

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4902120号
(P4902120)

(45) 発行日 平成24年3月21日(2012.3.21)

(24) 登録日 平成24年1月13日(2012.1.13)

(51) Int. Cl. F I
G06F 9/445 (2006.01) G06F 9/06 640A
G06F 13/00 (2006.01) G06F 13/00 530B

請求項の数 7 (全 25 頁)

| | |
|--|---|
| <p>(21) 出願番号 特願2005-10920 (P2005-10920) (22) 出願日 平成17年1月18日 (2005.1.18) (65) 公開番号 特開2005-259114 (P2005-259114A) (43) 公開日 平成17年9月22日 (2005.9.22) 審査請求日 平成20年1月18日 (2008.1.18) (31) 優先権主張番号 10/799, 351 (32) 優先日 平成16年3月12日 (2004.3.12) (33) 優先権主張国 米国 (US)</p> | <p>(73) 特許権者 500046438 マイクロソフト コーポレーション アメリカ合衆国 ワシントン州 9805 2-6399 レッドモンド ワン マイ クロソフト ウェイ (74) 代理人 100077481 弁理士 谷 義一 (74) 代理人 100088915 弁理士 阿部 和夫 (72) 発明者 デビッド ビー. デーガン アメリカ合衆国 98052 ワシントン 州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーシ ョン内</p> |
|--|---|

最終頁に続く

(54) 【発明の名称】 ソフトウェア更新を配信するためのシステム及び方法

(57) 【特許請求の範囲】

【請求項 1】

クライアントコンピュータへの配信のために通信ネットワークを介してソフトウェア更新を配信するためのソフトウェア更新配信システムであって、前記ソフトウェア更新配信システムは、

クライアントコンピュータにソフトウェア更新を配信するよう動作可能なルート更新サービスノードと、

クライアントコンピュータにソフトウェア更新を配信するよう動作可能な複数の子更新サービスノードと

を備え、前記複数の子更新サービスノードの各々は、

ソフトウェア更新を記憶するための更新ストアと、

前記子更新サービスノードが前記通信ネットワークを介して当該子更新サービスノードにとっての親更新サービスノードからソフトウェア更新を入手するためのおよび前記子更新サービスノードが前記通信ネットワークを介して当該子更新サービスノードにとっての子更新サービスノードにソフトウェア更新を配信するための、子更新サービスノードの更新ウェブサービスであって、前記親更新サービスノードは、上流に直接接続された、ルート更新サービスノードまたは別の子更新サービスノードのうちの1つを含む、子更新サービスノードの更新ウェブサービスと、

前記複数の子更新サービスノードの各々において確立された規則に従って、1つまたは複数の子更新サービスノードに配信可能なソフトウェア更新を決定するための子更新モ

ジュールと、

更新活動報告を生成して前記親更新サービスノードに送信するための報告モジュールと、

ソフトウェア更新を前記クライアントコンピュータに配信するためのクライアント更新モジュールであって、前記複数の子更新サービスノードの各々において確立された規則に従って、ソフトウェア更新を前記クライアントコンピュータに配信する、クライアント更新モジュールと

を備え、

前記ルート更新サービスノードおよび前記複数の子更新サービスノードは、前記ルート更新サービスノードが少なくとも1つの子更新サービスノードにとっての親更新サービスノードであるような階層方式で体系化され、前記複数の子更新サービスノードの各々は親更新サービスノードを有し、前記複数の子更新サービスノードのうちの第1の子更新サービスノードは、前記複数の子更新サービスノードのうちの第2の子更新サービスノードにとっての親更新サービスノードであり、

前記ルート更新サービスノードは、第1の管理アプリケーションプログラミングインタフェース（API）および第1の管理ユーザインタフェースを備え、前記第1の管理APIおよび第1の管理ユーザインタフェースは、前記ルート更新サービスノードの管理者から、ソフトウェア更新を前記複数の子更新サービスノードのうちの少なくとも幾つかに配信するための規則の第1のセットを受け取るように動作可能であり、

少なくとも前記第1の子更新サービスノードは、前記第1の管理APIとは別の第2の管理APIと、前記第1の管理ユーザインタフェースとは別の第2の管理ユーザインタフェースとを備え、前記第2の管理APIおよび第2の管理ユーザインタフェースは、ソフトウェア更新を前記第1の子更新サービスノードから少なくとも前記第2の子更新サービスノードに配信するための規則の第2のセットを受け取るように動作可能であり、前記規則の第2のセットは、第1のグループに属するところの第2の子更新サービスノードを定義し、

前記ルート更新サービスノードは、ソフトウェアプロバイダのコンピュータから受け取ることによって、第1のソフトウェア更新を入手し、前記複数の子更新サービスノードのうちの少なくとも1つは、自己の親更新サービスノードから前記第1のソフトウェア更新を入手することによって、配信のために前記第1のソフトウェア更新を入手することを特徴とするソフトウェア更新配信システム。

【請求項2】

前記ルート更新サービスノードは、

ソフトウェア更新を記憶するための更新ストアと、

前記ルート更新サービスノードが前記通信ネットワークを介してその子更新サービスノードにソフトウェア更新を配信するための、前記ルート更新サービスノードの更新ウェブサービスと、

前記通信ネットワークを介してそのソフトウェア更新を前記ルート更新サービスノードにソフトウェアプロバイダのコンピュータが送信するための、通信インタフェースである、ソフトウェアプロバイダインタフェースと

を備えることを特徴とする請求項1に記載のソフトウェア更新配信システム。

【請求項3】

前記更新ストアは、前記ソフトウェア更新の更新ペイロードが記憶される更新内容ストアと、前記ソフトウェア更新の更新メタデータが記憶される更新情報ストアとを備え、前記更新メタデータは、前記更新ペイロードを入手するための参照を含んだ前記ソフトウェア更新に対応する情報を含み、

前記第1の子更新サービスノードは、前記ソフトウェア更新の更新メタデータを前記ルート更新サービスノードから入手することと、続けて前記ソフトウェア更新の更新ペイロードを前記ルート更新サービスノードから入手することとによって、前記ソフトウェア更新を前記ルート更新サービスノードから入手することを特徴とする請求項1に記載のソフ

10

20

30

40

50

トウェア更新配信システム。

【請求項 4】

前記第 1 の子更新サービスノードは、前記ソフトウェア更新の更新ペイロードを前記ルート更新サービスノードからジャストインタイム方式で入手し、前記ジャストインタイム方式は、更新ペイロードが必要なときのみダウンロードされる方式であることを特徴とする請求項 3 に記載のソフトウェア更新配信システム。

【請求項 5】

前記ルート更新サービスノードは、ソフトウェア更新をクライアントコンピュータに配信するためのクライアント更新モジュールを更に備え、

前記クライアント更新モジュールは、前記ルート更新サービスノードの管理ユーザインタフェースを用いて前記ルート更新サービスノードの管理 API を介して前記ルート更新サービスノードの管理者が前記ルート更新サービスノードにおいて確立した規則に従い、ソフトウェア更新をクライアントコンピュータに配信することを特徴とする請求項 1 に記載のソフトウェア更新配信システム。

【請求項 6】

前記第 1 の子更新サービスノードは、新しい更新を求めて前記ルートノードに定期的にクエリすることにより、ソフトウェア更新を前記ルート更新サービスノードから定期的に入手するように構成することができることを特徴とする請求項 1 に記載のソフトウェア更新配信システム。

【請求項 7】

ソフトウェア更新配信システムの階層構造内で、ソフトウェア更新の配信を容易にする方法であって、前記ソフトウェア更新配信システムは、

階層更新サービスノードのルート更新サービスノードと、

前記ルート更新サービスノードの下流に階層構造で体系化された複数の子更新サービスノードであって、前記複数の更新サービスノードの各々は、ソフトウェア更新をクライアントコンピュータに配信するように動作可能である、複数の子更新サービスノードと、

前記更新サービスノードの階層内の更新サービスノードに接続されたクライアントコンピュータへの配信のための、ソフトウェア更新に対応するソフトウェア更新情報とを備え、

前記ルート更新サービスノードは、前記複数の子更新サービスノードのうち少なくとも 1 つの子更新サービスノードにとっての親更新サービスノードであり、前記ルート更新サービスノードを除いた各更新サービスノードは、親更新サービスノードを有し、当該親更新サービスノードは、上流に直接接続された、ルート更新サービスノードまたは別の子更新サービスノードのうちの一つを含み、

前記複数の子更新サービスノードのうち少なくとも第 1 の子更新サービスノードは、前記複数の子更新サービスノードのうち別の子更新サービスノードにとっての親更新サービスノードであり、

前記複数の子更新サービスノードの各々は、管理アプリケーションプログラミングインタフェース (API) および管理ユーザインタフェースを備え、前記複数の子更新サービスノードの各々に対応する各管理者は、当該管理 API および管理ユーザインタフェースを通じて、ソフトウェア更新を配信するための規則のセットを確立し、規則の第 1 のセットは、第 1 のグループを確立し、且つ、当該第 1 のグループが利用可能なソフトウェア更新のサブセットを指定し、

前記方法は、

更新ストア内にソフトウェア更新を、前記複数の子更新サービスノードの各々が、記憶するステップと、

通信ネットワークを介して更新ウェブサービスを通じて自己の親更新サービスノードからソフトウェア更新を前記複数の子更新サービスノードの各々が、入手するステップと、

前記確立された規則に従って、1 つまたは複数の子更新サービスノードに配信可能なソフトウェア更新を、前記複数の子更新サービスノードの各々が、決定するステップと、

10

20

30

40

50

前記複数の子更新サービスノードの各々が、更新活動報告を生成して、前記親更新サービスノードに送信するステップと、

前記複数の子更新サービスノードの各々が、下流に直接接続されている1つまたは複数の子更新サービスノードまたはクライアントコンピュータは当該複数の子更新サービスの各々からソフトウェア更新を入手することを許可されているか否かを、確立した前記規則のセットに従い決定するステップと、

