

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5058495号
(P5058495)

(45) 発行日 平成24年10月24日(2012.10.24)

(24) 登録日 平成24年8月10日(2012.8.10)

(51) Int.Cl.

G06F 12/00 (2006.01)

F 1

G06F 12/00 533J

請求項の数 15 (全 16 頁)

(21) 出願番号 特願2006-45851 (P2006-45851)
 (22) 出願日 平成18年2月22日 (2006.2.22)
 (65) 公開番号 特開2006-236350 (P2006-236350A)
 (43) 公開日 平成18年9月7日 (2006.9.7)
 審査請求日 平成21年2月20日 (2009.2.20)
 (31) 優先権主張番号 11/063,381
 (32) 優先日 平成17年2月22日 (2005.2.22)
 (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 バラン セス ラマン
 アメリカ合衆国 98052 ワシントン
 州 レッドモンド ワン マイクロソフト
 ウェイ マイクロソフト コーポレーション内

最終頁に続く

(54) 【発明の名称】ゴースト化による同期

(57) 【特許請求の範囲】

【請求項 1】

第1のシステム上の第1のファイルストアにファイルを記憶するステップ、
 前記ファイルに関するメタデータを前記第1のシステム上の第1のアイテムストア中に作成するステップであって、前記メタデータは前記ファイルの内容から導出され、ファイル支援アイテムに含まれるステップ、

第2のファイルストアと第2のアイテムストアとを有する第2のシステムを前記第1のシステムに接続し、接続されている間に前記メタデータに関する前記ファイルを同期させることなく前記第1のアイテムストアを前記第2のアイテムストアと同期させるステップであって、同期後の前記第2のアイテムストアが前記ファイルに関するメタデータのコピーを有するステップ、

前記第2のシステムを前記第1のシステムから切断し、切断されている間に、前記メタデータのコピーを編集するステップ、および、

前記第2のシステムを前記第1のシステムに再接続し、接続されている間に、前記ファイルを同期させることなく前記第1のアイテムストアを前記第2のアイテムストアと同期させるステップであって、同期後の前記第1のアイテムストアが前記メタデータのコピーに対する編集を有するステップ、

前記第1のアイテムストア中の前記編集を有するメタデータにしたがって、当該メタデータに対応する前記第1のファイルストア中のファイルの内容に前記編集を反映するステップ

10

20

を備えたことを特徴とする方法。

【請求項 2】

前記第1のアイテムストアを前記第2のアイテムストアと同期させる前記ステップは、
第1のバージョンのアイテムを前記第1のシステム上で見つけ、前記第1のバージョン
のアイテムに対応する第2のバージョンのアイテムを前記第2のシステム上で見つけるス
テップ、

前記第1のバージョンのアイテムと前記第2のバージョンのアイテムのいずれかを保持
されるバージョンと識別するステップ、および、

前記第1のアイテムストアと前記第2のアイテムストアの両方が前記保持されるバージ
ョンのアイテムを含むように、前記保持されるバージョンのアイテムをコピーするス
テップ

10

を備えたことを特徴とする請求項1に記載の方法。

【請求項 3】

前記識別するステップは、

更新インジケータを調べて、前記第1のバージョンのアイテムと前記第2のバージョン
のアイテムの一方が前回の同期の後で変化したと判定するステップを備えたことを特徴と
する請求項2に記載の方法。

【請求項 4】

前記ファイルのファイルタイプを決定するステップ、および、

前記ファイルタイプに基づいて所定のアイテムのメタデータ値を決定するステップ
を備えた方法によって、メタデータが導出されることを特徴とする請求項1に記載の方法
。

20

【請求項 5】

前記ファイルのファイルタイプを決定するステップ、および、

前記ファイルに含まれるデータの少なくとも一部を要約してメタデータ値を決定するス
テップ

を備えた方法によって、メタデータが導出されることを特徴とする請求項1に記載の方法
。

【請求項 6】

前記ファイル支援アイテムに関連する前記ファイルに対する変更を識別するステップ、
および、

前記ファイル支援アイテム中の前記メタデータに前記変更を反映するステップ
をさらに備えたことを特徴とする請求項1に記載の方法。

30

【請求項 7】

前記ファイル支援アイテム中の前記メタデータに対する変更を識別するステップ、およ
び、

前記ファイル支援アイテムに関連する前記ファイルに前記メタデータに対する変更を反
映するステップ

をさらに備えたことを特徴とする請求項6に記載の方法。

【請求項 8】

第1および第2のシステムを含むシステムを、

前記第1のシステム上の第1のファイルストアにファイルを記憶する手段、

前記ファイルの内容から前記メタデータを導出し、ファイル支援アイテムに含めること
で、前記ファイルに関係するメタデータを前記第1のシステム上のアイテムストア中に作
成する手段、

第2のファイルストアと第2のアイテムストアとを有する前記第2のシステムが前記第
1のシステムに接続されている間に、同期後の前記第2のアイテムストアが前記ファイル
に関係するメタデータのコピーを有するように、前記メタデータに関係する前記ファイル
を同期させることなく前記第1のアイテムストアを前記第2のアイテムストアと同期させ
る手段、

40

50

前記第2のシステムが前記第1のシステムから切断されている間に、前記メタデータのコピーを編集する手段を提供する手段、。

前記第2のシステムが前記第1のシステムに再接続されている間に、再同期後の前記アイテムストアが前記メタデータのコピーに対する編集を有するように、前記ファイルを同期させることなく前記第1のアイテムストアを前記第2のアイテムストアと再同期させる手段、および

前記第1のアイテムストア中の前記編集を有するメタデータにしたがって、当該メタデータに対応する前記第1のファイルストア中のファイルの内容に前記編集を反映する手段として機能させるプログラム。

