

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5009366号
(P5009366)

(45) 発行日 平成24年8月22日(2012.8.22)

(24) 登録日 平成24年6月8日(2012.6.8)

(51) Int.Cl.

F I

G O 6 F 12/00 (2006.01)

G O 6 F 12/00 5 3 3 J

G O 6 F 13/00 (2006.01)

G O 6 F 13/00 5 4 0 A

G O 6 F 12/00 5 4 6 M

請求項の数 13 (全 27 頁)

(21) 出願番号 特願2009-512123 (P2009-512123)
 (86) (22) 出願日 平成19年5月22日(2007.5.22)
 (65) 公表番号 特表2009-538475 (P2009-538475A)
 (43) 公表日 平成21年11月5日(2009.11.5)
 (86) 国際出願番号 PCT/US2007/012283
 (87) 国際公開番号 W02007/139824
 (87) 国際公開日 平成19年12月6日(2007.12.6)
 審査請求日 平成22年5月21日(2010.5.21)
 (31) 優先権主張番号 11/438,176
 (32) 優先日 平成18年5月22日(2006.5.22)
 (33) 優先権主張国 米国(US)

(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 ダニエル ビー. ウィトリオール
 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 インターナショナル パテンツ内

最終頁に続く

(54) 【発明の名称】 構造化ウェブサイトコンテンツの同期技術

(57) 【特許請求の範囲】

【請求項1】

クライアントデバイスとサーバデバイスとの間でデータを同期させる方法を実行するコンピュータ実行可能命令を含むコンピュータ読み取り可能な記憶媒体であって、

該コンピュータ実行可能命令は、コンピュータデバイスによって実行される時に、

前記クライアントデバイスによってサポートされる1つまたは複数の構造化コンテンツタイプをクライアントデバイスが識別するステップと、

前記サーバデバイスによってサポートされる1つまたは複数の構造化コンテンツタイプをクライアントデバイスが受信するステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる1つまたは複数の構造化コンテンツタイプをクライアントデバイスが識別するステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる1つまたは複数の構造化コンテンツタイプを同期させる同期要求をクライアントデバイスが送信するステップであって、該同期要求は、変更トークン以降に変更された第1の定義数のデータ項目を要求することを含み、該第1の定義数のデータ項目についての要求は GetListChangesSinceToken クエリ上の rowLimit プロパティを使用して行われる、ステップと、

前記第1の定義数よりも多いデータ項目が変更される時、要求された第1の定義数のデータ項目と、 ListItemCollectionPositionNext の値の開始から第2の定義数のデータ項目を要求するための ListItemCollectio

10

20

n P o s i t i o n N e x t の値とをクライアントデバイスが受信するステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる 1 つまたは複数の構造化コンテンツタイプの少なくとも最初のデータ項目をクライアントデバイスが修正するステップと、

前記少なくとも最初のデータ項目のサーババージョンをサーバデバイスからクライアントデバイスが受信するステップであって、該最初のデータ項目のサーババージョンはバージョン履歴を含む、ステップと、

前記最初のデータ項目のサーババージョンのバージョン履歴をクライアントデバイスが分析するステップであって、該最初のデータ項目のサーババージョンのバージョン履歴は、複数のユーザ識別子および変更番号のペアを含み、ユーザ識別子および変更番号のペアの各々におけるユーザ識別子は最初のデータ項目を編集したコンピュータを識別し、ユーザ識別子および変更番号のペアの各々における変更番号は該最初のデータ項目になされた変更の順番を示す、ステップと、

最初のデータ項目が第 3 のデバイスによって編集されたか否かを、前記最初のデータ項目のサーババージョンのバージョン履歴からクライアントデバイスが判定するステップと、

前記クライアントデバイスにて行われた修正が前記最初のデータ項目のサーババージョンに含まれることを、前記最初のデータ項目のサーババージョンのバージョン履歴からクライアントデバイスが判定するステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる前記 1 つまたは複数の構造化コンテンツタイプをクライアントデバイスが同期させるステップであって、該同期は少なくとも部分的に前記バージョン履歴に基づく、ステップと

を含む方法をコンピュータデバイスに実行させることを特徴とするコンピュータ読み取り可能な記憶媒体。

【請求項 2】

異なる構造化コンテンツタイプについての同期要求を送信する異なるハイパーリンクを前記サーバデバイスが生成するステップをさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 3】

前記サーバデバイスが、同期させる構造化コンテンツの量を決定し、該量に基づいて同期パラメータを設定し、および前記同期パラメータに従って前記クライアントデバイスに同期情報を送信するステップ

をさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 4】

前記クライアントデバイスがプロトコルハンドラを用いてウェブブラウザから同期要求を送信するステップをさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 5】

前記サーバデバイスがプロパティバッグを用いてデータスキーマコンフリクトを検出するステップをさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 6】

前記サーバデバイスがフィールド値マッピングを用いて言語コンフリクトを検出するステップをさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 7】

前記サーバデバイスが、同期間隔パラメータを生成し、前記クライアントデバイスに前記同期間隔パラメータを送信し、および前記同期間隔パラメータに従って前記クライアントデバイスから同期要求を受信するステップ

10

20

30

40

50

をさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 8】

前記サーバデバイスが、前記サーバデバイスによって格納されたコンテンツ変更履歴データまたは前の同期要求からの前の同期結果に基づいてクライアントデバイスに対する同期間隔パラメータを生成し、および前記クライアントデバイスに前記同期間隔パラメータを送信するステップ

をさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 9】

前記サーバデバイスが、内部プライベートネットワークから少なくとも 1 つの構造化コンテンツタイプとともに同期要求を送信するための第 1 のネットワークアドレスを生成するとともに、外部公衆網から少なくとも 1 つの構造化コンテンツタイプとともに同期要求を送信するための第 2 のネットワークアドレスを生成するステップをさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記憶媒体。

【請求項 10】

クライアントデバイスとサーバデバイスとの間でデータを同期させる方法であって、該方法は、

サーバコンテンツリストについての要求をクライアントデバイスが送信するステップであって、該サーバコンテンツリストは前記サーバデバイスによってサポートされる構造化コンテンツタイプを含む、ステップと、

クライアントデバイスが、前記サーバコンテンツリストを受信し、該サーバコンテンツリストをクライアントコンテンツリストと比較するステップであって、該クライアントコンテンツリストは該クライアントデバイスによってサポートされる構造化コンテンツタイプを含む、ステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる 1 つまたは複数の構造化コンテンツタイプをクライアントデバイスが識別するステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる 1 つまたは複数の構造化コンテンツタイプを同期させる同期要求をクライアントデバイスが送信するステップであって、該同期要求は、変更トークン以降に変更された第 1 の定義数のデータ項目を要求することを含み、該第 1 の定義数のデータ項目についての要求は `GetListChangesSinceToken` クエリ上の `rowLimit` プロパティを使用して行われる、ステップと、

前記第 1 の定義数よりも多いデータ項目が変更される時、要求された第 1 の定義数のデータ項目と、`ListItemCollectionPositionNext` の値の開始から第 2 の定義数のデータ項目を要求するための `ListItemCollectionPositionNext` の値とをクライアントデバイスが受信するステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる 1 つまたは複数の構造化コンテンツタイプの一つまたは複数のデータ項目をクライアントデバイスが修正するステップと、

前記一つまたは複数のデータ項目のサーババージョンをサーバデバイスからクライアントデバイスが受信するステップであって、該一つまたは複数のデータ項目のサーババージョンはバージョン履歴を含む、ステップと、

前記一つまたは複数のデータ項目のサーババージョンのバージョン履歴をクライアントデバイスが分析するステップであって、該一つまたは複数のデータ項目のサーババージョンのバージョン履歴は、複数のユーザ識別子および変更番号のペアを含み、ユーザ識別子および変更番号のペアの各々におけるユーザ識別子は一つまたは複数のデータ項目を編集したコンピュータを識別し、ユーザ識別子および変更番号のペアの各々における変更番号は該一つまたは複数のデータ項目になされた変更の順番を示す、ステップと、

一つまたは複数のデータ項目が第 3 のデバイスによって編集されたことを、前記一つま

10

20

30

40

50

たは複数のデータ項目のサーババージョンのバージョン履歴からクライアントデバイスが判定するステップと、

前記クライアントデバイスにて行われた修正が前記一つまたは複数のデータ項目のサーババージョンに含まれることを、前記一つまたは複数のデータ項目のサーババージョンのバージョン履歴からクライアントデバイスが判定するステップと、

前記一つまたは複数のデータ項目のサーババージョンを前記クライアントデバイスが自動的に格納するステップと

を含むことを特徴とする方法。

【請求項 11】

クライアントデバイスとサーバデバイスとの間で構造化コンテンツを同期させるシステムであって、

前記システムは、

プロセッサと、

該プロセッサに結合され、該プロセッサによって実行可能なコンピュータプログラム命令を備えるメモリと

を備え、

前記コンピュータプログラム命令は、前記プロセッサによって実行される場合に、

前記サーバデバイスによってサポートされる一つまたは複数の構造化コンテンツタイプをサーバデバイスが識別するステップと、

前記クライアントデバイスによってサポートされる一つまたは複数の構造化コンテンツタイプを識別するクライアントコンテンツリストをサーバデバイスが受信するステップと

、
サーバデバイスが、前記クライアントデバイスによってサポートされる前記一つまたは複数の構造化コンテンツタイプを前記サーバデバイスによってサポートされる前記一つまたは複数の構造化コンテンツタイプと比較して、共有コンテンツタイプリストを生成するステップであって、該共有コンテンツタイプリストは前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる一つまたは複数のコンテンツタイプを含む、ステップと、

前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる一つまたは複数の構造化コンテンツタイプを同期させる同期要求をサーバデバイスが受信するステップであって、前記クライアントデバイスおよび前記サーバデバイスの両方によってサポートされる一つまたは複数の構造化コンテンツタイプのバージョン履歴が該同期要求より前に第3のデバイスによって修正されたか否かについての判定は、同期させることより前に行われ、該同期要求は、GetListChangesSinceTokenクエリ上のrowLimitプロパティを使用する要求であって、変更トークン以降に変更された第1の定義数のデータ項目の要求を含む、ステップと、

前記第1の定義数よりも多いデータ項目が変更される時、要求された第1の定義数のデータ項目と、ListItemCollectionPositionNextの値の開始から第2の定義数のデータ項目を要求するためのListItemCollectionPositionNextの値とをクライアントデバイスに送信するステップと、

を含む方法を前記プロセッサに実行させることを特徴とするシステム。

【請求項 12】

前記サーバデバイス及び前記クライアントデバイスによってサポートされる構造化コンテンツタイプを同期させる同期要求を送信するように構成されたクライアント同期モジュールをさらに備えることを特徴とする請求項 11 に記載のシステム。

【請求項 13】

クライアントコンテンツリストを受信し、前記クライアントデバイスによってサポートされる構造化コンテンツタイプを前記サーバデバイスによってサポートされる構造化コンテンツタイプと比較し、一致している構造化コンテンツタイプとともに共有コンテンツリストを送信し、同期要求を受信して前記共有コンテンツリストに従って構造化コンテンツ

10

20

30

40

50

タイプを同期させるように構成されたサーバ同期モジュールをさらに備えることを特徴とする請求項 11 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンテンツの同期化 (synchronization) 技術に関する。