ソフトウェア更新をクライアントコンピュータに、前記複数の子更新サービスノードの各々が、前記決定に従い配信するステップと
を備え、

前記子更新サービスノードは、前記通信ネットワークを介して前記更新ウェブサービスを通じて自己の子更新サービスノードにソフトウェア更新を配信することを特徴とする方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はソフトウェアおよびコンピュータネットワークに関し、より詳細には、本発明は、ソフトウェア更新を配信するための更新配信システムアーキテクチャおよび方法に関する。

【背景技術】

【0002】

ほぼすべての市販ソフトウェア製品は、ソフトウェアの機能を修復または更新するための継続的な改訂プロセスを受ける。ソフトウェア製品の各改訂はしばしば、新しいファイルを追加すること、既存のファイルをより新しい改訂で置換すること、旧ファイルを削除すること、またはこれらの動作の様々な組合せを必要とする。この、ソフトウェア製品の古いファイルの置換、新しいファイルの追加、旧ファイルの削除や、機能を追加および/または削除することによってソフトウェア製品の構成および挙動を変更するといったプロセスを、以下では「製品を更新する」と言い、バイナリファイル、データファイル、更新命令、メタデータなどを含めた、製品を更新するのに使用されるデータ集合を、以下ではより単純に「更新（アップデート）」と言う。

【0003】

ソフトウェアプロバイダは、問題の修復、セキュリティ強化、または新機能の追加のためにソフトウェア製品の更新を作成すると、この更新をその顧客ベースに対して広く利用可能にしたいと思うことになる。更新が、製品中の欠陥を補正することやクリティカルなセキュリティ問題に対処することに向けられているときなど、ソフトウェアプロバイダがこの更新をできるだけ早く顧客のコンピュータにインストールさせたいと思うことは非常に多い。実際、ほとんどのソフトウェアプロバイダは、ソフトウェア更新をその顧客にできるだけ早く、またできるだけトラブルなく配信するというビジネス上の動機を有する。

【0004】

コンピュータ産業は、ネットワークに、特にインターネットに接続されたコンピュータの数の爆発的な増加を体験している。この爆発的な増加により、またインターネット接続を介して利用可能な通信能力により、インターネットは、ソフトウェアプロバイダがその顧客に更新を配信するための重要かつ不可欠なチャネルになった。実際、インターネットは、多くのソフトウェアプロバイダがその顧客にソフトウェア更新を配信するための主要な配信チャネルになっている。インターネットを介した電子更新配信は、ソフトウェアプロバイダのコスト全体を削減するので、またソフトウェア更新が入手可能になればすぐに顧客がそれを入手することができるので、インターネットを介してソフトウェア更新を配信するのが最もソフトウェアプロバイダの利益になることが多い。これらのソフトウェア更新がどんなユーザ介入もなしに自動的にインターネットを介して実施されることは、ますます多くなっている。

【0005】

10

20

30

40

50

インターネットは現在、ソフトウェアプロバイダからソフトウェア更新を配信するためのパイプとして一般的に使用されているが、いくつかの問題が頻繁に生じる。このような問題のうち2つとして、(1)更新配信インフラストラクチャ/リソースに関する効率と、(2)ソフトウェア更新の配信およびインストールに対する管理制御がある。

【0006】

配信リソースの効率に関しては、インターネットを含めてネットワークは、帯域幅としばしば呼ばれる通信リソースを、限りある量しか保有しない。限りある量の通信帯域幅は頻繁にボトルネックを生じ、特に、Microsoft CorporationのWindows(登録商標)ファミリのオペレーティングシステムおよび関連する生産性製品など、一般的なソフトウェア製品のソフトウェア更新に関してはそうである。このようなボトルネックは、インターネット全体に分散した複数のダウンロード位置でソフトウェア更新を入手可能にするときにも存在する。このようなボトルネックが生じる理由の1つは、インターネットによって利用可能になる体系化されていないアクセスモデルである。例えば、コンピュータAにおける第1ユーザがソフトウェア製品の最新ダウンロードを要求した場合、このダウンロードは、第1ユーザのISP(independent service provider)を経由する。さらに、要求は単一の個別化されたアクセスとして扱われ、このことは、要求が他のどんなネットワークトラフィックおよび/または要求からも独立し無関係であるものとして扱われることを意味する。したがって、やはり同じISPをたまたま有するコンピュータBにおける第2ユーザが、第1ユーザと同じダウンロードを要求した場合、第2ユーザからの要求も単一の個別化されたアクセスとして扱われる。この例では、各要求が別々に扱われたため、同じダウンロードが同じインフラストラクチャを介して2度にわたって送信されることになる。明らかに、ユーザの数が著しく増加した場合、限りある通信帯域幅はボトルネックになる。この例は非常によくみられるが、この場合に、ダウンロードをローカル位置でキャッシュすることができて、各ユーザ要求をローカルキャッシュから満たすことができているならば、はるかに効率的であつただろう。

【0007】

配信の制御に関しては、多くの組織、特に大規模な組織は、そのコンピュータへの更新配信を制御することに対する正当な理由を有する。例えば、残念ながら更新の中には、ソフトウェア製品の機能を「破壊」する欠陥を有するかまたは招くものがあり、これらの欠陥はしばしばバグと呼ばれる。これらの破壊された機能は、重要でない場合もあるが、ビジネスのミッションクリティカル(mission critical)な機能を混乱させる可能性が非常に多い。あるいは、特定のソフトウェア更新によってミッションクリティカルな機能が削除された場合、またはそれらの挙動が本質的に改変された場合に、それらが「破壊」される場合もある。ビジネスはそのミッションクリティカルな機能を失うわけにはいかないので、責任あるビジネスはまず、制御された環境内でいくらかの期間にわたって各ソフトウェア更新を評価およびテストし、その後、その残りのコンピュータに更新を公開することになる。この評価期間で、組織は、更新がミッションクリティカルな機能に悪影響を及ぼすかどうか妥当性検査することができる。更新がどんなミッションクリティカルな機能も低下させないことが十分に決定された後でのみ、更新を組織の残りのコンピュータに配信することが許可される。明らかに、ほとんどの組織は、そのコンピュータへのソフトウェア更新のインストールに対して制御を実施しなければならない。

【0008】

ビジネスまたは組織がしばしばソフトウェア更新を制御する必要があることの別の理由は、組織内のコンピュータ間の一貫性を保証することである。ワードプロセッサ用であろうとオペレーティングシステム用であろうと、すべてのコンピュータが動作する基盤となる標準化されたターゲットプラットフォームを有することは、情報サービス部門にとって非常に重要である。標準がなければ、ソフトウェアおよびコンピュータの保守は不要に複雑で困難になるであろう。

【0009】

10

20

30

40

50

ローカル制御が重要であることの別の理由は、課金のためである。大規模な組織では、ソフトウェアを個別にコンピュータにインストールすること、または特定のソフトウェア製品のライセンスを組織内の各コンピュータごとに個別に維持することは、非効率的であることが多い。そうではなく、単一のサイトライセンスが、組織がソフトウェア製品を多くのコンピュータ上で実行することを許可する。したがって組織は、サイトライセンスの下で製品を実行するコンピュータの数を報告する必要があるか、あるいはサイトライセンスの下で製品を実行するコンピュータの数を制限する必要がある場合がある。これらの理由はすべて、ソフトウェア更新配信に対するローカル制御をしばしば必要とする。

【0010】

【特許文献1】「Controlling Installation Update Behaviors On a Client Computer」という名称の米国特許出願、弁理人整理番号 M S F T - 1 - 2 2 4 5 9

10

【発明の開示】

【発明が解決しようとする課題】

【0011】

上に識別した、ソフトウェア更新配信に関する様々な問題に鑑みて、ソフトウェア更新の配信に対する制御を提供するため、およびソフトウェア更新の配信効率を高めるための、拡張可能なソフトウェア更新配信アーキテクチャが必要とされている。本発明は、従来技術にみられるこれらおよびその他の問題に対処するものである。

【課題を解決するための手段】

【0012】

20

本発明によれば、通信ネットワークを介してソフトウェア更新を配信するためのソフトウェア更新配信システムが提示される。このソフトウェア更新配信システムは、ルート更新サービスノードと、少なくとも1つの子更新サービスノードとを備える。本発明の一態様によれば、ルート更新サービスノードおよび少なくとも1つの子更新サービスノードは、階層方式で構成される。ルート更新サービスノードは、少なくとも1つの子更新サービスノードの親更新サービスノードである。ルート更新サービスノードを除く各更新サービスノードは、親更新サービスノードを有する。各子更新サービスノードはまた、ソフトウェア更新配信システム階層中の少なくとも1つの他の更新サービスノードに対する親更新サービスノードである場合もある。加えて、ルート更新サービスノードは、ソフトウェアプロバイダからソフトウェア更新を入手する。少なくとも1つの子更新サービスはそれぞれ、その親更新サービスノードからソフトウェア更新を入手する。

30

【0013】

本発明の追加の態様によれば、ソフトウェア更新をクライアントコンピュータに配信するための更新サービスノードと、子更新サービスノードとが提示される。更新サービスノードは、ソフトウェア更新を記憶するための更新ストアを備える。更新サービスノードはウェブサービスインタフェースも備え、このウェブサービスインタフェースを介して、更新サービスノードはその親更新サービスノードからソフトウェア更新を入手し、子更新サービスノードにソフトウェア更新を配信する。更新サービスノードはさらに管理アプリケーションプログラミングインタフェース (API) も備え、これを介して管理者は、管理ユーザインタフェースを使用して、ソフトウェア更新を子更新サービスノードに配信するための規則を確立する。更新サービスノードはさらに、確立した規則に従ってどのソフトウェアが子更新サービスノードへの配信に利用可能かを決定するための子更新モジュールも備える。

40

【0014】

本発明の他の態様によれば、ソフトウェア更新配信システム中の子更新サービスノードがその親更新サービスノードからソフトウェア更新を入手する方法が提示される。子更新サービスノードは、その親更新サービスノードからソフトウェア製品のリストを入手する。ソフトウェア製品のリストは、親更新サービスノードがソフトウェア更新を配信するソフトウェア製品を識別する。子更新サービスノードは、ソフトウェア更新が求められるソフトウェア製品を選択する。次いで子更新サービスノードは、更新が求められるソフトウ

