(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6210010号
(P6210010)

(45)発行日　平成29年10月11日(2017.10.11)　(24)登録日　平成29年9月22日(2017.9.22)

(51)Int.Cl.　F I
G06F 9/445 (2006.01)　G06F 9/06 610L

請求項の数 7 (全 23 頁)

(21)出願番号　特願2014-66174(P2014-66174)
(22)出願日　平成26年3月27日(2014.3.27)
(65)公開番号　特開2015-191278(P2015-191278A)
(43)公開日　平成27年11月2日(2015.11.2)
審査請求日　平成28年12月6日(2016.12.6)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100103528
弁理士　原田　一男
(72)発明者　飯倉　二美
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　松本　安英
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

審査官　坂庭　剛史

最終頁に続く

(54)【発明の名称】情報処理プログラム、装置及び方法

(57)【特許請求の範囲】
【請求項１】
ソフトウエアの被インストール装置から、前記ソフトウエアの識別子と、前記ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における前記被インストール装置の第１の構成データと、前記エラーの解消時における前記被インストール装置の第２の構成データとを取得し、
前記ソフトウエアの識別子に対応付けて格納部に前記ソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと前記第１の構成データ及び前記第２の構成データとに基づき、前記ソフトウエアの識別子及び前記エラーの識別子に対応する対処データを更新し、
前記格納部に前記既存の対処データが格納されていない場合には、前記第１の構成データ及び前記第２の構成データとの差分から対処データを生成し、前記ソフトウエアの識別子及び前記エラーの識別子に対応付けて前記格納部に格納する
処理を、コンピュータに実行させるためのプログラム。
【請求項２】
前記既存の対処データを更新する処理が、
前記第１の構成データと前記第２の構成データとの差分から第１の対処データを生成し、
前記ソフトウエアの識別子及び前記エラーの識別子に対応付けられている第１の既存の対処データに含まれないが前記第１の対処データに含まれる対処ステップが存在する場合

には、当該対処ステップを前記第１の既存の対処データに追加することで前記第１の既存の対処データを更新する
処理を含む請求項１記載のプログラム。
【請求項３】
前記既存の対処データを更新する処理が、
前記格納部において前記ソフトウエアの識別子及び前記エラーの識別子とは異なるエラーの識別子に対応付けられている第２の既存の対処データに係る第２の構成データと、更新後の前記第１の既存の対処データ又は前記第１の対処データに係る第２の構成データとの差を特定し、
前記差に対応する、更新後の前記第１の既存の対処データ又は前記第１の対処データにおける対処ステップを削除する
処理をさらに含む請求項２記載のプログラム。
【請求項４】
ソフトウエアの識別子とエラーの識別子と被インストール装置の識別子とを含む検索要求を受信し、
前記ソフトウエアの識別子及び前記エラーの識別子に対応付けられた対処データを前記格納部から抽出し、
抽出された前記対処データに含まれる対処ステップのうち、前記検索要求に係るエラー発生時における前記被インストール装置の第３の構成データにおいて対応するデータが含まれる対処ステップを削除し、
抽出された前記対処データを出力する
処理をさらに前記コンピュータに実行させるための請求項１乃至３のいずれか１つ記載のプログラム。
【請求項５】
前記エラー発生時及び前記エラーの解消時が、前記ソフトウエアにより出力されるログを読み出すことで特定される
請求項１乃至４のいずれか１つ記載のプログラム。
【請求項６】
ソフトウエアの被インストール装置から、前記ソフトウエアの識別子と、前記ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における前記被インストール装置の第１の構成データと、前記エラーの解消時における前記被インストール装置の第２の構成データとを取得し、
前記ソフトウエアの識別子に対応付けて格納部に前記ソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと前記第１の構成データ及び前記第２の構成データとに基づき、前記ソフトウエアの識別子及び前記エラーの識別子に対応する対処データを更新し、
前記格納部に前記既存の対処データが格納されていない場合には、前記第１の構成データ及び前記第２の構成データとの差分から対処データを生成し、前記ソフトウエアの識別子及び前記エラーの識別子に対応付けて前記格納部に格納する
処理を、コンピュータが実行する情報処理方法。
【請求項７】
ソフトウエアの被インストール装置から、前記ソフトウエアの識別子と、前記ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における前記被インストール装置の第１の構成データと、前記エラーの解消時における前記被インストール装置の第２の構成データとを取得する取得部と、
前記ソフトウエアの識別子に対応付けて格納部に前記ソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと前記第１の構成データ及び前記第２の構成データとに基づき、前記ソフトウエアの識別子及び前記エラーの識別子に対応する対処データを更新し、前記格納部に前記既存の対処データが格納されていない場合には、前記第１の構成データ及び前記第２の構成

データとの差分から対処データを生成し、前記ソフトウエアの識別子及び前記エラーの識別子に対応付けて前記格納部に格納する更新部と、
を有する情報処理装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ソフトウエアのインストール時に発生するエラーに対処するための技術に関する。
【背景技術】
【０００２】
今までインストールしたことがないソフトウエアをインストールする際、インストールの手順やパラメータの設定を誤ることによりエラーメッセージが出力されることがある。インストールしたことがないソフトウエアのエラーメッセージに対する対処方法が不明な場合、操作者は、典型的には、インターネットなどで検索する等して対処法を探すことになる。しかしながら、対処法が見つからなかったり、見つかったとしてもエラーが解決しない対処法であったりして、早期に適切な対処法にたどり着くのは難しい。
【０００３】
通常運用時におけるエラーに対して対処法を蓄積して再利用するような技術は存在しているが、インストール時に発生するエラーに着目したものはない。インストール時は、関連するソフトウエア構成が整っていないためにエラーが発生することがあるが、通常運用時にはソフトウエア構成が整っていないためにエラーが発生するわけではない。知識のない操作者にとっては、あるソフトウエアをインストールする際に求められるソフトウエア構成がどのような状態であってもそのための対処法が提示される方が好ましい。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２００５－３４６３３１号公報
【特許文献２】特開２００３－１１４８１１号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
従って、本発明の目的は、一側面によれば、複数の装置に対して共通して適用可能な対処データを生成できるようにするための技術を提供することである。
【課題を解決するための手段】
【０００６】
本発明に係る情報処理方法は、（Ａ）ソフトウエアの被インストール装置から、ソフトウエアの識別子と、ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における被インストール装置の第１の構成データと、エラーの解消時における被インストール装置の第２の構成データとを取得し、（Ｂ）ソフトウエアの識別子に対応付けて格納部にソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと第１の構成データ及び第２の構成データとに基づき、ソフトウエアの識別子及びエラーの識別子に対応する対処データを更新し、（Ｃ）格納部に既存の対処データが格納されていない場合には、第１の構成データ及び第２の構成データとの差分から対処データを生成し、ソフトウエアの識別子及びエラーの識別子に対応付けて格納部に格納する処理を含む。
【発明の効果】
【０００７】
一側面としては、複数の装置に対して共通して適用可能な対処データを生成できるようになる。
【図面の簡単な説明】

