(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2006-520966
(P2006-520966A)
(43)公表日　平成18年9月14日(2006.9.14)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 9/50 (2006.01)	G06F 9/46 462Z	

審査請求 有　予備審査請求 未請求　（全 39 頁）

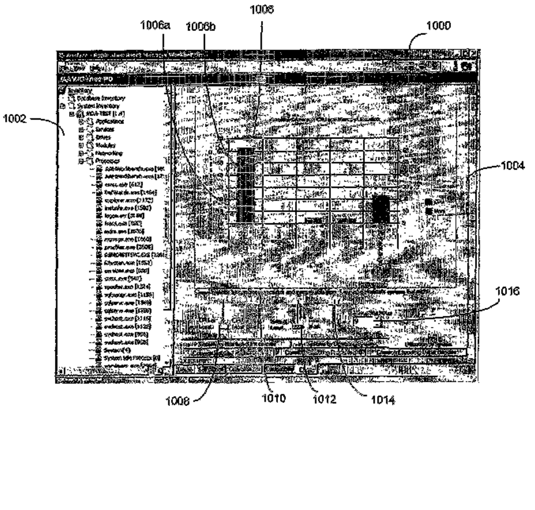
(21)出願番号	特願2006-507437(P2006-507437)
(86)(22)出願日	平成16年3月19日(2004.3.19)
(85)翻訳文提出日	平成17年11月14日(2005.11.14)
(86)国際出願番号	PCT/US2004/008667
(87)国際公開番号	WO2004/084083
(87)国際公開日	平成16年9月30日(2004.9.30)
(31)優先権主張番号	60/455,749
(32)優先日	平成15年3月19日(2003.3.19)
(33)優先権主張国	米国(US)

(71)出願人　591067923
ユニシス　コーポレイシヨン
ＵＮＩＳＹＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　ペンシルバニア州　１９４２４　ブルーベル，　ピイー．オー．ボツクス　５００　タウンシツプ　ライン　アンド　ユニオン　ミーテイング　ローズ（番地なし）
(74)代理人　100064746
弁理士　深見　久郎
(74)代理人　100085132
弁理士　森田　俊雄
(74)代理人　100083703
弁理士　仲村　義平

最終頁に続く

(54)【発明の名称】サーバの統合分析

(57)【要約】
サーバ等の第１のコンピュータデバイスの特徴を示す第１のデータセット（１００８）が、第２のコンピュータデバイスの特徴を示す第２のデータセット（１０１２）と比較され、それに対して分析される。コンピュータが読取可能な命令はデータセットに対して動作し、第２のデータセットにおける実質的に類似の特徴とは異なる第１のデータセットにおける少なくとも１つの特徴を決定することができる。出力装置（１０００）上の視覚的な表示（１００４）は、少なくとも１つの差の指示を与えて、ユーザが統合分析を容易に行なうことができるようにする。好ましくは、第１のおよび第２のデータセットは分析のためにリレーショナルデータベースにおいて記憶され、その特徴はリレーショナルデータベース上のＳＱＬクエリによって比較される。



【特許請求の範囲】
【請求項１】
　コンピュータデバイスを統合するための方法であって、
　第１のコンピュータデバイスの特徴を示す第１のデータセットを検索するステップと、
　第２のコンピュータデバイスの特徴を示す第２のデータセットを検索するステップと、
　前記第２のデータセットにおける実質的に同様の特徴とは異なる前記第１のデータセットにおける少なくとも１つの特徴を決定するステップと、
　少なくとも１つの差の視覚的な表示を与えるステップとを含む、方法。
【請求項２】
　前記第１のデータセットおよび前記第２のデータセットをリレーショナルデータベースにロードし、かつ前記リレーショナルデータベース上のＳＱＬクエリによって特徴を比較するステップを含む、請求項１に記載の方法。
【請求項３】
　コンピュータデバイスの特徴は、システムパラメータを示す情報を含む、請求項１に記載の方法。
【請求項４】
　前記システムパラメータは、プロセッサの数、利用可能なプロセッサ、プロセッサのレベル、装置、ディスクドライブの特徴、ディスクドライブの容量、システム名、ページサイズ、オペレーティングシステムのバージョン、構築されたオペレーティングシステム、およびネットワークの接続性、システムＣＰＵの利用、およびシステムメモリロードのうちの少なくとも１つを含む、請求項３に記載の方法。
【請求項５】
　コンピュータデバイスの特徴は、実行可能なプロセスパラメータを示す情報を含む、請求項１に記載の方法。
【請求項６】
　前記実行可能なプロセスパラメータは、ＣＰＵの利用、メモリの利用、アクティブなプロセス、アクティブなプロセスの従属性、プロセッサの使用、メモリの使用、プロセスの生成時間、プロセスＩＤ、プロセスの所有者、プロセスの処理、プロセスのバージョン、従属性のバージョン、プロセスのタイムスタンプ、プロセスの記述、および従属性の記述のうちの少なくとも１つを含む、請求項５に記載の方法。
【請求項７】
　コンピュータデバイスの特徴を示す情報は、コンピュータデバイスのデータベース定義パラメータを示す情報を含む、請求項１に記載の方法。
【請求項８】
　前記視覚的な表示は、少なくとも特徴に関する差のレベルを示すチャートを含む、請求項１に記載の方法。
【請求項９】
　前記視覚的な表示は、前記第１のデータセットの特徴を前記第２のデータセットの特徴と比較するテクストの表示を含む、請求項１に記載の方法。
【請求項１０】
　前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのリストを提示し、前記少なくとも１つのプロセスが前記第２のデータセットに存在するかどうかのインジケータを与える、請求項６に記載の方法。
【請求項１１】
　前記第１のデータセットにおけるプロセスのバージョンを前記第２のデータセットにおけるプロセスのバージョンと比較するインジケータをさらに含む、請求項６に記載の方法。
【請求項１２】
　前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのプロセスのリストを提示し、前記少なくとも１つのプロセスが前記第２のデータセットに存在するかど

うかのインジケータを与える、請求項６に記載の方法。

【請求項１３】

前記コンピュータデバイスのデータベース定義パラメータは、データベース名、ロール、ユーザ、エイリアス、デフォルト値、規則、機能、ユーザによって規定されたデータの種類、ユーザメッセージ、テーブル、ビュー、インデックス、拡張されたプロシージャ、記憶されたプロシージャ、およびトリガのうちの少なくとも１つを含む、請求項７に記載の方法。

【請求項１４】

前記第１のセットにおけるデータベースのログイン名を前記第２のセットにおけるデータベースのログイン名と比較するインジケータをさらに含む、請求項１３に記載の方法。

【請求項１５】

前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのテーブルのリストを提示し、少なくとも１つのテーブルが前記第２のデータセットに存在するかどうかのインジケータを与える、請求項１３に記載の方法。

【請求項１６】

前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのカラム名のリストを提示し、前記少なくとも１つのカラム名が前記第２のデータセットに存在するかどうかのインジケータを与える、請求項１３に記載の方法。

【請求項１７】

複数の第１のデータセットおよび複数の第２のデータセットを受信し、かつ時間が経つと前記第２のデータセットにおける実質的に同様の特徴とは異なる前記第１のデータセットにおける少なくとも１つの特徴を決定するステップをさらに含む、請求項１に記載の方法。

【請求項１８】

コンピュータデバイスを統合するためのシステムであって、

第１のコンピュータデバイスの特徴を示す第１のデータセットを記憶する記憶装置と、

第２のコンピュータデバイスの特徴を示す第２のデータセットを記憶する記憶装置と、

前記記憶装置に記憶され、かつ前記第２のデータセットにおける実質的に同様の特徴とは異なる前記第１のデータセットにおける少なくとも１つの特徴を決定することができるコンピュータが読取可能な命令と、

記憶装置に記憶され、かつ前記少なくとも１つの差の出力装置上で視覚的な表示を与えることができるコンピュータが読取可能な命令とを含む、システム。

【請求項１９】

前記第１のおよび第２のデータセットはリレーショナルデータベースに記憶され、少なくとも１つの特徴を決定するための前記コンピュータが読取可能な命令は、前記リレーショナルデータベース上のＳＱＬクエリによって特徴を比較する、請求項１８に記載のシステム。

【請求項２０】

コンピュータデバイスの特徴はシステムパラメータを示す情報を含む、請求項１８に記載のシステム。

【請求項２１】

前記システムパラメータは、プロセッサの数、利用可能なプロセッサ、プロセッサのレベル、装置、ディスクドライブの特徴、ディスクドライブの容量、システム名、ページサイズ、オペレーティングシステムバージョン、構築されたオペレーティングシステム、およびネットワークの接続性、システムのＣＰＵの利用、およびシステムメモリロードのうちの少なくとも１つを含む、請求項２０に記載のシステム。

【請求項２２】

コンピュータデバイスの特徴は、実行可能なプロセスパラメータを示す情報を含む、請求項１８に記載のシステム。

【請求項２３】

実行可能なプロセスパラメータは、ＣＰＵの利用、メモリの利用、アクティブなプロセス、アクティブなプロセスの従属性、プロセッサの使用、メモリの使用、プロセスの生成時間、プロセスＩＤ、プロセス所有者、プロセスの処理、プロセスのバージョン、従属性のバージョン、プロセスタイムスタンプ、プロセスの記述、および従属性の記述のうちの少なくとも１つを含む、請求項２２に記載のシステム。

【請求項２４】

コンピュータデバイスの特徴を示す情報は、コンピュータデバイスのデータベース定義パラメータを示す情報を含む、請求項１８に記載のシステム。

【請求項２５】

前記視覚的な表示は、少なくとも特徴に関する差のレベルを示すチャートを含む、請求項１８に記載のシステム。

【請求項２６】

視覚的な表示は、前記第１のデータセットの特徴を前記第２のデータセットの特徴と比較するテキストの表示を含む、請求項１８に記載のシステム。

【請求項２７】

前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのプロセスのリストを提示し、前記少なくとも１つのプロセスが前記第２のデータセットに存在するかどうかのインジケータを与える、請求項２３に記載のシステム。

【請求項２８】

前記第１のセットにおける前記プロセスのバージョンを前記第２のセットにおける前記プロセスのバージョンと比較するインジケータをさらに含む、請求項２３に記載のシステム。

【請求項２９】

前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのプロセスのリストを提示し、前記少なくとも１つのプロセスは前記第２のデータセットに存在するかどうかのインジケータを与える、請求項２３に記載のシステム。

【請求項３０】

前記第１のセットにおける前記プロセスのバージョンを前記第２のセットにおける前記プロセスのバージョンと比較するインジケータをさらに含む、請求項２３に記載のシステム。

【請求項３１】

コンピュータデバイスの特徴を示す情報は、コンピュータデバイスのデータベース定義パラメータを含む、請求項１８に記載のシステム。

【請求項３２】

前記コンピュータデバイスのデータベース定義パラメータは、データベース名、ロール、ユーザ、エイリアス、デフォルト値、規則、機能、ユーザによって規定されたデータの種類、ユーザメッセージ、テーブル、ビュー、インデックス、拡張されたプロシージャ、記憶されたプロシージャ、およびトリガのうちの少なくとも１つを含む、請求項３１に記載のシステム。

【請求項３３】

前記第１のセットにおける前記データベースログイン名を前記第２のセットにおける前記データベースログイン名と比較するインジケータをさらに含む、請求項３１に記載のシステム。

【請求項３４】

前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのテーブルのリストを提示し、前記少なくとも１つのテーブルが前記第２のデータセットに存在するかどうかのインジケータを与える、請求項３１に記載のシステム。

【請求項３５】

前記視覚的な表示は、前記第１のデータセットにおける少なくとも１つのカラム名のリストを提示し、前記少なくとも１つのカラム名が前記第２のデータセットに存在するかど

うかのインジケータを与える、請求項３１に記載のシステム。

【請求項３６】

第１のコンピュータデバイスの特徴を示す第１のデータセットに記憶される前記記憶装置は、複数の時間における前記第１のコンピュータデバイスの特徴を示す複数の第１のデータセットを含み、

第２のコンピュータデバイスの特徴を示す第２のデータセットを記憶する記憶装置は、複数の時間における前記第２のコンピュータデバイスの特徴を示す複数の第２のデータセットを含み、

前記コンピュータが読取可能な命令は記憶装置に記憶され、時間が経つと前記複数の第２のデータセットにおける実質的に同様の特徴とは異なる前記複数の第１のデータセットにおける少なくとも１つの特徴を決定することができる、請求項１８に記載のシステム。

【発明の詳細な説明】

【技術分野】

【０００１】

関連出願の相互参照

この出願は、全体が引用により援用される２００３年３月１９日に出願された米国仮出願番号第６０／４５５，７４９号、「サーバ統合のためのシステムおよびデータベースのインベントリの発見および分析」（“Discovery and Analysis of System and Database Inventories for Server Consolidation”）への優先権を主張する。

【０００２】

著作権表示／許可

この特許文献の開示の一部には、著作権保護の対象となる資料が含まれ得る。著作権所有者は、特許商標局の特許ファイルまたは特許記録に現われるように、特許文献または特許開示の何人による複製にも異議を唱えないが、それ以外ではいかなるすべての著作権を保有する。以下の表示はこの文献に適用される。著作権（Ｃ）２００４　ユニシス・コーポレイション（Unisys Corp.）

発明の分野

本発明は、計算機システムの分野に関し、より特定的にサーバの統合のためのシステムおよび方法に関する。

【背景技術】

【０００３】

発明の背景

企業において技術が一段と普及するにつれて、企業の組織は臨時の方法でサーバファームを作成してきた。たとえば、新しいアプリケーションが利用可能または必要になると、組織は新しいサーバを追加してそのアプリケーションのためのコンピューティングサポートを与えることが多い。多くの場合、サーバは、その特定のアプリケーションを実行するためだけの十分な計算能力を有する。このような臨時のサーバファームは、重複するアプリケーション、同じアプリケーションの多数のバージョン、冗長なデータ記憶および異なる計算能力という扱いにくい組合せとなる。その結果として、アプリケーションが複製され、ハードウェアの互換性がなくなる。場合によっては、企業はその計算インベントリを完全に理解さえしていないかも知れない。

【０００４】

理想的には、組織のサーバファームは、サーバおよびアプリケーションのより均質のグループであり、アプリケーションは最も効率的で効果的な方法でサーバに亘って十分にバランスをとられている。しかしながら、より典型的には、会社は計算機製品およびハードウェアの折衷的な組み合わせを有する。結果として、計算機システムが不十分になるだけではなく、職員がさまざまなハードウェアおよびソフトウェアのアプリケーションのすべてに熟達する必要があり負担がかかる。この問題に取り組むために、組織はそのアプリケーションを可用性およびスケーラビリティを向上させた、小数の大型サーバへと統合している。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

サーバの統合は、所有権の合計費用の削減、合理化されて管理可能な動作の生成、システムの信頼性の向上、容量利用の増加等を含むかなりの利益をもたらすことができる。サーバの統合は、フィジカルデバイスまたはサブシステムを追加することなく、処理および記憶容量を基準化する能力、ならびに必要に応じてリソースを区分し、割当てる柔軟性を企業に与えることができる。サーバの統合は、標準化されたコンピューティング環境をもたらし、プラットフォームの数を減らし、ソフトウェア製品およびシステムインターフェイスを統合し、動作およびシステム管理のプロシージャを集中化する。結果として職員研修が減る。

【０００６】

サーバの統合は、一般的に物理的統合または論理的統合とすることができる。物理的統合は、システムのスケーラビリティを拡張し、論理的統合は多数のアプリケーションまたはデータベースを集中型のアプリケーションまたはデータベースに移行する。さらに、物理的統合は、２つの主要なサブカテゴリ、サーバの統合および記憶の統合として考えることができる。物理的サーバの統合は、複数のサーバを利用し、そのオペレーティングシステムのインスタンスを大型のサーバの区分またはドメインに配置する。記憶装置の統合は、異なるソースからのデータを単一のリポジトリおよびフォーマットへと結合する。記憶装置は、データセンタにおける今日の最も重要な資産調達の検討材料の１つであり、その費用はサーバ費用と競合するまたはそれを超える可能性がある。記憶装置の経済的寿命は大半のサーバを超えるため、今日の記憶装置の決定は今後何年間の動作に影響を与える。

