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

特許第4914220号
(P4914220)

(45)発行日　平成24年4月11日(2012.4.11)　(24)登録日　平成24年1月27日(2012.1.27)

(51)Int.Cl.　　F I
G06F 11/00 (2006.01)　G06F 9/06 630B
G06F 9/445 (2006.01)　G06F 9/06 610Q
G06F 13/00 (2006.01)　G06F 13/00 530B

請求項の数 19　(全 38 頁)

(21)出願番号　特願2006-543794 (P2006-543794)
(86)(22)出願日　平成16年7月23日 (2004.7.23)
(65)公表番号　特表2007-514233 (P2007-514233A)
(43)公表日　平成19年5月31日 (2007.5.31)
(86)国際出願番号　PCT/US2004/023975
(87)国際公開番号　WO2005/060388
(87)国際公開日　平成17年7月7日 (2005.7.7)
審査請求日　平成19年7月23日 (2007.7.23)
(31)優先権主張番号　10/737,708
(32)優先日　平成15年12月15日 (2003.12.15)
(33)優先権主張国　米国(US)

前置審査

(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　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　110001243
特許業務法人　谷・阿部特許事務所
(74)復代理人　100115624
弁理士　濱中　淳宏
(74)復代理人　100156971
弁理士　稲　綾子

最終頁に続く

(54)【発明の名称】ソフトウェアアップデートのための通信方法およびシステム

(57)【特許請求の範囲】
【請求項１】
サーバコンピューティングデバイスからクライアントコンピューティングデバイスにソフトウェアアップデートを提供するための方法であって、
（ａ）前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスから、前記クライアントコンピューティングデバイス上に記憶されているクライアント許可モジュールの識別子を含む許可要求を取得するステップと、
（ｂ）前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスが特定のターゲットグループに関連付けられているかどうかを判定するステップであって、前記クライアント許可モジュールが、前記クライアントコンピューティングデバイスが前記特定のターゲットグループに関連付けられていることを示すとき、前記クライアントコンピューティングデバイスは、前記特定のターゲットグループに関連付けられていると判定されるステップと、
（ｃ）前記クライアントコンピューティングデバイスが前記特定のターゲットグループに関連付けられていると判定された場合に、前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスに前記特定のターゲットグループを識別するサーバクッキーを送信するステップと、
（ｄ）前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスから、前記サーバコンピューティングデバイス上に記憶されている少なくとも１つのソフトウェアアップデートを求める要求を受信するステップであって、前記要求は

、前記サーバクッキーを含み、前記要求は、前記クライアントコンピューティングデバイスがインストール済みのソフトウェアアップデートを記憶しているときは、該インストール済みのソフトウェアアップデートの識別子をさらに含み、前記クライアントコンピューティングデバイスが失敗したソフトウェアアップデートを記憶しているときは、該失敗したソフトウェアアップデートの識別子をさらに含む、ステップと、

（ｅ）前記少なくとも１つのソフトウェアアップデートを求める前記要求を受信したことに応答して、前記サーバコンピューティングデバイスが、該サーバコンピューティングデバイスが記憶している複数のソフトウェアアップデートのうち、前記サーバクッキーで識別される前記特定のターゲットグループに関連するソフトウェアアップデートを決定するステップであって、前記複数のソフトウェアアップデートはそれぞれ、該ソフトウェアアップデートをインストールするための少なくとも１つの条件を定義する適用規則と、該ソフトウェアアップデートをインストールする前に別のソフトウェアアップデートをインストールする必要があるときに該別のソフトウェアアップデートを識別する前提条件と、該ソフトウェアアップデートをインストールした後にさらにインストールすべき追加のソフトウェアアップデートがあるかどうかを示すコードとを含む命令コンポーネントを含む、ステップと、

（ｆ）前記サーバコンピューティングデバイスが、前記複数のソフトウェアアップデートのうち前記特定のターゲットグループに関連すると決定されたソフトウェアアップデートから、前記クライアントコンピューティングデバイスに提供するためのソフトウェアアップデートを選択し、該選択されたソフトウェアアップデートの前記命令コンポーネントを、前記クライアントコンピューティングデバイスに送信するステップであって、前記要求が、少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含まないとき、前記前提条件を含まない前記命令コンポーネントを有するソフトウェアアップデートを、前記クライアントコンピューティングデバイスに提供するためのソフトウェアアップデートとして選択し、前記要求が、少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含むとき、該少なくとも１つのインストール済みのソフトウェアアップデートの識別子に基づいて前記クライアントコンピューティングデバイスが各前提条件で示される前記別のソフトウェアアップデートを記憶していると判断したときに、該前提条件を含む前記命令コンポーネントを有するソフトウェアアップデートを、前記クライアントコンピューティングデバイスに提供するためのソフトウェアアップデートとして選択する、ステップと、

（ｇ）前記クライアントコンピューティングデバイスが、前記選択されたソフトウェアアップデートの前記命令コンポーネントを処理して、前記命令コンポーネント内の前記適用規則によって定義される前記条件を満たすかどうか判断し、前記クライアントコンピューティングデバイスが前記条件を満たさないとき、該条件を有する命令コンポーネントを、失敗したソフトウェアアップデートとして記憶するステップと、

（ｈ）前記クライアントコンピューティングデバイスが前記条件を満たすとき、前記命令コンポーネントの前記コードが、前記追加のソフトウェアアップデートがあることを示しているかどうかを判断するステップと、

（ｉ）前記命令コンポーネントの前記コードが前記追加のソフトウェアアップデートがないことを示している場合に、前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスから前記選択されたソフトウェアアップデートの局所的データコンポーネントを求める要求を受信したとき、前記選択されたソフトウェアアップデートの局所的データコンポーネントを、前記クライアントコンピューティングデバイスに送信するステップであって、前記局所的データコンポーネントは、前記選択されたソフトウェアアップデートの特徴と利点を記述する情報、および前記選択されたソフトウェアアップデートのインストールに関するテキストを含む、ステップと

（ｊ）前記クライアントコンピューティングデバイスが、前記選択されたソフトウェアアップデートの特徴と利点を記述する前記情報を、前記テキストに応じたメッセージとともにディスプレイに表示することにより、前記選択されたソフトウェアアップデートが前

記クライアントコンピューティングデバイスにインストール可能であることをユーザに通知するステップと
を含むことを特徴とする方法。
【請求項２】
前記許可要求は、以前に記憶されたサーバクッキーのデータを含み、前記サーバコンピューティングデバイスが前記クライアントコンピューティングデバイスに送信する前記サーバクッキーは、前記以前に記憶されたサーバクッキーの前記データを含むことを特徴とする請求項１に記載の方法。
【請求項３】
前記サーバコンピューティングデバイスのサーバ許可モジュールが、前記クライアント許可モジュールによって識別される前記特定のターゲットグループと同じターゲットグループを識別するとき、前記クライアントコンピューティングデバイスは、前記特定のターゲットグループに関連付けられていると判定されることを特徴とする請求項１に記載の方法。
【請求項４】
前記サーバクッキーは、有効期限時間を含み、
前記方法は、前記サーバクッキーの前記有効期限時間が経過しているかどうかを判定するステップをさらに含み、
前記サーバコンピューティングデバイスが、前記選択されたソフトウェアアップデートの前記命令コンポーネントを前記クライアントコンピューティングデバイスに送信するステップは、前記サーバクッキーの前記有効期限時間が経過していないと判定されたときにのみ実行されることを特徴とする請求項１に記載の方法。
【請求項５】
前記サーバクッキーは、クリアテキストフォーマットおよび暗号化されたフォーマットで前記有効期限時間を記憶し、前記サーバクッキーは、少なくとも１つのターゲットグループを識別する暗号化されたデータを含み、前記方法は、
前記クライアントコンピューティングデバイスが、前記サーバクッキーから前記クリアテキストフォーマットで記憶された前記有効期限時間を取得するステップと、
前記クライアントコンピューティングデバイスが、前記有効期限時間が経過していないかどうかを判定するステップと、
前記有効期限時間が経過していないと判定された場合に、前記クライアントコンピューティングデバイスが、前記サーバクッキーを前記少なくとも１つのソフトウェアアップデートを求める前記要求によって前記サーバコンピューティングデバイスに送信するステップと
をさらに含むことを特徴とする請求項１に記載の方法。
【請求項６】
前記クライアント許可モジュールは、ソフトウェアアプリケーションのインストレーション中に前記クライアントコンピューティングデバイス上に記憶されることを特徴とする請求項１に記載の方法。
【請求項７】
前記クライアント許可モジュールは、ソフトウェアアプリケーションのアップデートプロセス中に前記クライアントコンピューティングデバイス上に記憶されることを特徴とする請求項１に記載の方法。
【請求項８】
前記サーバ許可モジュールは、前記ソフトウェアアップデートサービスから動的に追加または除去するように構成されるソフトウェアプラグインであることを特徴とする請求項１に記載の方法。
【請求項９】
前記少なくとも１つのソフトウェアアップデートを求める前記要求は、前記クライアントコンピューティングデバイス上で実行される自動アップデートプログラムモジュールか

ら生成されることを特徴とする請求項１に記載の方法。
【請求項１０】
前記少なくとも１つのソフトウェアアップデートを求める前記要求は、手動アップデートを要求するユーザによって引き起こされることを特徴とする請求項１に記載の方法。
【請求項１１】
サーバコンピューティングデバイスからクライアントコンピューティングデバイスにソフトウェアアップデートを送信するための方法であって、
（ａ）前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスが特定のターゲットグループに関連付けられているかどうかを判定するステップであって、前記クライアントコンピューティングデバイスが、前記クライアントコンピューティングデバイスが前記特定のターゲットグループに関連付けられていることを示すクライアント許可モジュールを含むとき、前記クライアントコンピューティングデバイスは、前記特定のターゲットグループに関連付けられていると判定されるステップと、
（ｂ）前記クライアントコンピューティングデバイスが前記特定のターゲットグループに関連付けられていると判定された場合に、前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスに前記特定のターゲットグループを識別するサーバクッキーを送信するステップと、
（ｃ）前記クライアントコンピューティングデバイスが、前記サーバコンピューティングデバイスに、前記サーバコンピューティングデバイス上に記憶されている少なくとも１つのソフトウェアアップデートを求める要求を送信するステップであって、前記要求は、前記サーバクッキーを含み、前記要求は、前記クライアントコンピューティングデバイスが失敗したソフトウェアアップデートを記憶しているとき、該失敗したソフトウェアアップデートの識別子をさらに含む、ステップと、
（ｄ）前記少なくとも１つのソフトウェアアップデートを求める前記要求を受信したことに応答して、前記サーバコンピューティングデバイスが、該サーバコンピューティングデバイスが記憶している複数のソフトウェアアップデートのうち、前記サーバクッキーで識別される前記特定のターゲットグループに関連するソフトウェアアップデートを決定するステップであって、前記複数のソフトウェアアップデートはそれぞれ、該ソフトウェアアップデートをインストールするための少なくとも１つの条件を定義する適用規則と、該ソフトウェアアップデートをインストールする前に別のソフトウェアアップデートをインストールする必要があるときに該別のソフトウェアアップデートを識別する前提条件と、該ソフトウェアアップデートをインストールした後にさらにインストールすべき追加のソフトウェアアップデートがあるかどうかを示すコードとを含む命令コンポーネントを含む、ステップと、
（ｅ）前記サーバコンピューティングデバイスが、前記複数のソフトウェアアップデートのうち前記特定のターゲットグループに関連すると決定されたソフトウェアアップデートから、前記クライアントコンピューティングデバイスに提供するためのソフトウェアアップデートを選択し、該選択されたソフトウェアアップデートの前記命令コンポーネントを前記クライアントコンピューティングデバイスに送信するステップであって、前記要求が少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含まないときは、前記前提条件を含まない前記命令コンポーネントを有するソフトウェアアップデートを前記クライアントコンピューティングデバイスに提供するためのソフトウェアアップデートとして選択し、前記要求が少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含むとき、該少なくとも１つのインストール済みのソフトウェアアップデートの識別子に基づいて前記クライアントコンピューティングデバイスが各前提条件で示される前記別のソフトウェアアップデートを記憶していると判断したときに、該前提条件を含む前記命令コンポーネントを有するソフトウェアアップデートを前記クライアントコンピューティングデバイスに提供するためのソフトウェアアップデートとして選択する、ステップと、
（ｆ）前記クライアントコンピューティングデバイスが、前記選択されたソフトウェア

アップデートの前記命令コンポーネントを処理して、前記命令コンポーネント内の前記適用規則によって定義される前記条件を満たすかどうか判断し、前記クライアントコンピューティングデバイスが前記条件を満たさないとき、該条件を有する命令コンポーネントを、失敗したソフトウェアアップデートとして記憶するステップと、

（ｇ）前記クライアントコンピューティングデバイスが前記条件を満たすとき、前記命令コンポーネントの前記コードが、前記追加のソフトウェアアップデートがあることを示しているかどうかを判断するステップと、

（ｈ）前記命令コンポーネントの前記コードが前記追加のソフトウェアアップデートがないことを示しているとき、前記クライアントコンピューティングデバイスが、前記選択されたソフトウェアアップデートの局所的データコンポーネントを、前記サーバコンピューティングデバイスに要求するステップと、

（ｉ）前記サーバコンピューティングデバイスが、前記選択されたソフトウェアアップデートの局所的データコンポーネントを前記クライアントコンピューティングデバイスに送信するステップであって、前記局所的データコンポーネントは、前記選択されたデータソフトウェアアップデートの特徴と利点を記述する情報、および前記選択されたソフトウェアアップデートのインストールに関するテキストを含む、ステップと

（ｊ）前記クライアントコンピューティングデバイスが、前記選択されたソフトウェアアップデートの特徴と利点を記述する前記情報を、前記テキストに応じたメッセージとともにディスプレイに表示することにより、前記選択されたソフトウェアアップデートが前記クライアントコンピューティングデバイスにインストール可能であることをユーザに通知するステップと

を含むことを特徴とする方法。

【請求項１２】

前記サーバコンピューティングデバイスが、前記クライアントコンピューティングデバイスから前記クライアントコンピューティングデバイスに含まれる前記クライアント許可モジュールの識別子を含む許可要求を取得するステップをさらに含むことを特徴とする請求項１１に記載の方法。

【請求項１３】

前記サーバクッキーは、有効期限時間を含み、

前記方法は、前記サーバクッキーの前記有効期限時間が経過しているかどうかを判定するステップをさらに含み、

前記サーバコンピューティングデバイスが、前記選択されたソフトウェアアップデートの前記命令コンポーネントを前記クライアントコンピューティングデバイスに送信するステップは、前記サーバクッキーの前記有効期限時間が経過していないと判定されたときにのみ実行されることを特徴とする請求項１１に記載の方法。

【請求項１４】

前記サーバクッキーは、クリアテキストフォーマットおよび暗号化されたフォーマットで前記有効期限時間を記憶し、前記サーバクッキーは、少なくとも１つのターゲットグループを識別する暗号化されたデータを含み、前記方法は、

前記クライアントコンピューティングデバイスが、前記サーバクッキーから前記クリアテキストフォーマットで記憶された前記有効期限時間を取得するステップと、

前記クライアントコンピューティングデバイスが、前記有効期限時間が経過していないかどうかを判定するステップと、

前記有効期限時間が経過していないと判定された場合に、前記クライアントコンピューティングデバイスが、前記サーバクッキーを前記少なくとも１つのソフトウェアアップデートを求める前記要求によって前記サーバコンピューティングデバイスに送信するステップと

をさらに含むことを特徴とする請求項１１に記載の方法。

【請求項１５】

クライアントコンピューティングデバイスとサーバコンピューティングデバイスとの間

で通信する方法であって、
（ａ）前記クライアントコンピューティングデバイスが、前記サーバコンピューティングデバイスに、前記クライアントコンピューティングデバイスが属するターゲットグループを識別するクライアント許可モジュールの識別子を含む許可要求を発行するステップと、
（ｂ）前記クライアントコンピューティングデバイスが、前記サーバコンピューティングデバイスから前記ターゲットグループを識別するサーバクッキーを受信するステップと、
（ｃ）前記クライアントコンピューティングデバイスが、前記サーバコンピューティングデバイスに、ソフトウェアアップデートを求めるアップデート要求を発行するステップであって、前記アップデート要求は、前記サーバクッキーを含む、ステップと、
（ｄ）前記クライアントコンピューティングデバイスが、前記ターゲットグループに関連付けられた少なくとも１つのソフトウェアアップデートの命令コンポーネントを受信するステップであって、各命令コンポーネントは、該命令コンポーネントに対応するソフトウェアアップデートを前記クライアントコンピューティングデバイスがインストールするための条件を定義する少なくとも１つの適用規則を含む、ステップと、
（ｅ）前記クライアントコンピューティングデバイスが、前記少なくとも１つのソフトウェアアップデートのうち、前記命令コンポーネント内の前記少なくとも１つの適用規則によって定義される前記条件を満たすと判断した特定のソフトウェアアップデートの局所的データコンポーネントを、前記サーバコンピューティングデバイスに要求するステップと、
（ｆ）前記クライアントコンピューティングデバイスが、前記サーバコンピューティングデバイスから前記特定のソフトウェアアップデートの前記局所的データコンポーネントを受信するステップであって、前記局所的データコンポーネントは、前記特定のソフトウェアアップデートの特徴と利点を記述する情報、および前記特定のソフトウェアアップデートのインストールに関するテキストを含む、ステップと、
（ｇ）前記クライアントコンピューティングデバイスが、前記特定のソフトウェアアップデートの特徴と利点を記述する前記情報を、前記テキストに応じたメッセージとともにディスプレイに表示することにより、前記特定のソフトウェアアップデートが前記クライアントコンピューティングデバイスにインストール可能であることをユーザに通知するステップと
を含むことを特徴とする方法。
【請求項１６】
前記許可要求は、以前に記憶されたサーバクッキーのデータを含み、前記サーバコンピューティングデバイスによって発行された前記サーバクッキーは、前記以前に記憶されたサーバクッキーの前記データを含むことを特徴とする請求項１５に記載の方法。
【請求項１７】
前記サーバコンピューティングデバイスのサーバ許可モジュールが、前記クライアント許可モジュールの識別子によって識別される前記ターゲットグループと同じターゲットグループを識別する場合に、前記クライアントコンピューティングデバイスが前記ターゲットグループに関連すると判定されることを特徴とする請求項１５に記載の方法。
【請求項１８】
前記サーバクッキーは、有効期限時間を含み、
前記少なくとも１つのソフトウェアアップデートの命令コンポーネントを受信するステップは、前記アップデート要求内の前記サーバクッキーの前記有効期限時間が経過していない場合にのみ前記少なくとも１つのソフトウェアアップデートの命令コンポーネントを受信すること含むことを特徴とする請求項１５に記載の方法。
【請求項１９】
前記サーバクッキーは、前記有効期限時間をクリアテキストフォーマットおよび暗号化されたフォーマットで記憶し、前記サーバクッキーは、少なくとも１つのターゲットグル

