(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2006-268118
(P2006-268118A)
(43)公開日　平成18年10月5日(2006.10.5)

(51)Int.Cl.	F I		テーマコード（参考）
G06F 9/445 (2006.01)	G06F 9/06	610A	5B076 5B176

審査請求 有　請求項の数 11　ＯＬ　（全 13 頁）

(21)出願番号　特願2005-81719(P2005-81719)
(22)出願日　平成17年3月22日(2005.3.22)

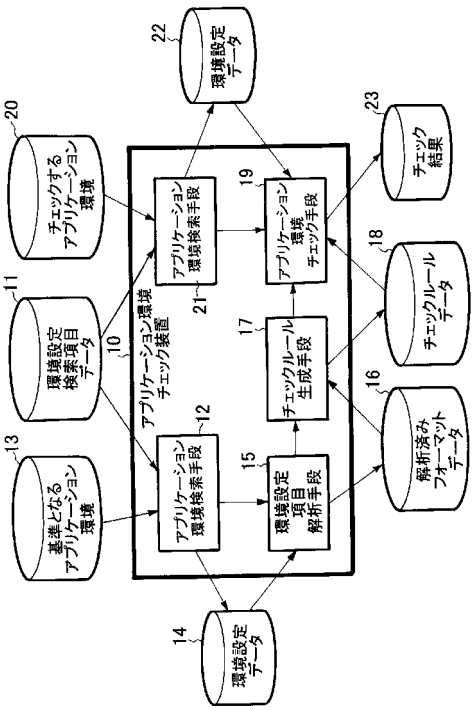
（特許庁注：以下のものは登録商標）
１．ＪＡＶＡ

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(74)代理人　100065385
弁理士　山下　穣平
(74)代理人　100122921
弁理士　志村　博
(74)代理人　100130029
弁理士　永井　道雄
(72)発明者　高根　宏之
東京都港区芝五丁目７番１号　日本電気株式会社内
(72)発明者　橋本　良太
東京都港区芝五丁目７番１号　日本電気株式会社内

最終頁に続く

(54)【発明の名称】アプリケーション環境チェック装置及び方法とそのプログラム

(57)【要約】　（修正有）
【課題】アプリケーション環境設定において、設定漏れやパラメータ設定の不整合を無くし、誤った設定値の検出を可能にする。
【解決手段】アプリケーション環境検索手段１２は、環境設定検索項目データファイル１１を読み込んで、基準となるアプリケーション環境情報１３を検索して環境設定データファイル１４を出力し、環境設定項目解析手段１５は、環境設定データファイル１４を読み込んで、解析済みフォーマットデータ１６を出力し、チェックルール生成手段１７は、解析済みフォーマットデータ１６を読み込み、チェックルールデータ１８を出力する。アプリケーション環境検索手段２１は、チェックするアプリケーション環境情報２０の環境設定データファイル２２を出力し、アプリケーション環境チェック手段１９は、環境設定データファイル２２とチェックルールデータ１８を比較して、そのチェック結果２３を出力する。
【選択図】図１