【０００７】

たとえば、所与のサーバが容量を超える場合、追加のアプリケーションをそのサーバに移動して、サーバの全体的な物理的な数を減じることができる。さらに、組織は典型的にシステムを５０％から６０％の利用で実行するように構成し、ピークの作業負荷のための余分の容量を残す。さまざまなサーバ上の、この使用されていない容量が大型のサーバファームにおけるサーバの数について考慮される場合に、無駄にされるリソースの量は莫大なものである可能性がある。サーバを統合することによって、使用されていない容量の量は、必要とされていないサーバの数と同様に大幅に低下する。

【０００８】

対象の特許文献は、サーバの統合の局面を自動化するためのさまざまな方法およびシステムについて記載している。

【課題を解決するための手段】

【０００９】

発明の概要

前述の特徴は、サーバファームにおけるサーバ等の複数のコンピュータデバイスで実行されるサービスを統合するためのシステムおよび方法によって提供される。システムおよび方法は、サーバ等の第１のコンピュータデバイスの特徴を示す第１のデータセットと、第２のコンピュータデバイスの特徴を示す第２のデータセットとを検索し、記憶することによって動作する。コンピュータが読取可能な命令は、データセットに対して動作し、第１のデータセットにおける少なくとも１つの特徴を決定することができ、第１のデータセットは、第２のデータセットにおける実質的に同様の特徴とは異なる。出力装置の視覚的な表示は、少なくとも１つの差の指示を与えて、ユーザが容易に統合の分析を行なうことができるようにする。好ましくは、第１のおよび第２のデータセットは分析のためにリレーショナルデータベースに記憶され、特徴はリレーショナルデータベース上のＳＱＬクエリによって比較される。

【００１０】

比較される特徴は、システム、実行可能なプロセス、およびデータベースのパラメータである。視覚的な表示は、特徴の差のレベルを示すチャート、または第１のデータの特徴

を第２のデータセットの特徴と比較するテキストの表示とすることができる。
【００１１】
たとえば、視覚的な表示は、第１のデータセットにおける少なくとも１つのプロセスのリストを提示することができ、少なくとも１つのプロセスが第２のデータセットに存在するかどうかのインジケータを提供する。さらに、インジケータは、第１のデータセットにおけるプロセスバージョンを第２のデータセットにおけるプロセスバージョンと比較する。さらに、視覚のインジケータは、第１のセットにおけるデータベースのログイン名を、第２のセットにおけるデータベースのログイン名と比較することができる。視覚的な表示は、第１のデータセットにおける少なくとも１つのテーブルのリストを提示し、少なくとも１つのテーブルが第２のデータセットに存在するかどうかのインジケータを提供することができる。同様のインジケータをテーブルまたは他のデータベースのパラメータに提供することができる。
【００１２】
特に、時間の経過を伴うその使用に関する単一のシステム計算機システムの比較では、２つのシステムを統合することができるかどうかを決定するために、時間の経過を伴う１つ以上の他のコンピュータデバイスの使用と比較することができる。
【００１３】
本発明に従った統合のシステムおよび方法は、添付の図面を参照して以下でさらに記載される。
【発明を実施するための最良の形態】
【００１４】
例示の実施例の詳細な説明
ここで本発明の例示の実施例の詳細な説明が図１から１９を参照してなされる。この説明は、本発明の可能な実施例の詳細な例を提供するが、これらの詳細は例示的なものであり、本発明の範囲の限界を定めることは意図されていないことに留意すべきである。
【００１５】
図１は、対象の発明の主な局面の概略図である。一般的に、統合サービス１１５は、第１のサーバファーム１１０に与えられて、ハードウェア、ソフトウェアおよびそのサーバファームにおけるデータのインベントリを作成する。その情報の局面を用いて、サーバファーム１１０を第２のサーバファーム１２０へと統合する。第２のサーバファーム１２０は、ハードウェア、ソフトウェア、データ、またはこれらのアイテムのある組合せの統合を表わし得る。統合サービス１１５は、どの特徴が第１のサーバファーム１１０に存在するかを発見し、冗長性、リソースの利用等を決定するための発見された特徴を分析する編成された方法を与え、さらに第２の統合されたサーバファームの配置を助けるためのツールを提供するプロセスを通して統合の局面を自動化する助けをする。
【００１６】
典型的なサーバファーム、たとえばサーバファーム１１０は、さまざまなサーバ１１０ａから１１０ｆを有し得る。例示のサーバファーム１１０におけるサーバ１１０ａから１１０ｆは、さまざまな製造業者、機能および電力等であり得る。さらに、示されるように、さまざまなサーバはアプリケーションおよびデータの混合を含む。たとえば、サーバ１１０ａはアプリケーションＡｐｐＡおよびＡｐｐＢを実行し、サーバ１１０ｂはアプリケーションＡｐｐＡ１を実行し、データベースデータ１を保持し、サーバ１１０ｃはアプリケーションＡｐｐＢ１を実行し、サーバ１１０ｄはアプリケーションＡｐｐＣを実行し、サーバ１１０ｅはアプリケーションＡｐｐＣ１を実行し、サーバ１１０ｆはアプリケーションＡｐｐＤを実行し、データベースデータ２を保持する。特に、さまざまなアプリケーションは、同じアプリケーションのさまざまなバージョンであり得る。たとえば、アプリケーションＡｐｐＡ１は、同じバージョンであっても異なるバージョンであっても、アプリケーションＡｐｐＡの別のインスタンスであってもよい。同様にアプリケーションＡｐｐＢ１は、アプリケーションＡｐｐＢの別のインスタンスであってもよい。さらに、データベースデータ１およびデータ２は、共通の複数のフィールドを有し得、２つのデータベ

ースを単一のデータベースへとマージすることができる。

【００１７】

上述のように、統合サービス１１５は、サーバファームをサーバファーム１２０へと統合するという主な目的のためにサーバファーム１１０に含まれるさまざまなサーバ、ハードウェア構成、アプリケーション、データベース等を発見するためのツールを提供する。

【００１８】

もちろん、機能性の一部が統合の際に故意に除去されなければ、サーバファーム１２０は、サーバファーム１１０によって以前に提供された機能性の少なくともすべてを提供する。統合されたサーバファーム１２０において、ハードウェアは、結合され、排除され、アップグレードされ得る。同様に、アプリケーションは、単一のサーバ上で実行するために統合され、排除され得、またはアップグレードされ、結合された単一のアプリケーションのさまざまなバージョン、たとえばアプリケーションＡｐｐＡおよびＡｐｐＡ１は、アプリケーションＡｐｐＡおよびＡｐｐＢへと統合され、ＡｐｐＢ１はアプリケーションＡｐｐＢへと統合されている。さらに、データベースデータ１およびデータ２はデータベースデータ１＋２へと統合されている。

【００１９】

図２は、統合管理システム１１７で動作する統合サービスの局面をさらに示している。統合システム１１７は、１つ以上のコンピュータデバイス上で実行する。コンピュータデバイスは、ネットワーク２１０を介してサーバファーム１１０に結合される。もちろん、統合システム１１７をサーバファームから分離して示しているのは例示のために過ぎない。当然のことながら、サービスは、サーバファーム内でまたはサーバファームなしでサーバまたはシステムを実行することができる。さらに、サーバファーム１１０および１２０は、分離したサーバファームとして示され、統合サービスが容易にする変形を示している。多くの例において、サーバファーム１２０は、サーバファーム１１０それ自体の更新および統合である。すなわち、サーバファームにおけるサーバの多くが再利用され、または統合されたサーバファームにおいて再配置される。

【００２０】

統合サービスの一部として実行される発見サービス２２０は、さまざまな発見サービス、たとえばアプリケーション／システムの発見、ＳＱＬサーバの発見等を含む。さまざまな発見サービスは、サーバファーム、たとえばサーバファーム１１０におけるさまざまなアセットを発見し、インベントリを作成するためにネットワーク２１０に亘って分配されるエージェントである。さまざまなサーバ、たとえば１１０ａから１１０ｆ上の発見された情報は、次に統合データベース２０６に記憶される。サーバファーム上のアセットの十分な部分が発見された後で、分析サービス２０４を用いて、サーバファームのさまざまな局面を分析することができる。最終的に、分析された情報を用いて、統合されたサーバファーム、たとえばサーバファーム１２０を管理および配置することができる。

【００２１】

主に、システムおよびアプリケーションエージェント、ならびに、ＳＱＬサービス発見エージェントの２種類のインベントリエージェントがある。他の種類のエージェントがあってもよい。たとえば、エージェントの種類は、オラクルデータベース、ＩＢＭデータベース、オブジェクト志向のデータベース等の情報を収集するように設計することができる。これらのエージェントは共にマイクロソフトウインドウズ（登録商標）の動作環境、ユニックス環境、またはリナックス（登録商標）環境におけるシステムハードウェア、アプリケーションおよびデータベースの構成に対する複数のデータポイントを捕える。システムおよびアプリケーションエージェントは、既存のアプリケーションを分析するために必要なデータポイントを検索するプロセスを補助して、統合のための適合性を決定し、統合されたアプリケーションインフラストラクチャの設計を補助する。システムおよびアプリケーションエージェントは、サーバ、アプリケーション、データベース、デバイス、プロセッサ、メモリを含む、さらにはシステムおよびアプリケーションエージェントインベントリモデル（以下で図１６Ａとの関連でさらに詳細に説明される）で規定されるようなこ

のような情報の関係を含む、クライアントの既存のサーバエステートの詳細なインベントリの獲得を容易にする。ＳＱＬサーバ発見エージェントは、ＳＱＬサーバデータベースの実現化例を分析するために必要なデータポイントを検索するためのプロセスを補助して、統合のための適合性を決定し、統合されたＳＱＬサーバインフラストラクチャの設計を補助する。データベース発見エージェントの動作は、マイクロソフトＳＱＬサーバを参照して本明細書に記載されるが、エージェントの説明および特徴は、オラクルシステムの特定の特徴に好適に適合されたオラクルデータベースシステムにも適用される。

【００２２】

ＳＱＬデータベースエージェントは、サーバ、ＳＱＬインスタンス、データベース、ユーザを含む、さらにデータベースインベントリモデル(以下で図１６Ｂとの関連でさらに詳細に説明される）で規定されたこのような情報の関係を含む、クライアントの既存のＳＱＬサーバエステートの詳細なインベントリの入手を容易にする。

【００２３】

図３は、発見プロセスを設定し開始するための例示の呼出画面を示している。ウインドウ３０２は、ユーザが発見プロセスを制御するのを可能にするためのさまざまなユーザインターフェイス機構を提供する。フォルダ部分３０４は、ユーザが、収集された発見データ、たとえばフォルダ“／ＡＡＭ／ｊｏｅ”の記憶場所を選択するのを可能にする。ターゲットボックス３０６は、選択されたターゲットサーバの名前を表示する。ボックス３０８は、選択されたフォルダにおけるファイルのリストを表示する。ツール部分３１０は、ユーザが使用する発見ツールを選択するのを可能にする。この例では、ユーザは「発見システム」を選択している。ユーザは、「発見データベース」等の代替の発見を選択するかもしれない。

【００２４】

特に、ターゲットボックス３０６は、ホスト名によってターゲットサーバを指定するための技術について示している。他の技術も可能である。たとえば、システム１１７はサーバのコンマで分離されたリストを受取ることができ、またはシステムはドメインコントローラに照会し、サーバファームにおけるＩＰアドレスのサブネットリストを獲得することができる。一般的に、サーバは、ホスト名、ホストリスト、ＴＣＰ／ＩＰサブネット、マイクロソフトアクティブディレクトリサイト名、またはドメイン名によって識別することができる。ホスト名によって、ユーザはインベントリのための単一のサーバを選択することができる。この例では、ユーザはホストマシンの名前、ユーザ名および管理者特権を有するパスワードを指定する。ホストリストによって、ユーザはインベントリのためのホストリストからサーバのグループを選択することができる。ＴＣＰ／ＩＰサブネットによって、ユーザは特定のＴＣＰ／ＩＰサブネット内のすべてのサーバを選択することができる。この例において、ユーザはサブネット内のすべてのシステムのネットワークサブネットアドレスおよびユーザ名、ならびに管理者特権を有するパスワードを入力している。サイト名によって、ユーザは特定のサイトにおけるすべてのサーバを選択することができる。この例において、ユーザはサイト内のすべてのシステムのためのサイト名およびユーザ名ならびに管理者特権を有するパスワードを入力している。ドメイン名によって、ユーザはドメインにおけるすべてのサーバを選択することができる。発見ツールのユーザは、ドメイン内のすべてのシステムのためのドメイン名およびユーザ名ならびに管理者特権を有するパスワードを入力しなければならない。サーバファーム、たとえばサーバファーム１１０におけるサーバアドレスのリストを決定した後で、システムはターゲットサーバ、たとえば１１０ａにログインして発見プロセスを呼出す。

【００２５】

一般的に、ユーザは、発見プロセスを完了するために管理者としてターゲットサーバにログインしなければならない。したがって、発見サービスは、管理者のアカウントおよびパスワードへのアクセスを有していなければならない。このアカウントおよびパスワードは、一般的にサーバファーム、たとえばサーバファーム１１０を通したサーバのすべてであるが、必ずしもそうではない。発見プロセスはそれが処理される際に各システムのアカ

ウント名およびパスワード情報を検索する。結果として、ログインプロセスは、ユーザ名およびパスワードを用いた後で発見プロセスを呼び出して、サーバファーム１１０内の複数のサーバ１１０ａ－１１０ｆの各々にログオンするように自動化することができる。発見の動作は、一般的に組織が、既存のユーザＩＤおよびパスワードを利用可能にするかまたは発見の標的となるサーバの新しいユーザＩＤおよびパスワードを生成することを要求する。ユーザＩＤは、プログラムをデバッグしてデバイスドライバをロードおよびアンロードするための権利を含む管理者特権を有するべきであり、発見のタスクが完了するとすぐにシステムから除去することができる。

【００２６】

発見ツールは、各指定されたサーバたとえば１１０ａへのリモートエージェントを開始して、そのシステム内で実行されるアプリケーションおよびプロセスのすべてについての情報を獲得する。エージェントは、獲得した情報をＸＭファイルとして統合コンピュータシステム１１７に書込み、ここでそれは統合データベース２０６に記憶される。リモートエージェントは次にターゲットサーバ、たとえば１１０ａから除去されて、それ自身のトレースを残さない。

【００２７】

発見プロセスは、一般的に、遠隔手順呼出（ＲＰＣ）、プロセス間通信（ＩＰＣ）、および名前付きパイプを使用して、１つのコンピュータデバイス（すなわち統合システム１１７をホストするコンピュータデバイス）上で実行される親プロセスをサーバコンピュータ、たとえば発見されている１１０ａと堅く結合する。ＲＰＣによってアプリケーションは遠隔から機能を呼出すことができる。したがって、ＲＰＣによってＩＰＣは、機能を呼出すのと同じくらい容易にする。ＲＰＣは単一のコンピュータまたはネットワーク上の異なるコンピュータ上のプロセスの間で動作する。

【００２８】

名前付きパイプを用いて、関連したプロセスではないプロセス間および異なるコンピュータ上のプロセス間のデータを転送する。典型的に、名前付きパイプサーバプロセスは、公知の名前またはそのクライアントに通信される名前を有する名前付きパイプを生成する。パイプの名前を知っている名前付きパイプクライアントプロセスは、その他の端部を開いて、名前付きパイプのサーバプロセスによって指定されたアクセスの制限を受けやすくすることができる。サーバおよびクライアントの双方がパイプに接続された後で、これらはパイプ上の読出および書込動作を行なうことによってデータを交換することができる。