ープを識別する暗号化されたデータを含み、前記方法は、
前記クライアントコンピューティングデバイスが、前記サーバクッキーから前記クリアテキストフォーマットで記憶された前記有効期限時間を取得するステップをさらに含むことを特徴とする請求項１８に記載の方法。
【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、詳細には、本発明は、ソフトウェアアップデートを管理し伝えるためのシステムおよび方法に関する。
【背景技術】
【０００２】
ほとんどの市販のソフトウェア製品は、継続的なリビジョンプロセスを受けて機能および／またはファンクションを修復し、またはアップグレードする。ソフトウェア製品またはソフトウェアコンポーネントの各リビジョンは、新しいファイルの追加および／または既存ファイルを新しいバージョンのファイルと置換することを必要とすることもある。ベンダが、ソフトウェア製品の問題を分離しその問題についての解決方法を作成した後に、ベンダは、その解決方法をアップデート中へと修正し、このアップデートを顧客に広く使用可能にしたいと望むはずである。ソフトウェアベンダは、ソフトウェアアップデートをできる限り速やかに、また問題なく顧客に対して配信すべきビジネスインセンティブ（ｂｕｓｉｎｅｓｓ　ｉｎｃｅｎｔｉｖｅ）を有する。
【０００３】
インターネットは、顧客がソフトウェア製品についての最終アップデートを取得するための重要なチャネルを提供する。インターネットの使用の爆発的な増加は、ソフトウェア製品およびアップデートがオンラインでダウンロードするために提供されるという、顧客による一般的な期待を生み出してきている。インターネットを使用してアップデートを配信することを推進することは、それがソフトウェアベンダのコストを低減し、その修正版がダウンロードのために使用可能になるとすぐに顧客が識別された問題についてのその修正版を取得できるようになるので、ソフトウェアベンダのためにもなる。インターネット上のこれらのベンダサイトは、あるアプリケーションについてのアップデートファイルを見出し、その所在を突き止めることを非常に簡単にするように設計することが可能である。ファイルダウンロードの技術的側面は、ユーザの観点からはほとんど見えなくなってしまっており、今では一般にオペレーティングシステムによって処理される。
【０００４】
従来のアプローチでは、ソフトウェアベンダは、ソフトウェアアップデートをダウンロードのための「パッケージ」として構築する。このパッケージは、一般にその設定プログラムおよび各製品アップデートファイルが、埋め込まれ、このパッケージをより小さなパッケージにするために圧縮された自己展開（ｓｅｌｆ－ｅｘｔｒａｃｔｉｎｇ、自己解凍）実行可能ファイルである。このパッケージのサイズは、一般に各変更ファイルの圧縮されたサイズとその抽出コードそれ自体のサイズを加えた合計である。実行すると、このパッケージは含まれている各ファイルを一時的なロケーションへと抽出し、次いでその設定プログラムを開始して各ファイルをシステムディレクトリ中の適切なロケーションにインストールする。圧縮形式で発送されたファイルは、ファイルがインストールされるときに伸張される。同じロケーション中の同じ名前の既存ファイルはどれも、この置換ファイルによって単に上書きされるはずである。
【０００５】
インターネットがソフトウェアアップデートの広範で速やかな配信を可能にしているにもかかわらず、ネットワーク伝送のその限られた帯域幅が問題を引き起こしてきている。大規模な一般のソフトウェアアプリケーションが、アップデートのダウンロードサイズを不当なほど大きくなるようにしてきている。通常は、製品の様々な問題についての多数の修正が、１つのアップデート中にグループ化されることになる。ベンダが定期的にソフト

ウェア製品をアップデートする場合、このアップデートパッケージのダウンロードサイズは、ユーザがすでに以前のアップデートからこれらのファイルを有しているという想定の下ではこのベンダがファイルを省略することができないので、増大し続けることになる。アップデートパッケージは、いくつかの全体ファイルを結合するので、たとえこれらのファイルが圧縮されている場合でさえ、非常に大きくなってしまうこともある。時には、最高速のモデム接続上でさえ、このダウンロードの帯域幅効率は低下する。

【０００６】

従来のダウンロードプロセスの時間を浪費する側面は、もちろん望ましくない。一部の場合においては、顧客は、これらのファイルのダウンロード中に長距離料金または接続時間料金を支払うこともある。接続時間のどのような短縮も、これらの顧客にとっての直接的な金銭的コストを低下させることになる。ベンダには一般に、ベンダが提供するダウンロードのサイズに関連した何らかの区別できるコストがあり、したがって、サイズを低減することは同様にベンダにとっても直接的な金銭的な利点をもたらす。ダウンロードのサイズを低減することは、その使用可能なネットワーク帯域幅を増大させることになり、ベンダは、既存のネットワークサーバ装置を用いて、より多くの顧客を満足させることができるようになる。

【０００７】

大規模なアップデートをダウンロードするためにかかる長い時間は、様々なネットワーク接続問題に対してこのダウンロードプロセスをよりぜい弱にもする。電話回線ノイズ、キャッチホン信号、および意図的ではないコマンドを含めて、インターネットセッションが時期尚早にも切断されてしまう可能性がある理由がいくつかある。一部のインターネットサービスプロバイダは、接続時間制限を強制し、ユーザが単一セッション中でオンラインとなることができる時間長が制限される。ネットワーク接続が遮断される場合にそのユーザが大規模なファイルをダウンロードしている場合には、ユーザは、初めからやり直す必要があり得る。ほとんどの一般のオペレーティングシステムおよびファイル転送プロトコルでは、このファイル転送を再開できるようにはなっておらず、したがって、任意の中間的進捗状況が失われてしまうはずであり、この転送は再開される必要がある。エラーの可能性は非常に多いので、多くのユーザには、オンラインでそのアップデートを取得することがほとんど不可能であることが分かる。アップデートパッケージのサイズがあまりにも大きすぎる場合には、ユーザが、アップデートパッケージを完全にダウンロードすることが決してできないこともある。

【０００８】

ソフトウェアアップデートのサイズを低減させ、帯域幅効率を増大させる１つの試みが、デルタパッチ（ｄｅｌｔａ　ｐａｔｃｈ）またはバイナリパッチ（ｂｉｎａｒｙ　ｐａｔｃｈ）の使用に関する。デルタパッチが、コンピューティングデバイスによって実行される場合に既存のファイルを修正する専用ソフトウェアコードに対応することが当業者には理解されよう。このデルタパッチが専用ソフトウェアコードを含むので、一意のデルタパッチは、ファイルの一意のバージョンごとに必要とされる。ソフトウェアアップデートに適用するときに、ソフトウェアアップデートサービス（ｓｏｆｔｗａｒｅ　ｕｐｄａｔｅ　ｓｅｒｖｉｃｅ）は、完全なアップデートされたファイルを伝送する代わりに、より小さなサイズのアップデートデルタパッチを伝送することができる。次いでこのアップデートされたデルタパッチを利用してこの既存のファイルを修正してそのアップデートされたファイルにする。

【発明の開示】

【発明が解決しようとする課題】

【０００９】

このアップデートデルタパッチは、ファイルをアップデートするために必要とされるデータ量を潜在的に減少させる可能性があるが、デルタパッチング（ｄｅｌｔａ　ｐａｔｃｈｉｎｇ）に対する現行のアプローチは、ファイルの多数のバージョンが存在する状況において、適用可能なデルタファイルの選択を管理する際に不十分である。一意のデルタパ

ッチがファイルのバージョンごとに必要とされるので、典型的なソフトウェアアップデートシステムではしばしば、ファイルの一意の各バージョンに対応する、たとえ数千でないとしても数百の一意のデルタパッチを必要とする可能性がある。１つのアプローチでは、デルタパッチングをサポートする一部のアップデートサービスは、すべての可能性のあるデルタパッチをクライアントコンピューティングデバイス（ｃｌｉｅｎｔ　ｃｏｍｐｕｔｉｎｇ　ｄｅｖｉｃｅ）に対して伝送する。しかし、このアプローチでは、一般に可能性のあるアップデートデルタパッチの数が増すにつれて、このソフトウェアアップデートを実施するために必要とされるデータ量が増大する。したがって、適用可能な可能性のあるデルタパッチ数は、その完全なアップデートされたファイルと同じサイズへと速やかに増大してしまう可能性がある。別のアプローチでは、ネットワーク化されたアップデートソフトウェアサービスが、クライアントマシンをスキャンしてクライアントマシンごとにどのデルタパッチが適用可能となり得るかを選択する。これによって伝送されるデルタパッチ情報量が減少するが、このアプローチは、そのクライアントマシンをスキャンし適用可能なデルタパッチを選択するための、このソフトウェアアップデートサービス上の追加のロジックを必要とする。この追加のロジックの使用は、このサービスが提供する必要があるそのシステムリソースを増大させる。さらに、このアプローチは一般に、従来のウェブサーバが一般に実現するなどのネットワークキャッシング（ｎｅｔｗｏｒｋ　ｃａｃｈｉｎｇ）の利用を妨げる。

【００１０】

前述の短所に加えて、既存のシステムでは、ハードウェアドライバなど、ある種のタイプのソフトウェアアップデートを配信することができない。当技術分野において知られているように、ハードウェアドライバに適用されるアップデートなど専用化されたソフトウェアアップデートは、ほとんどの専用化されたソフトウェアアップデートが特定のハードウェアを有するクライアントコンピュータ上でしか機能しないことになるので、大規模配信でユーザに提供することが困難である。ほとんどの場合において、例えばクライアントコンピュータが互換性のないハードウェアドライバのアップグレードを取得する場合には、このドライバアップグレードのインストレーションは、致命的なエラーを引き起こすこともあり、あるいはそのコンピュータが動作しないようにさえすることもある。

【００１１】

前述のことから簡単に理解されるように、サーバといくつかのクライアントの間でソフトウェアアップデートの改善された通信を有するシステムおよび方法が必要になっている。さらに、専用化されたアップデートを配信する場合に、アップデートサービスが特定のタイプのクライアントをターゲットにすることを可能にする改善されたメカニズムを有するソフトウェアアップデートのシステムおよび方法が必要になっている。

【課題を解決するための手段】

【００１２】

本発明は、ソフトウェアアップデートを管理するためのシステムおよび方法を対象としている。より詳細には、本発明は、このソフトウェアアップデートを選択し実施するために必要とされる帯域幅および処理リソースを最小にしながらソフトウェアアップデートの選択および実施を実行するためのシステムおよび方法を対象としている。本発明の一態様によれば、ソフトウェアアップデートを処理するためのシステムおよびコンポーネントアーキテクチャが提供される。

【００１３】

本発明の一態様によれば、ソフトウェアアップデートサービスからクライアントコンピューティングデバイスへとソフトウェアアップデートを伝えるための方法が提供される。この方法は、ソフトウェアアップデートサービスと通信するクライアントコンピューティングデバイスを含むコンピュータシステムにおいて実施することができる。この方法によれば、ソフトウェアアップデートサービスは、クライアントコンピューティングデバイスからの同期化要求を取得する。この同期化要求は、クライアントコンピューティングデバイスが、このインストール済みのソフトウェアアップデートを記憶する場合に、インスト

ール済みのソフトウェアアップデートの識別子を含んでいる。このソフトウェアアップデートサービスは、この同期化要求が、少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含むかどうかを判定する。このソフトウェアアップデートサービスが、この同期化要求が少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含むと判定した場合、このソフトウェアアップデートサービスは、このクライアントコンピューティングデバイスに伝えるための追加のソフトウェアアップデートを選択する。この追加のソフトウェアアップデートの選択は、この追加のソフトウェアアップデート中で定義される前提条件の満足に依存しており、ここで、この前提条件は、同期化要求が少なくとも１つのインストール済みのソフトウェアアップデートについての識別子を含んでいることを必要とする。代わりに、このソフトウェアアップデートサービスが、この同期化要求が少なくとも１つのインストール済みのソフトウェアアップデートの識別子を含んでいないと判定した場合には、このソフトウェアアップデートサービスは、このクライアントコンピューティングデバイスに伝えるための、前提条件を含まない第１のレベルのソフトウェアアップデートを選択する。次いで、このソフトウェアアップデートサービスは、このソフトウェアアップデートサービスからこのクライアントコンピューティングデバイスへとこの選択されたソフトウェアアップデートの命令コンポーネントを伝える。このクライアントコンピューティングデバイスが、この選択されたソフトウェアアップデートに記憶される適用規則の条件を満たす少なくとも１つのコンポーネントを含む場合には、次いでこのソフトウェアアップデートサービスは、このクライアントコンピューティングデバイス中のこの選択されたソフトウェアアップデートの命令コンポーネントをインストール済みのソフトウェアアップデートとして記憶する。

【００１４】

本発明の別の態様によれば、ソフトウェアアップデートサービスからクライアントコンピューティングデバイスへとソフトウェアアップデートを伝えるための方法が提供される。この方法は、ソフトウェアアップデートサービスと通信するクライアントコンピューティングデバイスを含むコンピュータシステムにおいて実施することができる。この方法によれば、ソフトウェアアップデートサービスは、このクライアントコンピューティングデバイスからの許可要求を取得する。この許可要求は、このクライアントコンピューティングデバイス上に記憶されるクライアント許可モジュールの識別子を含んでいる。このソフトウェアアップデートサービスは、このクライアントコンピューティングデバイスが、ターゲットグループに関連するかどうかを判定する。このクライアント許可モジュールが、このクライアントコンピューティングデバイスがそのターゲットグループに関連することを示す場合、このソフトウェアアップデートサービスは、このクライアントコンピューティングデバイスがそのターゲットグループに関連すると判定する。このクライアントコンピューティングデバイスがターゲットグループに関連すると判定された場合には、このソフトウェアアップデートサービスは、このソフトウェアアップデートサービスからこのクライアントコンピューティングデバイスへとそのターゲットグループを識別するサーバクッキーを伝える。このソフトウェアアップデートサービスは、このソフトウェアアップデートサービス上に記憶される少なくとも１つのソフトウェアアップデートを求める要求を取得し、この要求はこのサーバクッキーを含んでいる。この要求を取得することに応答して、このソフトウェアアップデートサービスは、ソフトウェアアップデートがこのサーバクッキー中で識別されるターゲットグループに関連するかどうかを判定する。さらに、このソフトウェアアップデートサービスが、このソフトウェアアップデートがこのサーバクッキー中で識別されるターゲットグループに関連すると判定した場合には、このソフトウェアアップデートサービスは、このソフトウェアアップデートサービスからこのクライアントコンピューティングデバイスへとこのソフトウェアアップデートを伝える。

【００１５】

本発明のさらなる態様によれば、サーバからクライアントコンピューティングデバイスへとデータ構造を伝えるための方法が提供される。この方法は、サーバと通信するクライアントコンピューティングデバイスを含むコンピュータシステムにおいて実施することが

できる。この方法によれば、このクライアントコンピューティングデバイスが、このクライアントコンピューティングデバイスがターゲットグループに関連することを示す許可モジュールを含む場合に、ソフトウェアアップデートサービスは、このクライアントコンピューティングデバイスが、そのターゲットグループに関連するかどうかを判定する。このクライアントコンピューティングデバイスがそのターゲットグループに関連すると判定された場合、このソフトウェアアップデートサービスは、このサーバからこのクライアントコンピューティングデバイスへとサーバクッキーを伝える。このサーバクッキーは、そのターゲットグループを識別する。このソフトウェアアップデートサービスは、このサーバ上で記憶される少なくとも１つのデータ構造を求める要求を取得する。この要求は、このサーバクッキーを含んでいる。この要求を取得することに応答して、このソフトウェアアップデートサービスは、データ構造がこのサーバクッキー中で識別されるそのターゲットグループに関連するかどうかを判定する。このソフトウェアアップデートサービスが、このデータ構造がこのサーバクッキー中で識別されるそのターゲットグループに関連すると判定した場合には、このソフトウェアアップデートサービスは、このサーバからこのクライアントコンピューティングデバイスへとこのデータ構造を伝える。