【特許請求の範囲】
【請求項１】
あらかじめ与えられた環境情報検索項目に従って基準となるアプリケーション環境情報を検索する第１のアプリケーション環境検索手段と、
前記第１のアプリケーション環境検索結果に含まれるアプリケーション環境設定情報を解析して、定型ルール化した解析済みフォーマットデータを作成する環境設定項目解析手段と、
前記解析済みフォーマットデータを読み込み、アプリケーション環境設定情報のチェックに必要なチェックルールデータを生成して出力するチェックルール生成手段とを備えることを特徴とするアプリケーション環境チェック装置。
【請求項２】
あらかじめ与えられた環境情報検索項目に従ってチェックするアプリケーション環境情報を検索する第２のアプリケーション環境検索手段と、
前記第２のアプリケーション環境検索結果に含まれるアプリケーション環境設定情報とチェックルールデータから取得した環境情報を比較して、そのチェック結果を出力するアプリケーション環境チェック手段とをさらに備えることを特徴とする請求項１に記載のアプリケーション環境チェック装置。
【請求項３】
前記アプリケーション環境情報には、環境変数、検索対象となるファイルおよびディレクトリの有無、検索対象となるファイルおよびディレクトリの属性、環境設定ファイルのパラメータテキスト文字列、ＯＳコマンド実行およびＡＰＩ実行で取得できる環境設定の文字列情報が含まれることを特徴とする請求項１又は２に記載のアプリケーション環境チェック装置。
【請求項４】
前記解析済みフォーマットデータには、各アプリケーション名に対応した、ルール分類、ルール識別子、ルール設定値が含まれ、
前記チェックルール生成手段は、解析済みフォーマットデータを読み込み、アプリケーション毎に設定された環境設定項目について、他のアプリケーションとルール分類及びルール識別子が同じかどうか比較し、
違う場合は、チェックルールを一意に区別するルール番号を各チェックルールに付けて保存し、
同じ場合は、同一チェックルールとして扱い、同じルール番号を付けて保存することを特徴とする請求項１又は２に記載のアプリケーション環境チェック装置。
【請求項５】
前記チェックルール生成手段は、ルール分類及びルール識別子が同じで、ルール設定値が異なれば、ルール不整合である旨を出力することを特徴とする請求項４に記載のアプリケーション環境チェック装置。
【請求項６】
前記アプリケーション環境チェック手段は、チェックルールおよび第２のアプリケーション環境設定情報を読み込んでそれらを比較し、
両者のルール識別子及びルール設定値が、いずれも同値であれば正常である旨の結果を出力し、いずれかが異なっていれば、不整合である旨の結果を出力することを特徴とする請求項４に記載のアプリケーション環境チェック装置。
【請求項７】
前記チェックルールには、ＯＳの環境変数が定義されていること、ＯＳの環境変数が特定の値である、あるいは特定の値が含まれていること、ファイルやディレクトリが存在すること、ファイルやディレクトリの属性が決められた値であること、環境設定ファイルのパラメータ文字列が決められた値であること、ＯＳコマンドやＡＰＩ実行の結果文字列が決められた値であること、が含まれることを特徴とする請求項１又は２に記載のアプリケーション環境チェック装置。

【請求項８】
あらかじめ与えられた環境情報検索項目に従って基準となる第１のアプリケーション環境情報を検索するステップと、
前記第１のアプリケーション環境検索結果に含まれるアプリケーション環境設定情報を解析して、定型ルール化した解析済みフォーマットデータを作成するステップと、
前記解析済みフォーマットデータを読み込み、アプリケーション環境設定情報のチェックに必要なチェックルールデータを生成して出力するステップとを含むことを特徴とするアプリケーション環境チェック方法。
【請求項９】
あらかじめ与えられた環境情報検索項目に従ってチェックする第２のアプリケーション環境情報を検索するステップと、
前記第２のアプリケーション環境検索結果に含まれるアプリケーション環境設定情報とチェックルールデータから取得した環境情報を比較して、そのチェック結果を出力するステップとをさらに含むことを特徴とする請求項８に記載のアプリケーション環境チェック方法。
【請求項１０】
あらかじめ与えられた環境情報検索項目に従って基準となる第１のアプリケーション環境情報を検索するステップと、
前記第１のアプリケーション環境検索結果に含まれるアプリケーション環境設定情報を解析して、定型ルール化した解析済みフォーマットデータを作成するステップと、
前記解析済みフォーマットデータを読み込み、アプリケーション環境設定情報のチェックに必要なチェックルールデータを生成して出力するステップとをコンピュータに実行させることを特徴とするアプリケーション環境チェック・プログラム。
【請求項１１】
あらかじめ与えられた環境情報検索項目に従ってチェックする第２のアプリケーション環境情報を検索するステップと、
前記第２のアプリケーション環境検索結果に含まれるアプリケーション環境設定情報とチェックルールデータから取得した環境情報を比較して、そのチェック結果を出力するステップとをさらにコンピュータに実行させることを特徴とする請求項１０に記載のアプリケーション環境チェック・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータのアプリケーション環境のチェックルールを生成して、他のアプリケーション環境をチェックする装置及び方法に関する。
【背景技術】
【０００２】
近年、ソフトウエアの役割分担による階層化・分離が進み、一台のハードウエアマシンに対して、ＯＳの他にシステム基盤を受け持つＲＤＢＭＳ、アプリケーションサーバ、Ｗｅｂサーバなどの複数のミドルウエアや、それらのＯＳ、ミドルウエアをシステム基盤として用いて動作するアプリケーション・フレームワークや業務アプリケーションがインストールされ、これらが１つのシステムとして協調動作するようになっている。
【０００３】
各アプリケーション・プログラムは、インストール時にどのマシンでも共通に使用する実行モジュール及び、実行モジュールが用いる各データファイルをディスク上に配置する。その他に、各マシン毎の異なったシステム環境に対応する為に外部より入力され設定されるパラメータを読み取って動作するように構成されている。
【０００４】
このようなアプリケーション環境下では、インストールに工数がかかる問題があった。さらに、ソフトウエアの階層化が進み、ＯＳとミドルウエアとフレームワーク及び業務ア

