステップ1607）の直後の特定時間内のイベント数をカウントする。起動時刻が『06/04/2008 08:29:23』であるとき、10秒内のイベントの数は1である。

10

【0072】

ステップ1609においては起動結果チェッカ134が、ステップ1608においてカウントされたイベント数をチェックする。それが0より大きい場合には、プロセスがステップ1611に進む。それ以外であればプロセスがステップ1610に進む。ステップ1610においては起動結果チェッカ134が、起動 後変数を成功にセットする。ステップ1611においては起動結果チェッカ134が、起動 後変数を失敗にセットする。イベントの数が1であることから、起動 後変数は失敗にセットされる。

【0073】

ステップ1612においては起動結果チェッカ134が、起動 前および起動 後変数の値を返す。この説明においては、起動 前の戻り値が成功であり、起動 後の値が失敗である。

20

【0074】

図17は、本発明の第1の実施態様に従って分析コンピュータ101内に常駐する原因構成変更分析132によって実行される場所の原因構成変更分析プロセスを図解したフローチャートの例である。原因構成変更分析132は、原因分析プログラム121からの呼び出しによってプロセスを開始する。図17に図解されているとおり、ステップ1701においては原因構成変更分析132が、パラメータとしてコンピュータIDおよびアプリケーション名を受領する。この説明においてはコンピュータIDを『Comp 001』とし、アプリケーション名を『DOC EDITOR』とする。ステップ1702においては原因構成変更分析132が、コンピュータID（ステップ1701）およびアプリケーション名（ステップ1701）の値とともに図18のサブルーチン呼び出す。このステップの時点で、レコード（911～971）が原因構成変更一時テーブル144内にストアされる。ステップ1703においては原因構成変更分析132が、ステップ1702の戻り値として構成変更リストを受領する。リスト内の項目は、構成項目、変更タイプ、および変更日時を含む。この説明においては、構成変更リストが『PRINTER DRIVER A - 追加 - 06/04/2008 08:20:11』、『PATCH 2322 - 追加 - 06/04/2008 07:43:11』、『VPN CLIENT v2.0 - 追加 - 06/03/2008 14:27:35』、および『VPN CLIENT v1.8 - 削除 - 06/03/2008 13:59:28』を含む。

30

40

【0075】

ステップ1704においては原因構成変更分析132が、ステップ1703において受領されたリスト内のすべての項目が処理されたか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ1705に進む。ステップ1705においては原因構成変更分析132が、リスト（ステップ1703）から1つの項目を取り出し、構成項目、変更タイプ、および変更日時を読み取る。ステップ1706においては原因構成変更分析132が、原因構成変更一時テーブル144を参照することによって、（構成項目（ステップ1705）=テーブル144内の構成項目）、かつ（変更タイプ（ステップ1705）=テーブル144内の変更タイプ）、かつ（テーブル144内

50

の起動 前 = 成功)、かつ(テーブル144内の起動 後 = 失敗)という条件を満たすレコードの数をカウントする。ステップ1707においては原因構成変更分析132が、原因構成変更一時テーブル144を参照することによって、(構成項目(ステップ1705) = テーブル144内の構成項目)、かつ(変更タイプ(ステップ1705) = テーブル144内の変更タイプ)という条件を満たすレコードの数をカウントする。ステップ1708においては原因構成変更分析132が、結果のレコードを原因構成変更テーブル146内に挿入する。このレコードは、構成項目(ステップ1705)、変更タイプ(ステップ1705)、変更日時(ステップ1705)、失敗レコードの数(ステップ1706の結果)、および関係するすべてのレコードの数(ステップ1707の結果)を含む。このステップの時点で、レコード(1011~1014)が原因構成変更テーブル146内に

10

#### 【0076】

図18は、図17のステップ1702における原因構成変更分析プロセスのサブルーチンを図解したフローチャートの例である。このサブルーチンは、原因構成変更分析132からの呼び出しによってプロセスを開始する。図18に図解されているとおり、ステップ1801においてこのサブルーチンは、コンピュータIDおよびアプリケーション名をパラメータとして受領する。この説明においてはコンピュータIDを『Comp 001』とし、アプリケーション名を『DOC EDITOR』とする。ステップ1802においてはこのサブルーチンが、原因構成変更一時テーブル144からすべてのレコードを取り出す。またこれは、構成項目と変更タイプのすべてのペアを読み取り、それらをリストとしてCONFIG LIST変数にセットする。このステップの時点で、CONFIG LIST変数が、『PRINTER DRIVER A - 追加』、『PATCH 2322 - 追加』、『VPN CLIENT v2.0 - 追加』、および『VPN CLIENT v1.8 - 削除』を含む。ステップ1803においてこのサブルーチンは、CONFIG LIST(ステップ1802)内と同じ構成項目903と変更タイプ904のペアを有するレコードを、ステップ1801において受領したコンピュータIDについてのレコードを除き、構成変更履歴テーブル143から抽出する。

20

#### 【0077】

ステップ1804においてはこのサブルーチンが、ステップ1803において抽出したすべてのレコードが処理されたか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ1805に進む。ステップ1805においてはこのサブルーチンが、ステップ1803において抽出したレコードから1つのレコードを取り出し、コンピュータID 901、変更日時902、構成項目903、および変更タイプ904を読み取る。ステップ1806においてはこのサブルーチンが、コンピュータID(ステップ1805)、アプリケーション名(ステップ1801)、および変更日時(ステップ1805)の値とともに起動結果チェッカ134を呼び出す。ステップ1807においてはこのサブルーチンが、ステップ1806の結果として起動 前および起動 後変数の値を受領する。結果は、構成変更の前および後においてアプリケーションがエラーをまったく伴うことなく起動できたか否かを示す。ステップ1808においてはこのサブルーチンが、コンピュータID 901、変更日時902、構成項目903、変更タイプ904、起動 前905、および起動 後906を含むレコードを作成する。またそのレコードを、原因構成変更一時テーブル144内に挿入する。

30

40

#### 【0078】

図19は、分析コンピュータ101内に常駐する修復構成変更分析133によって実行されるところの修復構成変更分析プロセスを図解したフローチャートの例である。修復構成変更分析133は、原因分析プログラム121からの呼び出しによってプロセスを開始する。図19に図解されているとおり、ステップ1901において修復構成変更分析133は、パラメータとしてアプリケーション名を受領する。この説明においてはアプリケーション名を『DOC EDITOR』とする。ステップ1902においては修復構成変更分析133が、アプリケーション名(ステップ1901)の値とともに図20のサブルー

50

チンを呼び出す。ステップ1903においては修復構成変更分析133が、(起動前1105 = 失敗)かつ(起動後1106 = 成功)という条件を満たすレコードを修復構成変更一時テーブル145(ステップ1902)から抽出する。ステップ1904においては修復構成変更分析133が、ステップ1903において抽出したレコードから重複を伴うことなく構成項目1103と変更タイプ1104のペアを抽出する。ステップ1905においては修復構成変更分析133が、ステップ1903において抽出したレコードからステップ1904において抽出したペアの中に構成項目1103および変更タイプ1104の値が含まれていないレコードを削除する。

【0079】

ステップ1906においては修復構成変更分析133が、ステップ1904において抽出したすべてのペアが処理されたか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ1907に進む。ステップ1907においては修復構成変更分析133が、1つのペアを取り出し、構成項目1103および変更タイプ1104の値を読み取る。ステップ1908においては修復構成変更分析133が、修復構成変更一時テーブル145を参照することによって、(構成項目(ステップ1907) = テーブル145内の構成項目)、かつ(変更タイプ(ステップ1907) = テーブル145内の変更タイプ)、かつ(テーブル145内の起動後 = 成功)という条件を満たすレコードの数をカウントする。ステップ1909においては修復構成変更分析133が、修復構成変更一時テーブル145を参照することによって、(構成項目(ステップ1907) = テーブル145内の構成項目)、かつ(変更タイプ(ステップ1907) = テーブル145内の変更タイプ)という条件を満たすレコードの数をカウントする。ステップ1910においては修復構成変更分析133が、結果のレコードを修復構成変更テーブル147内に挿入する。当該レコードは、構成項目(ステップ1907)、変更タイプ(ステップ1907)、成功レコードの数(ステップ1908の結果)、および関係するすべてのレコードの数(ステップ1909の結果)を含む。

【0080】

図20は、図19のステップ1902における修復構成変更分析プロセスのサブルーチンを図解したフローチャートの例である。このサブルーチンは、修復構成変更分析133からの呼び出しによってプロセスを開始する。図20に図解されているとおり、ステップ2001においてこのサブルーチンは、アプリケーション名をパラメータとして受領する。ステップ2002においてはこのサブルーチンが、起動後906の値が失敗であるレコードを原因構成変更一時テーブル144から抽出する。

【0081】

ステップ2003においてはこのサブルーチンが、ステップ2002において抽出したすべてのレコードが処理されたか否かをチェックする。イエスであれば、プロセスがステップ2012に進む。ノーであれば、プロセスがステップ2004に進む。ステップ2004においてはこのサブルーチンが、ステップ2002において抽出したレコードの中から1つのレコードを取り出し、コンピュータID 901および変更日時902を読み取る。ステップ2005においてはこのサブルーチンが、(コンピュータID 701 = コンピュータID(ステップ2004))、かつ(変更日時702 > 変更日時(ステップ2004))という条件を満たすレコードを構成変更履歴テーブル143から抽出する。

【0082】

ステップ2006においてはこのサブルーチンが、ステップ2005において抽出したすべてのレコードが処理されたか否かをチェックする。イエスであれば、プロセスがステップ2003に進む。ノーであれば、プロセスがステップ2007に進む。ステップ2007においてはこのサブルーチンが、ステップ2005において抽出したレコードから1つのレコードを取り出し、コンピュータIDおよび変更日時を読み取る。ステップ2008においてはこのサブルーチンが、コンピュータID(ステップ2007)、アプリケーション名(ステップ2001)、および変更日時(ステップ2007)の値とともに起動結果チェッカ134を呼び出す。ステップ2009においてはこのサブルーチンが、ステ

