

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4533474号
(P4533474)

(45) 発行日 平成22年9月1日 (2010.9.1)

(24) 登録日 平成22年6月18日 (2010.6.18)

| | |
|--------------------------------|-----------------------|
| (51) Int. Cl. | F I |
| G 0 6 F 15/00 (2006.01) | G 0 6 F 15/00 3 1 0 A |
| G 0 6 F 13/00 (2006.01) | G 0 6 F 13/00 3 5 1 H |

請求項の数 1 外国語出願 (全 24 頁)

| | | | |
|--------------|-----------------------------|-----------|------------------------|
| (21) 出願番号 | 特願平11-134712 | (73) 特許権者 | 597004720 |
| (22) 出願日 | 平成11年5月14日 (1999.5.14) | | サン・マイクロシステムズ・インコーポレ |
| (65) 公開番号 | 特開2000-67022 (P2000-67022A) | | ーテッド |
| (43) 公開日 | 平成12年3月3日 (2000.3.3) | | Sun Microsystems, Inc. |
| 審査請求日 | 平成18年5月11日 (2006.5.11) | | アメリカ合衆国 カリフォルニア州 95 |
| (31) 優先権主張番号 | 09/079499 | | 054 サンタ クララ ネットワーク |
| (32) 優先日 | 平成10年5月14日 (1998.5.14) | | サークル 4150 |
| (33) 優先権主張国 | 米国 (US) | (74) 代理人 | 100096817 |
| | | | 弁理士 五十嵐 孝雄 |
| | | (74) 代理人 | 100097146 |
| | | | 弁理士 下出 隆史 |
| | | (74) 代理人 | 100102750 |
| | | | 弁理士 市川 浩 |

最終頁に続く

(54) 【発明の名称】 コンピュータネットワーク内でデータを変換するための方法

(57) 【特許請求の範囲】

【請求項 1】

コンピュータ・ネットワークを通じてデータを送信する前に前記データを準備する方法であって、

サーバ・コンピュータが、前記クライアント・コンピュータ及びサーバ・コンピュータの間の接続を確立し、

クライアント・コンピュータが、クライアント・プロフィールを前記サーバ・コンピュータへ送信し、

サーバ・コンピュータが、第1の複数のデータ・アイテムを含む第1データ・セットをサーバ・コンピュータ上で検索し、

サーバ・コンピュータが、第2の複数のデータ・アイテムを含む第2データ・セットをサーバ・コンピュータ上で検索し、

前記第1データ・セット及び第2データ・セットはクライアント・コンピュータに関連していることと、

サーバ・コンピュータが、前記第1データ・セット内のデータ・アイテムを前記第2データ・セット内のデータ・アイテムへ関連付けし、これによって、それぞれが前記第1データ・セットにおける一のデータ・アイテムと前記第2データ・セットにおける一のデータ・アイテムとの間の関連性を特定する複数のデータ・アイテム・コレスポンドンスを形成し、

サーバ・コンピュータが、前記クライアント・コンピュータと前記サーバ・コンピュ

10

20

タとの間でデータを交換するために用いられるソフトウェアのバージョンを決定し、

サーバ・コンピュータが、各データ・アイテム・コレスポネンスにおいて特定された前記第1データ・セットにおける前記データ・アイテムおよび前記第2データ・セットにおける前記データ・アイテム間の優先順位を決定するために、一連のルールを前記複数のデータ・アイテム・コレスポネンスへ適用し、

サーバ・コンピュータが、少なくとも前記第1データ・セットの一部と少なくとも前記第2データ・セットの一部とを組み合わせることにより、第3の複数のデータ・アイテムを含む第3データ・セットを導出することを含み、前記第3のデータセットには、前記各データ・アイテム・コレスポネンスについて、前記優先順位に基づき前記第1データ・セットにおける前記データ・アイテムまたは前記第2データ・セットにおける前記データ・アイテムが含まれる、方法。

10

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はコンピュータ・ソフトウェア及びコンピュータ・ネットワーク・アプリケーションに関する。より詳細には、本発明は、クライアント・サーバ・アプリケーションおよびコンピュータ・ネットワーク内の複数のコンピュータ間におけるコンフィギュレーション・データの交換に関する。

【0002】

【従来の技術】

20

従来のコンピュータ・ネットワークの一類型では、クライアントと称される一連のパーソナル・コンピュータ（例：サン・スパーク・ワークステーションまたはIBMパーソナル・コンピュータ）は、1つ以上のサーバ・コンピュータに対して接続する必要がある。クライアント・コンピュータは、一般的に独立しており、また、ユーザ・アプリケーションを実行すると共にネットワーク・オペレーションを実施するために必要な情報の殆どを自身のメモリ内に有する。即ち、クライアント・コンピュータはソフトウェア及びハードウェアの両方の機能及び要件に関する自身のコンフィギュレーションの情報を含む。ネットワーク・ソフトウェア・アプリケーションへのアクセス、電子メールの送受信及びネットワーク・データベースの情報の検索及び格納などの様々な理由から、クライアント・コンピュータはネットワーク・サーバへ一般的にアクセスする。しかし、一般的に、特定のクライアント・コンピュータ固有の情報は、そのクライアント・コンピュータ上に存在する。この情報には、例えば、メモリ容量、データベース種類数またはハードウェア仕様が含まれることが可能であり、ハードウェア仕様の例としては、データベースの種類または別のプロセッサの種類が挙げられる。クライアント・コンピュータは比較的独立しており、かつ自身のコンフィギュレーション情報を格納している（従って、この情報はサーバ・コンピュータ上で入手できない）。このため、クライアント・コンピュータ上でのデータ管理及びアプリケーション管理のタスクは重荷になってきている。

30

【0003】

ネットワークのサーバ上に存在するアプリケーションに対する小さな変更または調整を、クライアント・コンピュータへ伝播することは可能であるが、任意の大幅なアップグレード若しくは調整または新しいアプリケーションのインストールは、全てのクライアントが、ネットワーク管理者によって各クライアント・コンピュータがアクセスされると共に更新されることを要求する結果をもたらす。幾つかの企業では、ネットワークへ接続されたコンピュータの数は何万台にも及んでおり、アプリケーション・ソフトウェアまたは一般コンフィギュレーション・ソフトウェアに対する大幅な改訂またはアップグレードのインストールの負担は、高い費用を要する非能率的な長い時間を要する作業となっている。更に、殆どのクライアント・コンピュータは独立しているため、それぞれ異なる場所にある複数のクライアント・コンピュータを使用することを必要とする各ユーザが、アプリケーション及びコンフィギュレーション・データに関するパーソナル・プリファレンス（個人設定）を維持することは困難である。即ち、ユーザはパーソナル・プリファレンスを自分

40

50

が常に使用するクライアント・コンピュータへデフォルトとしてインストール可能であるにもかかわらず、他のクライアント・コンピュータのデフォルトを変更することなくこれらのデフォルトを前記の他のクライアント・コンピュータへ複製できる。

【 0 0 0 4 】

別の種類のコンピュータ・ネットワーク・コンフィギュレーションは、メインフレーム・コンピュータへ一般的に接続されたダム・ターミナル、即ち、シン・クライアント ("thin client") を使用している。この種のネットワークでは、ほぼ全ての処理及びデータはメインフレーム・コンピュータ上に存在する。シン・クライアントはこれらのアクティビティを一切遂行しない。この種のコンフィギュレーションでは、クライアントに関する全ての情報はメインフレーム・コンピュータのコントロール下にある。クライアント及びメインフレームの間の接続が終了した場合、全ての処理が停止し、クライアントは全てのアクティビティを実施できなくなる。

10

【 0 0 0 5 】

前記のように、従来のネットワーク・コンフィギュレーションでは、新しいソフトウェアまたは新しいアプリケーションをインストールするプロセスは静的プロセスである。このコンフィギュレーションでは、各パーソナル・コンピュータのコンフィギュレーション情報は各クライアント・マシン上で定義される。従って、ネットワーク管理者は各パーソナル・コンピュータ上の各コンフィギュレーションを静的に定義する必要がある。従来のコンピュータ・ネットワーク・コンフィギュレーションでは、特定の各サブシステム、即ち、クライアントのコンフィギュレーション情報は、このクライアント内でハードコードされている。更に、ネットワーク・サーバへ接続された複数の独立クライアントを使用する従来のネットワーク・コンフィギュレーションの場合、新バージョンのインストールまたはソフトウェアの大幅アップグレード (アップグレードはサブシステムのコンフィギュレーションに関する知識またはサブシステムのコンフィギュレーションへのアクセスを必要とする) などのアプリケーション・メンテナンスは、通常、アプリケーションまたはネットワークをダウンさせることを必要とする。

20

【 0 0 0 6 】

複数のクライアントと、クライアントが様々な理由から必要とする情報を有する 1 つのサーバとを備えた従来のコンピュータ・ネットワークの場合、クライアントが必要とするサーバ上の全データ、即ち、クライアントに関するサーバ上の全データは、サーバからクライアントへ転送されることが多い。一般的に、これは大量のデータの移動をともなう。これらのデータのうちの殆どはクライアントによって使用されない、即ち、クライアントのオペレーションに必要な。全データをクライアントへ転送することは、非能率的であり、かつネットワーク・トラフィックを増大する。更に、クライアントはサーバから転送された自身に関連する全ての情報を格納する十分なメモリ及び記憶装置を有する必要がある。例えば、大容量の記憶装置を有さない PDA 及びスマート・カードなどのデバイスは、自身に関連する可能性のあるコンフィギュレーション情報を含む全ての情報を自身のメモリ内に格納できない。

30

【 0 0 0 7 】

従って、この種のコンフィギュレーション情報を中央リポジトリに格納することによって、クライアント・コンフィギュレーションの分散管理をサポートするシステムを有することが望ましい。これは、ネットワーク管理者がサブシステムのコンフィギュレーションをサーバから管理することと、アプリケーションに対する全ての変更をサーバから伝播することを可能にする。ネットワーク・ユーザが異なる複数のクライアントへそれぞれログオンでき、かつ該ユーザのパーソナル・プリファレンス及びプロフィールへネットワーク上の任意のクライアント上でアクセスできることが望ましい。更に、効率的で完全に機能するやり方で、データをサーバからクライアントへ転送することを可能にする方法を有することが望ましい。更に、クライアントが重複するデータを受信することがないように凝縮されたコンパクトなフォームのデータを、クライアントへ送信することが望ましい。

40

【 0 0 0 8 】

50

【発明の概要】

本発明の目的を達成すべく、ネットワーク内の複数のコンピュータ間におけるデータの格納及び交換を実施するためのデータ・フレームワーク及び関連するプロトコルを形成する方法、装置及びコンピュータ読み取り可能媒体を開示する。本発明の1つの態様に基づき、ルート・ノード層と、中間ノード層と、コンフィギュレーション・データを格納するデータ層とを含む n 分岐ツリー構造を有するデータ・スキーマを開示する。中間ノード層はコンピュータ・ネットワークのコンポーネント及び様々な特徴に関するカテゴリカル情報を含む多数のノードを有する。ツリー構造において、各中間ノード及びルート・ノードは、その下に連なる複数のノードへの分岐を有する。これらの下位ノードは子ノードと称される。データ・ノード層はツリーの最下位に位置し、かつコンピュータ・ネットワークのコンポーネント及び他の特徴に関する実際の固有コンフィギュレーション・データを含む。複数の中間ノード及びデータ・ノードのうちの特定の一部は永続メモリ領域を形成している。永続メモリ領域を形成しているデータ・ノード内の実際の固有コンフィギュレーション・データは、クライアント・コンピュータまたはサーバ・コンピュータ上で変更され、かつサーバ・コンピュータ上に格納される。これによって、関連する固有情報は不揮発性となり、かつ多数のクライアント・コンピュータによるアクセスが可能である。