【００１６】

本発明の前述の態様およびその多くの付随する利点については、以下の詳細なる説明を添付図面と併せて参照することによってさらに良く理解が進むにつれて、さらに簡単に認識されよう。

【発明を実施するための最良の形態】

【００１７】

一般的に説明すると、本発明は、ソフトウェアアップデートを管理するためのシステムおよび方法を対象としている。より詳細には、本発明は、ソフトウェアアップデートを選択し実施するために必要とされる帯域幅および処理リソースを最小にしながら、ソフトウェアアップデートの選択と実施を実行するためのシステムおよび方法を対象としている。本発明によれば、ソフトウェアアップデートは、特定のソフトウェアアプリケーションまたはオペレーティングシステムについてのアップデートに対応することが可能である。さらに、ソフトウェアアップデートは、ソフトウェアドライバ、またはシステムＢＩＯＳなどのファームウェアへのアップデートを含むこともできる。本発明の一態様によれば、これらのソフトウェアアップデートを処理するためのシステムおよびコンポーネントアーキテクチャが提供される。本発明の別の態様によれば、クライアントマシンのアップデートサービスとの許可および同期化を実施するためのアップデートプロトコルおよびインターフェースが提供される。本発明のさらなる態様によれば、インストレーションコンポーネント、およびデルタパッチを利用した様々なインストール済みのファイルをアップデートするための方法が提供される。しかし、本発明の追加の態様を本出願において提供することもできることが当業者には理解されよう。さらに、各識別された態様は、個別に、あるいは共通の発明態様の一部分として考えることができることが、当業者には理解されよう。

【００１８】

図１のソフトウェアアップデートシステム１００は、本発明によるソフトウェアアップデートシステム１００を示すブロック図である。一般的に説明すると、このソフトウェアアップデートシステム１００は、１つまたは複数のクライアントコンピューティングデバイス１１０、アップデートサービス１２０、および外部アップデートプロバイダ１３０を含むことができる。一般的に説明すると、このアップデートサービス１２０は、このクライアントコンピューティングデバイス１１０に伝えられそこでインストールされるソフトウェアアップデートの配信を記憶し管理する。このソフトウェアアップデートは、このアップデートサービス１２０によって、あるいは任意の数の外部アップデートプロバイダ１３０によって提供することができる。

【００１９】

このクライアントコンピューティングデバイス１１０と、このアップデートサービス１

２０と、この外部アップデートプロバイダ１３０は、ネットワーク１０１を介して電子的に通信する。このネットワークは、ＬＡＮ（ローカルエリアネットワーク）、またはＷＡＮ（ワイドエリアネットワーク）やインターネットなどのより大規模なネットワークでもよい。一般的に知られているソフトウェアを使用することにより、ソフトウェアアップデートシステム１００は、クライアントコンピューティングデバイス１１０とこのアップデートサービス１２０のサーバ１２１、１２２、１２３および１２４との間でドキュメント、コマンド、および他の知られているタイプの情報を交換するように構成することができる。当業者には理解されるように、図１に示すソフトウェアアップデートシステム１００は、本発明を実施するための１つの適切なシステムの簡略化された実施例であり、本発明は、この実施例だけには限定されない。

【００２０】

以下にさらに詳細に説明するように、一実施形態のアップデートサービス１２０は、いくつかのサーバを備えている。図１に示すように、このアップデートサービス１２０は、このアップデートサービス１２０のこれら全部のプロセスを管理し、このアップデートサービス１２０のサーバ１２１、１２２、１２３および１２４のプロセスを調整するためのアップデートサーバ１２１を含んでいる。許可サーバ１２２は、クライアントが要求するときに許可クッキーを生成し、次にこの許可クッキーを使用してサーバクッキーを生成し、このサーバクッキーは、クライアントコンピュータがこのアップデートサービス１２０によって生成されるアップデートにアクセスすることができるようにする。メタデータサーバ１２３は、アップデートサービス１２０が提供するこれらのアップデートに関する一般的な情報を提供する。このメタデータサーバ１２３は、本発明のシステムが特定のタイプのクライアントコンピュータまたは特定のグループのクライアントコンピュータについての特定のアップデートを識別することができるようにする。ダウンロードサーバ１２４は、アップデートサービス１２０が提供するソフトウェアアップデートに関連するデータファイルを配信するための１つまたは複数のソフトウェアコンポーネントを提供する。

【００２１】

外部アップデートプロバイダ１３０は、ソフトウェアアップデートを配信する１つまたは複数のサーバを含むことができる。外部アップデートプロバイダ１３０は、ソフトウェア、ソフトウェアアップデート、またはクライアントコンピュータのグループに対して配信すべき他のデータを提供するエンティティに関連づけることができる。例えば、外部アップデートプロバイダ１３０は、このアップデートサービス１２０を使用して１つまたは複数のソフトウェアアプリケーションについてのアップデートを配信したいと希望するサードパーティソフトウェア開発業者に関連づけることができる。別の実施例では、外部アップデートプロバイダ１３０は、ソフトウェアアップデートシステム１２０に関連づけることもできる。

【００２２】

クライアントコンピューティングデバイス１１０は、ソフトウェアアプリケーション１１４を記憶し実行するどのようなコンピューティングデバイスとすることもできる。このクライアントコンピューティングデバイス１１０は、それだけには限定されないが、ＰＣ（パーソナルコンピュータ）、ＰＤＡ（携帯型個人情報端末）、移動電話、２方向ページャなどを含めていくつかの異なるコンピュータ製品のうちのどの１つからでも形成することができる。当業者または他の人々に理解されるように、このクライアントコンピューティングデバイス１１０のアーキテクチャは適切な任意の形態を取ることができる。例えば、クライアントコンピューティングデバイス１１０は、ネットワーク１０１との通信を提供するためのネットワークインターフェースを含むことができる。このネットワークインターフェースは、有線または無線の任意のネットワーク接続と共に使用するように構成することができ、ＴＣＰ／ＩＰプロトコルなど適切な任意の通信プロトコルを用いて使用することができる。さらに、クライアントコンピューティングデバイス１１０は、処理ユニット、ディスプレイ、およびメモリユニットを含むことができる。このメモリユニットは、オペレーティングシステム１１６などのクライアントコンピューティングデバイス１１

０を動作させるために必要なプログラムコードを記憶することができる。さらに、このメモリユニットは、本発明のプロセスを制御し実行するためのアップデート管理コンポーネント１１２を記憶する。

【００２３】

このソフトウェアアップデートシステム１００は、実行される場合に、本発明を実施するソフトウェアプログラムを記憶する。実行される場合に、このソフトウェアアップデートシステム１００は、ソフトウェアアップデートを記憶し管理し選択的に伝える。以下でさらに十分に説明するように、多くの他の利点のうちでもとりわけ、本発明は、ソフトウェアアップデートを受信する資格を有するターゲットグループのクライアントコンピューティングデバイスを定義し選択するメカニズムを提供している。本発明は、ソフトウェアアップデートに関連するデータファイルをダウンロードするための改善されたメカニズムも提供している。

【００２４】

本発明を例示する目的のために、本発明の動作する実施例の詳細な説明が提供される。この動作する実施例を説明するに際して、ソフトウェアアプリケーションの特定のアップグレード、例えばメディアプレーヤバージョン６．０からメディアプレーヤバージョン７．０のアップグレードを言及することができるソフトウェアアップデートに対して言及を行う。当業者には理解されるように、かかるソフトウェアアップデートは、このソフトウェアアップデートに関連するいくつかのデータファイルの通信およびインストレーションを含むことができる。したがって、本発明を例示する目的では、ソフトウェアアップデートとソフトウェアアップデートを含む個別のデータファイルとの間で区別が行われる。

【００２５】

次に図２～６を参照して、クライアントコンピューティングデバイス１１０上で１つまたは複数のファイルをアップデートするためのソフトウェアアップデートシステム１００のコンポーネント間の例示の相互作用について説明する。図２を参照すると、ソフトウェアアップデートサービスは、１つまたは複数の外部アップデートプロバイダ１３０によるソフトウェアアップデート情報の伝送によって起動される。前述のように、外部アップデートプロバイダ１３０は、ソフトウェアアップデートシステム１００に関連づけることができる。代わりに、ソフトウェアアップデート情報は、サードパーティの外部アップデートプロバイダ１３０が伝送することもできる。本発明の例示の一実施形態においては、このソフトウェアアップデート情報は、ファイルをアップデートするために利用されるソフトウェアコード、ファイルを置換するために利用されるソフトウェアコード、このソフトウェアアップデートの適用性を決定するための様々な規則、および／またはこのソフトウェアアップデートを記述するディスプレイ情報を含むことが可能である。このソフトウェアアップデート情報の送信は、いつでも終了することができ、他の例示したソフトウェアアップデートコンポーネントの相互作用の開始と同時である必要はない。

【００２６】

外部アップデートプロバイダ１３０からのソフトウェアアップデート情報を受信すると、このアップデートサービス１２０は、１つまたは複数のデータを生成して、アップデート情報の伝送を実施する。このデータは、ファイルの異なるバージョンをアップデートするための１組のソフトウェアデルタパッチに対応するパッチストレージファイル（ｐａｔｃｈ　ｓｔｏｒａｇｅ　ｆｉｌｅ）を含むことができる。このデータはまた、このパッチストレージファイル中で見出される対応するデルタ（ｄｅｌｔａ）に対して特定のファイルバージョンをマッピングするインデックスに対応するパッチストレージマニフェスト（ｐａｔｃｈ　ｓｔｏｒａｇｅ　ｍａｎｉｆｅｓｔ）を含むこともできる。このデータは、以下にさらに非常に詳細に説明するように、このアップデートエージェント（ｕｐｄａｔｅ　ａｇｅｎｔ）が特定のソフトウェアアップデートデータを要求しインストールするために利用することになる情報に対応する自己展開ファイルをさらに含むことができる。パッチストレージファイル、パッチストレージマニフェスト、および自己展開ファイルの生成は、いつでも終了することができ他の例示したコンポーネントの相互作用と同時である

必要がないことが当業者には理解されよう。

【００２７】

ソフトウェアアップデート情報のクライアントに対する伝送を起動するために、クライアントコンピューティングデバイス１１０は、このアップデートサービス１２０に対して認証要求を起動する。本発明の例示の一実施形態においては、この認証要求は、クライアントコンピューティングデバイス１１０とアップデートサービス１２０の間のアップデートプロトコル相互作用に対応し、これについては、以下でさらに非常に詳細に説明することにする。この認証が完了すると、アップデートサービス１２０は、認証クッキーをクライアントコンピューティングデバイス１１０に送信する。次に図３を参照すると、この認証済みのクライアントコンピューティングデバイス１２０は、次いでこの使用可能なアップデートのアップデートサーバ１２０との同期化を起動する。本発明の例示の一実施形態においては、この同期化要求は、クライアントコンピューティングデバイス１１０とアップデートサービス１２０との間のアップデートプロトコル相互作用にも対応し、これについては、以下でさらに非常に詳細に説明することにする。この同期化が完了すると、クライアントコンピューティングデバイス１１０は、すべての適用可能なソフトウェアアップデートの情報、およびこのアップデートを記述する情報を受信する。しかし、本発明の例示の一実施形態においては、このアップデートをインスタンス化するソフトウェアコードは、ダウンロードされていない。

【００２８】

続けて図３を参照すると、このアップデートプロセス中のある時点で、インストールすべきアップデートの選択が受信される。本発明の例示の一実施形態においては、ユーザが、同期化中に受信するソフトウェアアップデート情報を提示され、適切なアップデートを選択するように求められることもある。代わりに、クライアントコンピューティングデバイス１１０が、すべての適用可能なソフトウェアアップデートを自動的に選択するように構成することもできる。さらに、クライアントコンピューティングデバイス１１０が、使用可能なソフトウェアアップデートのサブセットを自動的に選択することができるようにする何らかの規則を有することもある。さらにまた、ユーザが、インターネットのウェブページを介するなどアップデートサービス１２０と通信することによってアップデートの選択を開始することもできる。

【００２９】

次に図４を参照すると、アップデートエージェントがまだ存在していない場合、アップデート管理コンポーネント１１２は、クライアントコンピューティングデバイス１１０上でアップデートエージェント１１８をインスタンス化する。次いでアップデートエージェント１１８は、自己展開ファイルなどのソフトウェアアップデート情報パッケージの伝送を要求する。このアップデートエージェント１１８は、この自己展開ファイルを受信し、以下で説明するようにそのインストーラに対して任意のアップデートを実施する。さらに、アップデートエージェント１１８は、アップデートサービス１２０から任意の欠損した、または破損した情報を要求することができる。

【００３０】

次に図５を参照すると、アップデートエージェント１１８が、そのソフトウェアアップデート情報パッケージを受信した後に、アップデートエージェント１１８は、クライアントコンピューティングデバイス１１０上でインストールされるファイルのインベントリを実施する。このインベントリとそのソフトウェアアップデート情報パッケージの比較に基づいて、アップデートエージェント１１８は、どのデルタパッチまたは他のアップデート情報が、この選択されたアップデートを完了するために必要になるかを決定する。次いでアップデートエージェント１１８は、特定のデルタアップデートを求める要求を伝送する。本発明の一実施形態においては、ソフトウェアアップデートを求める要求は、直接ネットワーク接続を介して伝送される直接要求に対応することもあり、これは手動アップデートと呼ばれることになる。本発明の別の実施形態においては、ソフトウェアアップデートを求める要求は、顕在的なユーザアクションを必要とせずに伝送されるバックグラウンド

要求である場合もある。この実施形態は、自動アップデートと呼ばれることになる。

【００３１】

本発明の例示の一実施形態においては、このソフトウェアアップデートが、デルタパッチに対応する場合、アップデートエージェント１１８は、そのパッチストレージマニフェストによって識別される特定のデルタパッチを識別する要求をアップデートサービス１２０に対して伝送する。代わりに、デルタパッチが使用可能でない場合、またはいくつかのデルタパッチが失敗している場合に、アップデートエージェント１１８は、フォールバックプロシージャ（ｆａｌｌｂａｃｋ　ｐｒｏｃｅｄｕｒｅ）を起動することができる。このフォールバックプロシージャは、そのパッチストレージファイルから全体のアップデートファイルの完全コピーの伝送を求める要求を含むことができる。このフォールバックプロシージャはまた、自己完結型のパッケージから全体がアップデートされたファイルの完全コピーの伝送を求める要求を含むこともできる。

【００３２】

本発明の例示の一実施形態においては、アップデートサービス１２０のダウンロードサーバ１２４は、アップデートエージェント１１８からのこのソフトウェアアップデート要求を直接処理することができる。代わりに、この要求は、アップデートサービス１２０からのこの要求されたアップデートデルタパッチをいずれにしても受信している従来のウェブサーバなど任意の数の追加の外部ダウンロードサーバによって処理することもできる。例えば、ある会社では、内部サーバを利用してクライアントマシンをアップデートすることができる。さらに、この要求は、以前の要求を処理する際にそのアップデートデルタパッチの一部または全部がキャッシュされた外部ダウンロードサーバによって処理することもできる。したがって、この実施形態においては、このダウンロードは、「ＨＴＴＰ」（ｈｙｐｅｒ　ｔｅｘｔ　ｔｒａｎｓｆｅｒ　ｐｒｏｔｏｃｏｌハイパーテキスト転送プロトコル）データ要求をサービスすることが可能ないくつかの追加のダウンロードサーバに分散することができる。

【００３３】

図６を参照すると、このソフトウェアアップデート情報が受信された後に、アップデートエージェント１１８は、このデルタパッチをこのインストール済みのファイルとマージしてアップデートされたファイルを生成する。さらに、アップデートエージェント１１８は、このマージが正常にその適切なファイルをアップデートしたかどうか妥当性検査することができる。前述のように、デルタパッチを妥当性検査することができない場合には、アップデートエージェント１１８は、そのデルタパッチを再度要求することができ、あるいは、何回かの失敗の後に全体がアップデートされたファイルを要求することができる。アップデートエージェント１１８が、妥当性検査されたアップデートファイルを取得した後に、このファイルは、クライアントコンピューティングデバイス１１０上にインストールされる。

【００３４】

図７は、本発明による、クライアントコンピューティングデバイス１１０とソフトウェアアップデートサービス１２０の間の相互作用を示すソフトウェアアップデート処理ルーチン７００の流れ図である。ブロック７０２において、このソフトウェアアップデートサービス１２０は、このクライアントコンピュータ１１０に対してアクセスを許可する。本発明の例示の一実施形態においては、クライアントコンピュータに対するアクセスの許可は、特定のグループのコンピュータに関連するソフトウェアアップデートへのアクセスを可能にするサーバ発行クッキー（ｓｅｒｖｅｒ－ｉｓｓｕｅｄ　ｃｏｏｋｉｅ）の生成を含むことが可能である。この許可プロセスのより詳細な説明は、図８に関して説明することにする。

【００３５】

ブロック７０４において、このクライアントコンピュータ１１０とソフトウェアアップデートサービス１２０は、アップデート情報を同期化させる。本発明の例示の一実施形態においては、ソフトウェアアップデートサービス１２０は、このクライアントコンピュー