10

20

30

40

50

ップ2008の結果として起動前および起動後変数の値を受領する。ステップ2010においてはこのサブルーチンが、コンピュータID 1101、変更日時1102、構成項目1103、変更タイプ1104、起動前1105、および起動後1106を含むレコードを作成する。またそのレコードを修復構成変更一時テーブル145内に挿入する。ステップ2011においてはこのサブルーチンが、起動後の値(ステップ2009)が成功か否かをチェックする。イエスであれば、プロセスがステップ2003に戻る。ノーであれば、プロセスがステップ2006に戻る。ステップ2012においてはこのサブルーチンが、修復構成変更一時テーブル145上のレコードの重複を除去する。

【0083】

#### B. 第2の実施態様

図21は、本発明の第2の実施態様に従って原因構成変更テーブル146 21の例を示す。第2の実施態様においては、原因分析プログラム121が、過去に同じ分析が行われていた場合に、過去に分析およびストアが行われた結果を再使用して表示する。これを行うためには、図10の原因構成変更テーブル146を拡張する必要がある。図21における列1001~1005は、図10におけるものと同じである。それに加えてアプリケーション名2101および分析日時2102のための新しい列が導入されている。たとえば、レコード2111は、アプリケーション名『DOC EDITOR』についての分析が分析日時『06/05/2008 14:20:12』に行われたことを示す。アプリケーション名が『DOC EDITOR』であり、目標期間検出131の結果が同じとなるように分析条件が指定された場合に、原因分析プログラム121は、原因構成変更テーブル146 21に基づいて分析結果を表示することが可能である。同一の概念を、図12の修復構成変更テーブル147に対しても適用できる。

【0084】

図22は、本発明の第2の実施態様に従って原因分析プログラム121によって実行される原因分析を図解したフローチャートの例である。ステップ1401、1403、1404、1405、および1406は、図14におけるものと同じである。図14と図22の間の相違は以下のとおりである。ステップ1402 22(ステップ1402の代わり)においては原因分析プログラム(121)が、結果テーブル(原因構成変更テーブル146および修復構成変更テーブル147)を初期化しない。ステップ2201においては原因分析プログラム121が、アプリケーション名412およびステップ1403の構成変更とそれぞれ同じアプリケーション名および構成変更を含むレコードを、原因構成変更テーブル146 21(図21に示す)から検索する。ステップ2202においては原因分析プログラム121が、過去の結果のレコードが見つかったか否かをチェックする。イエスであれば、プロセスがステップ1406に進む。ノーであれば、プロセスがステップ1404に進む。ステップ1404においてストアされる結果は、原因構成変更テーブル146 21(図21)の概要に基づくものとする。

【0085】

#### C. 第3の実施態様

図23は、本発明の第3の実施態様に従って原因構成変更テーブル146 23の例を示している。第3の実施態様においては、原因分析プログラム121が構成変更の組み合わせに基づいて分析を行う。これを行うためには、図10の原因構成変更テーブル146を拡張する必要がある。図23に示されるとおり、列1001~1005は、図10におけるものと同じである。それに加えて組み合わせID 2301のための新しい列が導入される。ここにはレコード2311~2317が存在する。たとえばレコード2311は、各構成変更のすべての組み合わせを使用することによって分析が行われたことを示す。同一の概念を、図12の修復構成変更テーブル147に対して適用して拡張することができる。

【0086】

図24は、本発明の第3の実施態様に従って原因構成変更分析132によって実行される原因構成変更分析プロセスを図解したフローチャートの例である。ステップ1

10

20

30

40

50

701、1702、および1703は、図17おけるもの同じである。図17と図24の間における相違は、以下のとおりとなる。ステップ2401においては原因構成変更分析132が、構成変更1703のすべての組み合わせのリストを作成する。構成変更が『PRINTER DRIVER A - 追加』、『VPN CLIENT v2.0 - 追加』、および『VPN CLIENT v1.8 - 削除』である場合には、それらの組み合わせが次のとおりとなる。

- ・『PRINTER DRIVER A - 追加』、『VPN CLIENT v2.0 - 追加』、『VPN CLIENT v1.8 - 削除』
- ・『PRINTER DRIVER A - 追加』、『VPN CLIENT v2.0 - 追加』
- ・『PRINTER DRIVER A - 追加』、『VPN CLIENT v1.8 - 削除』
- ・『VPN CLIENT v2.0 - 追加』、『VPN CLIENT v1.8 - 削除』
- ・『PRINTER DRIVER A - 追加』
- ・『VPN CLIENT v2.0 - 追加』
- ・『VPN CLIENT v1.8 - 削除』

ステップ2402においては原因構成変更分析132が、ステップ2401において作成したリスト内のすべての項目が処理されたか否かをチェックする。イエスであれば、プロセスが終了する。ノーであれば、プロセスがステップ2403に進む。ステップ2403においては原因構成変更分析132が、リスト(ステップ2401)から1つの項目を取り出し、構成項目、変更タイプ、および変更日時を読み取る。ステップ2404においては原因構成変更分析132が、原因構成変更一時テーブル144を参照することによって、(構成項目と変更タイプのペア(ステップ2403))=テーブル144内のコンピュータにおける構成項目と変更タイプのペア)、かつ(テーブル144内のそのコンピュータにおけるもっとも最近の起動後の値=失敗)という条件を満たすコンピュータの数をカウントする。ステップ2405においては原因構成変更分析132が、原因構成変更一時テーブル144を参照することによって、(構成項目と変更タイプのペア(ステップ2403))=テーブル144内のコンピュータにおける構成項目と変更タイプのペア)という条件を満たすコンピュータの数をカウントする。ステップ2406においては原因構成変更分析132が、結果のレコードを原因構成変更テーブル146 23(図23)内に挿入する。

#### 【0087】

##### D. 第4の実施態様

図25は、本発明の第4の実施態様に従って全体のシステムのハードウェア・アーキテクチャの構成、ソフトウェア・モジュール、およびテーブルの例を図解している。ピアツーピア・アーキテクチャのための第4の実施態様においては、状況に応じてすべてのコンピュータがサーバおよびクライアントとなることが可能である。集中サーバは必要ない。各コンピュータ2501は、原因分析プログラム(121)、エージェント(161)、ログ情報(171)、およびログ収集プログラム(122)を有する。各コンピュータ内のログ情報(171)は、そのコンピュータのログ情報だけでなく、ほかのコンピュータのログ情報も含む。

#### 【0088】

当然のことながら、図1および2に図解されているシステム構成は、本発明の実装が考えられる情報システムの純粋な例示であり、本発明は特定のハードウェア構成に限定されない。本発明を実装するコンピュータおよびストレージ・システムは、上で述べた本発明の実装に使用されるモジュール、プログラム、およびデータ構造をストアし、読み出すことが可能な周知のI/Oデバイス(たとえば、CDおよびDVDドライブ、フレキシブルディスク・ドライブ、ハード・ドライブ等)を有することも可能である。これらのモジュール、プログラム、およびデータ構造は、その種のコンピュータ可読媒体上においてエン

コードすることが可能である。たとえば、本発明のデータ構造を、本発明で使用されるプログラムが常駐する1つまたは複数のコンピュータ可読媒体とは独立してコンピュータ可読媒体上にストアすることができる。システムの構成要素は、たとえば通信ネットワークといった、デジタル・データ通信の任意の形式または媒体によって相互接続が可能である。通信ネットワークの例は、ローカル・エリア・ネットワーク、ワイド・エリア・ネットワーク、たとえばインターネット、無線ネットワーク、ストレージ・エリア・ネットワーク、およびこれらの類を含む。

【0089】

説明においては、本発明の完全な理解を提供するために説明を目的として多くの詳細が示されている。しかしながら当業者には明らかとなるが、これらの特定の詳細のすべてが、本発明を実施するために必要とされるわけではない。これもまた注意されたいが、本発明は、一般にフローチャート、フロー図、構造図、またはブロック図として図示されるプロセスとして記述されることがある。フローチャートは、逐次的なプロセスとして動作を記述することができるが、動作の多くは並列の、または同時的な実行が可能である。それに加えて動作の順序は再編成することができる。

【0090】

この分野で周知のとおり上で述べた動作は、ハードウェア、ソフトウェア、またはソフトウェアとハードウェアの何らかの組み合わせによって実行することが可能である。本発明の実施態様の多様な側面は、回路および論理デバイス（ハードウェア）を使用して実装できるが、ほかの側面は機械可読媒体上にストアされて、プロセッサによって実行された場合に当該プロセッサに本発明の実施態様を遂行する方法を実行させることになるインストラクション（ソフトウェア）を使用して実装できる。さらに、本発明のいくつかの実施態様はハードウェアだけで実行されることがあり、ほかの実施態様はソフトウェアだけで実行されることがある。さらにまた、記述した多様な機能は、単一ユニット内において実行されることが可能であるか、または多数の構成要素にわたり、多くの方法で分散されることが可能である。ソフトウェアによって実行される場合には、方法を、コンピュータ可読媒体上にストアされるインストラクションに基づいて汎用コンピュータ等のプロセッサによって実行することができる。望ましい場合には、インストラクションを圧縮および/または暗号化されたフォーマットで媒体上にストアすることが可能である。

【0091】

以上から、本発明が、知識データベースの使用を伴うことなく構成変更を分析することによってアプリケーションの失敗に対する解決策を見つけ出すための方法、装置、およびコンピュータ可読媒体上にストアされるプログラムを提供することが明らかであろう。それに加えて特定の実施態様が図解され、この明細書の中で述べられてきたが、当業者は、同じ目的を達成すると算定される任意の取り合わせが開示された特定の実施態様の代用となり得ることを認識されよう。この開示は、本発明の任意の、およびすべての適応、または変形を保護することが意図されており、また以下の請求項の中に使用されている用語が、明細書の中に開示されている特定の実施態様に本発明を限定すると解釈されるべきでないことが理解されるものとする。むしろ本発明の範囲は、確立された請求項の解釈論に従って、請求項が権利を有する均等の全範囲とともに解釈されるべき以下の請求項によって完全に決定されるものとする。

【0092】

E. 第5の実施態様

第5の実施態様においては、目標コンピュータ内のエージェントが、アプリケーションのインスールを監視し、インストールを検知したタイミングでインストール開始イベントを分析コンピュータに通知する。本実施態様における分析コンピュータには、集計プログラムを追加する。集計プログラムでは通知されたアプリケーションについて原因構成変更テーブルに一致するアプリケーション名があれば、該当の分析結果を目標コンピュータに送信する。