10

【0009】

本発明の1つの実施形態では、データ・スキーマ、即ち、データ・フレームワークはクライアント・コンピュータ上に存在するクライアント・スキーマ・サブコンポーネントと、1つ以上のサーバ・コンピュータ上に存在するサーバ・スキーマ・サブコンポーネントとを含む。別の実施形態では、ユーザ・プリファレンス（ユーザ設定）及びプロフィールをサーバ・スキーマ内へ格納する。これによって、ネットワーク上のユーザはネットワーク上の複数のクライアント・コンピュータのうちの任意の1つを使用し、かつユーザ自身のプリファレンスへアクセスすることが可能になる。更に、クライアント・コンピュータ・プラットフォーム及びプロフィールの情報はサーバ・スキーマ内に格納される。更に別の実施形態では、前記の永続メモリ領域は多数のデータ・エントリを含み、各データ・エントリは名前と、関連するノードのリストと、プロパティ名と、関連するプロパティ値とを有する。

20

【0010】

本発明の別の態様では、サーバ・コンピュータを有するコンピュータ・ネットワーク内のクライアント・コンピュータに関する情報をアレンジし、かつ格納するためのデータ・スキーマを開示する。データ・スキーマはルート・ノードと、中間ノードを有する多数の中間ノード・レベルとを含む。各中間ノードはコンピュータ・ネットワーク及びクライアント・コンピュータに関するカテゴリカル情報を示す、即ち、格納している。更に、スキーマは中間ノード内のカテゴリカル情報に対応する実際のコンフィギュレーション情報を含む。クライアント・コンピュータを起動し、かつネットワークへ接続した際に決定された様々なプロフィール及びプリファレンスに対するクライアント・コンピュータの適合を、クライアント・コンピュータ上に存在するカテゴリカル情報及びコンフィギュレーション情報は可能にする。

30

【0011】

1つの実施形態では、多数の中間ノード・レベルは複数のトップ中間ノードを有するトップ中間ノード・レベルを含む。各トップ中間ノードは多数のデータ・エントリを有するデータスペースを形成している。別の実施形態では、複数のデータスペースのうちのソフトウェア・データスペースと称されるデータスペースは永続データスペースである。クライアント・コンピュータを切った際、即ち、機能停止した際、この永続データスペースはデータスペース内のデータ・エントリの保存を可能にする。

40

【0012】

本発明の別の態様では、データ・スキーマを格納するサーバ・コンピュータを有するコンピュータ・ネットワーク内の多数のクライアント・コンピュータに関するコンフィギュレーション情報をアレンジし、かつ格納するためのデータ・スキーマを開示する。このデー

50

タ・スキーマはルート・ノードと、複数の中間ノードを有する多数の中間ノード・レベルとを含む。各中間ノードはコンピュータ・ネットワークと、このコンピュータ・ネットワーク上のクライアント・コンピュータとに関するカテゴリカル情報を示す、即ち、格納する。更に、データ・スキーマのデータ・レベル内に配置された中間ノード内のカテゴリカル情報に対応する実際のコンフィギュレーション情報をデータ・スキーマはさらに含む。カテゴリカル情報及びコンフィギュレーション情報は、サーバ・コンピュータ上に存在し、これによって、クライアント・コンピュータからのリクエストを受けた際、サーバ・コンピュータがクライアント・コンピュータ・プロフィール及びプリファレンスを伝播することを可能にする。

【 0 0 1 3 】

10

1つの実施形態では、多数の中間ノード・レベルは複数のトップ中間ノードを有するトップ中間ノード・レベルを含む。各トップ中間ノードは多数のデータ・エントリを有するデータスペースを形成している。別の実施形態では、2つのデータスペース、即ち、マシン・データスペース及びユーザ・データスペースが存在し、これら2つのデータスペースは永続データスペースである。クライアント・コンピュータを切った際、即ち、機能停止した際、これらの永続データスペースはその中に含まれるデータ・エントリの保存を可能にする。別の実施形態では、マシン・データスペースはネットワーク上に存在する多数のコンピュータ・タイプに関するコンピュータ・コンフィギュレーション・データを格納する。そして、ユーザ・データスペースはコンピュータ・ネットワークへアクセスするために登録した多数のユーザに関するユーザ・コンフィギュレーション・データを格納する。

20

【 0 0 1 4 】

本発明の別の態様では、コンピュータ・ネットワーク上で伝送するデータを準備するための方法、装置及びコンピュータ読み取り可能媒体を提供する。クライアント・コンピュータのプロフィール及びプラットフォームに関する値を含むデータ・セットを、サーバ・コンピュータ上で検索する。更に、クライアント・コンピュータを使用するユーザのユーザ・プリファレンス及びユーザ・グループに関する値を含む別のデータ・セットを、サーバ・コンピュータ上で検索する。これら2つのデータ・セットは特定のクライアント・コンピュータへログオンするユーザに対応している点で互に関連している。2つのデータ・セット内のデータ・アイテムをプロパティ毎に関連付けすることによって、データ・アイテム・コレスポンドンス (data item correspondence) が形成される。複数のデータ・カテゴリー間における優先順位のリストを含む一連のルールを、データ・アイテム・コレスポンドンスへ適用する。ファイナル・データ・セットは前記の一連のルールをデータ・アイテム・コレスポンドンスへ適用することによって得られ、かつネットワークを通じて目的のコンピュータへ送信される。ファイナル・データ・セットをネットワークを通じて送信することによって、送信するデータ量を最初の2つのデータ・セットをそのまま全て送信した場合のデータ量よりも低減できる。

30

【 0 0 1 5 】

1つの実施形態では、接続がクライアント・コンピュータ及びサーバ・コンピュータの間で確立される。クライアント・プロフィールはクライアント・コンピュータからサーバ・コンピュータへ送信される。別の実施形態では、前記の一連のルールは、プリファレンスの順位を含む。最初の2つのデータ・セットにそれぞれ含まれるデータ・アイテムの中からファイナル・データ・セット内へ含めるデータ・アイテムを決定するために、この順位は使用される。別の実施形態では、クライアント・コンピュータはネットワーク・コンピュータである。

40

【 0 0 1 6 】

本発明の更に別の態様では、サーバ・コンピュータを有するネットワーク内のコンピュータのコンフィギュレーションを設定するための方法、装置及びコンピュータ読み取り可能媒体を提供する。プロフィールを有するコンピュータと、サーバ・コンピュータとの間の接続を確立する。前記のコンピュータのプロフィールをサーバへ送信する。次いで、サーバのコントロール下で格納されたコンフィギュレーション情報を、サーバは前記のプロフ

50

ィールに基づいて検索する。サーバ上で検索され、かつコンピュータへ送信されたコンフィギュレーション情報に基づいて、コンピュータのコンフィギュレーションを設定する。これによって、サーバ・コンピュータによるコンフィギュレーション情報の中央集中管理を促進する。

【 0 0 1 7 】

1つの実施形態では、プロフィールはクライアント・コンピュータに関するハードウェア情報と、そのクライアント・コンピュータのユーザに関する情報とを含む。別の実施形態では、クライアント・コンピュータ及びサーバ・コンピュータは、これら2つのコンピュータ間でのデータ交換に使用するソフトウェアのバージョン情報を交換する。使用するソフトウェアの正確なバージョンはサーバ・コンピュータによって決定される。更に別の実施形態では、クライアント・コンピュータはネットワーク・コンピュータである。更に別の実施形態では、サーバ・コンピュータはネットワーク上の多数のクライアント・コンピュータと、クライアント・コンピュータを使用するために登録したユーザとに関するコンフィギュレーション情報を格納する。

【 0 0 1 8 】

本発明とその更なる効果は添付図面に基づく以下の説明から最もよく理解できる。

【発明の実施の形態】

本発明の好ましい実施形態を以下に詳述する。好ましい実施形態の一例を添付図面に示す。本発明を好ましい実施形態に関連して詳述するが、これは本発明を1つの好ましい実施形態に制限することを意図するものではない。逆に、請求の範囲に定義する本発明の趣旨及び範囲に含まれる変形例、改良例及び等価例を包含することを意図する。

【 0 0 1 9 】

データ・フレームワーク、即ち、データ・スキーマと、このデータ・スキーマ内のデータをコンピュータ・ネットワーク内の複数のコンピュータ間で交換するための関連するプロトコルとを複数の図面に示す。コンフィギュレーション情報及び関連情報をシステム・データベース内で表示し、かつ格納するための階層構造、即ち、データ・スキーマを、本発明は開示する。本発明の1つの実施形態を示す目的で、ジャバ・システム・データベース (Java system database、略してJSD) を検討する。別の好ましい実施形態では、このシステム・データベースは別の種類のプラットフォーム上でのオペレーションが可能である。本実施形態のJSDは、少なくとも2つの主要サブコンポーネント、即ち、クライアント・スキーマ及びサーバ・スキーマと称されるサブスキーマを含む単一サブシステムである。本実施形態では、クライアントに関するデータは、クライアント・メモリ内に存在するクライアント・スキーマに格納される。各クライアントのコンフィギュレーション・データはネットワーク・サーバ上に存在するサーバ・スキーマに格納される。コンフィギュレーション・データはクライアント/サーバ・プロトコルを通じて2つのスキーマ、即ち、2つの階層構造間で交換される。正確な情報をサーバ・スキーマから取り出し、かつクライアント・マシン上のクライアント・スキーマへ転送、即ち、入力することと、逆に、情報をクライアント・スキーマからサーバ・スキーマへ転送することを、クライアント/サーバ・プロトコルは保証する。サブシステムとも称される各クライアントのコンフィギュレーション情報はサーバ・スキーマに格納される。これはクライアントに関するコンフィギュレーション情報をクライアント・マシン上でハードコードする、即ち、格納する従来のネットワークとは対照的である。ネットワーク管理者がネットワーク内の各コンピュータのコンフィギュレーション情報を単一サーバなどの中央リポジトリから管理することを、本発明のデータ・スキーマは可能にする。従って、サブシステム・コンフィギュレーションに関する知識及びサブシステム・コンフィギュレーションへのアクセスを必要とする任意のソフトウェアの更新、バージョン・アップグレードまたは新しいアプリケーションのインストールを、中央リポジトリから行い、かつ各クライアントへ伝播できる。アプリケーションの新しいアップグレードまたは新しいバージョンのインストールまたは伝播を目的としたメインテナンスを実施するために、クライアント・マシン上のユーザはアプリケーションを終了する必要があるというえ、ネットワークをダウンさせる必要もない。