50

エア製品を識別する要求を親更新サービスノードに送信する。これに回答して子更新サービスノードは、子更新サービスノードに入手可能な、選択したソフトウェア製品に対するソフトウェア更新を識別する更新リストを受け取る。次いで子更新サービスノードは、ソフトウェア更新リスト中の各ソフトウェア更新を親更新サービスノードから入手する。

【0015】

本発明の別の態様によれば、親更新サービスノード上で実行される、ソフトウェア更新を子更新サービスノードに配信する方法が提示される。ソフトウェア更新カタログを求める要求を子更新サービスノードから受け取る。ソフトウェア更新カタログは、親更新サービスノードがソフトウェア更新を配信するソフトウェア製品を識別するものであり、これを子更新サービスノードに返す。次いで、選択されたソフトウェア製品に対するソフトウェア更新のリストを求める要求を受け取る。これに回答して、親更新サービスノードは、選択されたソフトウェア製品に対するいずれかのソフトウェア更新が子更新サービスノードに入手可能かどうかを判定する。子更新サービスノードに入手可能であると判定された、選択されたソフトウェア製品に対するソフトウェア更新を識別する更新リストを、子更新サービスノードに返す。更新リストは、入手可能なソフトウェア更新を固有の更新識別子に従って識別する。更新リスト中で識別される入手可能なソフトウェア更新に対応するメタデータを求める更新要求を受け取る。更新要求は、入手可能なソフトウェア更新をその固有識別子で識別する。これに回答して、更新メタデータを子更新サービスノードに返す。更新メタデータは、対応する更新ペイロードを入手するための参照を含めた、ソフトウェア更新に対応する情報を含む。

【0016】

本発明に関する前述の態様および付随する多くの利益は、以下の詳細な説明を添付の図面と共に参照することによってよりよく理解されるようになるので、より容易に認識されるようになるであろう。

【発明を実施するための最良の形態】

【0017】

本発明の態様によれば、階層方式で構成された、ソフトウェア更新を配信するための更新配信システムが提示される。すでに述べたように本発明では、ソフトウェア更新は、古いファイルおよび/または旧ファイルを置換または削除すること、追加のファイルを加えること、ソフトウェア製品の構成を改変することなどを含む場合がある。加えて、技術的な意味および文字どおりの意味では特にソフトウェア製品のソフトウェア更新を対象とするわけではないが、本発明の態様は有利にも、ソフトウェア製品の配信に利用することもできる。したがって、以下の考察は更新配信システムにおけるソフトウェア更新の配信に関して行うが、これは例示のためにすぎず、本発明を限定するものと解釈すべきではない。

【0018】

図1は、本発明の態様により形成された例示的な更新配信システム100を示す図である。本発明によれば、図示の更新配信システム100などの更新配信システムの「最上部」には、ルート更新サービスノード102がある。ソフトウェアプロバイダ110などのソフトウェアプロバイダが、ルート更新サービスノード102に更新を送信することによって、更新配信システム100を介してそのソフトウェア更新を配信する。本発明の態様によれば、ソフトウェアプロバイダ110などのソフトウェアプロバイダは、インターネット108などのネットワークを介して、そのソフトウェア更新をルート更新サービスノード102に送信することができる。

【0019】

この例示的な更新配信システム100などの階層型更新配信システムは、ルート更新サービスノード102に加えて、少なくとも1つの他の更新サービスノードをおそらく含むことになる。図1に示すように、例示的な更新配信システム100は、ルート更新サービスノード102と2つの追加の更新サービスノード、すなわち更新サービスノード104および更新サービスノード106を含む。本発明によれば、各階層型更新配信システムは

、ルート更新サービスノード102の下でツリー状の構造に構成される。言い換えれば、更新配信システム中の各更新サービスノードは、0個またはそれ以上の子更新サービスノードを有する。したがって、例示的な更新配信システム100では、各親更新サービスノードすなわちルート更新サービスノード102および更新サービスノード104が1つの子しか有さないように示されているが、これは例示のためにすぎず、本発明を限定するものと解釈すべきではない。さらに、本発明の一実施形態によれば、ルート更新サービスノード102を除いて、更新配信システム中の各更新サービスノードは、1つの親更新サービスノードを有する。したがって図1に示すように、更新サービスノード104はルート更新サービスノード102の子ノードであり、更新サービスノード106は更新サービスノード104の子ノードである。代替の一実施形態では、子更新サービスノードは、複数の親更新サービスノード(図示せず)から更新を受け取ることができる。図からわかるように、ルート更新サービスノード102を除いて、各更新サービスノードは、子更新サービスノードと親更新サービスノードの両方になることができる。

10

【0020】

例示的な更新配信システム100に示すように、ルート更新サービスノード102は、インターネット108を介して更新サービスノード104と通信する。ただし、これは例示にすぎず、本発明を限定するものと解釈すべきではないことを理解されたい。更新配信システム中の各更新サービスノードは、何らかの通信ネットワークを介してその親および/または子と通信できればよい。したがって、更新サービスノード104は、インターネット108を介してその親であるルート更新サービスノード102と通信するが、別法として、ローカルエリアネットワーク124を介して、更新サービスノード106などのその子更新サービスノードと通信することもできる。

20

【0021】

図1にはまた、更新サービスノード106がローカルエリアネットワーク124のサブネットワーク126内にあるのも示す。例として、ローカルエリアネットワーク124は、組織の一般的な企業ネットワークに対応するものとして、更新サービスノード104は、その親であるルート更新サービスノード102への接続を介した、企業から更新配信システム100へのリンクを表す。さらに、サブネットワーク126は、テスト/評価グループや、リモートに位置するオフィスや、ミッションクリティカルなグループなど、企業ネットワーク内のコンピュータの識別可能グループに対応するものとして、

30

【0022】

ルート更新サービスノード102と更新サービスノード104および106との両方を含めて、各更新サービスノードは、子更新サービスノードとクライアントコンピュータの両方にソフトウェア更新を配信するように構成されることを理解されたい。図1に示すように、例示的な更新配信システム100は、クライアントコンピュータ112~122を含む。ルート更新サービスノード102を含めた各更新サービスノードは、ローカル構成情報に従って、子更新サービスノードおよびクライアントコンピュータに更新を配信する。一実施形態では、管理者がグループを定義し、これらのグループに更新配信規則を関連付ける。各更新サービスノードは、少なくとも1つの配信グループを有する。

40

【0023】

更新配信システムがどのように動作するかを示すための例として、ローカルエリアネットワーク124がビジネス組織の企業ネットワークに対応すると仮定する。本発明の一実施形態によれば、更新サービスノード104上の管理者が、企業ネットワーク124に対して複数の配信グループを定義することができる。配信グループには、更新サービスノード106とクライアントコンピュータ120および122とを含むサブネットワーク126に対応する評価グループが含まれ、これは、更新の適正を全般的企業ネットワーク124に対して評価するための評価グループである。配信グループにはまた、更新サービスノ

50

ード104とクライアントコンピュータ114~118とを含む全般的企業グループが含まれる。

【0024】

評価グループに関して管理者は、更新サービスノード106をメンバとして含め、更新が入手可能になればすぐに評価グループのメンバに更新が配信されるように、このグループに規則を関連付ける。あるいは、全般的企業グループに関して管理者は、クライアントコンピュータ114~118を加え、管理者によって特に許可された場合にのみ更新が全般的企業グループのメンバに配信されるように規則を関連付ける。また、子更新サービスノード106に対する管理者が、評価サブネットワーク126中のクライアントコンピュータ120および122からなるデフォルトグループを生み出すとも仮定する。新しいソフトウェア更新があれば、すぐにこのグループに配信することができる。

10

【0025】

上の例を続けるが、ソフトウェアプロバイダ110が、ソフトウェア更新をルート更新サービスノード102に送信する。ルート更新サービスノード102において確立された規則に従って、更新は最終的に企業更新サービスノード104に配信される。企業更新サービスノード104は、更新を受け取ると、管理者によって確立された規則により、更新を評価グループのメンバ(子更新サービスノード106としてのみ定義されている)に配信するが、全般的企業グループに更新を配信する許可が特に下りるまでは、全般的企業グループには更新を提供しないでおく。

【0026】

20

上の例を続けるが、評価更新サービスノード106は、更新を受け取ると、各定義グループごとに更新を処理する。この例では、評価更新サービスノード106は1つのグループだけを有する。ただし前述のように、実際の実装形態では、複数の定義グループがあって、各グループが配信規則に関連した固有のセットに関連していてもよい。この例では、評価更新サービスノード106はすぐに、更新をクライアントコンピュータ120および122への配信に利用可能にする。今やクライアントコンピュータ120および122を更新することができ、評価期間/プロセスを開始することができる。

【0027】

上の例を続けるが、企業更新サービスノード104上の管理者は、更新が企業ネットワーク124全体への配信に適することに十分納得すると、更新が全般的企業グループのメンバに配信されるのを明示的に許可する。企業更新サービスノード104は、これに対応して、更新をクライアントコンピュータ114~118に入手可能にする。評価更新サービスノード106は全般的企業グループにも含まれてよいことを理解されたい。しかし、評価更新サービスノード106はすでに更新されているので、更新を評価サブネットワーク126に配信するための追加の更新関連動作は必要ない。

30

【0028】

上の例でわかるように、本発明は、ローカル配信制御およびダウンロード効率の点で大きな利益をもたらす。ローカル配信制御に関する前述の態様に加えて、通信帯域幅のかなりの節約も実現する。例えば、図1に示した例示的な企業ネットワーク124は5つのクライアントコンピュータを含むが、ソフトウェアプロバイダの更新は、ルート更新サービスノード102から企業更新サービスノード104に一度ダウンロードされただけである。明らかにこの場合、更新サービスノードによってサービスされるクライアントコンピュータの数が増加しても、親更新サービスノードと子更新サービスノードとの間の通信帯域幅の使用は一定のままであり、それにより、通常なら使用されるであろう通信帯域幅の量が大幅に削減される。加えて、この更新配信システムは、拡張可能であるとともにスケール可能である。更新配信システムは少なくとも2つの方式で拡張可能である。すなわち、任意の数の子更新サービスノードを親更新サービスノードに追加することができ、子更新サービスノードは親更新サービスノードになることもできる。したがって、個別の必要を満たすように更新配信システムの各サブツリーを適合させることができる。