ティングデバイス１１０に対して特定のソフトウェアアップデートを記述するメタデータを送信する。このメタデータは、ユーザがインストレーションについての１つまたは複数のアップデートを選択できるようにする使用可能なソフトウェアアップデートを記述する情報を含んでいる。この同期化プロセスのより詳細な説明については、図９および図１０に関して以下で説明することにする。ブロック７０６において、クライアントコンピューティングデバイス１１０は、ダウンロードする適用可能なアップデートの選択を得る。本発明の例示の一実施形態においては、適用可能なアップデートのこの選択は、ユーザの選択を実施するいくつかの固有のユーザインターフェースの利用に対応する可能性がある。ユーザインターフェースの選択については、図１１に関してさらに非常に詳細に説明することにする。

【００３６】

ブロック７０８において、クライアントコンピューティングデバイス１１０は、適用可能なソフトウェアアップデートのこのユーザ選択を処理し、ソフトウェアアップデートサービス１２０とインターフェースして特定のアップデート情報を要求する。本発明の例示の一実施形態においては、クライアントコンピューティングデバイス１１０は、１つまたは複数の適用可能なアップデートデルタパッチを選択し要求する。次いでクライアントコンピューティングデバイス１１０上のアップデートエージェント１１８は、この要求されたデータを処理してこの選択されたソフトウェアアップデートを実施することができる。ブロック７１０において、このルーチン７００は終了する。

【００３７】

図８を参照して、クライアントコンピューティングデバイス１１０に対してアクセスを許可するための、ブロック７０２（図７）に対応するプロトコル図８００について次に説明することにする。本発明の例示の一実施形態においては、ソフトウェアアップデートサービス１２０は、拡張ターゲティングメカニズム（ｅｘｔｅｎｓｉｂｌｅ　ｔａｒｇｅｔｉｎｇ　ｍｅｃｈａｎｉｓｍ）を利用してクライアントコンピューティングデバイス１１０のアップデートおよび他のソフトウェアへのアクセスを制御する。このソフトウェアアップデートサービス１２０は、特定のソフトウェアアップデートを１つまたは複数のターゲットグループのクライアントコンピューティングデバイス１１０に関連づけるメカニズムを組み込んでいる。例えば、ソフトウェアアップデートサービス１２０は、特定のハードウェアデバイスを有する特定のブランドのクライアントコンピューティングデバイス１１０だけに特定のハードウェアドライバアップデートのアクセスを制限することができる。かかる実施例においては、このソフトウェアアップデートサービス１２０は、特定のブランド名および特定のハードウェアデバイスを有するクライアントコンピューティングデバイス１１０のターゲットグループを定義し、この特定のソフトウェアダウンロードの送信をそのターゲットグループだけに制限することができる。

【００３８】

本発明の例示の一実施形態においては、拡張ターゲティングメカニズムは、１つまたは複数のターゲットグループに対するクライアントコンピューティングデバイスの会員資格を定義するソフトウェアコンポーネント（「許可プラグイン」）の使用によって実行される。クライアントコンピューティングデバイス１１０上の許可プラグインの存在は、そのクライアントコンピューティングデバイスがその許可プラグインの特定のターゲットグループに属するかどうかを規定する。例えばあるターゲットグループは、特定のソフトウェアアプリケーションについての有効な「ＰＩＤ」（ｐｒｏｄｕｃｔ　ｉｄｅｎｔｉｆｉｃａｔｉｏｎ製品識別）番号を有するすべてのコンピュータを含むことができる。かかる実施例においては、図８に関して以下でより詳細に説明するように、許可プラグイン８２６をそのクライアント中でインストールしてそのクライアントコンピューティングデバイスのメモリモジュールからＰＩＤを読み取り、この得られたＰＩＤを対応するＰＩＤサーバプラグイン８２９へと渡すことができる。本明細書中でＰＩＤバリデータ（ＰＩＤ　ｖａｌｉｄａｔｏｒ）８２９とも呼ばれるこの対応するＰＩＤプラグインは、１つまたは複数の方法を利用してこの受信されたＰＩＤが有効かどうかを判定する。このクライアントコ

ンピューティングデバイス１１０上に記憶されているＰＩＤが有効であると判定された後に、このサーバは、このクライアントコンピューティングデバイス１１０が有効なＰＩＤを有するターゲットグループのメンバであることを示すサーバクッキーを生成する。別の実施例においては、ターゲットグループは、ベータテストコンピュータ（ｂｅｔａ　ｔｅｓｔ　ｃｏｍｐｕｔｅｒ）として指定されるクライアントコンピューティングデバイスを含むことができる。

【００３９】

本発明の例示の一実施形態においては、ソフトウェアアップデートサービス１２０の許可サーバ１２２は、この許可サーバが認識することになるクライアントコンピューティングデバイスの１組のターゲットグループを定義するいくつかのサーバ許可プラグインを含んでいる。各サーバ許可プラグインは、クライアントコンピューティングデバイス１１０上に記憶された対応するクライアント許可プラグインとデータをやりとりするコンポーネントを含んでいる。同様にして、各クライアントコンピューティングデバイス１１０は、そのクライアントが属するターゲットグループを識別する１つまたは複数のクライアント許可プラグインを含んでいる。本発明の例示の一実施形態においては、このクライアント許可プラグインは、オペレーティングシステムのインストレーションやアップグレードなど、ソフトウェアアプリケーションのインストレーションまたはアップグレード中に各クライアントコンピューティングデバイス中にインストールすることができる。さらに、このサーバ許可プラグインは、ソフトウェアアップデートへのアクセスを制御したいと希望するアドミニストレータが動的にインストールし、または取り除くことができる。クライアントコンピューティングデバイス１１０および許可サーバ１２２上に記憶されたこの許可プラグインは、実際のソフトウェアプラグインとすることも可能であり、あるいはこの許可プラグインは、動的にリンクされたライブラリ中へとハードコード化することも可能である。

【００４０】

図８に示すように、この許可サーバ１２２は、３つの実施例のサーバ許可プラグイン、すなわち（１）すべてのコンピュータを含むターゲットグループ（以下で、「オールコンピュータターゲットグループ（Ａｌｌ　Ｃｏｍｐｕｔｅｒ　ｔａｒｇｅｔ　ｇｒｏｕｐ）」）を定義する第１のサーバ許可プラグイン８２８、（２）有効なＰＩＤを有するコンピュータを含むターゲットグループ（以下で、「ＰＩＤターゲットグループ」）を定義する第２のサーバ許可プラグイン８２９、および（３）ベータテストコンピュータを含むターゲットグループ（以下で、「ベータターゲットグループ」）を定義する第３のサーバ許可プラグイン８３０を含んでいる。また図８に示すように、クライアントコンピューティングデバイス１１０は、２つのクライアント許可プラグイン、すなわち（１）クライアントコンピューティングデバイス１１０が、オールコンピュータターゲットグループのメンバであることを示す第１のクライアント許可プラグイン８２５、および（２）クライアントコンピューティングデバイス１１０が、ＰＩＤターゲットグループのメンバであることを示す第２のクライアント許可プラグイン８２６を含んでいる。この実施例においては、クライアントコンピューティングデバイス１１０は、このクライアントコンピューティングデバイス１１０がそのベータターゲットグループのメンバであることを示す許可プラグインを含んでいない。当業者には理解されるように、各クライアント許可プラグイン８２５および８２６は、クライアントコンピューティングデバイス１１０上で１つまたは複数のファンクションを実行してこの妥当性検査プロセスを支援するように構成することができる。例えば、この第２のクライアント許可プラグイン８２６は、クライアントコンピューティングデバイス１１０のメモリを検査してインストール済みのソフトウェアアプリケーションについてのＰＩＤを検証し、または取得するように構成することができる。

【００４１】

図８に示すように、許可サブルーチン７０２は、クライアントコンピューティングデバイス１１０が構成要求８０３を許可サーバ１２２に伝える場合に開始される。本発明の例示の一実施形態においては、この構成要求８０３は、この許可サーバ１２２上に記憶され

たその許可プラグインを記述する情報を取得するように構成された任意の適切なソフトウェアコンポーネントから形成される。当業者には理解されるように、構成要求８０３は、「ＧｅｔＣｏｎｆｉｇ」と呼ばれる、知られている方法を利用することができる。この構成要求８０３を受信することに応答して、許可サーバ１２２は、構成応答８０４を伝え、この構成応答は、許可サーバ１２２上に記憶されたすべての許可プラグインを識別する情報を含んでいる。一実施形態においては、この構成応答８０４は、許可サーバ１２２上に記憶されるすべての許可プラグインを識別し記述するストリングのアレイを含んでいる。本実施例においては、この構成応答８０４は、第１のサーバ許可プラグイン８２８、第２のサーバ許可プラグイン８２９、および第３のサーバ許可プラグイン８３０を識別する情報を含んでいる。

【００４２】

ブロック８０５において、クライアントコンピューティングデバイス１１０は、この構成応答８０４を受信することに応答して１つまたは複数の許可クッキーを生成する。ブロック８０５のプロセスにおいて、クライアントコンピューティングデバイス１１０は、マッチングしたクライアントとサーバ許可プラグインの各対ごとに許可クッキーを生成する。したがって、本実施例においては、第１のクライアント許可プラグイン８２５と第１のサーバ許可プラグイン８２８が共にオールコンピュータターゲットグループに関連するので、第１のクライアント許可プラグイン８２５は、このオールコンピュータターゲットグループに関連する第１の許可クッキーを生成する。さらに、第２のクライアント許可プラグイン８２６と第２のサーバ許可プラグイン８２９が共にＰＩＤターゲットグループに関連するので、第２のクライアント許可プラグイン８２６は、このＰＩＤターゲットグループに関連する第２の許可クッキーを生成する。クライアントコンピューティングデバイス１１０は、それがベータターゲットグループのメンバであることを示す許可プラグインをもたないので、第３の許可クッキーは生成されない。

【００４３】

当業者には理解されるように、ブロック８０５のプロセスの一実施形態は、当技術分野において「ＧｅｔＡｕｔｈＣｏｏｋｉｅ」と呼ばれる一般に知られているソフトウェアの方法を使用することを含んでいる。この各許可クッキーの生成は、追加の処理を含むことができることも理解されよう。例えば、第２のクライアント許可プラグイン８２６は、このクライアントのシステムレジストリ（ｓｙｓｔｅｍ　ｒｅｇｉｓｔｒｙ）に記憶された情報を検査してＰＩＤを検索しこのＰＩＤをこの許可クッキー中に含めるように構成することができる。他の実施例においては、ブロック８０５のプロセスは、他のコンピュータまたはデバイスと通信するためのプロセスを含むことができる。例えば、クライアント許可プラグインは、サウンドカード、スキャナ、ビデオカードなどのデバイスと通信してこのデバイスの型およびモデルを取得することができる。限定のない他の実施例においては、クライアント許可プラグインは、フィンガプリントリーダ（ｆｉｎｇｅｒｐｒｉｎｔ　ｒｅａｄｅｒ）などのセキュリティデバイスと通信して、ユーザを記述する情報を取得することができる。

【００４４】

一般に、クライアント許可プラグインは、クライアントコンピューティングデバイス１１０、またはクライアントコンピューティングデバイス１１０に通信可能に接続された他の任意のコンピューティングデバイスの任意のコンポーネントから構成情報を読み取ることができる。他の実施例においては、クライアント許可プラグインは、１つまたは複数のパブリックまたは専用のＡＰＩ（アプリケーションプログラミングインターフェース）を利用してこの対応するサーバプラグインによって妥当性検査されるクライアントからの情報を収集し暗号化するように構成することができる。かかる実施例においては、ＰＩＤバリデータプラグイン８２６は、専用のＡＰＩを使用してこのクライアントのＰＩＤを暗号化して、この暗号化済みのＰＩＤを暗号解読と妥当性検査のためにサーバに渡す。他の実施形態においては、他のクライアント許可プラグインは、フィンガプリントリーダや声紋などの生物測定学の測定を利用して許可クッキーを構築して妥当性検査のためにサーバに

渡すことができる。さらに他の実施例においては、クライアント許可プラグインは、ウェブサービスまたは他の任意のサービスを呼び出して許可証明書または他の任意タイプのデータをこの許可サーバ１２２に伝えることができる。

【００４５】

本発明の例示の一実施形態においては、各許可クッキーは、関連するターゲットグループを識別するストリングを含んでいる。例えば、ストリングは、特定の許可クッキーがこのＰＩＤターゲットグループに関連することを示すことができる。各許可クッキーは、クライアントとサーバの間でデータを伝えるためのデータセクションも含んでいる。例えば、このＰＩＤターゲットグループに関連する許可クッキーは、実際のＰＩＤを含むデータセクションを有することができる。当業者には理解されるように、このデータセクションは、バイトアレイなどの任意の形式で記憶される任意のタイプのデータを含むことができる。例えば、クライアントおよびサーバ上のプラグインが、パブリックキー（ｐｕｂｌｉｃ　ｋｅｙ）およびプライベートキー（ｐｒｉｖａｔｅ　ｋｅｙ）の通信を必要とする場合には、かかるデータは、１つまたは複数の許可クッキーのデータセクション中で暗号化することができる。

【００４６】

クライアントコンピューティングデバイス１１０が対応するクライアントおよびサーバの許可プラグインの各対ごとに許可クッキーを生成した後に、このクライアントコンピューティングデバイスは、この生成済みの許可クッキーをこの許可サーバ１２２に対して伝える。図８に示すように、クライアントコンピューティングデバイス１１０は、クッキー要求８０６においてこの許可クッキーを伝える。このクッキー要求８０６は、ブロック８０５のプロセス中で生成された許可クッキーのアレイを伝えるための適切な任意のフォーマットを含んでいる。この許可方法７０２のこの一部分の一実施形態は、当技術分野において「ＧｅｔＣｏｏｋｉｅ」と呼ばれる一般的に知られているソフトウェアの方法の使用を含むことができる。

【００４７】

一実施形態においては、クッキー要求８０６は、クライアントコンピューティングデバイス１１０のメモリに記憶された他の許可サーバクッキーも含んでいる。以下の説明からさらに簡単に理解されるように、クライアントコンピューティングデバイス１１０のこのメモリは、この許可ルーチン７００の以前の実行において作成された以前の許可サーバクッキーを記憶していることもある。クッキー要求８０６においてこの記憶済みの許可サーバクッキーを提供することにより、このクライアントコンピューティングデバイス１１０は、許可サブルーチン７０２の以前の実行で認可されたそのアクセス権を維持することができるようになる。本実施例においては、このクライアント中に許可サーバクッキーが記憶されていないので、このクッキー要求８０６は、このオールコンピュータターゲットグループに関連する第１の許可クッキーと、このＰＩＤターゲットグループに関連する第２の許可クッキーを含んでいる。

【００４８】

次に、ブロック８０７に示すように、このクッキー要求８０６を受信することに応答して、許可サーバ１２２は、サーバクッキーを生成する。一実施形態においては、この受信した許可クッキーごとに適切な許可プラグインに対して呼出しを行って、サーバクッキーデータが生成される。各サーバ許可プラグインによって生成されるサーバクッキーデータは、この受信された許可クッキー中で識別されるターゲットグループごとの識別子を含んでいる。本実施例においては、クッキー要求８０６が、このオールコンピュータターゲットグループに関連する第１の許可クッキーと、このＰＩＤターゲットグループに関連する第２の許可クッキーとを含んでいるので、許可サーバ１２２は、これらの各ターゲットグループについての識別子を含むサーバクッキーデータを生成する。許可サーバ１２２上において、以前のサーバクッキーがそのクッキー要求８０６において受信されている場合には、このサーバクッキーデータを以前のサーバクッキーのデータと組み合わせて新しいサーバクッキーを生成する。一実施形態においては、この新しいサーバクッキーは、トリプ

ルＤＥＳ（Ｔｒｉｐｌｅ　ＤＥＳ）などのパブリックに利用可能な暗号方法の使用によって暗号化される。

【００４９】

本発明の例示の一実施形態においては、このサーバクッキーは、１つまたは複数の関連するターゲットグループを識別する暗号化された情報を含むことが可能である。さらに、このサーバクッキーは、有効期限データ（ｅｘｐｉｒａｔｉｏｎ　ｄａｔａ）を含むことができ、これは、クリアテキストフォーマットでも暗号化フォーマットでも記憶される。クリアテキストフォーマットで記憶される有効期限データは、このサーバクッキーの有効期限切れについて監視するためにクライアントコンピューティングデバイス１１０によって使用される。この暗号化フォーマットで記憶される有効期限データは、ソフトウェアアップデートサービス１２０によって使用されて、このクライアントコンピューティングデバイス１１０が、特定のターゲットグループに関連するアップデートを受信することが許可されているかどうかを判定する。一実施形態においては、このサーバクッキーの有効期限データは、このサーバクッキーで識別されるすべてのターゲットグループに適用される。代わりに、またはこの全体のサーバクッキーに適用される有効期限の時間に追加して、このサーバクッキーは、それぞれ、個別のターゲットグループに適用できる複数の有効期限データを含むことができる。当業者には理解されるように、各サーバクッキーは、追加のデータを含むことができる。例えば、サーバクッキーは、この許可サブルーチン７０２の最後の実行のタイムスタンプなどのクライアント状態情報を記憶するように構成することができる。