【背景技術】

【0002】

ネットワークソフトウェアアプリケーションは、通常、アプリケーションデータ用及び/またはユーザデータ用の集中ネットワークデータベースを維持する。場合によっては、デバイスは、集中データベースのサブセットをネットワークデータベースからダウンロードまたは複製して、次に、ネットワーク接続を切ることができる。例えば、無線携帯端末は、ウェブサイトなどの集中データベースから、カレンダー及び連絡先をダウンロードすることができる。ネットワークデータベースにより維持されている情報が修正されるかまたは、情報の複製されたサブセット自体が修正される場合、両方のデータ記憶場所におけるかかる変更を更新するのに同期イベントが必要かもしれない。例えば、ユーザがカレンダーアプリケーションに新規のアポイントメントを追加すると仮定する。携帯端末がネットワークデータベースとの接続を確立すると、ネットワークデータベースに記憶されたカレンダー情報は、携帯端末からの修正されたデータを反映するために更新されることを必要とするかもしれない。逆もまた同様である。

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、ネットワークデータベースと同期しようとするデバイス数と同様に、アプリケーションデータの量が増加するにつれて、同期イベントはますます時間集中及び帯域幅集中が生じるかもしれない。従って、これらの課題及び他の課題を解決するために改良された同期技術が必要になるかもしれない。

【課題を解決するための手段】

【0004】

この概要は、簡易化した形式でいくつかの概念を説明し、当該概念の詳細は以下の実施形態で詳しく説明する。この概要は、特許請求の範囲に記載された発明の重要な特徴または本質的特徴を特定することを意図していないし、特許請求の範囲に記載された発明の範囲を限定するために用いられることを意図していない。

【0005】

様々な実施形態は、概して、物理的エンティティまたは論理的エンティティ同士の間で構造化コンテンツを同期化する技術を対象にすることができる。より詳しくは、様々な実施形態は、クライアントとサーバとの間の同期化技術を対象にすることができる。例えば、いくつかの実施形態では、サーバとクライアントとの間で共有されている特定の構造化コンテンツタイプを、同期動作の前に識別することができる。該構造化コンテンツタイプに対応する構造化コンテンツは、識別された時点で同期され得る。このように、同期は、クライアント及び/またはサーバによって記憶されているデータセット全体よりはむしろ、構造化コンテンツタイプによって識別されるような共有構造化コンテンツに焦点を合わせることができる。従って、同期時間及び資源を削減し、それによって全体的なデバイス動作及びユーザに対するネットワークサービスを強化することができる。

【0006】

1つの実施形態において、クライアントは発見動作または識別動作を実行することができる。例えば、サーバは、サーバコンテンツリスト要求を受信することができる。サーバは、サーバでサポートされる構造化コンテンツタイプを表している構造化コンテンツ値とともにサーバコンテンツリストを検索して送信することができる。クライアントは、サーバコンテンツリストを受信し、クライアントでサポートされる構造化コンテンツタイプを

表している構造化コンテンツ値を有するクライアントコンテンツリストをサーバコンテンツリストと比較することができる。クライアントは、同期要求及び／または共有コンテンツリストを送信して、サーバ及びクライアントでサポートされる構造化コンテンツタイプを同期させることができる。サーバは、同期要求及び／または共有コンテンツリストを受信して、それに応じて同期動作を実行することができる。

【 0 0 0 7 】

1つの実施形態において、サーバは発見動作または識別動作を実行することができる。例えば、サーバはクライアントコンテンツリストを受信することができる。サーバは、サーバコンテンツリストを検索して、両方のコンテンツリストを比較することができる。サーバは、該比較に従ってクライアント及びサーバでサポートされている構造化コンテンツタイプを表している構造化コンテンツ値を有する共有コンテンツリストを生成することができる。サーバは、共有コンテンツリストをクライアントに送信することができる。クライアントは、共有コンテンツリストを受信し、同期要求を送信して共有コンテンツリストに従って構造化コンテンツタイプを同期させることができる。サーバは、同期要求を受信し、該同期要求に従って同期動作を実行することができる。他の実施形態が開示され、特許請求の範囲に記載される。

【 発明を実施するための最良の形態 】

【 0 0 0 8 】

図1は、ネットワークの1つの例示的な実施形態を示す。図1は、ネットワーク100のブロック図を示す。ネットワーク100は、様々な実施形態を実施するのに適した一般的なネットワークアーキテクチャを示すことができる。ネットワーク100は、複数の要素を含むことができる。ある要素は、特定の動作を実行する任意の物理的構造または論理的構造を含むことができる。所定のデザインパラメータまたは性能制約に要求されるように、各々の要素を、ハードウェア、ソフトウェア、またはそれらの任意の組合せとして実施することができる。ハードウェア要素の例には、プロセッサ、マイクロプロセッサ、回路、回路素子（例えば、トランジスタ、抵抗器、コンデンサ、インダクタなど）、集積回路、特定用途向け集積回路（ASIC）、プログラム可能論理回路（PLD）、デジタルシグナルプロセッサ（DSP）、フィールドプログラマブルゲートアレイ（FPGA）、記憶装置、論理ゲート、レジスタ、半導体デバイス、チップ、マイクロチップ、チップセットなどを含むことができる。ソフトウェアの例には、任意のソフトウェアコンポーネント、プログラム、アプリケーション、コンピュータプログラム、アプリケーションプログラム、システムプログラム、マシンプログラム、オペレーティングシステムソフトウェア、ミドルウェア、ファームウェア、ソフトウェアモジュール、ルーチン、サブルーチン、関数、方法、インタフェース、ソフトウェアインタフェース、アプリケーションプログラミングインタフェース（API）、命令セット、演算コード、コンピュータコード、コードセグメント、コンピュータコードセグメント、ワード、値、シンボルまたはそれらの任意の組合せを含むことができる。図1に示すネットワーク100は、特定のトポロジにおける限定された要素数を有しているが、ネットワーク100は、所定の実施に要求されるように、代わりのトポロジにおいてより多くの要素またはもっと少ない要素を含んでもよいということが理解されるだろう。実施形態は、上記の説明に限定されない。

【 0 0 0 9 】

例えば、図1に示すように、ネットワーク100は、コンピュータアレイ140、サーバ150、ネットワーク160、ロードバランサ170、及びクライアントデバイス180-1-sなどの様々な要素を含むことができる。例えば、1つの実施形態において、サーバ150は、ウェブサーバとして実施されてもよい。ウェブサーバは、クライアントデバイス（例えば、クライアント180-1-s及び／またはコンピュータデバイス122-1-n）からハイパーテキスト転送プロトコル（HTTP）要求を受理してウェブページを供給することができるコンピュータデバイスを含んでもよい。ウェブページは、通常、ハイパーテキストマークアップ言語（HTML）、拡張マークアップ言語（XML）、拡張可能ハイパーテキストマークアップ言語（XHTML）、マイクロソフトウェア（登録

10

20

30

40

50

商標)マークアップ言語(W o r d M L)などのマークアップ言語の何らかの形式を用いて生成される文書である。ネットワーク160は、伝送制御プロトコル及びインターネットプロトコル(T C P / I P)などの1つまたは複数のインターネットプロトコルを用いるパケット網を含んでもよい。ロードバランサ170は、コンピューティング資源が効率的な方法で用いられるような方法でネットワーク化されたコンピュータサーバの集合(例えば、コンピュータアレイ140など)に仕事を割り当てるデバイスを含むことができる。ロードバランサ170は、例えば、コンピュータ、サーバ、仮想サーバ、ネットアプライアンスなどを用いて実施されてもよい。

【0010】

様々な実施形態において、ネットワーク100は、様々なコンピュータデバイスを含むことができる。1つの実施形態では、例えば、コンピュータアレイ140は、複数のコンピュータデバイス122-1-nを含むことができる。同様に、クライアントデバイス180-1sも、様々なタイプのコンピュータデバイスとして実施され得る。コンピュータデバイスの例には、コンピュータ、コンピュータシステム、コンピュータサブシステム、ワークステーション、ターミナル、サーバ、ウェブサーバ、仮想サーバ、パーソナルコンピュータ(PC)、デスクトップコンピュータ、ラップトップコンピュータ、ウルトララップトップコンピュータ、ポータブルコンピュータ、ハンドヘルドコンピュータ、携帯情報端末(PDA)、モバイルコンピュータデバイス、携帯電話、一体型携帯電話/PDA、デジタルビデオデバイス(例えば、デジタルカメラ、ビデオカメラまたはカムコーダ)、デジタルオーディオデバイス(例えば、MP3プレーヤ)、片方向ページャ、両方向ページャ、前述の例のいずれかの仮想インスタンス化、及び他の任意の電子デバイス、電気機械デバイス、もしくは電気デバイスをも含むことができるが必ずしもこれらに限定されるわけではない。実施形態は、上記の説明に限定されない。

【0011】

例えば、1つの実施形態では、コンピュータアレイ140は、サーバファームとして実現され得る。コンピュータデバイス122-1-nは各々、サーバ、仮想サーバ、バーチャルマシン、シングルボードコンピュータ(SBC)、サーバブレード、ウェブサーバなどを示す。サーバファームは、通常、企業によって通常維持されるコンピュータサーバの一群であって、1台のマシンの能力を超えたサーバのニーズを達成する。しばしば、サーバファームは、1つのタスクに割り当てられたプライマリサーバ及びバックアップサーバの両方を有し、プライマリサーバの障害の場合には、バックアップサーバがプライマリサーバの機能を引き継ぐ。サーバファームは、例えば、ウェブホストサービスを提供するためによく用いられる。ウェブホストサービスは、情報、画像、映像、音声、テキスト、動画、映画、ピクチャ、またはワールドワイドウェブ(WWWまたは「ウェブ」)を介してアクセス可能なウェブコンテンツの任意の他の形式を記憶するオンラインシステムを個人及び組織に提供する、インターネットホストサービス的一种である。ウェブホストは、通常データセンタにおいて、自社のクライアントに対して自社が所有するサーバ上のスペース及びインターネット接続を提供する会社である。ウェブホストはまた、自社で所有しておらず自社のデータセンタに位置すべきサーバに対するインターネットへのデータセンタのスペース及び接続を提供することができる。

【0012】

図1は、コンピュータデバイス122-1のより詳細なブロック図を更に示している。コンピュータデバイス122-1は、任意のコンピュータデバイス122-1-nの代表であってよい。更に、コンピュータデバイス122-1はまた、任意のクライアントデバイス180-1sの代表であってもよい。その最も基本的な構成106では、コンピュータデバイス122-1は、通常、少なくとも1つの処理装置102及びメモリ104を含む。メモリ104は、揮発性メモリ及び不揮発メモリを含む、データを記憶することができる任意の機械可読媒体またはコンピュータ可読媒体を用いて実施され得る。例えば、メモリ104は、リードオンリメモリ(ROM)、ランダムアクセスメモリ(RAM)、ダイナミックRAM(DRAM)、ダブルデータレートDRAM(DDRAM)、同期型D

10

20

30

40

50

RAM (SDRAM)、スタティックRAM (SRAM)、プログラマブルROM (PROM)、消去可能ROM (EPROM)、電氣的消去可能ROM (EEPROM)、フラッシュメモリ、強誘電性ポリマーメモリなどのポリマーメモリ、オボニックス (ovonic) メモリ、相変化メモリもしくは強誘電性メモリ、SONOS (シリコン/酸化膜/窒化膜/酸化膜/シリコン: oxide-nitride-oxide-silicon) メモリ、磁気カードもしくは光カード、または、情報を記憶するのに適した任意の他のタイプの媒体など、を含むことができる。図1に示すように、メモリ104は、例えば、1つ又は複数のアプリケーションプログラム118、サーバ同期モジュール120、及び付随データなどの、様々なソフトウェアプログラムを記憶することができる。