40

【0029】

50

本発明によって実現される別の利益は、管理者の担当するクライアントコンピュータおよびネットワークにわたって一貫したソフトウェアプラットフォームを管理者が確立および施行するのが容易なことである。例えば、管理者がクライアントコンピュータにわたる一貫したソフトウェアプラットフォームを保証する現在のプロセスは、一般に次のとおりである。どんなソフトウェアプラットフォームにするか、すなわちどの更新をインストールすべきであってどの更新をインストールすべきでないかを決定し、各クライアントコンピュータ上のソフトウェアプラットフォームのステータスを決定し、適切なソフトウェア更新をクライアントコンピュータにプッシュして、インストールされるようにする。最初にかかる手間の後も、管理者は、ソフトウェアプラットフォームが一貫性を保つようにするために、このプロセスを頻繁に繰り返さなければならない。これらの従来技術の手法とは対照的に、前述の更新配信システムを利用することによって、かつ、クライアントマシンを異なるターゲットグループに分類するためのインフラストラクチャサポート、および異なる更新を異なるターゲットグループに向けることができること（後述する）を利用することによって、管理者は、管理者が担当するネットワークおよび/またはクライアントコンピュータのソフトウェア更新インストールに関する所望の状態を指定することができる。管理者は、任意の数のクライアントコンピュータおよび/またはクライアントコンピュータグループに対して、管理API（これも後述する）を介して所望の状態を指定する。所望の更新状態を確立/指定した後は、管理者は、更新配信システムに依拠して更新配信システム全体への更新配信を実施することができる。更新配信システムは、システム中のクライアントが管理者によって指示された所望の状態に適合することを保証するように機能する。定期的な動作を介して、更新配信システムは、所望の状態に適合しないクライアントコンピュータを適合させることを絶えず試みる。加えて、強制インストール属性やインストール期限付きインストール属性など、特定のインストール属性をソフトウェア更新に関連付けることによって、更新配信システムは、一貫したソフトウェアプラットフォームの維持に関連する実施詳細の大部分から管理者を解放する。インストール属性をソフトウェア更新に関連付けてクライアントコンピューティングデバイス上のインストール挙動を制御することに関するより詳細な記述は、同時係属であり本願の譲受人に譲渡された2004年3月12日出願の米国特許出願（特許文献1）に記載されている。

【0030】

図2は、本発明の態様により形成された、企業更新サービスノード104（図1）や評価更新サービスノード106（図1）などの更新サービスノード200の例示的な論理コンポーネントを示すブロック図である。図2に示すように、更新サービスノード200は、更新ウェブサービス202、クライアント更新モジュール204、子更新モジュール206、報告モジュール208を備える。例示的な更新サービスノード200はまた、認証/許可モジュール210、管理アプリケーションプログラミングインタフェース（API）212、更新内容ストア214、管理ユーザインタフェース218、更新情報ストア216も備える。

【0031】

更新ウェブサービス202は、クライアントコンピュータ、子更新サービスノード、親更新サービスノードが更新サービスノードと通信できるためのウェブサービスの共通セットを提供する。例えば図1を参照すると、子/評価更新サービスノード106が親/企業更新サービスノード104からソフトウェア更新を入手するためには、クライアントは親の更新ウェブサービス202を介して通信する。同様に、ルート更新サービスノード102などの親更新サービスノードが、更新を含めてその子更新サービスノード104に通信すべき情報を有するとき、親更新サービスノードは、子の更新ウェブサービス202を介して通信する。

【0032】

クライアント更新モジュール204は、更新サービスノードに記憶された更新および更新情報に関する、クライアントコンピュータと更新サービスノード200との間の通信を扱う。限定しないが、これらの更新関連の通信には、クライアント要求に回答して更新を

10

20

30

40

50

配信すること、入手可能なソフトウェア製品および関連する更新のリストをクライアントコンピュータに提供することが含まれる。クライアント更新モジュール204はまた、クライアントコンピュータが特定の更新を入手することが許可されているかどうかを関連の配信規則に従って判定することも担い、クライアントコンピュータによるアクセスが許可されている更新関連の情報でクライアントコンピュータに応答する。

【0033】

子更新モジュール206は、親更新サービスノードとその子更新サービスノードとの間の更新関連の通信を扱う。限定しないが、これらの更新関連の通信には、子更新サービスノードに入手可能なソフトウェア製品および関連する更新のリストを識別すること、ならびに、子更新サービスノードからの更新要求に応答することが含まれる。子更新モジュール206は、子更新サービスノードが特定の更新を入手することが許可されているかどうかを関連の配信規則に従って判定することを担い、子更新サービスノードによるアクセスが許可されている更新関連の情報で子更新サービスノードに応答する。

10

【0034】

報告モジュール208は、どのグループが特定の更新を受け取ったかまたは受け取っていないか、どのクライアントコンピュータが更新をダウンロード/インストールしたかまたはしていないか、どの更新が更新サービスノード上で入手可能かなど、更新関連の報告を生成する。これらの報告は、管理者などによって内部で使用することもでき、また、親の更新サービスインタフェース202を介して親更新サービスノードに送信することもできる。前述のように、課金や保守の目的で、特定の更新がどのクライアントコンピュータにインストールされているかを企業が決定する必要がしばしばある。報告モジュール208によって生成された情報/報告を、これらの報告の基礎にすることができる。

20

【0035】

認証/許可モジュール210は、認証を担う。すなわち、特定のクライアントコンピュータまたは子更新サービスノードの識別を決定すること、および、更新サービスノード200における入手可能な更新へのアクセスがクライアントコンピュータまたは子更新サービスノードに許可されているかどうかを判定することを担う。更新サービスノード上の更新へのアクセスを認証および許可されているクライアントコンピュータおよび子更新サービスノードに対しては、認証/許可モジュール210は許可トークンを発行する。許可トークンは、更新の入手に関連して使用しなければならないものである。許可トークンの発行および使用については、後で図4Aおよび4Bに関してより詳細に述べる。

30

【0036】

管理API212は、更新サービスノード200の制御を実施するため、および更新を最終的に記憶し配信するためのアプリケーションインタフェースを表す。更新ウェブサービス202がクライアントコンピュータおよび子更新サービスノードから様々な更新関連の要求を受け取ると、最終的にこれらの要求は、直接に、またはクライアント更新モジュール204および子更新モジュール206を介して間接的に、管理API212への呼び出しに分割される。管理ユーザインタフェース218と共に、または更新サービスノード200にインストールされた他の何らかのプログラムであって管理API212を使用するように適切に構成されたプログラムと共に、管理者は最終的に、この更新サービスノード、ならびに任意の子更新サービスノードおよびクライアントコンピュータのための更新プロセスのすべての面を制御する。

40

【0037】

管理ユーザインタフェース218を介して、管理者は、更新サービスノード200を管理API212によって構成および維持することができる。したがって、管理ユーザインタフェース218を介して、管理者は、グループを作成、修正、削除し、また各グループに対する規則を関連付ける。さらに管理者は、管理ユーザインタフェース218を使用して、クライアントコンピュータまたは子更新サービスノードがどのグループに属するかを確立する。管理者はまた、管理ユーザインタフェース218を介して、クライアントコンピュータまたは子更新サービスノードへの更新の配信を明示的に許可すること、新しい更

50

新があるかどうか定期的にその親更新サービスノードに照会するよう更新サービスノード 200 を構成すること、報告パラメータを構成し内部報告を見ることなどを行うことができる。前述のように、管理ユーザインタフェース 218 は、管理者が更新サービスノード 200 の各面に対する制御を実施できるようにするが、管理ユーザインタフェース 218 の代わりに、更新サービスノード 200 上にある、管理 API 212 と共に動作するように適切に適合された別のアプリケーションを使用してもよい。

【0038】

前述のように、本発明の一実施形態によれば、更新サービスノード 200 は、更新内容ストア 214 と更新情報ストア 216 の両方を備える。更新内容ストア 214 は、バイナリやパッチファイルなど、ソフトウェア更新を表す実際のファイルを記憶する。対照的に、更新情報ストア 216 は、更新内容ストア 214 に記憶された更新ファイルを含めて更新サービスノード 200 上で入手可能な更新に対応する情報およびメタデータを記憶する。一実施形態によれば、更新内容ストア 214 と更新情報ストア 216 は両方ともリレーショナルデータベースである。この例示的な更新サービスノード 200 は 2 つのデータストアを有するように示してあるが、本発明はこのように限定されるべきではない。代替の一実施形態では、更新内容ストア 214 と更新情報ストア 216 の両方を単一の情報ストアに結合することもできる。

【0039】

本発明の態様によれば、ソフトウェア更新が物理的に更新内容ストア 214 に記憶されていなくても、更新を更新サービスノード 200 上で「入手可能」なものとしてクライアントコンピュータおよび子更新サービスノードに対して提示することができる。より具体的には、実際の更新ファイルをすぐに更新サービスノード 200 にダウンロードして記憶するのではなく、親更新サービスノードまたは他のどこかにある更新ファイルを参照するリンクを、代わりに更新サービスノードに記憶することができる。したがって、クライアントコンピュータが更新を要求した場合、または子更新サービスノードが実際の更新を要求した場合に、更新は親更新サービスノードから提供され、クライアントコンピュータまたは子更新サービスノードへの送達に備えて更新内容ストア 214 に記憶される。このタイプの更新アクセスがジャストインタイム (just-in-time) ダウンロードと呼ばれることは、当業者なら理解するであろう。このようにすれば、「入手可能」な更新は、それが実際に要求されるまでは、様々なネットワークチャネルを介して配信する必要はない。本発明の態様によれば、更新サービスノード 200 の管理者は、ソフトウェア更新をジャストインタイム方式で入手するかどうかを選択的に決定することができる。