【００５０】

生成された後に、この許可サーバクッキー８０９は、許可サーバ１２２からクライアントコンピューティングデバイス１１０に伝えられる。次に、ブロック８１１に示すように、このサーバクッキーは、その後にクライアントコンピューティングデバイス１１０のメモリに記憶される。クライアントコンピューティングデバイス１１０が、このサーバクッキーの少なくとも１つのコンポーネントが、有効期限切れになったことを判定した場合には、このクライアントコンピューティングデバイスは、この許可方法７０２を再実行して新しいサーバクッキーを取得することができる。前述のように、この許可方法７０２の後続の各実行において、クライアントコンピューティングデバイス１１０は、このクッキー要求８０６においてその記憶されたサーバクッキーを許可サーバ１２２に対して伝えることができる。一実施形態においては、サーバが、クライアントにこのサーバ構成が変更されたこと、すなわち新しい許可プラグインが追加されたことを通知しない限り、このクライアントは、要求８０３を送信する必要がない。

【００５１】

本発明の別の態様によれば、このソフトウェアアップデートサービス１２０は、メタデータサーバ１２３とクライアントコンピューティングデバイス１１０の間でアップデート情報を同期化させるための同期化サブルーチンを提供することができる。一意のソフトウェアアップデート階層を使用することによって、この同期化サブルーチンは、特定のクライアントコンピューティングデバイスに適用される特定のアップデートを効率的に識別することが可能である。さらに、この許可サブルーチン７０２において生成されるこのサーバクッキーの使用によって、この同期化サブルーチンは、特定のターゲットグループに関連するアップデートに対するアクセスを選択的に認可することが可能である。

【００５２】

本発明の例示の一実施形態によれば、各ソフトウェアアップデートは３つのコンポーネント、すなわち（１）命令コンポーネント、（２）局所的データコンポーネント、および（３）データコンポーネントを含んでいる。当業者には理解されるように、各アップデートは、１つまたは複数の前述のコンポーネントを有することができる。例えば、アップデートが、命令コンポーネント、局所的データコンポーネント、およびデータストリームコンポーネントを含むこともある。別の実施例においては、アップデートは、クライアントコンピューティングデバイスの１つまたは複数の条件をテストするための１つの命令コン

ポーネントしか含まないこともある。このソフトウェアアップデートの様々なコンポーネントについて以下でより詳細に説明する。

【００５３】

一般的に説明すると、この命令コンポーネントは、２つのサブコンポーネント、すなわち（１）クライアントコンピューティングデバイス１１０がテストすべき１つまたは複数の条件を定義する適用規則、および（２）個別のアップデートの適切なインストレーションのために必要とされる１つまたは複数のアップデートを識別する１組の前提条件を含んでいる。以下で説明するように、この適用規則は、１台のコンピュータに関連したいくつかの条件を定義することが可能であり、これらの各条件は、任意の論理演算子を使用することによって他の条件に関連づけることが可能である。例えば、この命令コンポーネントは、コンピュータが特定のバージョンのＷｉｎｄｏｗｓ（登録商標）をインストールしているかどうかを判定する適用規則を含むこともできる。また以下で説明するように、この１組の前提条件は、以前にインストールされている必要がある１つまたは複数のアップデートを識別することが可能である。例えば、図９を参照して以下でさらに詳細に説明するように、個別のアップデートは、個別のアップデートの適切なインストレーションのために必要とされる他のアップデートをリストにした前提条件を含むこともある。他の実施例においては、図９にも示すように、この１組の前提条件は、より複雑な前提条件の規則を定義するための論理演算子の使用を含むことも可能である。

【００５４】

この命令コンポーネントはまた、特定のアップデートに依存する他のアップデートが存在するかどうかを示す、ブールフラグ（Ｂｏｏｌｅａｎ　ｆｌａｇ）などのコードも含む。図では、特定のアップデートに依存する他のアップデートが存在しない場合には、アップデートは、ＬＥＡＦアップデートであると考えられる。あるアップデートがＬＥＡＦであるかどうかを示すために使用されるこのブールフラグは、関連したアップデートが追加され、または除去されるときに、メタデータサーバ１２３によって動的にアップデートされる。

【００５５】

各アップデートの局所的データコンポーネントは、このアップデートを記述する一般的な情報を含んでいる。例えば、この局所的データコンポーネントは、このアップデートの特徴および利点を記述する情報を含むことができる。この局所的データコンポーネントはまた、このアップデートのインストレーションプロシージャのテキスト記述を含むこともできる。さらに、この局所的データコンポーネントは、このアップデートに関連した他の任意のデータまたは情報を含むこともできる。例えば、その局所的データは、アップデートが、高優先順位のアップデートであることを示すことができる。別の実施例においては、この局所的データは、アップデートが他のソフトウェアアップデートと共にインストールすることができないことを示すメッセージなど、特定のインストレーションメッセージを提供することができる。この局所的情報は、ユーザに対してその含まれる情報の表示を可能にするフォーマットにすることもできる。

【００５６】

各アップデートのデータコンポーネントは、そのアップデートの１つまたは複数のバイナリデータストリーム（ｂｉｎａｒｙ　ｄａｔａ　ｓｔｒｅａｍ）を含んでいる。一実施形態において、各アップデートのデータコンポーネントは、実行可能ファイル、ドキュメント、リンクされたライブラリなど、１つまたは複数のデータファイルに関連づけることができる。以下でさらに詳細に説明するように、各アップデートは、データファイルの組合せに関連づけることができ、これらのデータファイルのそれぞれは、クライアントが使用するソフトウェアの実際のアップグレード、インストレーション、または修正を実施する。例えば、アップデートのインストレーションは、選択されたアップデートを完了するために必要とされるすべての情報を含む１つのＣＡＢファイルの使用によって実施することができる。代わりに、このアップデートのインストレーションは、クライアントコンピューティングデバイス上に記憶される１つまたは複数のファイルをアップデートするため

に使用されるいくつかの個別のアップデートの使用によって実施することができる。

【００５７】

本発明の例示の一実施形態においては、ソフトウェアアップデートは、このソフトウェアアップデートの制御された配信を可能にする階層の形で構成することが可能である。一般的に説明すると、アップデートのこの階層は、アップデートの間の関係を定義しており、特にどのアップデートが他のアップデートに依存しているかを示している。図示するために、実施例の１組のアップデートが、図９に提供され示されている。図に示すように、サンプルのアップデート９００の階層は、ベースの組のアップデート９０１、第２の組のアップデート９０２、および第３の組のアップデート９０３を含んでいる。一般的に、ベースの組のアップデート９０１中の各アップデートは、他のアップデートのインストレーションを必要とする前提条件を有していない。しかし、第６のアップデート９２１は、この第１のアップデート９１１、第２のアップデート９１２、および第３のアップデート９１３のインストレーションを必要とする前提条件を含んでいる。第７のアップデート９２２は、第４のアップデート９１４のインストレーションを必要とする前提条件を含んでいる。第８のアップデート９３１は、第６のアップデート９２１、および第５のアップデート９１５のインストレーションを必要とする前提条件を含んでいる。したがって、第８のアップデート９３１はまた、第１のアップデート９１１、第２のアップデート９１２、および第３のアップデート９１３のインストレーションを必要とする。本発明を例示するために、このサンプルの１組のアップデート９００のうちのすべてのアップデートが、すべてオールコンピュータターゲットグループおよびＰＩＤターゲットグループに関連していることが想定されている。

【００５８】

また図９に示すように、また以下でさらに詳細に説明するように、各アップデートは、このアップデートのインストレーションについての条件を指定する適用規則を含んでいる。例えば、第１のアップデート９１１は、オペレーティングシステムの英語バージョンのインストレーションを必要とする。第２のアップデート９１２は、Ｗｉｎｄｏｗｓ（登録商標）ＸＰバージョンＳＰ１のインストレーションを必要とする。別の実施例においては、第６のアップデート９２１は、ＸＰ　ＰＡＴＣＨ１と呼ばれるソフトウェアパッチのインストレーションを必要とする。したがって、この適用規則が満たされていない場合には、クライアントコンピューティングデバイス１１０は、このアップデートをインストールしないはずである。

【００５９】

図９はまた、特定のアップデートに依存する他のアップデートがあるかどうかを示す命令コンポーネントのセクションを示している。このセクションは、ＬＥＡＦと呼ばれ、これは、ソフトウェアアップデートが一連の関連したアップデートの最後のアップデートであることを示すブール値（Ｂｏｏｌｅａｎ　ｖａｌｕｅ）の形式とすることができる。本発明を例示する目的では、他のアップデートが、特定のアップデートを前提条件としてリストアップしない場合には、特定のアップデートは、ＬＥＡＦアップデートになる。図９に示すように、第７のアップデート９２２と、第８のアップデート９３１が、このたった２つのＬＥＡＦアップデートである。

【００６０】

図１０は、本発明に従って形成される同期化サブルーチン７０４（図７）のプロトコル図を示している。一般的に説明すると、この同期化サブルーチン７０４は、クライアントコンピューティングデバイス１１０と、メタデータサーバ１２３などのサーバとの間である種のアップデートの命令コンポーネントを選択的に交信してクライアントコンピューティングデバイス１１０に適用することが可能なアップデートを識別する。図１０に示すように、アップデートを起動するために、クライアントコンピューティングデバイス１１０は、まずインストール済みのアップデートを処理し、このクライアントにとって使用可能な１つまたは複数のアップデートを要求する同期化要求１０５１をメタデータサーバ１２３に対して伝える。この同期化要求１０５１を受信することに応答して、このメタデータ

サーバ１２３は、いくつかのアップデートをこのクライアントコンピューティングデバイス１１０に対して戻す。以下の説明からさらに簡単に理解されるように、クライアントコンピューティングデバイス１１０は、この同期化要求１０５１の通信に先立って局所的に記憶されたデータを処理する。

【００６１】

クライアントコンピューティングデバイス１１０は、各受信されたアップデートの命令コンポーネントを処理してこの適用規則中で定義されるその条件を満たすことができるかどうかを判定する。個別のアップデート中で定義されるこの条件が満たされる場合、同期化サブルーチン１０５０を例示する目的では、個別のアップデートが「インストール済み」であり、このインストール済みのアップデートがこのクライアントのアップデートキャッシュの第１のコンポーネントに保存される。他方、個別のアップデート中で定義されるこの条件が満たされない場合には、個別のアップデートは、「失敗した」と考えられ、この失敗したアップデートが、このクライアントのアップデートキャッシュの第２のコンポーネントに保存される。この同期化サブルーチン１０５０のこの説明において、アップデートがインストールされる場合には、この適用規則のこれらの前提条件およびこの条件が満たされていることを仮定することができる。このサブルーチンを説明するためのアップデートのインストレーションは、このアップデートに関連するこのデータファイルが、クライアントコンピューティングデバイス１１０において実際にインストールされることを必ずしも意味していない。

【００６２】

一実施形態においては、このクライアントのアップデートキャッシュのこれらの２つのコンポーネントを使用してこの受信済みのアップデートを分類する。第１のコンポーネントが、インストール済みの非ＬＥＡＦアップデートを記憶するために使用され、第２のコンポーネントが、クライアントが受信する他のすべてのアップデート、すなわちインストールされなかったアップデートを記憶するために使用される。このアップデートキャッシュの第２のコンポーネントは、すべてのＬＥＡＦアップデートのストレージも含んでいる。以下でさらに詳細に説明するように、このアップデートキャッシュに記憶されるアップデートが、メタデータサーバ１２３に伝えられこれによって処理されてクライアントコンピューティングデバイス１１０上でインストレーションのために使用可能な他の関連したアップデートを識別することができる。

【００６３】

ここで図１０に戻って、アイテム１０５１、１０５５、および１０６０として示されている同期化要求の細部について次に説明することにする。当業者には理解されるように、この同期化要求は、アップデートを要求するいくつかの異なるデバイス、プロセス、アプリケーション、ユーザ起動されたコマンドのうちの１つによって起動することができる。この同期化要求は、アップデートのリストを要求するユーザ、クライアントエージェントが起動する自動アップデート、あるいはメタデータサーバ１２３またはアップデートサービス１２０からの情報を要求する他の任意のソフトウェアコンポーネントによって起動することができる。一実施形態においては、この同期化要求は、許可ルーチン７０２から生成された許可サーバクッキーなどの許可サーバクッキーを含んでいる。このサーバクッキーの使用により、このサーバは、そのクライアントが１つまたは複数のターゲットグループのメンバであるかどうかを判定できるようになる。

【００６４】

各同期化要求は、このクライアントのアップデートキャッシュに記憶されたアップデートごとの識別子を含むこともできる。より詳細には、１つまたは複数のアップデートが、アップデートキャッシュに記憶されている場合、この同期化要求は、インストール済みの非ＬＥＡＦアップデートについての識別子を有する第１のコンポーネントと、ＬＥＡＦアップデート、失敗したアップデート、インストールされなかった他のアップデートなど、他のすべてのアップデートについての識別子を有する第２のコンポーネントとを含んでいる。このアップデート識別子は、それだけには限定されないが、整数アレイを含めて任意

のフォーマットにすることができる。代わりに、このクライアントのアップデートキャッシュにアップデートが記憶されていない場合には、この同期化要求は、アップデート識別子を用いて構成されていない。同期化要求がアップデート識別子を用いて構成されていない場合に、この同期化要求は、このクライアントコンピューティングデバイス１１０がキャッシュされたアップデートをまったくもたないことを示す指示を提供する。

【００６５】

図１０に示すように、第１の同期化要求１０５１が、クライアントコンピューティングデバイス１１０からメタデータサーバ１２３へと伝えられる。本実施例においては、これがこの方法の最初の実行であることを考えれば、このクライアントのアップデートキャッシュは、どのようなアップデートも含んでいないことになる。したがって、この第１の同期化要求１０５１は、キャッシュされたアップデートについての識別子を含んでいない。この同期化要求を受信することに応答して、ブロック１０５２に示すように、メタデータサーバ１２３は、この同期化要求が少なくとも１つのアップデート識別子を含むかどうかを判定する。この同期化要求がアップデート識別子を含んでいないと判定された場合、メタデータサーバ１２３は、クライアントコンピューティングデバイス１１０に伝えるための第１のレベルのアップデートを選択することによって応答する。前述のように、第１のレベルのアップデートは、他のアップデートを識別する前提条件をもたないどのようなアップデートも含むことができる。

【００６６】

代わりに、同期化要求が、少なくとも１つのアップデート識別子を含んでいると判定された場合には、メタデータサーバ１２３は、このサーバの記憶されたアップデートの前提条件を検査してこのクライアントに配信するための追加のアップデートを選択する。一実施形態においては、メタデータサーバ１２３は、条件が満たされた前提条件を有するアップデートを選択する。この前提条件の検査において、このサーバは、このクライアント上にインストールされた非ＬＥＡＦアップデートの識別子を含む同期化要求の第１のコンポーネントのアップデートを使用する。

【００６７】

条件が満たされた前提条件を有するアップデートを選択するのに加えて、このサーバはまた、この同期化要求の第２のコンポーネント中で識別されるアップデートを使用してこの選択されたアップデートにフィルタをかける。より詳細には、この同期化要求の第２のコンポーネント中で識別されるインストールされていないアップデート、ＬＥＡＦアップデート、および失敗したアップデートを使用して１つまたは複数の選択されたアップデートにフィルタをかける。本発明のこの特徴によって、本発明のシステムおよび方法は、メタデータサーバ１２３上に記憶されるアップデートの複数回の伝送を回避することができるようになる。

【００６８】

本実施例に戻ると、この第１の同期化要求１０５１は、アップデート識別子を含んでいないので、このメタデータサーバ１２３は、クライアントコンピューティングデバイス１１０に伝えるためのベースレベルのアップデート９０１を選択する。図９に示すアップデートのサンプルの組を参照すると、このベースレベルのアップデート９０１は、９１１、９１２、９１３、９１４および９１５と呼ばれるアップデートを含んでいる。

【００６９】

ブロック１０５２の処理において、メタデータサーバ１２３はまた、この同期化要求１０５１に含まれる許可サーバクッキーを検査して、クライアントコンピューティングデバイス１１０に関連するターゲットグループを識別する。メタデータサーバ１２３はまた、ブロック１０５２のプロセスで選択されたこのアップデートのターゲットグループも検査する。次いで、ブロック１０５２のこのプロセスは、この受信済みの許可サーバクッキー中で識別されるターゲットグループに関連していないすべての選択されたアップデートをフィルタで取り除く。本実施例においては、選択されたアップデート９１１、９１２、９１３、９１４および９１５のすべてが、ＰＩＤターゲットグループおよびオールコンピュ

ータターゲットグループに関連するので、これらの選択されたアップデートのすべてがクライアントコンピューティングデバイス１１０に対して送信される。

【００７０】

次いでメタデータサーバ１２３は、同期化応答１０５３中でこの選択されたアップデートをクライアントコンピューティングデバイス１１０に伝える。一般的に、各同期化応答は、このサーバ１２０が送信する各アップデートの命令コンポーネントを含んでいる。したがって、本実施例においては、第１の同期化応答１０５３は、９１１、９１２、９１３、９１４および９１５と呼ばれるこれらのアップデートについての命令コンポーネントを含んでいる。一実施形態においては、各同期化応答は、各アップデートの局所的データコンポーネントまたはデータコンポーネントを含んでいない。