プリケーションに分化してきているが、環境設定においては相互影響する部分が拡大しているため、下記問題が顕在化してきた。

【０００５】

（１）各アプリケーションの環境設定項目が個別に存在して、複雑化しており、利用者あるいはシステム構築者が、アプリケーション・インストール説明書あるいは個別アプリケーションのインストーラにしたがって、手作業で１つ１つパラメータ設定していた。

【０００６】

（２）複数のアプリケーション間で、相互依存性が高まり、整合が取れるように、各パラメータの意味を理解して設定していた。

【０００７】

上記２つの理由から、設定項目の不整合が発生しやすくなっており、マシン環境の調査確認が必要となり、システム構築上多大な工数がかかることが問題となってきた。

【０００８】

特に大勢の開発者がいたときに、ＯＳ上でミドルウエアやフレームワークから成る同一の開発環境を多くのマシン上に構築しなければならないときなどは、個々のアプリケーション（ミドルウエアやフレームワーク）を良く知らない開発者が同じアプリケーション環境を構築する為に、各アプリケーションの説明書を土台にして、開発環境を同一化する必要がある個々の業務システム開発プロジェクトで、複数アプリケーションの整合性を指示するインストール手順書を作成する必要があるなど、アプリケーションを個別に環境設定していたのでは、多大なコストがかかるようになっている。

【０００９】

従来の技術で対応しようとすると、個別のアプリケーションごとにチェックプログラムを作成する必要があった。さらに、システムが変更されアプリケーション構成が変更される毎に、個別のチェックプログラムも変更する必要があった。

【００１０】

また、エージェントにより個人ユーザシステム内のアプリケーション・プログラムの設定情報を収集してシステム別監査情報としてプロバイダ装置に送信し、折り返し、プロバイダ装置から送出されたシステム別設定情報に含まれる最新の設定情報をアプリケーション・プログラムに設定する情報環境設定システムがある（例えば、特許文献１参照）。

【特許文献１】特開２００３－２６３３２５号公報

【発明の開示】

【発明が解決しようとする課題】

【００１１】

しかしながら、基準となるシステム環境下のアプリケーション群の設定情報を全て抜き出すことや、他のマシンでアプリケーション群が正しく設定されているか確認すること、あるいは確認する為のチェック項目を全て洗い出すことが難しかった。

【００１２】

上記のことを行うには、人手で各アプリケーションの設定項目を調べてチェックするか、あるいは個別のアプリケーションのチェックプログラムを作成する必要があった。

【００１３】

そこで本発明は、基準となるアプリケーション環境から環境設定情報を採取してチェックルールを自動的に生成し、他のアプリケーション環境の正しさを確認することができるアプリケーション環境チェック装置及び方法とそのプログラムを提供することを目的とする。

【課題を解決するための手段】

【００１４】

上述の課題を解決するため、本発明は、環境設定検索項目データファイルから読み込んだ環境情報の検索項目に従い、基準となるアプリケーション環境情報を検索して、検索結果を第１の環境設定データファイルに出力する第１のアプリケーション環境検索手段と、前記第１の環境設定データファイルから環境情報を読み込み、環境設定データを解析して

、定型ルール化した解析済みフォーマットデータを出力する環境設定項目解析手段と、前記解析済みフォーマットデータを読み込み、アプリケーション環境設定情報のチェックに必要なチェックルールデータを生成して出力するチェックルール生成手段とを備えることを特徴とする。

【００１５】

さらに、前記環境設定検索項目データファイルを読み込んで、チェックするアプリケーション環境情報を検索して、検索結果を第２の環境設定データファイルに出力する第２のアプリケーション環境検索手段と、前記第２の環境設定データファイルとチェックルールデータから取得した環境情報を比較して、そのチェック結果を出力するアプリケーション環境チェック手段とを備えることを特徴とする。

【発明の効果】

【００１６】

本発明によれば、システム上でインストールに伴うアプリケーションの環境設定で、検索項目の設定を変更するだけで、対象となったシステムに対して以下の効果が期待できる。

【００１７】

・環境設定漏れが無くなる

・パラメータ設定の不整合が無くなる

・誤った設定値の検出が可能となる（複数アプリケーション間の不整合や依存ファイル名の異同等）

【００１８】

以上から、アプリケーション環境設定やアプリケーション環境確認のコスト軽減が見込める。また、複数アプリケーション間での不整合の原因が速やかに検出可能となることで、システム構築の効率を向上することができる。