【0013】

コンピュータデバイス122-1はまた、構成106を超えたさらなる特徴及び/または機能性を有することができる。例えば、コンピュータデバイス122-1は、着脱自在の記憶装置108及び固定式記憶装置110を含むことができ、それらも、前述のように様々なタイプの機械可読媒体またはコンピュータ可読媒体を含むことができる。コンピュータデバイス122-1は、また、例えばキーボード、マウス、ペン、音声入力デバイス、タッチ入力デバイスなどの1つまたは複数の入力デバイス114を有することができる。ディスプレイ、スピーカ、プリンタなどの1つまたは複数の出力デバイス116も、コンピュータデバイス122-1に含まれ得る。

【0014】

コンピュータデバイス122-1は、コンピュータデバイス122-1が他のデバイスと通信するのを可能にする1つまたは複数の通信接続部112を更に含むことができる。通信接続部112は、例えば1つまたは複数の通信インタフェース、ネットワークインタフェース、ネットワークインターフェースカード (NIC)、ラジオ、無線送信器/受信器 (送受信器)、有線通信媒体及び/または無線通信媒体、物理的コネクタなどの様々なタイプの標準通信要素を含むことができる。通信媒体は、通常、コンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを搬送波または他の移送メカニズムなどの変調されたデータ信号内の他のデータを具体化し、任意の情報配信媒体を含む。用語「変調されたデータ信号」は、信号内に情報をコード化するような方法で設定されたかまたは変更されたその特徴セットのうちの1つまたはいくつかを有する信号を意味する。一例であってこの例に限らないが、通信媒体は、有線通信媒体及び無線通信媒体を含む。有線通信媒体の例は、導線、ケーブル、金属リード、プリント基板 (PCB)、バックプレーン、スイッチ構成、半導体物質、ツイストペア線、同軸ケーブル、光ファイバ、伝播信号などを含むことができる。無線通信媒体の例は、音響、無線周波数 (RF) スペクトル、赤外線及び他の無線媒体を含むことができる。本明細書において用いられる用語である機械可読媒体及びコンピュータ可読媒体は、記憶媒体及び通信媒体の両方を含むことが意図されている。

【0015】

コンピュータデバイス122-1-nの各々は、複数のデータベースを含むことができる。コンピュータデバイス122-1に関して詳細に示すように、コンピュータデバイス122-1は、データベース124、126に結合され得る。各々のデータベースは、コンピュータデバイス122-1、アプリケーションプログラム118、サーバ同期モジュール120などに対する様々なタイプのデータを記憶することができる。1つの実施形態では、例えば、データベース124は、クライアントコンテンツリスト130、サーバコンテンツリスト132、共有コンテンツリスト134などを記憶することができる。クライアントコンテンツリスト130は、1つまたは複数のクライアント180-1-sなどのクライアントにサポートされている構造化コンテンツタイプを表している構造化コンテンツ値を含むことができる。サーバコンテンツリスト132は、1つまたは複数のサーバ122-1-nなどのサーバにサポートされている構造化コンテンツタイプを表している構造化コンテンツ値を含むことができる。共有コンテンツリスト134は、1つまたは複数のクライアント180-1-s及び1つまたは複数のサーバ122-1-nにより共有されている

10

20

30

40

50

構造化コンテンツタイプを表している構造化コンテンツ値を含むことができる。他のタイプのデータ及びデータベースを、コンピュータデバイス 122-1 で実施することができるが、実施形態は、上記の説明に限定されない。

【0016】

一般的な動作において、ネットワーク 100 は、コンピュータレイ 140 のうちの 1 つまたは複数のコンピュータデバイス 122-1-n によって記憶されるかまたは実行される 1 つまたは複数のアプリケーションプログラムと、 1 つまたは複数のクライアントデバイス 180-1-s によって記憶されるかまたは実行される 1 つ以上のアプリケーションプログラムとの間の同期動作を容易にすることができる。コンピュータデバイス 122-1-n の各々がサーバベースのウェブアプリケーションプログラムのような 1 つまたは複数のアプリケーションプログラム 118 を実行すると仮定する。サーバベースのウェブアプリケーションプログラムの例は、本件特許出願人により製造されているウィンドウズ（登録商標）SHAREPOINT（登録商標）サービス（WSS）バージョン 2.0、バージョン 3.0、及び改良版（本明細書では「WSS アプリケーション」または「シェアポイントアプリケーション」と呼ぶ）を含むことができる。WSS バージョン 3.0 は、例えば、クライアント 180-1-s などの様々なクライアントに対する基本的チームサービス、ウェブポータル及びイントラネット機能を提供する一組の統合アプリケーションプログラムである。WSS バージョン 3.0 に対するユーザデータまたはコンテンツデータは、データベース 124、126 で記憶され得る。更に、1 つまたは複数のコンピュータデバイス 122-1-n は、例えば、シェアポイントポータルサーバ（SPS）2003、または、オフィスシェアポイントサービス（OSS）2006 などの、WSS バージョン 3.0 上に構築されるサードパーティアプリケーションを実施することができる。いくつかの実施形態が、一例として WSS アプリケーションプログラム及び付随するサードパーティプログラムを参照して開示されるが、単一のコンピュータデバイスまたは複数のコンピュータデバイスによって実行されるいかなるアプリケーションプログラムも、本明細書で開示されている同期技術を用いて同期することができるということが理解されるだろう。実施形態は、上記の説明に限定されない。

【0017】

WSS バージョン 3.0 は、複数のユーザに対するプラットフォームを提供して効率的な方法でデータを共有する。例えば、ユーザはチームに組織され、各々のチームは、共有カレンダー、連絡先リスト、作業リスト、電子メール、掲示板、文書などを有することができる。かかる共有データの有用性を増すために、ネットワーク 100 の様々な要素を、間欠的に（intermittently）同期されるデータベースシステム（ISDB）として組織することができる。ISDB は、様々な物理的クライアントエンティティまたは論理的クライアントエンティティが、クライアントデバイス 180-1 などのクライアントデバイスに対してデータベース 124、126 から情報の一部をダウンロードまたは複製することを許容する。クライアントデバイス 180-1 が本件特許出願人によって製造されるマイクロソフトアウトLOOK（商標）のようなアプリケーションを含むと仮定する。マイクロソフトアウトLOOK は、WSS バージョン 3.0 を用いて維持されるようにサーバ 122-1 から共有データをインポートすることができる。従って、ユーザは、マイクロソフトアウトLOOK のローカライズ版を実行するハンドヘルドコンピュータデバイスなどの携帯用デバイスを有し、サーバ 122-1 から共有データの一部をダウンロードすることができる。クライアントデバイス 180-1 は、アポイントメントの追加、連絡先の削除、作業の変更、ノートの生成、プロジェクトスケジュールの更新などで、サーバ 122-1 との常時接続なしに（例えば、オフライン）局所的に複製されたデータを修正することができる。一方、他のユーザも、同様にデータベース 124、126 から局所的に複製されたデータを修正することができる。データ完全性を維持するために、クライアントデバイス 180-1 は、サーバ 122-1 と周期的にまたは間欠的に同期して、両方のデバイス間で共有される任意の構造化コンテンツまたはデータセットに対する変更を反映することを必要とするかもしれない。

10

20

30

40

50

【 0 0 1 8 】

同期動作を実行するために、各々のクライアントデバイス 1 8 0 - 1 - s は、クライアント同期モジュール 1 8 2 を含むことができ、各々のサーバ 1 2 2 - 1 - n は、サーバ同期モジュール 1 2 0 を含むことができる。同期モジュール 1 2 0、1 8 2 は、様々な統合レベルを有していて、サーバ 1 2 2 及びクライアントデバイス 1 8 0 がそれぞれ、両方のデバイスでサポートされている構造化コンテンツまたはデータセットを同期させることを可能にすることができる。1つの実施形態では、例えば、クライアントデバイス 1 8 0 は、サーバ 1 2 2 によって実行されるような W S S アプリケーションとある程度ではあるが統合されるマイクロソフトアウトルッククライアントアプリケーションを含むことができる。かかる統合は、マイクロソフトアウトルックと W S S アプリケーションとの間の同期データを含むことができる。例えば、W S S アプリケーションは、W S S アプリケーションに記憶される共有カレンダー、連絡先、作業、電子メール及び掲示板を含むことができる。様々な実施形態は、マイクロソフトアウトルックユーザに対するユーザモデル、ドライバ、仕様、データスキーマ及び同期インフラを定義して、W S S アプリケーションデータと相互に作用することができる。いくつかの実施形態では、ほんの一例としてマイクロソフトアウトルックと W S S アプリケーションとの間で構造化コンテンツを同期するように記述されているが、一組の所定のデザインパラメータ及び性能制約に要求されるように、構造化コンテンツの同期はいかなるアプリケーションプログラム及びコンピューティングプラットフォームに対しても実行され得るということが認められるだろう。実施形態は、上記の説明に限定されない。

10

20

【 0 0 1 9 】

サーバ 1 2 2 - 1 及びクライアントデバイス 1 8 0 - 1 などの 2 つ以上のアプリケーション及び/またはデバイスを同期させようとするときに、いくつかのデザイン上の考慮すべき事柄がある。例えば、データの中央保管所としてウェブサイトを用いるために、データは、ウェブサイトサーバ上で最小の負荷を生成しつつ、素早く他のアプリケーションに同期しなければならない。更に、ユーザはしばしば、データ全体のうちの小さい範囲だけに興味があり、従って、ユーザは、興味のある関連データに対する同期動作を開始するために容易に発見できる場所を必要とする。更に、異なる同期クライアントは、サーバからの特定のタイプのデータと同期することができるだけであるかもしれない。同様に、いくつかの同期クライアントは、ウェブサイトサーバでサポートされないデータタイプまたは値をサポートするかもしれない。その一方で、ウェブサイトサーバは、同期クライアントでサポートされないデータタイプまたは値をサポートするかもしれない。別の例では、ウェブサイト及び同期クライアントは、異なった言語で実行されているかもしれない。また別の例では、複数の同期クライアントは、サーバに接続されていないが潜在的に同じデータを更新することができ、従って、データが同期イベントの間に失われないことを確実にする手段が取られることを必要とする。更に別の例では、ユーザは、プライベートネットワーク内及びインターネットなどの公衆網内の両方で同じウェブサイトと同期することを必要とするかもしれない。また別の例では、サーバからのコンテンツを同期させているクライアントは、コンテンツが変化してサーバ上の余分の負荷または同期したクライアント上の古いコンテンツのいずれかをもたらす頻度を正確に予測することができない。最終的に、サーバ管理者は、クライアントを同期させることを介してサーバに発生する負荷の制御を可能にすることを所望する。

30

40

【 0 0 2 0 】

これらの課題及び他の課題を解決するために、サーバ 1 2 2 及び/またはクライアントデバイス 1 8 0 は、構造化コンテンツを利用する改善された同期技術を実施することができる。構造化コンテンツは、幅、長さ、ビットサイズ、バイトサイズ、構文、フィールド、値、コード、フラグなどの 1 つまたは複数の画定された構造上のパラメータを有している任意のタイプの媒体コンテンツ（例えば、テキスト、音声、映像、画像、ピクチャ、動画、シンボル、キャラクタ、数、アイコン、その他）を示すことができる。構造化コンテンツは、アプリケーションデータ及び/またはユーザデータを含み、マイクロソフトアウ