【0093】

図26は第5の実施態様における目標コンピュータ内に配置されるエージェント161の機能ブロック図と分析コンピュータ101における機能ブロックのうち、本実施態様に関連する部分を示す図である。分析コンピュータには図2に示す機能構成に、集計プログラム2601を追加する。集計プログラムは原因分析プログラムが作成した原因構成変更テーブル146と修復構成変更テーブル147を読み出して利用する。なお、集計プログラム2601は分析コンピュータ101のメモリ112に記録され、CPU111によって実行されることは言うまでもない。

【0094】

エージェント161には、アプリケーションインストールの監視と通知を行うアプリケーション監視手段2602、分析結果の受信処理と保存、ユーザインターフェース2606を介しての出力を行う分析情報管理手段2603を持つ。受信した情報は問題構成変更テーブル2604、対策構成変更テーブル2605として保存される。

10

【0095】

エージェントプログラムの処理フローについて説明する(図示せず)。インストーラプログラムの起動などでアプリケーションインストールの開始を検知すると、該当のアプリケーション名、変更タイプいとして追加という情報を含む構成変更イベントを分析コンピュータに送信する。

【0096】

エージェント161は、分析コンピュータからの分析結果情報を受信すると、分析結果情報に基づいて問題構成変更テーブル2604、又は/及び対策構成変更テーブル2605を作成及び更新する。そして、エージェント161は、結果に基づく画面を作成し、出力する。

20

【0097】

ここで、エージェントはアプリケーションのインストール処理を中断させてもよい。その場合、ユーザに中断を知らせるユーザインタフェースを提供する。インストールを中止させている場合は、分析結果の受信待ちを行い、一定時間受信がなければ、インストール処理を再開する。受信した分析結果の画面への出力処理時に、「インストールを続行しますか?」メッセージとともに、ユーザがインストールを続行または中止を選択可能なユーザインタフェースを提供する。

【0098】

次に問題構成変更テーブル2604について説明する。問題構成変更テーブルは、分析コンピュータ101が分析した問題となる可能性のある構成変更を示したテーブルである。問題構成変更テーブル2604は、インストール対象(これからインストールしようとしているものを対象に含めてもよい)のアプリケーション毎に分析コンピュータが分析処理を行った時間、一つ以上のレコードを持つ。なお、当該レコードは以下の属性値を持つ。

30

\* 原因の可能性のある構成変更を示す値

\* 失敗率。

【0099】

次に対策構成変更テーブル2605について説明する。対策構成変更テーブルは、分析コンピュータ101が分析した、問題を解決する可能性のある構成変更を示したテーブルである。対策構成変更テーブルは、インストール対象(これからインストールしようとしているものを対象に含めてもよい)のアプリケーション毎に分析コンピュータが分析処理を行った時間、一つ以上のレコードを持つ。なお、当該レコードは以下の属性値を持つ。

40

\* アプリケーションの問題を解決する可能性のある構成変更を示す値

\* 成功率。

【0100】

図27には分析コンピュータにおける集計プログラムのフローチャートを示す。

【0101】

以下は分析結果情報に問題となる構成変更を含め、送信する場合の説明である。この場

50



合、図27のステップ2706乃至ステップ2709の実行は省略する。

【0102】

(ステップ2701)集計プログラムは、構成変更イベントを受信する。

【0103】

(ステップ2702)集計プログラムは、受信したアプリケーション名について、原因構成変更テーブルを検索する。

【0104】

(ステップ2703)集計プログラムは、ステップ2702の検索にて、受信したアプリケーション名に一致するアプリケーション名のレコードがあるか否かを判断する。

【0105】

(ステップ2704)集計プログラムは、各構成項目レコードについて、全事例数に対する失敗事例数の割合を確信度として算出する。

【0106】

(ステップ2705)集計プログラムは、次に確信度に基づいて送信するレコードを抽出する。

【0107】

そして抽出したレコードの情報を、分析結果情報に含め、構成変更イベント通知の送信元である目標コンピュータに送信して(ステップ2710)、処理を終了する。

【0108】

なお、ステップ2705の抽出方法としては、閾値を用いる方法と、特定数に限定する方法がある。

【0109】

(方法1)閾値を用いる方法の場合、集計プログラムは管理者または目標コンピュータの利用者から確信度の閾値を受信し、当該閾値をメモリ112に記録することで閾値の管理を行う。確信度について閾値と比較し、閾値以上の構成項目のレコードのみを抽出する。

【0110】

(方法2)特定数に限定する場合、予め特定数(例えば3)を定義しておく。算出した確信度に基づき上位3つまでレコードを抽出する。閾値や特定数の定義がNULLの場合は全レコードとする。

【0111】

図28はエージェント161が問題構成変更テーブル2604を参照し、ビデオI/F154を介して表示器157に表示する出力例を示す。上段には、インストール対象の該アプリケーションに対して、起動失敗の原因となる可能性のある構成項目についての情報を表示する。構成項目とその変更タイプ毎(2801)に、確信度として失敗率を表示し(2802)、さらに確信度によって、エンドユーザに対するメッセージ(2803)を出力する。メッセージは確信度の値により、例えば75%以上で「<警告>この構成変更が行われている場合、アプリケーションの起動が失敗する可能性が高いため、対策を行ってください」、50%~74%で「<注意>この構成変更が行われている場合、アプリケーションの起動が失敗する可能性があります」、50%未満で「<情報>この構成変更はアプリケーションに影響ありません」とする。

【0112】

また原因構成変更テーブルの情報だけでなく、修復構成変更テーブルについても第2の実施態様により、図21と同様にアプリケーション名毎にその構成項目と修復事例数、全事例数がテーブルに保存される。図27のステップ2706乃至ステップ2709はこのような処理を実現するためのステップである。

【0113】

(ステップ2706)集計プログラムは、修復構成変更テーブルを検索して、該当するアプリケーションの情報を抽出する。

【0114】

10

20

30

40

50

(ステップ2707)集計プログラムは、ステップ2706の検索にて、受信したアプリケーション名に一致するアプリケーション名のレコードがあるか否かを判断する。

【0115】

(ステップ2708)集計プログラムは、各構成項目レコードについて、全事例数に対する失敗事例数の割合を確信度として算出する。

【0116】

(ステップ2709)集計プログラムは、次に確信度に基づいて送信するレコードを抽出し、分析結果情報に含める。

【0117】

エージェント161では、分析結果情報を受信し、分析結果情報に含まれるレコードを用いて対策構成変更テーブルを作成または更新する。本実施例においては、目標コンピュータにおける該アプリケーションにおける問題は未発生であることから、対策については、トラブル情報と同時に出力してもよいし、別にユーザインタフェースによる要求があった場合に出力してもよい。

10

【0118】

図29はエージェント161が対策構成変更テーブル2705を参照し、ビデオI/F154を介して表示器157を該当アプリケーションの起動失敗に対する対策の有効性として表示する出力例を示す。構成項目とその変更タイプ毎(2901)に、確信度として対策の成功率を表示し(2902)、さらに確信度によって、エンドユーザに対するメッセージ(2903)を出力する。メッセージは、例えば「この構成変更により、アプリケーションの起動失敗の問題を修復できる可能性があります」とする。

20

【0119】

第5の実施態様の変形例として、送信するレコードの抽出方法について、確信度に基づく方法ではなく、構成変更イベント送信元のコンピュータの構成に基づく方法について説明する。ステップ2703までは同様であり、原因構成変更テーブルのアプリケーション名が一致するレコードを抽出する。次に集計プログラムは図7に示す構成変更履歴テーブルを参照し、コンピュータIDが送信元コンピュータに一致するレコードについて、前記のステップ2703で抽出されたレコードにおける各構成項目と変更タイプについても一致するレコードがあるかを検索する。構成項目と変更タイプが一致するレコードのみ、送信するレコードとする。

30

【0120】

これら送信するレコードの抽出は必要最低限の情報を提供するために行う。確信度が低いものや、該当コンピュータの構成に関係のない構成項目に関する情報を削除することで、目標コンピュータへの転送量を減らすことが可能である。

【0121】

さらに、第5の実施態様の別の例として、アプリケーションのインストール時のみでなく、パッチの適用やソフトウェアの削除など構成項目の変更時に、情報を提供する方法を説明する。エージェントではアプリケーションインストールだけでなく、パッチ適用やドライバ更新、削除といった構成項目の変更を監視し、検知時には分析コンピュータに構成変更イベントとして通知する。ここでは、アプリケーション名として構成項目を、また追加、削除といった変更タイプを構成変更イベントに含める。

40

【0122】

第1の実施態様においてパッチ追加やソフトウェアの削除は、他のアプリケーションの動失敗の原因候補となる構成項目として原因構成変更テーブルに記録されている。よって、集計プログラムでは、構成変更イベントを受信し、受領したアプリケーション名について、図27に示したフローにおいて、原因構成変更テーブルを検索した結果、アプリケーション名としての記録がない場合、構成項目を検索する。さらに、変更タイプに一致するものがあるかを検索して、抽出し、エージェントに送信する。

【0123】

この場合の目標コンピュータでの画面出力情報を図30に示す。構成変更の対象として

50

いるソフトウェア名に対して、影響する構成項目とその変更タイプ毎(3001)に、確信度として対策の成功率を表示し(3002)、さらに確信度によって、エンドユーザに対するメッセージ(3003)を出力する。

【0124】

同一の概念を修復構成変更テーブルに対しても適用できる。

もう一つの変形例として、構成項目が組み合わされた原因構成変更テーブルの情報を利用する方法を説明する。第3の実施態様に示されている構成変更の組み合わせに基づいて行われた分析結果は、図23に示す原因構成変更テーブルに保存されている。ここで、図23のテーブルを拡張し、アプリケーション名および分析日時の列を導入する(特に図示せず)。図24における原因構成変更分析の結果を保存する際に、対象としたアプリケーション名と分析した日時を保存する。

10

【0125】

本実施例における集計プログラムにおける処理で、図27に示したフローと異なる点は、ステップ2702において図21の原因構成変更テーブルを検索する代わりに、図23を拡張した原因構成変更テーブルを検索する点である。拡張した原因構成変更テーブルからアプリケーション名が一致するレコードを抽出し、その組み合わせID毎に、確信度として全事例数に対する失敗事例数を算出する。確信度に基づき、送信する組み合わせIDのレコードを決定する。