【請求項9】

10

ファイルを有する第1のファイルストア、

前記ファイルに関するメタデータであって前記ファイルの内容から導出されたメタデータを含むファイル支援アイテムを有する第1のアイテムストア、

第2のファイルストアと第2のアイテムストアとを有する第2のシステムとの接続および前記第2のシステムからの切断に適した接続手段と、

前記第2のシステムと接続されている間に、同期後の前記第2のアイテムストアが前記ファイルに関するメタデータのコピーを有するように、前記メタデータに関する前記ファイルを同期させることなく前記第1のアイテムストアを前記第2のアイテムストアと同期させる手段、。

前記第2のシステムから切断されている間に、前記メタデータのコピーを編集する手段を提供する手段、。

20

前記第2のシステムと再接続されている間に、再同期後の前記アイテムストアが前記メタデータのコピーに対する編集を有するように、前記ファイルを同期させることなく前記第1のアイテムストアを前記第2のアイテムストアと再同期させる手段、および

前記第1のアイテムストア中の前記編集を有するメタデータにしたがって、当該メタデータに対応する前記第1のファイルストア中のファイルの内容に前記編集を反映する手段を備えたことを特徴とするシステム。

【請求項10】

前記第1のアイテムストアを前記第2のアイテムストアと同期させる前記手段は、

第1のバージョンのアイテムを前記第1のシステム上で見つけ、前記第1のバージョンのアイテムに対応する第2のバージョンのアイテムを前記第2のシステム上で見つける手段、。

30

前記第1のバージョンのアイテムと前記第2のバージョンのアイテムのいずれかを保持されるバージョンと識別する手段、および、

前記第1のアイテムストアと前記第2のアイテムストアの両方が前記保持されるバージョンのアイテムを含むように、前記保持されるバージョンのアイテムをコピーする手段を備えたことを特徴とする請求項9に記載のシステム。

【請求項11】

前記識別する手段は、

更新インジケータを調べて、前記第1のバージョンのアイテムと前記第2のバージョンのアイテムの一方が前回の同期の後で変化したと判定する手段を備えたことを特徴とする請求項10に記載のシステム。

40

【請求項12】

前記ファイルのファイルタイプを決定する手段、および、

前記ファイルタイプに基づいて所定のアイテムのメタデータ値を決定する手段を備え、これによりメタデータが導出されることを特徴とする請求項9に記載のシステム。。

【請求項13】

前記ファイルのファイルタイプを決定する手段、および、

前記ファイルに含まれたデータの少なくとも一部を要約することによりメタデータ値を

50

決定する手段

を備え、これによりメタデータが導出されることを特徴とする請求項 9 に記載のシステム

。

【請求項 1 4】

前記接続手段は、

前記ファイル支援アイテムに関連する前記ファイルに対する変更を識別し、

前記ファイル支援アイテム中の前記メタデータに前記変更を反映する

ようにさらに構成されていることを特徴とする請求項 9 に記載のシステム。

【請求項 1 5】

前記ファイル支援アイテム中の前記メタデータに対する変更を識別する手段、および、10

前記ファイル支援アイテムに関連する前記ファイルに前記メタデータに対する変更を反映する手段

をさらに備えたことを特徴とする請求項 1 4 に記載のシステム。

【発明の詳細な説明】**【技術分野】****【0 0 0 1】**

本発明は一般にコンピュータデータベースに関し、より詳細にはデータベースの同期に関する。

【背景技術】**【0 0 0 2】**

同期されたデータベースは、別々に使用できるが相互に対して時々更新できるデータベースである。例えば、ポータブルコンピュータ上のファイルシステムが、オフィスでサーバのファイルシステムと接続される場合がある。ポータブルコンピュータがオフィス外で使用されるとき、ユーザが作業を継続できるように所定のデータをポータブルコンピュータに記憶することができる。一部のファイルは、ポータブルコンピュータ上で変更および更新されることがあり、サーバ上でも同様である。

【0 0 0 3】

ポータブルコンピュータが再びオフィスに持ち込まれたとき、ポータブルコンピュータ上とサーバ上のファイル構造を同期させることができる。ポータブルコンピュータからの更新済みファイルをサーバに転送することができ、同様にサーバからの更新済みファイルをポータブルコンピュータに転送することができる。ファイル構造が非常に多くのデータを含むときは、同期するのに非常に長い時間かかることがあり、ポータブルコンピュータ上とサーバコンピュータ上の両方で多量のストレージ空間を必要とすることがある。

【発明の開示】**【発明が解決しようとする課題】****【0 0 0 4】**

したがって、長時間かかり空間を消費する同期を必要とせずに、接続状況と非接続状況で有用な作業をデータベースに対して実施することのできる、システムおよび方法を提供することが有効である。

【課題を解決するための手段】**【0 0 0 5】**

本発明は、メタデータと生データとを有するデータベースを同期させるためのシステムおよび方法を提供する。メタデータは、2つのシステム間で完全に同期させることができ、両方のシステムにコピーが記憶される。生データは選択的に同期させることができ、したがって、クライアントシステムはすべての生データをローカルに記憶する必要はない。切断時、クライアントシステムは、ローカルに記憶すべき生データの所定部分を識別することができ、このようなデータは、サーバシステムに再接続されたときに同期させることができる。メタデータは、生データに関連する様々なアイテムを含むことができる。したがって、これらのメタデータは、生データがないときでも、様々な有用な機能をクライアントシステムに提供することができる。生データを操作する、または、生データの変更を40

50

生じさせる一部の機能を含むメタデータの操作は、「オフライン」で、生データなしで実施することができる。後続の再同期イベントが発生したとき、これらの変更を生データに反映することができる。

【発明を実施するための最良の形態】

【0006】

本発明は様々な修正および代替形態が可能であるが、本発明の特定の実施形態を図面に例示し、本明細書で詳細に説明する。しかし、本発明を開示される特定の実施形態に限定する意図はなく、反対に、本発明は、特許請求の範囲によって定義される本発明の趣旨および範囲内に含まれるすべての変更形態、均等形態、および代替形態に及ぶことを理解されたい。一般に、これらの実施形態は、本発明の具体的な発明的態様または特徴を強調するためを選択されたものである。10