50

トルック、マイクロソフトアクセス（商標）、マイクロソフトフロントページ（商標）、マイクロソフトワンノート（商標）、マイクロソフトパワーポイント（商標）、マイクロソフトワード（商標）、マイクロソフトビジオ（商標）、W S Sアプリケーション、S P Sアプリケーションなどの任意の数のアプリケーションプログラムをサポートすることができる。例えば、1つの実施形態では、構造化コンテンツは、例えば、カレンダーリスト、作業リスト、連絡先リスト、ノートまたはメモのリスト、メールリストなどから1つまたは複数の項目などのマイクロソフトアウトルックに対するアプリケーションデータ及び/またはユーザデータを含むことができる。サーバ同期モジュール120及びクライアント同期モジュール182は、統合されて、サーバ122とクライアント180との間で共有される構造化コンテンツタイプを識別することができる。次に、サーバ122とクライアント180との間で共有される構造化コンテンツタイプに対応する構造化コンテンツ情報を同期させることができる。

10

【0021】

構造化コンテンツを用いて同期動作を実行することにより、従来の同期技術に勝るいくつかの利点を提供することができる。様々な実施形態において、例えば、デバイス同士の間の同期動作を開始することを、多くの方法で改善することができる。例えば、ネットワークアドレス及び/またはハイパーリンクが、同期クライアントに対して生成されて、サーバ及び/またはクライアントでサポートされる特定のコンテンツタイプに基づいて、同期動作を開始することができる。異なるハイパーリンクが生成されて、異なるコンテンツの範囲に対する同期動作を開始することができる。別の例では、同期動作は、同期されているコンテンツの量に基づいて異なって開始されてもよい（例えば、ヘッダだけを同期させることにに対してコンテンツセット全体を同期させる）。また別の例では、プロトコルハンドラを、ウェブブラウザから同期動作を開始するために用いてもよい。

20

【0022】

一旦同期動作が開始されると、様々な実施形態はコンテンツが実際に同期する方法を改善することができる。例えば、サーバ側のチェンジログが、変更されたコンテンツだけを同期させるために用いられてもよい。トークンが前後にパスされて、前の同期イベント以後になされた新規な変更だけを得ることができる。別の例では、同期動作は、構造化スキーマの個々に要求されたかまたは画定されたフィールドをサポートするだけであってもよい。また別の例では、同期動作は、フォルダ階層において生成される階層的スコープのような異なった階層的スコープをサポートしてもよい。また別の例では、同期図が生成されて、様々なフィールド値によって決定されるフィルタリングされた一組のコンテンツを表示することができる。また別の例では、いくつかの実施形態は、ページング技術を用いて非常に大きいデータセットの同期をサポートすることができる。別の例では、いくつかの実施形態は、バージョン履歴を用いてコンフリクト検出及び自動解消を実行することができる。もう1つの例では、いくつかの実施形態は、コンフリクト解消のために「E - T a g s」を用いている添付ファイルの同期をサポートすることができる。別の例では、いくつかの実施形態は、プロパティバッグを用いて同期ウェブサイトとクライアントとの間のスキーマの不整合をサポートすることができる。更なる例では、いくつかの実施形態は、フィールド値マッピングを用いてサーバ/クライアント言語不整合をサポートすることができる。また別の例では、いくつかの実施形態は、読み出し/送信/リフレッシュのパターンを用いて読み出し及び更新をサポートすることができる。最終的に、いくつかの実施形態は、内部プライベートネットワークまたは外部公衆網からウェブサイトにアクセスするために、同期することができる代替りのU R L（universal resource locator）をクライアントに提供することができる。

30

40

【0023】

同期動作を更に改善するために、様々な実施形態は、クライアントデバイスをウェブサイト同期させるべきときに関するタイミングの問題を改善することができる。例えば、いくつかの実施形態は、望ましい同期間隔を同期クライアントに提供することによってクライアント負荷及び/またはサーバ負荷を管理することができる（例えば、クライアント

50

はP分ごとに同期化する)。望ましい同期間隔は、最小同期間隔、最大同期間隔、または、いくつかの中間値を示すことができる。望ましい同期間隔は、サーバに記憶されたコンテンツ変更履歴データ(例えば、格子同期)、またはクライアントからの前の同期要求の結果(例えば、指数関数的減衰)に基づいていてもよい。

【0024】

構造化コンテンツをサーバ122と同期させるために、クライアントデバイス180は、SharePoint.StssyncHandler.ActiveX.Controlを用いることができる。このコントロールは、マイクロソフトアウトLOOKとともにクライアントデバイス180上にインストールされ、マイクロソフトアウトLOOKがクライアントデバイス180上にインストールされてリストの特定のタイプに同期させることができる否かを示す。例えば、WSSバージョン3.0では、同期され得るリストのタイプには、カレンダーリスト、連絡先リスト、文書リスト、作業リスト、掲示板リストなどを含む。

10

【0025】

上記の実施形態に対する動作を、以下の図及び添付の例を参照して更に説明することができる。いくつかの図は、論理フローを含むことができる。本明細書に示したかかる図は特定の論理フローを含むことができるが、本明細書において開示したような一般的な機能が実行され得る方法の例を単に提供するだけであるということが理解されるだろう。更に、所定の論理フローは、特に明記しない限り示される命令で必ずしも実行されなければならないというわけではない。更に、所定の論理フローはハードウェア要素、プロセッサによって実行されるソフトウェア要素、またはそれらの任意の組み合わせによって実行され得る。実施形態は、上記の説明に限定されない。

20

【0026】

図2は、論理フローの1つの実施形態を示す。図2は、論理フロー200を示す。論理フロー200は、ネットワーク100、コンピュータデバイス122-1-n、及び/またはクライアントデバイス180-1-sなどの、本明細書において開示されている1つまたは複数の実施形態で実行される動作の代表であってもよい。いくつかの実施形態では、例えば、サーバとクライアントとの間で共有される特定の構造化コンテンツタイプを、ブロック202における同期動作の前に識別することができる。構造化コンテンツタイプに対応する構造化コンテンツは、ブロック204で一旦識別されると、次に同期され得る。このように、同期は、クライアント及び/またはサーバによって記憶された一組のデータ全体よりむしろ、構造化コンテンツタイプによって識別されるような共有された構造化コンテンツに焦点を合わせることができる。従って、同期時間及び資源を削減し、それによってユーザに対する全体的なデバイス動作及びネットワークサービスを向上させることができる。

30

【0027】

図3は、第1のメッセージフローの1つの実施形態を示す。図3は、クライアントが識別動作を実行するメッセージフロー300を示す。図3に示すように、クライアントデバイス180は、サーバ122にメッセージ302を送信することができる。メッセージ302はサーバコンテンツリスト要求を含む。サーバ122は、メッセージ302を受信して、データベース124からサーバコンテンツリスト132を検索することができる。サーバ122はクライアントデバイス180にメッセージ304を送信することができる。メッセージ304は、サーバ122でサポートされる構造化コンテンツタイプを表す構造化コンテンツ値とともにサーバコンテンツリスト132を含む。クライアントデバイス180は、サーバコンテンツリスト132を受信して、クライアントコンテンツリスト130を検索することができる。クライアントコンテンツリスト130は、クライアントでサポートされる構造化コンテンツタイプを表している構造化コンテンツ値を含むことができる。クライアントデバイス180は、サーバコンテンツリスト132をクライアントコンテンツリスト130と比較して一致するコンテンツタイプを検出することができる。クライアントデバイス180は、共有コンテンツリスト134を生成して、サーバ122にメ

40

50

ッセージ306を送信することができる。メッセージ306は、同期要求及び/または共有コンテンツリスト134を含むことができる。サーバ122はメッセージ306を受信し、サーバ122及びクライアントデバイス180は、同期動作を開始して、矢印308で示したように、サーバ122及びクライアント180でサポートされる構造化コンテンツタイプを同期させることができる。

【0028】

図4は、第2のメッセージフローの1つの実施形態を示す。図4は、サーバが識別動作を実行するメッセージフロー400を示す。図4に示すように、サーバ122は、メッセージ402を介してクライアントコンテンツリスト130を受信することができる。サーバ122は、サーバコンテンツリスト132を検索して、両方のコンテンツリストを比較することができる。サーバ122は、比較に従ってクライアントデバイス180及びサーバ122でサポートされる構造化コンテンツタイプを表している構造化コンテンツ値を有する共有コンテンツリスト134を生成することができる。サーバ122は、メッセージ404を介してクライアントデバイス180に共有コンテンツリスト134を送信することができる。クライアントデバイス180は、共有コンテンツリスト134を受信し、メッセージ406を介して共有コンテンツリスト134に従って構造化コンテンツタイプを同期させる同期要求を送信することができる。サーバ122は同期要求を受信し、サーバ122及びクライアントデバイス180は、矢印408で示したように、同期要求に従って同期動作を実行することができる。

【0029】

様々な実施形態において、デバイス同士の間の同期動作を開始することを、多くの方法で改善することができる。例えば、ネットワークアドレス及び/またはハイパーリンクを、同期クライアントに対して生成し、サーバ及び/またはクライアントでサポートされる特定のコンテンツタイプに基づいて同期動作を開始することができる。異なるハイパーリンクが生成されて、異なるコンテンツ範囲に対する同期動作を開始することができる。1つの実施形態では、サーバ同期モジュール120は、ハイパーリンクを生成して、少なくとも1つの構造化コンテンツタイプとともに同期要求を送信することができる。その結果、ユーザは、特定の構造化コンテンツタイプをサーバ122と継ぎ目なく選択的に同期させることができる。

【0030】

様々な実施形態において、同期されているコンテンツ量に基づいて異なった形で同期動作を開始することができる。例えば、1つの実施形態では、サーバ同期モジュール120は、同期させる構造化コンテンツの量を決定することができる。サーバ同期モジュール120は、決定された量に基づいて同期パラメータを設定することができる。サーバ122-1は、同期パラメータに従ってクライアント180-1に同期情報を送信することができる。例えば、構造化コンテンツタイプが電子メールメッセージであると仮定する。決定された量がサーバ122-1とクライアント180-1との間で同期されるべき大容量の電子メールメッセージである場合、サーバ122-1は、同期パラメータを設定して、電子メールコンテンツ全体よりも電子メールヘッダだけを最初に同期させるべきであるということを示すことができる。逆に、決定された量が小容量の電子メールメッセージである場合、サーバ122-1は、同期パラメータを設定して、電子メールコンテンツ全体がクライアント180-1と同期するべきであるということを示すことができる。別の例では、構造化コンテンツタイプが文書であると仮定する。決定された量がサーバ122-1とクライアント180-1との間で同期されるべき大容量の文書である場合、サーバ122-1は、同期パラメータを設定して、各々の文書の本文全体よりも各々の文書の特定の部分だけが最初に同期されるべきであるということを示すことができる。逆に、決定された量が小容量の文書である場合、サーバ122-1は、同期パラメータを設定して、各々の文書のコンテンツ全体がクライアント180-1と同期するべきであるということを示すことができる。この同じ技術は、また、文書の全体サイズを調べることにより用いられてもよく、大容量の文書は一回に送信される部分だけを有し、小容量の文書は1つのトランザクシ