【００７１】

次に、ブロック１０５４に示すように、クライアントコンピューティングデバイス１１０は、各受信済みのアップデートの命令コンポーネントを処理してこの適用規則で定義される条件を満たすことができるかどうかを判定する。再度図９を参照すると、このクライアントコンピューティングデバイス１１０は、受信されたアップデート９１１～９１５の命令コンポーネントを処理する。本発明を例示するために、クライアントコンピューティングデバイス１１０のオペレーティングシステムは、Ｗｉｎｄｏｗｓ（登録商標）バージョンＸＰ　ＳＰ１の英語のインストレーションであることがこの実施例において想定されている。また、クライアントコンピューティングデバイス１１０は、ＤｅｌｌのＰＣであり、３２ビットのＸ８６プロセッサを作動させていることも想定されている。したがって、このサンプルの１組のアップデートの命令コンポーネントの処理においては、このコンピュータが英語のＯＳを含んでいるので、クライアントコンピューティングデバイス１１０は、第１のアップデート９１１中で定義される条件が満たされるはずであると判定するはずである。このオペレーティングシステムがＷｉｎｄｏｗｓ（登録商標）バージョンＸＰ　ＳＰ１であるので、この第２のアップデート９１２中で定義される条件も満たされるはずである。クライアントコンピューティングデバイス１１０が、Ｘ８６プロセッサを作動させているのでこの第３のアップデート９１３中で定義される条件も満たされるはずである。クライアントコンピューティングデバイス１１０がＤｅｌｌのＰＣであるので第５のアップデート９１５中で定義される条件も満たされるはずである。その結果、第１のアップデート９１１、第２のアップデート９１２、第３のアップデート９１３、および第５のアップデート９１５はすべて、このクライアントのアップデートキャッシュの第１のコンポーネントに保存される。クライアントコンピューティングデバイス１１０は６４ビットのＸ８６プロセッサを作動させていないので第４のアップデート９１４中で定義される条件は、満たされないはずである。したがって、この第４のアップデート９１４は、失敗したアップデートであると考えられ、このクライアントのアップデートキャッシュの第２のコンポーネントに保存される。

【００７２】

図１０に戻ると、ブロック１０５４の処理において、クライアントコンピューティングデバイス１１０はまた、後続の同期化要求が必要とされるかどうかを判定する。一実施形態においては、少なくとも１つのこの受信済みのアップデートが、それがＬＥＡＦアップデートではないことを示す場合、後続の同期化要求が必要であることが決定される。本実施例においては、受信済みのアップデートのすべてがＬＥＡＦアップデートではないので、後続の同期化要求が必要とされることが決定される。したがって、クライアントコンピューティングデバイス１１０は、後続の同期化要求１０５５をメタデータサーバ１２３に伝える。

【００７３】

以上で概要を述べたように、同期化要求は、このクライアントのアップデートキャッシュに記憶されるアップデートごとの識別子を含んでいる。したがって、本実施例において、後続の同期化要求１０５５は、第１のアップデート９１１、第２のアップデート９１２、第３のアップデート９１３、および第５のアップデート９１５が、このクライアント上

にインストールされていることを示す第１のデータコンポーネントを含んでいる。さらに、後続の同期化要求１０５５は、第４のアップデート９１４が、このクライアント上に正常にインストールされていないことを示す第２のデータコンポーネントを含んでいる。

【００７４】

後続の同期化要求１０５５を受信することに応答して、以上で概要を述べたように、メタデータサーバ１２３は、後続の同期化要求１０５５が、少なくとも１つのアップデート識別子を含むかどうかを判定する。後続の同期化要求が少なくとも１つのアップデート識別子を含むと判定された場合、メタデータサーバ１２３はすべての記憶されているアップデートの前提条件を検査してこのクライアントに配信するための追加のアップデートを選択する。

【００７５】

再度本実施例を参照すると、ブロック１０５６の処理において、その前提条件が満たされるので、メタデータサーバ１２３は、第６のアップデート９２１を選択するはずである。より詳細には、図９に示すように、第１のアップデート９１１、第２のアップデート９１２、および第３のアップデート９１３のインストレーションを必要とするその前提条件が満たされるので、第６のアップデート９２１が、クライアントコンピューティングデバイス１１０に伝えるために選択される。第７のアップデート９２２および第８のアップデート９３１は、これらの前提条件が満たされないので、このクライアントに伝えるために選択されないはずである。より詳細には、この同期化要求１０５５は、第７のアップデート９２２についての前提条件である、第４のアップデート９１４についての識別子を含んでいない。さらに、同期化要求１０５５は、第８のアップデート９３１についての前提条件である第６のアップデート９２１についての識別子を含んでいない。

【００７６】

図１０に戻ると、同期化サブルーチン１０５０は、後続の応答１０５７において選択されたアップデートをメタデータサーバ１２３からクライアントコンピューティングデバイス１１０へと伝えることによって継続される。再度本実施例を参照すると、この後続の応答１０５７は、第６のアップデート９２１に関連した情報を含むはずであり、これは後続の応答１０５７においてクライアント１１０に伝えられる。

【００７７】

後続の応答１０５７を受信すると、クライアントコンピューティングデバイス１１０は、後続の応答１０５７の命令コンポーネントを処理する。ブロック１０５４のプロセスと同様に、クライアントコンピューティングデバイス１１０は、各受信されたアップデートの命令コンポーネントを処理して、この適用規則中で定義される条件が満たされるかどうかを判定する。本実施例においては、ＸＰ　ＰＡＴＣＨ１がこのクライアントコンピューティングデバイス中でインストールされると想定する場合に、第６のアップデート９２１は、インストールすべきと考えられ、このアップデートは、クライアントコンピューティングデバイス１１０のこのアップデートキャッシュに書き込まれる。この第６のアップデート９２１がＬＥＡＦアップデートではないので、クライアントコンピューティングデバイス１１０は、このクライアントのアップデートキャッシュの第１および第２のコンポーネントに記憶されるすべてのアップデートを含む別の同期化要求１０６０を送信する。この同期化要求１０６０も、許可サーバクッキーを含んでいる。

【００７８】

本実施例においては、メタデータサーバ１２３の前述の処理を使用することによって、この同期化要求１０６０は、ブロック１０６１において処理され、ここでは、このサーバは、第８のアップデート９３１を選択する。この同期化要求１０６０は、第５および第６のアップデート９１５および９２１がクライアントコンピューティングデバイス１１０中にインストールされることを示すので、この第８のアップデート９３１が選択される。この第８のアップデート９３１は、この許可サーバクッキー中で識別される同じターゲットグループに関連することを想定すると、第８のアップデート９３１の命令コンポーネントが、別の応答１０６２中においてクライアントコンピューティングデバイス１１０に対し

て伝えられる。次いで、第８のアップデート９３１が、ブロック１０５４および１０５９のプロセスと同様にしてブロック１０６３において処理される。この応答１０６２の受信済みのアップデートがすべて、ＬＥＡＦアップデートであるので、後続の同期化要求が、メタデータサーバ１２３に対して返信されない。

【００７９】

クライアントコンピューティングデバイス１１０において、この受信済みのアップデートのすべてがＬＥＡＦアップデートであると判定された後に、またはアップデートが応答１０６２中で受信されない場合には、同期化サブルーチン１０５０は、クライアントコンピューティングデバイス１１０からメタデータサーバ１２３へとドライバ同期化要求１０６４を伝える。当業者には理解されるように、このドライバ同期化要求１０６４は、クライアントコンピューティングデバイス１１０中にインストールされるハードウェアをすべて記述する情報と、このインストール済みのソフトウェアを記述する情報を含むことができる。以前のソフトウェア同期化要求（１０５１、１０５５、および１０６０）と同様に、このドライバ同期化要求１０６４は、このインストール済みのアップデートをサーバに対して伝えることができる。さらに、このクライアント上に現在キャッシュされているすべてのドライバアップデートが、もしある場合には、このサーバに対して伝えられる。

【００８０】

このドライバ同期化要求１０６４を受信することに応答して、メタデータサーバ１２３は、このクライアント上にまだキャッシュされていない、クライアントコンピューティングデバイス１１０に適用されるすべてのドライバアップデートを送信することによって応答する。その前提条件および条件が満たされる場合には、ドライバアップデートは、応答１０６５においてクライアントコンピューティングデバイス１１０に対して送信される。このドライバアップデートを伝える応答１０６５は、各アップデートの命令コンポーネントを伝えることが好ましい。次いでこのドライバアップデートは、このクライアントコンピューティングデバイスのアップデートキャッシュに書き込まれる。

【００８１】

このドライバアップデートを含むこの応答１０６５を受信した後に、この同期化サブルーチン１０５０は、この受信済みのソフトウェアアップデートおよびハードウェアアップデートのそれぞれの局所的データを求める要求１０６６を送信する。以上で概要を述べたように、各アップデートの局所的データコンポーネントは、このアップデートを記述する一般的な情報を含んでいる。例えば、この局所的データコンポーネントは、このアップデートの特徴および利点を記述する情報を含むことができる。この局所的データコンポーネントはまた、このアップデートのインストレーションプロシージャのテキスト記述を含むこともできる。さらに、この局所的データコンポーネントは、このアップデートに関連した他の任意のデータまたは情報を含むこともできる。

【００８２】

したがって、この受信済みのソフトウェアアップデートおよびハードウェアアップデートのそれぞれのこの局所的データを求める要求１０６６を受信すると、メタデータサーバ１２３は、このクライアントのこのアップデートキャッシュに保存されたすべてのこの受信済みのソフトウェアアップデートおよびハードウェアアップデートについてのすべての局所的データを送信することによって応答する。受信した後に、この局所的データをソフトウェアアプリケーションによって処理して、これらのアップデートのうちのどれをインストールする必要があるかを決定することができる。代わりに、この受信済みの局所的データをユーザに対して表示して、このクライアントコンピューティングデバイス１１０にとって使用可能なすべてのアップデートについてそのユーザに通知することができる。一実施形態においては、この受信済みの局所的データは、ウェブページ上に表示することが可能である。本実施例においては、局所的データは、第６および第８のアップデート９２１および９３１についてこのクライアントが受信することができる。局所的データがベースアップデート９１１、９１２、９１３、および９１５に記憶されている場合には、これらのアップデートについてのこの局所的データも、このクライアントは受信するはずであ

る。

【００８３】

図１１は、このクライアントにとって使用可能なこれらのアップデートに関連する局所的データの一実施例を表示するウェブページ１１００の一実施例を示している。図では、ウェブページ１１００は、あるアップデートの第１の詳細な記述１１０５と、別のアップデートの第２の詳細な記述１１０６とを含んでいる。また図に示すように、各アップデートは、それぞれこれらのアップデートのユーザ選択を受け取るための選択メカニズム１１０３および１１０４に関連している。また図に示すように、ウェブページ１１００は、ユーザが、メタデータサーバ１２３やダウンロードサーバ１２４などのサーバに対する、アップデートの選択の通信を制御できるようにする制御ボタン１１０１を用いて構成されている。

【００８４】

本発明の一態様において、このクライアントは、いくつかのプロセスを実施してこのウェブページ１１００の表示を拡張する。例えば、クライアントコンピューティングデバイス１１０は、各アップデートのこの局所的データを検査して、特定のアップデートが高優先順位であるかどうかを判定する。かかる特徴は、局所的データ中に、または特定のアップデートの他のコンポーネント中に、この特定のアップデートが高優先順位アップデートまたは緊急アップデートであることを示すテキストを見つけることによって実施することができる。クライアントコンピューティングデバイス１１０が高優先順位アップデートまたは緊急アップデート検出する場合には、このクライアントは、このページの最上部セクションなど、ウェブページ１１００の可視セクションにこの高優先順位アップデートを表示する。さらに、このクライアントは、このアップデートが高優先順位アップデートであることを示す、専用化されたテキストメッセージ１１２０などの視覚インジケータを生成することができる。

【００８５】

クライアントコンピューティングデバイス１１０はまた、各アップデートの局所的データを検査して、特定のアップデートが、排他的インストレーションを必要とするかどうか、すなわち別のアップデートのインストレーションファイルと同時にはインストールすることができないインストレーションファイルを有するアップデートであるかどうかを判定することができる。かかる特徴は、局所的データ中に、または特定のアップデートの他のコンポーネント中に、この特定のアップデートが排他的インストレーションを必要とすることを示すテキストを見つけることによって実施することができる。クライアントコンピューティングデバイス１１０が、かかるアップデートを検出する場合には、このクライアントは、排他的インストレーションを必要とするアップデートの記述を有する、図１１に示すテキストメッセージ１１２２などの視覚インジケータを表示する。

【００８６】

図７に戻ると、このソフトウェアアップデートルーチン７００は、ブロック７０６で継続され、ここで、クライアントコンピューティングデバイス１１０は、これらのアップデートの選択を受け取る。以上で指摘したように、制御ボタン１１０１の動作に応答して、１つまたは複数のアップデートの選択をメタデータサーバ１２３またはダウンロードサーバ１２４が取得することが可能である。１つまたは複数のアップデートの選択を受け取った後に、このソフトウェアアップデートルーチン７００は、ブロック７０８で継続され、ここでこの選択されたソフトウェアアップデートが処理される。

【００８７】

本発明のさらに他の態様によれば、ソフトウェアアップデートサービス１２０は、このソフトウェアアップデートサービスとクライアントコンピューティングデバイス１１０の間で情報を選択し伝送する方法を提供することができる。図１２Ａおよび１２Ｂは、本発明による、要求されたソフトウェアを検索しインストールするための、クライアントコンピューティングデバイス１１０によって実施されるソフトウェアアップデート処理サブルーチン１２００を示している。前述のように、このソフトウェアアップデート処理サブル

ーチン１２００は、ソフトウェアアップデートの選択が生成され受信された後に実施することができる。図１２Ａを参照すると、ブロック１２０２において、アップデート管理コンポーネント１１１が、アップデートエージェント１１８をインスタンス化する。本発明の例示の一実施形態において、このアップデートエージェント１１８は、要求されたソフトウェアアップデートを完了するため、このアップデートエージェントのインストレーションコンポーネントの要求されたバージョンを生成するため、既存のファイルをデルタパッチとマージすることによってアップデートされたファイルを生成するため、および／またはアップデートされたファイルのインストレーションを開始するために、どのようなソフトウェアアップデート情報が必要とされるかを決定するための専用ソフトウェアコンポーネントである。アップデートエージェント１１８が、すでにインスタンス化されている場合には、ブロック１２０２は、省略することができる。

【００８８】

ブロック１２０４において、アップデートエージェント１１８は、アップデートサービス１２０からソフトウェアアップデート情報を取得する。本発明の例示の一実施形態においては、アップデートサービス１２０が送信するこのソフトウェアアップデート情報は、このアップデートエージェントが利用することができる様々なデータを含む、自己展開ファイルなどのパッケージの形態である。一態様によれば、このパッケージは、特定のソフトウェアアップデートに対応するすべてのファイルのリストを含むことが可能である。さらに、このパッケージは、アップデートすべき特定のバージョンのファイルをアップデートサービス１２０上のそのパッチストレージファイルに記憶される対応するソフトウェアアップデートデルタパッチに対してマッピングするパッチストレージマニフェストの少なくとも一部分のコピーを含むことが可能である。このパッケージはまた、このインストレーションを完了するために必要とされるインストレーションコンポーネントのバージョンの識別を含むことが可能な、アップデートすべきファイルごとのインストレーション情報を含むことも可能である。さらにこのパッケージはまた、アップデートエージェント１１８についてのインストレーションコンポーネントまたはクライアントコンピューティングデバイス１１０上にすでに記憶されているインストレーションコンポーネントのバージョンをアップデートするためのデルタパッチを含むことが可能である。さらにまた、このパッケージは、このアップデートエージェントにソフトウェアアップデートが正常であったかどうかを判定できるようにする検証情報を含むことが可能である。例えば、この検証情報は、比較するためのアップデートファイルについての基準ハッシュ値を含むことが可能である。このアップデートエージェント１１８は、パッケージのその内容を検証することもできる。

【００８９】

判断ブロック１２０６において、テストを実施してアップデートエージェント１１８がアップデートを実施するためにこのインストレーションコンポーネントのバージョンをアップデートする必要があるかどうかを判定する。この自己展開ファイル中のインストレーションコンポーネントの完全なコピーの送信により、ソフトウェアアップデートごとの、アップデートサービス１２０が送信するデータ量を増大させる可能性があることが当業者には理解されよう。したがって本発明の例示の一実施形態においては、インストレーションコンポーネントのベースラインバージョンは、このクライアントコンピューティングデバイスに記憶し、インストレーションコンポーネントデルタパッチを介してこの現行のソフトウェアアップデートの要件について特にアップデートすることができる。したがって、この自己展開ファイル中のこのインストレーション情報は、クライアントコンピューティングデバイス１１０上において、含められた任意のインストレーションコンポーネントアップデートをこのインストレーションコンポーネントのこのベースラインバージョンとマージする必要があるか否かをアップデートエージェント１１８に指示する。アップデートが必要とされる場合、図１３に関して以下でさらに非常に詳細に説明するように、ブロック１２０８において、アップデートエージェント１１８は、このベースラインインストレーションコンポーネント（ｂａｓｅｌｉｎｅ　ｉｎｓｔａｌｌａｔｉｏｎ　ｃｏｍｐｏ