【発明を実施するための最良の形態】

【００１９】

次に、本発明の最良の形態について図面を参照して説明する。

【実施例】

【００２０】

図１は、本発明のアプリケーション環境チェック装置の機能ブロックによる構成を示す。アプリケーション環境チェック装置１０は、サーバ、ワークステーション、パーソナルコンピュータ等の情報処理装置であり、メモリに格納された制御プログラムにより以下の各実行手段として機能する。

【００２１】

アプリケーション環境チェック装置１０は、アプリケーション環境情報をシステム上から取り出して出力するアプリケーション環境検索手段１２およびアプリケーション環境検索手段２１と、アプリケーション環境検索手段１２で取り出したアプリケーション環境設定データを解析して、個別のルールを導き出す環境設定項目解析手段１５と、それらのルールを組み合わせて、アプリケーション環境設定情報のチェックに必要なルールを生成するチェックルール生成手段１７とを備える。

【００２２】

さらにチェックルール生成手段１７およびアプリケーション環境検索手段２１で得られた情報に基づき、アプリケーション環境をチェックするアプリケーション環境チェック手段１９を備える。

【００２３】

なお、チェックルールデータ１８を生成するまでの工程と、そのチェックルールに基づき、アプリケーション環境をチェックして、チェック結果２３を出力する工程とを別のコンピュータで処理することもできる。

【００２４】

またアプリケーション環境チェック装置１０の入出力ファイルとして、以下の構成を備

える。
【００２５】
アプリケーション環境検索手段１２およびアプリケーション環境検索手段２１の入力となる環境設定検索項目データファイル１１。アプリケーション環境検索手段１２の出力で、環境設定項目解析手段１５の入力となる環境設定データファイル１４。環境設定項目解析手段１５の出力で、チェックルール生成手段１７の入力となる解析済みフォーマットデータ１６。チェックルール生成手段１７の出力で、アプリケーション環境チェック手段１９の入力となるチェックルールデータ１８。
【００２６】
アプリケーション環境検索手段２１の出力となり、アプリケーション環境チェック手段１９の入力となる環境設定データファイル２２。アプリケーション環境チェック手段１９の出力となるチェック結果２３。
【００２７】
次に、図２のフローチャートを参照して、本発明のアプリケーション環境チェック方法について説明する。
【００２８】
（ステップ１０１）
まず、アプリケーション環境検索手段１２は、検索する環境情報の項目を環境設定検索項目データファイル１１から読み込む。
【００２９】
検索される環境情報の種類には、以下のものがある。
【００３０】
・環境変数
環境変数とは、ＯＳやＯＳのシェルなどに設定されているシステムの属性を記録している変数である。変数の名前と意味はあらかじめ決まっているため、環境変数を読み込めばシステムの設定がある程度分かるようになっている。
【００３１】
ＯＳの環境変数は、ＯＳ上で動作するアプリケーションソフトから、システムコールやＯＳの標準ＡＰＩ(Application Programming Interface)などを通じて簡単に値を取得できるようになっている。
【００３２】
・検索対象となるファイルおよびディレクトリの有無
・検索対象となるファイルおよびディレクトリの属性
例えばオーナー、タイムスタンプ、バージョン、パーミッションなど
・環境設定ファイルのパラメータテキスト文字列
例えば　/etc/hosts、/etc/passwd、$ORACLE_HOME/init.oraのテキスト内容など
・ＯＳコマンド実行およびＡＰＩ実行で取得できる環境設定の文字列情報
例えば　hostnameコマンド実行時の返却される文字列など
【００３３】
また、WINDOWS（米国Microsoft Corporationの登録商標）の場合のレジストリ情報や.iniファイルを例に挙げると、レジストリに登録された情報は、Win32APIで取得できるため、上記ＯＳコマンド実行およびＡＰＩ実行で取得できる環境設定の文字列情報に相当し、.iniファイルに記述されたパラメータ文字列は、環境設定ファイルのパラメータテキスト文字列に相当する。但し.iniファイルの情報取得は、Win32APIを用いることも可能である。
【００３４】
これらの検索対象となる項目を環境設定検索項目データファイル１１に記述しておく。環境設定検索項目データファイル１１には、以下の内容が記述される。
・検索対象となる環境変数の名称
・検索有無の対象のファイル・ディレクトリのパス