【0040】

上の図 2 の記述では、例示的な更新サービスノード 200 の様々なコンポーネントを例示しているが、更新サービスノードのその他のコンポーネントが存在してもよいことを理解されたい。さらに、前述のコンポーネントは、必ずしも実際のコンポーネントではなく論理コンポーネントとして理解されたい。実際の実装形態では、実装決定に従って、上に識別したコンポーネントを共に組み合わせることもでき、かつ/または他のコンポーネントと組み合わせることもできる。加えて、更新サービスノード 200 はネットワーク上のサーバコンピュータと見なすことができるが、実際の実装形態では、更新サービスノードは任意の数のタイプのコンピューティングデバイス上で実現することができることを理解されたい。例えば、各更新サービスノード 200 は、単一のスタンドアロンコンピュータシステム上で、あるいは複数のコンピューティングデバイスを含む分散コンピューティングシステム上で実施および/またはインストールすることができる。

【0041】

図 3 は、本発明の態様により形成された、図 1 に示したルート更新サービスノード 102 などのルート更新サービスノード 300 の例示的な論理コンポーネントを示すブロック図である。更新サービスノード 200 のコンポーネント (図 2) と同様、ルート更新サービスノード 300 は、更新ウェブサービス 202、子更新モジュール 206、認証/許可モジュール 210 を備える。加えて、例示的なルート更新サービスノード 300 は、管理

10

20

30

40

50

API 212、更新内容ストア 214、更新情報ストア 216 も備える。任意選択で、ルート更新サービスノード 300 はまた、クライアント更新モジュール 204、報告モジュール 208、管理ユーザインタフェース 218 を備えることもできる。

【0042】

クライアント更新モジュール 204 は、ルート更新サービスノードがソフトウェア更新をクライアントコンピュータに直接提供するかどうかに応じた、ルート更新サービスノード 300 のためのオプションのコンポーネントである。例えば図 1 を参照すると、ルート更新サービスノード 102 は、クライアントコンピュータ 112 に直接にサービスするルート更新サービスノードとして、オプションのクライアント更新モジュール 204 を備えることになる。しかし、ルート更新サービスノード 300 がクライアントコンピュータに直接にサービスしない場合は、クライアント更新モジュール 204 は省くことができる。

10

【0043】

報告モジュール 208 は、ルート更新サービスノード 300 に対してはオプションである。というのは、ルート更新サービスノード 300 には、更新報告を提供する対象となる親更新サービスノードがないからである。しかし、ルート更新サービスノードの管理者にとって更新報告が望ましい場合は、任意選択で報告モジュール 208 を含めることができる。

【0044】

更新サービスノード 200 (図 2) に含まれる論理コンポーネントを備えることに加えて、ルート更新サービスノード 300 は、ソフトウェアプロバイダインタフェース 302 も備える。ソフトウェアプロバイダインタフェース 302 は、ソフトウェアプロバイダ 110 (図 1) がソフトウェア更新を直接にルート更新サービスノード 300 に送信するため、および間接的に例示的な更新配信システム 100 に送信するための通信インタフェースを提供する。

20

【0045】

図 2 の更新サービスノード 200 と同様、上の図 3 の記述では、例示的なルート更新サービスノード 300 の様々なコンポーネントを例示している。しかし、ルート更新サービスノードのその他のコンポーネントが存在してもよいことを理解されたい。さらに、前述のコンポーネントは、必ずしも実際のコンポーネントではなく論理コンポーネントとして理解されたい。実際の実装形態では、実装決定に従って、上に識別したコンポーネントを共に組み合わせることもでき、かつ/または他のコンポーネントと組み合わせることもできる。加えて、更新サービスノード 200 はネットワーク上のサーバコンピュータと見なすことができるが、実際の実装形態では、更新サービスノードは任意の数のタイプのコンピューティングデバイス上で実現することができることを理解されたい。例えば、ルート更新サービスノード 300 は、単一のスタンドアロンコンピュータシステム上で、あるいは複数のコンピューティングデバイスを含む分散コンピューティングシステム上で実施および/またはインストールすることができる。

30

【0046】

更新がどのようにルート更新サービスノードから更新配信システム 100 全体に配信されるかをよりよく理解するために、親更新サービスノードと子更新サービスノードとの間の例示的なやりとりは例示するに値する。図 4 は、本発明の態様による、ソフトウェア更新を親更新サービスノードから子更新サービスノードに伝搬する際の、親更新サービスノード 402 と子更新サービスノード 404 との間の例示的なやりとり 400 を示すブロック図である。図からわかるように、例示的な図 400 は半分に分かれており、左半分は親更新サービスノード 402 の動作およびイベントに対応し、右半分は子更新サービスノード 404 の動作およびイベントに対応する。

40

【0047】

図 4 に関する考察ではさらに、親更新サービスノード 402 は更新配信システム 100 中のルート更新サービスノードであってもよく、そうでなくてもよいことを理解されたい。加えてこの考察では、管理者によって明示的に許可されない限り子更新サービスノード

50

404がソフトウェア更新を受け取ることができないように、親更新サービスノード402が管理者によって構成されていると仮定する。

【0048】

例示的なやりとり400に示すように、イベント406で開始し、親更新サービスノード402は、ソフトウェアプロバイダ110からソフトウェア更新を受け取る。これは、親更新サービスノードがルート更新サービスノード102である場合は直接に受け取り、そうでない場合は更新配信システム100を介して間接的に受け取る。親更新サービスノード402がソフトウェアプロバイダ110からソフトウェア更新を受け取ってからいくらか後の時点で、子更新サービスノード404は、親更新サービスノードからソフトウェア更新を入手するプロセスを開始する。

10

【0049】

一実施形態では、子更新サービスノード404は、親更新サービスノード402から入手可能なソフトウェア更新を定期的に自動入手するように構成することができる。より具体的には、管理者は、親更新サービスノード402上で入手可能な最新のソフトウェア更新を定期的に入手するように、管理ユーザインタフェース218を介して子更新サービスノード404を選択的に構成することができる。一例として管理者は、子更新サービスノード404を構成して、毎日および/または毎時、その親更新サービスノード402から最新のソフトウェア更新を入手するようにすることができ、また、自動更新プロセスを開始する時刻を指定することができる。その他の定期的スケジュールおよび基準を使用することもできる。同様に、管理者は、管理ユーザインタフェース218を介して手動で更新プロセスを開始することもできる。

20

【0050】

更新プロセスを開始するために、イベント408で、子更新サービスノード404は、親更新サービスノード402から認証および許可を受ける。親更新サービスノード402から認証および許可を受けることは、ソフトウェア更新の配信に対する制御の一要素をもたらし、許可された更新サービスノードだけに更新配信を制限する。認証および許可の技術は当技術分野で周知であり、任意の数の技術を使用して、親更新サービスノード402によって子更新サービスノード404を認証および許可することができる。本発明はいずれか1つの技術に限定されない。

【0051】

親更新サービスノード402による正しい認証および許可の後、イベント410で、親更新サービスノード402は、許可トークンを子更新サービスノード404に返す。一実施形態によれば、許可トークンは、限られた時間量にわたって親更新サービスノードと更新活動をさらに実施する許可を子更新サービスノード404に与える、時間に影響されるトークンである。したがって、子更新サービスノード404が親更新サービスノード402によって正しく認証および許可されない場合は、許可トークンは返されず、子更新サービスノードは、認証および許可を除く他のどんな更新関連の活動も実施することができない。同様に、更新トークンが失効した後は、子更新サービスノード404は、再認証および再許可を除くそれ以上のどんな更新関連の活動も親更新サービスノード402と実施することができない。

30

40

【0052】

許可トークンを受け取った後、イベント412で、子更新サービスノード404は、製品更新カタログを求める要求を許可トークンと共に親更新サービスノードに送信する。製品更新カタログは、親更新サービスノード402がソフトウェア更新を配信するソフトウェア製品のリストまたは目録を表す。

【0053】

本発明の態様によれば、子更新サービスノード404は、その親更新サービスノード402上で入手可能なすべてのソフトウェア更新を伝搬する必要はない。例えば、図1の例示的な更新配信システムを参照すると、企業更新サービスノード104は、ルート更新サービスノード102上で入手可能なソフトウェア製品の一部だけに対するサイトライセン

50

スしか有さない場合がある。このため、企業更新サービスノード104は、ルート更新サービスノード102で入手可能なソフトウェア更新のほとんどは使用しないので、これらすべてを入手する必要はない。したがって、更新サービスノード上の管理者は、どのソフトウェア製品更新が更新サービスノード上で入手可能かを選択的に確立することができる。ソフトウェア製品を選択することに加えて、管理者は、更新プロセスで消費されるネットワーク帯域幅の量を最小限に抑えるために、ソフトウェア製品に関して追加のより特定の特徴を選択的に識別することもできる。例えば、ソフトウェア製品を選択することに加えて、管理者は、選択したソフトウェア製品に対するソフトウェア更新の英語バージョンだけを要求することを選択的に指定することができる。同様に管理者は、特定の属性を有するソフトウェア更新だけをダウンロードすることを選択的に指定することもできる。これらの特定の属性の例として、管理者は、選択したソフトウェア製品に対して、セキュリティ関連のソフトウェア更新だけ、またはソフトウェアプロバイダ110がクリティカルと見なすソフトウェア更新だけを要求することができる。要約すると、本発明の態様によれば管理者は、ソフトウェア更新の任意の識別可能な属性または特徴に従って、選択したソフトウェア製品に対するソフトウェア更新のリストを要求することができる。

10

【0054】

本発明の一態様によれば、親更新サービスノード402から入手する更新製品カタログは、子更新サービスノード404が各製品の更新を配信するように構成されていてもいなくても、更新が入手可能であるソフトウェア製品すべてを識別する。しかし、本発明の代替の一態様によれば、親更新サービスノード402から入手する更新製品カタログは、要求元である子更新サービスノードが更新を配信するように構成されているソフトウェア製品だけを識別する。例えば、どのソフトウェア製品が製品更新カタログにリストされるかを制限することは、子更新サービスノード404が属する1つまたは複数のグループに従って決定することができる。