【００２９】

発見は、システム情報およびサーバファームにある指定されたサーバ上のプロセスを実行することに関する情報を取り入れて、情報を図２のデータベース２０６に記憶するプロセスである。発見動作が各ターゲットサーバ上で終了すると、エージェントはサーバから除去され、外部のシステムからのサーバへのリンクが終了される。要約すれば、発見動作のトレースは組織のシステム内に残されるべきではない。

【００３０】

多数の発見は、特定の時間間隔における発見をスケジュールすることによって行われて、特定の時間のみに実行されるアプリケーションまたはプロセスを捕えることができる、または発見動作は手動で再び実行することができる。発見動作が繰返される毎に、サーバＸＭＬファイルの新しいバージョンが作成される。すべての修正はバージョン履歴に記憶され、そこで利用可能である。

【００３１】

アプリケーションおよびプロセス発見によって発見された情報の種類は、所与のシステム上のプロセッサの数、所与のシステム上で利用可能なプロセッサ、プロセッサのレベルおよび修正、デバイス、ディスクドライブの特徴および容量等のハードウェアの情報を含む。発見されたシステムの情報は、システム名、ページサイズ、オペレーティングシステムのバージョン、構築されたオペレーティングシステム、ネットワークの接続性等を含む。発見されたプロセスおよび依存情報は、アクティブなプロセスおよびその関連付けられ

た依存度（構成要素および構成の双方）、システムおよびプロセスレベルの双方でのプロセッサの使用、システムおよびプロセスレベル双方でのメモリの使用、プロセスの生成時間、プロセスＩＤ、プロセス所有者、プロセスの処理、プロセスおよび依存度のバージョンおよびタイムスタンプ、プロセスおよび依存度の説明を含む。

【００３２】

　ＳＱＬサーバデータベースの発見は、ＳＱＬサーバの統合を容易にするように設計される。これは情報収集および分析プロセスの多くを自動化する。これはプロセス発見を通して収集された情報を補足する。収集された情報は、顧客の既存のＳＱＬサーバエステート、すなわちサーバ、インスタンス、データベース、ユーザ等の詳細なインベントリである。収集された情報はデータベース２０６に記憶され、分析プロセスの間に統合システム１１７によって使用される。

【００３３】

　図４は、発見プロセスの局面をさらに示している。ターゲットサーバ、たとえば１１０ａは、好ましくは、全体的な発見プロセスの一部としてＧＵＩインターフェイスを通して選択される。選択された発見エージェント４０６は、特権のあるユーザアカウントを有するターゲットサーバ上でプッシュされて、クライアントマシン上でＸＭＬファイルフォーマットへの情報の収集を開始する。ＸＭＬファイルは、トラッキングバージョンを有する統合データベース２０６に記憶される。ロードプロセスの一部として、ＸＭＬファイルにおける情報が読出され、一連の関連する記録へと変形されて、クエリためにキャッシュデータベースに記憶される。

【００３４】

　統合データベース２０６を用いて、ターゲットのＳＱＬサーバから収集された情報を記憶する。データベースの種類は、好ましくはリレーショナルデータベースである。統合データベース２０６に加えておよびそれと混合しないために、ターゲットデータベース、たとえばターゲットＳＱＬサーバデータベースがある。このようなデータベースは、インベントリが取出されるインスタンスである。これらのデータベースにアクセスするために、データベース発見プロセスは、対象のＳＱＬサーバ上のＳＱＬ管理者特権アカウントを要求する。

【００３５】

　ＳＱＬサーバのインスタンスに接続するために、典型的に、ＳＱＬサーバのインスタンスが実行しているコンピュータのネットワーク名およびインスタンス名（これは特定のインスタンスが発見される場合のみに必要である）を含む2つまたは3つの情報が要求される。

【００３６】

　最初に、ログインした後で統合システム１１７は、プロシージャをターゲットサーバ、たとえば１１０ａにコピーする。特に、それはリモートサービスが実行可能なプログラム４０４をサーバコンピュータ上のａｄｍｉｎ＄のシェアにコピーする。その後で、４つの名前付きパイプ４０２が、図４に示されるようにリモートサービス４０４および統合システム１１７の間で開始される。４つの名前付きパイプ４０２、stdin, stdout, stderrおよびcontrolを用いて、統合システム１１７およびサーバ１１０ａ間の通信を容易にする。リモートサービス４０４は、名前付きパイプ４０２を用いて統合システム１１７およびサーバ１１０ａ間の接続を確立する。名前付きパイプ４０２が確立された後で、発見プロシージャ４０６、たとえば図３のツールボックス３１０から選択された発見プロシージャは、サーバ１１０ａにコピーされる。

【００３７】

　発見プロセス４０６がターゲットサーバ１１０ａの代わりであるときに、制御パイプを用いて発見プロシージャ４０６を実行する。名前付きパイプ４０２、すなわちstdin, stdout, stderrおよびcontrolは、発見プロシージャにルーティングされる。発見プロセス４０６は、以下でより完全に説明されるように適切なインベントリの収集を実施し、ターゲットサーバ１１０ａのアセットを説明するデータを含むＸＭＬファイルに送返する。その

後で発見プロセス４０６は終了し、次に好ましくはシャットダウンされ、ターゲットサーバ１１０ａから除去される。このプロセスは、サーバファーム１１０、たとえば１１０ｂ、１１０ｃ等における残っているサーバのために繰返される。

【００３８】

アプリケーションおよびシステムの発見エージェントがターゲットサーバ１１０ａ上で始動すると、さまざまなシステム呼出を用いてプロセスおよびＤＬＬ情報が収集される。ウインドウズ（登録商標）２０００のサーバオペレーティングシステム環境におけるすべてのプロセスのリストを得るために、以下の呼出が用いられる。

【００３９】

【数１】

```
ULONG (__stdcall *NtQuerySystemInformation)(
                        ULONG SystemInformationClass,
                        PVOID SystemInformation,
                        ULONG SystemInformationLength,
                        PULONG ReturnLength
                        );
```

【００４０】

NtQuerySystemInformationは、さまざまな種類のシステム情報を検索する内部のウインドウズ（登録商標）の機能である。

【００４１】

SystemInformationClassは、検索されるべきシステム情報の種類を示している。この情報は、システムにおけるプロセスの数、プロセスによって用いられる処理の数を含む各プロセスのリソースの使用についての情報、ピークのページファイルの使用、およびプロセスが割当てられるメモリページの数等を含む。

【００４２】

SystemInformationは、要求された情報が戻されるバッファを指す。この情報のサイズおよび構造は、SystemInformationClassのパラメータの値に依存して変化する。

【００４３】

SystemInformationLengthは、SystemInformationパラメータによってバイトで示されるバッファのサイズである。

【００４４】

ReturnLengthは、関数が、要求される情報の実際のサイズを書込む場所への選択的なポインタである。

【００４５】

どのＤＬＬがプロセスによってロードされるかについての情報を得るために開始アドレスを提供する別の呼出が用いられる。この呼出は以下のとおりである。

【００４６】

【数２】

```
ULONG (__stdcall *NtQueryInformationProcess)(
                        PVOID ProcessHandle,
                        INT ProcessInformationClass,
                        PVOID ProcessInformation,
                        ULONG ProcessInformationLength,
                        PULONG ReturnLength );
```

【００４７】

ProcessHandleは、情報が検索されるプロセスへの処理を指定する。

【００４８】

ProcessInformationClassは、検出されるプロセス情報の種類を指定する。このパラメ

ータは、指定されたプロセスがデバッグされているかどうかを決定するために用いることのできるＰＥＢ構造へのポインタ、および指定されたプロセスまたはプロセスがＷＯＷ６４環境（ＷＯＷ６４は、Ｗｉｎ３２ベースのアプリケーションが６４ビットのウィンドウズ（登録商標）で実行するのを可能にするｘ８６エミュレータである）で実行しているかどうかを識別するためにシステムによって用いられる一意の値のどちらも検索することができる。

【００４９】

ProcessInformationは、関数が、要求された情報を書込むアプリケーションを呼出すことによって供給されるバッファへのポインタである。

【００５０】

ProcessInformationLengthは、ProcessInformationパラメータによってバイトで示されたバッファのサイズである。

【００５１】

ReturnLengthは、機能が、要求された情報のサイズに戻る変数へのポインタである。

【００５２】

このようにして収集された情報は、ＸＭＬファイルに入れられて統合コンピュータシステム１１７に返送される。以下のＸＭＬはこのようなＸＭＬファイルの一部の例を示している。

【００５３】

【数３－１】


```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Discovery type="Process">
<PE_SysInfo ID="5008DJUL1030-SI" discoverVersion="2.0.0"
captureTimeGMT="21:10:30 30 Oct 2003" captureTimeNumeric="1067548230"
systemName="USMV-MUTSCHGO" systemMake="Dell Computer Corporation"
systemModel="Dell WORKSTATION PWS360" osMajorVersion="5"
osMinorVersion="1" osBuild="2600" osRev="Service Pack 1" pageSize="4096"
allocationGranularity="65536" totalMemory="1072689152"
availableMemory="634216448" totalVirtualMemory="2147352576"
availableVirtualMemory="2111578112" totalPageFile="2581708800"
availablePageFile="2110324736" memoryLoad="40" cpuLoad="1.7"
systemDirectory="C:\WINDOWS\System32\">
  <PE_SysInfoEx ID="EI" parent_ID="5008DJUL1030-SI" servicePackMajor="1"
servicePackMinor="0" productType="PROD_WORKSTATION"
installedPkg="PKG_SINGLEUSERTS" />
<PE_HdweInfo ID="HI" parent_ID="5008DJUL1030-SI" numberOfProcessors="2"
availableProcessorMask="3" processorLevel="15" processorRevision="521">
  <PE_ProcessorSpeed parent_ID="HI" procNum="0" speed="2992" />
  <PE_ProcessorSpeed parent_ID="HI" procNum="1" speed="2992" />
<PE_Device parent_ID="HI" deviceLocation="LPT1" cmpLocation="">
  <deviceName>Printer Port Logical Interface</deviceName>
  </PE_Device>
<PE_Device parent_ID="HI" deviceLocation="USB Device" cmpLocation="">
  <deviceName>ViewSonic Color Pocket PC V37</deviceName>
  </PE_Device>
<PE_Device parent_ID="HI" deviceLocation="" cmpLocation="0,0,0,1">
  <deviceName>HL-DT-ST RW/DVD GCC-4480B</deviceName>
  </PE_Device>
...
<PE_AppCatalogItem parent_ID="5008DJUL1030-SI" appName="PowerDVD"
appVersion="" publisher="" msiGuid="{6811CAA0-BF12-11D4-9EA1-
0050BAE317E1}">
  <installLocation />
  <installSource />
  </PE_AppCatalogItem>
<PE_AppCatalogItem parent_ID="5008DJUL1030-SI" appName="Easy CD Creator 5
Basic" appVersion="5.3.4.21" publisher="Roxio Inc" msiGuid="{609F7AC8-C510-
11D4-A788-009027ABA5D0}">
  <installLocation />
  <installSource />
  </PE_AppCatalogItem>
<PE_AppCatalogItem parent_ID="5008DJUL1030-SI" appName="Microsoft Office
2000 SR-1 Premium" appVersion="9.00.9327" publisher="Microsoft Corporation"
msiGuid="{00000409-78E1-11D2-B60F-006097C998E7}">
  <installLocation />
  <installSource>\\usmv-sms\UITSoftware\STD2000.S2A\</installSource>
  </PE_AppCatalogItem>
<PE_AppCatalogItem parent_ID="5008DJUL1030-SI" appName="Microsoft SQL
Server 2000" appVersion="8.00.761" publisher="Microsoft" msiGuid="">
  <installLocation>C:\Program Files\Microsoft SQL Server\MSSQL</installLocation>
```


(続く)

【００５４】

【数３－２】


```
  <installSource />
  </PE_AppCatalogItem>

...
<PE_Process ID="Proc.1588" parentSystem_ID="5008DJUL1030-SI"
processName="AGENTSRV.EXE" processId="1588" depth="5" affinityMask="3"
processOwner="NT AUTHORITY\SYSTEM" parentProcess_ID="Proc.772"
startTime="09:21:25 29 Oct 2003" startTimeNumeric="1067448085"
handleCount="119" basePriority="8" cpuTime="1441406250"
percentCpuTime="0.1" hasServices="true" ownProcess="false"
peakVirtualSize="187858944" pageFaultCount="127666"
peakWorkingSetSize="107339776" workingSetSize="184320"
quotaPeakPagedPoolUsage="51872" quotaPagedPoolUsage="50056"
quotaPeakNonPagedPoolUsage="20446" quotaNonPagedPoolUsage="4400"
pagefileUsage="18952192" peakPagefileUsage="107180032"
privatePageCount="18952192" version="7.0.3.0892" usedModules="Mod.0 Mod.1
Mod.2 Mod.3 Mod.4 Mod.5 Mod.6 Mod.7 Mod.8 Mod.9 Mod.10 Mod.11 Mod.12
Mod.13 Mod.14 Mod.15 Mod.16 Mod.17 Mod.18 Mod.19 Mod.20 Mod.21 Mod.22
Mod.23 Mod.24 Mod.25 Mod.26 Mod.27 Mod.28 Mod.29 Mod.30">
  <description>Agent Service Module</description>
  <fullPath>C:\Legato Connected\AGENTSRV.EXE</fullPath>
  <commandLine>"c:\Legato Connected\AgentSrv.EXE" -asv</commandLine>
  </PE_Process>
<PE_Process ID="Proc.772" parentSystem_ID="5008DJUL1030-SI"
processName="SERVICES.EXE" processId="772" depth="4" affinityMask="3"
processOwner="NT AUTHORITY\SYSTEM" parentProcess_ID="Proc.728"
startTime="09:21:21 29 Oct 2003" startTimeNumeric="1067448081"
handleCount="365" basePriority="9" cpuTime="79843750" percentCpuTime="0.0"
hasServices="true" ownProcess="false" peakVirtualSize="54595584"
pageFaultCount="4985" peakWorkingSetSize="7499776"
workingSetSize="4673536" quotaPeakPagedPoolUsage="58560"
quotaPagedPoolUsage="35612" quotaPeakNonPagedPoolUsage="14264"
quotaNonPagedPoolUsage="11040" pagefileUsage="3964928"
peakPagefileUsage="4517888" privatePageCount="3964928" version="5.1.2600.0
(xpclient.010817-1148)" usedModules="Mod.217 Mod.1 Mod.2 Mod.9 Mod.5 Mod.6
Mod.3 Mod.4 Mod.66 Mod.218 Mod.212 Mod.219 Mod.84 Mod.220 Mod.73 Mod.221
Mod.17 Mod.18 Mod.50 Mod.37 Mod.89 Mod.65 Mod.19 Mod.222 Mod.30 Mod.223
Mod.7 Mod.42">
  <description>Services and Controller app</description>
  <fullPath>C:\WINDOWS\SYSTEM32\SERVICES.EXE</fullPath>
  <commandLine>C:\WINDOWS\system32\services.exe</commandLine>
  </PE_Process>

...
<PE_Module version="1.02.0814.0000" ID="Mod.392" parent_ID="5008DJUL1030-
SI" base="1505034240" size="36864" memoryMapped="false"
creationTime="8/29/2002 2:00 AM">
  <moduleDescription>WinInet Soap Connector Library</moduleDescription>
  <path>C:\Program Files\Common Files\MSSoap\Binaries\WISC10.DLL</path>
  <imageName>WISC10.DLL</imageName>
  </PE_Module>
...
  </PE_SysInfo>
  </Discovery>
```