【0126】

F. 第6の実施態様

20

第5の実施態様では、以前に該当アプリケーションに関する原因分析要求があり、分析が行われた場合にその結果を目標コンピュータのエンドユーザに提供するものである。ここで、分析が行われていない場合や、最新の分析結果を提供するために、本実施態様では構成変更イベントを受信した時点での分析を実行し、結果を提供する方法を示す。

【0127】

分析目標コンピュータにおいてアプリケーションの起動失敗があつてからの分析ではなく、アプリケーションをインストールする場合に、該当アプリケーションについて他のコンピュータで起動に問題のある事例があるかを分析する。

【0128】

図31は本発明の第6の実施態様における分析コンピュータにおける集計プログラムによって実行される処理のフローチャートの例である。集計プログラムは、図27と同様にエージェントからのインストール開始イベント通知を受信し(ステップ2701)、送信元コンピュータIDと通知に含まれるアプリケーション名を取り出す(ステップ2702)。次に集計プログラムは、コンピュータIDとアプリケーション名をパラメータとして原因分析プログラム121(a)を呼び出す(ステップ3103)。ここで、原因分析プログラムの処理は第1の実施態様における図14の処理とは一部で異なるため、図32で説明する。

30

【0129】

集計プログラムでは、原因分析プログラム121(a)の終了後、第5の実施態様と同様に、結果として得られた原因構成変更テーブルについて、構成項目毎に確信度として全事例数に対する失敗事例数を算出(ステップ2704)し、確信度に基づいて送信するレコードを抽出して(ステップ2705)、エージェントに送信する(ステップ2710)。

40

【0130】

また、原因分析プログラム(a)では第1の実施態様と同様に、修復構成変更テーブルを作成してもよい。その場合、結果として得られた修正構成変更テーブルについても同様に確信度を算出して、確信度に基づき送信するレコードを決定し、エージェントに送信する。

【0131】

図32に本実施態様における原因分析プログラム121(a)のフローチャートを示す。

(ステップ3201)原因分析プログラムは、集計プログラムからコンピュータIDおよ

50

びアプリケーションIDを受領する。

【0132】

(ステップ3202)原因分析プログラムは、一時テーブルおよび結果テーブルを初期化しておく。

【0133】

(ステップ3203)原因分析プログラムは、次に構成変更履歴テーブルを読み出し、受領したコンピュータIDと同じコンピュータIDのレコードを抽出する。ここで、構成変更履歴テーブルのデータが長期間保存されている場合、抽出対象を一定期間前までのレコードに限定してもよい。

【0134】

(ステップ3204)原因分析プログラムは、抽出した結果を原因構成変更一時テーブルに保存する。このコンピュータIDに対しては起動チェックを行わないため、保存したレコードの起動 - 前、起動 - 後カラムはNULLのままにしておく。

【0135】

(ステップ3205)原因分析プログラムは、次に他の目標コンピュータの事例を分析する。分析処理は第1の実施態様と同様であるため、ここで図18のサブルーチンを呼び出す。

【0136】

(ステップ3206)原因分析プログラムは、サブルーチンの戻り値として構成変更リストを受領する。リスト内の項目は、構成項目、変更タイプ、および変更日時を含む。

【0137】

(ステップ3207)原因分析プログラムは、リスト内の項目について、全て処理されたかを確認する。処理されていない場合は、ステップ3208に進む。

【0138】

(ステップ3208)原因分析プログラムは、サブルーチン(3205)の結果、作成された原因構成変更一時テーブルを参照し、次の条件を満たすレコード数をカウントする。構成変更リストの構成項目、変更タイプとテーブルの構成項目、変更タイプが同じで、さらに「起動 - 前 = 成功」かつ「起動 - 後 = 失敗」という条件とする。

【0139】

(ステップ3209)原因分析プログラムは、原因構成変更一時テーブルを参照し、ステップ3201で受領したコンピュータIDについてのレコードを除いて、構成変更リストの構成項目、変更タイプとテーブルの構成項目、変更タイプが同じレコードの数をカウントする。

【0140】

(ステップ3210)原因分析プログラムは、結果のレコードを原因構成変更テーブルに登録する。原因構成変更テーブルは、図21で示したテーブル構成とし、ステップ3201で受領したアプリケーション名、分析日時、構成変更リスト(ステップ3206)に含まれる構成項目、変更タイプ、変更日時と、ステップ3208のカウント値を失敗事例の数、ステップ3209のカウント値を全事例の数として登録する。

【0141】

第6の実施態様の変形例として、該当アプリケーションによる原因構成変更テーブルが既にある場合、分析日時に基づき、一定期間以上前のものでなければ、新たに分析は行わず、該当テーブルの情報を使用する。この場合、図31に示す集計プログラムの処理において、ステップ2702の後に、原因構成変更テーブルを検索し、該当アプリケーションのレコードが見つかった場合に、分析日時を確認する。一定期間以内であれば、原因分析プログラムの呼び出しは行わずに、該当するレコードを使用して、構成項目毎の確信度算出とそれに続く処理(ステップ2704~2706)を行う。分析日時が一定期間以内でない場合はステップ3103に進む。

【0142】

第5または第6の実施態様の変形例として、エージェントが保持するデータを出力する

10

20

30

40

50

方法について説明する。分析コンピュータより受信し、保持する問題構成変更テーブルおよび対策構成変更テーブルには、アプリケーション毎に、原因構成項目または修復構成項目とその確信度を含む。エージェントはアプリケーション名入力を受け付けるユーザインタフェースを提供し、ユーザがアプリケーション名を入力すると、該アプリケーションに関する情報がテーブルにあるか検索し、情報があれば、図28や図29に示した例のように出力する。

#### 【0143】

さらに第6の別の実施態様として、分析コンピュータにおいて定期的に集計プログラムを起動し、情報を更新する方法がある。集計プログラムでは、図31の処理と異なり、起動時にコンピュータIDやアプリケーション名を受領しない。そのため、保存する原因構成変更テーブルにあるアプリケーション名を順番に取り出し、全アプリケーションについて、コンピュータ毎に、前回分析日時以降の構成変更項目を抽出し、原因分析を実施する。定期的に集計した情報は原因構成変更テーブルを更新して保存し、定期的または構成変更イベント受信時やエージェントからの要求時に、目標コンピュータに送信する。

10

#### 【0144】

またアプリケーションのインストールや、パッチの適用といった構成変更の場合以外にも、ユーザ操作が影響するアプリケーションの起動失敗事例の分析を行う方法がある。この場合、分析コンピュータのログ収集プログラムは、ユーザの操作ログを収集し、操作履歴テーブルで管理する。エージェントでは、構成変更監視手段により、OSの設定変更など、ユーザの操作を監視し、特定の操作が行われた場合に、分析コンピュータに通知する。分析コンピュータにおいて集計プログラムと原因分析プログラムにより、操作履歴テーブルを検索し、同様の操作を行った後にアプリケーションの起動失敗が発生している事例を分析し、確信度を集計する。結果をエージェントに送信し、エージェントでは画面に出力する。

20

#### 【0145】

以上により、コンピュータにおける構成変更に起因するアプリケーションの問題について、知識データベースを使用しない分析コンピュータによる分析結果を、エンドユーザが構成変更を行うタイミングで提供し、エンドユーザに対して問題への対応を促す支援を実現できる。

#### 【0146】

以上の第5及び第6の実施態様において、複数の目標コンピュータと分析コンピュータとを有する計算機システムにおけるアプリケーション導入方法であって、前記複数の目標コンピュータに含まれ、所定のアプリケーションをインストール済みかつ起動済みである、一つ以上の第1目標コンピュータは、前記所定のアプリケーション起動以前に行った複数の構成変更の情報を含むログを、前記分析コンピュータに送信し、前記分析コンピュータは、前記ログを受信し、前記ログに基づいて構成変更の種別毎に、当該構成変更後に所定のアプリケーション起動が失敗した割合である起動失敗率を計算することを説明した。

30

#### 【0147】

また、前記複数の目標コンピュータに含まれ、前記一つ以上の第1目標コンピュータ以外である、第2目標コンピュータは：(1)前記分析コンピュータから前記所定のアプリケーションに関する構成変更の種別毎の起動失敗率を含む第1の情報を受信し、(2)前記起動失敗率に基づいて、前記所定のアプリケーションの起動が失敗する原因となる構成変更の種別を表示する、ことを説明した。

40

#### 【0148】

なお、前記第1の情報に含まれる起動失敗率は、前記分析コンピュータが前記ログに基づいて検知した全ての構成変更の種別の一部の種別の各々についての起動失敗率であって、前記一部の種別は、前記起動失敗率に基づいて前記分析コンピュータが選択したものでよいことを説明した。

#### 【0149】

また、前記分析コンピュータは、前記ログに基づいて前記構成変更の種別毎に、当該構

50

成変更後に前記所定のアプリケーションの起動が成功した割合である起動成功率を計算し、前記第2目標コンピュータは、(3)前記分析コンピュータから前記所定のアプリケーションに関する構成変更の種別毎の起動成功率を含む第2の情報を受信し、(4)前記起動成功率に基づいて、前記所定のアプリケーションの起動が成功する原因となる構成変更の種別を表示してもよいことを説明した。

【0150】

また、前記第2目標コンピュータは、前記所定のアプリケーション以外の複数のアプリケーションをインストール済みであり、前記第2目標コンピュータは、前記全ての構成変更の種別に含まれる所定の種別について、起動が失敗する可能性のあるアプリケーションを前記所定のアプリケーション及び前記複数のアプリケーションから選択し、選択したアプリケーションの識別子を表示してもよいことを説明した。

10

【0151】

また、前記種別とは、構成項目又は/及び変更タイプが一例であることを説明したが、他の例としては構成変更操作の種類であってもよい。

【0152】

なお、以上の説明では「aaaテーブル」、「aaaリスト」、「aaaDB」、「aaaキュー」等の表現にて本発明の情報を説明したが、これら情報は必ずしもテーブル、リスト、DB、キュー、等のデータ構造以外で表現されていてもよい。そのため、データ構造に依存しないことを示すために「aaaテーブル」、「aaaリスト」、「aaaDB」、「aaaキュー」等について「aaa情報」と呼んでもよい。さらに、各情報の内容を説明する際に、「識別情報」、「識別子」、「名」、「名前」、「ID」という表現を用いたが、これらについてはお互いに置換が可能である。