【０００８】
【図１】図１は、実施の形態に係るシステムの概要を示す図である。
【図２】図２は、運用管理サーバの構成例を示す図である。
【図３】図３は、第１の例においてパッケージ管理ＤＢに格納されるデータの例を示す図である。
【図４】図４は、第１の例において構成ＤＢに格納されるデータの例を示す図である。
【図５】図５は、第１の例においてログデータとパッケージ管理ＤＢと構成ＤＢとの関係を示す図である。
【図６】図６は、第２の例においてパッケージ管理ＤＢに格納される初期的なデータの例を示す図である。
【図７】図７は、第２の例におけるログデータとパッケージ管理ＤＢと構成ＤＢとの関係を示す図である。
【図８】図８は、第２の例においてパッケージ管理ＤＢに格納されるデータの例を示す図である。
【図９】図９は、第２の例において構成ＤＢに格納されるデータの例を示す図である。
【図１０】図１０は、第３の例においてパッケージ管理ＤＢに格納される初期的なデータの例を示す図である。
【図１１】図１１は、第３の例におけるログデータとパッケージ管理ＤＢと構成ＤＢとの関係を示す図である。
【図１２】図１２は、第３の例においてパッケージ管理ＤＢに格納されるデータの例を示す図である。
【図１３】図１３は、第３の例において構成ＤＢに格納されるデータの例を示す図である。
【図１４】図１４は、ソフトウエアパッケージのインストール時における処理の処理フローを示す図である。
【図１５】図１５は、ソフトウエア管理表の一例を示す図である。
【図１６】図１６は、エラー検出時の処理の処理フローを示す図である。
【図１７】図１７は、サーバから運用管理サーバへ送信するデータのフォーマット例を示す図である。
【図１８】図１８は、対処１におけるエラー発生時の構成データ例を示す図である。
【図１９】図１９は、対処１におけるエラー解消時の構成データ例を示す図である。
【図２０】図２０は、対処２におけるエラー発生時の構成データ例を示す図である。
【図２１】図２１は、対処２におけるエラー解消時の構成データ例を示す図である。
【図２２】図２２は、対処３におけるエラー発生時の構成データ例を示す図である。
【図２３】図２３は、対処３におけるエラー解消時の構成データ例を示す図である。
【図２４】図２４は、対処データの更新のための処理の処理フローを示す図である。
【図２５】図２５は、対処１についての初期対処データを示す図である。
【図２６】図２６は、対処２についての初期対処データを示す図である。
【図２７】図２７は、対処３についての初期対処データを示す図である。
【図２８】図２８は、対処データの更新例を示す図である。
【図２９】図２９は、対処４についての初期対処データを示す図である。
【図３０】図３０は、対処データの更新例を示す図である。
【図３１】図３１は、対処データの更新のための処理の処理フローを示す図である。
【図３２】図３２は、対処データの更新例を示す図である。
【図３３】図３３は、対処ＤＢに登録される対処データの一例を示す図である。
【図３４】図３４は、ある装置の構成データを示す図である。
【図３５】図３５は、別のある装置の構成データを示す図である。
【図３６】図３６は、検索処理の処理フローを示す図である。
【図３７】図３７は、ある装置の場合に提示される対処候補例を示す図である。
【図３８】図３８は、別のある装置の場合に提示される対処候補例を示す図である。

【図３９】図３９は、コンピュータの機能ブロック図である。
【発明を実施するための形態】
【０００９】
本発明の実施の形態に係るシステムの概要を図１に示す。ネットワーク２００には、あるソフトウエアパッケージのインストール先となる１又は複数のサーバ３００が接続されており、さらに本実施の形態に係る主要な処理を実行する運用管理サーバ１００も接続されている。ユーザ端末４００は、サーバ３００に対してソフトウエアパッケージをインストールする際にユーザにより操作される端末装置であり、ユーザ端末４０２のようにサーバ３００に接続されている場合もあれば、ユーザ端末４０１にようにネットワーク２００に接続されている場合もある。なお、インストール時にエラーが発生した場合に、ユーザ端末４００のオペレータは、ユーザ端末４００に運用管理サーバ１００にアクセスさせ、対処方法のデータを取得するようになっている。
【００１０】
サーバ３００は、収集部３０６と、構成管理部３０４と、パッケージ管理部３０８と、構成データベース（ＤＢ）３０５と、パッケージ管理ＤＢ３０９とを有する。
【００１１】
ユーザ端末４００のオペレータは、サーバ３００に、ソフトウエアパッケージをインストールする。インストール時に、例えばＯＳ（Operating System）から通知を受けた収集部３０６は、そのソフトウエアパッケージに含まれるプログラム３０１のログ出力先（例えばログファイル３０２）及びパラメータの格納場所（例えばパラメータファイル３０３）のデータを例えばソフトウエアパッケージのパッケージ情報から取得して、ソフトウエア管理表３０７に登録する。本実施の形態では、プログラム３０１が出力するログをモニタするので、ログの出力先が特定できない場合には、収集部３０６は、ソフトウエア管理表３０７にソフトウエアパッケージのデータを登録しない。また、パラメータ格納場所のデータが取得できない場合にも、ソフトウエア管理表３０７にソフトウエアパッケージのデータを登録しない。
【００１２】
構成管理部３０４は、例えばソフトウエア管理表３０７から、プログラム３０１のパラメータ格納場所（例えばパラメータファイル３０３）を特定し、そのパラメータ格納場所からパラメータを読み出して、構成ＤＢ３０５に格納する。但し、構成管理部３０４は、ソフトウエア管理表３０７ではなく別途取得したパラメータ格納場所からパラメータを読み出して、構成ＤＢ３０５に格納しても良い。パッケージ管理部３０８は、例えばＯＳからインストールされたソフトウエアパッケージのデータを取得して、パッケージ管理ＤＢ３０９に格納する。
【００１３】
収集部３０６は、ログの出力先からログを読み出し、エラーの発生を検出すると、構成ＤＢ３０５及びパッケージ管理ＤＢ３０９から構成データを読み出す。また、収集部３０６は、ログからエラーの解消を検出すると、再度構成ＤＢ３０５及びパッケージ管理ＤＢ３０９から構成データを読み出す。そして、収集部３０６は、ソフトウエアパッケージ名、エラーメッセージ（エラーの識別子の相当）、エラー発生時の構成データ及びエラー解消時の構成データを、運用管理サーバ１００に送信する。さらに、収集部３０６は、エラーが発生した時又は定期的に、構成ＤＢ３０５及びパッケージ管理ＤＢ３０９から構成データを読み出して、運用管理サーバ１００へ送信する。
【００１４】
図２に、運用管理サーバ１００の機能ブロック構成を示す。運用管理サーバ１００は、受信部１０１と、第１構成データ格納部１０２と、対処データ更新部１０３と、対処ＤＢ１０４と、第２構成データ格納部１０５と、検索部１０６とを有する。
【００１５】
受信部１０１は、各サーバ３００から、エラー発生時及びエラー解消時の構成データを受信すると第１構成データ格納部１０２に格納し、ある時点の構成データを受信すると第

２構成データ格納部１０５に格納する。
【００１６】
対処データ更新部１０３は、初期対処データ生成部１０３１と、第１更新部１０３２と、第２更新部１０３３とを有する。
【００１７】
初期対処データ生成部１０３１は、第１構成データ格納部１０２に格納されているエラー発生時及びエラー解消時の構成データから初期的な対処データを生成する。第１更新部１０３２は、対処ＤＢ１０４に既に登録されている、同じソフトウエアパッケージ且つ同じエラーメッセージについての対処データを、初期的な対処データで更新する。第２更新部１０３３は、対処ＤＢ１０４に既に登録されている、同じソフトウエアパッケージ且つ異なるエラーメッセージについての対処データで、今回のエラーメッセージについての対処データを更新する。
【００１８】
検索部１０６は、ユーザ端末４００から、ソフトウエアパッケージの識別子と、エラーメッセージと、サーバ３００の識別子（ユーザ毎に構成が異なる場合にはユーザ識別子を含む）とを含む検索要求を受信すると、対処ＤＢ１０４に格納されている対処データと、第２構成データ格納部１０５に格納された構成データとから、実行すべき対処ステップの候補を抽出して、ユーザ端末４００に返信する。
【００１９】
次に、図３乃至図１３を用いて、サーバ３００に対して行われるインストールの手順について具体例を説明する。
【００２０】
ここではソフトウエアパッケージａｐｐｔｏを導入する場合を説明する。このａｐｐｔｏを導入する場合には、以下のような手順が実行される。
（１）ｌｉｂｘｙｚ－５．１をインストール
（２）ｏｐｅｎｄｂをインストール
（３）ｏｐｅｎｄｂのパスワードをセット（例えばｓｅｃｒｅｔ）
（４）ｏｐｅｎｄｂをスタート
（５）ａｐｐｔｏをインストール
（６）ａｐｐｔｏのｃｏｎｎｅｃｔｔｏパラメータに、ｌｏｃａｌｈｏｓｔ：３３６０をセット
（７）ａｐｐｔｏのｄｂｐａｓｓｗｄパラメータのセット（（３）と同じパスワード）
（８）ａｐｐｔｏをスタート
【００２１】
このような正しい操作を行うと、パッケージ管理ＤＢ３０９には、インストールしたソフトウエアパッケージの順で図３に示すようなデータが格納されるようになる。図３の例では、ソフトウエアパッケージ名と、そのバージョンとが登録される。上で述べた（１）（２）（５）に対応するデータが登録されている。
【００２２】
また、構成ＤＢ３０５には、図４に示すようなデータが、操作の順に登録される。図４の例では、コンポーネント名と、パラメータと、そのパラメータの値とが登録されるようになっている。上で述べた（３）（６）（７）についてのデータが登録されている。
【００２３】
これに対してログファイル３０２と、パッケージ管理ＤＢ３０９及び構成ＤＢ３０５に格納されているデータとの対応関係を図５に示す。
【００２４】
上で述べた例ではエラーは発生していないので、ａｐｐｔｏのログファイル３０２には、（８）ａｐｐｔｏをスタートに応じて［ｉｎｆｏ］ｐｒｅｐａｒｉｎｇ及び［ｉｎｆｏ］ｓｔａｒｔｉｎｇのみが登録される。一方、パッケージ管理ＤＢ３０９には上で述べたように（１）（２）（５）に対応するデータが登録される。よって、パッケージ管理ＤＢ