20

【0055】

イベント414で、親更新サービスノード402は、製品更新カタログを子更新サービスノード404に返す。イベント416で、子更新サービスノード404は、最新の更新が現在望まれる製品を製品更新カタログから選択する。製品更新カタログは、子更新サービスノード404が配信するソフトウェア製品だけをリストしたものとすることができるものの、子更新サービスノードは、異なるソフトウェア製品を異なる時にまたは異なる定期的スケジュールで入手するように構成できることに留意されたい。

30

【0056】

イベント418で、子更新サービスノード404は、子更新サービスノード404が現在更新を求めている選択された製品を識別する更新同期要求を、許可トークンと共に送信する。同期要求には、子更新サービスノード404上の製品に対して入手可能な最新の更新を識別する情報が含まれる。製品に対する最新の更新を識別する情報を、以下では「更新アンカー」と呼ぶ。各ソフトウェア製品についての更新アンカーは、通常は更新情報ストア216(図2)に記憶される。一実施形態では、更新アンカーは、改訂番号と改訂番号に関連する日付とを含む。

40

【0057】

更新同期要求に回答して、イベント420で、親更新サービスノード402は、子更新サービスノード404に入手可能な新しい更新があればどれが入手可能かを決定する。前述のようにこの決定は、特定のソフトウェア更新に関連する具体的な規則、子更新サービスノード404がメンバである1つまたは複数のグループ、ならびに更新アンカーに基づく。この例では、前述のように、前に受け取ったソフトウェア更新は子更新サービスノード404に対して明示的に許可されていない。このため、イベント406で受け取ったソフトウェア更新は、子更新サービスノード404に「入手可能」とであると判定されない。したがって、イベント422で、イベント406で受け取ったソフトウェア更新を識別せずに更新リストを子更新サービスノード404に返す。本発明の態様によれば、更新リストは、同期要求に従って、親更新サービスノード402上で「入手可能な」すべての更新

50

を識別する。一実施形態では、更新リストは、「入手可能な」各更新情報を、更新に関連する固有識別子で識別する。

【0058】

イベント424で、更新リストが空なので、すなわち親更新サービスノード402上で現在入手可能な更新がないので、子更新サービスノード404の更新プロセスは、単に所定時間量にわたって遅延するかスリープ状態になる。この例では、この遅延期間中にイベント426で、親更新サービスノード402における管理者は、イベント406で受け取ったソフトウェア更新が子更新サービスノード404に配信されるのを許可する。

【0059】

イベント428(図4B)で、子更新サービスノード404は再び自動更新プロセスを開始して、親更新サービスノード402から認証および許可を受ける。これに回答して、イベント430で、親更新サービスノード402は、許可トークンを子更新サービスノード404に返す。

【0060】

イベント432で、子更新サービスノード404は、製品更新カタログを求める要求を許可トークンと共に親更新サービスノード402に送信する。イベント434で、親更新サービスノード402は、製品更新カタログを子更新サービスノード404に返す。イベント436で、子更新サービスノード404は、更新が望まれる製品を更新カタログから選択する。イベント438で、子更新サービスノード404は、選択した製品を識別する更新同期要求を、許可トークンと共に送信する。

【0061】

前にイベント406で受け取ったソフトウェア更新を入手することが子更新サービスノード404に許可されているので、イベント440で、親更新サービスノード402は、ソフトウェア更新が子更新サービスノードに「入手可能」とであると判定し、対応する更新情報を更新リストに含める。したがってイベント442で、親更新サービスノード402は、イベント406で受け取ったソフトウェア更新を今や識別する更新リストを、子更新サービスノード404に返す。

【0062】

親更新サービスノード402上の「入手可能」な更新を識別する更新リストによって、子更新サービスノード404は今や、ソフトウェア更新を入手するのに必要な情報を有する。本発明の一実施形態によれば、子更新サービスノード404は、2つの部分でソフトウェア更新を親更新サービスノード402から入手する。すなわち、更新メタデータの入手と、更新内容またはファイル(以下では更新ペイロードと呼ぶ)の入手である。本発明の追加の態様によれば、更新メタデータは、ソフトウェア更新の関連特徴を記述する。限定しないがこれらの特徴には、更新を一意に識別する更新識別子、ソフトウェア更新に関連する改訂番号情報、ソフトウェア更新を優先事項と見なすべきかどうか、言語特有の情報、他のソフトウェア更新との関係、ダウンロードのための更新ペイロード位置、インストールハンドラルーチンなどが含まれる。

【0063】

ソフトウェア更新全体を2つの部分で、すなわち更新メタデータと更新ペイロードでダウンロードすることがしばしば有益である理由のいくつかには、更新ペイロードが更新メタデータよりもかなり大きい場合が多いこと、および、更新ペイロードが常にすぐ必要とされるとは限らないこと、すなわち必要とされるとしてもクライアントコンピュータへのインストールのためにすぐ必要とされるとは限らないことがある。したがって、本発明の一実施形態によれば、更新ペイロードは更新メタデータと別に、必要なときだけダウンロードする。当業者はこのダウンロード技法をレイジー(lazy)ダウンロード、あるいはジャストインタイムダウンロードとして理解するであろう。本発明の態様によれば、管理者は、更新ペイロードをジャストインタイム方式で入手するように、または更新メタデータを入手した後すぐに入手するように、更新サービスノードを構成することができる。さらに、代替の一実施形態では、更新メタデータと更新ペイロードの両方を一緒にダウン

10

20

30

40

50

ロードすることもできる。

【0064】

図4Bに示すように、更新リスト中で更新が識別されると、イベント444で、子更新サービスノード404は、「入手可能」なソフトウェア更新の更新メタデータを、更新リスト中のその固有識別子に従って要求する。親更新サービスノード402との他のほとんどの通信やりとりと同様、更新要求は許可トークンと共に送信する。図示の例ではすべての更新メタデータを1つのアクセスでダウンロードしているが、本発明の代替の態様（図示せず）では、更新メタデータを複数のアクセスでダウンロードすることもできることに留意されたい。例えば、第1のアクセスではまず、適用可能性規則（applicability rule）や他のソフトウェア更新に対する依存性など、ソフトウェア更新が適用可能かどうかおよび/または望ましいかどうかを判定するのに必要な更新メタデータの要素だけをまずダウンロードする。次いで、更新が適用可能および/または望ましいと判定された後で、残りの更新メタデータを入手することができる。これに回答して、イベント446で、親更新サービスノード402は、ソフトウェア更新の更新メタデータを子更新サービスノード404に返し、子更新サービスノード404は、この更新メタデータを更新情報ストア216に記憶する。

10

【0065】

任意選択で、イベント448で、子更新サービスノード404は、更新ペイロードを親更新サービスノード402からダウンロードする要求を送信する。これに回答して、イベント450で、親更新サービスノード402は、更新ペイロードを子更新サービスノード404に返し、子更新サービスノード404は、これを更新内容ストア214に記憶する。代替の一実施形態では、子更新サービスノード404は、更新メタデータ中で指定された、親更新サービスノード402ではない場合のある記憶位置から更新ペイロードをダウンロードする。

20

【0066】

今や子更新サービスノード404上で更新活動が行われたので、イベント452で、子更新サービスノードは、たった今行った更新活動を概説する更新報告を生成して、親更新サービスノード402に送信する。その後、子更新サービスノード404は、更新プロセスの実行が次に計画されるまで（図示せず）、再び遅延する。

【0067】

当業者なら理解するであろうが、前述のイベントは例示のためにすぎず、イベントおよび状況の例示的なセットの特定の1つを反映するものである。明らかだが、具体的な詳細および状況に従ってその他のイベントが行われ、それが前述のイベントの何らかの変形を引き起こす場合もある。加えて、子更新サービスノード404が最新の「入手可能」なソフトウェア更新を親更新サービスノード402から入手している間、子更新サービスノードは同時にその子更新サービスノードからの更新要求を処理している場合もあることを理解されたい。したがって、上の一連のイベントは例示にすぎず、本発明を限定するものではないと見なすべきである。

30

【0068】

図5は、図1の企業更新サービスノード104などの子更新サービスノード上で実行される、その親更新サービスノードから定期的に更新を入手するための例示的なルーチン500を示す流れ図である。ブロック502で開始し、子更新サービスノードは、「入手可能」な更新の同期済み更新リストを親更新サービスノードから入手する。「入手可能」な更新の同期済み更新リストを親更新サービスノードから入手することについて、図6に関して以下に述べる。

40

【0069】

図6は、図5の例示的なルーチン500で使用するのに適した、「入手可能」な更新の同期済み更新リストを親更新サービスノードから入手するための例示的なサブルーチン600の流れ図である。ブロック602で開始し、前に図4Aおよび4Bに関して論じたように、子更新サービスノードは親更新サービスノードから認証および許可を受け、正しい

50

認証および許可に回答して、許可トークンを受け取る。ブロック604で、子更新サービスノードは、許可トークンと共に、親更新サービスノードとの通信パラメータを確立する。通信パラメータを確立することにより、親と子の更新サービスノードは、親と子の両方が理解する共通基盤を正しく確立することができる。限定しないが通信パラメータには、通信更新プロトコルまたはバージョンや、製品分類などが含まれる。

【0070】

親更新サービスノードとの通信パラメータを確立した後、ブロック606で、子更新サービスノードは、親更新サービスノードが更新を提供/配信するソフトウェア製品を記述する製品更新カタログを入手する。ブロック608で、子更新サービスノードは、更新が現在求められているソフトウェア製品更新を選択する。ブロック610で、子更新サービスノードは、許可トークンと、選択したソフトウェア製品に関連し現在の改訂およびすでに子更新サービスノード上にある更新を識別する「アンカー」との両方を含む、更新同期要求を親更新サービスノードに送信する。