【0007】

本明細書全体を通じて、同一の参照番号は、図の記述全体を通して同一の要素を表す。

【0008】

要素が「接続」または「結合」されるものとして言及されるとき、これらの要素は、相互と直接に接続または結合されてもよく、あるいは1つまたは複数の介在要素があってもよい。

【0009】

本発明は、デバイス、システム、方法、および／またはコンピュータプログラム製品として具体化することができる。したがって、本発明の一部またはすべては、ハードウェアおよび／またはソフトウェア（ファームウェア、常駐ソフトウェア、マイクロコード、ストートマシン、ゲートアレイなどを含む）に組み入れができる。さらに、本発明は、コンピュータによって使用可能な、またはコンピュータ可読の記憶媒体上のコンピュータプログラム製品の形態をとることができ、この媒体には、命令実行システムによって、または命令実行システムと共に使用される、コンピュータによって使用可能な、またはコンピュータ可読のプログラムコードが組み入れられる。本明細書のコンテキストでは、コンピュータによって使用可能な、またはコンピュータ可読の媒体は、命令実行システム、装置、またはデバイスによって、あるいはそれらと共に使用されるプログラムを、収録する、記憶する、通信する、伝搬する、または搬送することのできる、任意の媒体とすることができます。20

【0010】

コンピュータによって使用可能な、またはコンピュータ可読の媒体は、例えば、電子、磁気、光学、電磁、赤外線、または半導体の、システム、装置、デバイス、または伝搬媒体とすることができますが、これらに限定されるものではない。コンピュータ可読媒体には、例として、コンピュータ記憶媒体および通信媒体を含めることができるが、これらに限定されるものではない。

【0011】

コンピュータ記憶媒体には、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、またはその他のデータなどの情報を記憶するための任意の方法または技術で実現された、揮発性および不揮発性、着脱可能および着脱不能の媒体が含まれる。コンピュータ記憶媒体には、RAM、ROM、EEPROM、フラッシュメモリ、または他のメモリ技術、CD-ROM、デジタル多用途ディスク（DVD）、または他の光記憶、磁気カセット、磁気テープ、磁気ディスク記憶、または他の磁気記憶装置、あるいは、所望の情報を記憶するのに使用でき命令実行システムからアクセスできるその他の任意の媒体が含まれるが、これらに限定されるものではない。プログラムは、例えば紙媒体または他の媒体を光学的に走査することによって電子的に取り込んでから、必要ならば適切な方式でコンパイル、解釈、または他の方法で処理し、次いでコンピュータメモリに記憶することができるので、コンピュータによって使用可能な、またはコンピュータ可読の媒体は、プログラムが印刷された紙媒体または別の適切な媒体であってもよいことに留意されたい。40

【0012】

10

30

30

40

50

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、またはその他のデータを、搬送波やその他の移送機構などの被変調データ信号に組み入れるものであり、通信媒体には任意の情報伝達媒体が含まれる。用語「被変調データ信号」は、信号中の情報が符号化される形で1つまたは複数の特性が設定または変更される信号を意味する。例えば、通信媒体には、有線ネットワークや直接有線接続などの有線媒体と、音響、RF、赤外線などの無線媒体およびその他の無線媒体とが含まれるが、これらに限定されるものではない。上記のいずれかの組合せもコンピュータ可読媒体の範囲内に含まれる。

【0013】

本発明がコンピュータ実行可能命令の一般的なコンテキストで具体化されるとき、この実施形態は、1つまたは複数のシステム、コンピュータ、またはその他のデバイスによって実行されるプログラムモジュールを含むことができる。一般に、プログラムモジュールは、特定のタスクを実施するか、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。通常、プログラムモジュールの機能は、様々な実施形態で望まれるように結合または分散させることができる。

【0014】

図1に、同期されるデータベースシステムを示す本発明の一実施形態100を例示する。サーバシステム102とクライアントシステム104とが、通信バス106で接続される。

【0015】

サーバシステム102は、アイテムリポジトリ110を含むストレージシステム108を有する。アイテム112、114、および116は、それぞれ、パラメータ118、122、および124からなる。アイテム112はファイル120に関係する。同様に、アイテム116はファイル126に関係する。

【0016】

クライアントシステム104は、アイテムリポジトリ130を含むストレージシステム128を有する。アイテム132、134、および136は、それぞれ、パラメータ138、142、および144からなる。アイテム132は、ゴースト化されたファイル(ghosted file)140に関係する。アイテム136は、同期されたファイル146に関係する。

【0017】

サーバシステム102およびクライアントシステム104は、データを記憶および操作することのできるコンピュータシステムまたはその他の任意のデバイスとすることができます。一実施形態では、サーバシステム102はサーバコンピュータであってよく、クライアントシステム104はポータブルラップトップコンピュータなどのクライアントシステムであってよい。他の実施形態では、クライアントシステム104はポータブルメディアプレーヤ、パーソナル情報マネージャ、または、少なくとも何らかのデータ記憶装置を有するその他のデバイスであってよい。

【0018】

通信バス106は、サーバ102とクライアント104とが通信できるための任意のタイプの通信機構とすることができます。例として、イーサネット(登録商標)、トークリング、シリアルまたはパラレル通信などの様々な有線通信プロトコル、様々な無線プロトコル、電話、あるいはその他の任意の通信媒体が挙げられるが、これらに限定されるものではない。クライアントシステム104がラップトップコンピュータである実施例では、通信バス106はインターネット接続とすることができます。場合によっては、様々な暗号化プロトコルを使用することもできる。

【0019】

ストレージデバイス108および128は、ハードディスクストレージシステムとすることができます、あるいは、フラッシュメモリなどの固体メモリを含めたその他の任意のタイプの記憶媒体とすることができます。場合によっては、記憶媒体は不揮発性メモリとするこ