10

20

30

40

50

ョンで全体が送信される。これらは単に2、3の例であって、構造化コンテンツタイプの他の特徴を用いて同期パラメータを設定することができる。

【0031】

様々な実施形態において、プロトコルハンドラを、ウェブブラウザから同期動作を開始するために用いることができる。一旦、共有構造化コンテンツタイプが識別されると、クライアント180-1のウェブブラウザは、前述のように、1つまたは複数のURLを提供して同期動作を開始することができる。これを実現するために、クライアント同期モジュール182は、プロトコルハンドラを用いて、サーバ122-1に異なった同期要求を送信することができる。

【0032】

一旦同期動作が開始されると、様々な実施形態によりコンテンツが実際に同期する方法を改善することができる。例えば、サーバ側のチェンジログを用いて、変更コンテンツだけを同期させることができる。例えば、1つの実施形態では、サーバ同期モジュール120は、同期要求を受信して、特定の構造化コンテンツタイプを同期させることができる。サーバ同期モジュール120は、構造化コンテンツタイプによって示された構造化コンテンツがチェンジログを用いて修正されたか否かを判断することができる。サーバ122-1は、この判断に基づいてクライアントに同期情報を送信することができる。

【0033】

読み出し/書き込み同期について、画定された技術が、WSSアプリケーションから変更を同期させるためにマイクロソフトアウトLOOKに必要である。このことは、チェンジログを用いかつウェブサービスAPIを付随することで実行することができる。例えば、WSSバージョン3.0は、仮想サーバ、サイト及びリスト毎にチェンジログをサポートする。このことは、特に、追加され、編集され、削除され、名前を変えられ、さらに移動された項目に対してポーリングする能力をマイクロソフトアウトLOOKに許容することができる。更に、ウェブサービスが実施されて、チェンジログ項目についてのバルクの「get」を実行することをマイクロソフトアウトLOOKに許容することができる。

【0034】

様々な実施形態は、構造化コンテンツの複数のバージョンがそれぞれに修正されたか否かをバージョン履歴を用いて判断することができる。いくつかの実施形態は、バージョン履歴を用いてコンフリクト検出及び自動解消を実行することができる。バージョン履歴は、ピアツーピア環境において特定のクラスの同期コンフリクトを自動的に解消するために用いられるメカニズムである。特に、それは、ある項目に対する恒常的な連続的変更の結果であるコンフリクトを自動的に解消することができる。

【0035】

一例として、第1のユーザが、オフィスで用いられるデスクトップコンピュータなどの第1のクライアントデバイス180-1に項目を同期させると仮定する。第1のユーザは、項目に2、3の変更を行うが、サーバ122-1へそれらの変更をコピーしない。第1のユーザは、第1のクライアントデバイス180-1をPDAなどの第2のクライアントデバイス180-2と同期させて、オフィスを出て家に向かう。第1のユーザが帰宅すると、第1のユーザは、第2のクライアントデバイス180-2を家庭用PCなどの第3のクライアントデバイス180-3と同期させる。第1のユーザは、第3のクライアントデバイス180-3からサーバ122-1へ該項目を同期させる前に、該項目に更なる変更を行う。その翌日、第1のユーザは、仕事に行き、第1のクライアントデバイス180-1をサーバ122-1と同期させる。第1のクライアントデバイス180-1は、それが有する更新された項目をサーバ122-1に送信しようとするが、サーバ122-1は、第1のクライアントデバイス180-1との前の同期以来の該項目に対する変更を検出し、従って、コンフリクトをもたらす。しかしながら、コンフリクトは、本当のコンフリクトではない。なぜなら、すでにサーバ122-1に存在する項目の更新バージョンは、第1のユーザが第3のクライアントデバイス180-3をサーバ122-1と同期させる場合に、第1のクライアントデバイス180-1上でなされた変更を含んでいるからである。いくつ

10

20

30

40

50

かの実施形態は、バージョン履歴を用いてこのようにシナリオを検出し、自動的にコンフリクトを解消することができる。

【 0 0 3 6 】

上記のシナリオでは、該項目に関する単一のバージョン番号は、コンフリクトを解消するのに不十分であるかもしれない。例えば、もし第1のユーザが第3のクライアントデバイス180-3からサーバ122-1と同期しなかった場合どうなるだろうか。それどころか、第2のユーザが該項目に対して関係のない変更を行ったと仮定する。この場合、本当にコンフリクトが存在する。単一のバージョン番号は、2つのシナリオを識別するのに不十分である。バージョン履歴技術は、追加の情報を記憶して、この課題を解決するのを助ける。

10

【 0 0 3 7 】

特に、バージョン履歴技術を、項目の複数のバージョン（例えば、バージョンA及びバージョンB）があるか否か、及び1つのバージョンは別のバージョンでなされた変更を含むか否か（例えば、バージョンBは、バージョンAでなされた変更を含むか）を判断するために用いてもよい。この場合、問題の項目のテキスト審査は、難しくて不正確であると判明するかもしれない。例えば、バージョンBはバージョンAからの変更を本当に含んでいるが、バージョンBの変更のうちの1つはバージョンAにおいて加えられたラインを除去することであったシナリオを検討する。

【 0 0 3 8 】

バージョン履歴技術を、著者のリストと、変更がどの順番でなされたかを示すそれらの各々に対するインクリメントする変更番号とを記録するために用いることができる。この場合の「オーサー」は、グローバルユーザ識別子（GUID）で表されるコンピュータである。従って、バージョン履歴特性は、次のように、GUID及び変更番号の連結リストを含むことができる。

20

{ 199A4AEA-A573-40CB-BB3C-7A66C0375104:1, 201B4AEA-A573-40CB-BB3C-7A66C0375104:2, 185D4AEA-A573-40CB-BB3C-7A66C0375104:3 }

【 0 0 3 9 】

特定のコンピュータデバイスが所定の項目を編集するたびに、適当な同期モジュール（例えば、同期モジュール120及び/または182）は、最後の変更番号をとり、その値を1増加して、そのGUIDの次にそれを書く。コンフリクトは、各々のバージョンに対する最も大きい変更セットを比較することによりバージョンAとバージョンBとの間で自動的に解消される。「最も大きい変更セット」は、最も大きい数を持つGUIDと番号との対を含むことができる。バージョンAに対する最も大きい変更セットがバージョンBに含まれ、バージョンBに対する最も大きい変更セットがバージョンAに含まれない場合、バージョンBはより新規であって、バージョンBが勝利者であるようにコンフリクトが解消される。しかしながら、バージョンAに対する最も大きい変更セットがバージョンBに含まれず、バージョンBに対する最も大きい変更セットがバージョンAに含まれない場合、バージョンBは、バージョンAになされた全ての編集を含んでいない。この場合、ユーザは、手作業でコンフリクトを解消することが必要であるかもしれない。

30

【 0 0 4 0 】

以前の例を引き続き述べると、第1のユーザが第1のクライアントデバイス180-1に項目を同期させると仮定する。サーバ同期モジュール120は、データベース124のバージョン履歴を維持することができる。クライアント同期モジュール182も、クライアントデバイス180-1-sのバージョン履歴を維持することができる。この時点で、バージョン履歴は以下のように現れ得る。

40

バージョン履歴：{Server_GUID:1}

次に、第1のユーザは、第1のクライアントデバイス180-1上で項目を編集する。バージョン履歴は、そのとき次の通りに現れ得る。

バージョン履歴：{Server_GUID:1, Work_GUID:2}

第1のユーザは、第2のクライアントデバイス180-2に、さらに、第2のクライア

50

ントデバイス 1 8 0 - 2 から第 3 のクライアントデバイス 1 8 0 - 3 に、項目を同期させる。項目に対する編集が存在しないので、バージョン履歴において対応する変更がない。第 1 のユーザは、第 3 のクライアントデバイス 1 8 0 - 3 上で項目を編集する。バージョン履歴は、以下のように更新される。

バージョン履歴：{Server_GUID:1, Work_GUID:2, Home_GUID:3}

第 1 のユーザは、第 3 のクライアントデバイス 1 8 0 - 3 からサーバ 1 2 2 - 1 に同期化する。

【 0 0 4 1 】

第 1 のユーザは、その翌日に仕事に戻り、第 1 のクライアントデバイス 1 8 0 - 1 をサーバ 1 2 2 - 1 と同期させる。第 1 のクライアントデバイス 1 8 0 - 1 のクライアント同期モジュール 1 8 2 は、以下のようなバージョン履歴を有することができる。

{Server_GUID:1, Work_GUID:2}

サーバ 1 2 2 - 1 のサーバ同期モジュール 1 2 0 は、以下のようなバージョン履歴を有することができる。

{Server_GUID:1, Work_GUID:2, Home_GUID:3}

バージョン A 及びバージョン B に対するバージョン履歴の比較により A における最も大きい変更セット (Work_GUID:2) は、バージョン B に含まれるが、バージョン B に対する最も大きい変更セット (Home_GUID:3) は、バージョン A に含まれないということが明らかになる。従って、サーバ同期モジュール 1 2 0 は、バージョン B はバージョン A になされた全ての編集を含むが、バージョン A はバージョン B になされた全ての変更を含まないということを知っている。バージョン B は、バージョン A の上位集合であって、それゆえに、コンフリクトに勝つ。

【 0 0 4 2 】

第 1 のユーザは、第 3 のクライアントデバイス 1 8 0 - 3 からサーバ 1 2 2 - 1 と同期しなかったが、それどころか、第 2 のユーザは、サーバ 1 2 2 - 1 に存在する項目に対する変更を行ったと仮定する。バージョン履歴を、このシナリオからコンフリクトを検出するために用いることができる。この場合、第 4 のクライアントデバイス 1 8 0 - 4 のクライアント同期モジュール 1 8 2 は、以下のバージョン履歴を有することができる。

{Server_GUID:1, Work_GUID:2}

その一方で、サーバ 1 2 2 - 1 のサーバ同期モジュール 1 2 0 は、以下のバージョン履歴を有することができる。

{Server_GUID:1, User2_Work_GUID:2}

バージョン A 及びバージョン B に対するバージョン履歴の比較で、バージョン A の最も大きい変更セット (Work_GUID:2) はバージョン B に含まれず、バージョン B の最も大きい変更セット (User2_Work_GUID:2) はバージョン A に含まれないということがここで明らかになる。従って、バージョン B はバージョン A の上位集合ではなくかつバージョン A はバージョン B の上位集合ではなく、サーバ同期モジュール 1 2 0 は、解消することを必要とする本当のコンフリクトを検出した。

【 0 0 4 3 】

様々な実施形態は、プロパティバッグを用いてデータスキーマのコンフリクトを検出することができる。いくつかの実施形態は、プロパティバッグを用いて、サーバ 1 2 2 とクライアント 1 8 0 との間のスキーマの不整合をサポートすることができる。プロパティバッグとは、アプリケーションプログラム及び/またはデバイスで用いる様々なデータスキーマに対する一組のフィールド定義を示すことができる。プロパティバッグは、固定されたまたはハードコードされたフィールドのセットよりはむしろ必要に応じてフィールドを定義するために用いられ得る。例えば、クライアントデバイス 1 8 0 によって実行されるマイクロソフトアウトLOOKに対するタスクスキーマは、サーバ 1 2 2 の W S S アプリケーションが利用可能でない特定フィールドを必要とするかもしれないし、逆もまた同様である。プロパティバッグは、アドホックとしてマイクロソフトアウトLOOK及び W S S アプリケーションの両方が用いることができる様々なリストまたはアプリケーションに対す