10

【0071】

更新同期要求に回答して、ブロック612で、子更新サービスノードは、子更新サービスノードに現在何が記憶されているかに応じて親更新サービスノード上の「入手可能」なソフトウェア更新に従って同期がとられた更新リストを、親更新サービスノードから入手する。前述のように更新リストは、子更新サービスノードに「入手可能」な親更新サービスノード402上のソフトウェア更新を、固有識別子で識別する。その後、例示的なサブルーチン600は終了する。

20

【0072】

再び図5を参照すると、同期済み更新リストを親更新サービスノードから入手した後、判定ブロック504で、いずれかのソフトウェア更新が現在、親更新サービスノードからダウンロードするのに「入手可能」かどうかを判定する。この判定は、同期済み更新リスト中にいずれかの更新識別子がリストされているかどうかに従って行う。ダウンロードするのに現在「入手可能」なソフトウェア更新がない場合は、例示的なルーチン500は遅延ブロック510に進み、次の更新期間が発生するまで遅延/スリープ状態になる。あるいは、親更新サービスノードからダウンロードするのに「入手可能」な更新がある場合は、ブロック506で、子更新サービスノードは親更新サービスノードから更新を入手する。「入手可能」な更新を親更新サービスノードから入手することについて、図7に関して以下に述べる。

30

【0073】

図7は、図5の例示的なルーチン500で使用するのに適した、「入手可能」な更新を親更新サービスノードから入手するための例示的なサブルーチン700の流れ図である。ブロック702で開始し、更新リスト中の第1の更新識別子を選択する。ブロック704で、子更新サービスノードは、選択した更新識別子に対応する更新メタデータを親更新サービスノードから入手し、更新情報ストア216に記憶する。

【0074】

一実施形態によれば、ブロック706で、子更新サービスノードは、選択した更新識別子に対応する更新ペイロードを親更新サービスノードから入手して、更新内容ストア212に記憶する。任意選択で、更新内容はすぐに子更新サービスノードにダウンロードしなくてもよい。前述のように、子更新サービスノードは選択的に、更新を親更新サービスノードからジャストインタイム方式でダウンロードするように構成することができる。このオプション処理によれば、図7に示すように、例示的なサブルーチン700はブロック704からブロック706に進むのではなく、任意選択でブロック704から判定ブロック708に進む。

40

【0075】

選択した更新識別子に対する更新メタデータ、および任意選択で更新ペイロードを入手した後、判定ブロック708で、更新リスト中に追加の更新識別子があるかどうかを判定する。追加の更新識別子がある場合は、ブロック710で、更新リスト中の次の更新識別

50

子を選択し、サブルーチン700は追加の処理のためにブロック704に戻る。ルーチン700は、判定ブロック708で更新リスト中に更新識別子がこれ以上ないと判定されるまで継続する。更新識別子がこれ以上ないと判定されると、例示的なサブルーチン700は終了する。

【0076】

再び図5に戻ると、子更新サービスノードは、「入手可能」な更新を親更新サービスノードから入手した後、ブロック508で、更新活動を親更新サービスノードに報告する。その後、遅延ブロック510で、例示的なルーチン500は、次の更新期間まで所定時間量にわたって遅延ノスリープ状態になり、次いでブロック502に進み、上に識別した更新手順を繰り返す。

10

【0077】

図5に示すように、判定ブロック504で、親更新サービスノード上に「入手可能」な更新がないときでも、子更新サービスノードは任意選択で、その更新活動を親更新サービスノード報告するように構成することができる。この代替構成によれば、入手可能な更新がないとき、例示的なルーチン500はブロック508に進んで更新活動を報告することができる。

【0078】

図8は、親更新サービスノード上で実施される、子更新サービスノードからの更新同期要求に回答して「入手可能」な更新を識別する同期済み更新リストを生成するための例示的なルーチン800の流れ図である。ブロック802で開始し、親更新サービスノードは、「入手可能」な更新を識別する更新リストを求める更新同期要求を、子更新サービスノードから受け取る。ブロック804で、更新同期要求中で識別される第1のソフトウェア製品を選択する。

20

【0079】

判定ブロック806で、識別されるソフトウェア製品に対する入手可能な更新があるかどうかを判定する。この判定は、更新情報ストア216に記憶されたソフトウェア製品に関するメタデータに従って、また、子更新サービスノードから提供された更新アンカーに従って、また、子更新サービスノードが属するグループに関連する配信規則に従って行う。この判定に従って、「入手可能」な更新がある場合は、ブロック808で、「入手可能」な更新に関連する固有の更新識別子を更新リストに書き込む。「入手可能」な更新の固有の更新識別子を更新リストに書き込んだ後、判定ブロック810で、更新同期要求中で識別される追加のソフトウェア製品がまだあるかどうかを判定する。更新同期要求中に追加のソフトウェア製品がある場合は、ブロック814で、親更新サービスノードは、更新同期要求中で識別される次のソフトウェア製品を選択し、判定ブロック806に戻って、選択したソフトウェア製品に対する「入手可能」な更新があるかどうかを判定する。あるいは、更新同期要求中で識別されるソフトウェア製品がそれ以上ない場合は、ブロック814で、更新リストを子更新サービスノードに返す。その後、例示的なサブルーチン800は終了する。

30

【0080】

好ましい実施形態を含めて、本発明の様々な実施形態を図示および記述したが、本発明の趣旨および範囲を逸脱することなく様々な変更をこれに加えることができることを理解されたい。

40

【0081】

独占所有または特権を請求する本発明の実施形態は、特許請求の範囲に定義する。

【図面の簡単な説明】**【0082】**

【図1】本発明の態様により形成された例示的な更新配信システムを示す図である。

【図2】本発明の態様により形成された更新サービスノードの例示的な論理コンポーネントを示すブロック図である。

【図3】本発明の態様により形成されたルート更新サービスノードの例示的な論理コンポ

50

ーネットを示すブロック図である。

【図4A】本発明の態様による、ソフトウェア更新を親更新サービスノードから子更新サービスノードに提供する際の、親更新サービスノードと子更新サービスノードとの間の例示的なやりとりを示すブロック図である。

【図4B】本発明の態様による、ソフトウェア更新を親更新サービスノードから子更新サービスノードに提供する際の、親更新サービスノードと子更新サービスノードとの間の例示的なやりとりを示すブロック図である。

【図5】子更新サービスノード上で実行される、その親更新サービスノードから定期的に更新を入手するための例示的なルーチンを示す流れ図である。

【図6】図5の例示的なルーチンで使用するのに適した、更新カタログを親更新サービスノードから入手するための例示的なサブルーチンの流れ図である。

【図7】図5の例示的なルーチンで使用するのに適した、ソフトウェア更新を親更新サービスノードから入手するための例示的なサブルーチンの流れ図である。

【図8】子更新サービスノードからの更新要求を処理するための例示的なルーチンの流れ図である。

【符号の説明】

【0083】

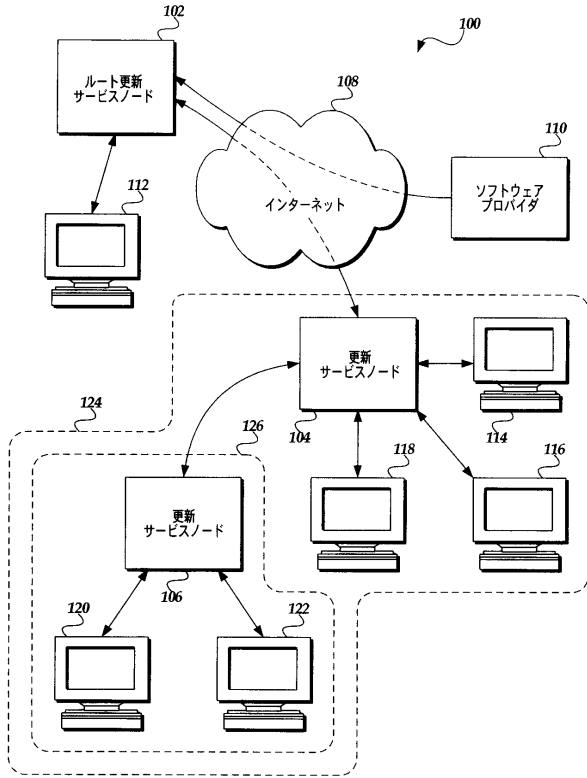
- 102 ルート更新サービスノード
- 104 更新サービスノード
- 106 更新サービスノード
- 108 インターネット
- 110 ソフトウェアプロバイダ
- 202 更新ウェブサービス
- 204 クライアント更新モジュール
- 206 子更新モジュール
- 208 報告モジュール
- 210 認証/許可モジュール
- 212 管理API
- 214 更新内容
- 216 更新情報
- 218 管理ユーザインタフェース
- 302 ソフトウェアプロバイダインタフェース