【 0 0 2 0 】

図 1 はコンピュータ・ネットワーク・コンフィギュレーションのコンポーネントを示すブロック図であり、本発明の 1 つの実施形態に基づくシステムワイド・データ・スキーマを示す。本実施形態では、システムワイド・データ・スキーマを、ネットワーク 1 0 7 の一部を構成するクライアント・マシン 1 0 5 上に存在するクライアント・スキーマ 1 0 3 からなるジャバ・システム・データベース J S D (1 0 1) として説明する。サーバ・スキーマ 1 1 1 はネットワーク 1 0 7 の一部を構成するサーバ・コンピュータ 1 0 9 上に存在する。

【 0 0 2 1 】

図 2 は本発明の 1 つの実施形態に基づくクライアント・スキーマ階層構造 1 0 3 を表す n 分岐ツリー構造を示す。クライアント・スキーマ階層構造 1 0 3 及びサーバ・スキーマ階層構造 1 1 1 は n 分岐ツリーを用いて示す。ルート・エントリ 2 0 1 がツリーのルートに設けられている。ルート・エントリ 2 0 1 はデータを全く含んでいない。更に、この階層構造内では、ルート・エントリ 2 0 1 は自身を参照する唯一のノード・フレームワークである。クライアント・スキーマ 1 0 3 内のノードの第 1 レベル 2 0 3 に属する一連のノードは、汎用クライアント・スキーマ内のネームスペースをそれぞれ定義している。階層構造内の第 1 レベル 2 0 3 はルート・エントリ 2 0 1 の直下にあり、かつ複数のネームスペース・エントリを有する。

【 0 0 2 2 】

本実施形態では、6 つのネームスペースが汎用クライアント・スキーマ 1 0 3 内に存在する。別の好ましい実施形態では、特定のネットワーク・コンフィギュレーションの要求に基づいて、これより更に多い数または更に少ない数のネームスペースを設け得る。本実施形態では、J S D クライアントの標準トップレベル・ネームスペースは、ソフトウェア (SOFTWARE) 、デバイス (DEVICE) 、インターフェース (INTERFACE) 、ソフトウェア・コンフィギュレーション (SOFTWARE CONFIGURATION) 、エイリアス (ALIAS) 及びテンプレ (TEMP.) である。例えば、ソフトウェア・ネームスペースはノード 2 0 5 に始まり、かつこのノード 2 0 5 から分岐する全てのノード及びデータを含む。階層構造の層 2 0 3 に属する固有エントリは固有ネームスペースを形成するサブツリーのルートである。ソフトウェアなどの 1 つのネームスペースに属する全てのエントリは、クライアント 1 0 5 のソフトウェア・アプリケーションのコンフィギュレーション・データに関するエントリである。本発明のデータ・スキーマに属する各エントリは固有名称と、子 (任意のエントリの真下のエントリ) のリストと、一連の組 (a set of tuples) とからなる。各組はプロパティ名及び関連するプロパティ値を含む。例えば、ワードプロセッシング・プログラムでは、プロパティ名を“フォント”とし、プロパティ値をパレンティノ (Palentino) とし得る。同様に、デバイス・ネームスペース 2 0 7 の下に位置する全てのエントリは、クライアント・スキーマ 1 0 3 を有するクライアント・マシン 1 0 5 のコンフィギュレーション情報に関するエントリである。階層構造内の各エントリはサブツリー内のエントリとしても機能するとともに、下位エントリ、即ち、子ノードを有するサブツリーのルートとしても機能する。層 2 0 3 内の各ネームスペースは本願に対応する米国出願と同じ日に仮出願され、かつ同じ譲受人へ譲渡されたジャバ・システム・データベース (JAVA SYSTEM DATABASE) と称される出願に開示されており、この出願の内容は本開示をもって本明細書中に開示したものとする。

【 0 0 2 3 】

図 2 に示すように、ツリー内の各エントリは単一の親を有し、かつ幾つかの子ノードを有する。ソフトウェア・ネームスペース 2 0 9 などのネームスペースは、クライアント 1 0 5 などの特定のクライアントのソフトウェア上のコンフィギュレーション・データに関する複数のエントリを含む特別に指定されたサブツリーである。図 2 に示すように、本実施形態では、各ネームスペースはスーパー・ルートとも称されるルート・エントリの直下に常に位置する。別の好ましい実施形態では、ネームスペースを階層構造の別のレベルで形成可能であり、ネームスペースをルート 2 0 1 の直下に配置する必要はない。J S D スキ

10

20

30

40

50

ーマ・クライアントの標準ネームスペースはクライアント・コンピュータのスタートアップ、即ち、ブーティング・プロシージャ中に形成される。本実施形態の各ネームスペースはジャバ・プラットフォームの全てのインプリメンテーションで使用可能である。6つのネームスペースはジャバ・プラットフォームが初期化する周知のネームスペースである。他の動的に構築されたネームスペースを標準データベース・ネームスペースへ初期化後に追加しても良い。

【0024】

各ネームスペースはデフォルト・ネームスペース・マネージャによって管理される。ネームスペース・マネージャはエントリをネームスペース内で格納する方法及びこのエントリへアクセスする方法をコントロールする。更に、ネームスペース・マネージャはネームスペース内の任意のエントリのセキュリティ属性、記憶装置属性及びオーナーシップ属性をエクスポートする標準インターフェースを提供する。

10

【0025】

例えば、ソフトウェア・ネームスペース209は、インストール済みシステム・サービス及び/または使用可能システム・サービスのリストを有する。このシステム・サービスの例としては、デバイス・ドライバ、ユーザー・アプリケーション及びユーザー・コンフィギュレーション情報が挙げられる。サーバがソフトウェア・ネームスペース内の全てのエントリに対するバックアップ・ストアを提供するため、ソフトウェア・ネームスペースはクライアント・スキーマ内の唯一の永続ネームスペースである。一時ネームスペースとは逆の意味を有する永続ネームスペース、即ち、永続エントリは永続記憶場所への保存を要するエントリである。永続エントリの例としては、永続記憶装置への格納を要するユーザー環境に関するコンフィギュレーション情報が挙げられる。ユーザがログオンする際、環境を再設定する手間を省くために、ユーザが前回保存した環境を検索する必要がある。永続エントリは永久記憶場所への保存及び永久記憶場所からの検索が可能なエントリである。ネームスペース形成時に、永続ネームスペース及び一時ネームスペースは静的に分離される。永続エントリが一時ネームスペース内に存在することと、一時エントリが永続ネームスペース内に存在することの少なくともいずれか一方は許されない。本実施形態では、永続エントリはリモートJSDサーバに格納される。本実施形態では、4つのカテゴリ、即ち、アプリケーション(application)、システム(system)、サービス(service)及びパブリック(public)がソフトウェア・ネームスペースの下に存在する。ジャバ・プラットフォームを使用する本実施形態では、ソフトウェア・ネームスペースに含まれる幾つかのエントリはジャバ固有の命名法を使用して形成され、ジャバとは無関係な他のエントリは特定のアプリケーションに基づく命名法を使用して形成される。本実施形態では、アイ・ビー・エム(IMB)、サン(Sun)またはロータス(Lotus)などの企業名はそれぞれcom.ibm、com.Sun及びcom.Lotusなどの名前を与えられている。これらの企業名は企業固有情報を特徴づける。企業エントリの下にエントリ名は企業固有のものである。

20

30

【0026】

前記のように、本実施形態では、ソフトウェア・ネームスペース209は6つのネームスペースのうちの永続記憶を有する唯一のネームスペースである。そして、デバイス・ネームスペース207等のその他のネームスペースは一時記憶を有する。クライアント・コンピュータの電源を切ると、これらのネームスペース内のエントリは消失する。これは本実施形態において事実であり、この理由としては、これら5つの一時ネームスペースがクライアント・コンピュータ固有のデータを格納する点が挙げられる。本実施形態では、ソフトウェア・ネームスペースはコンピュータを切った後での保存を要するアプリケーション・コンフィギュレーション情報を含んでいる。

40

【0027】

4つのカテゴリ、即ち、アプリケーション、システム、サービス及びパブリックがソフトウェア・ネームスペースの下に設けられている。アプリケーション・カテゴリを例にとってみた場合、エントリcom.Netscape213は企業の固有名称(例:ネットスケープ(Netscape))を有し、ネットスケープの1つの製品であるネットスケープ・ナビゲータ(Nets

50

cape Navigator) のエントリが com.Netscape エントリ 3 1 3 の下に存在する。ネットスケープ・ナビゲータに関する企業固有情報 2 1 7 がエントリ 2 1 5 の下に存在する。

【 0 0 2 8 】

エントリ 2 1 9 , 2 2 1 , 2 2 3 は他のベンダのエントリであり、これらのエントリはエントリ 2 1 5 に類似したエントリをそれぞれ有する。本実施形態では、デバイス・ネームスペース 2 2 5 の構造は、クライアント上に存在する入出力バス及びデバイスのうちの幾つかまたは全てを示す。換言するならば、バス及びデバイスの物理的接続性は複数のエントリからなるツリーとして表される。そして、このツリーでは、特定のバスが親であり、複数のリーフ・エントリは複数のデバイス上のコンフィギュレーション・データをそれぞれ含む。

10

【 0 0 2 9 】

ソフトウェア・ネームスペースでは、階層構造のリーフ・ノード・レベルはデータ 2 2 7 を有する。データ 2 2 7 はコンフィギュレーション固有のデータである。そして、データ 2 2 7 は、アプリケーション(例: ネットスケープ・ナビゲータ)が望むリーフ・ノード・レベル内の特定のデータの順位に基づいて構成されている。ワードプロセッシング・アプリケーションの場合、リーフ・ノード・エントリは、フォント、辞書定義及び他のワードプロセッシング・タイプ・コンフィギュレーション・データなどの固有情報を含み得る。

【 0 0 3 0 】

JSD のサーバ・スキーマ・コンポーネント内の複数のネームスペースは永続記憶空間である。即ち、これらのネームスペースはクライアント・コンピュータを切った後でも存続する。本実施形態では、2 つのネームスペース、即ち、マシン(machine)及びユーザ(user)がサーバ・スキーマ内に存在する。図 3 は本発明の 1 つの実施形態に基づく JSD サーバ・スキーマの構造を示すブロック図である。このブロック図は図 1 のサーバ・コンピュータ 1 0 9 及びサーバ・スキーマ 1 1 1 の詳細を示す。ルート・エントリ 3 0 1 (ルート・エントリ 3 0 1 は本実施形態のコンフィギュレーション・ネームスペースも表している)が n 分岐ツリーのトップに位置する。前記のように、2 つのネームスペースがサーバ・スキーマ内に存在する。エリア 3 0 3 はマシン・ノード 3 0 5 を有するマシン・ネームスペースを表す。エリア 3 0 7 はユーザ・ノード 3 0 9 を有するユーザ・ネームスペースを表す。

20

30

【 0 0 3 1 】

本実施形態では、マシン・ネームスペース 3 0 3 は 3 つのカテゴリによって形成されている。別の好ましい実施形態では、ネットワークのプラットフォーム及び要件に基づいて、マシン・ネームスペース 3 0 3 は 3 つより更に多い数または更に少ない数のサブカテゴリを有し得る。3 つのカテゴリ、即ち、サブレンジはプラットフォーム(Platform) 3 1 1、識別子(identifier) 3 1 3 及びプロフィール(profile) 3 1 5 である。サンマイクロシステムズ及び IBM コーポレーションなどの特定のコンピュータ製造業者を指す多数のエントリがプラットフォーム 3 1 1 の下に存在する。これは図 4 に詳細に示す。