・属性検索の対象のファイル・ディレクトリのパス
・環境設定ファイルのパス
・ＯＳコマンドおよびＡＰＩ実行でアプリケーション環境検索手段１２が解釈するキーワード
【００３５】
(ステップ１０２）
次に読み込んだ検索項目に従い、アプリケーション環境検索手段１２は、基準となるアプリケーション環境１３の環境情報を検索する。
【００３６】
上記の環境情報検索に関しては、ＯＳあるいはアプリケーションに備わっているコマンドおよびＡＰＩを用いて取得する。例えばWINDOWSならDirコマンド、UNIX（米国American Telephone and Telegraph Companyの登録商標）ならlsコマンドでファイル・ディレクトリ属性は取得できる。また例えば、各ＯＳで異なるがenvコマンド、setenvコマンドというコマンドで、環境変数の名称を引数に指定し、環境変数は取得できる。
【００３７】
（ステップ１０３）
未検索の環境設定検索項目が残っている場合は、環境情報検索を繰り返す。
【００３８】
（ステップ１０４）
環境設定検索項目データファイル１１に記述されている検索項目の全てについて検索終了後に、検索結果を環境設定データファイル１４に出力する。
【００３９】
（ステップ１０５）
次に環境設定項目解析手段１５は、環境設定データファイル１４から環境情報を読み込む。
【００４０】
（ステップ１０６）
そして、各ＯＳやシステム環境、またＯＳコマンドやＡＰＩの各情報取得手段によって、環境設定データは異なった形式で情報が出力されている。これら情報取得手段ごとの出力形式に対応した解析処理を行い、環境情報を統一したフォーマットにする。例えば図３の形式で、アプリケーション名、ルール分類、ルール識別子、ルール設定値の項目を持って出力される。
【００４１】
ここでルール分類とは、アプリケーション設定情報の種類を意味し、例えば、ファイル有無、ファイルバージョン、環境変数などである。同一種類の情報ならば比較可能である。ルール識別子とは、アプリケーション設定情報の具体的な設定項目あるいは具体的な設定項目の名称を意味する。ルール設定値とは、そのアプリケーション設定項目で設定した値を意味する。これらはチェックルールを構成する各要素となる。
【００４２】
（ステップ１０７）
未解析の環境設定データが残っている場合は、解析処理を繰り返して環境情報を統一したフォーマットにする。
【００４３】
（ステップ１０８）
環境設定データファイル１４の全データ解析終了後に、統一されたフォーマットの環境情報を解析済みフォーマットデータ１６に出力する。
【００４４】
（ステップ１０９）
次にチェックルール生成手段１７は、解析済みフォーマットデータ１６を読み込み、アプリケーション毎に設定された環境設定項目について、他のアプリケーションとルール分

類及びルール識別子が同じかどうか比較する。
【００４５】
（ステップ１１０および１１１および１１２）
違う場合は、チェックルールを一意に区別するルール番号を各チェックルールに付けて保存する。同じ場合は、同一チェックルールとして扱い、同じルール番号を付けて保存する。ルール分類及びルール識別子が同じで、ルール設定値が異なれば、ルール不整合である旨を出力して警告する。
【００４６】
例えば図４の場合、下記２行において、ルール分類及びルール識別子が等しいが、ルール設定値が異なっている為、警告が出力される。
【００４７】
oracle　環境変数　JAVA_HOME　C:\jdk142　ルール３不整合
weblogic　環境変数　JAVA_HOME　C:\jdk141　ルール３不整合
【００４８】
（ステップ１１３および１１４）
解析済みフォーマットデータ１６のすべての環境設定項目について、繰り返し同一ルールかの判定を行い、環境設定項目全ての解析が終了した後、保存された結果はチェックルールファイル１８として出力される。
【００４９】
チェックルールには、以下のものがあるが、これらはルール分類、ルール識別子、ルール設定値によって構成されている。以下、個別に各場合の例を説明する。
【００５０】
・ＯＳの環境変数が定義されている。
例えば、TEMP変数が定義されている場合である。この場合、ルール分類は「環境変数」、ルール識別子は「TEMP」、ルール設定値は、*(ワイルドカード：文字列全般、何の値でも良い)となる。
【００５１】
・ＯＳの環境変数が特定の値である、あるいは特定の値が含まれている。
例えばパスに特定ソフトのインストールディレクトリが設定されている場合である。例として、環境変数JAVA_HOMEがC:\jdk142と設定されている場合である。この場合、ルール分類は「環境変数」、ルール識別子は「JAVA_HOME」、ルール設定値は「C:\jdk142」となる。
【００５２】
・ファイルやディレクトリが存在する。
例としてC:\oracle8\bin\oracle.exeが存在する場合とする。
この場合、ルール分類は「ファイル有無」、ルール識別子は「C:\oracle8\bin\oracle.exe」、ルール設定値は「有り」となる。
【００５３】
・ファイルやディレクトリの属性が決められた値である。
例えば、サイズが特定の値、読み書きできる、あるいは読み取り専用などの場合である。例として C:\oracle8\bin\oracle.exeのタイムスタンプが2001/01/30 11:32と指定されている場合である。この場合、ルール分類は「ファイル属性タイプスタンプ」、ルール識別子は「C:\oracle8\bin\oracle.exe」、ルール設定値は「2001/01/30 11:32」となる。
【００５４】
・環境設定ファイルのパラメータ文字列が決められた値である。
例としてC:\oracle8\bin\init.oraパラメータファイルの中にdb_block_sizeが8192と設定されている場合とする。この場合、ルール分類は「環境設定ファイルパラメータ文字列」、ルール識別子は「C:\oracle8\bin\init.ora,db_block_size」、ルール設定値は「db_block_size=8192」となる。
【００５５】