３０９から、（１）に対応する操作である対処ステップ「Ｉｎｓｔａｌｌ　ｌｉｂｘｙｚ－５．１」と、（２）に対応する操作である対処ステップ「Ｉｎｓｔａｌｌ　ｏｐｅｎｄｂ－４．２」と、（５）に対応する操作である対処ステップ「Ｉｎｓｔａｌｌ　ａｐｐｔｏ－１．５」とが得られる。

【００２５】

また、構成ＤＢ３０５には上で述べたように（３）（６）（７）に対応するデータが登録される。よって、構成ＤＢ３０５から、（３）に対応する操作である対処ステップである「ｏｐｅｎｄｂ，ｐａｓｓｗｄ＝ｓｅｃｒｅｔ」と、（６）に対応する操作である対処ステップ「ａｐｐｔｏ，ｃｏｎｎｅｃｔｔｏ＝ｌｏｃａｌｈｏｓｔ：３３６０」と、（７）に対応する操作である対処ステップ「ａｐｐｔｏ，ｄｂｐａｓｓｗｄ＝ｓｅｃｒｅｔ」とが得られる。

【００２６】

一方、正しい操作が行われない場合も示す。例えば、（１）のｌｉｂｘｙｚのインストールを忘れて、（３）のｏｐｅｎｄｂのパスワードのセットを忘れ、無駄にｌｉｂａｂｃ－３．３をインストールしてしまった場合を示す。本例では、さらにｏｐｅｎｄｂについては既にインストール済みであるものとする。すなわち、１０：１４の時点で、図６に示すようなパッケージ管理ＤＢ３０９の状態になっている。

【００２７】

図７に、ａｐｐｔｏのログを示す。この例では、１０：１５でａｐｐｔｏ－１．５をインストールして、いきなりａｐｐｔｏをスタートさせたので、ａｐｐｔｏのログには、エラーメッセージ「ｍｉｓｓｉｎｇ　ｌｉｂｒａｒｙ」が登録される。これに対して、オペレータは、１０：１７にｌｉｂｘｙｚ－５．１をインストールすると共に１０：１８にｌｉｂａｂｃ－３．３をインストールしている。従って、パッケージ管理ＤＢ３０９には、図８に示すようなデータが格納されるようになる。この時点で、パッケージ管理ＤＢ３０９からは、既にインストールされていた分を除き、図７に示すような２つの対処ステップが得られる。

【００２８】

１０：１９で再度ａｐｐｔｏをスタートさせたので、ａｐｐｔｏのログには、エラーメッセージ「ｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｔｏ　ｄａｔａｂａｓｅ」が登録される。これに対して、オペレータは、ｏｐｅｎｄｂ及びａｐｐｔｏについてのパラメータの設定を行うので、構成ＤＢ３０５に図９に示すようなデータが格納される。

【００２９】

このようにしてからａｐｐｔｏをスタートさせると、１０：２３に［ｉｎｆｏ］ｓｔａｒｔｉｎｇというログが格納される。これによってエラーが解消したことが分かる。

【００３０】

また、別の例も示しておく。例えば、図１０に示すように、既にｏｐｅｎｄｂ及びｌｉｂｘｙｚがインストール済みであるものとする。

【００３１】

図１１に、ａｐｐｔｏのログを示す。この例では、１０：１５でｏｐｅｎｄｂについてパラメータの設定を行い、１０：１６でａｐｐｔｏをインストールしている。なお、この段階で、パッケージ管理ＤＢ３０９に図１２に示すようなデータが格納される。

【００３２】

さらに、１０：１７で、ａｐｐｔｏについてパラメータの設定し、１０：１８でａｐｐｔｏをスタートさせた。しかしながら、ａｐｐｔｏについてパラメータｄｂｐａｓｓｗｄの設定が行われていないので、エラーメッセージ「ｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｔｏ　ｄａｔａｂａｓｅ」がログとして登録される。

【００３３】

これに対して、オペレータは、１０：１９でａｐｐｔｏのパラメータｄｂｐａｓｓｗｄの設定を行うが、パスワードの綴りが間違っているので、１０：２０で再度ａｐｐｔｏをスタートさせても同じエラーメッセージが出力され、ログファイル３０２に登録される。

これに対して、オペレータは、１０：２１に正しいパラメータの設定を行ったので、１０：２２でａｐｐｔｏをスタートさせると、［ｉｎｆｏ］ｓｔａｒｔｉｎｇというログが格納される。これによってエラーが解消したことが分かる。１０：２１で、パラメータの修正が行われているので、構成ＤＢ３０５には、図１３に示すようなデータが格納される。

【００３４】

このようにパッケージ管理ＤＢ３０９及び構成ＤＢ３０５には、ソフトウエアパッケージをインストールして動作させるために行われた操作に対応するデータが蓄積される。

【００３５】

図７の例では、１０：１６でエラーが発生した時点におけるパッケージ管理ＤＢ３０９及び構成ＤＢ３０５の内容と、１０：２３でエラーが解消した時点におけるパッケージ管理ＤＢ３０９及び構成ＤＢ３０５の内容との差は、図７の３行目から８行目までに対処ステップとして示されており、この部分を抽出することで、エラーに対するこの場合の対処データとなる。

【００３６】

同様に、図１１の例では、１０：１８でエラーが発生した時におけるパッケージ管理ＤＢ３０９及び構成ＤＢ３０５の内容と、１０：２２でエラーが解消した時点におけるパッケージ管理ＤＢ３０９及び構成ＤＢ３０５の内容との差は、７行目の対処ステップで表される。５行目の対処ステップは、構成ＤＢ３０５ではマージされているので、現れない。従って、７行目を抽出することで、エラーに対するこの場合の対処データとなる。

【００３７】

このような特質に基づき、以下に示す処理を行うことで、複数の装置で適用可能な対処データを抽出して、再利用できるようにする。

【００３８】

まず、サーバ３００側の処理について、図１４乃至図１７を用いて説明する。

【００３９】

まず、図１４を用いて、ソフトウエアパッケージのインストール時における収集部３０６の処理について説明する。

【００４０】

まず、収集部３０６は、例えばＯＳからの通知により、新規ソフトウエアパッケージのインストールを検出する（図１４：ステップＳ１）。ここで、パッケージ管理部３０８も、例えばＯＳからの通知により、新規ソフトウエアパッケージのインストールを検出して、パッケージ管理ＤＢ３０９に登録する。

【００４１】

そして、収集部３０６は、そのソフトウエアパッケージのパッケージ情報に、ログの出力先が規定されているか否かを判断する（ステップＳ３）。ログの出力先が規定されていない場合には、本ソフトウエアパッケージについては構成データを抽出できないので、処理を終了する。一方、ログの出力先が規定されている場合には、収集部３０６は、パッケージ情報からログの出力先（例えばログファイル３０２）を特定する（ステップＳ５）。

【００４２】

また、収集部３０６は、パッケージ情報に、パラメータの格納場所が規定されているか否かを判断する（ステップＳ７）。本実施の形態では、パラメータの格納場所が規定されていない場合には、処理を終了する。

【００４３】

一方、パラメータの格納場所が規定されていれば、収集部３０６は、パラメータの格納場所（例えばパラメータファイル３０３）を特定する（ステップＳ９）。

【００４４】

そして、収集部３０６は、ソフトウエア管理表３０７に、ソフトウエア名と、ログ出力先と、パラメータ格納場所とを登録する（ステップＳ１１）。そして処理を終了する。

【００４５】

例えば、ソフトウエア管理表３０７の一例を図１５に示す。図１５の例では、上で述べ

たように、ソフトウエア名と、ログ出力先と、パラメータ格納場所とが格納される。このようなソフトウエア管理表３０７に従って、収集部３０６は、ログの監視を行う。また、構成管理部３０４は、パラメータファイル３０３を特定して、更新があれば、構成ＤＢ３０５を更新する。