10

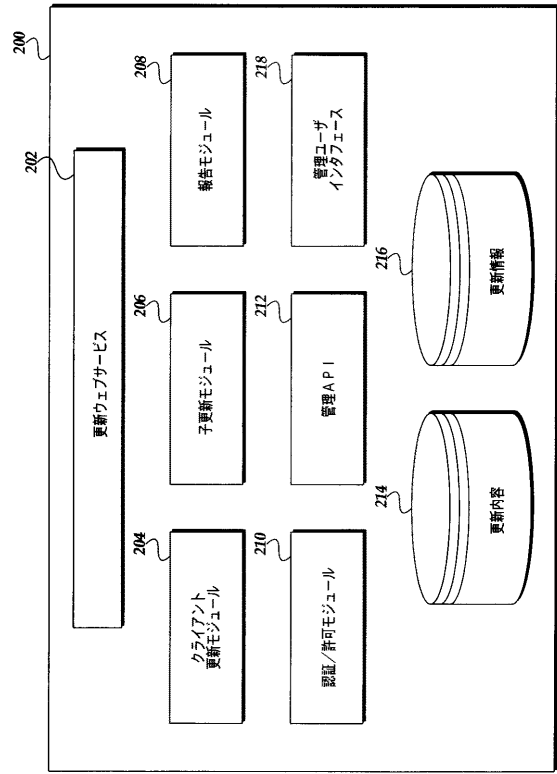
20

30

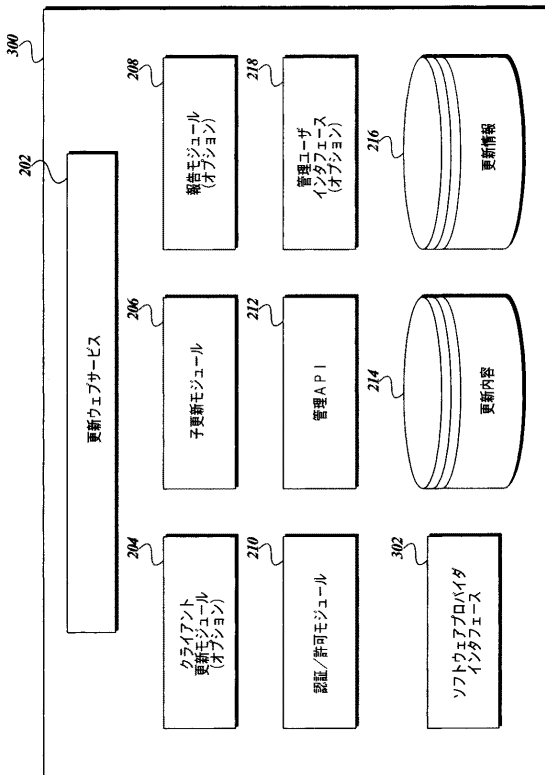
【図1】



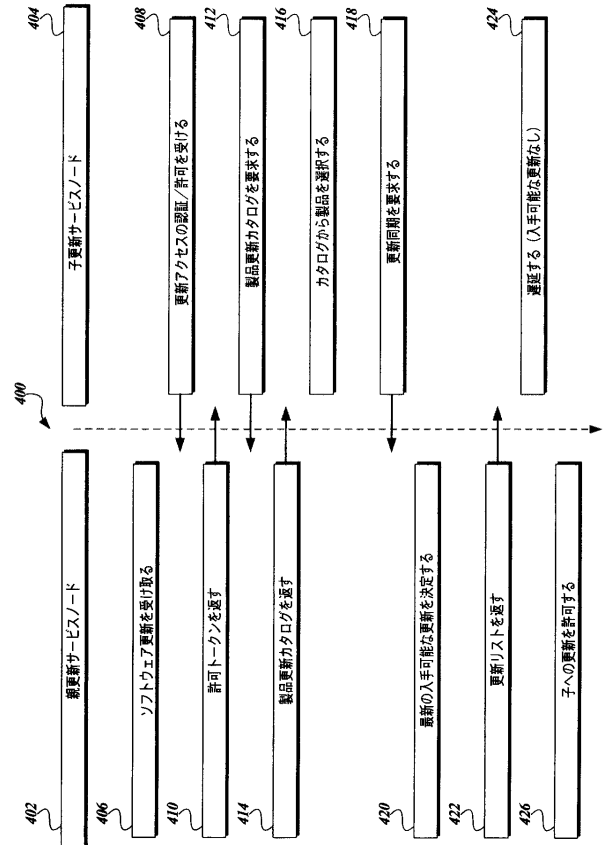
【図2】



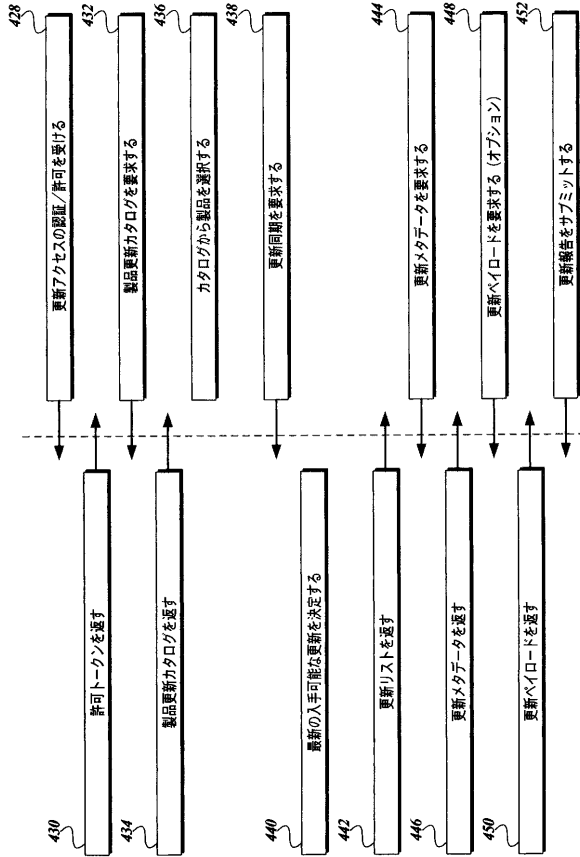
【図3】



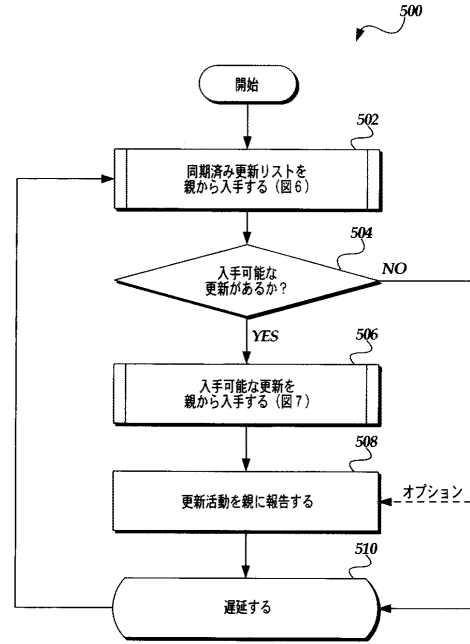
【図4A】



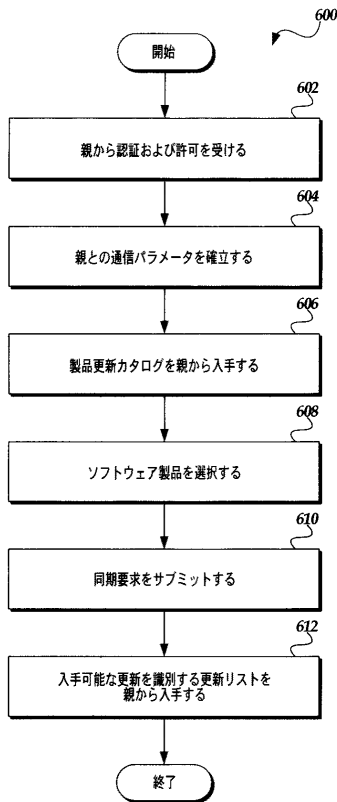
【図4B】



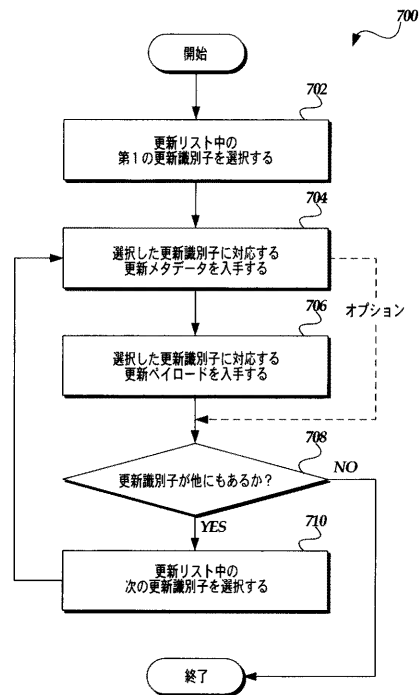
【図5】



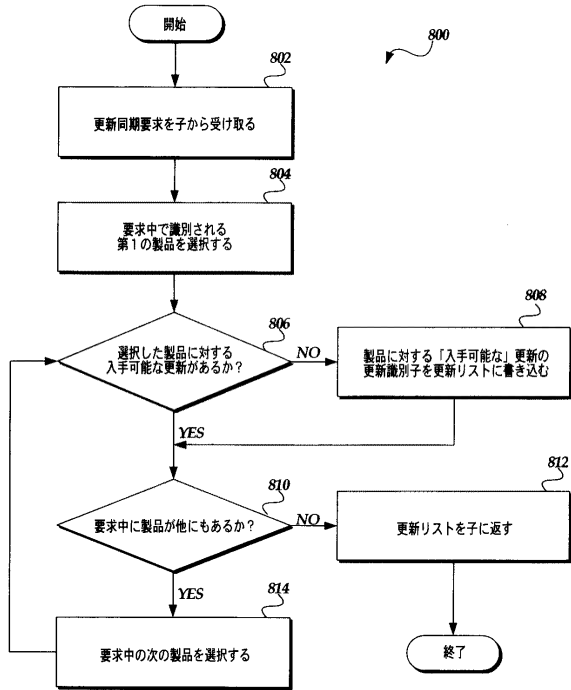
【図6】



【図7】



【図8】



フロントページの続き

- (72)発明者 デビッド シー . ヘネシー
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 エドワード エフ . レウス
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 ジェyson ジェイ . ロバーツ
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 チアンポー フー
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 リー チャールズ マ
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 マーク シェパード
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 マジャーラ エヌ . モハメド
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 ニルマル アール . ソイ
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 トータム アール . スリラム
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 バラダラジャン エス . シルピラマライ
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 ビジャサンカル ラマン
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内
- (72)発明者 シャン ヘ
 アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
 イクロソフト コーポレーション内

審査官 林 毅

- (56)参考文献 特開2003 - 348515 (JP, A)
 特開2002 - 032284 (JP, A)
 特開2002 - 007255 (JP, A)
 特開平10 - 269085 (JP, A)
 特表平06 - 509430 (JP, A)
 特開2003 - 150342 (JP, A)
 特開2001 - 265603 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 9 / 4 4 5
G 0 6 F 1 3 / 0 0