【 0 0 3 2 】

図 4 はサーバ・スキーマ 1 1 1 内のマシン・ネームスペース 3 0 3 のツリー構造を示すブロック図である。前記のように、ベンダ固有サブレンジ 4 0 1 , 4 0 3 がプラットフォーム 3 1 1 のカテゴリの下に存在する。このレベルにおけるエントリ数はネットワーク内で使用されているコンピュータの製造業者の数に基づいている。エントリ 4 0 5 , 4 0 7 で示すような多数のエントリが com.Sun などの各製造業者の下に存在する。各エントリは製造業者が製造したコンピュータの特定のモデル、即ち、タイプを示す。例えば、コンピュータ・タイプ J D M 1 が com.Sun の下に存在し、コンピュータ・タイプ N S 1 0 0 0 及び N S 2 0 0 0 が com.IBM の下に存在する。各コンピュータ・タイプ、即ち、モデルの下には、このコンピュータのアプリケーション・コンフィギュレーションを特定するリーフ・ノード 4 0 9 が存在する。タイプ・サブカテゴリにおいて、リーフ・エントリ、即ち、リーフ・ノード内のアプリケーション・コンフィギュレーション情報は、特定のコンピュ

40

50

ータ、即ち、親エントリに示されるコンピュータへ適用可能な全てのコンフィギュレーションを含む。

【0033】

識別子ルート・エントリ313を有する識別子サブカテゴリの下には、ネットワーク107内の各コンピュータの固有識別子411を含むエントリが存在する。本実施形態では、各コンピュータのMACアドレスを固有識別子として使用する。特定のクライアント識別子411の下にデータ413は、このクライアント識別子411に対応する特定のコンピュータ固有のアプリケーション・コンフィギュレーション情報である。識別子411の下のコンフィギュレーション・データ413は特定のユーザがコンフィギュレーションを設定した特定のコンピュータへ適用される。このため、このコンフィギュレーション・データ413はプラットフォーム・カテゴリの下のコンフィギュレーション・データ409と区別し得る。本実施形態では、識別子カテゴリの下の複数の固有識別子411と、プラットフォーム・カテゴリの下の複数のエントリとの間には、相互参照（図示略）が存在する。即ち、特定の識別子から特定の種類のコンピュータへの参照、即ち、リファレンスが存在する。特定の固有識別子が示すプラットフォーム、即ち、コンピュータの種類をサーバが決定することを、これは可能にする。

10

【0034】

プロフィール・ルート・エントリ315を有するプロフィール・カテゴリの下には、ネットワーク内のコンピュータの特定のカテゴリまたは用途を示す複数のエントリが存在する。企業内の部門などを示し得る特定のプロフィールに関するコンフィギュレーション情報はプロフィールのサブカテゴリの下に含まれ得る。これらの例としては、2つのノード415, 417が挙げられ、これら2つのノード415, 417はファイナンス（Finance）及びセールス（Sales）のプロフィールをそれぞれ表す。ファイナンス・ノード415の下には、ファイナンス・プロフィールに関するデータを含むアプリケーション固有データ419が存在する。固有識別子からプラットフォーム・エントリへのリファレンスと同じように、固有識別子からプロフィール・エントリへのリファレンスも存在する（適用可能な場合）。即ち、経理部で使用するコンピュータまたは受付係ターミナルとしてのみ使用するコンピュータなどのように、特定のコンピュータが特定のプロフィールを有する場合、このコンピュータの識別子から適切なプロフィール・エントリへのリファレンスが存在する。

20

30

【0035】

図5は本発明の1つの実施形態に基づくユーザ・ネームスペースを示すブロック図である。本実施形態では、ユーザ・ネームスペース307は2つのカテゴリ、即ち、ユーザ（users）及びグループ（groups）を有する。ユーザ・ノード（users node）317はノード501, 503, 505などのコンピュータ・ネットワーク上の各ユーザの名前を表す。各ユーザのノードの下には、符号507で示すような各ユーザのパーソナル・プリファレンスを含む固有コンフィギュレーション・データが存在する。例えば、ワードプロセッシング・アプリケーションでは、特定のユーザ・プリファレンスはデフォルトのフォント及びドキュメントのサイズであり得る。各ユーザがネットワーク107上の任意のコンピュータを使用し、かつこのコンピュータ上で自身のパーソナル・コンフィギュレーションを使用することを、このカテゴリは可能にする。例えば、このユーザがワードプロセッシング・プログラムを起動した場合、そのユーザのプリファレンスがコンピュータの通常のユーザのデフォルトに代わってデフォルトとなる。ユーザ・ネームスペース内のもう一方のカテゴリはグループ・ノード（groups node）319を有するグループ・カテゴリである。このカテゴリは特定のユーザのグループに関するエントリを含む。グループは様々なカテゴリを含み得る。この例としては、ノード509, 511に示すような企業内の部門、即ち、企業内の各従業員を他の従業員と区別するカテゴリが挙げられる。本実施形態では、ユーザ・カテゴリ317の下に各ユーザ503, 505から1つ以上の特定のグループへの参照ポイントが存在する（適切な場合）。

40

【0036】

50

本発明では、複数の図に示すクライアント/サーバ・プロトコルに基づいて、データはコンピュータ・ネットワーク内のクライアント及びサーバの間で交換される。図1のクライアント105及びサーバ109の間に位置する矢印108で示すこのプロトコルは、クライアント及びサーバの間における接続の確立及びデータの交換を行うための一連のルール及びプロシージャである。本実施形態におけるサーバは様々なプロトコルを使用する外部サービス及び記憶媒体との間におけるデータ交換及びコミュニケーションが可能である。JSDサーバがサポートするプロトコルは、サーバのバックエンドに位置する様々な永続プロトコルである。その一方、永続記憶装置、データ・レジストリ、ファイル・サーバ及びデータ・ディレクトリなどの外部エンティティ及びJSDサーバの間のデータ交換を実施するために、JSDサーバは多数の様々なプロトコルをサポート可能であり、JSDクライアント及びJSDサーバの間のクライアント/サーバ・プロトコルはこれらの様々なプロトコルをサポートする十分な汎用性及び幅を有する。このような状況において、クライアント/サーバ・プロトコルは小さな“フットプリント”を含むべく設計されている。サーバへ接続された外部エンティティとの間における通信に必要な複数のプロトコルを管理する複雑な仕組みを設けることにより、これらのプロトコルを管理する負担を各クライアントへ課することがなくなる。従って、以上詳述したモデルは2本タイヤ・モデルであり、第1タイヤはJSDクライアント及びJSDサーバの間における本発明のクライアント/サーバ・プロトコルであり、第2タイヤはJSDサーバと、様々な外部サービス及びデータ記憶エンティティとがサポートする複数のプロトコルである。

【0037】

図6～図8はクライアント及びサーバの間の接続を確立し、次いで、コンフィギュレーション情報を交換する本発明の1つの実施形態に基づくプロセスを示すフローチャートである。図6のステップ601では、コンピュータ・ネットワーク内のクライアント・コンピュータは、ネットワークとの間のコミュニケーション・リンクの開設を試みる。このステップでは、サーバへ接続するために、クライアントはディスカバリ・パケットをネットワークへ送信する。本実施形態では、このデータ・パケットはクライアントの情報を含むDhcp（ダイナミック・ホスト・コンフィギュレーション・プロトコル）ディスカバリ・パケットである。ステップ903では、ネットワーク・サーバはディスカバリ・パケットを受信する。本実施形態では、クライアント・コンピュータはクライアント・スキーマ103を有し、サーバ・コンピュータは図1において詳述したJSD101を全体で定義するサーバ・スキーマ111を有する。ステップ605では、サーバ・コンピュータはディスカバリ・パケットの送信者のIPアドレス（固有識別子の1つの例）を読み取る。本実施形態では、IPアドレスはデータ・パケットを送信するクライアントの固有識別子として使用される。別の好ましい実施形態では、MAC IDなどの別の識別子を固有識別子として使用できる。

【0038】

ステップ607では、クライアントがサーバとの間の接続を確立したことを告げるパケットを、サーバはクライアントへ送信する。クライアントがサーバと現在通信中であることを、サーバはクライアントへ実質的に伝える。ステップ609では、クライアントはネゴシエーション・データ・パケットをサーバへ送信する。このステップは図9において詳述する。ネゴシエーション・パケットの目的は、クライアント及びサーバ間の通信に使用するソフトウェアのバージョンを確定することにある。ソフトウェアのバージョンを確定した後、クライアントは自身の固有プロフィール情報をステップ611においてサーバへ送信する。多くの例において、一般的に、この固有プロフィール情報はクライアントのPROMへハードコードされている。例えば、クライアントの固有プロフィール情報は、サンJDM1ワークステーションまたはIBM NS1000コンピュータなどのコンピュータのタイプを含み得る。サーバがプロフィール情報をクライアントから受信した後、図3のステップ613において、サーバはこのプロフィールをプラットフォーム・サブカテゴリ311の下のマシン・ネームスペース303内の固有プロフィールとマッチさせることを試みる。サーバ・スキーマのマシン・ネームスペース303はプラットフォームのカテ

ゴリ 3 1 1 を有する。プラットフォームのカテゴリ 3 1 1 はコンピュータ製造業者が製造した特定のタイプのコンピュータの固有プロフィール情報を格納する。図 7 のステップ 6 1 3 では、サーバはクライアントのプロフィールをマシン・ネームスペース内で探索する。ステップ 6 1 5 では、クライアントが固有プロフィールを有するか否かを決定するために、サーバはクライアント固有識別子を使用する。前記のように、マシン・ネームスペース内では、識別子のカテゴリはネットワーク内の全てのクライアントの固有識別子を含む。クライアントが受付係またはセールスなどの固有プロフィールを有する場合、固有識別子から対応するプロフィールへの参照ポイントが存在する。

【 0 0 3 9 】

ステップ 6 1 7 では、サーバはプロフィール・データ（もし存在する場合）と、プラットフォーム・データとを 1 つにまとめる、即ち、結合する。そして、サーバは 1 つにまとめたデータ・エントリ、即ち、合体データ・エントリをクライアントのソフトウェア・ネームスペースへ送信する。プロフィール・データ及びプラットフォーム・データを合体するプロセスは図 1 0 及び図 1 1 に詳細を示す。本実施形態では、コンピュータのプラットフォームに関するデータは、コンピュータ固有のプロフィール内のデータによってオーバーライドされる（このコンピュータのコンピュータ固有プロフィールが存在する場合）。ステップ 6 1 9 では、クライアントは、自身のクライアント・スキーマ 1 0 3 内のソフトウェア・ネームスペース 2 0 9 をサーバから送信された合体データ・エントリで占める。このポイントにおいて、クライアント・コンピュータのブートアップ・ステージは完了する。

