

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4753693号
(P4753693)

(45) 発行日 平成23年8月24日 (2011. 8. 24)

(24) 登録日 平成23年6月3日 (2011. 6. 3)

(51) Int. Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 3 3 J

G 0 6 F 12/00 5 4 5 Z

G 0 6 F 12/00 5 3 1 D

請求項の数 13 (全 15 頁)

(21) 出願番号 特願2005-330703 (P2005-330703)
 (22) 出願日 平成17年11月15日 (2005. 11. 15)
 (65) 公開番号 特開2006-178932 (P2006-178932A)
 (43) 公開日 平成18年7月6日 (2006. 7. 6)
 審査請求日 平成20年11月6日 (2008. 11. 6)
 (31) 優先権主張番号 11/025, 165
 (32) 優先日 平成16年12月20日 (2004. 12. 20)
 (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 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】 スナップショットなしでアイテムを同期化するシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

ハードウェア/ソフトウェアインターフェースシステムにおけるピアコンピュータシ
 ステム間の混合要素を同期化する方法であって、

第1のピアコンピュータシステム上の混合要素を識別するステップであって、前記混合
 要素は、構造化データコンポーネント及び非構造化ファイルストリームを備え、前記ハー
 ドウェア/ソフトウェアインターフェースは、前記構造化データコンポーネントのスナッ
 プショット作成を可能にし、前記非構造化ファイルストリームのスナップショット作成を
 可能にしないものであるステップと、

前記構造化データコンポーネントのスナップショットを作成するステップであって、 i
) 前記混合要素の同期化に関連する継続時間中に、ターゲット構造化データコンポーネン
 トに関連する変更がコピー構造化データコンポーネントに対して行われるように、前記構
 造化データコンポーネントを前記コピー構造化データコンポーネントにコピーするステッ
 プ、 i i) 前記構造化データコンポーネントが同期ピアにコピーされている間は前記構
 造化データコンポーネントに対する変更を防ぐステップ、 i i i) 前記構造化データを同期
 ピアにコピーする間に前記構造化データに対して変更を行うであろうプロセスのために前
 記構造化データコンポーネントの第1のコピーを作成し、かつ、前記プロセスが前記構
 造化データに対してではなく前記第1のコピーに対して変更を行うことを可能にするステッ
 プ、及び、 i v) 前記構造化データコンポーネントが前記同期ピアにコピーされた後に、
 前記構造化データを前記第1のコピーで更新するステップ、を含むステップと、

10

20

前記スナップショットを作成するステップと同時に、前記ファイルストリームに関連する第 1 の変更値を判定するステップと、

前記構造化データコンポーネントの変更値および前記ファイルストリームの前記第 1 の変更値に基づいて、前記混合要素が第 2 のピアコンピュータにコピーされるべきか否かを判定するステップと、

前記非構造化ファイルストリームを前記第 2 のピアコンピュータへのコピーのために準備するために、前記非構造化ファイルストリームをロックするステップであって、前記ファイルストリームの第 2 の変更値に関連するものであるステップと、

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変更されなかったか否かを判定するために、前記ファイルストリームの前記第 1 の変更値を前記ファイルストリームの前記第 2 の変更値と比較するステップと、

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変更されなかった場合に、前記非構造化ファイルストリームを前記第 2 のピアコンピュータにコピーするステップと

を含むことを特徴とする方法。

【請求項 2】

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変わった場合は、同期ピアに例外をもたらし、かつ、前記非構造化ファイルストリームをアンロックするステップをさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記非構造化ファイルストリームをロックするステップは、トランザクションリード動作によって達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 4】

前記非構造化ファイルストリームをロックするステップ中に前記非構造化ファイルストリームをロックすることができない場合は、同期ピアに例外をもたらし、(a) 前記非構造化ファイルストリームがロックされた後に前記非構造化ファイルストリームの第 2 の変更値を判定するステップと、(b) 前記第 1 の変更値と比較された前記第 2 の変更値が、前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間に前記非構造化ファイルストリームに対する変更が生じなかったことを示す場合は、前記非構造化ファイルストリームを前記同期ピアにコピーし、コピーが終了した後は、前記非構造化ファイルストリームをアンロックするステップとを実行しないことを特徴とする請求項 1 に記載の方法。

【請求項 5】

ハードウェア制御装置によって実施される方法であって、前記ハードウェア制御装置は、前記方法の各ステップを実施するための手段を備えることを特徴とする請求項 1 に記載の方法。