【００４６】

次に、図１６を用いて、エラー検出時の処理を説明する。

【００４７】

収集部３０６は、ログファイル３０２から未処理のログを読み出す（ステップＳ２１）。例えば新たに登録されたログを読み出す。また、収集部３０６は、エラー発生後且つエラー解消前の状態であるか否かを表す対処中フラグの状態をチェックする（ステップＳ２３）。

【００４８】

そして、収集部３０６は、対処中フラグがオフとなっており且つログにエラーが発生したか否かを判断する（ステップＳ２５）。エラー発生はエラーメッセージが登録されているか否かで判断する。

【００４９】

この条件を満たす場合には、エラー解消後に新たにエラーが発生した場合を示しているので、収集部３０６は、対処中フラグをオンにセットする（ステップＳ２７）。また、収集部３０６は、対処前（エラー発生時）の構成データを、構成ＤＢ３０５及びパッケージ管理ＤＢ３０９から取得する（ステップＳ２９）。具体的には、構成ＤＢ３０５から、その時点における構成情報を読み出し、パッケージ管理ＤＢ３０９からパッケージ管理データを読み出す。そして処理はステップＳ３９に移行する。

【００５０】

一方、対処中フラグがオフとなっており且つエラーが発生したと判断しなかった場合、収集部３０６は、対処中フラグがオンで且つログが正常稼働中であることを示しているか否かを判断する（ステップＳ３１）。正常稼働中は、上で述べた例では［ｉｎｆｏ］ｓｔａｒｔｉｎｇといった予め定められたメッセージが登録されているか否かで判断する。この条件を満たさない場合には、処理はステップＳ３９に移行する。

【００５１】

一方、対処中フラグがオンで且つログが正常稼働中である場合には、エラーが解消したことになるので、収集部３０６は、対処中フラグをオフにセットする（ステップＳ３３）。また、収集部３０６は、対処後（エラー解消後）の構成データを、構成ＤＢ３０５及びパッケージ管理ＤＢ３０９から取得する（ステップＳ３５）。具体的には、構成ＤＢ３０５から、その時点における構成情報を読み出し、パッケージ管理ＤＢ３０９からパッケージ管理データを読み出す。

【００５２】

そして、収集部３０６は、対処前後の構成データを、運用管理サーバ１００へ送信する（ステップＳ３７）。例えば図１７に示すようなデータを、運用管理サーバ１００へ送信する。

【００５３】

図１７の例では、ユーザＩＤ及びサーバＩＤと、ソフトウエア名と、エラーメッセージと、対処前の構成データ（パッケージ管理データ及び構成情報）と、対処後の構成データ（パッケージ管理データ及び構成情報）とを含む。

【００５４】

そして、収集部３０６は、処理終了が指示されたか否かを判断する（ステップＳ３９）。処理終了でなければ処理はステップＳ２１に戻る。一方、処理終了が指示されると処理は終了する。

【００５５】

このような処理を行うことで、エラー発生時の構成データと、エラー解消後の構成データとを確実に取得でき、運用管理サーバ１００へ送信できる。

【００５６】
次に、運用管理サーバ１００における処理について図１８乃至図３３を用いて説明する。まず、以下の説明で用いる具体例について説明しておく。
【００５７】
図１８は、エラーメッセージｍｉｓｓｉｎｇ　ｌｉｂｒａｒｙのエラーが発生したときの構成データ例を示す図である。なお、構成ＤＢ３０５における構成情報については省略されている。一方、図１９は、エラーメッセージｍｉｓｓｉｎｇ　ｌｉｂｒａｒｙのエラーが解消した時の構成データ例を示す図である。図１８と同様に、構成ＤＢ３０５における構成情報については省略されている。そして、パッケージ管理ＤＢ３０９におけるパッケージ管理データにおいて太線で囲まれた部分が、図１８に示したエラー発生時とは異なる部分である。なお、図１８及び図１９で示した構成データを、対処１の構成データと呼ぶことにする。
【００５８】
図２０は、エラーメッセージｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｄａｔａｂａｓｅのエラーが発生したときの構成データ例を示す図である。なお、構成ＤＢ３０５における構成情報については省略されている。一方、図２１は、エラーメッセージｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｄａｔａｂａｓｅのエラーが解消した時の構成データ例を示す図である。構成情報については、太線で囲まれた追加された部分のみ具体的に示している。そして、構成ＤＢ３０５における構成情報において太線で囲まれた部分が、図２０に示したエラー発生時とは異なる部分である。なお、図２０及び図２１で示した構成データを、対処２の構成データと呼ぶことにする。
【００５９】
図２２は、エラーメッセージｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｄａｔａｂａｓｅのエラーが発生したときの構成データ例を示す図である。なお、構成ＤＢ３０５における構成情報については一部省略されている。一方、図２３は、エラーメッセージｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｄａｔａｂａｓｅのエラーが解消した時の構成データ例を示す図である。そして、構成ＤＢ３０５における構成情報において太線で囲まれた部分が、図２２に示したエラー発生時とは異なる部分である。なお、図２２及び図２３で示した構成データを、対処３の構成データと呼ぶことにする。
【００６０】
次に、図２４乃至図３３を用いて運用管理サーバ１００において対処データを対処ＤＢ１０４に登録する又は対処ＤＢ１０４に登録済みの対処データを更新する際の処理について説明する。
【００６１】
受信部１０１は、サーバ３００から、対処前後の構成データを受信し、第１構成データ格納部１０２に格納する（図２４：ステップＳ４１）。なお、この段階で対処後の構成データを、第２構成データ格納部１０５に格納するようにしても良い。
【００６２】
そして、対処データ更新部１０３の初期対処データ生成部１０３１は、対処前後の構成データの差分を抽出して、初期対処データを生成する（ステップＳ４３）。
【００６３】
例えば対処１の場合、図２５に示すような初期対処データが生成される。図２５の例では、図１９において太線で示したパッケージ管理データの差分に対応する対処ステップ１及び２が含まれる。なお、区別するため、ソフトウエア名及びエラーメッセージについても共に示している。
【００６４】
同様に、対処３の場合、図２７に示すような初期対処データが生成される。図２７の例では、図２３において太線で示した構成データの差分に対応する対処ステップ１が含まれる。なお、区別するため、ソフトウエア名及びエラーメッセージについても共に示している。

【００６５】
さらに、対処２の場合、図２６に示すような初期対処データが生成される。図２６の例では、図２１において太線で示した構成データの差分に対応する対処ステップ１乃至３が含まれる。なお、区別するため、ソフトウエア名及びエラーメッセージについても共に共に示している。
【００６６】
そして、対処データ更新部１０３の第１更新部１０３２は、同じソフトウエアパッケージで発生する同じエラーについて対処データが対処ＤＢ１０４に登録されているか判断する（ステップＳ４５）。
【００６７】
この条件が満たされる場合には、第１更新部１０３２は、既存の対処データに含まれないが初期対処データに含まれる対処ステップを、既存の対処データに追加する（ステップＳ４７）。そして処理はステップＳ４９に移行する。一方、同じソフトウエアパッケージで発生する同じエラーについて対処データが未だ登録されていない場合には、処理はステップＳ４９に移行する。
【００６８】
例えば、最初に、ソフトウエアパッケージ「ａｐｐｔｏ」のエラーメッセージ「ｃｏｕｌｄ　ｎｏｔ　ｃｏｎｎｅｃｔ　ｄａｔａｂａｓｅ」についての対処３から生成された初期対処データ（図２７）がそのまま対処ＤＢ１０４に登録されているものとする。次に同じソフトウエアパッケージで同じエラーメッセージについての対処２から生成された初期対処データ（図２６）が生成されると、図２６における初期対処データにおける対処ステップ１及び２が、ステップＳ４７の条件を満たすことが分かる。従って、既存の対処データに、初期対処データにおける対処ステップ１及び２が追加される。すなわち、図２８に示すようなデータに対処データが更新される。なお、対処後の構成データについては、後の処理で用いられるので、この例の場合対処２の構成データ（図２１）が保持される。
【００６９】
また、この後に、図２９に示すような対処４の初期対処データが得られたものとする。そうすると、対処４の初期対処データにおける対処ステップ１はステップＳ４７の条件を満たすが、対処ステップ２はステップＳ４７の条件を満たさないことが分かる。従って、ステップＳ４７を行えば、図３０に示すように対処データが更新される。すなわち、図３０における対処ステップ２が挿入される。なお、同じ種類のコンポーネント名の対処ステップが並ぶように、対処データ内でソートされている。
【００７０】
このような処理を行うことで、エラー発生時の状態によってこれまで抽出されなかった対処ステップを、いずれの装置においても漏れなく行うように追加することができるようになる。
【００７１】
図２４の処理の説明に戻って、第２更新部１０３３は、同じソフトウエアパッケージで発生する異なるエラーについて対処データが登録済みであるか否かを判断する（ステップＳ４９）。この条件を満たす場合には、処理は端子Ａを介して図３１の処理に移行する。
【００７２】
一方、この条件を満たさない場合には、対処データ更新部１０３は、更新後の対処データ（同じソフトウエアパッケージに対して何も登録されていなかった場合には初期対処データそのもの）を、対処ＤＢ１０４に登録する（ステップＳ５１）。なお、対処データに対応する対処後の構成データについても、対処ＤＢ１０４に登録する。そして処理は終了する。
【００７３】
次に、図３１を用いて端子Ａの後の処理について説明する。第２更新部１０３３は、未処理のエラーについて登録済みの対処データを特定する（ステップＳ５３）。今回のエラーに対して複数の異なるエラーについて対処データが登録されている場合があるので、例