ｎｅｎｔ）をアップデートする。

【００９０】

このアップデートエージェントがこのインストレーションコンポーネントをアップデートした後に、またはこのインストレーションコンポーネントが、アップデートを必要としない場合には、ブロック１２１０において、アップデートエージェント１１８は、クライアントコンピューティングデバイス１１０上にインストールされたファイル、およびこの特定のバージョンのファイルのインベントリを実施する。本発明の例示の一実施形態において、アップデートエージェント１１８は、この選択されたアップデートに対応するものとしてこのパッケージ中で識別されたすべてのファイルについてクライアントコンピューティングデバイス１１０のファイルシステムに照会することができる。代わりに、アップデートエージェント１１８が、最近インベントリを実施したことがある場合には、このインベントリのキャッシュされたバージョンを利用することができる。ブロック１２１２において、アップデートエージェント１１８は、この要求されたアップデートを完了するためにどのようなソフトウェアアップデート情報が必要とされるかを識別する。本発明の例示の一実施形態において、そのパッチストレージマニフェストは、必要とされるデルタパッチに対するインストール済みのファイルのバージョンのマッピングを含んでいる。したがって、デルタパッチングが利用可能な場合、アップデートエージェント１１８は、このマッピングを利用してこのパッチストレージファイル内の特定のデルタパッチおよびそのオフセットロケーションを識別することになる。代わりに、デルタパッチが利用可能でなく実施できない場合には、アップデートエージェント１１８は、ダウンロードするために全体のファイルを識別することができる。

【００９１】

次に図１２Ｂを参照すると、ブロック１２１４において、このアップデートエージェント（ｕｐｄａｔｅ　ｄａｔｅ　ａｇｅｎｔ）は、識別されたソフトウェアアップデート情報を求める要求を送信する。本発明の例示の一実施形態において、アップデートエージェント１１８は、このアップデートサービス１２０のダウンロードサーバ１２４に対してこのパッチストレージファイルから必要とされる特定の範囲のパッチを示すことによって特定のデルタパッチを求める要求を送信することができる。前述のように、このパッチストレージファイルは、各デルタパッチがそのパッチストレージファイルに伴うそのロケーションによって識別される多数の適用可能デルタパッチを含んでいる。このパッチストレージファイルが一部の実施形態ではかなり大きいこともあるので、アップデートエージェント１１８は、このパッチストレージマニフェストから指示されるように、このパッチストレージファイル中の特定のロケーションからのデータだけしか求めない要求を利用することができる。本発明の代替実施形態においては、アップデートエージェント１１８は、アップデートファイルの全体コピーおよび／またはこのパッチストレージファイルの完全コピーを要求することができる。

【００９２】

本発明の代替実施形態においては、アップデートサービス１２０に排他的に関連づけることができない別のダウンロードサーバが、このアップデートエージェント１１８の要求を処理することができる。この実施形態においては、この要求パッチストレージファイルは、全部または一部分をネットワーク上で任意の数の追加のダウンロードサーバに対して伝送することができる。この追加のダウンロードサーバは、プライベートネットワーク上でクライアントをアップデートするために利用されるプライベートネットワークの一部分とすることができる。さらに、この追加のダウンロードサーバは、パブリックネットワークの一部分とすることができる。プライベートネットワーク環境においては、これらのダウンロードサーバは、クライアント要求を処理するためのパッチストレージファイルの完全コピーを取得することができる。代わりに、ダウンロードサーバは、他のクライアントからの以前のデータ要求を処理する際にこのパッチストレージファイルの一部分をキャッシュすることもでき、このキャッシュデータを利用してこのダウンロードを履行することができる。したがって、この追加のダウンロードサーバは、アップデートサービス１２０

のダウンロードサーバ１２４上における通信歪み（ｃｏｍｍｕｎｉｃａｔｉｏｎ ｓｔｒａｉｎ）を低減することが可能である。

【００９３】

ブロック１２１６において、アップデートエージェント１１８は、この要求されたアップデート情報を受信する。本発明の例示の一実施形態においては、この要求されたアップデート情報は、２つのアプローチで伝送することができる。手動アップデートと呼ばれる第１のアプローチでは、このアップデート要求は、直接ＨＴＴＰデータ配信応答（ｄｉｒｅｃｔ ＨＴＴＰ ｄａｔａ ｄｅｌｉｖｅｒｙ ｒｅｓｐｏｎｓｅ）を求める要求と共にアップデートサービス１２０に対して送信される。このアプローチにおいては、アップデートサービス１２０は、使用可能な全帯域幅のすべてを利用してこの要求されたデータをアップデートエージェント１１８に対して送信することができる。自動アップデートと呼ばれる第２のアプローチでは、このアップデート要求は、間接ＨＴＴＰデータ配信応答（ｉｎｄｉｒｅｃｔ ＨＴＴＰ ｄａｔａ ｄｅｌｉｖｅｒｙ ｒｅｓｐｏｎｓｅ）を求める要求と共にアップデートサービス１２０に対して送信される。この応答においては、アップデートサービス１２０は、この要求されたデータをバックグラウンドプロセス（ｂａｃｋｇｒｏｕｎｄ ｐｒｏｃｅｓｓ）として送信する。このバックグラウンドプロセスは、使用可能な帯域幅の最小量を利用するようにして実施することができる。さらに、このバックグラウンドプロセスは、このダウンロードプロセス中に中断し、次に利用可能な時に再開することができる。バックグラウンドプロセスを介して要求されたデータを伝送するためのシステムおよび方法についての説明は、参照により本明細書中に組み込まれている、２０００年2月１６日に出願の「System and Method for Transferring Data Over a Network」という名称の同一出願人による同時係属の米国特許出願第０９／５０５，７３５号明細書中に説明されている。フォアグラウンドまたはバックグラウンドのデータ配信は、この選択されたソフトウェアアップデートの優先順位を必ずしも反映したものではなく、代わりにこのアップデート情報を取得するためにどのようにして帯域幅が割り付けられるかを反映したものであることが当業者には理解されよう。

【００９４】

要求された情報をこのアップデートサービスから受信した後に、ブロック１２１８において、アップデートエージェント１１８は、このデルタパッチをこの対応するインストール済みのファイルとマージする。本発明の例示の一実施形態においては、アップデートエージェント１１８は、このインストール済みのファイルの元のバージョンをキャッシュして、この選択されたファイルがこのダウンロードおよびマージングプロセス中に変化しないようにする。さらに、このインストール済みのファイルのキャッシュされた元のバージョンを使用してこの選択されたアップデートをアンインストール（ｕｎｉｎｓｔａｌｌ）することができる。

【００９５】

判断ブロック１２２０において、テストを実施してこのアップデートされたファイルが有効であるかどうかを判定する。本発明の例示の一実施形態においては、アップデートエージェント１１８は、ハッシングアルゴリズムを利用して、アップデート情報パッケージから得られ、有効なファイルアップデートに対応する基準ハッシュ値を現在の修正されたファイルからのハッシュと比較することができる。これらのハッシュがマッチングしない場合、現行の修正ファイルは有効ではない。いくつかの代替の妥当性検査アルゴリズムのうち任意の１つを利用することもできることが当業者には理解されよう。このアップデートファイルが有効でない場合には、このサブルーチン１２００はブロック１２１４に戻り、ここでこのアップデートエージェントは、アップデート情報を再度要求することができる。代わりに、アップデートエージェント１１８が、このアップデートファイルを数回にわたって生成しようと試みてもうまくいかない場合には、このアップデートエージェントは、いくつかのフォールバックプロシージャ（ｆａｌｌｂａｃｋ ｐｒｏｃｅｄｕｒｅ）の１つを実施することができる。本発明の一実施形態において、このアップデートエージェント１１８は、このパッチストレージファイルに記憶されこのパッチストレージマニフ

ェストから識別されたアップデートファイルの完全コピーをアップデートサービス１２０から要求することができる。本発明の他の実施形態においては、アップデートエージェント１１８は、アップデートサービス１２０から自己完結型のファイル中のこのアップデートファイルのコピーを要求することができる。本発明のさらに他の実施形態においてサブルーチン１２００は、その他の場合には失敗することもある。

【００９６】

この選択されたファイルが有効になった後に、判断ブロック１２２２において、テストを実施して追加のダウンロードが何か必要とされるかどうかを判定する。本発明の例示の一実施形態においては、サブルーチン１２００は、以前に選択されたダウンロードの終了後に追加のダウンロードがあるかについて絶えずチェックする反復ループに入る。このダウンロード中にファイルの状態が変化する場合、アップデートエージェント１１８は、この新しい状態変化について追加のダウンロードを要求し続けるはずである。追加のダウンロードが必要とされる場合には、ブロック１２２４において、アップデートエージェント１１８は、別のインベントリを実施し、すべての適用可能なデルタパッチを識別する。次いでこのサブルーチン１２００はブロック１２１４へと戻る。

【００９７】

すべてのこの要求されたアップデートのダウンロードが完了した後に、判断ブロック１２２４において、テストを実施して、このクライアントマシンの状態が変化しているかどうかを判定する。本発明の例示の一実施形態においては、アップデート情報のダウンロードおよびマージングと、このアップデートされたファイルの実際のインストレーションの間で時間が経過してしまうこともある。したがってこのアップデートファイルをインストールするのに先立ってこのアップデートエージェントは、このクライアントコンピューティングデバイス状態が変化しているかどうかを判定する。この状態が変化している場合には、このファイルアップデートは有効でないこともあり、このアップデートはブロック１２２６において失敗する。代わりに、状態の変化が起こっていなければ、アップデートエージェント１１８は、ブロック１２２８においてこのアップデートされたファイルをインストールし、サブルーチン１２００は、ブロック１２３０において戻る。

【００９８】

次に図１３を参照して、ブロック１２０８（図１２Ａ）に対応するベースラインインストレーションコンポーネントをアップデートするための、クライアントコンピューティングデバイス１１０によって実施されるサブルーチン１３００について説明することにする。判断ブロック１３０２において、テストを実施して新しいベースラインインストレーションコンポーネントが、アップデートサービス１２０からアップデートエージェント１１８に伝送される自己展開ファイル中に含まれるかどうかを判定する。本発明の例示の一実施形態において、このベースラインインストーラ（ｂａｓｅｌｉｎｅ　ｉｎｓｔａｌｌｅｒ）をアップデートするために必要とされるデルタパッチが、アップデートされたインストレーションコンポーネントの伝送とサイズが同等である場合、新しいベースラインインストレーションコンポーネントが伝送されることになる。アップデートされたインストレーションコンポーネントが含まれる場合には、ブロック１３０４において、このアップデートエージェントは、このアップデートされたベースラインインストレーションコンポーネントを新しいインストレーションコンポーネントとしてインストールする。さらに、この新しいアップデートされたインストレーションコンポーネントは、クライアントコンピューティングデバイス１１０のメモリに保存されて追加のアップデートについてのベースラインインストーラとしての役割を果たすことができる。ブロック１３０６においてこのサブルーチンは戻る。

【００９９】

アップデートされたベースラインインストレーションコンポーネントが、この自己展開ファイル中に含まれていない場合には、ブロック１３０８においてアップデートエージェント１１８は、この自己展開ファイルからベースラインインストレーションコンポーネントデルタパッチを取得する。本発明の例示の一実施形態において、このベースラインイン

ストレーションコンポーネントデルタパッチは、このベースラインインストレーションコンポーネントとマージされてアップデートされたベースラインインストレーションコンポーネントを生成することができるソフトウェアコードに対応する。したがって、ブロック１３１０において、このアップデートエージェントは、このベースラインインストレーションコンポーネントデルタパッチをこのベースラインインストレーションコンポーネントとマージする。ブロック１３１２において、次いでアップデートエージェント１１８は、このアップデートされたベースラインインストレーションコンポーネントを現在のインストレーションコンポーネントとして指定する。本発明の例示の一実施形態において、このインストレーションが完了した後にはこのアップデートされたインストレーションコンポーネントは、保存されないことになる。この実施形態によれば、アップデートエージェント１１８は、クライアントコンピューティングデバイス１１０のメモリ中に限られた数のベースラインインストレーションコンポーネントしか保持しない。したがって、このアップデートエージェントは、インストレーションごとに、一時的なアップデートされたインストレーションコンポーネントを生成する。各クライアントコンピューティングデバイス１１０は、限られた数のベースラインインストレーションコンポーネントにしか対応することができないので、アップデートサービス１２０には、クライアントコンピューティングデバイスごとに１つのベースラインインストレーションコンポーネントデルタパッチを伝送する必要があるだけである。ブロック１３１４において、このサブルーチン１３００は、戻る。

【０１００】

本発明の好ましい実施形態について示し説明してきたが、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こで行うことができることが理解されよう。例えば、本明細書中で説明した例示の実施例は、ソフトウェアアップデートに適用されるが、本発明の範囲は、ソフトウェアアップデートに関連した情報の配信および通信以外の他の使用を含んでいる。したがって、特定の主題がこの開示中において明確に除外されない限り、本発明の範囲が、ソフトウェアアップデート以外の、またはソフトウェアアップデートに追加の任意タイプのデータの配信および通信に適用されることを理解されたい。

【図面の簡単な説明】

【０１０１】

【図１】本発明による、クライアントコンピュータ、およびアップデートソフトウェアを提供するアップデートサービスを含むソフトウェアアップデートシステムを示すブロック図である。

【図２】本発明による、アップデートサービスを用いたクライアントコンピューティングデバイスの認証を示す、図１のソフトウェアアップデートシステムのブロック図である。

【図３】本発明による、クライアントコンピューティングデバイスとアップデートサービスの間の使用可能なアップデートの同期化を示す、図１のソフトウェアアップデートシステムのブロック図である。

【図４】本発明による、アップデートサービスからクライアントコンピューティングデバイスへのソフトウェアアップデート情報の伝送を示す、図１のソフトウェアアップデートシステムのブロック図である。

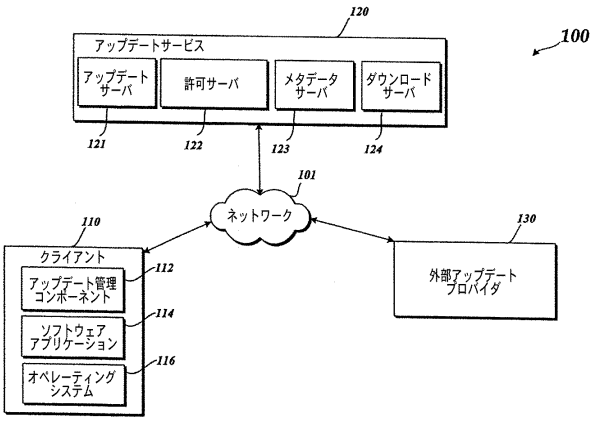
【図５】本発明による、クライアントコンピューティングデバイスによるアップデート情報の処理および選択を示す、図１のソフトウェアアップデートシステムのブロック図である。

【図６】本発明による、クライアントコンピューティングデバイスによるデルタパッチのマージング、およびアップデートファイルのインストレーションを示す、図１のソフトウェアアップデートシステムのブロック図である。

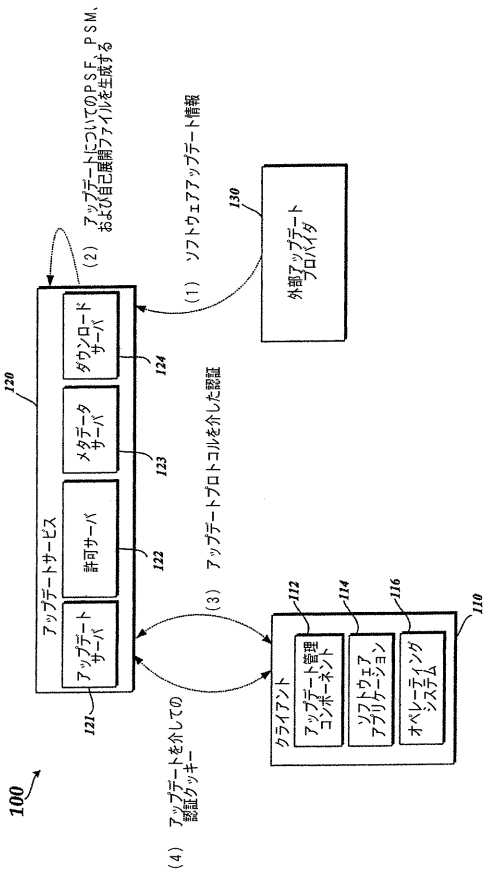
【図７】本発明による、クライアントコンピューティングデバイス上におけるインストレーションのために使用可能なソフトウェアアップデートを識別するための、クライアントコンピューティングデバイスおよびアップデートサービスによって実施されるソフトウェアアップデートルーチンを示す流れ図である。

【図８】本発明による、アップデートサービス上に記憶されるアップデートに対する選択的なアクセスを提供するための許可ルーチンのプロトコル図である。
【図９】本発明による、許可ルーチンを示す実施例の１組のソフトウェアアップデートのブロック図である。
【図１０】本発明による、選択グループのソフトウェアアップデートをソフトウェアアップデートサービスからクライアントコンピューティングデバイスへと伝えるための同期化ルーチンのプロトコル図である。
【図１１】本発明による、個別のクライアントコンピューティングデバイスにとって使用可能なソフトウェアアップデートのリストを表示するためのグラフィックユーザインターフェースの例示のセクションを示すブロック図である。
【図１２Ａ】本発明による、要求されたソフトウェアを検索しインストールするためのクライアントコンピューティングデバイス１１０によって実施されるソフトウェアアップデート処理サブルーチン１２００の一部分を示す図である。
【図１２Ｂ】本発明による、要求されたソフトウェアを検索しインストールするためのクライアントコンピューティングデバイス１１０によって実施されるソフトウェアアップデート処理サブルーチン１２００の一部分を示す図である。
【図１３】本発明による、ベースラインインストレーションコンポーネントをアップデートするための、クライアントコンピューティングデバイスによって実施されるサブルーチンを示す流れ図である。