20

【0153】

なお、分析コンピュータは複数のコンピュータであってもよい。なお、目標コンピュータ102のメモリ152とディスク153は区別せずにひとまとめに記憶資源(すなわち記憶するデバイス)として扱っても良い。同様に、分析コンピュータ101のメモリ112とディスク113は区別せずにひとまとめに記憶資源として扱っても良い。

【0154】

なお、以上の説明では「プログラム」を主語として説明を行う場合があった、プログラムはプロセッサによって実行されることで定められた処理をメモリ及び通信ポート(通信制御デバイス)を用いながら行うため、プロセッサを主語とした説明としてもよい。また、プログラムを主語として開示された処理は管理サーバ等の計算機、情報処理装置が行う処理としてもよい。また、プログラムの一部または全ては専用ハードウェアによって実現されてもよい。

30

【0155】

また、各種プログラムはプログラム配布サーバや、計算機が読み取り可能な記憶メディアによって各計算機にインストールされてもよい。

【符号の説明】

【0156】

- 101 分析コンピュータ
- 102 分析目標コンピュータ
- 103 LAN
- 111 CPU
- 112 メモリ
- 113 ディスク
- 114 ビデオ・インターフェース
- 115 ネットワーク・インターフェース
- 116 システム・バス
- 117 表示器
- 121 原因分析プログラム

40

50

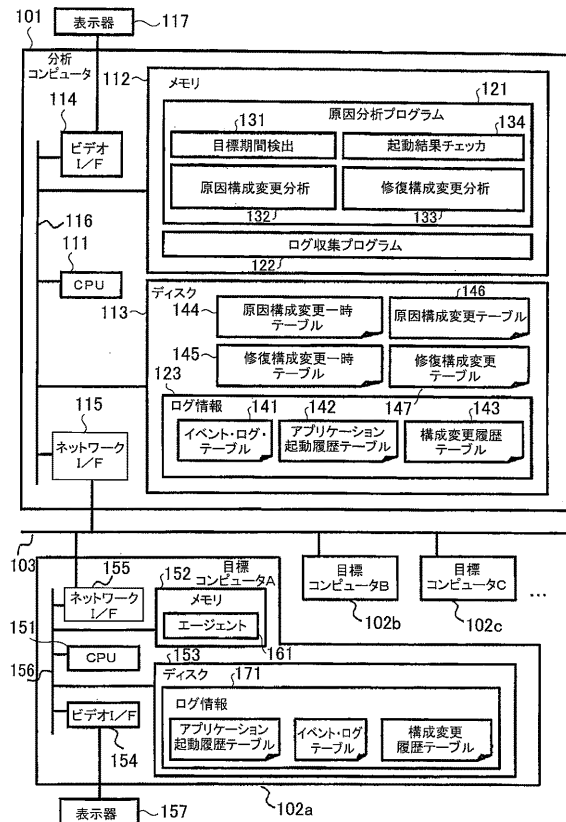
- 1 2 2 ログ収集プログラム
- 1 2 3 ログ情報
- 1 3 1 目標期間検出
- 1 3 2 原因構成変更分析
- 1 3 3 修復構成変更分析
- 1 3 4 起動結果チェッカ
- 1 4 1 イベント・ログ・テーブル
- 1 4 2 アプリケーション起動履歴テーブル
- 1 4 3 構成変更履歴テーブル
- 1 4 4 原因構成変更一時テーブル
- 1 4 5 修復構成変更一時テーブル
- 1 4 6 原因構成変更テーブル
- 1 4 6 2 1 原因構成変更テーブル
- 1 4 6 2 3 原因構成変更テーブル
- 1 4 7 修復構成変更テーブル
- 1 5 1 CPU
- 1 5 2 メモリ
- 1 5 3 ディスク
- 1 5 4 ビデオ・インターフェース
- 1 5 5 ネットワーク・インターフェース
- 1 5 6 システム・バス
- 1 5 7 表示器
- 1 6 1 エージェント
- 1 7 1 ログ情報

10

20

【図 1】

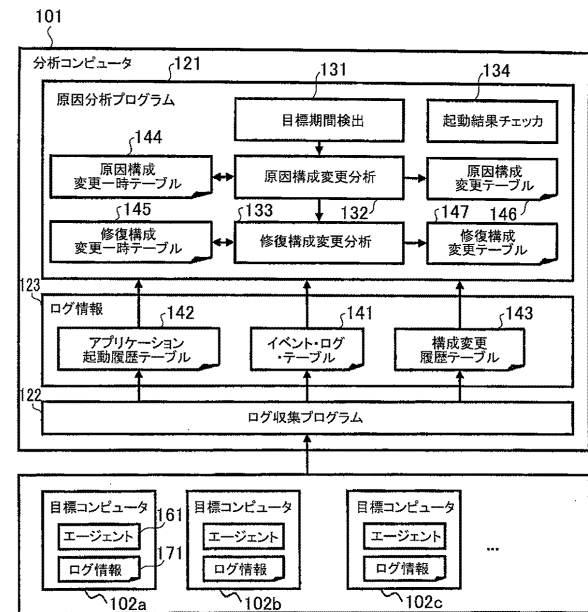
【図 1】



クライアント・サーバ・アーキテクチャのための例示的なハードウェアおよび論理構成

【図 2】

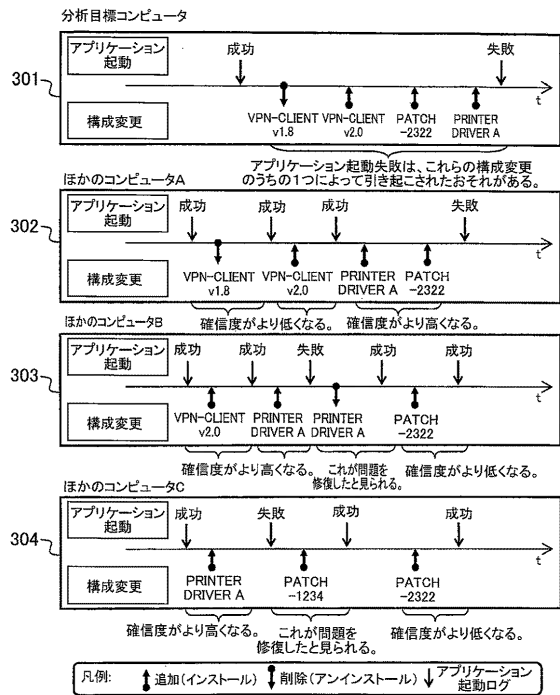
【図 2】



情報システム内の機能的な関係

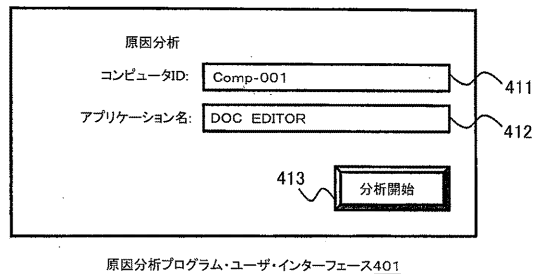
【図3】

【図3】



【図4】

【図4】



【図5】

【図5】

可能性のある原因構成変更

構成項目	変更タイプ	変更日時	失敗率対成功率 失敗数対成功数	
			失敗率	成功率
PRINTER DRIVER A	追加	06/04/2008 08:20:11	80.00%	20.00%
VPN-CLIENT v2.0	追加	06/03/2008 14:27:35	525	13 / 367
VPN-CLIENT v1.8	削除	06/02/2008 13:27:35	2.89%	97.11%
PATCH-2322	追加	06/04/2008 07:43:11	1.78%	98.22%

確信度: 高 (511-514), 低 (515)

可能性のある原因構成変更

構成項目	変更タイプ	修復率対非修復率 修復数対非修復数	修復率対非修復率 修復数対非修復数	
			修復率	非修復率
PRINTER DRIVER A	削除	87.50%	12.50%	
PATCH-1234	追加	62.50%	37.50%	
MEMORY=1024	追加	16.67%	83.33%	

確信度: 高 (531-532), 低 (534)

分析結果ユーザ・インターフェース501

【図6】

【図6】

コンピュータID	日時	イベント・タイプ
Comp-001	06/05/2008 13:41:19	Event-001
Comp-001	06/05/2008 13:35:43	Event-001
Comp-001	06/04/2008 08:29:24	Event-001
...	...	...
Comp-002	05/02/2008 10:29:25	Event-001
Comp-002	05/02/2008 07:28:39	Event-001
...	...	...
Comp-003	05/02/2008 11:29:25	Event-001
...	...	...
Comp-006	05/02/2008 11:29:25	Event-001
...	...	...
Comp-007	05/15/2008 13:26:25	Event-001
Comp-007	05/12/2008 15:23:48	Event-001
...	...	...

イベント・ログ・テーブル141



【図7】

【図7】

701	702	703	704
コンピュータID	変更日時	構成項目	変更タイプ
Comp-001	06/04/2008 08:20:11	PRINTER DRIVER A	追加
Comp-001	06/04/2008 07:43:11	PATCH-2322	追加
Comp-001	06/03/2008 14:27:35	VPN-CLIENT v2.0	追加
Comp-001	06/02/2008 13:27:35	VPN-CLIENT v1.8	削除
Comp-001	06/01/2008 10:41:21	1024MB	追加
Comp-001	05/30/2008 16:10:13	DESKTOP SEARCH	追加
...	...	...	...
Comp-002	06/04/2008 08:49:46	PATCH-2322	追加
Comp-002	06/03/2008 16:53:11	PRINTER DRIVER A	追加
Comp-002	05/21/2008 14:25:48	MEDIA PLAYER	追加
Comp-002	04/24/2008 11:37:34	VPN-CLIENT v2.0	追加
Comp-002	04/23/2008 10:20:17	VPN-CLIENT v1.8	削除
Comp-002	04/08/2008 10:20:35	VPN-CLIENT v1.8	追加
...	...	...	...
Comp-003	05/06/2008 07:43:11	PATCH-2322	追加
Comp-003	05/04/2008 14:27:35	VPN-CLIENT v1.8	追加
Comp-003	05/04/2008 08:30:32	PRINTER DRIVER A	削除
Comp-003	05/01/2008 08:20:11	PRINTER DRIVER A	追加
...	...	...	...
Comp-004	05/06/2008 07:43:11	PATCH-2322	追加
Comp-004	05/04/2008 14:27:35	VPN-CLIENT v2.0	追加
Comp-004	05/01/2008 08:20:11	VPN-CLIENT v1.8	追加
...	...	...	...
Comp-005	05/06/2008 07:43:11	PATCH-2322	追加
Comp-005	05/04/2008 14:27:35	VPN-CLIENT v2.0	追加
...	...	...	...
Comp-006	05/05/2008 10:14:54	PATCH123	追加
Comp-006	05/01/2008 08:20:11	PRINTER DRIVER A	追加
...	...	...	...