【００５５】

ＳＱＬサーバ発見エージェントがターゲットサーバ１１０ａを開始すると、以下の行為が行なわれる。

【００５６】
　１　エージェントはターゲットマシン１１０ａのＳＱＬサーバ名およびバージョンを入手する。
【００５７】
　２　ターゲットマシン１１０ａのＳＱＬサーバの各インスタンスにつき以下の情報が入手される。
【００５８】
　・データベーススキーマの提示が決定され、各データベースにつきテーブル、ビュー、インデックス、ロール等のスキーマ情報が収集される。
【００５９】
　・ユーザログイン、許可およびロール
　・マスタｄｂにおけるユーザオブジェクト
　・データベースの名前ならびにログインおよびデータベースのクライアント名
　・ＳＱＬ構成設定
　・照合設定、
　・ジョブおよびタスク
　・ＳＱＬ警告
　・複製
　・ＤＴＳパッケージリスト
　・データベースサイズおよびログサイズの情報
　一般的に、入手されたデータを用いて多数のサーバ上のデータベースを複製するためのデータベースオブジェクト間の差を検出する。以下のデータベースオブジェクトが比較のために入手される。
【００６０】
　ロール、ユーザ、エイリアス、デフォルト値、規則、機能、ユーザによって規定されたデータの種類、ユーザメッセージ、テーブル、ビュー、インデックス、拡張されたプロシージャ、記憶されたプロシージャおよびトリガが入手される。この情報を入手するのに利用可能な複数の方法がある。好ましい方法はＴ－ＳＱＬを用いてシステムテーブルからカタログ情報を収集する。以下の記述はマイクロソフトコーポレーション（Microsoft Corporation）から入手可能なＳＱＬサーバの実現化例を示している。それにもかかわらず、全体的な技術はオラクルデータベースシステム等の他のデータベースシステムにも適用可能である。
【００６１】
　プロシージャを記憶したＳＱＬサーバの利用可能なシステムを用いて情報を入手する。たとえば、Systemprocessおよびsysdatabaseテーブルに対する結合クエリは以下のような情報の一部を入手する。
【００６２】
【数４】

```
SELECT dbs.[name], [program_name],[loginame] FROM
[master].[dbo].[sysprocesses] procs,  [master].[dbo].[sysdatabases] dbs  Where
procs.[dbid] = dbs.[dbid] And Len([program_name]) > 0
```

【００６３】
　機能は如何なるユーザオブジェクトのためのマスタデータｄｂを問合せる。システムに記憶されたプロシージャを用いてデータを入手する。機能は、マスタデータベースにおけるユーザタイプのオブジェクトおよびその記述とともに見つけられるオブジェクトを探し、コンテンツがＸＭＬファイルに書込まれてキャッシュデータベースに記憶される。
【００６４】

【数５】

```
SELECT    CONVERT(char(32), host_name()) as MachineName,
ServerName = CASE  @@servername WHEN null THEN CONVERT(char(32),
host_name()) ELSE CONVERT(char(32), @@servername) END,  o.name as
StoredProcName,  u.name as OwnerName   FROM master..sysobjects o,
master..sysusers u   WHERE       o.uid = u.uid and  o.type = 'P' and o.category =
0 and  o.name <> 'sp_helpsql'
```

【００６５】
　１つよりも多いサーバにおける重複名および相反する許可のような潜在的なログインの問題を識別するために、この機能は利用可能な記憶されたプロシージャを介してログインおよび許可を入手する。
【００６６】
　各インスタンスにつき、ログインのリストおよびそのインスタンス内の各データベースのロールを得る。
【００６７】
　sp_configure等からの構成情報が抽出され、ＳＱＬサーバの特定のバージョンのデフォルト設定に対して比較される。
【００６８】
　ＳＱＬサーバ機能ServerPropertyを用いて、以下に示されるような製品のバージョン、エディション、サービスパック、照合等を収集する。
【００６９】
【数６】

```
Select  CONVERT(char(32), serverproperty('Collation') )as 'Collation',
CONVERT(char(32), serverproperty('Edition'))as 'Edition',
CONVERT(char(32), serverproperty('Engine Edition') )as 'Engine Edition',
CONVERT(char(32), serverproperty('InstanceName') )as 'InstanceName',
CONVERT(char(32), serverproperty('IsClustered'))as 'IsClustered',
CONVERT(char(32), serverproperty('IsFullTextInstalled'))as 'IsFullTextInstalled',
CONVERT(char(32), serverproperty('IsIntegratedSecurityOnly'))as
'IsIntegratedSecurityOnly',
CONVERT(char(32), serverproperty('IsSingleUser'))as 'IsSingleUser',
CONVERT(char(32), serverproperty('IsSyncWithBackup'))as 'IsSyncWithBackup',
CONVERT(char(32), serverproperty('LicenseType'))as 'LicenseType',
CONVERT(char(32), serverproperty('MachineName'))as 'MachineName',
CONVERT(char(32), serverproperty('NumLicenses'))as 'NumLicenses',
CONVERT(char(32), serverproperty('ProcessID'))as 'ProcessID',
CONVERT(char(32), serverproperty('ProductVersion'))as 'ProductVersion',
CONVERT(char(32), serverproperty('ProductLevel'))as  'ProductLevel',
CONVERT(char(32), serverproperty('ServerName'))as 'ServerName'
```

For non-2000 SQL Server some of these fields will be null.

【００７０】
　非２０００ＳＱＬサーバについて、これらのフィールドの一部はナルである。
【００７１】
　以下の機能は、ｍｓｄｂのsysjobsテーブルを介してジョブのリストを、sysAlertテーブルを介して警告のリストを、およびインスタンスのsysOperatorsを介してオペレータのリストを入手する。
【００７２】
　ジョブ：
【００７３】

【数７】

```
Select CONVERT(char(32), host_name()) as MachineName, ServerName = CASE
@@servername  WHEN null THEN CONVERT(char(32), host_name()) ELSE
CONVERT(char(32), @@servername)  END,* from msdb..sysjobs
```

【００７４】
警告：
【００７５】
【数８】

```
SELECT CONVERT(char(32), host_name()) as MachineName, ServerName = CASE
@@servername WHEN null THEN CONVERT(char(32), host_name()) ELSE
CONVERT(char(32), @@servername) END,[id],[Name],Event_source,
Event_category_id, Event_id, Message_id, Severity,
Enabled,Delay_between_responses, Last_occurrence_date, Last_occurrence_time,
Last_response_date, Last_response_time, Notification_message,
Include_event_description, Database_name, Event_description_keyword,
Occurrence_count, Count_reset_date, Count_reset_time, Job_id, Has_notification,
Flags, Performance_condition, Category_id, '' as Event_category_name, '' as
Delay_between_notifications, '' as Task_id, '' as Has_email_notification, '' as
Has_pager_notification FROM msdb..sysalerts
```

【００７６】
オペレータ：
【００７７】
【数９】

```
DECLARE @SQLVersion varchar(4)
SELECT @SQLVersion = SUBSTRING(@@version, 23, 4)
--Extract the information, dependant on SQL version
IF (@SQLVersion = '6.50')
        SELECT CONVERT(char(32), host_name()) as MachineName, ServerName =
CASE @@servername WHEN null THEN CONVERT(char(32), host_name()) ELSE
CONVERT(char(32), @@servername) END,*, '' as Netsend_address, '' as
Last_netsend_date, '' as Last_netsend_time, '' as Category_id FROM
msdb..sysoperators
ELSE
        IF (@SQLVersion = '7.00') or (@SQLVersion = '2000')
                SELECT CONVERT(char(32), host_name()) as MachineName,
ServerName = CASE @@servername WHEN null THEN CONVERT(char(32),
host_name()) ELSE CONVERT(char(32), @@servername) END,* FROM
msdb..sysoperators
```

【００７８】
複製が可能な場合には、情報がデータベース上で収集され、複製のロール（発行者、配布者、加入者）および複製の種類とともにリスト、サーバ、インスタンスおよびｄｂ名において報告される。システムの記憶プロシージャ‘sp_helpreplicationdboption’を用いて複製情報を入手する。ＤＴＳパッケージ情報を入手するために、以下のＳＱＬステートメントが実行される。
【００７９】

【数１０】

```
DECLARE @SQLVersion varchar(4)
DECLARE @SQLString varchar(255)
SELECT @sqlversion = SUBSTRING(@@version, 23, 4)
IF (@SQLVersion = '6.50')
        select ''
ELSE
        IF (@SQLVersion = '7.00')
                IF @@ServerName is not Null
                        SELECT @SQLString = 'SELECT CONVERT(char(32),
host_name()) as MachineName, CONVERT(char(32), @@servername) as
ServerName,name,id,versionid,cast(description AS char(25)) as ShortDescription,
categoryid,createdate,owner, owner_sid, "" as PackageType from
msdb..sysdtspackages'
                ELSE
                        SELECT @SQLString = 'SELECT CONVERT(char(32),
host_name()) as MachineName, CONVERT(char(32), host_name()) as
ServerName,name,id,versionid,cast(description AS char(25)) as ShortDescription,
categoryid,createdate,owner, owner_sid, "" as PackageType from
msdb..sysdtspackages'
        ELSE
                IF @SQLVersion = '2000'
                        IF @@ServerName is not Null
                                SELECT @SQLString = 'SELECT CONVERT(char(32),
host_name()) as MachineName, CONVERT(char(32), @@servername) as
ServerName,name,id,versionid,cast(description AS char(25)) as ShortDescription,
categoryid,createdate,owner, owner_sid,packagetype from msdb..sysdtspackages'
                        ElSE
                                SELECT @SQLString = 'SELECT CONVERT(char(32),
host_name()) as MachineName, CONVERT(char(32), host_name()) as
ServerName,name,id,versionid,cast(description AS char(25)) as ShortDescription,
categoryid,createdate,owner, owner_sid,packagetype from msdb..sysdtspackages'
EXEC(@SQLString)
```

【００８０】

各データベースのデータベースサイズおよびログサイズを得るために、（使用されかつ自由な）ｄｂサイズおよび（使用されかつ自由な）ログサイズを用いて、server/instance/dbnameで報告する。以下は、各データベースに行き、情報の一部を得るために記憶されたプロシージャ‘sp_spaceused’を実行するサンプルコードである。

【００８１】

【数１１】

```
DECLARE AllDatabases CURSOR FOR SELECT name FROM sysdatabases --WHERE
dbid > 4
OPEN AllDatabases
DECLARE @DBNameVar VARCHAR(128)
DECLARE @Statement VARCHAR(255)
FETCH NEXT FROM AllDatabases INTO @DBNameVar
WHILE (@@FETCH_STATUS = 0)
BEGIN
  SELECT @Statement = 'USE ' + @DBNameVar + CHAR(13)
    + ' exec sp_spaceused'
  EXEC (@Statement)
  FETCH NEXT FROM AllDatabases INTO @DBNameVar
END
CLOSE AllDatabases
DEALLOCATE AllDatabases
```

【００８２】

ログサイズ情報を入手するために、以下のＳＱＬステートメントが使用される。

【００８３】

【数１２】

```
DBCC SQLPERF(LOGSPACE) WITH NO_INFOMSGS
```

【００８４】

入手したデータベースの情報は、ＸＭＬファイルにフォーマットされ、統合システム１１７に返信される。このようなおよびＸＭＬファイルの例示の一部は以下のとおりである。

【００８５】

【数１３】

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Discovery type="Database">
<DD_Server machineName="USMV-VAZEHGMM1" windowsVersion="5.1.2600
Service Pack 1 Build 2600" discoverVersion="2.0.0" processorCount="1"
processorActiveMask="" OS_Name="Windows_NT 5" systemName="USMV-
VAZEHGMM1" systemManufacturer="Dell Computer Corporation" systemModel="Dell
OPTIPLEX GX260" systemType="x86" processor="x86 Family 15 Model 2 Stepping 4
GenuineIntel ~ 2000 MHz" BIOSVersion="DELL - 6" locale="United States"
timeZone="Pacific Standard Time" windowsDirectory="C:\WINDOWS"
bootDevice="\Device\HarddiskVolume3" systemDirectory="C:\WINDOWS\System32"
physicalMemory="1046524.00" availablePhysicalMemory="102700.00"
virtualMemory="2097024.00" availableVirtualMemory="2040440.00"
pagefileSpace="0.00">
 ...

<DD_Database serverName="USMV-VAZEHGMM1\DESKTOPSERVER"
dbName="Analysis" owner="sa" created="Sep 3 2003" status="Status=ONLINE,
Updateability=READ_WRITE, UserAccess=MULTI_USER, Recovery=SIMPLE,
Version=539, Collation=SQL_Latin1_General_CP1_CI_AS, SQLSortOrder=52,
IsAutoClose, IsAutoShrink, IsTornPageDetectionEnabled, IsAutoCreateStatistics,
IsAutoUpdateStatistics" compatibility_level="80" logSize="0.00"
logspaceUsed="0.00" lStatus="" dbSize="24.06" unalloc_s="1.91"
reserv_s="21664.00" data_s="15552.00" index_s="5736.00" unused_sp="376.00"
transPublish="0" mergePublish="0" dbOwner="True" readOnly="False">

<DD_SchemaInfo>
<DD_Table serverName="USMV-VAZEHGMM1\DESKTOPSERVER"
dbName="Analysis" tableName="DD_Alert">
 <DD_Column serverName="USMV-VAZEHGMM1\DESKTOPSERVER"
dbName="Analysis" tableName="DD_Alert" columnName="fileVersion" colid="1"
coltype="nvarchar" collen="510" colprec="255" colscale="0" isnullable="0"
collation="SQL_Latin1_General_CP1_CI_AS" />
 </DD_Table>
 <DD_User serverName="USMV-VAZEHGMM1\DESKTOPSERVER"
dbName="Analysis" loginName="" groupName="" userName="guest" />
 </DD_SchemaInfo>
 </DD_Database>

 </DD_Instance>
 </DD_Server>
 </Discovery>
```

【００８６】

ここに、スキーマ情報の部分のみのより詳細なＸＭＬのレイアウトがある。

【００８７】

ＳＱＬインスタンス内の各データベースにつき、情報を含む<schemaInfo>と呼ばれる要素がある。

【００８８】

【数１４】

```
<SchemaInfo
  <TableInfo
   <ColumnInfo  name = "columnName goes here"
                    Description = "column description goes here"  />
   <ColumnInfo  name = "columnName goes here"
                    Description = "column description goes here"  />
   <ColumnInfo  name = "columnName goes here"
                    Description = "column description goes here"  />

     .......
     ....... more columns
   <TriggerInfo  name = "triggerName "  Description = " trigger description" />
     .......
     .......additional triggers

   <ConstraintInfo  name = "constraintName "
                         Description = "constraint description" />

     .......
     .......additional constraints

   <IndexInfo  name = "indexName "  Description = "index description" />
     .......
     .......additional indexes

  </TableInfo>
  ..........
  ..........additional  tables go here

  <ViewInfo  name = "viewName goes here"
               Description = "view description goes here"  </ViewInfo>
  .......
  ........ more views

  <UdtInfo  name = "UDTName goes here"
              Description = "UDT description "  </UdtInfo>
   .......