【 0 0 4 0 】

ステップ 6 2 1 では、ユーザはクライアント・コンピュータへログオンし、これによって、ログオン・フェイズを開始する。このポイントにおいて、ユーザ名がサーバへ送信される。ステップ 6 2 3 では、サーバ・スキーマ内のユーザ・ネームスペース 3 0 7 内のコンフィギュレーションを検索するために、サーバはユーザ名を受け入れる。前記のように、本実施形態では、サーバ・スキーマ 1 1 1 は 2 つのネームスペース、即ち、マシン・ネームスペース 3 0 3 及びユーザ・ネームスペース 3 0 7 を有する。ユーザ・ネームスペースの下には、2 つのカテゴリ、即ち、ユーザ 3 1 7 及びグループ 3 1 9 が存在する。ステップ 6 2 3 では、サーバはユーザ名を使用してユーザ・コンフィギュレーションを探索する。図 8 のステップ 6 2 5 では、ユーザ・エントリからグループ・カテゴリ内のエントリへの参照ポイントをチェックすることにより、サーバはユーザがグループ・ノード 3 1 9 によって表されるグループに属するか否かを決定する。ステップ 6 2 7 では、ユーザ・コンフィギュレーション・エントリ及びグループ・コンフィギュレーション・エントリは単一ユーザ・エントリへ合体され、かつクライアントへ送信される。本実施形態では、ユーザ・コンフィギュレーション・エントリ内のデータはユーザ・グループ・コンフィギュレーション・エントリ内の対応する全てのデータをオーバーライドする。ユーザ・エントリからのデータ及びグループ・エントリからのデータの合体のプロセスは図 1 0 及び図 1 1 に詳述する。ステップ 6 2 9 では、クライアントはユーザ・エントリを受信し、必要であれば、プラットフォーム / プロフィール・エントリが以前に占めていたクライアント・スキーマ内のソフトウェア・ネームスペース内のデータをオーバーライドする。本実施形態では、ステップ 6 2 7 においてクライアントへ送信されたユーザ・コンフィギュレーション・エントリは、ステップ 6 1 9 で送信された対応する全てのデータをオーバーライドする。本実施形態では、クライアント・スキーマ内のソフトウェア・ネームスペースは永続データを含む唯一のソフトウェア・ネームスペースであるため、このソフトウェア・ネームスペースのみがサーバから送信されたデータ、即ち、サーバ・コンピュータ上の永久記憶装置、即ち、不揮発性記憶装置内に存在するデータによって占められることに注意する必要がある。別の好ましい実施形態では、別のネームスペースはコンピュータ・ネットワークの要求に基づいて永続データを含み得る。ステップ 6 3 1 では、クライアント・コンピュータのユーザは特定のアプリケーションを開始可能である。このステージでは、特定のアプリケーションの適切なコンフィギュレーション・データを獲得するために、クライ

ント・コンピュータのクライアント・スキーマのソフトウェア・ネームスペースを探ることにより、クライアント・コンピュータは前記のアプリケーションを開始可能である。このステージにおいて、アプリケーションをクライアント・コンピュータ上で実行するために必要な全てのコンフィギュレーション・データがサーバからクライアントへ転送されている。更に、クライアント・コンピュータが使用するコンフィギュレーション・データのみが、サーバ上で実施された合体の結果としてサーバからクライアントへ転送される。

【0041】

図9はクライアント及びサーバの間のコミュニケーション・リンクをネゴシエート、即ち、折衝するプロセスを示す本発明の1つの実施形態に基づくフローチャートである。このフローチャートは図6のステップ609の詳細を示す。ステップ701では、クライアントは自身のソフトウェアのバージョン情報をサーバへ通知する。このバージョン情報はクライアントのハードウェア・コンポーネントをさらに示し得る。このデータはTCP/IPを通じてデータ・パケットとして送信される。このステップにおいて、クライアントは自身が使用するオペレーティング・ソフトウェアのバージョンをサーバへ単に通知する。ステップ703では、サーバはこのバージョン情報を調べ、次いで、サーバがサポートし得るオペレーティング・ソフトウェアのバージョンをクライアントへ通知する。これを行うことにより、コンフィギュレーション・データをクライアント及びサーバの間で交換するために使用するオペレーティング・ソフトウェアのバージョンをサーバは指令する。

【0042】

図10は図6のステップ617、627の詳細を示すフローチャートである。このフローチャートは特定のクライアントに関するサーバ上のコンフィギュレーション・データを合体する本実施形態のプロセスを示す。データ・エントリ内の特定のデータ・アイテムを別のネームスペースまたはカテゴリ内の対応するデータ・アイテムによってオーバーライドし得るため、データは図11に示す階層構造に基づいて合体される。ステップ801では、サーバはプラットフォーム・エントリを自身のマシン・ネームスペース内のプラットフォーム・サブツリーから検索する。プラットフォーム・エントリはクライアント・コンピュータ・タイプに関する固有情報を含む。ステップ803では、クライアント・マシン固有識別子からプロフィール・カテゴリ内の特定のプロフィールへの参照ポイントが存在するか否かをチェックすることにより、サーバはクライアントが対応するプロフィール・エントリを有するか否かを決定する。クライアントがプロフィール・エントリを有する場合、サーバはこのプロフィール・エントリをステップ805で検索する。ステップ805では、このプロフィール・エントリ内の値及びプラットフォーム・エントリ内の値を合体する。これらのプロフィール・エントリ及びプラットフォーム・エントリ内に含まれるプロパティの値は、図11の階層構造に基づいてプロパティ毎に合体される。本実施形態では、プラットフォーム・エントリ内にマッチするプロパティ値を有するプロフィール・エントリ内の値は、プロフィール・エントリ値によってオーバーライドされる。エントリの合体はプロパティ毎に実施される。ステップ803において、クライアントがプロフィール・エントリを有していないことが決定した場合、コントロールはステップ807へ移行する。

【0043】

ステップ807において、サーバはユーザ・エントリをサーバ・スキーマ内のユーザ・ネームスペースから検索する。ステップ809では、ログオンしたユーザがグループ319のいずれかに属するか否かを、サーバは決定する。ステップ811では、サーバはグループ・エントリを検索する。しかし、ここでもサーバはグループ・エントリ内の値をユーザ・エントリ内の値によってプロパティ毎にオーバーライドする。ステップ813では、サーバはステップ805からのプロフィール／プラットフォーム・エントリ内の値を合体ユーザ・エントリ内の値によってオーバーライドする。ユーザがグループに属さないことをサーバがステップ809で決定した場合、ステップ813へ移行する。ステップ815では、クライアントへログオンするユーザに関する合体データ・エントリはクライアント・コンピュータへ送信され、プロセスは完了する。

【 0 0 4 4 】

図 1 1 はデータを合体するために使用する本発明の 1 つの実施形態に基づく階層構造を示すブロック図である。最上位のランクに位置し、これによって、他の全てのエントリに優先するエントリは、ブロック 8 1 7 に示すユーザ・エントリである。このエントリの下には、次にランクの高いエントリ、即ち、ブロック 8 1 9 で示すユーザ・グループ・エントリが存在する。グループ・エントリに含まれるプロパティ内の値はユーザ・エントリ内のマッチするプロパティの値によってオーバーライドされる。ブロック 8 2 1 はクライアント・プロフィール・エントリである。グループ・エントリ内のプロパティの値はクライアント・プロフィール・エントリ内のマッチするプロパティ値をオーバーライドする。本実施形態の階層構造の最下位には、ブロック 8 2 3 に示すクライアント・プラットフォーム・エントリが存在する。このエントリ内の値は他の 3 つのカテゴリのうちのいずれか 1 つに含まれるマッチするプロパティの値によってオーバーライドされる。別の好ましい実施形態では、優先順位を変更可能であり、前記の実施形態のエントリの数より更に多い数または更に少ない数のエントリを階層構造内に含み得る。

10

【 0 0 4 5 】

クライアント/サーバ・プロトコルの 1 つの特徴は、ハートビート・メカニズムと称される。クライアント及びサーバの間の接続が切断または中断された際、クライアントまたはサーバが通知を受けることを、ハートビート・メカニズムは可能にする。図 1 2 及び図 1 3 はサーバ及びクライアントの間の接続が依然として生きていることを示すために、信号をサーバからクライアントへ周期的に送信する本発明の 1 つの実施形態に基づくプロセスを示すフローチャートである。一般的に、ハートビート・メカニズムはネゴシエーション・フェイズ(図 9 のステップ 6 0 9)の直後に開始され、かつクライアント及びサーバの間の接続が続く限り継続される。クライアント及びサーバの間の接続が切断または中断された際、クライアントまたはサーバが通知を受けることは重要な特徴である。ステップ 9 0 1 では、サーバはデータ・パケットをクライアントへ送信する。これは一般的にネゴシエーション・フェイズの直後に最初に発生し、接続が続く限り一定のインターバルで送信される。本実施形態では、ハートビート・デーモンはクライアント及びサーバの両方で実行されるとともに、クライアント/サーバ・プロトコル内の実質的に 1 つのスレッドである。別の好ましい実施形態では、ハートビート・デーモンはサーバまたはクライアントで実行可能であるとともに、マルチスレッド・プロセスであり得る。サーバは接続の状態情報を維持し、かつデータ・パケットをクライアントへ送信した時刻等のデータを含む。ステップ 9 0 3 では、クライアントからのリターン・データ・パケット、即ち、ハートビートが存在するか否かを、サーバは決定する。ステップ 9 0 5 では、ハートビートが存在する場合、クライアントからのデータ・パケットはサーバへ送信される。ステップ 9 0 1 において、設定された時間間隔(例: 1 0 分)の後で、サーバはデータ・パケットをクライアントへ再び送信する。別の好ましい実施形態では、時間間隔を、これより更に長くしたり、あるいは、更に短くすることができる。

20

30

【 0 0 4 6 】

クライアントがデータ・パケットをサーバへ送り返さなかったこと、即ち、ハートビートが存在しないことがステップ 9 0 3 で確定した場合、リターン・データ・パケットが存在しないことを表示するために、サーバは状態情報及びストラクチャ(ディレクトリ・キャッシュ)を使用する。これはステップ 9 0 7 で実施される。接続が切断されたことを表示するために、ダーティ・ビットを前記のストラクチャ内に配置する。ステップ 9 0 9 では、サーバ及びクライアントは再接続を試みる。この時点において、クライアント上での更新が必要か否かを確認するために、サーバは自身の状態ストラクチャ内のダーティ・ビットをチェックする。これはクライアントに関するサーバ上のコンフィギュレーション情報が変更された場合に発生し得る。ステップ 9 1 1 では、サーバはクライアント・コンフィギュレーションを更新する。クライアントはサーバ上のディレクトリ・キャッシュ状態ストラクチャに類似する状態情報を示すストラクチャをさらに有する。そして、このクライアントの状態ストラクチャは更新をクライアントからサーバへ送信すべきか否かを示すた

40

50

めのダーティ・ビットを含み得る。従って、ステップ913では、更新をサーバへ送信すべきか否かを確認するために、クライアントは自身のダーティ・ビットをチェックする。ステップ915では、クライアントはサーバ・スキーマを更新する。このポイントにおいて、クライアント及びサーバの間の再接続が確立され、ネゴシエーション・フェイズの完了後、ハートビート・メカニズムはオペレーションを再開する。