えば登録順に、登録済みの対処データを処理するものである。

【００７４】

そして、第２更新部１０３３は、処理対象の対処データに対応する対処後の構成データと、特定された対処データに対応する対処後の構成データとを比較して、差分を特定する（ステップＳ５５）。

【００７５】

例えば、図２８に示すような対処データが登録済みの対処データであり、対応する対処後の構成データが図２３に示すような構成データであるものとする。これに対して、対処１についての対処データ（図２５）が初期対処データとしてそのまま図３１の処理対象となった場合、対応する対処後の構成データは図１９に示すようなものであるとする。

【００７６】

そうすると、図２３と図１９を比較して、構成情報については差異がないとすると、対処１のパッケージ管理データにおいて「ｌｉｂａｂｃ」が余分に追加されているということが分かる。

【００７７】

ステップＳ５５の処理で差分が特定されなければ処理はステップＳ６１に移行する。一方、ステップＳ５５の処理で差分が特定されると、第２更新部１０３３は、特定された差分に対応する対処ステップを、対処データから削除する（ステップＳ５９）。具体的には、図２５において対処ステップ２が、差分に対応する対処ステップなので、その対処ステップが削除されて、図３２に示すような対処データに更新される。なお、構成データについては更新せずそのまま用いる。

【００７８】

そして、第２更新部１０３３は、未処理のエラーが存在するか否かを判断する（ステップＳ６１）。未処理のエラーが存在する場合には、処理はステップＳ５３に戻る。一方、未処理のエラーが存在しない場合には、端子Ｂを介して図２４の処理に戻る。

【００７９】

以上のような処理を行うことで、どのような装置に対しても共通して適用可能な対処データが生成されるようになる。

【００８０】

なお、パスワードのような具体的な値が対処ステップに含まれる場合には、そのまま対処ＤＢ１０４に登録するのは好ましくないので、図３３に示すように、「％ｓ」のように変数化している。

【００８１】

次に、検索時の処理について図３４乃至図３８を用いて説明する。なお、図３３に示すような対処データが対処ＤＢ１０４に登録されており、第２構成データ格納部１０５には、図３４及び図３５のような構成データが格納されているものとする。

【００８２】

図３４の例では、ユーザ「ｕ０７００」のサーバ「ｉ３０００」の構成情報を示しており、図３５の例では、ユーザ「ｕ０８００」のサーバ「ｉ４０００」の構成情報を示している。このように、エラー発生時の構成情報に差異があるものとする。

【００８３】

図３６を用いて処理内容を説明する。まず、検索部１０６は、ユーザ端末４００から、エラー内容（ソフトウエア名及びエラーメッセージ）及びユーザデータ（ユーザＩＤ及びサーバＩＤ）を受信する（ステップＳ１０１）。場合によっては、この段階で構成データを受信するようにしても良い。

【００８４】

そして、検索部１０６は、エラー内容で対処ＤＢ１０４を検索することで、対応する対処データを抽出する（ステップＳ１０５）。ここで、対処データが抽出できなければ（ステップＳ１０５：Ｎｏルート）、検索部１０６は、対処候補無しを、ユーザ端末４００へ送信する（ステップＳ１０７）。そして処理は終了する。

【００８５】
一方、対処データが抽出されると（ステップＳ１０５：Ｙｅｓルート）、検索部１０６は、抽出された対処データに含まれる対処ステップのうち、ユーザデータで第２構成データ格納部１０５において特定される構成データにおいて、対応する構成データが含まれている対処ステップを削除する（ステップＳ１０９）。
【００８６】
そして、検索部１０６は、残余の対処ステップを対処候補として含むデータを、ユーザ端末４００に送信する（ステップＳ１１１）。ユーザ端末４００は、このようなデータを受信し、表示装置に表示する。
【００８７】
例えば、図３７上段に示すように、ユーザ「ｕ０７００」のサーバ「ｉ３０００」について、ソフトウエア名「ａｐｐｔｏ」でエラーメッセージが「ｃｏｕｌｄ ｎｏｔ ｃｏｎｎｅｃｔ ｄａｔａｂａｓｅ」である旨の検索要求を受信すると、図３４のような構成データが読み出される。対処データは図３３に示すような対処データが抽出される。図３４の構成データからは、ｏｐｅｎｄｂに対してパスワードは設定済みであることが分かるので、図３３の対処データのうち対処ステップ１は削除される。従って、対処ステップ２及び３のみが残る。従って、図３７下段に示すように、ユーザ端末４００の表示装置には、対処ステップ２及び３が対処候補１及び２として提示されるようになる。
【００８８】
一方、図３８上段に示すように、ユーザ「ｕ０８００」のサーバ「ｉ４０００」について、ソフトウエア名「ａｐｐｔｏ」でエラーメッセージが「ｃｏｕｌｄ ｎｏｔ ｃｏｎｎｅｃｔ ｄａｔａｂａｓｅ」である旨の検索要求を受信すると、図３５のような構成データが読み出される。対処データは図３３に示すような対処データが抽出される。図３５の構成データからは、ｏｐｅｎｄｂに対してパスワードは設定済みで、ａｐｐｔｏについてｃｏｎｎｅｃｔｔｏというパラメータには設定が成されていることが分かるので、図３３の対処データのうち対処ステップ１及び２は削除される。従って、対処ステップ３のみが残る。従って、図３８下段に示すように、ユーザ端末４００の表示装置には、対処ステップ３が対処候補１として提示されるようになる。
【００８９】
以上のようにすれば、オペレータがインストールしたソフトウエアパッケージ及びその装置の構成に応じた適切な対処候補が提示されるようになる。
【００９０】
以上本発明の実施の形態を説明したが、本発明はこれに限定されるものではない。例えば、図１及び２に示した機能ブロック構成は一例であって、プログラムモジュール構成と一致しない場合もある。また、処理フローについては、処理結果が変わらない限り、処理ステップの実行順番を入れ替えたり、複数の処理ステップを並列に実行するようにしても良い。
【００９１】
さらに、サーバ３００の一部の処理については、運用管理サーバ１００で行うようにしても良い。
【００９２】
さらに、上で述べた実施の形態はソフトウエアのインストール時を想定しているが、通常運用時のエラーへの対処についても適用可能な場合もある。
【００９３】
また、運用管理サーバ１００の構成については、複数のコンピュータによって機能分担を行う場合もある。
【００９４】
なお、上で述べた運用管理サーバ１００やサーバ３００は、コンピュータ装置であって、図３９に示すように、メモリ２５０１とＣＰＵ（Central Processing Unit）２５０３とハードディスク・ドライブ（ＨＤＤ：Hard Disk Drive）２５０５と表示装置２５０９