  ........ more user-defined types

  <FunctionInfo  name = "functionName goes here"
                   Description = "function description goes here"   </FunctionInfo>
  ........
  .........  more user-defined functions

  <SPInfo  name = "stored-procedureName goes here"
            Description = "Stored-procedure description goes here"   </SPInfo>
   ........
  .........  more user stored-defined procs

  <DefaultsInfo  name = "defaultName goes here"
                  Description = "default description goes here"   </DefaultInfo>
   ........
  .........  more defaults in here

  <RuleInfo  name = "ruleName goes here"
               Description = "rule description goes here"  </RuleInfo>
   ........
   .........  more rules in here

  <UserInfo  name = "userName goes here"
              Description = "user description goes here"   </UserInfo>
   ........
   .........  more user info in here

  <UserMsgInfo  name = "userMsgName goes here"
                  Description = "userMsg description goes here"   </UserMsgInfo>
   ........
   .........  more user messages info in here

</SchemaInfo>
```

【００８９】

特定のサーバの情報が発見された後で、プロセスは、サーバファーム、たとえば１１０

における関心のあるサーバのすべてが発見されるまで、別のサーバ、たとえば１１０ｂのために繰返される。十分な数のサーバが発見された後で、それよりも、かなりの数のサーバが発見された後で、分析ツールを用いて統合プロセスの局面を補助することができる。

【００９０】

分析ツールは発見プロセスの間に得られる情報からの報告を解釈し生成する。各ファイルの修正を含む発見ファイルのいずれも開くことができる。したがって、分析プロセスは、発見されたサーバアセットの如何なるサブセットにも焦点を当てるように調整することができる。発見ファイルの組が一旦開かれると、分析ツールは分析されているシステムおよびプロセスの数を要約する。

【００９１】

本明細書において以下で、アプリケーション、データベース等が１つ以上の他のターゲットサーバへ移動するサーバ統合の文脈において分析が説明されるが、分析の局面および実際にここに記載されるツールの多くは単一のサーバに適用される。すなわち、サーバの局面は、異なる時点でそれ自体と比較することができる。したがって、上述の発見されたＸＭＬファイルは、サーバによって時間とともに維持されることに注目することが重要である。これによって２つの形態の時間をベースにした分析が可能になる。一方の場合に、使用中のプロセスおよびサーバのためにロードするシステムを、時間とともに変化する中で調査することができる。他方の場合に、サーバは統合のアクティビティが生じた後でそれ自身と比較することができる。これによって統合を後退させることができる。たとえば、アプリケーションおよびその従属性がソースサーバから統合ターゲットサーバに移動され、かつアプリケーションおよびその従属性の一部またはすべてが後でソースサーバから除去された場合に、ここに記載された分析ツールによって特徴のすべてを、サーバのインベントリの１つのバージョンを同じサーバのインベントリの異なるバージョンと比較する際に、適用することができる。このようにして、ユーザは初期のシステムの状態に戻ることができる。同様に、システムを用いてどのインベントリが特定のサーバに追加され、どのバージョンで追加が行なわれたのかを追跡することができる。このようにして、分析ツールによってユーザは、利用基準を超えさせたかもしれないサーバにどのアプリケーションが追加されたかを迅速に識別することができる。重要な点は、ここに記載されたツールは、統合の目的のためにソースサーバをターゲットサーバと比較するというコンテクスト以外のコンテクストにも適用されるということである。

【００９２】

アプリケーションの統合およびアプリケーションの共存のための機会を強調表示する報告を生成することができる。たとえば、共通プロセスレポートは、サーバファーム内の２つ以上のシステムで実行されるプロセスをリストする。共通のプロセスに関連付けられたアプリケーションは統合の候補である。分析ツールは、如何なる態様においても如何なる記憶された属性に記憶されるカスタムリポート出力を与える。

【００９３】

レポートは以下のデータ要素のいずれかのクエリーに基づいて生成することができる。

【００９４】

・ハードウェア情報
・所与のシステムのプロセッサの数
・所与のシステムの利用可能なプロセッサ
・プロセッサレベルおよび修正
・ＰＣＩバスのデバイス
・システム上の非ネットワークディスクドライブおよびドライブの特徴
・システム情報
・システム名
・オペレーティングシステムの修正
・構築されたオペレーティングシステム
・全体的なおよび利用可能なメモリ

・アプリケーション
・アプリケーション名
・アプリケーションバージョン
・プロセス
・プロセス名およびプロセスＩＤ
・プロセス所有者
・プロセスの従属性
・プロセスおよび従属性の記述
・プロセスおよび従属性のバージョンおよびタイムスタンプ
・実際のメモリおよび仮想のメモリ
・メモリページング
・プロセッサの使用
・実際のＣＰＵ時間
・プロセスで開かれた処理の数

図５は、統合のために収集されたデータの分析に関わる一般的なプロセスのフロー図を示している。この図はアプリケーションの統合の例を用いている。しかしながら、データ統合のための非常に類似したプロセスが生じる。明らかに、所与のサーバ上のアプリケーションおよびデータのすべてが他のサーバに統合される場合に、そのサーバは全体的にサーバからの除去の候補であり、物理的な統合をもたらす。

【００９５】

最初に、関心のある単数のまたは複数のサーバのデータが発見されたかどうかの判定を行なう（ステップ５０２）。潜在的な統合の候補のサーバを決定するために最初の高レベルの分析が行なわれる（ステップ５０４、５０６）。このプロセスは、分析ユーザインターフェイスの図と関連して以下でより完全に説明される。ステップ５０８で、統合の潜在的な利益に関する判定が行なわれる。潜在的な利益がある場合に、統合に必要なデータのすべてが収集される（ステップ５１０）。これは既に行なわれているかも知れず、そうであればこのステップをスキップすることができる。しかしながら、アプリケーションおよびその従属モジュールのすべてまたはデータベースならびにそのテーブルおよびカラムのすべて等の統合のために必要な詳細な情報のすべてが利用可能でなければならない（ステップ５１２）。その後で分析を行なって候補のサーバ上で共通の構成要素、たとえば候補のサーバ間で共通のアプリケーションおよびモジュールの数を判定する。次に、潜在的な統合のグループ化のリストが作成され、たとえばＥメールのアプリケーションは１つのマシン上で共にグループ化することができる（ステップＳ５１４，５１６）。

【００９６】

候補のアプリケーションおよび／またはデータベースが識別された後で、ばらつきがあるかどうか従属性が比較され、たとえば１つの候補サーバ上にＤＬＬがあり、他のサーバ上にＤＬＬと同じバージョンがある（ステップ５１８、５２０）。アプリケーションおよび／またはデータベースが統合された後で、統合されたサーバの性能値が測定されて、追加されたタスクを行なうための容量を有していることを保証する（ステップ５２２，５２５）。その後で、プロセス全体を繰返し、さらなる統合が有益なものかどうかを判定するために、統合されたサーバファームのための新しい情報を発見することができる。

【００９７】

図６は、統合分析で用いるための例示のユーザインターフェイス（ＵＩ）の例を示している。ウインドウ６００は、ユーザが関心のあるサーバファーム、たとえば１１０におけるサーバから収集した発見情報のさまざまなファイルを閲覧するためのインターフェイスを提供する。この目的のために、ウインドウ６００はフォルダに配置されたサーバ情報の階層的に配置されたサーバ情報のカタログを有するペイン６０２を有する。フォルダのうちの１つを選択することによって、ペイン６０２に表示され、ユーザはさまざまなサーバから収集された（上述した）ＸＭＬファイルのカタログを有するペイン６０４に提示される。特に、ＸＭＬファイルの各々はタイムスタンプ６０６とバージョン番号６０８とを含

む。これによって、情報を、異なる時間に同じサーバで発見し、サーバの変化を監視することができる。

【００９８】

図７は、ユーザが発見されたサーバのインベントリのすべてを見るのを可能にすることによって、サーバの統合の分析を支援するＵＩの一部の例を示している。ウインドウ７００は、２つのペイン７０２および７０３に分割される。ペイン７０２は、サーバのための発見された情報の階層的なビューを提供している。ここでたとえば、ユーザはサーバＯＴＧ－ＳＹＳ－３のシステムインベントリの階層的なビューを開いて、特にアプリケーションおよびアドービアクロバット５．０（７０４）を選択している。そのアプリケーションの属性７０６および対応する値７０８はペイン７０３に表示される。

【００９９】

図８は、選択されたサーバのアプリケーションの共通性の図形を提示することによってサーバ統合の分析を支援するＵＩの一部の例を示している。ウインドウ８００は、３つの円グラフ８０２、８０４および８０６の図を示している。円グラフ８０２は、異なる色またはシェーディングで現れる異なるおよび同じバージョンを有するアプリケーションとともに、１つよりも多いサーバに現れるアプリケーションを図示している。ここでたとえば、円グラフ８０２は、選択されたサーバ上でアプリケーションの非常に高い共通性があることを示し、統合によって利益を獲得し得ることを暗示している。同様に、円グラフ８０６はプロセスの共通性の量を示し、この例における高い共通性を示している。円グラフ８０４は、関心のあるサーバにおけるプロセスの従属性の共通性を図示している。共通性の詳細は、図９に詳細に示されるようにより詳細に見ることができる。

【０１００】

図９は、プロセスの共通性に関してさらなる詳細を与えるＵＩの例示の部分を示している。ウインドウ９００は、２つのペイン９０２および９０４に分割される。ペイン９０２は、統合分析を行なうためにサーバファームにおけるサーバのリスト、たとえばサーバファーム１１０を示している。ペイン９０４は、プロセス名９０６によってプロセスのリストを提供する。ペイン９０４は、発見情報の修正９１０とともに、プロセス９０８がどのサーバ上にあるかを示している。このウインドウ９００から、ユーザはどのサーバが共通のキープロセスを実行しているかを決定することによって統合のために候補のサーバをさらに分析することができる。

【０１０１】

追加の分析機能は、メモリおよびプロセッサのロードの指示を与え、アンダーロードまたはオーバロードされるサーバを識別する助けをする。アンダーロードされるサーバは、そのアプリケーションを別のサーバへと統合させる候補であり得る。さらに、既にオーバロードされたサーバは、統合において追加のアプリケーションを受け入れるのに適切な候補ではなく、実際には別のサーバに移動されたそのアプリケーションの１つ以上から利益を獲得し得る。図１０は、ＣＰＵおよびメモリの利用を表示するための例示のＵＩを示している。ウインドウ１０００は２つのペイン１００２および１００４を有する。ペイン１００２は、サーバのインベントリの階層的なリストを提供する。ペイン１００４は、組合された平均的なＣＰＵおよびシステムにおけるサーバのためのメモリの利用を示すディスプレイを提供し、互換性の分析を助ける。バー１００６は、特定のサーバ上のＣＰＵおよびメモリロードを図形で示し、ＣＰＵロードを示す部分１００６ａと、メモリロードを示す部分１００６ｂとを有する。スライド１００８および１０１０は、ユーザが結果をフィルタリングすることができる機構を提供する。すなわちスライド１００８を設定することによって、ユーザは、最小のＣＰＵの利用がスライダによって設定されたしきい値よりも低いディスプレイからこれらのシステムを排除することができ、スライド１０１０を設定することによって、ユーザは、ＣＰＵの利用がスライダによって設定された最大のＣＰＵの利用しきい値を超えるシステムを排除することができる。同様に、スライド１０１２および１０１４により、ユーザは、最小および最大のしきい値を設定することによってメモリの利用をフィルタリングすることができる。フィルタリングによってユーザは統合の候

補であるソースサーバを迅速に識別することができる。最小使用可能時間スピンボックス１０１６を変更して、最後の再始動以降の動作の時間が、示された時間数よりも低いディスプレイからこれらのシステムを排除することができる。

【０１０２】

図１１は、サーバ統合のために与えられる分析ツールに関するさらなる詳細を提供する。ここで、ウインドウ１１１０は２つのペイン１１０２および１１０４を提供する。ペイン１１０２は、サーバファーム、たとえばシステムおよびアプリケーション発見ツールによって発見されたサーバファーム１１０におけるサーバのすべてをリストする。ペイン１１０４は、ラジオボタン１１０４および１１０６によってユーザがプロセスまたはシステムの互換性を選択するための機構を提供する。この例では、ユーザはシステムの互換性分析を選択している。その後でユーザは、ソースシステム１１０８、たとえば統合のためのサーバ候補および１つ以上のターゲットシステム１１１０を選択することができる。ソースシステムプロセスはボックス１１１２における表示である。