構成変更履歴テーブル143

【図8】

【図8】

801	802	803
コンピュータID	起動日時	アプリケーション名
...	...	...
Comp-001	06/05/2008 13:41:18	DOC EDITOR
Comp-001	06/05/2008 13:35:42	DOC EDITOR
Comp-001	06/04/2008 08:29:23	DOC EDITOR
Comp-001	06/03/2008 12:13:14	MAILER
Comp-001	06/03/2008 12:10:54	VPN18
Comp-001	06/03/2008 12:03:54	ANTI-VIRUS
Comp-001	06/03/2008 12:03:59	BACKUP-AGENT
Comp-001	06/02/2008 14:26:03	DOC EDITOR
Comp-001	06/02/2008 14:22:47	MAILER
Comp-001	06/02/2008 10:29:23	DOC EDITOR
...	...	...
Comp-002	05/02/2008 10:29:23	DOC EDITOR
Comp-002	05/02/2008 07:27:23	DOC EDITOR
...	...	...
Comp-003	05/06/2008 07:45:23	DOC EDITOR
Comp-003	05/05/2008 10:30:24	DOC EDITOR
Comp-003	05/02/2008 11:29:23	DOC EDITOR
Comp-003	05/01/2008 07:45:23	DOC EDITOR
Comp-003	04/28/2008 10:29:23	DOC EDITOR
...	...	...
Comp-004	05/06/2008 07:45:23	DOC EDITOR
Comp-004	05/02/2008 11:29:23	DOC EDITOR
Comp-004	05/01/2008 07:45:23	DOC EDITOR
Comp-004	04/28/2008 10:29:23	DOC EDITOR
...	...	...
Comp-005	05/06/2008 07:45:23	DOC EDITOR
...	...	...

アプリケーション起動履歴テーブル142

【図9】

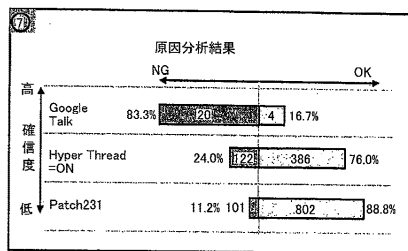
【図9】

901	902	903	904	905	906
コンピュータID	変更日時	構成項目	変更タイプ	起動-前	起動-後
Comp-001	06/04/2008 08:20:11	PRINTER DRIVER A	追加	成功	失敗
Comp-001	06/04/2008 07:43:11	PATCH-2322	追加	成功	失敗
Comp-001	06/03/2008 14:27:35	VPN-CLIENT v2.0	追加	成功	失敗
Comp-001	06/03/2008 13:59:28	VPN-CLIENT v1.8	削除	成功	失敗
...	...	...	...	...	...
Comp-002	06/04/2008 08:49:46	PATCH-2322	追加	成功	失敗
Comp-002	06/03/2008 16:53:11	PRINTER DRIVER A	追加	成功	失敗
Comp-002	04/24/2008 11:37:34	VPN-CLIENT v2.0	追加	成功	成功
Comp-002	04/23/2008 10:20:17	VPN-CLIENT v1.8	削除	成功	成功
...	...	...	...	...	...
Comp-003	05/06/2008 07:43:11	PATCH-2322	追加	成功	成功
Comp-003	05/01/2008 08:20:11	PRINTER DRIVER A	追加	成功	失敗
...	...	...	...	...	...
Comp-004	05/06/2008 07:43:11	PATCH-2322	追加	成功	成功
Comp-004	05/04/2008 14:27:35	VPN-CLIENT v2.0	追加	成功	成功
...	...	...	...	...	...
Comp-005	05/06/2008 07:43:11	PATCH-2322	追加	不明	成功
Comp-005	05/04/2008 14:27:35	VPN-CLIENT v2.0	追加	不明	成功
...	...	...	...	...	...
Comp-006	05/01/2008 08:20:11	PRINTER DRIVER A	追加	成功	失敗
...	...	...	...	...	...
Comp-007	05/11/2008 08:20:11	PRINTER DRIVER A	追加	成功	失敗
...	...	...	...	...	...

原因構成変更一時テーブル144

【図10】

【図10】



1001	1002	1003	1004	1005
構成項目	変更タイプ	変更日時	失敗事例の数	全事例の数
PRINTER DRIVER A	追加	06/04/2008 08:20:11	12	15
VPN-CLIENT v2.0	追加	06/03/2008 14:27:35	13	367
VPN-CLIENT v1.8	削除	06/03/2008 13:59:28	11	370
PATCH-2322	追加	06/04/2008 07:43:11	15	830

原因構成変更テーブル146

【図 1 1】

【図 11】

1101 コンピュータID	1102 変更日時	1103 構成項目	1104 変更タイプ	1105 起動-前	1106 起動-後	
Comp-001	06/04/2008 08:20:11	PRINTER DRIVER A	追加	成功	失敗	~1111
Comp-001	06/04/2008 07:43:11	PATCH-2322	追加	成功	失敗	~1112
Comp-001	06/03/2008 14:27:35	VPN-CLIENT v2.0	追加	成功	失敗	~1113
...	...	...	...	...	...	
Comp-002	06/04/2008 08:49:46	PATCH-2322	追加	成功	失敗	~1121
...	...	...	...	...	...	
Comp-003	05/04/2008 08:30:32	PRINTER DRIVER A	削除	失敗	成功	~1131
...	...	...	...	...	...	
Comp-006	05/05/2008 10:14:54	PATCH-123	追加	失敗	成功	~1141
...	...	...	...	...	...	

修復構成変更一時テーブル145

【図 1 2】

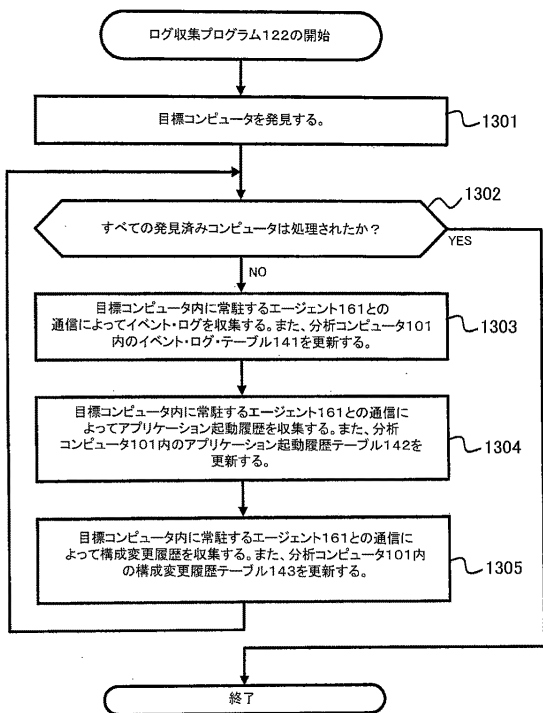
【図 12】

1201 構成項目	1202 変更タイプ	1203 修復事例の数	1204 全事例の数	
PRINTER DRIVER A	削除	29	33	~1211
PATCH-1234	追加	4	10	~1212
MEMORY=2048	追加	1	9	~1213