10

20

30

40

50

るフィールドを定義するために用いられ得る。このように、サーバ122とクライアント180との間のスキーマの不整合を、両方のデバイスで実行される実際のアプリケーションプログラムを必ずしも更新または変更することなく修正することができる。

【0044】

様々な実施形態は、フィールド値マッピングを用いて言語コンフリクトを検出することができる。いくつかの実施形態は、フィールド値マッピングを用いてサーバ122-1とクライアント180-1との間の言語不整合をサポートすることができる。クライアントデバイス180-1及びサーバ122-1が同じ言語でない国際的なシナリオをサポートするために、この特徴は望ましいかもしれない。例えば、マイクロソフトアウトルックは、「状態 (Status)」フィールド及び「優先度 (Priority)」フィールドの値に基づくタスク上で実行する論理を有する。サーバ122-1は、サーバの言語に基づいてこれらのフィールドに異なる値を書き込むことができる。例えば、英語では、状態フィールドは、「開始されていない (Not Started)」、「据え置きされた (Deferred)」、「完了した (Completed)」等の表現を含むことができる。マイクロソフトアウトルックは、マイクロソフトアウトルックが理解する値にこれらの値を確実にマッピングする方法を知ることが必要である。1つの実施形態では、マッピングセクションを、マイクロソフトアウトルックが書き込まれた値を自分で理解する値にマッピングするために用いることができるリストに対するスキーマに加えることができる。

【0045】

様々な実施形態は、1つまたは複数のページング技術を用いて構造化コンテンツタイプを同期させることができる。いくつかの実施形態は、様々なページング技術を用いてサーバ122-1とクライアント180-1との間の非常に大容量のデータセットの同期をサポートすることができる。サーバ122-1及びクライアント180-1に対する性能を向上させるために、ページ要求は、リスト内で変更された一組の項目を得るために用いられてもよい。変更された全ての項目を返送するよりはむしろ、サーバ122-1は最後のQ項目 (例えば、最後の100項目) だけを返送してもよい。クライアント180-1がサーバ122-1とのより遅い通信接続を有する場合、またはサーバ122-1がリストの数千または何百万もの項目を有する場合に、この特徴は特に役立つ。項目の大容量は、例えば、クライアントデバイス180-1のマイクロソフトアウトルックとサーバ122-1との間のアドレスブック構造化コンテンツタイプを同期させる場合、特に、性能のボトルネックを引き起こすかもしれない。

【0046】

所定のページング技術を、あらゆる方法を用いて実行することができる。1つの実施形態では、例えば、サーバ同期モジュール120は、返送用の最大項目数を特定する `GetListChangesSinceToken` ウェブサービス上の「`rowLimit`」と称されるプロパティをサポートすることができる。例えば、「`rowLimit = 100`」は、変更トークンの値以降に更新された最初の100項目を返送する。プロパティが特定されない場合、変更トークン以降に変更された全ての項目を返送することがある。

【0047】

サーバ122-1のWSSアプリケーションが項目制限付きの要求を受信すると、削除動作は項目制限から外されてもよい。パラメータは、リストクエリーに渡されて、`rowLimit` に返送される項目数を制限することができる。この数よりも少ない項目が変わった場合、更新された変更トークンが項目とともに返送される。しかしながら、この数より多い項目が変わった場合、変更トークンは更新されず、最初は通過した同じ変更トークンが返送され得る。`ListItemCollectionPositionNext` に対する値がまた返送されてもよい。`ListItemCollectionPositionNext` に対する値を受信するとすぐ、クライアントデバイス180-1は、変更トークン及びこの値の両方を送信するとともに、サーバ122-1に再クエリーしなければならない。サーバ122-1は、`ListItemCollectionPositionNext` の開始から次の `rowLimit` 項目数を返送するために新しいクエリーを実

行することができる。この「If」ブロックは、次に必要に応じて再実行されてもよい。

【0048】

クライアント180-1が更新を処理する前に必ず削除処理をする必要があるということとは注目に値する。クライアント180-1がすでに削除した項目に対する削除を受信した場合、クライアント180-1はエラーを食い止めて次へ進まなければならない。クライアント180-1がサーバ122-1で削除された項目に対する更新をしようとした場合、クライアント180-1はエラーを食い止めて動作を続けなければならない。

【0049】

様々な実施形態は、内部プライベートネットワークまたは外部公衆網（例えば、公衆インターネット）を介して、サーバ122-1と同期するためにクライアント180-1が利用することができる代替のURLをクライアント180-1に提供することができる。例えば、1つの実施形態では、サーバ同期モジュール120は、内部プライベートネットワークから少なくとも1つの構造化コンテンツタイプを有する同期要求を送信するための第1のネットワークアドレス及び、外部公衆ネットワークから少なくとも1つの構造化コンテンツタイプとともに同期要求を送信するための第2のネットワークアドレスを生成することができる。この第1のアドレス及び第2のアドレスは、例えば、ハイパーリンクを用いて実行されるURLを含むことができる。

【0050】

一例であるが、サーバ同期モジュール120は、同期要求に応答して返送されたデータとともに代替のドメインマッピングを渡すことができる。このことは、ユーザがプライベートネットワーク（例えば、企業イントラネット）内部のリストに進み、マイクロソフトアウトLOOKなどの局所的なアプリケーションにリストを同期させるシナリオを許容する。次に、ユーザは、帰宅し、ハイパーテキスト転送プロトコル（HTTP）を介してリモートプロセスジャコール（RPC）を用いてメールに接続し、それらのシェアポイントリストが同様に同期するのを求める。サーバ122-1（例えば、シェアポイントサイト）がまた、公衆網（例えば、インターネットなどのエクストラネット）で公開される場合、URLに対するドメインが異なる場合であっても、ユーザはリストを同期させることができる。

【0051】

代替のドメインマッピングを、いくつもの方法で実施することができる。1つの実施形態では、例えば、代替のドメインを、以下の順序で返送することができる。

[Intranet], [Default], [Extranet], [Internet], [Custom]

これらのマッピングを、「listitems」タグ上の「AlternateUrls」属性で返送することができる。マッピングが存在しない場合、同じ命令は、以下のように欠落したドメインに対して挿入されるコンマを用いて返送され得る。

(http://intranet, https://default, https://extranet.com, , http://custom)

これらの5つの代替のドメインマッピングは、全て、集中管理で画定され得る。ドメインは、例えば、URLのまさに最初の部分、例えば「http://www.microsoft.com」または「http://msw」、を含むことができる。クライアントアプリケーション（例えば、マイクロソフトアウトLOOK）は、これらのドメインを解析して、クライアントデバイス180-1-sについての状態に基づいてどれを使用するかについて決定する責任を負う。通常、クライアント180-1は、第1のドメインから始めて、順番に残りのドメインを処理する。

【0052】

同期動作を更に改善するために、様々な実施形態は、クライアントデバイスをウェブサイトといつ同期させるかについてのタイミング問題を改善することができる。例えば、いくつかの実施形態は、同期クライアントに望ましい同期間隔（例えば、クライアントはR分毎に同期化する）を提供することによってクライアント負荷を管理することができる。望ましい同期間隔は、最小同期間隔、最大同期間隔、またはいくつかの中間値を表すことができる。1つの実施形態では、例えば、サーバ同期モジュール120は、同期間隔バラ

10

20

30

40

50

メータを生成することができる。サーバ同期モジュール 120 は、クライアント 180-1 に同期間隔パラメータを送信することができる。サーバ 122-1 は、同期間隔パラメータに従ってクライアント 180-1 から同期要求を受信することができる。

【0053】

望ましい同期間隔を用いることにより、いくつかの利点を提供することができる。望ましい同期間隔の使用は、スケールを改善することができ、多数のマイクロソフトアウトルックのクライアント（例えば、10万クライアント）により、シェアポイントサーバファーム（例えば、コンピュータレイ 140）が使用不能になることはない。例えば、マイクロソフトアウトルックが、クライアントデバイス 180-1 に 60 分毎に各々の構造化コンテンツリストを同期させる、デフォルトの同期間隔を有すると仮定する。各々のユーザが最終的には最高 10 の同期されたリストを加えると仮定すると、サーバ 122-1 に対するヒットの総数は、 $10 \times 24 \times 100,000$ / 日に達するかもしれない。それは 1 日 24 時間毎秒ほぼ 278 ヒットということになる。ほぼ 60 分ごとに同期化する RSS (Really Simple Synchronization) 頻度 (feeds) は、この課題を更に悪化させるかもしれない。

【0054】

様々な技術を、所定のサーバ 122-1-n 上の作業負荷を減らすために用いることができる。1つの実施形態では、例えば、キャッシュリスト及び/またはサイトに対する修正された時間を、フロントエンドで実施することができる。最後にキャッシュされたリストまたはサイトが変更された時間以来の変更を求める要求がサーバ 122-1 に達する場合、サーバ 122-1 は変更がなかったということを知って、データベース 124、126 をクエリーすることを完全に回避することができる。キャッシュは、例えば、ほぼ 5 分毎といったある規則的な間隔で有効期限が切れることを必要としてもよい。キャッシュされるリストまたはサイトの数は、例えば、ほぼ 100 のリスト及び/またはサイトなどのメモリ資源を収容することに限定される必要があるかもしれない。

【0055】

所定のサーバ 122-1-n 上の作業負荷を減らす別の技術は、日付/時間適応同期技術を実施することであってもよい。サーバ同期モジュール 120 は、クライアントデバイス 180-1 に同期間隔パラメータを送信することにより、所定のマイクロソフトアウトルッククライアントによってなされた多数の同期要求を減じることができる。同期間隔パラメータは、勤務時間、夜間及び週末などの日付/時間情報を織り込んだリスト変更履歴データを利用している次のリスト変更時間の蓋然性の予測に基づいて変化する。

【0056】

様々な実施形態において、望ましい同期間隔は、サーバに記憶されたコンテンツ変更履歴データに基づいていてもよい。1つの実施形態では、例えば、サーバ同期モジュール 120 は、サーバ 122-1 によって記憶されたコンテンツ変更履歴データに基づいてクライアント 180-1 に対する同期間隔パラメータを生成することができる。サーバ同期モジュール 120 は、クライアント 180-1 に同期間隔パラメータを送信することができる。

【0057】

コンテンツ変更履歴データに基づく技術の 1つの例は、格子ベースの適応同期（「格子同期」）と称されてもよい。シェアポイントサーバ上の各々のリストは異なる使用特徴を有する。多くの要因は、リスト使用及び更新頻度に影響を及ぼすかもしれない。かかる要因の例は、リストタイプ、サイトタイプ、リストのサイズ、リストカスタマイズの量、サイトユーザ数、地理的位置及びユーザ分布、サイト訪問の総数、時間帯、サイトのプロジェクトの周期におけるそのときの位置（例えば、開始、中間、終了、またはアーカイブ）などを含むことができる。多くのこれらの要因は、構成ベースであり、利用パターンについての予測を、このデータに基づいて行うことができる。しかしながら、多くのこれらの要因はまた、時間ベースである。時間ベースの要因は、利用パターンに重大な影響を及ぼすかもしれない。サイトは、通常、実際は周期的であり、ウェブサイトが永久に同じ割合