10

20

30

40

50

とができ、このような場合以外では、揮発性メモリを使用することができる。任意のタイプのデータ記憶媒体を使用することができる。

【0020】

アイテム112、114、および116は、様々なアプリケーションにとって有用なメタデータおよびデータを含むことができる。典型的な汎用コンピュータシステムの実施例では、様々なアイテムは、コンピュータシステム上で動作可能なアプリケーションによって使用できる任意のタイプのデータに関係することができる。例えば、電話帳(phone contacts)、カレンダ上のイベント、実施すべき作業などに関係する情報を記憶するために、アイテムを作成することができる。

【0021】

各アイテムは様々なパラメータを含み、これらのパラメータは、そのタイプのアイテムに特有の値とすることができます。例えば、電話帳アイテムは、相手の名前、会社、住所、および電話番号を記憶するパラメータを含むことができる。アイテムの目的および機能に応じて、種々のタイプのアイテムが種々のパラメータを有することができる。

【0022】

他のアイテムは、特定の生データファイルに関係することができ、これは例えばアイテム112および116がそれぞれファイル120および126と関係を有するなどである。このような場合、アイテムのパラメータの一部またはすべては、ファイルから導出することができる。このようなアイテムは、ファイル支援アイテム(file backed item)と呼ぶことができる。例えば、オーディオ録音を含むファイルは、アーティスト、トラック名、録音の長さ、およびその他のパラメータなどのメタデータを含むファイル支援アイテムを有することができる。同様に、ワードプロセッサ文書を含むファイルは、文書の題名、作成者の名前、主題、およびその他の分類パラメータを含む、関連のファイル支援アイテムを有することができる。

【0023】

ファイル支援アイテムは、関係するファイルを有し、このファイルに関する様々なメタデータを含むことができる。これらのメタデータは、ファイルディレクトリを分析することによって確認できる導出データに加えて、非導出データを含むことができる。導出データは、生データファイルから特定の情報を選別して様々なパラメータ値を決定する分析ルーチンを実行することによって作成することができる。このようなルーチンは、様々なファイルのタイプごとに異なる場合がある。例えば、ワードプロセッサ文書に関連するファイル支援アイテムは、文書を走査して、題名、目次、キーワード、作成者、またはその他のパラメータを決定することによって作成することができる。これらのメタデータは、生データ内に直接含まれる。

【0024】

場合によっては、情報を生データファイルから選別して、または外部データベースに対して照会して、または他の方法で処理して、適切なパラメータ値を決定することができる。例えば、オーディオ録音は、アーティスト、曲名、アルバム、ジャンル、および元の発行日を決定できるように、オンラインデータベースに対して照合することができる識別子を有することができる。このようなメタデータは、生データ自体に導出メタデータが含まれていなくても、生データの内容に基づいて導出される。

【0025】

一部の実施形態では、データベース中の各ファイルが、1つの関係するファイル支援アイテムを有することができる。他の実施形態では、単一のファイル支援アイテムが複数の生データファイルに関係することができ、あるいは、単一の生データファイルが複数の関連するファイル支援アイテムを有することができる。

【0026】

非導出データは、生データ自体からは導出されない他の情報を含むことができる。例えば、オーディオ録音に関係するパラメータは、録音に対するユーザの評価、または録音が再生された回数を含むことができる。

10

20

30

40

50

【 0 0 2 7 】

このシステムは、導出メタデータと生データファイルの一方に対する変更が、他方に反映(propagate)されるように構成することができる。例えば、ワードプロセッサ文書が編集されているときに文書の題名に変更が施された場合、後にこの題名変更を、関連するファイル支援アイテムのメタデータ中で更新することができる。同様に、ファイル支援アイテムが新しい題名変更で更新された場合、この文書の題名を、逆に生データファイルに反映することができる。

【 0 0 2 8 】

本明細書の目的上、ファイル支援アイテムという用語は、アイテムという総称用語のサブセットを示している。ファイル支援アイテムに関するすべての記述および制限は、すべてのアイテムに当てはまるべきであり、総称的なアイテムに関する記述および制限のすべてではないが、その一部は、ファイル支援アイテムという用語に当てはまる場合がある。ファイル支援アイテムという用語は、少なくとも1つの関連ファイルの生データから何らかの方式で導出される少なくとも一部のメタデータを含むアイテムを指すべきである。

10

【 0 0 2 9 】

クライアントシステム104は、サーバシステム102と同期される。同期されるとき、サーバのアイテムリポジトリ110がクライアントのアイテムリポジトリ130と同期される。最初の同期時は、サーバシステム102は単に、アイテムリポジトリ110全体をクライアントシステム104にコピーすればよい。

20

【 0 0 3 0 】

後続の同期イベントは、前回の同期イベントの後にどのアイテムが更新または変更されたかを検出することを含むことができる。アイテムが更新されているが、反対のシステム上の対応するアイテムは前回の同期から変更されていない場合は、更新されたアイテムを反対のシステムにコピーすることができる。アイテムと、反対のシステム上の対応するアイテムとが両方とも更新されている場合は、一方または他方のアイテムを選択する決定を行わなければならない。一部の実施形態では、どちらのアイテムを保持するかを決定するための規則を使用することができ、他の実施形態では、ユーザに照会することができる。ファイルの同期も、アイテムの同期と同じ方法に従うことができる。

【 0 0 3 1 】

一部の実施形態では、同期の際に、各メタデータアイテムまたは生データファイルにインジケータを割り当てることができる。インジケータを使用して、アイテムまたは生データが更新されているか否かを判定することができる。一部の実施形態では、インクリメンタルなカウンタをバージョン識別子として使用することができる。メタデータまたは生データが更新されるごとにカウンタをインクリメントすることができ、それにより、同期イベントが発生したとき、サーバおよびクライアントのメタデータまたは生データのバージョン識別子を比較して、同期の後で更新が行われたかどうかを判定することができる。