に接続される表示制御部２５０７とリムーバブル・ディスク２５１１用のドライブ装置２５１３と入力装置２５１５とネットワークに接続するための通信制御部２５１７とがバス２５１９で接続されている。オペレーティング・システム（ＯＳ：Operating System）及び本実施例における処理を実施するためのアプリケーション・プログラムは、ＨＤＤ２５０５に格納されており、ＣＰＵ２５０３により実行される際にはＨＤＤ２５０５からメモリ２５０１に読み出される。ＣＰＵ２５０３は、アプリケーション・プログラムの処理内容に応じて表示制御部２５０７、通信制御部２５１７、ドライブ装置２５１３を制御して、所定の動作を行わせる。また、処理途中のデータについては、主としてメモリ２５０１に格納されるが、ＨＤＤ２５０５に格納されるようにしてもよい。本技術の実施例では、上で述べた処理を実施するためのアプリケーション・プログラムはコンピュータ読み取り可能なリムーバブル・ディスク２５１１に格納されて頒布され、ドライブ装置２５１３からＨＤＤ２５０５にインストールされる。インターネットなどのネットワーク及び通信制御部２５１７を経由して、ＨＤＤ２５０５にインストールされる場合もある。このようなコンピュータ装置は、上で述べたＣＰＵ２５０３、メモリ２５０１などのハードウエアとＯＳ及びアプリケーション・プログラムなどのプログラムとが有機的に協働することにより、上で述べたような各種機能を実現する。

【００９５】

以上述べた本実施の形態をまとめると、以下のようになる。

【００９６】

本実施の形態に係る情報処理方法は、（Ａ）ソフトウエアの被インストール装置から、ソフトウエアの識別子と、ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における被インストール装置の第１の構成データと、エラーの解消時における被インストール装置の第２の構成データとを取得し、（Ｂ）ソフトウエアの識別子に対応付けて格納部にソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと第１の構成データ及び第２の構成データとに基づき、ソフトウエアの識別子及びエラーの識別子に対応する対処データを更新し、（Ｃ）格納部に既存の対処データが格納されていない場合には、第１の構成データ及び第２の構成データとの差分から対処データを生成し、ソフトウエアの識別子及びエラーの識別子に対応付けて格納部に格納する処理を含む。

【００９７】

このような処理によれば、複数の装置に共通して適用可能な対処データを、構成データから蓄積してゆくことができるようになる。

【００９８】

なお、上で述べた既存の対処データを更新する処理が、（ｂ１）第１の構成データと第２の構成データとの差分から第１の対処データを生成し、（ｂ２）ソフトウエアの識別子及びエラーの識別子に対応付けられている第１の既存の対処データに含まれないが第１の対処データに含まれる対処ステップが存在する場合には、当該対処ステップを第１の既存の対処データに追加することで第１の既存の対処データを更新する処理を含むようにしても良い。

【００９９】

これによって漏れが生じないように対処ステップを追加させることができるようになる。

【０１００】

さらに、上で述べた既存の対処データを更新する処理が、（ｂ３）格納部においてソフトウエアの識別子及びエラーの識別子とは異なるエラーの識別子に対応付けられている第２の既存の対処データに係る第２の構成データと、更新後の第１の既存の対処データ又は第１の対処データに係る第２の構成データとの差を特定し、（ｂ４）上記差に対応する、更新後の第１の既存の対処データ又は第１の対処データにおける対処ステップを削除する処理をさらに含むようにしても良い。これによって必須とは言えない対処ステップを除去できるようになる。

【０１０１】
さらに、本情報処理方法は、（Ｄ）ソフトウエアの識別子とエラーの識別子と被インストール装置の識別子とを含む検索要求を受信し、（Ｅ）ソフトウエアの識別子及びエラーの識別子に対応付けられた対処データを格納部から抽出し、（Ｆ）抽出された対処データに含まれる対処ステップのうち、検索要求に係るエラー発生時における被インストール装置の第３の構成データにおいて対応するデータが含まれる対処ステップを削除し、（Ｈ）抽出された対処データを出力する処理をさらに含むようにしても良い。
【０１０２】
これによって、被インストール装置の状態に応じた対処データが、オペレータに提示され、効果的な対処が成されるようになることが期待される。
【０１０３】
なお、エラー発生時及びエラーの解消時が、ソフトウエアにより出力されるログを読み出すことで特定される場合もある。このようにすれば、正確にエラー発生時及びエラー解消時を検出できる。
【０１０４】
なお、上で述べたような処理をコンピュータに実行させるためのプログラムを作成することができ、当該プログラムは、例えばフレキシブル・ディスク、ＣＤ－ＲＯＭなどの光ディスク、光磁気ディスク、半導体メモリ（例えばＲＯＭ）、ハードディスク等のコンピュータ読み取り可能な記憶媒体又は記憶装置に格納される。なお、処理途中のデータについては、ＲＡＭ等の記憶装置に一時保管される。
【０１０５】
以上の実施例を含む実施形態に関し、さらに以下の付記を開示する。
【０１０６】
（付記１）
ソフトウエアの被インストール装置から、前記ソフトウエアの識別子と、前記ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における前記被インストール装置の第１の構成データと、前記エラーの解消時における前記被インストール装置の第２の構成データとを取得し、
前記ソフトウエアの識別子に対応付けて格納部に前記ソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと前記第１の構成データ及び前記第２の構成データとに基づき、前記ソフトウエアの識別子及び前記エラーの識別子に対応する対処データを更新し、
前記格納部に前記既存の対処データが格納されていない場合には、前記第１の構成データ及び前記第２の構成データとの差分から対処データを生成し、前記ソフトウエアの識別子及び前記エラーの識別子に対応付けて前記格納部に格納する
処理を、コンピュータに実行させるためのプログラム。
【０１０７】
（付記２）
前記既存の対処データを更新する処理が、
前記第１の構成データと前記第２の構成データとの差分から第１の対処データを生成し、
前記ソフトウエアの識別子及び前記エラーの識別子に対応付けられている第１の既存の対処データに含まれないが前記第１の対処データに含まれる対処ステップが存在する場合には、当該対処ステップを前記第１の既存の対処データに追加することで前記第１の既存の対処データを更新する
処理を含む付記１記載のプログラム。
【０１０８】
（付記３）
前記既存の対処データを更新する処理が、
前記格納部において前記ソフトウエアの識別子及び前記エラーの識別子とは異なるエラ

ーの識別子に対応付けられている第２の既存の対処データに係る第２の構成データと、更新後の前記第１の既存の対処データ又は前記第１の対処データに係る第２の構成データとの差を特定し、
前記差に対応する、更新後の前記第１の既存の対処データ又は前記第１の対処データにおける対処ステップを削除する
処理をさらに含む付記２記載のプログラム。
【０１０９】
（付記４）
ソフトウエアの識別子とエラーの識別子と被インストール装置の識別子とを含む検索要求を受信し、
前記ソフトウエアの識別子及び前記エラーの識別子に対応付けられた対処データを前記格納部から抽出し、
抽出された前記対処データに含まれる対処ステップのうち、前記検索要求に係るエラー発生時における前記被インストール装置の第３の構成データにおいて対応するデータが含まれる対処ステップを削除し、
抽出された前記対処データを出力する
処理をさらに前記コンピュータに実行させるための付記１乃至３のいずれか１つ記載のプログラム。
【０１１０】
（付記５）
前記エラー発生時及び前記エラーの解消時が、前記ソフトウエアにより出力されるログを読み出すことで特定される
付記１乃至４のいずれか１つ記載のプログラム。
【０１１１】
（付記６）
ソフトウエアの被インストール装置から、前記ソフトウエアの識別子と、前記ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における前記被インストール装置の第１の構成データと、前記エラーの解消時における前記被インストール装置の第２の構成データとを取得し、
前記ソフトウエアの識別子に対応付けて格納部に前記ソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと前記第１の構成データ及び前記第２の構成データとに基づき、前記ソフトウエアの識別子及び前記エラーの識別子に対応する対処データを更新し、
前記格納部に前記既存の対処データが格納されていない場合には、前記第１の構成データ及び前記第２の構成データとの差分から対処データを生成し、前記ソフトウエアの識別子及び前記エラーの識別子に対応付けて前記格納部に格納する
処理を、コンピュータが実行する情報処理方法。
【０１１２】
（付記７）
ソフトウエアの被インストール装置から、前記ソフトウエアの識別子と、前記ソフトウエアのインストール中に発生したエラーの識別子と、当該エラーの発生時における前記被インストール装置の第１の構成データと、前記エラーの解消時における前記被インストール装置の第２の構成データとを取得する取得部と、
前記ソフトウエアの識別子に対応付けて格納部に前記ソフトウエアのインストール時に行うべき対処ステップを含む既存の対処データが格納されている場合には、当該既存の対処データと前記第１の構成データ及び前記第２の構成データとに基づき、前記ソフトウエアの識別子及び前記エラーの識別子に対応する対処データを更新し、前記格納部に前記既存の対処データが格納されていない場合には、前記第１の構成データ及び前記第２の構成データとの差分から対処データを生成し、前記ソフトウエアの識別子及び前記エラーの識別子に対応付けて前記格納部に格納する更新部と、