【0047】

本発明のクライアント/サーバ・プロトコルの別の特徴としては、サーバ上で実施された特定のアプリケーションの更新を、このアプリケーションを使用するクライアントへ通知する能力が挙げられる。図14は本発明の1つの実施形態に基づくクライアント/サーバ・プロトコル内のイベント通知のプロセスを示すフローチャートである。ステップ1301では、サーバ・スキーマをアプリケーションの調整、即ち、変更によって更新する。即ち、サーバ・スキーマを一般コンフィギュレーション更新によって変更する。ステップ1003では、変更されたコンフィギュレーション・データを使用するクライアントを確認するために、即ち、サーバ・スキーマ内で実施された一般コンフィギュレーションの変更の通知を望む可能性のあるクライアントを確認するために、サーバは自身のキャッシュ・ディレクトリを調べる。サーバ・キャッシュ・ディレクトリはサーバに接続されているクライアントが使用するリストである。ステップ1005では、サーバは更新、即ち、変更を適切なクライアントへ斉同報する。ステップ1007では、クライアントは更新の影響を受けるクライアント側のアプリケーションを決定する、即ち、クライアントは一般コンフィギュレーションの更新が自身に関連するか否かを決定する。ステップ1009では、更新の影響を受けるアプリケーションが、全ての変更の通知を受けるための登録を行っているか否かを、クライアントは決定する。アプリケーションが通知を受けるべく登録されている場合、コントロールはステップ1011へ移行する。ステップ1011において、このアプリケーションは更新の通知を受ける。このステージにおいて、プロセスは完了する。アプリケーションが変更の通知を受けるべく登録されていない場合、このアプリケーションは変更、即ち、更新を無視し、通知を受けることなくオペレーションを継続する。

【0048】

本発明のクライアント/サーバ・プロトコルの別の特徴としては、特定のオペレーションの実行または不実行を保証する能力が挙げられる。この特徴はクライアント/サーバ・プロトコルの原子性と称される。更新をクライアント・スキーマ及びサーバ・スキーマ内で行うための2フェイズ・コミット・プロシージャ(two-phase commit procedure)を、クライアント/サーバ・プロトコルは実行する。第1フェイズでは、スキーマのサブツリー内の複数のエントリのうちの1つを更新するために、このサブツリーのルートをロックする。第2フェイズでは、サブツリー内の複数のエントリのうちの1つを更新し、次いで、このサブツリーのルートをアンロックする。サブツリーのルートをロックすることにより、他のクライアントまたはアプリケーションはサブツリー内のルートの下に位置する全てのエントリ及びノードへアクセスできなくなる。ロックされたサブレンジ内のエントリの更新を試みる別のクライアント・アプリケーションはブロック・トランザクションまたはアンブロック・トランザクションを起こす。アンロックの発生を待ち、次いで、自身の更新をクライアント・アプリケーションが望む場合、トランザクションはブロック・トランザクションと称される。このシナリオでは、更新は待ち行列内へ配置される。アプリケーションがロックの除去を待つことを望まない場合、トランザクションはノンブロック・トランザクションと称され、トランザクションはアプリケーションへ戻される。このメカニズムを通じて、クライアント/サーバ・プロトコルは、リクエストされたオペレーション、即ち、更新を実施すること、即ち、トランザクションをブロックすることと、オペレーションを実施しないこと、即ち、トランザクションをブロックしないことのうちのいずれか一方をクライアント・アプリケーションに対し保証できる。

【0049】

前記のように、コンピュータ・システム内に格納されたデータを使用する様々なコンピュ

10

20

30

40

50

ータ実行オペレーションを本発明は用いる。これらのオペレーションは物理量の物理操作を必要とするオペレーションを含む（但し、このオペレーションに限定されない）。一般的に、必ずしも必要でないが、これらの量は格納、転送、結合、比較及び他の操作が可能な電気信号または磁気信号の形態をなす。本発明の一部を構成する本明細書中に開示するオペレーションは、効果的なマシン・オペレーションである。実施する操作は生成（producing）、識別（identifying）、実行（running）、決定（determining）、比較（comparing）、実行（executing）、ダウンロード（downloading）または検出（detecting）等の用語で示されることが多い。特に、共通の用法を確立するために、これらの電気信号または磁気信号をビット、値、エレメント、変数、キャラクターまたはデータ等として示すと都合が良い。しかし、これらの用語またはこれらに類似する用語の全ては適切な物理量に関連すべきであり、かつこれらの物理量に適用された都合の良いラベルにすぎない点を覚えておく必要がある。

10

【 0 0 5 0 】

更に、本発明は前記のオペレーションを実施するためのデバイス、システムまたは装置に関する。システムは要求された目的のために特別に構築可能である。また、システムは汎用コンピュータとすることが可能である。汎用コンピュータに格納されたコンピュータ・プログラムによって、この汎用コンピュータは選択的に作動または設定される。前記の複数のプロセスは特定のコンピュータまたは他のコンピューティング装置に固有のものではない。特に、本明細書の開示内容に基づいて記述されたプログラムを様々な汎用コンピュータと併用可能である。これに代えて、要求されたオペレーションを実施するために、更に特別なコンピュータ・システムを形成することは更に都合が良い。

20

【 0 0 5 1 】

図 1 5 は本発明の 1 つの実施形態に基づく処理の実施に適した汎用コンピュータ・システム 1 1 0 0 のブロック図である。図 1 5 は汎用コンピュータ・システムの 1 つの例を示す。本発明の処理を実施するために、他のコンピュータ・システム・アーキテクチャ及びコンフィギュレーションを使用し得る。以下に詳述する様々なサブシステムからなるコンピュータ・システム 1 1 0 0 は少なくとも 1 つのマイクロプロセッサ・サブシステム（中央処理装置、即ち、CPUとも称される）1 1 0 2 を含む。即ち、CPU 1 1 0 2 はシングルチップ・プロセッサまたは複数のプロセッサによって実現し得る。CPU 1 1 0 2 はコンピュータ・システム 1 1 0 0 のオペレーションを制御する汎用デジタル・プロセッサである。メモリから検索した命令を使用して、CPU 1 1 0 2 は入力データの受信及び操作と、出力デバイス上でのデータの出力及び表示とを制御する。

30

【 0 0 5 2 】

CPU 1 1 0 2 は一般的にランダム・アクセス・メモリ（RAM）からなる第 1 の一次記憶装置 1 1 0 4 へメモリ・バス 1 1 0 8 を通じて双方向接続されている。更に、CPU 1 1 0 2 は一般的にリード・オンリ・メモリ（ROM）からなる第 2 の一次記憶装置 1 1 0 6 へメモリ・バス 1 1 0 8 を通じて単方向接続されている。当該技術分野でよく知られているように、一次記憶装置 1 1 0 4 は汎用記憶領域及び作業メモリとして使用可能であり、さらには入力データ及び処理済みデータを格納するためにも使用できる。更に、CPU 1 1 0 2 上のオペレーションを処理するためのデータ及び命令を格納する以外に、一次記憶装置 1 1 0 4 は例えばスレッド及びプロセスの形態でプログラミング命令及びデータを格納可能である。更に、データ及び命令をメモリ・バス 1 1 0 8 を通じて双方向に高速転送するために、一次記憶装置 1 1 0 4 は一般的に使用される。同様に、当該技術分野でよく知られているように、CPU 1 1 0 2 がその機能を果たすために使用する基本オペレーティング命令、プログラム・コード、データ及びオブジェクトを一次記憶装置 1 1 0 6 は一般的に含む。データ・アクセスが双方向または単方向のいずれを必要とするかなどの条件に基づいて、一次記憶装置 1 1 0 4 , 1 1 0 6 は以下に詳述する任意の適切なコンピュータ読み取り可能記憶媒体を含み得る。CPU 1 1 0 2 は頻繁に必要なデータをキャッシュ・メモリ 1 1 1 0 内へ超高速で直接検索し、かつ格納できる。

40

【 0 0 5 3 】

50

取り外し可能大容量記憶装置 1 1 1 2 はコンピュータ・システム 1 1 0 0 のための別のデータ記憶能力を提供し、かつペリフェラル・バス 1 1 1 4 を通じて CPU 1 1 0 2 へ両方向または単方向に接続されている。例えば、CD-ROM として一般的に知られている特定の取り外し可能大容量記憶装置はデータを単方向で CPU 1 1 0 2 へ送信する。その一方、フロッピー・ディスクはデータを CPU 1 1 0 2 へ双方向に送信し得る。記憶装置 1 1 1 2 は磁気テープ、フラッシュ・メモリ、搬送波に組み込まれた信号、PC カード、ポータブル大容量記憶装置、ホログラフィック記憶装置及び他の記憶装置等のコンピュータ読み取り可能媒体を更に含み得る。固定大容量記憶装置 1 1 1 6 は別のデータ記憶能力を提供し、かつペリフェラル・バス 1 1 1 4 を通じて CPU 1 1 0 2 へ双方向に接続されている。大容量記憶装置 1 1 1 6 の最も一般的な例としては、ハード・ディスク・ドライブ
10
が挙げられる。一般的に、これらの媒体へのアクセスは一次記憶装置 1 1 0 4 , 1 1 0 6 へのアクセスより遅い。CPU 1 1 0 2 が頻繁に使用しない他のプログラミング命令及びデータ等を大容量記憶装置 1 1 1 2 , 1 1 1 6 は一般的に格納する。必要に応じて、大容量記憶装置 1 1 1 2 , 1 1 1 6 内に保持された情報は、一次記憶装置 1 1 0 4 (例: RAM) の一部を構成するバーチャル・メモリとして標準的に組み込み可能である。

【 0 0 5 4 】

記憶装置サブシステムへの CPU 1 1 0 2 のアクセスを提供する以外に、ペリフェラル・バス 1 1 1 4 は他のサブシステム及びデバイスへのアクセスを提供するために使用できる。本実施形態では、これらは、ディスプレイ・モニタ 1 1 1 8 及びアダプタ 1 1 2 0、プリンタ・デバイス 1 1 2 2、ネットワーク・インターフェース 1 1 2 4、補助入出力装置
20
インターフェース 1 1 2 6、サウンド・カード 1 1 2 8 及びスピーカー 1 1 3 0、並びに必要とされる他のサブシステムを含む。

【 0 0 5 5 】

図示するように、ネットワーク接続を使用することにより、ネットワーク・インターフェース 1 1 2 4 は CPU 1 1 0 2 を別のコンピュータ、コンピュータ・ネットワークまたはテレコミュニケーション・ネットワークへ接続可能にする。前記の方法のステップを実行するうえで、CPU 1 1 0 2 はデータ・オブジェクトまたはプログラム命令などの情報を別のネットワークからネットワーク・インターフェース 1 1 2 4 を通じて受信するか、または情報をネットワーク・インターフェース 1 1 2 4 を通じて別のネットワークへ送信し得る。CPU で実行する複数の命令のシーケンスに代表される情報は、搬送波に組み込まれたコンピュータ・データ信号などの形態で別のネットワークに対して送受信可能である。
30
インターフェース・カードまたはこれに類似するデバイスと、CPU 1 1 0 2 によって実行される適切なソフトウェアとは、コンピュータ・システム 1 1 0 0 を外部ネットワークへ接続し、かつデータを標準プロトコルに基づいて転送するために使用できる。即ち、本発明の方法は CPU 1 1 0 2 上で単独で実行し得る。その一方、処理の一部を共有する遠隔 CPU と協働することにより、本発明の方法をインターネット、イントラネットワーク若しくはローカル・エリア・ネットワーク等のネットワークを通じて実行し得る。別の大容量記憶装置 (図示略) をネットワーク・インターフェース 1 1 2 4 を通じて CPU 1 1 0 2 へ接続し得る。