10

20

30

40

50

で用いられることはめったにない。更に、人々はスケジュール通りに働く傾向がある。従って、各時間枠が8時間から成っている場合であっても、午後10時から午前6時までに発生する仕事量は、午前9時から午後5時までの間に発生する量より通常少ない。人々はまた、週末に働きたがらない。仕事の性質及び仕事の地理的分布がこれらの要因に影響するかもしれないということは注目するに値する。従って、前もってそれらをハードコードしようとするよりむしろ使用パターンに適應させることが重要であるかもしれない。サーバへの要求数を減らすのに最も劇的な影響を及ぼし、なおユーザの期待を満足させる提案は、時間ベースのアルゴリズムを含む必要があり、人々が働きかつそれによって所定のサイトを用いる方法に敏感である必要がある。

【0058】

このような観点から、1つの設計的な考慮として、クライアントデバイス180-1に、クライアントデバイス180-1が次に同期動作を実行しなければならない望ましい時間を提供することが考えられる。クライアントデバイス180-1またはサーバ122-1のいずれかで変更がある場合、クライアントデバイス180-1はサーバ122-1と同期するだけでよい。従って、確率分析法を用いて、理想的な同期間隔を決定することができる。管理者は、クライアントデバイス180-1-sが再同期化するのを望む蓋然性閾値を調整することによって理にかなった方法でネットワーク100の動作を調整することができる。例えば、コンテンツの10%が更新される可能性がある場合に、サーバ管理者は、蓋然性閾値を調整してクライアントデバイス180-1に同期させるようになすことができる。そのことがあまりに多くのサーバ負荷をもたらす場合、管理者は50%の蓋然性閾値に閾値を調整することができる。ネットワーク性能が許容可能なパラメータ範囲内になるまで、管理者はこの処理・調整を細かく行ってもよい。

【0059】

場合によっては、将来のリスト利用のための蓋然性閾値は、使用履歴データに基づいていてもよい。例えば、リストは、データの集合または、1つまたは複数の構造化コンテンツタイプからなる項目を含んでいてもよい。使用は、しばしばパターンに従う。人々は生来習慣の奴隷であり、構造化作業環境は、それが予測可能性を生じさせるように、このことをさらに促進する。過去の使用を表すために、時間を格子に分割することができる。一週間の期間を仮定すると、y軸は当該週の曜日であり、一方、x軸は日毎の時間を示すことができる。格子の各セルにおいて、ブール値を、その期間にリストが更新されたか否かについて指し示すよう設定することができる。ここで、値「1」は、リストがこの期間に修正されたことを表し、値「0」はリストがこの期間に修正されなかったことを表す。例えば、月曜日の列で午前9:30 - 午前9:40のセルに注目すると、セルの値が1である場合、リストは月曜日の午前9:30 - 午前9:40の間に修正された。次の「1」は、月曜日の午後3:10 - 午後3:20までないかもしれない。先に進むと、1週間のサイト使用を監視することは、完成した格子をもたらす。リストがより頻繁に更新されるほど、格子はより多くの1値を有し、リストがそれほど頻繁に更新されないと、より多くの0値を有する。クライアントデバイス180-1が月曜日の午前9:05にサーバ122-1と同期すると仮定する。何かが変更されたか否かを見るクエリーを実行した後に、サーバ122-1は、サーバ122-1がその次の同期イベントを実行しなければならない時間に関してクライアントデバイス180-1に対するリコメンデーションを行うことをそのとき必要とする。履歴データを分析すると、更新が月曜日の午前9:30 - 午前9:40に生じたように見える。この履歴データに基づいて、サーバ122-1は、クライアントデバイス180-1に午前9:30 - 午前9:40の間にその次の同期を実行するように指示することができる。アルゴリズムで、値1に達するまで、格子を行き来することによってこのことを達成することができる。通過した各セルに対して、セル時間サイズを、同期同士の間の望ましい時間に加えることができる。この例では、9:00 - 9:10のセル、9:10 - 9:20のセル、及び9:20 - 9:30のセルを通過した。従って、サーバ122-1は、30分という望ましい同期間隔を返送することができる。

【0060】

この点において、午前 9 : 30 - 午前 9 時 40 分のリコメンデーションは、開始点に過ぎない。より長い期間が、リコメンデーションをより細かくする。前の例を続けると、リストが第 2 週の午前 9 : 22 に更新されたと仮定する。第 3 週の間に、クライアントデバイス 180-1 は、午前 9 : 05 にサーバ 122-1 との同期を再度実行し、サーバ 122-1 は、更新されたリコメンデーションを行うことを必要とする。格子を行き来すると、午前 9 : 20 - 9 : 30 のセルに値 1、さらに午前 9 : 30 - 9 : 40 のセルに値 1 がある。格子アルゴリズムは、そのときのセルから始まって、値 1 を有するセルに達するまで、各々のセルの通過で 10 分を加える。午前 9 : 00 - 9 : 10 のセルを通過し、さらに午前 9 : 10 - 9 : 20 のセルを通過する。サーバ 122-1 は、ここでクライアント 180-1 に 20 分という値を返送することができる。しかしながら、履歴証拠に基づいて、リストが 20 分で更新されるか否かを示す蓋然性が生成されてもよい。午前 9 : 05 の開始時間から午前 9 : 30 の完了時間まで 2 つの可能な月曜日において、リストは 1 つの月曜日にだけ更新された。この履歴データに基づいて、蓋然性（確率）を $1/2$ 、即ち 50 % と算出することができる。10 分を望ましい同期間隔に加えることができ、蓋然性分析を再び実行することができる。午前 9 : 05 の開始時間から午前 9 : 40 の完了時間までデータの 2 つの月曜日については、リストは 2 回更新された。従って、30 分の蓋然性は、 $1/2 + 1/2 = 2/2$ 、即ち 100 % である。より多くのデータを用いることにより、精密なリコメンデーションをもたらす。それは、サーバ管理者が、サーバ 122-1-n から情報を要求しているクライアントデバイス 180-1-s にどのくらいの頻度で満足しているか、に対して、ユーザが直ちに最新のデータを有することがどれくらい重要か、についてインフォームドデシジョン（informed decision）になる。例えば、サーバ管理者が 50 % 以下で閾値を設定する場合、サーバ 122-1 は 20 分の同期間隔を返送してもよい。しかしながら、閾値が 50 - 100 % の間で設定された場合、サーバは 30 分の同期間隔を返送してもよい。

【0061】

同期間隔を生成する一般化されたアルゴリズムは、以下のように示すことができる。

P は蓋然性を示し

T はユーザ定義済み閾値を示し

S は各セルのタイムスパンを示し、上記の例では、これは 10 分である

C はそのときのセルカウントを示し

G [] は格子であり、G [0] は現在時間における値を表すと仮定する

M は格子のセルについての最大値であり、上記の例では、これは週の数である

P = 0 ;

For (C = 0 ; P < T ; C + +)

P = P + G [C] / M ;

return C * S

戻り値が各々のクライアント要求によって再計算される必要がない点は注目に値する。むしろ、戻り値は一時期キャッシュに格納され、次の間隔が始まるまで有効であるとみなされる。上記の例では、それは最高 10 分である。

【0062】

実際の格子を構築するために、項目が修正されるたびに、リストの最終修正時間がチェックされる。リストの最終修正時間が現在の時間と同じタイムスパンで起こる場合、変更は書き込まれる必要はない。例えば、午前 9 : 00 から午前 9 : 10 までの間の同じ期間に起こる 2 つの変更は、1 つの更新を必要とするだけである。このことは、ロックコンテンション及びスケールに役立ち、書き込み数を最小にしておく。リストの最終修正時間が現在の時間以外の異なるタイムスパンで起こった場合、現在のタイムスパンのセルの値を、「1」増加することができる。場合によっては、WSS バージョン 3.0 がすでにこれを処理しているにもかかわらず、リストに対する修正時間は同様に更新されることを必要とする。

【0063】

格子アルゴリズムは、どれくらいのデータを記憶することができるか、さらに、どんな傾向が特定のアプリケーションの利用において共通か、に基づいて多くの側面で拡張可能である。x 方向において、各セルのタイムスパンは、1 分に減じるかまたは 1 時間以上まで伸ばしてもよい。y 次元において、格子は拡張されて一月の各日に対して異なった列を含んでいてもよい。または、一日を表している 1 つの列だけに格子を折り畳んでもよい。実際、2 日を有することにより少し知的な方法でそれを折り畳むことができ、そこで、一方は週日を表し、他方は週末日を表す。格子が 1 2 時間のシフト勤務を表している場合、格子は、各々が開始から終了までの 1 2 時間シフトを表している y 列を有することができる。

【 0 0 6 4 】

前に述べたように、サイト使用は、通常、予測可能な周期に付随する。これらの周期的期間は、何ヶ月かにわたってしばしばゆっくり変化する。格子アルゴリズムを洗練させるために、最近のデータは、古いデータよりも良いインジケータであるかもしれない。例えば、通常、6 ヶ月前の月曜日に起こったリスト更新は、昨日起こったものより重要でなくなる。従って、より大きいデータの集合を維持し、時間に基づいて異なったより大きい集合の範囲内でデータに重みをかける機能を適用することによって、格子アルゴリズムを洗練することができる。あるいは、メモリ資源は、旧データを期限切れにすることによってより効果的に用いられ得る。例えば、格子が 8 週間毎にリセットされると仮定する。上記した前の例で定義された格子では、各セルの最大値は 8 になるだろう。従って、各セルを、ほんの 3 つのビットで表すことができる。

【 0 0 6 5 】

しかしながら、格子をリセットすることは、また、格子アルゴリズムの結果に影響を及ぼすことができるデータも除去する。この問題は、期間までに互いにオフセットされる 2 つの格子を生成することにより軽減することができる。例えば、第 1 の格子は、週 1 においてデータの記録を開始し、一方、第 2 の格子は、週 4 においてデータの記録を開始することができる。この点で、サーバ同期モジュール 1 2 0 は、蓋然性の算出のために第 1 の格子を用いることができる。週 8 において、第 1 の格子をリセットすることができ、サーバ同期モジュール 1 2 0 は、第 2 の格子を用いるために切り替えて、蓋然性の算出を実行することができる。サーバ同期モジュール 1 2 0 は、再び第 1 の格子で記録を開始することができる。このように、サーバ同期モジュール 1 2 0 は、1 つの格子から次まで絶えず循環することができ、決してゼロから開始する必要はない。この場合、データフローの不連続は、4 週間隔で起こる。円滑さの重要性により、このことはより多くの格子を追加することにより軽減することができる。例えば、8 つの格子は、各々 1 週のより軽微な不連続性をもたらすかもしれない。複数の格子を用いることにより最近のイベントにより多くの力を与えて、蓋然性を生成するためにこのシステムをまた用いることができる、という点に注目する。

【 0 0 6 6 】

格子アルゴリズムの性能を更に改善するために、最大同期間隔及び最小同期間隔が設定されてもよい。最小同期間隔は恐らくセル幅のサイズにデフォルトで設定され、それは、上記の例では 1 0 分である。最大間隔のサイズは、システムの信頼レベルによる。いずれも、閾値とともに管理者によって変更可能でなければならない。これらの値は、使用の間、システムを洗練するように調整されることができるが、それがすでに使用中だった後で、格子構造（例えば、x 軸及び y 軸上の時間の表現）を変更するために邪魔となるかもしれない。格子が最小限の使用可能状態に投入されるまで、デフォルト時間は最初の同期に対して設定されなければならない。