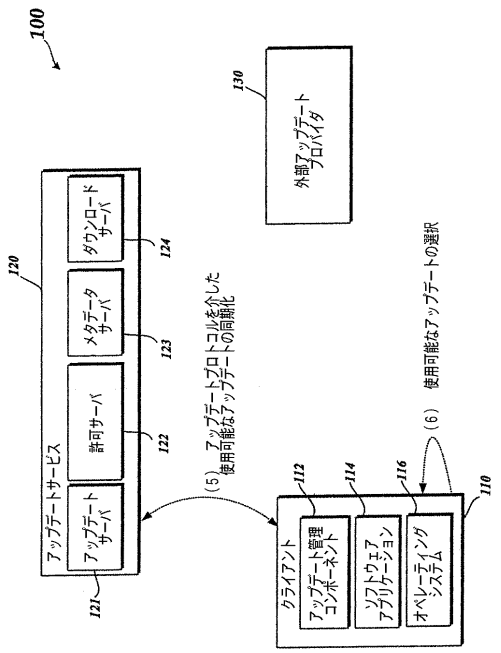
【図１】



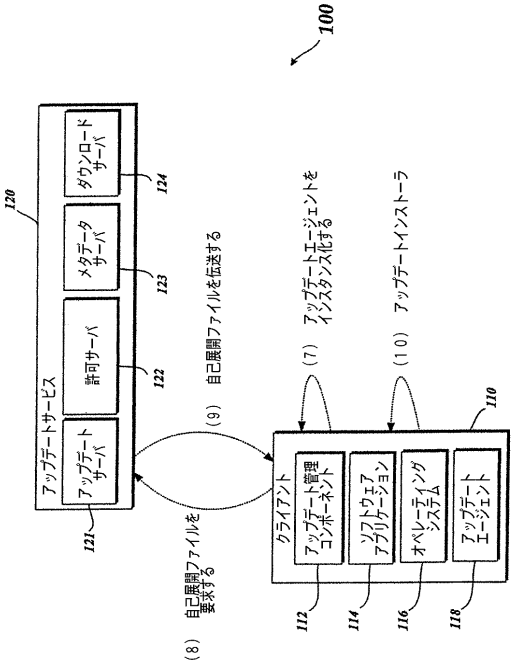
【図２】



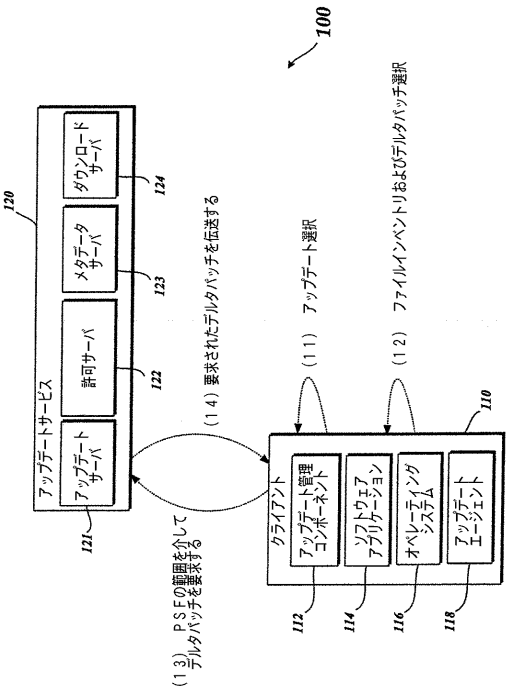
【図３】


100
120
アップデートサービス
アップデートサーバ
許可サーバ
メタデータサーバ
ダウンロードサーバ
121
122
123
124
130
外部アップデートプロバイダ
(5) アップデートプロトコルを介した使用可能なアップデートの同期化
(6) 使用可能なアップデートの選択
クライアント
アップデート管理コンポーネント
ソフトウェアアプリケーション
オペレーティングシステム
アップデートエージェント
110
112
114
116

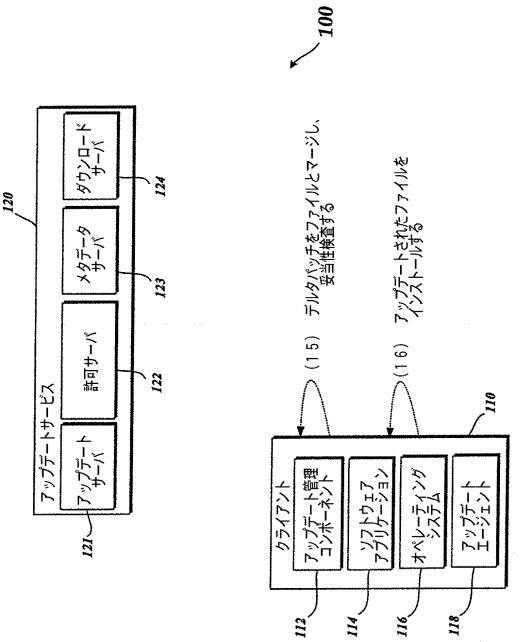
【図４】


100
120
アップデートサービス
アップデートサーバ
許可サーバ
メタデータサーバ
ダウンロードサーバ
121
122
123
124
(8) 自己展開ファイルを要求する
(9) 自己展開ファイルを伝送する
(7) アップデートエージェントをインスタンス化する
(10) アップデートインストーラ
クライアント
アップデート管理コンポーネント
ソフトウェアアプリケーション
オペレーティングシステム
アップデートエージェント
110
112
114
116
118

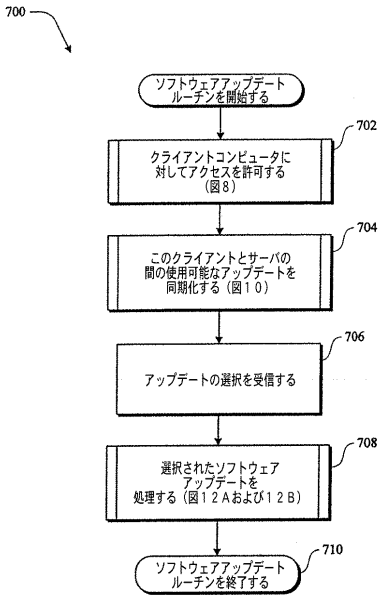
【図５】


100
120
アップデートサービス
アップデートサーバ
許可サーバ
メタデータサーバ
ダウンロードサーバ
121
122
123
124
(13) PSFの範囲を介してデルタパッチを要求する
(14) 要求されたデルタパッチを伝送する
(11) アップデート選択
(12) ファイルインベントリおよびデルタパッチ選択
クライアント
アップデート管理コンポーネント
ソフトウェアアプリケーション
オペレーティングシステム
アップデートエージェント
110
112
114
116
118

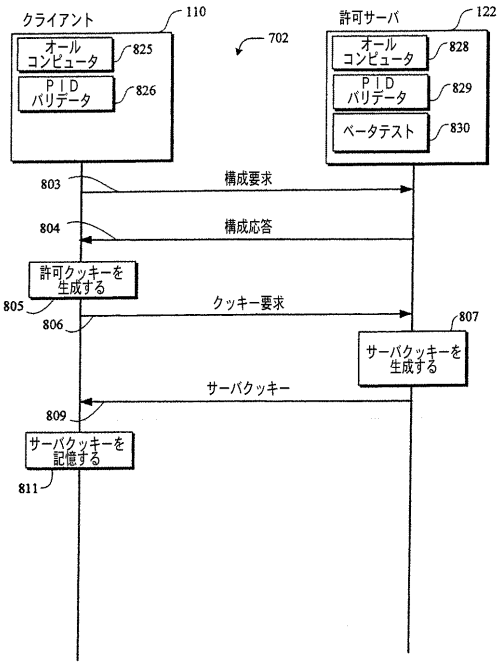
【図６】


100
120
アップデートサービス
アップデートサーバ
許可サーバ
メタデータサーバ
ダウンロードサーバ
121
122
123
124
(15) デルタパッチをファイルとマージし、妥当性検査する
(16) アップデートされたファイルをインストールする
クライアント
アップデート管理コンポーネント
ソフトウェアアプリケーション
オペレーティングシステム
アップデートエージェント
110
112
114
116
118

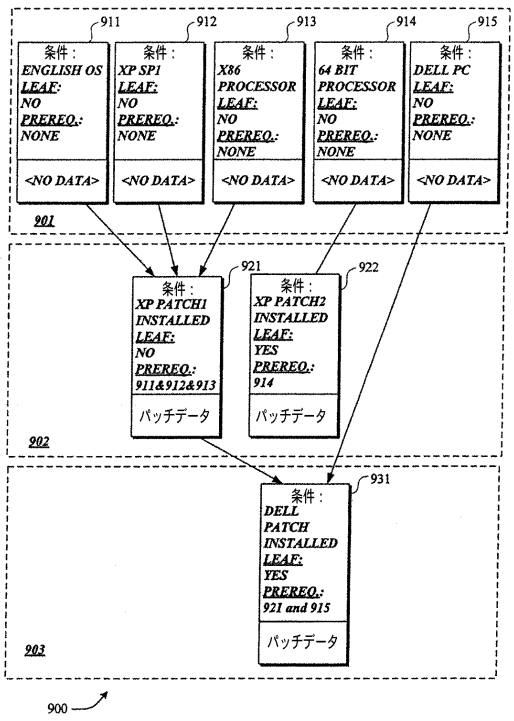
【図７】


700
ソフトウェアアップデート
ルーチンを開始する
クライアントコンピュータに
対してアクセスを許可する
（図８）
702
このクライアントとサーバの
間の使用可能なアップデートを
同期化する（図１０）
704
アップデートの選択を受信する
706
選択されたソフトウェア
アップデートを
処理する（図１２Ａおよび１２Ｂ）
708
ソフトウェアアップデート
ルーチンを終了する
710

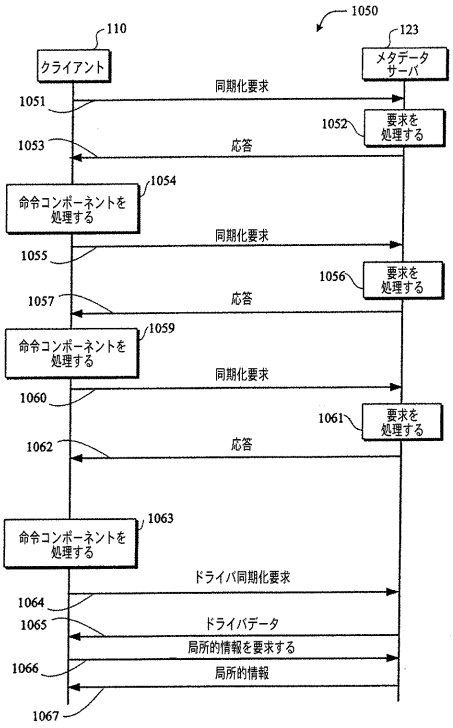
【図８】


クライアント
110
オール
コンピュータ
825
ＰＩＤ
バリデータ
826
702
許可サーバ
122
オール
コンピュータ
828
ＰＩＤ
バリデータ
829
ベータテスト
830
803
構成要求
804
構成応答
許可クッキーを
生成する
805
クッキー要求
806
807
サーバクッキーを
生成する
サーバクッキー
809
サーバクッキーを
記憶する
811

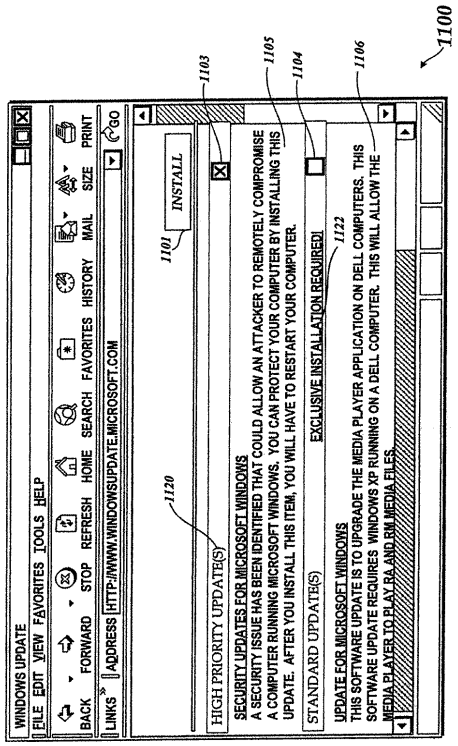
【図９】


911
912
913
914
915
条件：
ENGLISH OS
LEAF:
NO
PREREQ.:
NONE
条件：
XP SP1
LEAF:
NO
PREREQ.:
NONE
条件：
X86
PROCESSOR
LEAF:
NO
PREREQ.:
NONE
条件：
64 BIT
PROCESSOR
LEAF:
NO
PREREQ.:
NONE
条件：
DELL PC
LEAF:
NO
PREREQ.:
NONE
<NO DATA>
901
921
922
条件：
XP PATCH1
INSTALLED
LEAF:
NO
PREREQ.:
911&912&913
条件：
XP PATCH2
INSTALLED
LEAF:
YES
PREREQ.:
914
パッチデータ
902
931
条件：
DELL
PATCH
INSTALLED
LEAF:
YES
PREREQ.:
921 and 915
903
900

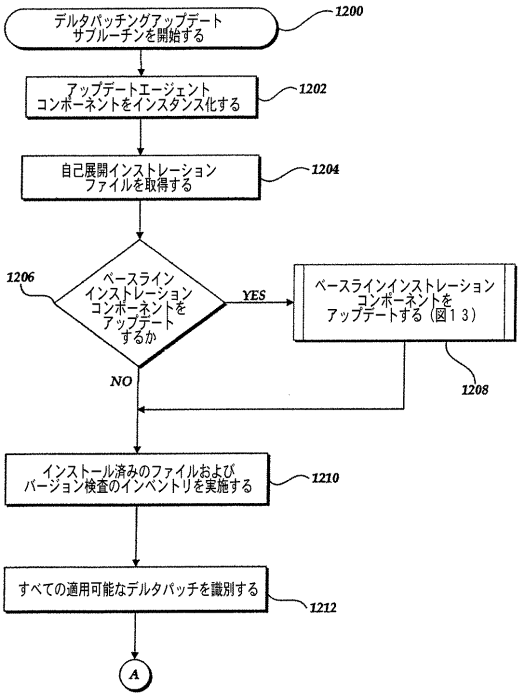
【図１０】


1050
110
クライアント
123
メタデータ
サーバ
1051
同期化要求
1052
要求を
処理する
1053
応答
命令コンポーネントを
処理する
1054
1055
同期化要求
1056
要求を
処理する
1057
応答
命令コンポーネントを
処理する
1059
1060
同期化要求
1061
要求を
処理する
1062
応答
命令コンポーネントを
処理する
1063
ドライバ同期化要求
1064
1065
ドライバデータ
局所的情報を要求する
1066
局所的情報
1067

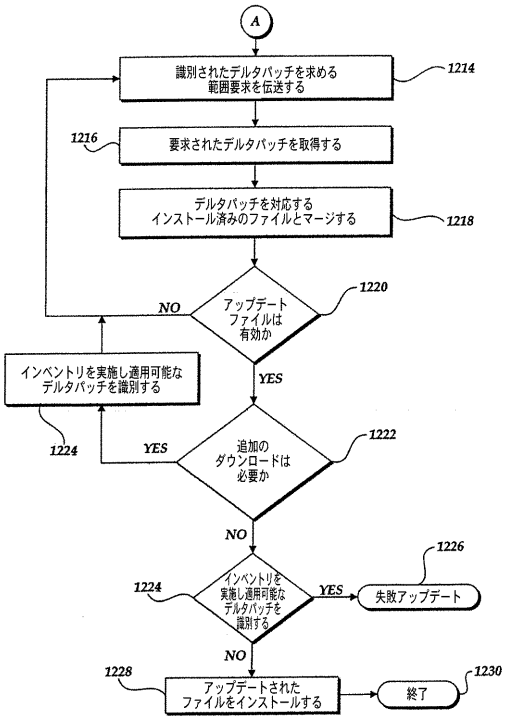
【図１１】



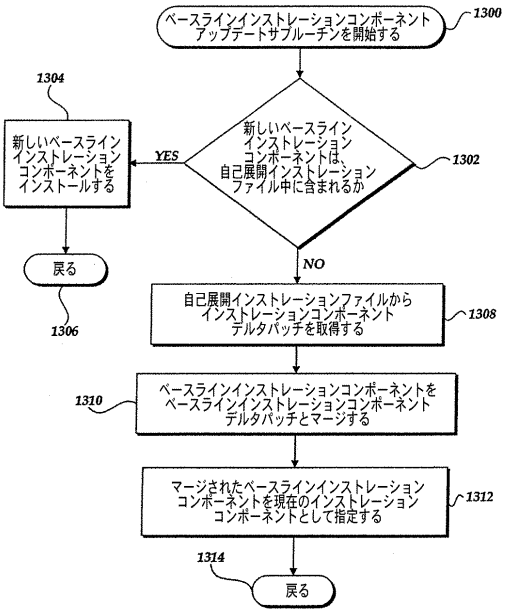
【図１２Ａ】



【図１２Ｂ】



【図１３】



フロントページの続き

(72)発明者 マイケル エドワード メーレマンズ
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 アーロン アベブッシュ
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 ジェーソン 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 ケン ショーマン
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 マザール モハンマド
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 ジョセフ ジー. ダジー
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

審査官 林 毅

(56)参考文献 特開平１１－２７２４５４（ＪＰ，Ａ）
特表２００１－５０８５７５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 11/00
G06F 9/445
G06F 13/00