【 0 0 5 6 】

マイクロホン、タッチ・ディスプレイ、トランスデューサ・カード・リーダー、テープ・リーダー、音声認識装置、手書き文字認識装置、バイオメトリクス・リーダー、カメラ、ポータブル大容量記憶装置及び他のコンピュータ等の装置に対する CPU 1 1 0 2 によるデータの送受信を可能にする汎用インターフェース及びカスタム・インターフェースを補助入出力装置インターフェース 1 1 2 6 は表す。
40

【 0 0 5 7 】

キーボード 1 1 3 6 またはポインタ・デバイス 1 1 3 8 からの入力を受信し、さらにはデコードしたシンボルをキーボード 1 1 3 6 またはポインタ・デバイス 1 1 3 8 から CPU 1 1 0 2 へ送信するために、キーボード・コントローラ 1 1 3 2 がローカル・バス 1 1 3 4 を通じて CPU 1 1 0 2 へ接続されている。ポインタ・デバイスはマウス、スタイラス
50

、トラック・ボールまたはタブレットであり得る。そして、ポインタ・デバイスはグラフィカル・ユーザー・インターフェースとのインターフェースに効果的である。

【 0 0 5 8 】

更に、本発明の実施形態は様々なコンピュータ実行オペレーションを実施するためのプログラム・コードを含むコンピュータ読み取り可能媒体を有するコンピュータ記憶装置に関する。コンピュータ読み取り可能媒体は、コンピュータ・システムによる後からの読み取りが可能なデータを格納し得る任意のデータ記憶装置である。媒体及びプログラム・コードは本発明の目的のために特別に設計され、かつ構築されたものであるか、またはコンピュータ・ソフトウェア技術分野の当業者によく知られたものであり得る。コンピュータ読み取り可能媒体の例としては、ハード・ディスク、フロッピー・ディスク及び磁気テープなどの磁気媒体と、CD-ROMディスクなどの光媒体と、光フロッピー・ディスクなどの磁気光媒体と、特定用途向け集積回路(AASIC)、プログラム可能論理回路(PLD)、ROMデバイス及びRAMデバイスなどの特別に構成されたハードウェア・デバイスとを含めた前記の全ての媒体が挙げられる(但し、これらに限定されない)。コンピュータ読み取り可能媒体は搬送波に組み込まれたデータ信号として結合コンピュータ・システムのネットワーク上に分散させ得る。従って、コンピュータ読み取り可能コードは分散した形態で格納及び実行できる。プログラム・コードの例としては、コンパイラなどによって形成されたマシン・コード、またはインタプリタを使用して実行できる高レベル・コードを含むファイルが挙げられる。

【 0 0 5 9 】

前記のハードウェア・エレメント及びソフトウェア・エレメントが標準的なデザイン及び構成を有することを当業者は認める。本発明に適した他のコンピュータ・システムは別のサブシステムまたは更に少ない数のサブシステムを含み得る。更に、メモリ・バス1108、ペリフェラル・バス1114及びローカル・バス1134は複数のサブシステムをリンクするために使用する任意の相互接続方式の実例である。例えば、ローカル・バスはCPUを固定大容量記憶装置1116及びディスプレイ・アダプタ1120へ接続するために使用可能である。図15に示すコンピュータ・システムは本発明に適したコンピュータ・システムの例である。本発明のクライアント・コンピュータまたはサーバ・コンピュータを実現するために、サブシステムの別のコンフィギュレーションを有する他のコンピュータ・アーキテクチャを使用し得る。本発明の別の好ましい実施形態では、クライアント・コンピュータはネットワーク・コンピュータ、即ち、NCである。機能性及び記憶能力の点において、このコンピュータは独立型コンピュータとして機能し得る完全に独立したファット・クライアント・コンピュータ("fat" client computer)と、サーバ・コンピュータ、即ち、メインフレーム・コンピュータにほぼ完全に依存するダム・クライアントとの間に位置する。更に別の好ましい実施形態では、パーソナル・デジタル・アシスタント(PDA)及びハンドヘルド・コンピュータなどの限られたメモリ・ストレージを有するコンピュータ以外に、クライアント・スキーマはジャバ・プラットフォームを実行可能なスマート・カード及び他のスマート・アプライアンスなどの非コンピュータ・デバイス上に存在し得る。

【 0 0 6 0 】

以上、理解を容易にする目的で、本発明をある程度詳しく説明したが、特定の変更及び改良を本発明の請求の範囲を逸脱することなく実施しても良い。更に、本発明のプロセス及び装置の両方を実現する他の方法があることを認識する必要がある。従って、本明細書に開示した複数の実施形態は例示目的であって、限定目的ではない。更に、本発明は本明細書に開示する詳細部分に限定されることなく、請求の範囲及びそれに等価な範囲内で変更し得る。

【図面の簡単な説明】

【図1】コンピュータ・ネットワーク・コンフィギュレーションのコンポーネントを表すブロック図であり、本発明の1つの実施形態に基づくシステムワイド・データ・スキーマを示す。

【図 2】本発明の 1 つの実施形態に基づくクライアント・スキーマ階層構造を示す n 分岐ツリー構造を表す図である。

【図 3】本発明の 1 つの実施形態に基づく J S D サーバ・スキーマの構造を示すブロック図である。

【図 4】サーバ・スキーマ内のマシン・ネームスペースのツリー構造を示すブロック図である。

【図 5】本発明の 1 つの実施形態に基づくユーザ・ネームスペースを示すブロック図である。

【図 6】クライアント及びサーバの間の接続を確立し、かつコンフィギュレーション情報を交換する本発明の 1 つの実施形態に基づくプロセスを示すフローチャートである。

10

【図 7】クライアント及びサーバの間の接続を確立し、かつコンフィギュレーション情報を交換する本発明の 1 つの実施形態に基づくプロセスを示すフローチャートである。

【図 8】クライアント及びサーバの間の接続を確立し、かつコンフィギュレーション情報を交換する本発明の 1 つの実施形態に基づくプロセスを示すフローチャートである。

【図 9】本発明の 1 つの実施形態に基づくクライアント及びサーバの間のコミュニケーション・リンクをネゴシエートするプロセスを示すフローチャートである。

【図 10】図 7 および図 8 のステップ 6 1 7 , 6 2 7 の詳細を示すフローチャートである。

【図 11】データを合体するために使用する本発明の 1 つの実施形態に基づく階層構造を示すブロック図である。

20

【図 12】サーバ及びクライアントの間の接続が依然として生きていることを示すために、信号をサーバからクライアントへ周期的に送信する本発明の 1 つの実施形態に基づくプロセスを示すフローチャートである。

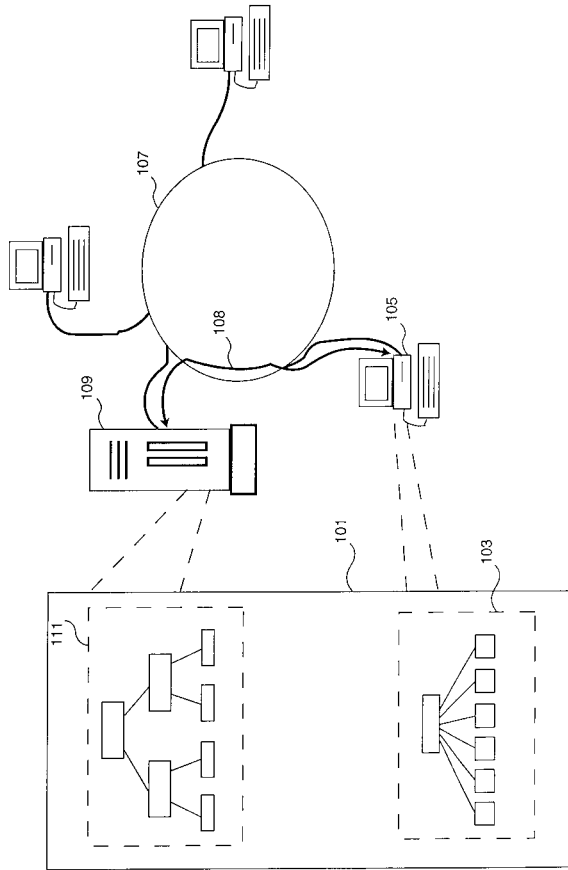
【図 13】サーバ及びクライアントの間の接続が依然として生きていることを示すために、信号をサーバからクライアントへ周期的に送信する本発明の 1 つの実施形態に基づくプロセスを示すフローチャートである。

【図 14】本発明の 1 つの実施形態に基づくクライアント / サーバ・プロトコル内のイベント通知のプロセスのフローチャートである。

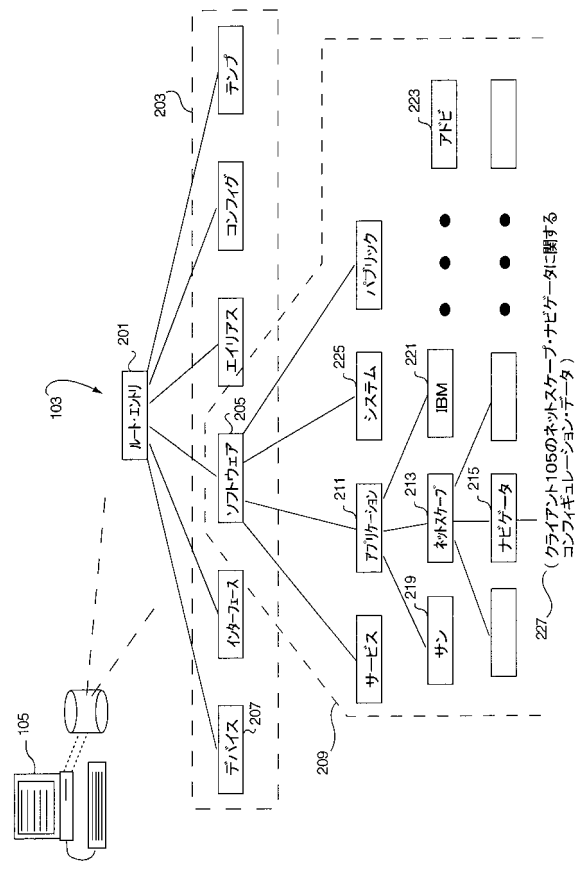
【図 15】本発明の実施形態の実現に適する一般的なコンピュータ・システムのブロック図である。