・ＯＳコマンドやＡＰＩ実行の結果文字列が決められた値である。
例としてＯＳコマンドhostnameによって取得できるホスト名がAPSERVER01の場合とする。この場合、ルール分類は「コマンドＡＰＩ出力情報」、ルール識別子は「hoatname」、ルール設定値は「APSERVER01」となる。

【００５６】

(ステップ１１５および１１６および１１７および１１８）

次にアプリケーション環境検索手段２１が環境情報の項目を環境設定検索項目データファイル１１を読み込んで環境設定データファイル２２を出力する点は、検索対象となるアプリケーション環境が基準となるアプリケーション環境１３から、チェックするアプリケーション環境２０に変わるだけでアプリケーション環境検索手段１２と同じであるから、説明は省略する。

【００５７】

（ステップ１１９）

次にアプリケーション環境チェック手段１９は、チェックルール１８および環境設定データファイル２２を読み込む。

【００５８】

（ステップ１２０）

そして、チェックルール１８とアプリケーション環境２０から取得した環境情報である環境設定データファイル２２を比較する。

【００５９】

ルール分類及びルール識別子が一致すれば、ルール設定値を比較し、同値であれば正常である旨の結果を返し、異なっていれば、不整合である旨の結果を返す。

【００６０】

例えば、チェック結果２３の例として図５の場合、チェックされるアプリケーション環境にディレクトリC:\oracle8\tempが存在しない場合、以下のように不整合となる。

【００６１】

AP名	ルール分類	ルール識別子	ルール設定値	チェック結果
oracle	ディレクトリ有無	C:\oracle8\temp	有り	不整合

他のアプリケーションについては環境設定が同じ場合で結果が正常となる。

【００６２】

また、チェックルール生成時点で、同一ルールのルール設定値が異なって不整合となっているルールは判定保留となる。

【００６３】

また、環境設定データを全てチェック完了した後も、ルールに対応した環境設定データが無い場合、環境設定がされていないと考えられるので、それらの結果を返す。

【００６４】

また、ルール分類が同じで、ルール識別子が異なるがルール設定値が同じである場合、２つの異なったルールは、実質同じものである可能性があり、それらを判別し結果を出力しておく。

【００６５】

(ステップ１２１）

未チェックの環境設定データが残っている場合は、チェック処理を繰り返す。

【００６６】

(ステップ１２２）

次にルールとチェック結果を併記したそれらの出力全てをチェック結果２３に出力する。

【００６７】

上記処理により、アプリケーションの環境情報を漏れなく確認し、環境設定が正しいことを確認することができる。

【００６８】

上記実施例の説明では、UNIXおよびWINDOWS上でのアプリケーション環境を挙げたが、環境情報の採取方法については、各ＯＳ提供のＡＰＩやファイルシステムを用いて取得するもので、特定のオペレーティングシステム環境に限定されるものではない。