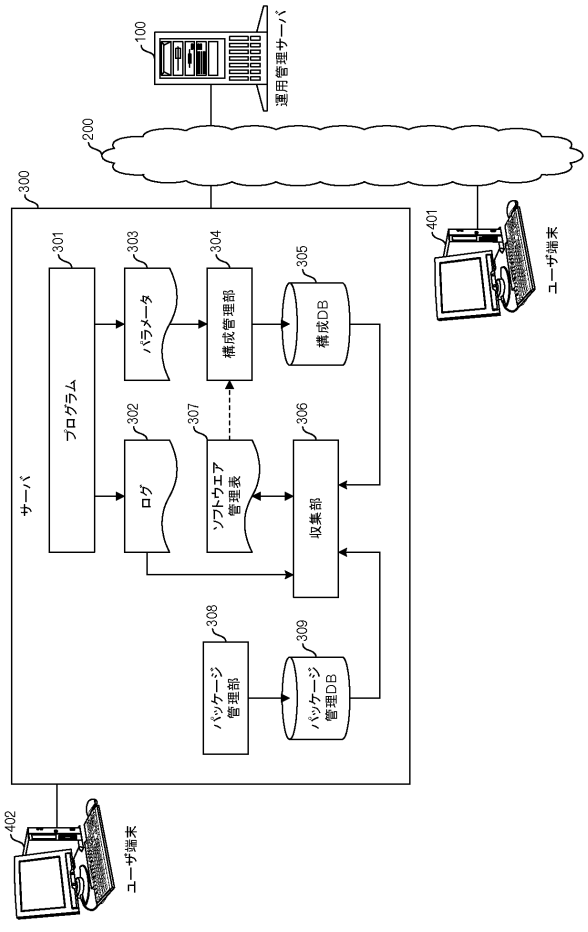
を有する情報処理装置。

【符号の説明】

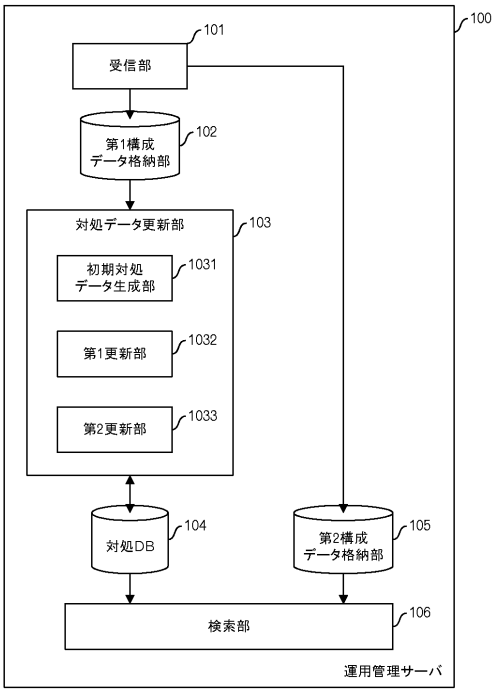
【０１１３】

１０１　受信部
１０２　第１構成データ格納部
１０３　対処データ更新部
１０４　対処ＤＢ
１０５　第２構成データ格納部
１０６　検索部

【図１】



【図２】



【図３】

パッケージ名	バージョン
libxyz	5.1
opendb	4.2
appto	1.5

【図４】

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
appto	dbpasswd	secret

【図５】

タイムスタンプ	apptoのログ	パッケージ管理の対処ステップ	構成情報の対処ステップ
2013/6/7 10:15		Install libxyz-5.1	
2013/6/7 10:16		Install opendb-4.2	
2013/6/7 10:17			opendb, passwd=secret
2013/6/7 10:19		Install appto-1.5	
2013/6/7 10:20			appto, connectto=localhost:3360
2013/6/7 10:21			appto, dbpasswd=secret
2013/6/7 10:22	[info] preparing		
2013/6/7 10:23	[info] starting		

【図６】

パッケージ名	バージョン
opendb	4.2

【図７】

タイムスタンプ	apptoのログ	パッケージ管理の対処ステップ	構成情報の対処ステップ
2013/6/7 10:15		Install appto-1.5	
2013/6/7 10:16	[info] preparing [error] missing library		
2013/6/7 10:17		Install libxyz-5.1	
2013/6/7 10:18		Install libabc-3.3	
2013/6/7 10:19	[error] could not connect to database		
2013/6/7 10:20			opendb, passwd=secret
2013/6/7 10:21			appto, connectto=localhost:3360
2013/6/7 10:22			appto, dbpasswd=secret
2013/6/7 10:23	[info] starting		

エラー

先ほどのエラーは解消、次のエラー

二番目のエラー解消

【図８】

パッケージ名	バージョン
opendb	4.2
appto	1.5
libxyz	5.1
libabc	3.3

【図９】

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
appto	dbpasswd	secret

【図１０】

パッケージ名	バージョン
opendb	4.2
libxyz	5.1

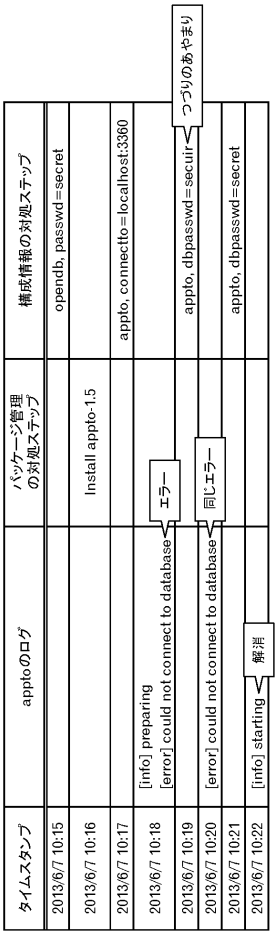
【図１２】

パッケージ名	バージョン
opendb	4.2
libxyz	5.1
appto	1.5

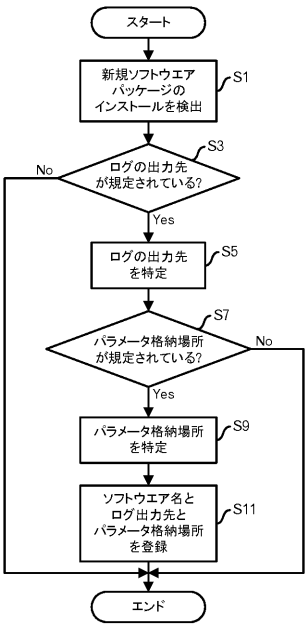
【図１３】

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
appto	dbpasswd	secret

【図１１】



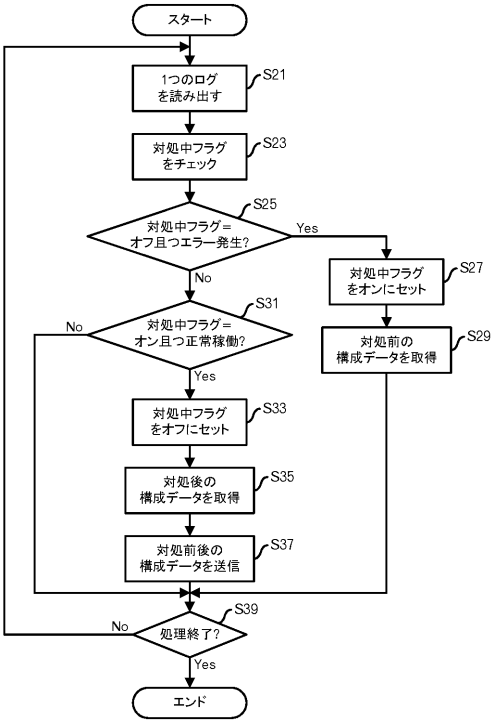
【図１４】



【図１５】

ソフトウェア名	ログ出力先	パラメータ格納場所
appto	/var/log/appto.log	/etc/appto/appto.cnf
⋮	⋮	⋮

【図１６】



【図１７】

ユーザ・サーバ	u0500, i1000
ソフトウェア名	appto
エラーメッセージ	missing library
対処前の構成データ	パッケージ管理データ パッケージ名 / バージョン 構成情報 コンポーネント名 / パラメータ / 値
対処後の構成データ	パッケージ管理データ パッケージ名 / バージョン 構成情報 コンポーネント名 / パラメータ / 値