30

【 0 0 3 2 】

他の実施形態では、タイムスタンプをバージョンインジケータとして使用することができる。タイムスタンプは、日付および時刻を含むことができ、あるいは、コード化された時間信号など別の時間インジケータとすることもできる。さらに他の実施形態では、単純なフラグが、アイテムまたは生データのセットが更新されているか否かを示すことができる。

40

【 0 0 3 3 】

図1において、アイテム112、114、および116はそれぞれアイテム132、134、および136と同期される。同様に、ファイル126はファイル146と同期される。

【 0 0 3 4 】

ファイル140は、ファイル120と同期されないが、「ゴースト化」されているものとして示されている。ファイル140は、ディレクトリシステム中のプレースホルダ(placeholder)など、ファイルの属性の一部を有することができますが、ファイル140の基

50

礎をなす生データは、クライアントシステム104上で利用可能とすることはできない。クライアントシステム104がサーバシステム102から切断されているとき、ファイル140に関係するアイテム132は利用可能とすることができますが、ファイル140中の生データは利用可能とすることはできない。

【0035】

ファイル140に関連するメタデータはファイル支援アイテム132に含まれているので、一部の機能および操作は、クライアントシステム104のユーザに利用可能とすることができます。ファイル140に含まれるオーディオファイルの例を用いると、クライアントシステム104は、アイテム132に含まれるメタデータのみを使用して、様々なオーディオトラックを表示でき、個々のトラックを評価でき、トラックをプレイリストまたは他の有用なグループに分類でき、その他の操作を実施できる。10

【0036】

加えて、アイテム132に含まれるメタデータを使用して、ファイル140の様々な側面を操作することができる。例えば、アイテム132を使用して、ファイル140がゴースト化されるのではなく同期されるべきであることを示すことができる。このような指示により、次の同期イベント時にファイル120がファイル140にコピーされるようにすることができる。別の例では、アイテム132を使用して、ファイル140を異なるディレクトリまたは階層構造に移動することができる。同様に、アイテム132を使用して、ファイル140が削除されるべきであることを示すことができる。

【0037】

アクションを行う対象となるデータと、後の実行のために記憶できるアクションを実行する一機構との両方としてアイテム132を使用することにより、その他多くのアクションを示すことができる。オーディオ録音の例を用いると、あるジャンルの音楽に関する照会の結果として、アイテム132を表示することができる。ユーザは、データベースから削除するためにアイテム132を選択することができる。ファイル140は単にゴースト化されているだけで、生データファイルではないので、対応するアイテム112およびファイル120の削除は、次の同期までは行えない。したがって、削除操作は、次の同期が行われるまでパラメータとしてアイテム132に記憶しておくことができる。20

【0038】

ファイル140などのファイルをゴースト化することの実際的な利点の1つは、多くの生データファイルを有する大きなデータベースが、クライアントストレージ128が保持できるよりもずっと大きくなれることである。しかし、アイテムリポジトリ130に記憶された様々なアイテムにはメタデータが含まれているので、データの有用性の多くは依然として存在する。30

【0039】

一部の実施形態では、ファイル120および126に含まれる生データは、特に知る必要があるときにだけ同期される機密扱いの、または秘密の生データファイルとすることができます。ファイルに関係するメタデータは、非機密扱い、または生データよりも低いレベルの機密扱いとすることができます。例えば、競合的な機密情報、秘密軍事情報、またはその他の機密情報を含む生データファイルをサーバシステム102上で保持し、必要でない限りクライアントシステム104に転送されないようにすることができます。40

【0040】

他の実施形態では、生データは、サーバシステム102とクライアントシステム104との間で転送するのに長時間を要する場合のある、並外れて大きいファイルであることがある。例えば、生データファイルが、ギガバイト単位のサイズであるほんの少数の動画ファイルであった場合、ムービーライブリの同期は、とりわけ通信バス106のスループットに応じて、数分間も、さらには数時間もかかることがある。

【0041】

アイテムリポジトリ110および130は、メタデータおよび他のデータを記憶するのに有用な任意のタイプのデータストレージとすることができます。一部の実施形態では、ア50

アイテムリポジトリ 110 および 130 は、1つのディレクトリ中の単一ファイルであってもよく、あるいは複数ディレクトリ中の複数ファイルであってもよい。他の実施形態では、アイテムリポジトリ 110 および 130 は、その他の任意のタイプのデータ記憶方式で生データと共に記憶することができる。

【0042】

図 2 は、ファイルからメタデータを生成することができる方法の例示的な一実施形態 200 のフローチャートである。プロセスはブロック 202 で開始する。ブロック 204 で、ファイルがファイルシステムに記憶される。ブロック 206 で、ファイルに関するアイテムが作成される。ブロック 208 でファイルが特別なタイプである場合は、ブロック 210 で、分析プログラムを実行してメタデータが生成される。ブロック 212 で、メタデータはアイテムに記憶される。ブロック 208 でファイルが特別なタイプでない場合、またはブロック 212 でメタデータがアイテムに記憶されたときは、ブロック 214 で、アイテムはリポジトリに記憶される。ブロック 216 でプロセスは終了する。
10

【0043】

実施形態 200 は、生データからメタデータを生成することができる一方法を例示している。実施形態 200 では、生データは、コンピュータシステム中のファイルの形態をとることができる。ファイルが所定のタイプである場合、生データを分析することによって、1つまたは複数のメタデータパラメータ値を決定することができる。これらのパラメータはアイテムに記憶される。

【0044】

実施形態 200 は、アプリケーションが生データファイルを作成したときに有用だが、メタデータは第 2 のアプリケーションで有用となり得る。例えば、生データファイルがワードプロセッシングプログラムによって作成された場合、作成者、題名、主題などの様々なメタデータパラメータを生データから抽出することができる。このメタデータは、文書管理アプリケーションが、文書に関する様々なパラメータを照会、表示、および操作するのに使用することができる。
20