【請求項 6】

ハードウェア / ソフトウェアインターフェースシステムにおけるピアコンピュータシステム間の混合要素を同期化するためのコンピュータ可読記憶媒体であって、少なくとも 1 つのプロセッサによって実行された場合に、

第 1 のピアコンピュータシステム上の混合要素を識別するステップであって、前記混合要素は、構造化データコンポーネント及び非構造化ファイルストリームを備え、前記ハードウェア / ソフトウェアインターフェースは、前記構造化データコンポーネントのスナップショット作成を可能にし、前記非構造化ファイルストリームのスナップショット作成を可能にしないものであるステップと、

前記構造化データコンポーネントのスナップショットを作成するステップであって、 i) 前記混合要素の同期化に関連する継続時間中にターゲット構造化データコンポーネント

10

20

30

40

50

に関連する変更がコピー構造化データコンポーネントに対して行われるように、前記構造化データコンポーネントを前記コピー構造化データコンポーネントにコピーするステップ、
i i) 前記構造化データコンポーネントが同期ピアにコピーされている間は、前記構造化データコンポーネントに対する変更を防ぐステップ、
i i i) 同期ピアに前記構造化データをコピーする間に前記構造化データに対して変更を行うであろうプロセスのために前記構造化データコンポーネントの第 1 のコピーを作成し、かつ、前記プロセスが前記構造化データに対してではなく前記第 1 のコピーに対して変更を行うことを可能にするステップ、及び、
i v) 前記構造化データコンポーネントが前記同期ピアにコピーされた後に、前記構造化データを前記第 1 のコピーで更新するステップ、を含むステップと、

前記スナップショット作成と同時に前記ファイルストリームに関連する第 1 の変更値を判定するステップと、

前記構造化データコンポーネントの変更値および前記ファイルストリームの前記第 1 の変更値に基づいて、前記混合要素が第 2 のピアコンピュータにコピーされるべきか否かを判定するステップと、

前記非構造化ファイルストリームを前記第 2 へのピアコンピュータのコピーのために準備するために、前記非構造化ファイルストリームをロックするステップであって、前記ファイルストリームの第 2 の変更値に関連するものであるステップと、

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変更されなかったか否かを判定するために、前記ファイルストリームの前記第 1 の変更値を前記ファイルストリームの前記第 2 の変更値と比較するステップと、

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変更されなかった場合に、前記非構造化ファイルストリームを前記第 2 のピアコンピュータにコピーするステップと

を含む方法を実施するコンピュータ実行可能命令を記憶していることを特徴とするコンピュータ可読記憶媒体。

【請求項 7】

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変わった場合は、同期ピアに例外をもたらし、前記非構造化ファイルストリームをアンロックするステップをさらに含む方法を実施するためのコンピュータ実行可能命令をさらに含むことを特徴とする請求項 6 に記載のコンピュータ可読記憶媒体。

【請求項 8】

前記非構造化ファイルストリームをロックするステップはトランザクションリード動作によって達成されるコンピュータ実行可能命令をさらに備えることを特徴とする請求項 6 に記載のコンピュータ可読記憶媒体。

【請求項 9】

前記非構造化ファイルストリームをロックするステップ中に前記非構造化ファイルストリームをロックすることができない場合は、同期ピアに例外をもたらし、
(a) 前記非構造化ファイルストリームがロックされた後に前記非構造化ファイルストリームの第 2 の変更値を判定するステップと、
(b) 前記第 1 の変更値と比較された前記第 2 の変更値が、前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間に前記非構造化ファイルストリームに対する変更が生じなかったことを示す場合は、前記非構造化ファイルストリームを前記同期ピアにコピーし、コピーが終了した後は、前記非構造化ファイルストリームをアンロックするステップとを実行しないコンピュータ実行可能命令をさらに備えることを特徴とする請求項 6 に記載のコンピュータ可読記憶媒体。

【請求項 10】

ハードウェア/ソフトウェアインターフェースシステムにおけるピアコンピュータシス

10

20

30

40

50

テム間の混合要素を同期化するためのシステムであって、

第1のピアコンピュータシステム上の混合要素を識別するステップであって、前記混合要素は、構造化データコンポーネント及び非構造化ファイルストリームを備え、前記ハードウェア/ソフトウェアインターフェースは、前記構造化データコンポーネントのスナップショット作成を可能にし、前記非構造化ファイルストリームの前記スナップショット作成を可能にしないものであるステップと、