【 0 0 6 7 】

この方法を、RSS などの他の同期プロトコルに用いることができる。それはまた、コンテンツをキャッシュに格納する検索クローラー、インデクサー、及びプロキシサーバなどの領域に適用できる。格子アルゴリズムはクライアントで実施可能だが、クライアントはサーバでなされる全ての変更を知らないので、精度はたぶん低下するだろう。格子アル

10

20

30

40

50

ゴリズムはまた、クライアント クライアントの同期において用いられてもよい。それはまた、同期方法に基づく通知に関して用いられてもよく、そこで、通知はめったにコンテンツの変更を処理しないし、このシステムはより頻繁にコンテンツの変更を処理する。

【 0 0 6 8 】

格子方法は、所定の実施により、メモリ資源に様々な影響を及ぼすことができる。データ記憶の予測を、以下のように行うことができる。

2 0 分の幅

7 つの区別できる日

テーブル毎に 8 週間

2 つのテーブル

= 3 7 8 バイト

1 0 分の幅

2 列 (週日 / 週末)

テーブル毎に 6 週間

2 つのテーブル

= 2 8 8 バイト

10

【 0 0 6 9 】

更に間隔が履歴データに基づいて望ましい同期間隔を生成するために、望ましい同期間隔はまた、クライアントからの前の同期要求の結果に基づいていてもよい。1 つの実施形態では、例えば、サーバ同期モジュール 1 2 0 は、クライアント 1 8 0 - 1 による前の同期要求からの前の同期結果に基づいてクライアントに対する同期間隔パラメータを生成することができる。サーバ同期モジュール 1 2 0 は、クライアント 1 8 0 - 1 に同期間隔パラメータを送信することができる。

20

【 0 0 7 0 】

一例として、サーバ同期モジュール 1 2 0 は、機能ベースの適応同期を実行して、同期間隔パラメータを生成することができる。所定のマイクロソフトアウトルッククライアントによって生成される同期要求の数は、何らかの形式の指数関数的減衰 / 指数関数的成長の機能を用いて生成され得る。例えば、最後の同期間隔をとる。変更が起こった場合、指数関数により時間間隔を減少させる。しかしながら、変更が起こらなかった場合、指数関数機能により時間間隔を増やす。

30

【 0 0 7 1 】

他の技術も、同期間隔パラメータを生成するために用いることができる。例えば、最後の修正の適応同期技術を用いることができる。この技術は、リストに対して最後に修正された時間に基づいて厳密に変化する同期間隔をクライアント 1 8 0 - 1 に送信することにより、マイクロソフトアウトルッククライアントによってなされる同期要求の数を減らす。例えば、リストが最後の 1 時間以内に修正された場合、2 0 分で再び同期する。リストが 1 時間以上前だが 1 日未満前に修正された場合、2 時間で再び同期する。リストが 1 日以上前に修正された場合、一日で再び同期する。別の例では、複数のリスト技術に対する単一のクエリーも用いることができる。この技術は、同時に重要な全てのリスト G U I D を渡す 1 つの要求を生成することをクライアント 1 8 0 - 1 に許容することによってマイクロソフトアウトルッククライアントによってなされる同期要求の数を減らす。最後の修正時間は全てのこれらのリストに対して点検されてパッチ処理されて、サーバ同期モジュール 1 2 0 は、どのリストに明確に同期させるべきかについてクライアントデバイス 1 8 0 - 1 に知らせることができる。また別の例では、電子メール通知ベースの同期技術を用いることができる。現在の同期技術は様々なポーリング技術を用いることができ、クライアントは定期的にまたは半定期的にサーバをポーリングする。1 つの実施形態では、「隠し電子メール警報」を送信することによって同期イベントが必要であるということ、サーバ 1 2 2 - 1 はクライアント 1 8 0 - 1 に通知することができる。このことを、既存の警報インフラに利用することができる。警報はユーザによって必ずしも見られるというわけではなく、それは、ユーザが同期動作を実行することを選択する時にシェアポイントにおい

40

50

て自動的に生成される。このことは、頻度の低いリスト変更に対して特に効果的であり得る。しかしながら、クライアントデバイス 180-1 は、この特定の技術を実施するために着信してくるメール要求の受信をサポートすることを必要とする。

【0072】

多数の具体的な詳細は、実施形態の完全な理解を提供するために本明細書において説明された。しかしながら、実施形態はこれらの具体的な詳細なしに実施されてもよいということが当業者に理解されるだろう。他の例では、周知の動作、コンポーネント、及び回路を、実施形態を不明瞭にしないために詳述しなかった。本明細書において開示される特定の構造上の詳細及び機能の詳細は典型例であって、必ずしも実施形態の範囲を限定するというわけではないということが当業者に認められるだろう。

10

【0073】

「特定の実施形態」または「1つの実施形態」に対する何らかの言及は、実施形態と関連して開示された特定の特性、構造または特徴が少なくとも1つの実施形態に含まれるということの意味するということは注目に値する。明細書の様々な場所における「1つの実施形態では」というフレーズの出現は、必ずしも全て同じ実施形態に言及しているというわけではない。

【0074】

いくつかの実施形態は、それらの派生物とともに「結合される (coupled)」及び「接続される (connected)」という表現を用いて開示され得る。これらの用語は互いに同義語ではないということが理解されなければならない。例えば、いくつかの実施形態は、用語「接続される」を用いて説明されて、2つ以上の要素が互いに直接物理的または電氣的に接触していることを示す。別の例では、いくつかの実施形態は、用語「結合される」を用いて説明されて、2つ以上の要素が直接物理的または電氣的に接触していることを示す。しかしながら、用語「結合される」は、また、2つ以上の要素が互いに直接接触していないが、なお互いに協力または相互作用するということの意味することができる。実施形態は、上記の説明に限定されない。

20

【0075】

例えば、いくつかの実施形態は、マシンによって実行される場合に、実施形態に従って方法及び/または動作をマシンに実行させることができる命令または命令セットを記憶することができる、機械可読媒体または物品 (article) を用いて実施されてもよい。例えば、かかるマシンは、任意の適当な処理プラットフォーム、計算プラットフォーム、コンピュータデバイス、コンピュータデバイス、コンピュータシステム、処理システム、コンピュータ、プロセッサ等を含んでもよく、ハードウェア及び/またはソフトウェアの任意の適当な組合せを用いて実施することができる。マシン可読媒体または物品は、例えば、任意の適当なタイプのメモリユニット、メモリデバイス、メモリデバイス、メモリ物品、メモリ媒体、記憶デバイス、記憶物品、記憶媒体及び/または、例えばメモリ、着脱自在または固定媒体、消去可能または消去不能媒体、書き込み可能または再書き込み可能媒体、デジタルまたはアナログ媒体、ハードディスク、フロッピー (登録商標) ディスク、CD-ROM (読み出し専用コンパクトディスク)、CD-R (書き込み可能コンパクトディスク)、CD-RW (書き換え可能コンパクトディスク)、光ディスク、磁気媒体、光磁気媒体、着脱自在メモ리카ードまたはディスク、様々なタイプのデジタル多用途ディスク、テープ、カセットなどの記憶デバイスを含むことができる。

30

40

【0076】

本発明は構造上の特徴及び/または方法論的動作に特有の言語で説明されたが、添付の請求の範囲において画定された本発明が上述の特定の特徴または動作に必ずしも限定されるというわけではないということが理解されるべきである。むしろ、上述の特定の特徴及び動作は、請求項を実施する例の形で開示されている。

【図面の簡単な説明】

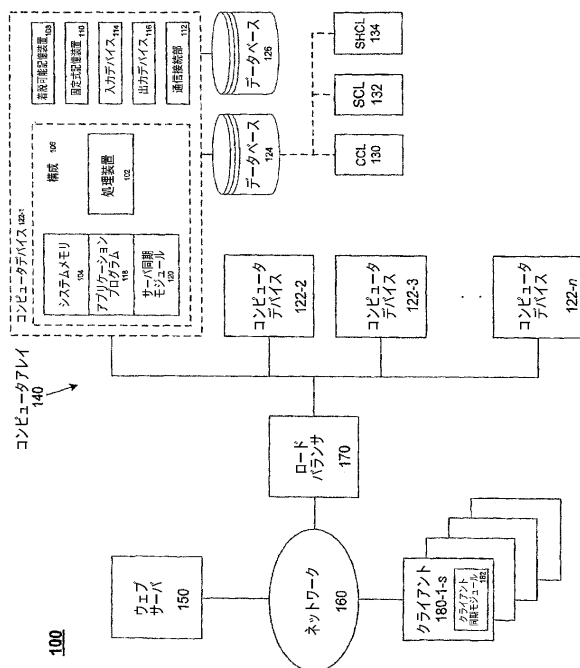
【0077】

【図1】ネットワークの例示的な実施形態を示す図である。

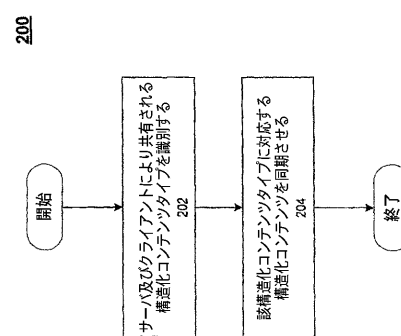
50

- 【図 2】論理フローの例示的な実施形態を示す図である。
 【図 3】第 1 のメッセージフローの例示的な実施形態を示す図である。
 【図 4】第 2 のメッセージフローの例示的な実施形態を示す図である。

【図 1】

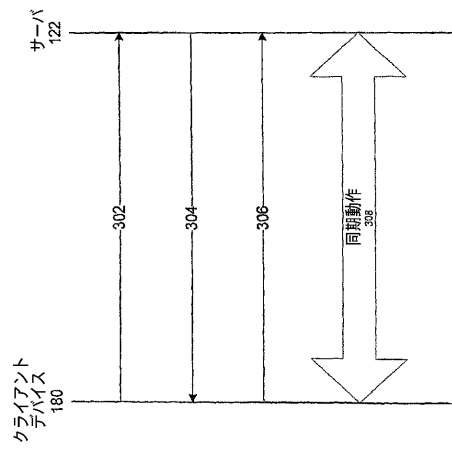


【図 2】



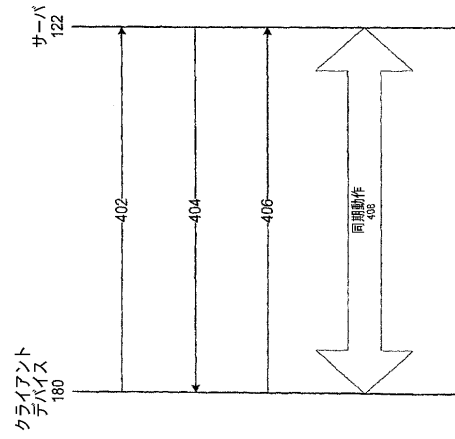
【図 3】

300



【図 4】

400



フロントページの続き

(72)発明者 ホルヘ フェレイラ
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 インターナショナル パテント内

審査官 田川 泰宏

(56)参考文献 ドッジ マーク, Microsoft Office Excel 2003 オフィシャルマ
ニュアル, 日経BPソフトプレス 松崎 稔, 2004年 7月12日, 第1版, p.725-748

(58)調査した分野(Int.Cl., DB名)

G06F 12/00

G06F 13/00