【０１０３】

図１２は、ソースサーバをターゲットサーバへと統合した結果の、表示インジケータによる分析をさらに詳述している。ウインドウ１２００は、図１１に示されるようにウインドウ１１００でなされた選択の結果を示す。ウインドウ１２００は、選択されたソースサーバＯＴＧ－ＴＥＳＴ－ＳＲＶ３（１．２）をターゲットサーバＯＴＧ－ＴＥＳＴ－ＳＲＶ２（１．２）へと統合した結果を表示している。ターゲットシステムはカラム１２０２に表示される。カラム１２０４は、どれだけ多くのＤＬＬがソース上で同じかを示し、ターゲットサーバおよびカラム１２０６は、どれだけ多くの共通のＤＬＬが異なっているかを示す。共通のＤＬＬは、たとえばウインドウズシステム３２のディレクトリに置かれることによって、システムにおけるすべてのアプリケーションによって用いられるものである。カラム１２０８は、統合前の目標のロードの割合を示し、カラム１２１０は統合後のターゲットのロードの割合を示す。ソースサーバからのＣＰＵの利用値を、ターゲットサーバの処理電力に正規化する。同様に、カラム１２１４および１２１６は、ターゲットマシンのメモリへの影響を表示する。ソースサーバからのメモリロード値は、ターゲットサーバ上のメモリのサイズに正規化される。この表示によってユーザは、ソースサーバのターゲットサーバへの統合がターゲットサーバを利用目標内に保つかどうかを迅速に決定し、ソースサーバから移動されたアプリケーションをサポートするためにどれだけ多くの追加のＤＬＬがターゲットサーバにロードされる必要があるかの指示を与える。

【０１０４】

システムの互換性に加えて、プロセスの互換性は、どのサーバを統合するかを決定する際に考慮すべき重要な事項である。プロセスの互換性の詳細な選択１１０６が図１１のペイン１１００で行なわれるときに、ソースシステムプロセスのリストボックス１１１２が使用可能になり、ユーザは１つ以上のプロセスを選択する。次にユーザはターゲットシステムリストボックス１１１０から単一のターゲットサーバを選択する。図１３は、プロセス分析の結果を表示し、かつユーザがプロセスの互換性を決定するのを助けるＵＩを提供する。ウインドウ１３００は、ソースおよびターゲットサーバ上の共通のＤＬＬの互換性および差の比較を表示する。カラム１３０２は共通のＤＬＬ名を表示し、カラム１２０４はそのバージョンを表示し、カラム１３０６はそのカラムがターゲットサーバ上に存在する（“１”）か存在しない（“０”）かを示す。さらに、ＤＬＬがターゲットサーバに存在しても、カラム１３０８はソースおよびターゲットのバージョンが同じ（“１”）か異なる（“０”）かの指示を与える。ターゲットシステム上のＤＬＬのバージョンが異なるときに、カラム１３１０はターゲットシステムで発見されたバージョンを含む。ここに示されるように、ソース上のＤＬＬの多くはターゲットサーバ上にも存在する。しかしながら、ターゲットバージョンはソースバージョンに一致しない。カラム１３０４および１３１０は、ソースおよびターゲットのＤＬＬバージョンそれぞれを与える。このようにして、ユーザはターゲットバージョンがＤＬＬの新しいバージョンであるかどうかを迅速に決定し、恐らく更新する必要性を減少させることができる。

【０１０５】
図１４および１５は、データベースの統合のコンテクストにおいて上述したのと同じ分析ツールの多くを提供する。サーバ上のアプリケーションおよびプロセスを統合するのに加えて、データベースの統合は統合の重要な局面でもある。データベースの統合は、データベースのスキーマがさまざまなサーバ上のデータベースまたはデータベースのインスタンス間でどのように異なるかについての理解を必要とする。より特定的に、データベースの統合は、同一ではない複数のデータベースが、組み合わせることができる共通の情報を十分有し得るという認識によって利用可能であり得る。この共通性は、少なくとも始めは、ターゲットデータベースがソースデータベースにおけるカラムのすべてまたはソースデータベースの十分な数のカラム、ならびにカラムおよび／またはテーブルをソースデータベースから追加する能力を有することを必要とする。その後で、トリガ、記憶されたプロシージャ、警告等をターゲットデータベースに移動させる等の追加の必要性に応えることができる。
【０１０６】
図１４は、共通のＳＱＬサーバログインの高位レベルの図を示している。この例において、ウインドウ１４００は２つのペイン１４０２および１４０４に分割される。ペイン１４０２は上記に示されたような発見の際にサーバにつき収集されたデータベースインベントリのリストを提供する。ペイン１４０４は、サーバファーム、たとえば１１０における多数のサーバで発見された共通のＳＱＬログインのすべてをリストする。カラム１４０６はデータベースのためのログイン名を与える。カラム１４０８はインスタンス名を与える。したがって、ユーザは共通のログイン名を有するどのデータベースがどのサーバ上にあるかを容易に決定することができる。
【０１０７】
データベースの互換性の詳細の選択１１１４が図１１のペイン１１００において行なわれるときに、ユーザはデータベースの互換性の分析を行なうことができる。図１５はデータベースの互換性を分析する必要のある追加の情報を与える。この例で、ウインドウ１５００は２つのペイン１５０２および１５０４を提供する。ペイン１５０２はペイン１４０２と同様である。ペイン１５０４は、テーブル名およびカラム名のリストを提供し、スキーマの共通性および差を示す。カラム１５０８は、問題となるテーブルのための、テーブル名およびカラム名のリストを提供する。カラム１５０６は、カラム１５０８にリストされたアイテムがデータベーステーブルであるかまたはデータベースカラムであるかを識別するアイテムの種類を提供する。カラム１５１０は、カラム１５０８におけるアイテムがターゲットサーバ上に存在する（“１”）か存在しない（“０”）かの指示を与える。カラム１５１２は、ソースおよびターゲット上のアイテムが互換性がある（“１”）か、互換性がない（“０”）か、またはそれを決定することができない（“？？？”）かの指示を当たる。
【０１０８】
図１６Ａおよび１６Ｂは、上述の分析ツールの実現についてのさらなる詳細を示す。特に、選択されたシステムおよびデータベースインベントリの選択されたＸＭＬファイルがデータベース２０６にロードされる（図２を参照）。次に、ＳＱＬのクエリをデータベースにおけるデータに対して実行し、分析を行なう、すなわち１つのサーバにおけるインベントリを別のサーバにおけるインベントリと比較する。図１６Ａは、収集されたＸＭＬデータを記憶するために用いることのできるスキーマ２０６ａの高レベルの図を示している。スキーマは使用することのできるテーブルの種類を示している。ＸＭＬデータは、ＸＭＬバルクロードまたは他のＳＱＬＸＭＬコマンド等の技術を知ることによってＳＱＬデータベースにロードすることができる。
【０１０９】
好ましくは、より柔軟なアプローチが用いられる。このような実現化例において、ＸＭＬローダはマイクロソフトＸＭＬパーザを用いてＸＭＬコンテンツをデータセットにパーズする。次にデータセットを用いて関係レコードを確立し、リレーショナルデータベース

、たとえばデータベース２０６に記憶する。

【０１１０】

スキーマ２０６ａは、システム名、製造元、型番、システムメモリ情報等の情報、ならびにデータのソース、すなわちどのＸＭＬファイルかおよびバージョン番号かについての情報を含むSysinfoテーブル１６０２を含む。HardwareInfoテーブル１６０４は、プロセッサの数および利用可能なプロセッサ等のサーバハードウェア情報を含む。ネットワークテーブル１６０８は、ＮＩＣ識別子、ＩＰアドレス等のさまざまなネットワーク情報を含む。デバイステーブル１６１０はデバイス名等のハードウェアデバイスに関する情報を含む。ドライブテーブル１６０６は、合計のバイトストレージ、バイトフリー、ボリューム名等のサーバドライブ情報を含む。アプリケーションテーブル１６１２は、アプリケーション名およびバージョン番号等の情報を含む。プロセステーブル１６１４は、プロセス所有者、ＣＰＵ利用情報、メモリ利用情報等のプロセスに関する情報を含む。モジュールテーブル１６１８は、モジュールサイズ、モジュール名等のモジュール情報を含む。プロセスモジュール関連テーブル１６１６はモジュールを親プロセスに関連付ける。

【０１１１】

スキーマ２０６ａは、アプリケーションの統合といったシステムインベントリ分析を行なうのに有益である。データベース分析に関して、図１６Ｂはデータインベントリ ＸＭＬファイルとともに用いるための高位レベルのスキーマを示している。このように、上述のようにさまざまなサーバから発見された選択されたデータベースＸＭＬファイルは、スキーマ２０６ｂに従ってデータベース２０６にロードされる。サーバテーブル１６２０はどのサーバが、発見されたデータベースを維持しているかを識別する情報を保持する。インスタンステーブル１６２２は、サーバ、たとえばＳＱＬサーバ６．０およびＳＱＬサーバ７．０にインストールされたデータベースサーバの１つ以上のインスタンスの名前に関する情報を保持する。各インスタンスにつき、データベーステーブル１６２４はそのインスタンス内の１つ以上のデータベースに関する情報を含む。テーブル１６２４における各データベースについて、「テーブル」テーブル１６２６はテーブル名のすべてを有し、カラムテーブル１６２８は所与のテーブルのためのカラムのすべてを保持する。プロシージャテーブル１６３２は、データベースで用いられる記憶されたプロシージャの名前等の情報を保持する。機能テーブル１６３６は、データベースに関連付けられた機能名のリストを保持する。トリガテーブル１６４０は、データベースに関連付けられたトリガ名のリストを保持する。ＤＢロールテーブル１６４４は、データベースに関連付けられたデータベースのロールのリストを保持する。さらに、インスタンステーブル１６２２における各インスタンスにつき、ＤＴＳＰパッケージテーブルは、パッケージおよび所有者の名前等のデータベースに関連付けられたデータ送信サービスパッケージに関連する情報を保持する。ログインテーブル１６３８は、ユーザ名等のログイン情報を保持する。最終的に、サーバロールテーブル１６４２は、メンバ名およびメンバのＳＩＤ等のサーバロールに関連する情報を保持する。

【０１１２】

分析が完了して統合の候補が識別された後で、ターゲットサーバに移動および／またはロードされなければならないかなりの数のファイルがあるかも知れない。図１７および１８は、少なくとも新しいアセットのターゲットサーバへの配置の局面を自動化する助けをするサブジェクトシステムの局面を示している。図１７は、例示のアセット配置ＵＩを示している。ウインドウ１７００はドロップダウンボックス１７０２を有し、配置ツールが選択される。セレクトボックス１７０８は、アセットが配置されるべきターゲットサーバを識別するための機構を提供する。ペイン１７０６は、ターゲットサーバ上で配置されるべきさまざまなアセットのすべてを識別する。特に、ボックス１７０４は、ユーザにターゲットサーバ上のアセットの配置と関連して用いられる配置のルールを規定するための機能を提供する。

【０１１３】

ユーザが配置ルールが用いられるべきであると決定した後で、選択する定義ボタン１７

０５を選択するとルールのエディタが開始される。図１８はルールエディタをさらに示している。ウインドウ１８００は、以下のテンプレートを含む予め規定されたルールテンプレートの例示のリストを提供する。

【０１１４】

ドライブ上の最小ディスクスペースをチェックする。

【０１１５】

最小メモリ（ＲＡＭ）をチェックする。

【０１１６】

プロセッサの最小数をチェックする。

【０１１７】

このアプリケーションのコピーが既にインストールされているかどうかをチェックする。

【０１１８】

競合するアプリケーションがインストールされていないことを確認する。

【０１１９】

要求されたアプリケーションが既にインストールされていることを確認する。

【０１２０】

もちろん、対象のシステムのこの局面の範囲から逸脱することなく他のルールテンプレートを規定することができる。

【０１２１】

図１９は配置システムの局面をさらに示している。ここで、本明細書において上述のように、統合情報が収集され分析されている。その後で、統合されたサーバファーム１２０が配置される。この目的のために、実行可能要素、２進数およびインストールを実行するのに必要なファイルの本質的にすべてが、セットアップファイルを有するフォルダへと配置される。典型的に、これはフォルダ毎の単一のアプリケーションであるが、このように制限される必要はない。さらに、配置のためのテンプレートが選択される。たとえば、最小のメモリが選択された場合、ユーザは最小のメモリ要件、たとえば５１２ＭＢを規定する。同様に、他の選択されたテンプレートのためのパラメータ、たとえば2プロセッサ、ディスクスペースの１ギガバイト等が規定される。ある時点で、配置のためターゲットサーバが選択される。図１９に示されるように、サーバ１２０ａおよび１２０ｂが選択された。代わりにすべてのドメインが選択され得る。システムの発見の局面と関連して上述したように、ターゲットシステムのアセットが発見される。これは初期の統合プロセスの一部として実行されるか、または独立して実行することができる。

【０１２２】

発見された情報を含む関連するＸＭＬファイルはパーズされて、規定されたルールと比較される。ルールにパスした場合に、ファイルはターゲットサーバまたはサーバに送信されて、インストールおよび遠隔プロシージャの呼出しが行なわれてインストールを開始させる。好ましくは、送信されたインストールファイルは送信の前に圧縮され、ターゲット上で圧縮解除される。好ましくは、圧縮は、送信の前に構成ファイルをｚｉｐし、ターゲットサーバで構成フォルダをｕｎｚｉｐすることによって行なわれる。ｕｎｚｉｐプログラムは、たとえば自己抽出ファイルとしてｕｎｚｉｐプログラムをバンドルすることによってプロセスの一部として送信され得る。

【０１２３】

好ましくは、規定されたルールの検査はＸＭＬファイルに対してＸＰＡＴＨクエリによって行なわれる。たとえば、発見と関連して先に規定した例示のＸＭＬファイルを用いると、以下のＸＭＬの抜粋に適用された場合に、プロセッサの数のＸＰＡＴＨクエリは“２”に戻る。

【０１２４】

【数１５】