【0045】

別の実施形態では、テスト装置によって、数値テストデータを含む生データファイルを作成することができる。ブロック 210 で、分析プログラムが動作して、最大値および最小値などの測定要約パラメータ (measurement summary parameter) をデータから抽出することができる。テスト装置、テスト主題、サンプルサイズ、または他のメタデータを識別するパラメータなど、追加のメタデータを別個にアイテムに加えることもできる。
30

【0046】

図 3 は、同期シーケンスの一実施形態 300 の図表現である。このプロセスは、初期接続 301 と、システムが切断されている期間 325 と、再接続時 326 のアクションとを示す。

【0047】

初期接続 301 で、サーバ 302 とクライアント 304 が接続される。サーバ 302 は、アイテムリポジトリ 310 中のアイテム 308 を含むストレージシステム 306 を有する。アイテム 308 の一部は、関係するファイル 312 および 314 を有する。初期接続 301 の間、クライアントストレージ 316 は、サーバ 302 からのアイテム 308 のコピーであるアイテム 320 を含む同期されたアイテムリポジトリ 318 でポピュレートされる。2つのサーバ参照ファイル (server referenced file) 312 および 314 は、クライアント 304 上ではファイル 322 および 324 としてゴースト化される。
40

【0048】

初期接続 301 の間、アイテムリポジトリ 310 と 318 が同期され、生データファイル 312 および 314 は、クライアント上ではゴースト化ファイル 322 および 324 としてゴースト化される。

【0049】

ブロック 325 は、接続が断たれているときのシステムの状態を示す。サーバ 302 は
50

ストレージ306を維持する。クライアント304は、様々なメタデータおよび他の機能を実施できるようにするアイテムリポジトリ318を含むストレージ316を有する。切断されている間、ゴースト化ファイル324を、後に同期するために選択することができる。

【0050】

ブロック326は、再接続時のシステムを示す。サーバ302とクライアント304は再接続される。ストレージシステム306と316との間で、アイテムリポジトリ310および318が同期される。同期ルーチンの一部として、ファイル324の生データを切断時でも利用可能とすることができるように、ゴースト化ファイル324をファイル314に置き換える。

10

【0051】

実施形態300は、ゴースト化ファイルを完全に使用可能なファイルに切り替える一方法を例示している。実施形態300では、ゴースト化ファイルは、「非ゴースト化」されるように、または次の同期時に完全に機能するファイルとなるように選択することができる。生データファイルに関するアイテムおよびメタデータは、ブロック325に示すオフラインモードで利用可能である。これらのアイテムおよびメタデータは、特定の生データファイルをオフラインで利用可能にするために選択することを含めて、多くの様々な機能に使用することができる。

【0052】

図4は、ゴースト化ファイルを同期させる方法の例示的な一実施形態400のフローチャートである。ブロック402で、クライアントとサーバが接続される。ブロック404で、クライアントとサーバとの間の変更が同期される。接続されている間、ブロック406でクライアントは、サーバ上で直接的にアイテムを操作、編集、追加、および削除することができ、さらにサーバ上で直接的に生データファイルにアクセスすることができる。ブロック408で、クライアントは、リモートで使用する生データファイルを識別することができ、ブロック410で、これらの生データファイルはクライアントに同期される。

20

【0053】

ブロック412でクライアントとサーバが切断されると、ブロック414で、クライアントは、アイテムリポジトリのローカルコピーを使用してアイテムを操作、編集、追加、および削除することができ、さらに、同期された生データファイルにローカルでアクセスすることができる。ブロック416で、クライアントはまた、ゴースト化された生データファイルをリモートで使用するために識別することもできる。さらに、ブロック418で、クライアントは、生データファイルを同期から除外することもできる。このアクションは、ブロック418で生データを実際にクライアントデータストアから削除することを含むことができる。

30

【0054】

ブロック402でクライアントとサーバが再接続されると、ブロック404で、アイテムに対する変更があればそれらの変更は同期され、さらに、前に同期された生データファイルに対する変更があればそれらの変更も同期される。「非ゴースト化」されたファイル、またはオフライン利用可能としてタグ付けされたファイルがあれば、ブロック404でこれらのファイルもまた同期させることができる。

40

【0055】

生データおよびメタデータを含むデータベースは、2段階で同期される。第1段階では、メタデータがサーバとクライアントとの間で同期され、生データはサーバシステム上に保持される。クライアントシステムは、接続または切断されているとき、クライアントシステム上で利用可能となるべき1つまたは複数の生データファイルを選択することを含めてどのような目的でもメタデータを使用および操作することができる。同期の第2段階は、選択された生データをクライアントシステムに同期せることである。同期された生データは、切断されているときにクライアントシステム上で操作、変更、および使用することができる。再接続時、メタデータまたは生データに施された変更があれば、それらの変

50

更はシステム間で同期される。

【0056】

実施形態300および400の一実施例は、音楽ファイルのためのファイル構造を含むことができる。ファイル312および314は、MP3フォーマットまたはその他のデジタルフォーマットなど、デジタル化されたオーディオファイルとすることができます。各オーディオファイルについてのアイテム308は、アーティスト、トラック名、および録音の長さを含むことができる。アイテムは、アイテムリポジトリ310に記憶することができ、アイテムリポジトリ318と完全に同期させることができる。

【0057】

図3に示された状態325のように接続が断たれた後、クライアントシステム304のユーザは、リポジトリ318中のアイテムを閲覧、操作、および変更することができる。例えばユーザは、アイテムをアーティストごとにソートしたり、一部のアイテムを選択したり、アイテムをオーディオプレイリストに配置したりすることができます。オーディオファイルがデータストア316内にあっても、ユーザは、オーディオファイルを使用する多くの様々な作業を実施することができる。ユーザはさらに、同期されるようにファイル324にフラグを立てるこどもできる。