【図１８】

パッケージ管理DB

パッケージ名	バージョン
opendb	4.2
appto	1.5

構成DB

コンポーネント名	パラメータ	値
システムパラメータなど(省略)		

【図１９】

パッケージ管理DB

パッケージ名	バージョン
opendb	4.2
libxyz	5.1
libabc	3.3
appto	1.5

構成DB

コンポーネント名	パラメータ	値
システムパラメータなど(省略)		

【図２０】

パッケージ管理DB

パッケージ名	バージョン
opendb	4.2
libxyz	5.1
libabc	3.3
appto	1.5

構成DB

コンポーネント名	パラメータ	値
システムパラメータなど(省略)		

【図２１】

パッケージ管理DB

パッケージ名	バージョン
opendb	4.2
libxyz	5.1
libabc	3.3
appto	1.5

構成DB

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
appto	dbpasswd	secret
その他のシステムパラメータなど(省略)		

【図２２】

パッケージ管理DB

パッケージ名	バージョン
opendb	4.2
libxyz	5.1
appto	1.5

構成DB

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
システムパラメータなど(省略)		

【図２３】

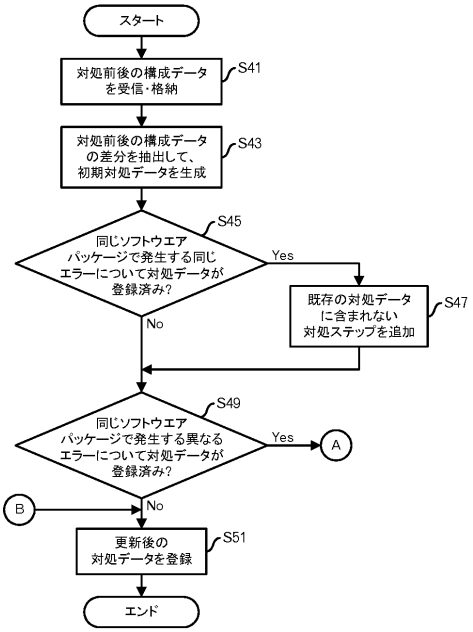
パッケージ管理DB

パッケージ名	バージョン
opendb	4.2
libxyz	5.1
appto	1.5

構成DB

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
appto	dbpasswd	secret
システムパラメータなど(省略)		

【図２４】



【図２５】

ソフトウェア名	appto
エラーメッセージ	″missing library″
対処ステップ1	Install libxyz-5.1
対処ステップ2	install libabc-3.3

【図２６】

ソフトウェア名	appto
エラーメッセージ	″could not connect to database″
対処ステップ1	opendb, passwd=secret
対処ステップ2	appto, connectto=localhost:3360
対処ステップ3	appto, dbpasswd=secret

【図２７】

ソフトウェア名	appto
エラーメッセージ	″could not connect to database″
対処ステップ1	appto, dbpasswd=secret

【図２８】

ソフトウェア名	appto
エラーメッセージ	″could not connect to database″
対処ステップ1	opendb, passwd=secret
対処ステップ2	appto, connectto=localhost:3360
対処ステップ3	appto, dbpasswd=secret

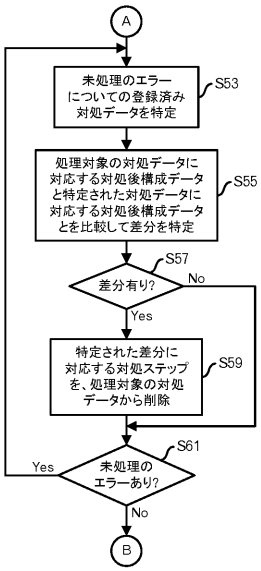
【図２９】

ソフトウェア名	appto
エラーメッセージ	″could not connect to database″
対処ステップ1	opendb, port=3360
対処ステップ2	appto, connectto=localhost:3360

【図３０】

ソフトウェア名	appto
エラーメッセージ	″could not connect to database″
対処ステップ1	opendb, passwd=secret
対処ステップ2	opendb, port=3360
対処ステップ3	appto, connectto=localhost:3360
対処ステップ4	appto, dbpasswd=secret

【図３１】



【図３２】

ソフトウェア名	appto
エラーメッセージ	″missing library″
対処ステップ1	Install libxyz-5.1

【図３３】

ソフトウェア名	appto
エラーメッセージ	″could not connect to database″
対処ステップ1	opendb, passwd=%s
対処ステップ2	appto, connectto=localhost:3360
対処ステップ3	appto, dbpasswd=%s

【図３４】

u0700,i3000の構成情報

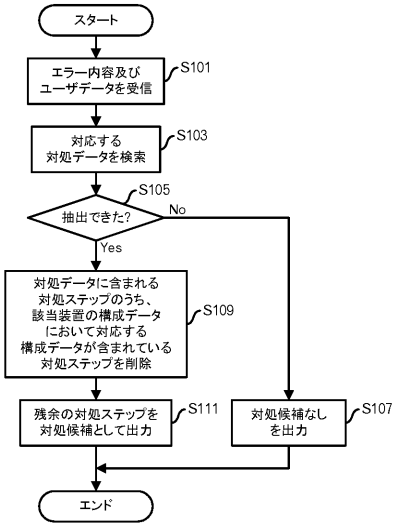
コンポーネント名	パラメータ	値
opendb	passwd	secret
システムパラメータなど(省略)		

【図３５】

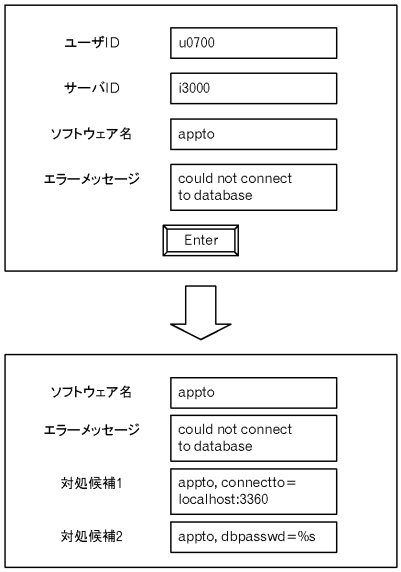
u0800,i4000の構成情報

コンポーネント名	パラメータ	値
opendb	passwd	secret
appto	connectto	localhost:3360
システムパラメータなど(省略)		

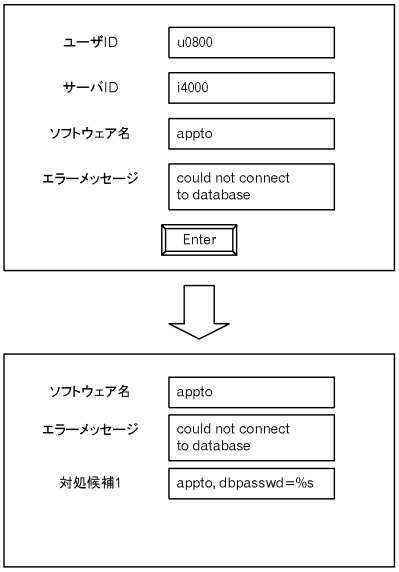
【図３６】



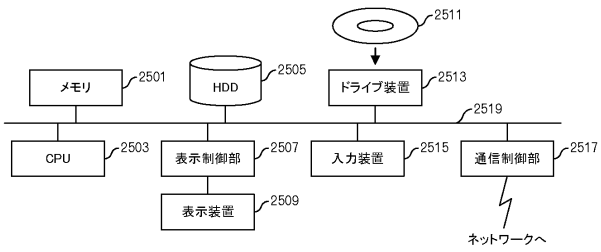
【図３７】



【図３８】



【図３９】



フロントページの続き

(56)参考文献 特開２００５－１２２３８３（ＪＰ，Ａ）
特開２００４－１８５０８４（ＪＰ，Ａ）
特開２００５－３４６３３１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４４５
Ｇ０６Ｆ　１１／０７