修復構成変更テーブル147

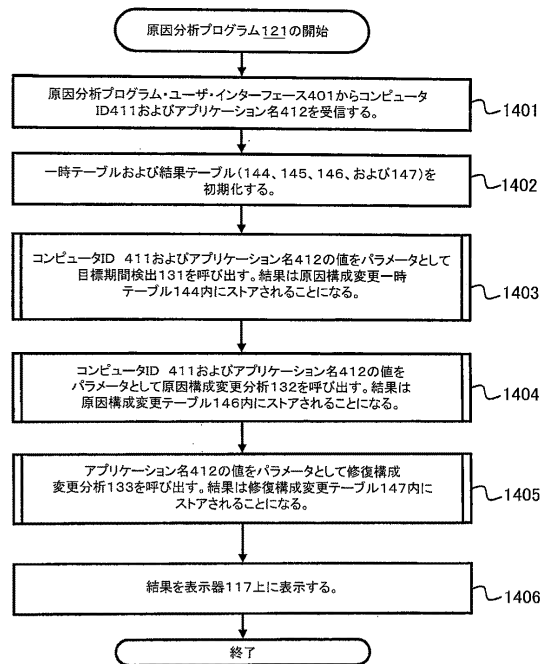
【図 1 3】

【図 13】



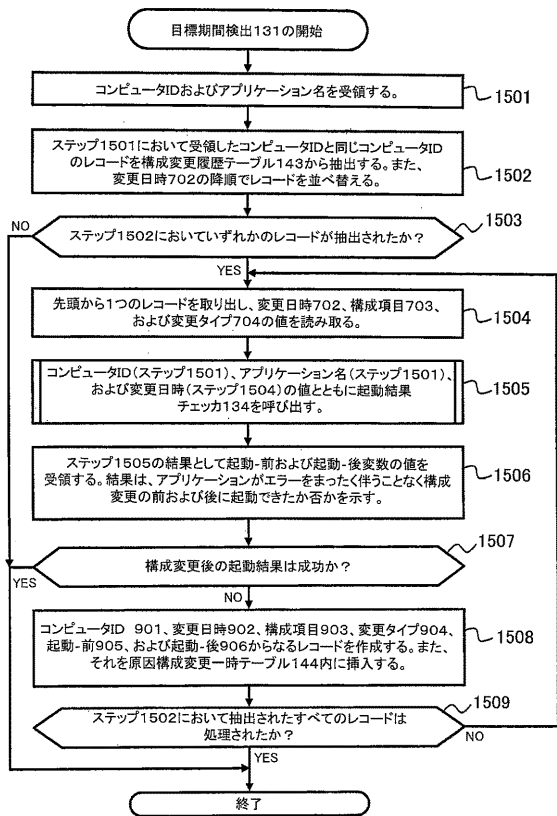
【図 1 4】

【図 14】



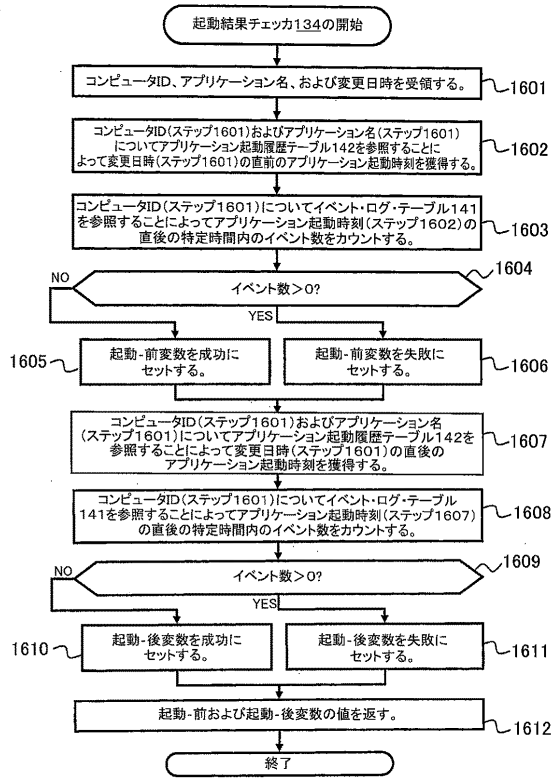
【図 15】

【図 15】



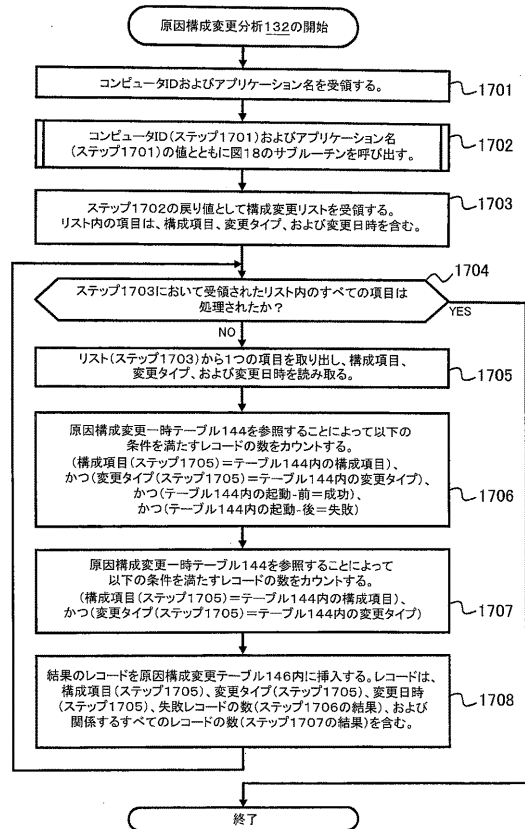
【図 16】

【図 16】



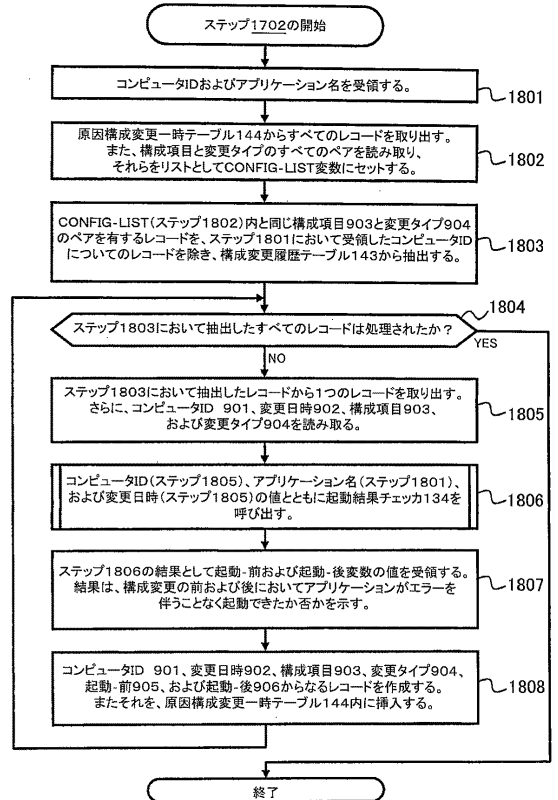
【図 17】

【図 17】



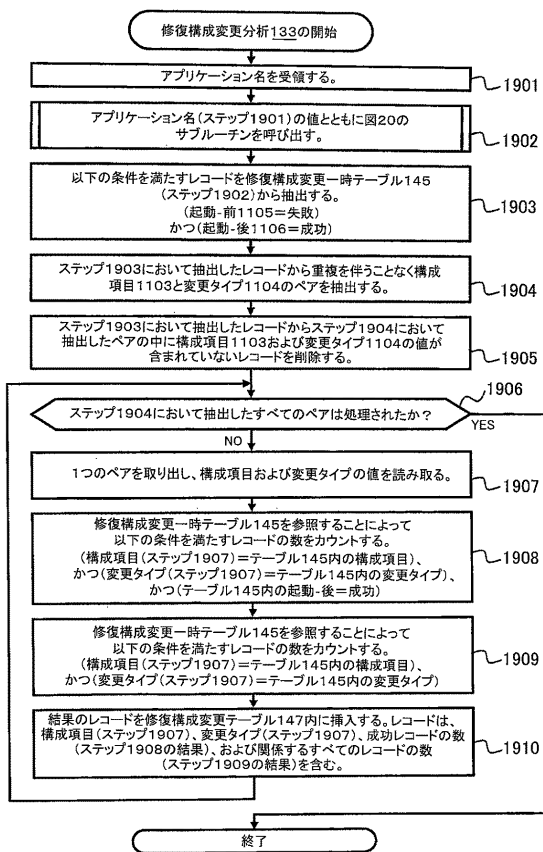
【図 18】

【図 18】



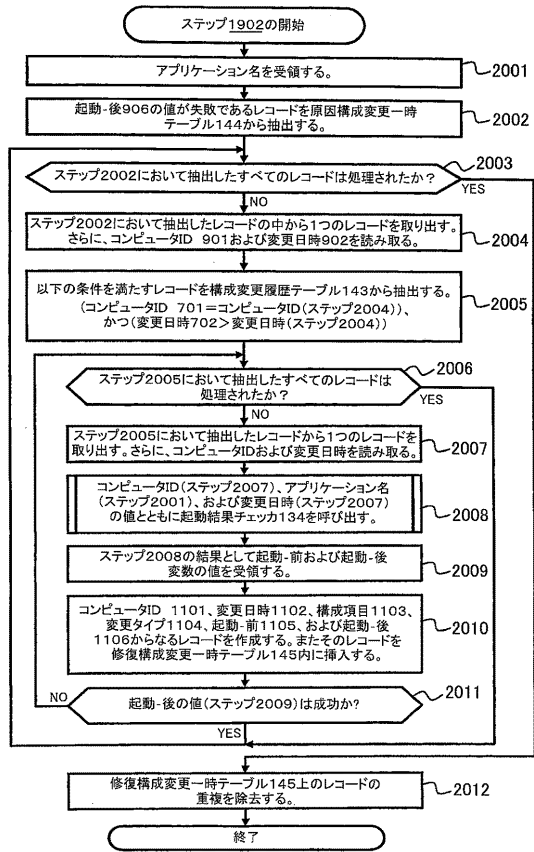
【図 19】

【図 19】



【図 20】

【図 20】



【図 21】

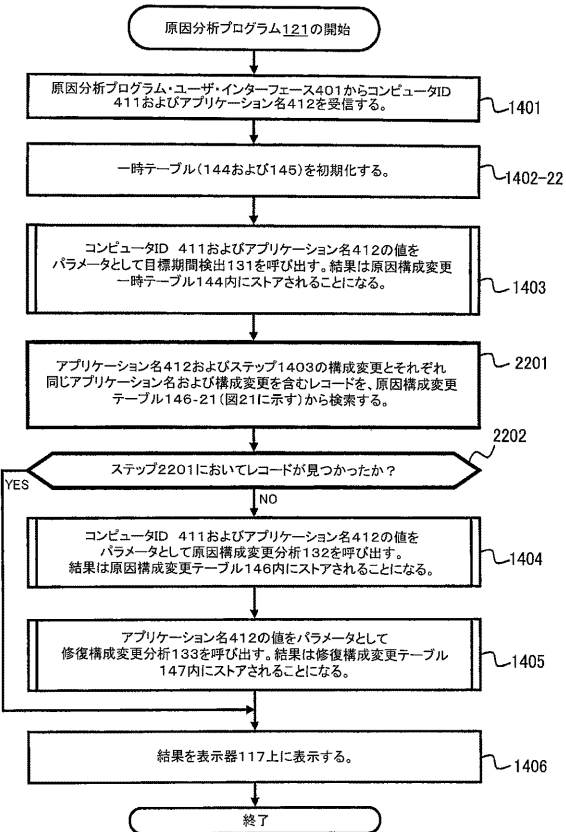
【図 21】

アプリケーション名	分析日時	構成項目	変更タイプ	変更日時	失敗事例の数	全事例の数
DOC EDITOR	06/05/2008 14:20:12	PRINTER DRIVER A	追加	06/04/2008 08:20:11	12	15
		VPN-CLIENT v2.0	追加	06/03/2008 14:27:35	13	367
		VPN-CLIENT v1.8	削除	06/03/2008 13:59:25	11	370
		PATCH-2322	追加	06/04/2008 07:43:11	15	830
WEB BROWSER	06/05/2008 14:20:12	...	...	...	...	...
...	...	...	...	...	...	...