30

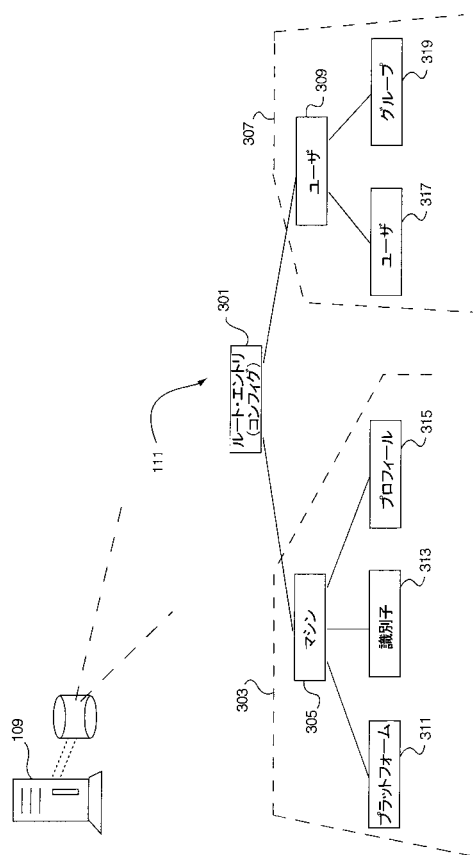
【図 1】



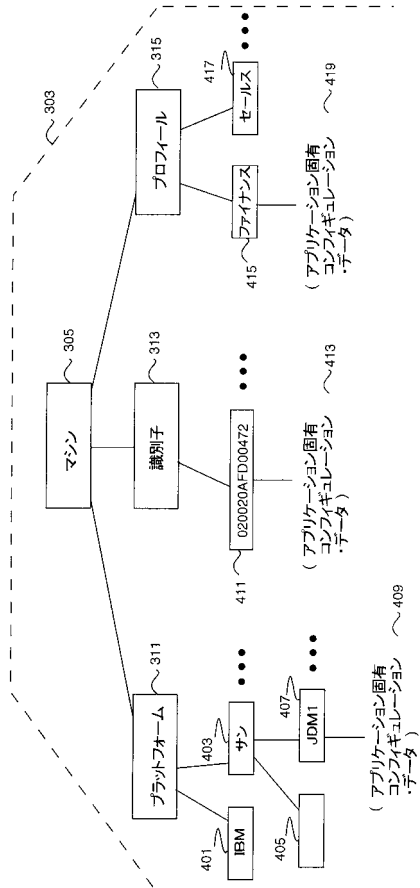
【図 2】



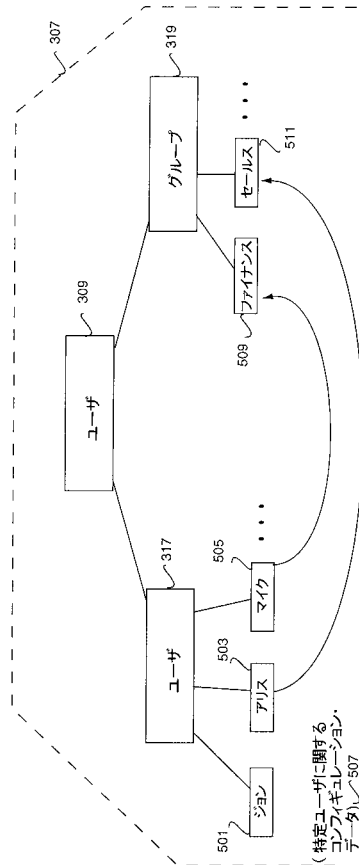
【図 3】



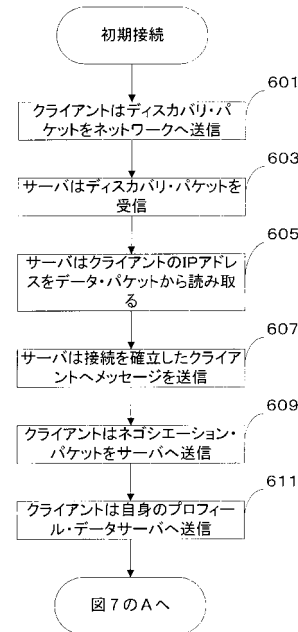
【図 4】



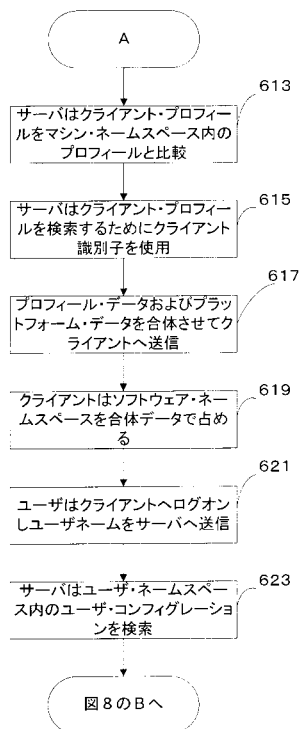
【図 5】



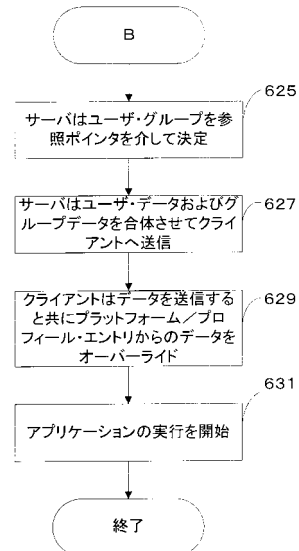
【図 6】



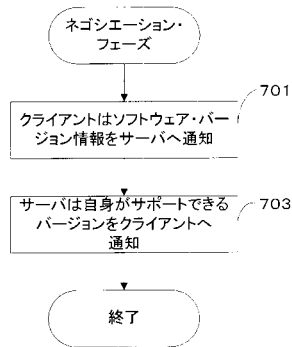
【図 7】



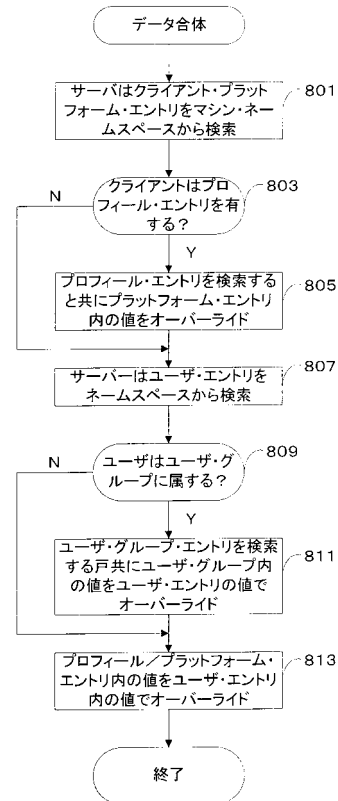
【図 8】



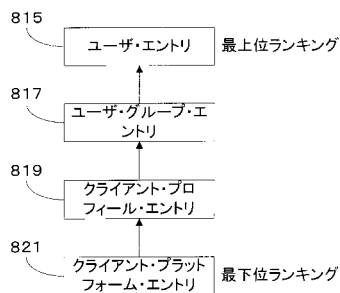
【図 9】



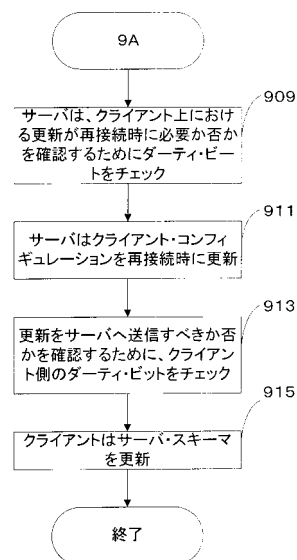
【図 10】



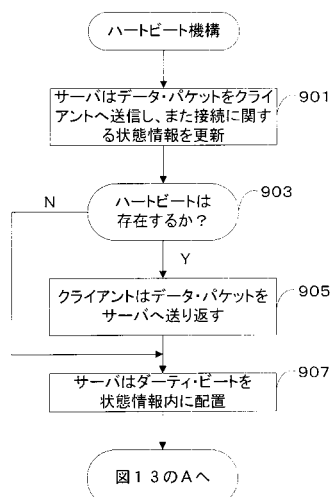
【図 11】



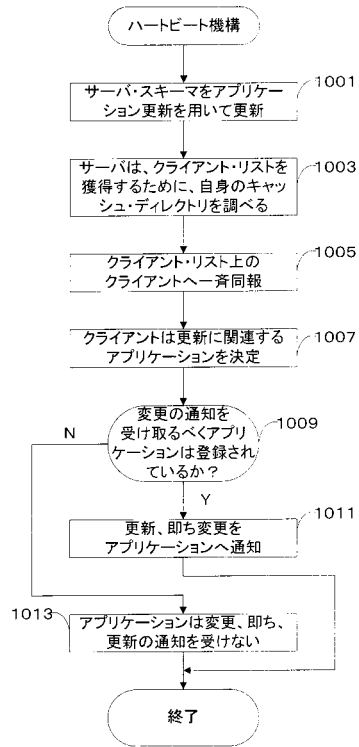
【図 13】



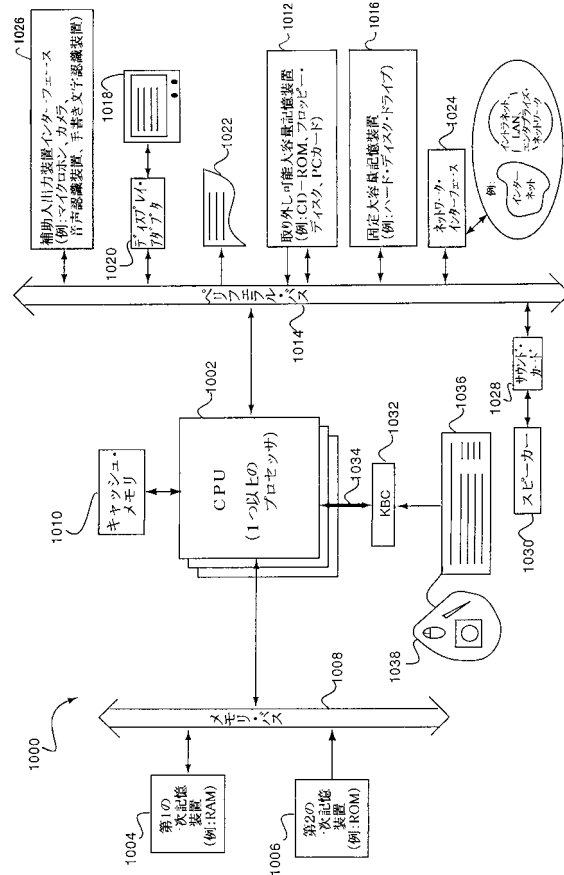
【図 12】



【 図 1 4 】



【 図 1 5 】



フロントページの続き

(74)復代理人 100102989

弁理士 井上 佳知

(72)発明者 バーナード・エー．・トラバサット

アメリカ合衆国 カリフォルニア州 9 4 1 0 9 サン・フランシスコ, カリフォルニア・ストリート, 2 0 5 5 アpartment 4 0 2

(72)発明者 トム・サウルポー

アメリカ合衆国 カリフォルニア州 9 5 1 2 0 サン・ホセ, プレット・ハーテ・ドライブ, 6 9 3 8

(72)発明者 ジェフリー・エー．・シュミット

アメリカ合衆国 カリフォルニア州 9 5 0 0 6 ボウルダー・クリーク, ロッジ・ロード, 5 3 5

(72)発明者 グレゴリー・エル．・スローター

アメリカ合衆国 カリフォルニア州 9 4 3 0 6 パロ・アルト, エマーソン・ストリート, 3 3 2 6

審査官 桑原 雅子

(56)参考文献 英国特許出願公開第 0 2 3 1 1 3 8 9 (G B , A)

特開平 1 0 - 0 7 4 1 4 7 (J P , A)

特開昭 6 3 - 2 0 5 7 4 7 (J P , A)

特開平 0 5 - 2 9 8 2 2 3 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F 15/00

G06F 13/00