```
<PE_HdweInfo ID="HI" parent_ID="5008DJUL1030-SI" numberOfProcessors="2"
availableProcessorMask="3" processorLevel="15" processorRevision="521">
  <PE_ProcessorSpeed parent_ID="HI" procNum="0" speed="2992" />
  <PE_ProcessorSpeed parent_ID="HI" procNum="1" speed="2992" />
 <PE_Device parent_ID="HI" deviceLocation="LPT1" cmpLocation="">
  <deviceName>Printer Port Logical Interface</deviceName>
  </PE_Device>
 <PE_Device parent_ID="HI" deviceLocation="USB Device" cmpLocation="">
  <deviceName>ViewSonic Color Pocket PC V37</deviceName>
  </PE_Device>
 <PE_Device parent_ID="HI" deviceLocation="" cmpLocation="0,0,0,1">
  <deviceName>HL-DT-ST RW/DVD GCC-4480B</deviceName>
  </PE_Device>
 ...
```


【０１２５】

類似のＸＰＡＴＨクエリは他のルール値にも適用することができる。

【０１２６】

上述の配置は、統合のコンテクスト以外のコンテクストにも使用され得る。たとえば、会社はその組織全体を通して複数のクライアントマシンにわたるアプリケーションを配置したいかも知れない。上記の技術によって、単一の配置セットアップは、規定されたルールに応じる選択されたマシンでのアプリケーションを自動的にインストールすることができる。

【０１２７】

上述の統合は例示の説明に過ぎず、すべてのサーバの統合においてアプリケーションおよびデータベースが統合されることを示すことを意図するものではない。むしろ、この例は、可能であり得る統合の兆候を示すことを意図している。包括的なテーマは、統合１１５がハードウェア、ソフトウェア、サーバファーム１１０等のサーバファーム上のデータのインベントリを決定しかつそのハードウェア、ソフトウェアおよびデータの統合を簡略化するためのツールを提供することである。

【０１２８】

以下に記載される本発明の実施例の要素は、ハードウェア、ファームウェア、ソフトウェアまたはその如何なる組合せによっても実現され得る。ハードウェアという用語は、一般的に電気的、電磁的、光学的、電気光学的、機械的、電気機械的な部品等の物理的構造を有する要素のことを指し、一方でソフトウェアという用語は一般的に、論理構造、方法、プロシージャ、プログラム、ルーチン、プロセス、アルゴリズム、公式、関数、式等のことを指す。ファームウェアという用語は一般的に、ハードウェア構造（たとえばフラッシュメモリ、ＲＯＭ、ＥＲＯＭ）において実現または実施される、論理構造、方法、プロシージャ、プログラム、ルーチン、プロセス、アルゴリズム、公式、関数、式等のことを指す。ファームウェアの例は、マイクロコード、書込可能な制御記憶装置、およびミクロプログラム構造を含み得る。ソフトウェアまたはファームウェアで実現されるときに、本発明の実施例の要素は、本質的に必要なタスクを行なうためのコードセグメントである。ソフトウェア／ファームウェアは、本発明の一実施例で記載される動作を実行するための実際のコード、または動作をエミュレートもしくはシミュレートするためのコードを含み得る。プログラムまたはコードセグメントは、プロセッサまたは機械がアクセス可能な媒体において記憶されるか、または送信媒体上で、搬送波において実現されるコンピュータデータ信号、もしくは搬送波によって変調される信号によって送信される。「プロセッサが読取可能なまたはアクセス可能な媒体」または「機械が読取可能なまたはアクセス可能な媒体」は、情報を記憶、送信または転送することのできる如何なる媒体も含み得る。プロセッサが読取可能なまたは機械がアクセス可能な媒体の例は、電気回路、半導体メモリ

デバイス、読取専用メモリ（ＲＯＭ）、フラッシュメモリ、消去可能なＲＯＭ（ＥＲＯＭ）、フロッピー（登録商標）ディスク、コンパクトディスク（ＣＤ）、ＲＯＭ、光ディスク、ハードディスク、光ファイバ媒体、無線周波数（ＲＦ）リンク等を含む。コンピュータデータ信号は、電子ネットワークチャネル、光ファイバ、エアリンク、電磁リンク、ＲＦリンク等の送信媒体上で伝播することができる如何なる信号も含み得る。コードセグメントは、インターネット、イントラネット等のコンピュータネットワークを介してダウンロードされ得る。機械がアクセス可能な媒体は製品において実現され得る。機械がアクセス可能な媒体は、機械がアクセスするときに、機械に以下に記載する動作を行なわせるデータを含み得る。機械がアクセス可能な媒体は、そこに埋込まれたプログラムコードを含み得る。プログラムコードは、以下に記載される動作を行なう機械が読取可能なコードを含み得る。「データ」という用語は、ここでは機械が読取可能な目的のために符号化される如何なる種類の情報のことも指す。したがって、それはプログラム、コード、データ、ファイル等を含み得る。

【０１２９】

本発明の実施例のすべてまたは一部は、ハードウェア、ソフトウェア、もしくはファームウェア、またはその如何なる組合せによっても実現され得る。ハードウェア、ソフトウェア、ファームウェア要素は、互いに結合された複数のモジュールを有し得る。ハードウェアモジュールは、機械的、電気的、光学的、電磁的または如何なる物理的接続によっても別のモジュールに結合される。ソフトウェアモジュールは、関数、プロシージャ、方法、サブプログラム、またはサブルーチン呼出し、ジャンプ、リンク、パラメータ、変数および引数、関数リターン等によって別のモジュールに結合される。ソフトウェアモジュールは別のモジュールに結合されて、変数、パラメータ、引数、ポインタ等を受信し、および／または結果、更新された変数、ポインタ等を生成もしくは渡す。ファームウェアモジュールは、上述のハードウェアおよびソフトウェアの結合の方法の如何なる組合せによっても別のモジュールに結合される。ハードウェア、ソフトウェア、またはファームウェアモジュールは、別のハードウェア、ソフトウェア、またはファームウェアモジュールのいずれか１つに結合され得る。モジュールは、プラットホーム上で実行されるオペレーティングシステムと相互作用するためのソフトウェアドライバまたはインターフェイスであってもよい。モジュールは、ハードウェアデバイスへのおよびそこからのデータを構成、設定、初期化、送信および受信するためのハードウェアドライバであってもよい。装置は、ハードウェア、ソフトウェアおよびファームウェアモジュールの如何なる組合せも含み得る。

【０１３０】

本発明の実施例は、通常フローチャート、フロー図、構造図またはブロック図として示されるプロセスとして記載され得る。フローチャートは一連のプロセスとしての動作を記載し得るが、動作の多くは並列してまたは同時に行なうことができる。さらに、動作の順序は再び配置され得る。プロセスはその動作が完了したときに終了され得る。

【０１３１】

多くの追加の修正が本発明の新しい教示および利点から実質的に逸脱することなく例示の実施例においても可能であることを当業者は容易に認識するであろう。いかなるこのような修正も、別掲の例示のクレームによって規定されるような本発明の範囲内に含まれることが意図される。

【図面の簡単な説明】

【０１３２】

【図１】サーバファーム統合の例示の図を示している。

【図２】図１における統合で用いられるような統合システムのさらなる詳細を示している。

【図３】サーバ統合の発見の局面を呼出すための例示のユーザインターフェイスを示す図である。

【図４】システムの発見の配置の局面を示すブロック図である。

【図５】全体的なサーバ統合を示す高レベルのフロー図である。