原因構成変更テーブル146-21

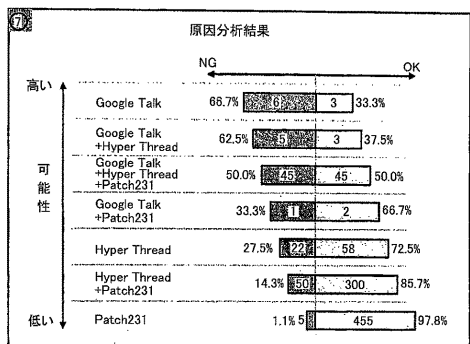
【図 22】

【図 22】



【図 23】

【図 23】

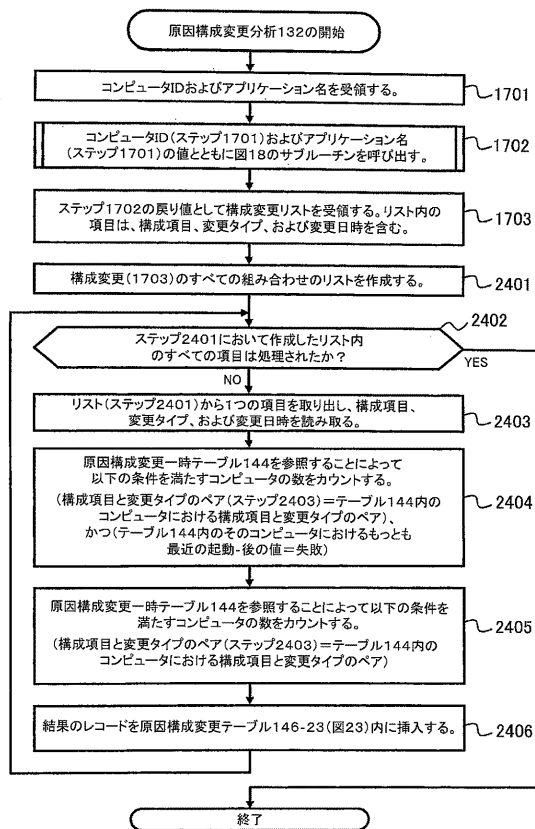


組み合わせ ID	構成項目	変更タイプ	変更日時	失敗事例の数	全事例の数
2311	PRINTER DRIVER A	追加	06/04/2008 08:20:11	45	90
	VPN-CLIENT v2.0	追加	06/03/2008 14:27:35		
	VPN-CLIENT v1.8	削除	06/03/2008 13:59:28		
2312	PRINTER DRIVER A	追加	06/04/2008 08:20:11	5	8
	VPN-CLIENT v2.0	追加	06/03/2008 14:27:35		
2313	VPN-CLIENT v2.0	追加	06/03/2008 14:27:35	50	300
	VPN-CLIENT v1.8	削除	06/03/2008 13:59:28		
2314	PRINTER DRIVER A	追加	06/04/2008 08:20:11	1	2
	VPN-CLIENT v1.8	削除	06/03/2008 13:59:28		
2315	PRINTER DRIVER A	追加	06/04/2008 08:20:11	6	9
2316	VPN-CLIENT v2.0	追加	06/03/2008 14:27:35	5	455
	VPN-CLIENT v1.8	削除	06/03/2008 13:59:28		
2317	VPN-CLIENT v1.8	削除	06/03/2008 13:59:28	22	58

原因構成変更テーブル146-23

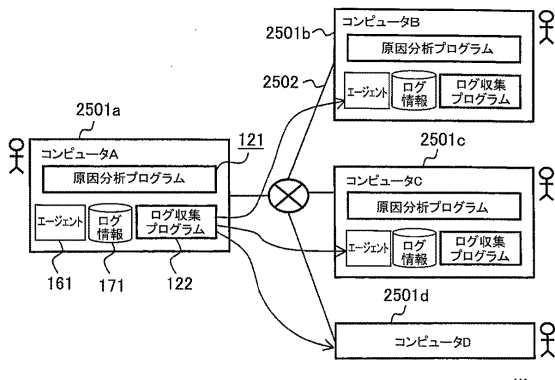
【図 24】

【図 24】



【図 25】

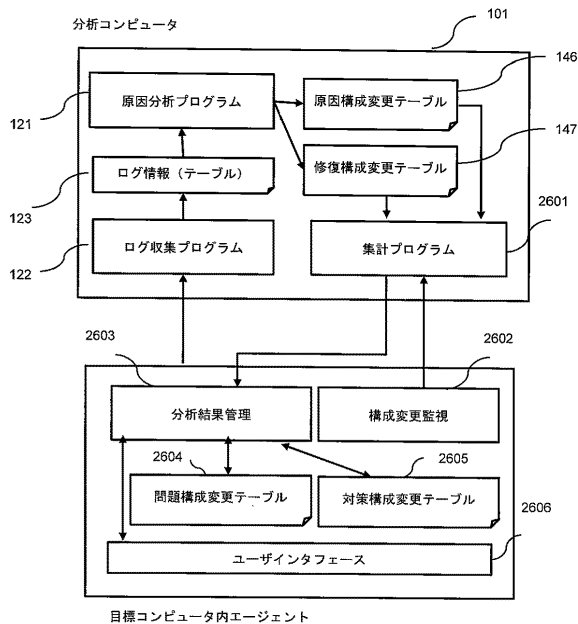
【図 25】



ピア・ツー・ピア・アーキテクチャのための例示的なハードウェアおよび論理構成

【図 26】

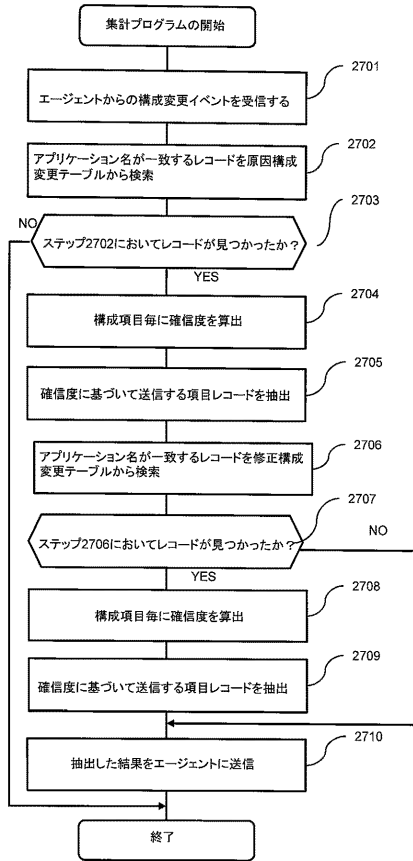
【図 26】



目標コンピュータ内エージェント

【図 27】

【図 27】



【図 28】

【図 28】

【トラブル情報】  
ソフトウェア名 DOG EDITOR 集計日06/05/2008

原因の可能性のある構成変更	失敗率	メッセージ
PRINTER DRIVER A 追加	80%	<警告> × × ×
VPN-CLIENT v2.0 追加	3.4%	<情報>...

【図 29】

【図 29】

【対策情報】  
ソフトウェア名 DOG EDITOR 集計日06/05/2008

可能性のある構成変更	成功率	メッセージ
PRINTER DRIVER A 削除	87.5%	× × ×
PATCH-1234 追加	62.5%	...

【図 30】

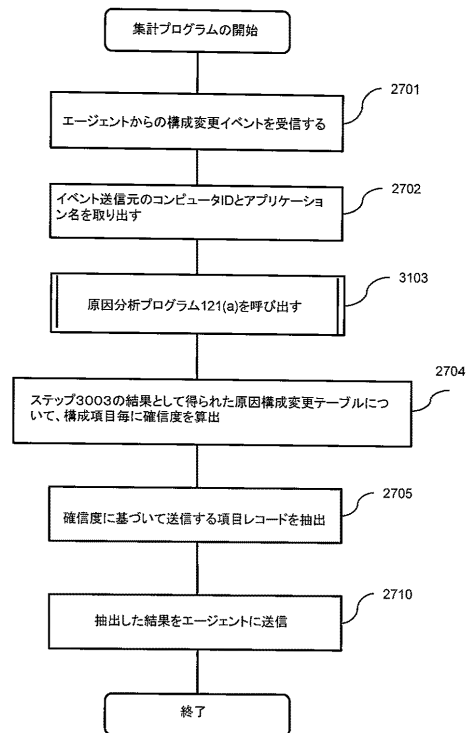
【図 30】

【トラブル情報】  
ソフトウェア名 PATCH-2322 追加 集計日06/05/2008

影響するアプリケーション	失敗率	メッセージ
DOG EDITOR	1.7%	<情報> × × ×
WEB BROWSER	...	...

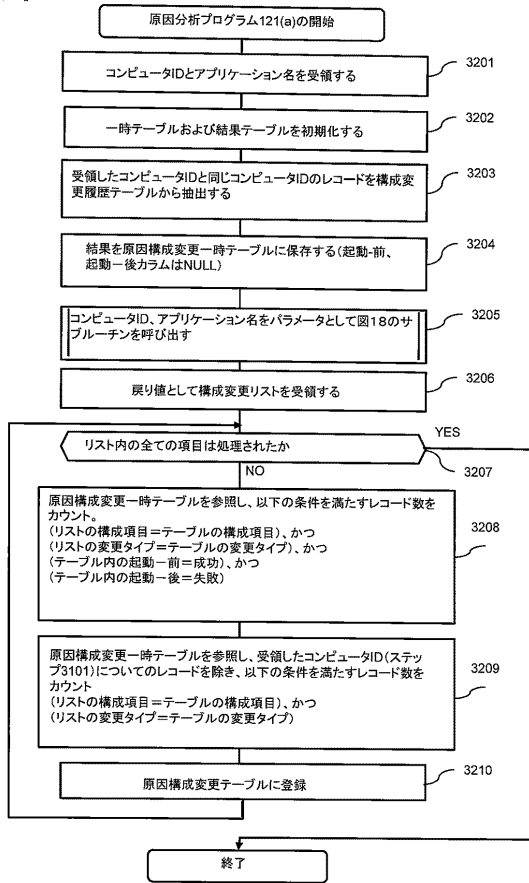
【図 31】

【図 31】



【図 32】

【図 32】



## フロントページの続き

- (72)発明者 菅内 公德  
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所 システム開発研究所内
- (72)発明者 増石 哲也  
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア事業部内
- (72)発明者 藤田 高広  
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア事業部内
- (72)発明者 小野 能嗣  
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア事業部内

審査官 多胡 滋

- (56)参考文献 特開2007-257184(JP,A)  
特開2007-293393(JP,A)  
特開2010-086516(JP,A)  
特開2004-013208(JP,A)  
特開平11-327945(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/445  
G06F 11/30  
G06F 11/34