【産業上の利用可能性】

【００６９】

本発明は、オペレーティングシステムを基盤として、その基盤上で動作するアプリケーション群がある環境ならば実施可能な為、例えば、オペレーティングシステムと１つないし複数のアプリケーションからなる組み込み系ソフトウエア環境でも実施できる。

【図面の簡単な説明】

【００７０】

【図１】本発明の機能ブロック構成図である。

【図２】本発明の動作を示すフローチャートである。

【図３】解析済みフォーマットデータの一例を示す図である。

【図４】生成したチェックルールの一例を示す図である。

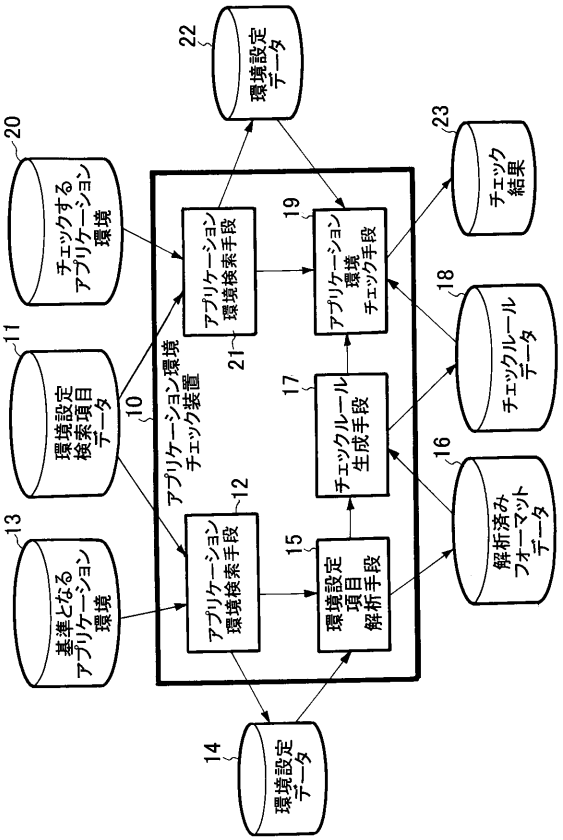
【図５】チェック結果の一例を示す図である。

【符号の説明】

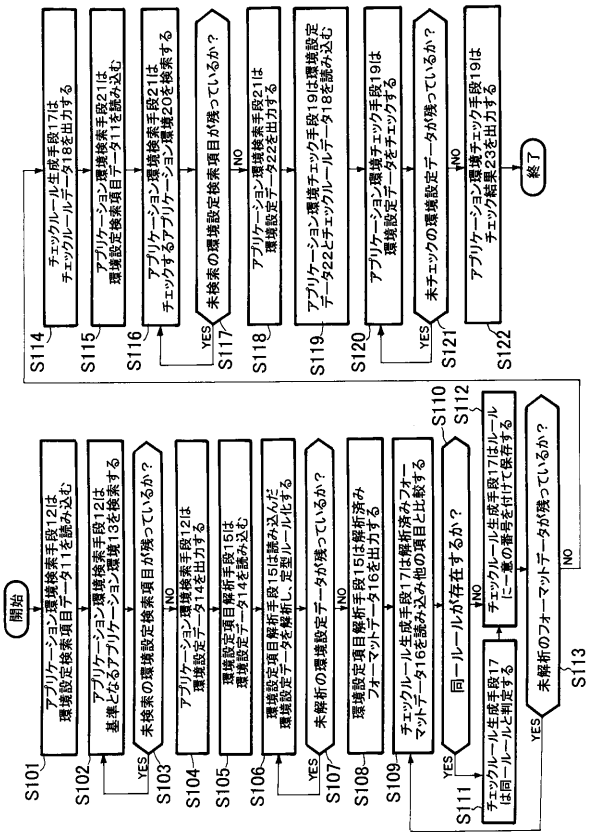
【００７１】

１０　アプリケーション環境チェック装置
１１　環境設定検索項目データファイル
１２　アプリケーション環境検索手段
１３　基準となるアプリケーション環境
１４　環境設定データファイル
１５　環境設定項目解析手段
１６　解析済みフォーマットデータ
１７　チェックルール生成手段
１８　チェックルールデータ
１９　アプリケーション環境チェック手段
２０　チェックするアプリケーション環境
２１　アプリケーション環境検索手段
２２　環境設定データファイル
２３　チェック結果