10

【0058】

システムが状態325で切断されている間、ファイル322および324のためにプレースホルダを使用することができる。ファイルは「ゴースト化」され、ディレクトリ構造内で見えていてもよいし、見えなくてよい。例えば、ゴースト化ファイルは、灰色に消されて表示されてもよいし、または半透明に表示されてもよい。

20

【0059】

状態326で接続が復元されると、アイテムリポジトリ310および318に対する変更があればそれらの変更は同期され、次いで、ファイル324など、フラグが立てられたゴースト化ファイルがあればそれらのファイルも同期される。完全に同期されたファイル324は、次いでクライアント304に転送され、クライアント304がサーバ302から切断されたときでも使用可能である。

【0060】

図5は、ファイル支援アイテムを有する同期されるシステムを処理する方法の例示的な一実施形態500のフローチャートである。ブロック502で、データストアが同期され、次いでブロック504で、クライアントとサーバが切断される。ブロック506で、サーバはファイル支援アイテムについてメタデータと生データファイルの両方を含むので、サーバ上では両方に対して変更を施すことができる。ブロック508で、クライアントシステムはファイル支援アイテムのメタデータ部分しか有さないので、クライアント上ではメタデータに対する変更しかできない。ブロック510で、クライアントとサーバが再接続されたとき、すべての変更が同期される。ブロック512で各ファイルにつき、ブロック514でファイルが更新されている場合は、ブロック516で変更をメタデータに反映する。ブロック518で各メタデータにつき、ブロック520でメタデータが更新されている場合は、ブロック522で変更を生データファイルに反映する。すべての変更が反映された後、ブロック524で、操作は接続された状態で再開する。

30

【0061】

実施形態500は、ファイル支援アイテムに対する変更、またはファイル支援アイテムに関連する生データに対する変更を最新に維持することができる一方法である。ファイル支援アイテムの生データと関連メタデータの一方に対する変更を使用して、他方が更新される。

【0062】

例えば、複数のワードプロセッシング文書ファイルがそれぞれ、ファイル支援アイテムに関連する場合がある。生データファイルから、章題、題名、キーワード、作成者などのパラメータを導出して、ファイル支援アイテム中のメタデータとして記憶することができる。データストアが同期されるとき、クライアントシステムには、メタデータを有するフ

40

50

ファイル支援アイテムが記憶され、サーバには、ファイル支援アイテムと生データの両方が記憶される。

【0063】

クライアントとサーバが切断されているとき、サーバ上では、ワードプロセッサ文書を直接的に編集することによって生データファイルに変更を施すことができる。さらに、ファイル支援アイテムを編集することによってメタデータに変更を施すこともできる。クライアント側では、生データは利用できないので、メタデータに対する変更しかできない。この例では、サーバ上で、1つのワードプロセッサ文書を編集して、新しい章を文書に追加することができる。生データファイルから章題を導出して、ファイル支援アイテムのメタデータに追加することができる。同様に、クライアントシステム上で、別のファイル支援アイテム中の作成者パラメータを変更することができる。

【0064】

クライアントとサーバが再接続されたとき、サーバ上のファイルに加えられた追加の章などの生データファイルに対する変更が、関連するファイル支援アイテムのメタデータに反映される。場合によっては、変更は、サーバ上では変更が行われたときに反映することができるが、クライアントには再接続まで反映されないようにすることができる。

【0065】

同様に、上記の例における作成者変更などのクライアント上のメタデータに対する変更は、クライアントとサーバが再接続されたときにサーバ上の生データファイル中に反映されることになる。ワードプロセッサ文書が開かれ、ワードプロセッサ文書は生データファイルの中で作成者の値を含む部分を見つけ、古い値を新しい値で上書きすることができる。

【0066】

本発明に関する上記の記載は、例示および説明の目的のために提示されている。この記載は、網羅的なものではなく、開示した正確な形態に本発明を限定するものでもなく、以上の教示に鑑みて他の変更形態および変形形態も可能である。本発明の原理およびその実際的な適用例を最も適切に説明して、それにより、企図される特定の使用に適する様々な実施形態および様々な変更形態において当業者が本発明を最もよく利用できるようにするために、各実施形態は選択および記載されている。添付の特許請求の範囲は、従来技術によって制限される場合を除き、本発明の他の代替形態を含むように解釈されることが意図されている。