【図６】発見されたサーバ情報の階層的なフォルダ図を示す例示的なユーザインターフェイスを示す図である。
【図７】サーバ上で発見されたアプリケーションの詳細を表示するための例示のユーザインターフェイスを示す図である。
【図８】サーバファームにおけるサーバ間の共通性および差を決定する分析を助ける例示のユーザインターフェイスを示す図である。
【図９】サーバ間でのアプリケーションの共通性に関するさらなる分析の詳細を与える例示のユーザインターフェイスを示す図である。
【図１０】ＣＰＵの利用およびメモリ制約によってサーバを見るための例示のユーザインターフェイスを示す図である。
【図１１】統合分析のためのソースおよびターゲットシステムを選択するための例示のユーザインターフェイスを示す図である。
【図１２】ソースサーバをターゲットサーバに統合した結果を示す例示のユーザインターフェイスを示す図である。
【図１３】プロセス分析の結果を表示する例示のユーザインターフェイスである。
【図１４】データベース統合において用いるための例示のユーザインターフェイスであり、共通のＳＱＬログインに関する情報を提供する図である。
【図１５】データベース統合で用いるための例示のユーザインターフェイスであり、テーブルおよびカラムの互換性に関する情報を提供する図である。
【図１６Ａ】システムおよびアプリケーションの互換性の分析において用いるためのシステムおよびアプリケーションのデータベースモデルの例を示す図である。
【図１６Ｂ】データベースの互換性および統合分析で用いるためのデータベースモデルの例である。
【図１７】サーバ統合におけるアプリケーションの配置等のネットワークにおけるコンピュータシステムへアプリケーションを配置する際に用いられる例示のユーザインターフェイスを示す図である。
【図１８】アプリケーションの配置と関連した配置のルールを選択するための例示のユーザインターフェイスを示す図である。
【図１９】サーバ統合アプリケーションにおけるアプリケーションの配置を示すブロック図である。

【図１】

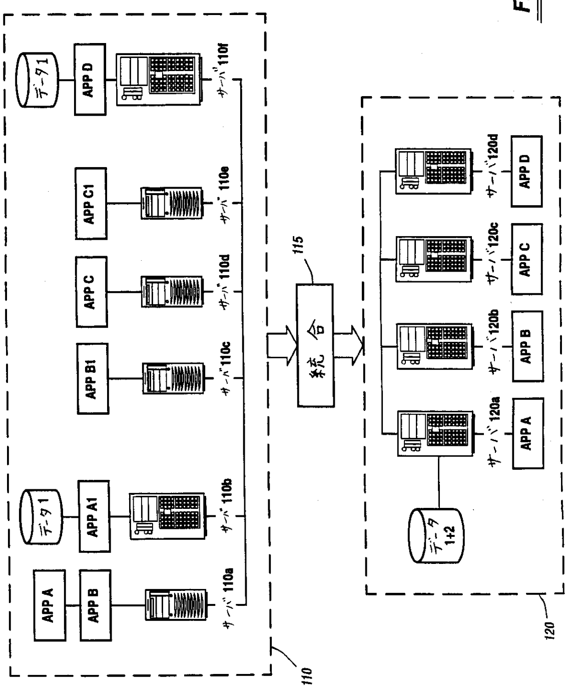


Figure 1

【図２】

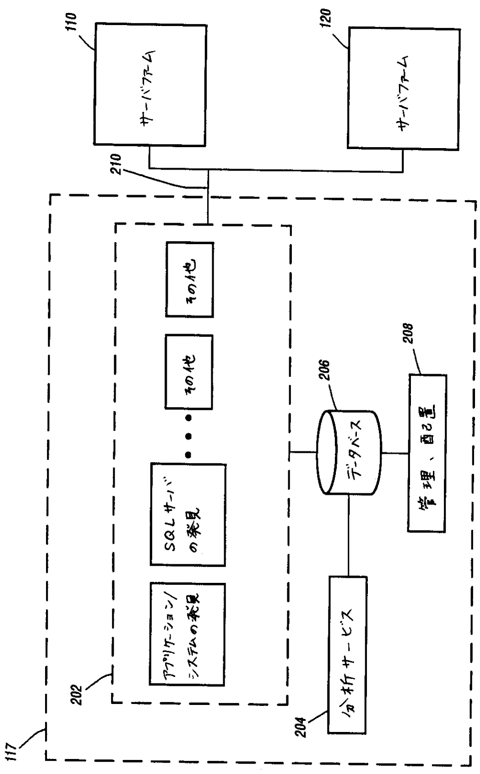


Figure 2

【図３】

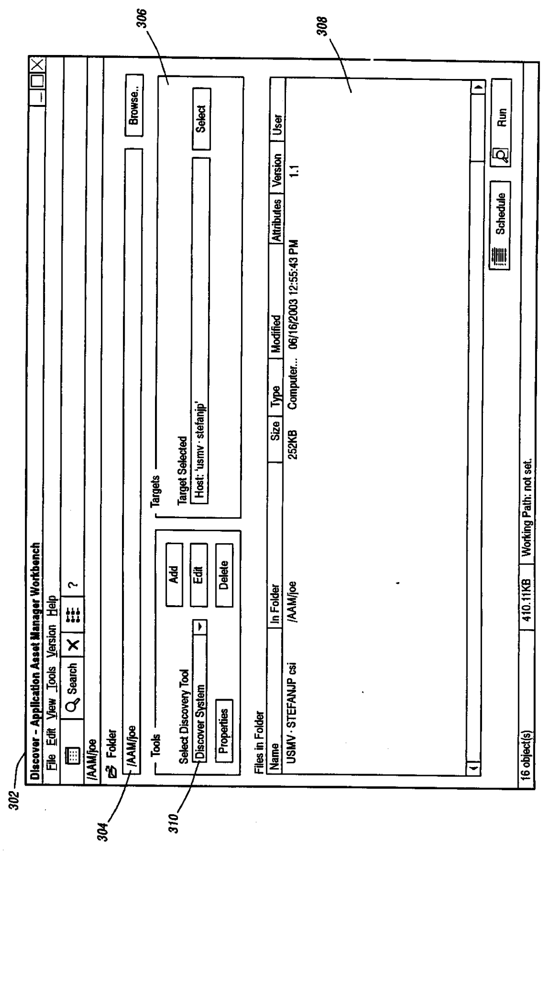


Figure 3

【図４】

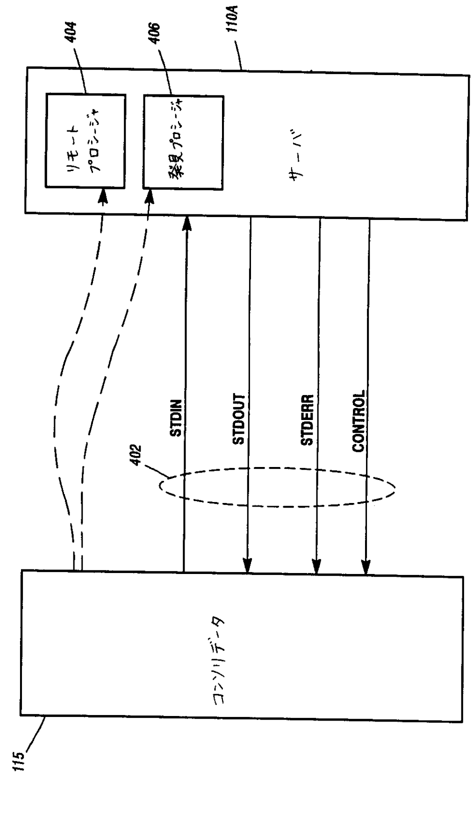


Figure 4

【図５】

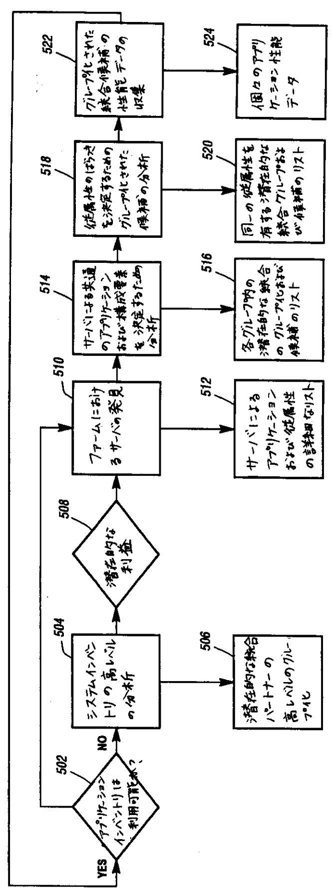


Figure 5

【図６】

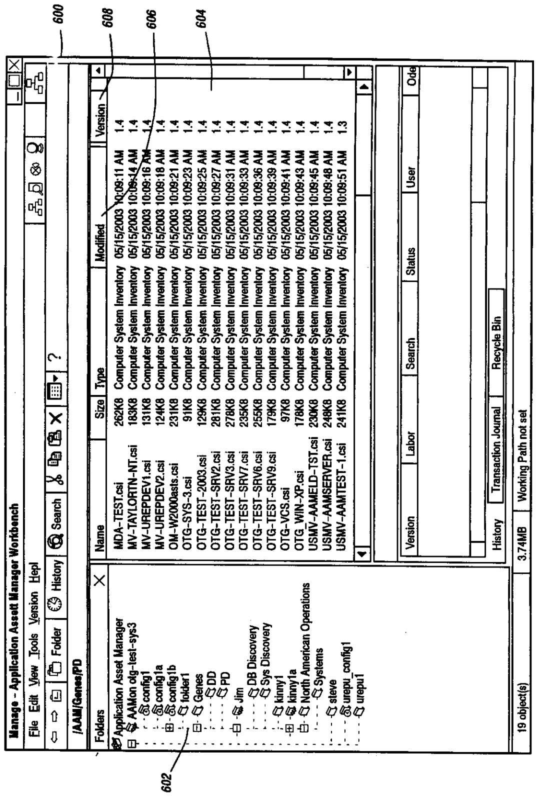


Figure 6

【図７】

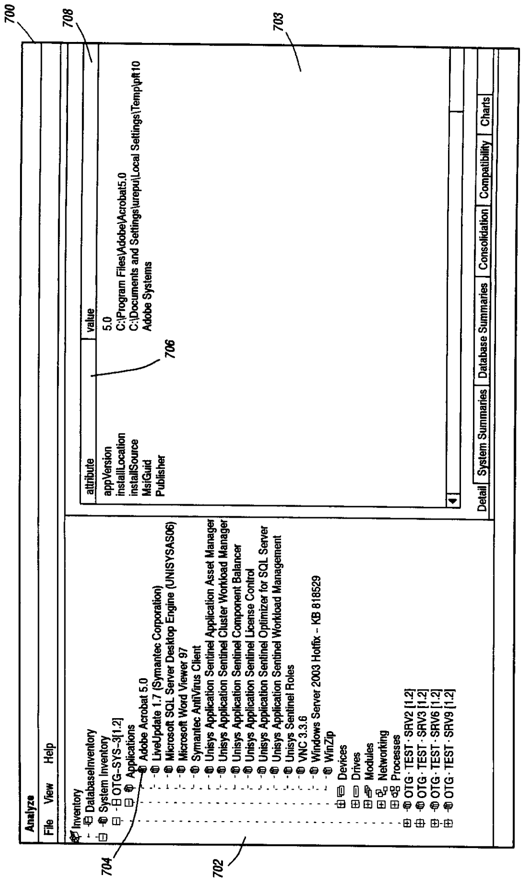


Figure 7

【図８】

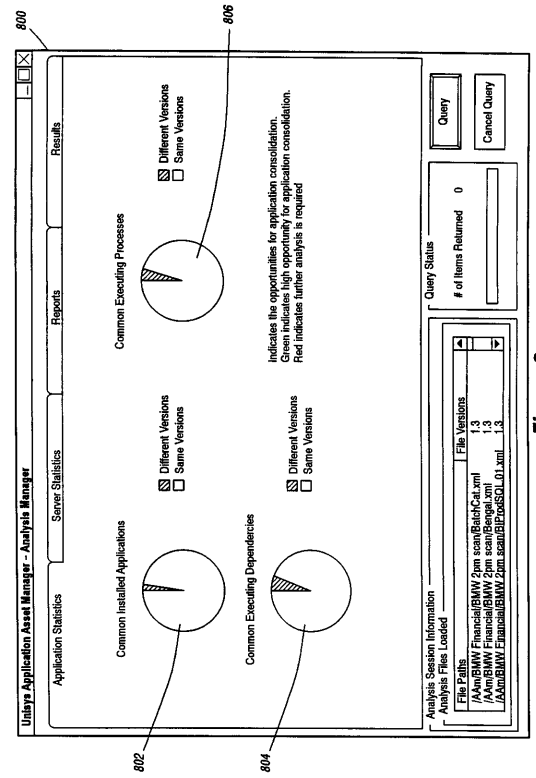


Figure 8

【図９】

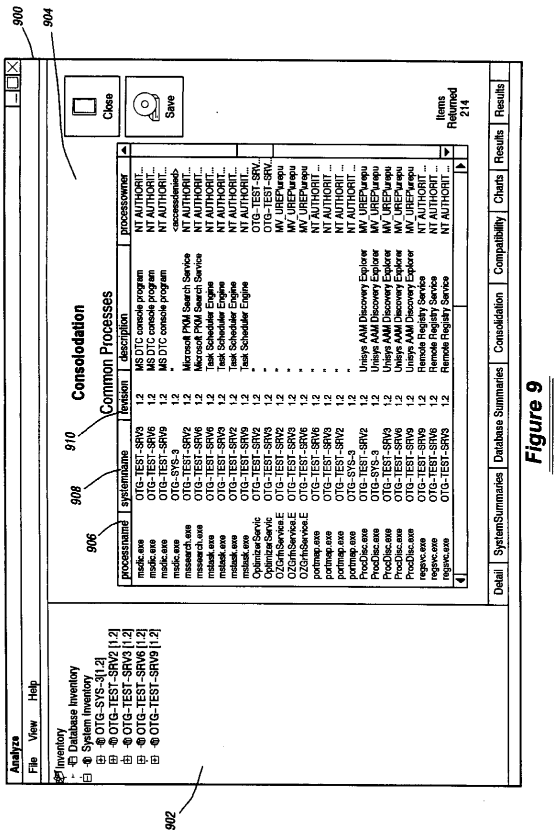


Figure 9

【図１０】

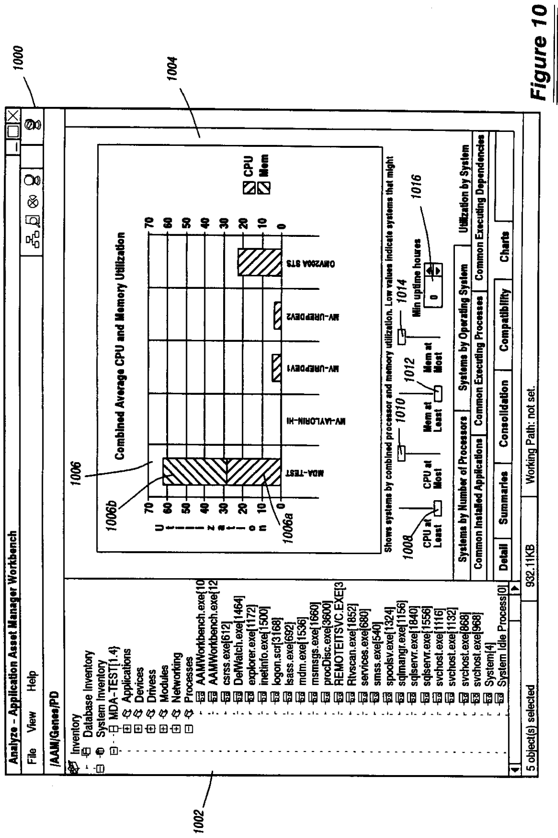


Figure 10

【図１１】

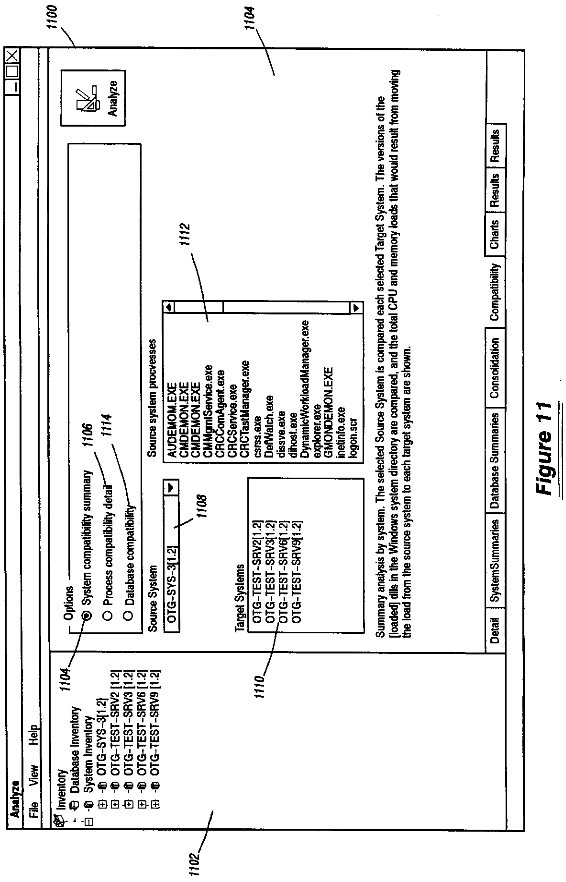


Figure 11

【図１２】

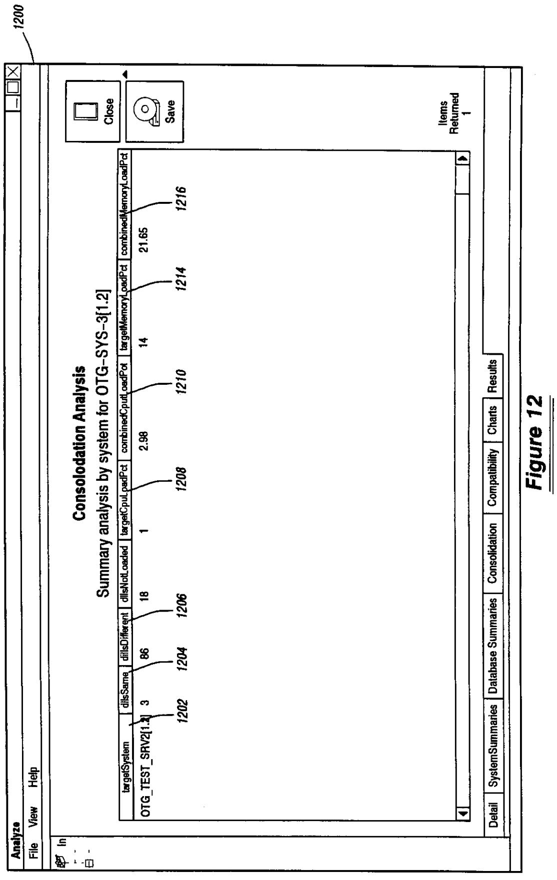


Figure 12

【図１３】

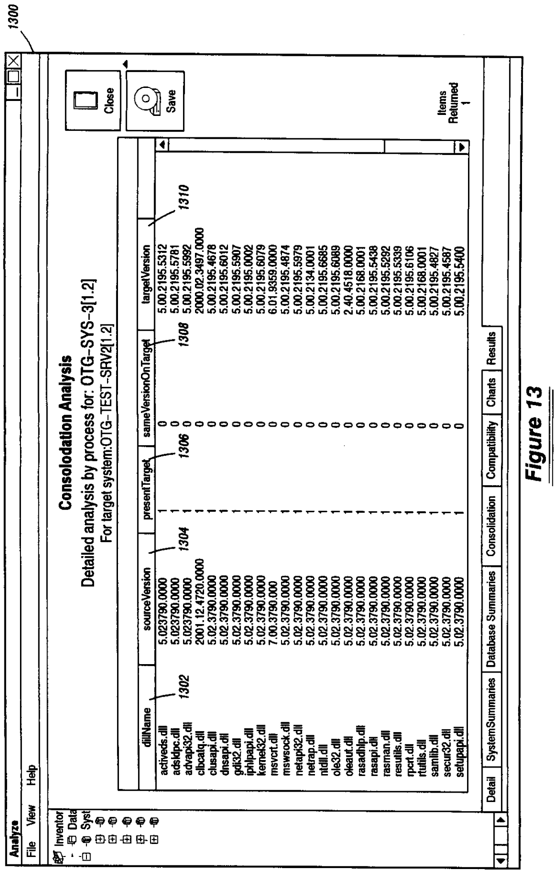


Figure 13

【図１４】

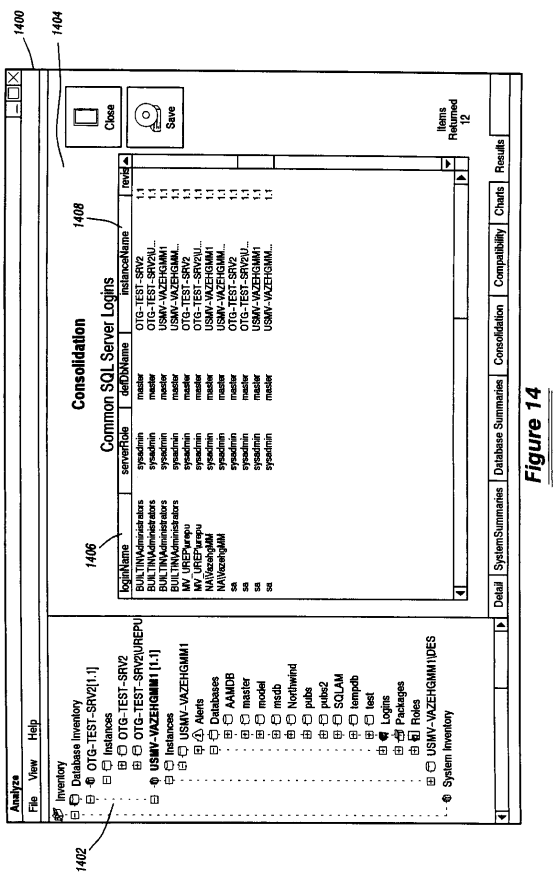


Figure 14

【図１５】

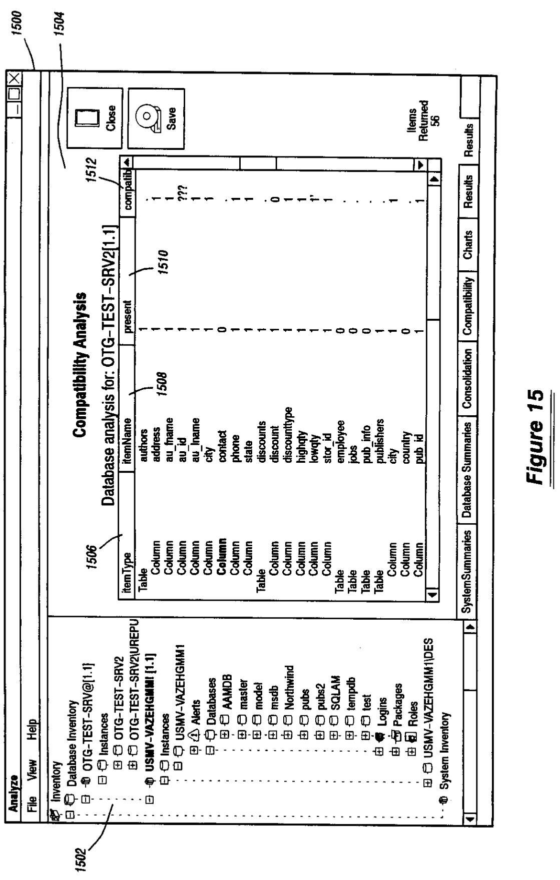


Figure 15

【図１６Ａ】

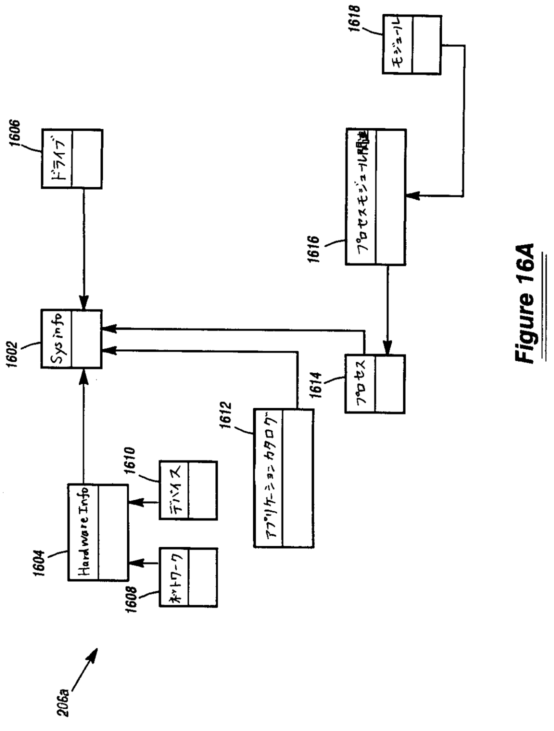


Figure 16A

【図１６Ｂ】

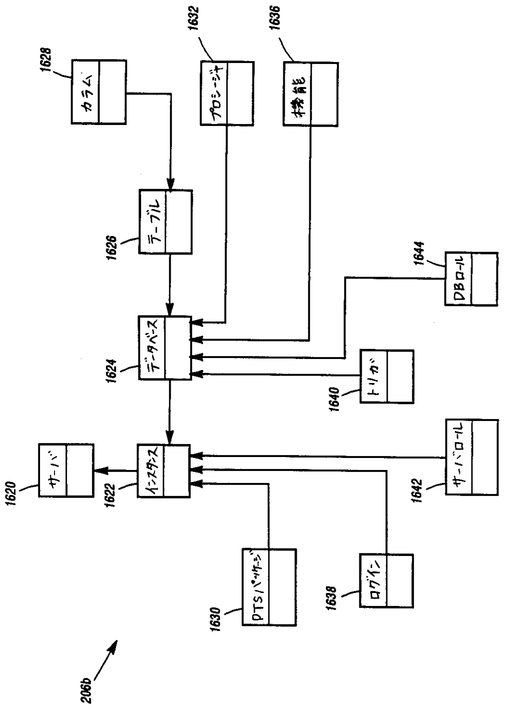


Figure 16B

【図１７】

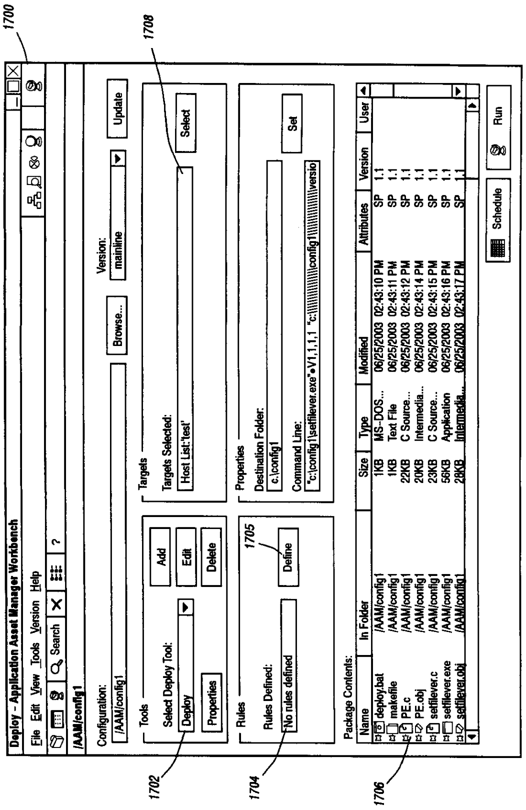


Figure 17

【図１８】

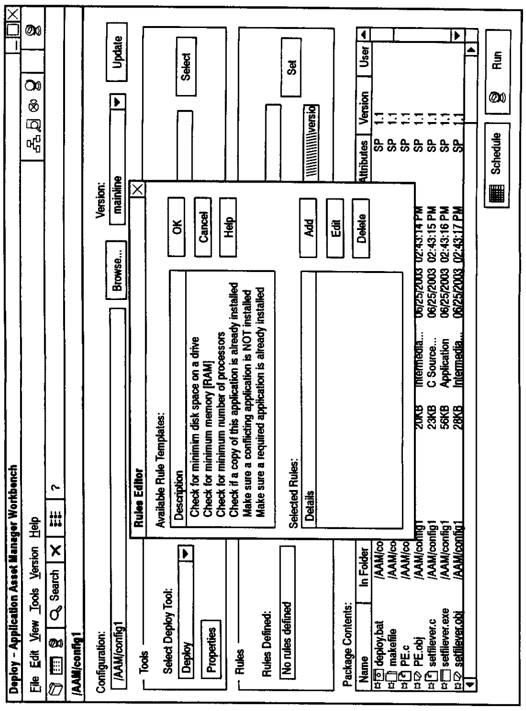


Figure 18

【図１９】

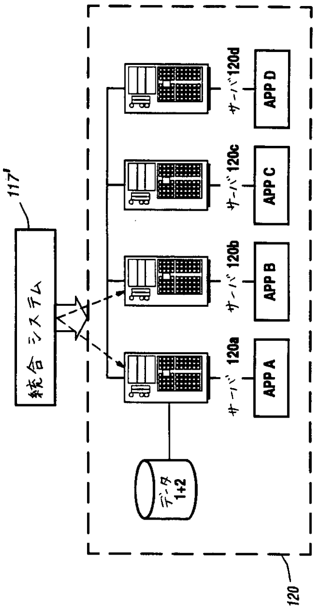


Figure 19

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US04/08667

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 15/16
US CL : 709/217
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 709/217

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,E	US 6,754,703 B1 (SPRING) 22 June 2004 (22.06.2004), column 3, lines 45-67 and column 4, lines 1-31.	1-36
A	US 6,076,106 A (HAMNER et al.) 13 June 2000 (13.06.2000), column 1, lines 55-67.	1-36
A,E	US 6,711,127 B1 (GORMAN et al.) 23 March 2004 (23.03.2004), column 2, lines 1-35.	1-36

☐ Further documents are listed in the continuation of Box C.　☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 July 2004 (23.07.2004)	17 AUG 2004
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450 Alexandria, Virginia 22313-1450 Facsimile No. (703)305-3230	Authorized officer Jeffrey A. Gaffin Telephone No. (703) 305 3900

Form PCT/ISA/210 (second sheet) (July 1998)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IT,LU,MC,NL,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NA,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,YU,ZA,ZM,ZW

(74)代理人 100096781
弁理士 堀井 豊
(74)代理人 100098316
弁理士 野田 久登
(74)代理人 100109162
弁理士 酒井 將行
(72)発明者 ステファニク，ジョゼフ・ピィ
アメリカ合衆国、９２６７３ カリフォルニア州、サン・クレメンテ、テラッツァ・エスコンディダ、６８０５
(72)発明者 ムチュラー，ユージーン・オゥ，ザ・サード
アメリカ合衆国、９２６７２ カリフォルニア州、サン・クレメンテ、マラカイ、３９
(72)発明者 クライダー，アンソニー・エル
アメリカ合衆国、９２６８８ カリフォルニア州、ランチョ・サンタ・マルガリータ、モハーベ・ウェイ、１０
(72)発明者 バゼグー，マフムード・エム
アメリカ合衆国、９２６７７ カリフォルニア州、ラグーナ・ニゲル、フラワリッジ・サークル、１