【図１】



【図２】



【図３】

解析済みフォーマットデータの例

AP名	ルール分類	ルール識別子	ルール設定値
oracle	環境変数	ORACLE_HOME	C:¥oracle8
oracle	環境変数	TEMP	C:¥oracle8¥temp
oracle	環境変数	JAVA_HOME	C:¥jdk142
oracle	ファイル有無	C:¥oracle8¥bin¥oracle.exe	有り
weblogic	ファイル有無	C:¥weblogic¥binweblogic.jar	有り
weblogic	環境変数	JAVA_HOME	C:¥jdk141
oracle	ディレクトリ有無	C:¥oracle8¥temp	有り
oracle	ファイルバージョン	oracle.exe	8.1.7
OS共通	コマンドAPI出力情報	host name	APSERVERO1
APO1	コマンドAPI出力情報	APO1 Version	3.51
Java	コマンドAPI出力情報	Java_Version	1.4.2
oracle	ファイル属性パーミッション	C:¥oracle8¥bin¥oracle.exe	rwxr-xr-x
oracle	ファイル属性タイムスタンプ	C:¥oracle8¥bin¥oracle.exe	2001/01/30/ 11:32
oracle	ファイル属性ファイル容量	C:¥oracle8¥bin¥oracle.exe	1234567
oracle	環境設定ファイルパラメータ文字列	C:¥oracle8¥init.ora, db_block_size	db_block_size=8192

【図４】

生成したチェックルールの例

AP名	ルール分類	ルール識別子	ルール設定値	ルール化状況
oracle	環境変数	ORACLE_HOME	C:¥oracle8	ルール1
oracle	環境変数	TEMP	C:¥oracle8¥temp	ルール2
oracle	環境変数	JAVA_HOME	C:¥jdk142	ルール3不整合
oracle	ファイル有無	C:¥oracle8¥bin¥oracle.exe	有り	ルール4
weblogic	ファイル有無	C:¥weblogic¥binweblogic.jar	有り	ルール5
weblogic	環境変数	JAVA_HOME	C:¥jdk141	ルール3不整合
oracle	ディレクトリ有無	C:¥oracle8¥temp	有り	ルール6
oracle	ファイルバージョン	oracle.exe	8.1.7	ルール7
OS共通	コマンドAPI出力情報	host name	APSERVERO1	ルール8
APO1	コマンドAPI出力情報	APO1 Version	3.51	ルール9
Java	コマンドAPI出力情報	Java_Version	1.4.2	ルール10
oracle	ファイル属性パーミッション	C:¥oracle8¥bin¥oracle.exe	rwxr-xr-x	ルール11
oracle	ファイル属性タイムスタンプ	C:¥oracle8¥bin¥oracle.exe	2001/01/30/ 11:32	ルール12
oracle	ファイル属性ファイル容量	C:¥oracle8¥bin¥oracle.exe	1234567	ルール13
oracle	環境設定ファイルパラメータ文字列	C:¥oracle8¥init.ora, db_block_size	db_block_size=8192	ルール14

【図５】

チェック結果の例

AP名	ルール分類	ルール識別子	ルール設定値	チェック結果
oracle	環境変数	ORACLE_HOME	C:¥oracle8	正常
oracle	環境変数	TEMP	C:¥oracle8¥temp	正常
oracle	環境変数	JAVA_HOME	C:¥jdk142	判定保留
oracle	ファイル有無	C:¥oracle8¥bin¥oracle.exe	有り	正常
weblogic	ファイル有無	C:¥weblogic¥bin¥weblogic.jar	有り	正常
weblogic	環境変数	JAVA_HOME	C:¥jdk141	判定保留
oracle	ディレクトリ有無	C:¥oracle8¥temp	有り	不整合
oracle	ファイルバージョン	oracle.exe	8.1.7	正常
OS共通	コマンドAPI出力情報	host name	APSERVERO1	正常
APO1	コマンドAPI出力情報	APO1 Version	3.51	正常
Java	コマンドAPI出力情報	Java_Version	1.4.2	正常
oracle	ファイル属性パーミッション	C:¥oracle8¥bin¥oracle.exe	rwxr-xr-x	正常
oracle	ファイル属性タイムスタンプ	C:¥oracle8¥bin¥oracle.exe	2001/01/30/ 11:32	正常
oracle	ファイル属性ファイル容量	C:¥oracle8¥bin¥oracle.exe	1234567	正常
oracle	環境設定ファイルパラメータ文字列	C:¥oracle8¥init.ora, db_block_size	db_block_size=8192	正常

フロントページの続き

Ｆターム(参考) 5B076 AA20
5B176 AA20