【図面の簡単な説明】

【0067】

【図1】同期されるデータベースシステムを示す一実施形態のブロック図である。

【図2】ファイルに関するアイテムを作成する方法を示す一実施形態を例示するフローチャート図である。

【図3】ゴースト化ファイルを同期させるためのイベントシーケンスを示す一実施形態のブロック図である。

【図4】ゴースト化ファイルを同期させる方法を示す一実施形態のフローチャート図である。

【図5】生データファイルおよびメタデータに対する変更を同期させる方法を示す一実施形態のフローチャート図である。

【符号の説明】

【0068】

112、114、116、132、134、136 アイテム

118、122、124、138、142、144 パラメータ

120、126 ファイル

140 ゴースト化ファイル

146 同期されたファイル

308、320 アイテム

10

20

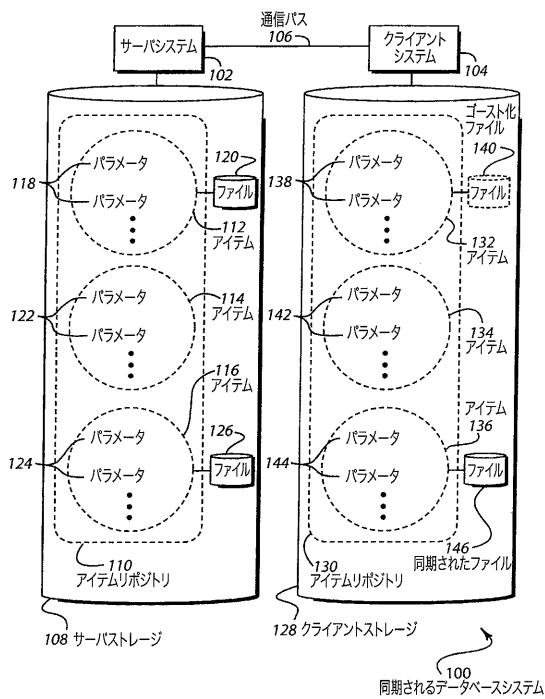
30

40

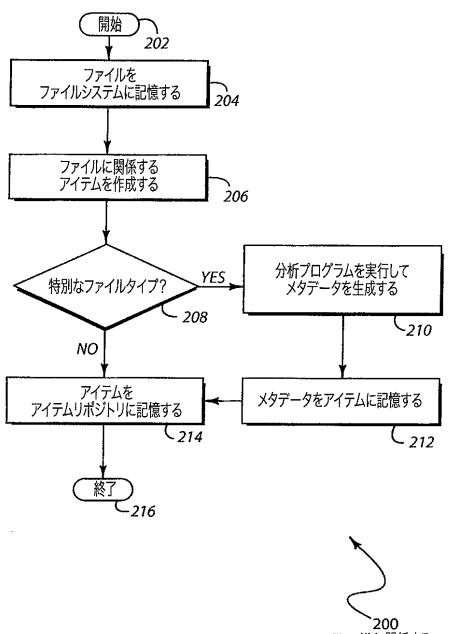
50

- 3 1 2、3 1 4 ファイル
 3 2 2 ゴースト化ファイル
 3 2 4 同期されるように選択されたゴースト化ファイル

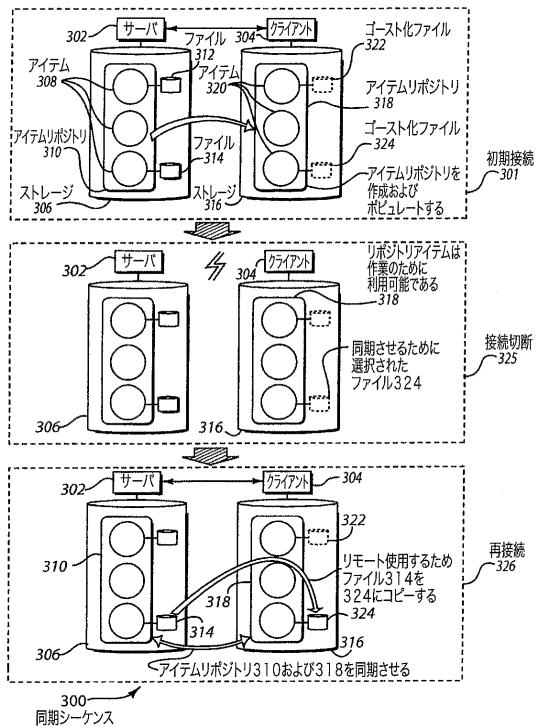
【図1】



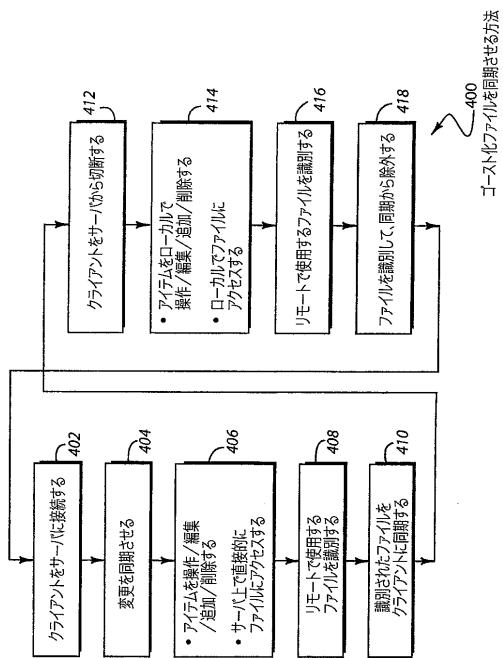
【図2】



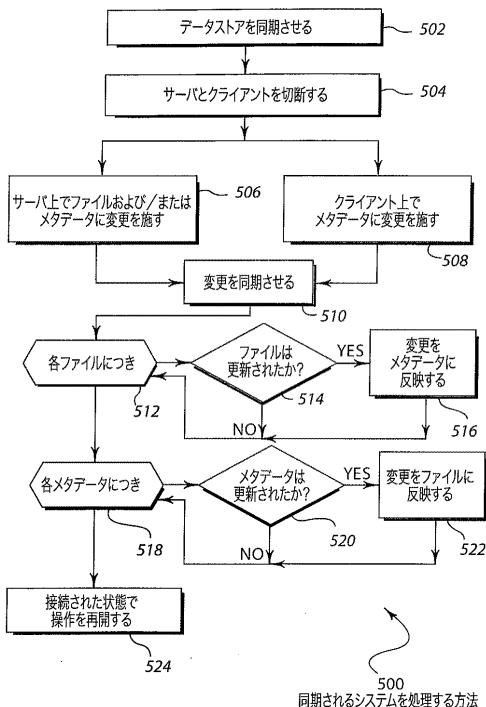
【図3】



【図4】



【図5】



フロントページの続き

(72)発明者 イレーナ フーディス

アメリカ合衆国 98052 ワシントン州 レッドmond 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 レフ ノヴィック

アメリカ合衆国 98052 ワシントン州 レッドmond 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 オーケチュクウ シー . エシェルオ

アメリカ合衆国 98052 ワシントン州 レッドmond 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 ラジエシュ エム . ラオ

アメリカ合衆国 98052 ワシントン州 レッドmond 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 ユンシン ウー

アメリカ合衆国 98052 ワシントン州 レッドmond 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 池田 聰史

(56)参考文献 特開2004-265418 (JP, A)

特開2005-032249 (JP, A)

特開2004-334858 (JP, A)

特開2002-149464 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/00

JSTPlus (JDreamII)