前記構造化データコンポーネントのスナップショットを作成するステップであって、
i) 前記混合要素の同期化に関連する継続時間中にターゲット構造化データコンポーネントに関連する変更がコピー構造化データコンポーネントに対して行われるように、前記構造化データコンポーネントを前記コピー構造化データコンポーネントにコピーするステップ、
ii) 前記構造化データコンポーネントが同期ピアにコピーされている間は、前記構造化データコンポーネントに対する変更を防ぐステップ、
iii) 同期ピアに前記構造化データをコピーする間に前記構造化データに対して変更を行うであろうプロセスのために前記構造化データコンポーネントの第1のコピーを作成し、かつ、前記プロセスが前記構造化データに対してではなく前記第1のコピーに対して変更を行うことを可能にするステップ、及び、
iv) 前記構造化データコンポーネントが前記同期ピアにコピーされた後は、前記構造化データを前記第1のコピーで更新するステップ、を含むステップと、

前記スナップショット作成と同時に前記ファイルストリームに関連する第1の変更値を判定するステップと、

前記構造化データコンポーネントの変更値および前記ファイルストリームの前記第1の変更値に基づいて、前記混合要素が第2のピアコンピュータにコピーされるべきか否かを判定するステップと、

前記非構造化ファイルストリームを前記第2のピアコンピュータへのコピーのために準備するために前記非構造化ファイルストリームをロックするステップであって、前記ファイルストリームの第2の変更値に関連するものであるステップと、

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変更されなかったか否かを判定するために、前記ファイルストリームの前記第1の変更値を前記ファイルストリームの前記第2の変更値と比較するステップと、

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変更されなかった場合は、前記非構造化ファイルストリームを前記第2のピアコンピュータにコピーするステップと

のための少なくとも1つのサブシステムを備えることを特徴とするシステム。

【請求項11】

前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化ファイルストリームがロックされた時までの間の時間中に前記非構造化ファイルストリームが変わった場合は、同期ピアに例外をもたらし、前記非構造化ファイルストリームをアンロックするステップのための少なくとも1つのサブシステムをさらに備えることを特徴とする請求項10に記載のシステム。

【請求項12】

前記非構造化ファイルストリームをロックするステップは、トランザクションリード動作によって達成されるものである少なくとも1つのサブシステムをさらに備えることを特徴とする請求項10に記載のシステム。

【請求項13】

前記非構造化ファイルストリームをロックするステップ中に前記非構造化ファイルストリームをロックすることができない場合は、同期ピアに例外をもたらし、
(a) 前記非構造化ファイルストリームがロックされた後に前記非構造化ファイルストリームの第2の変更値を判定するステップと、
(b) 前記第1の変更値と比較された前記第2の変更値が、前記構造化データコンポーネントのスナップショットが作成された時から前記非構造化フ

ファイルストリームがロックされた時までの間に前記非構造化ファイルストリームに対する変更が生じなかったことを示す場合は、前記非構造化ファイルストリームを前記同期ピアにコピーし、コピーが終了した後は、前記非構造化ファイルストリームをアンロックするステップとを実行しない少なくとも1つのサブシステムをさらに備えることを特徴とする請求項10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、一般には複数のコンピュータシステム間でのデータの同期化に関し、より詳細には、同期化中に大きな非構造化混合要素(mixed elements)(ME) 10
ファイルストリームのスナップショットの作成が可能なハードウェア/ソフトウェアインターフェースシステム環境におけるピアコンピュータシステム間で構造化データおよび非構造化ファイルストリームを含むMEを同期化することに関する。

【背景技術】

【0002】

ピアトゥピアの同期化環境およびその基礎を成すハードウェア/ソフトウェアインターフェースシステムがある。ここでは、とりわけ、格納可能な情報の自律ユニット(アイテム)が、同期化のための基本データ変更要素(basic data change element)を構成するいくつかのコンポーネント変更ユニット(component 20
change units)(CU)を含む、ハードウェア/ソフトウェアインターフェースシステム(例えば、オペレーティングシステムがあるが、ただしこれに限定されない)について説明する。各CUは、同期化システムが、ピアコンピュータシステム上で更新される必要がある変更はどれなのかを追跡し、他のピアコンピュータシステムから受け取ったどの変更が適用されるべきかを追跡できるようにする、関連するメタデータを有する。

【0003】

ある種のハードウェア/ソフトウェアインターフェースシステムでは、格納可能な情報の自律ユニット(例えばアイテム)の一部は、2種類のもの、つまり構造化されたもの、または混合されたものとして行うことができる。構造化要素(SE)は、十分に構造化されたデータだけを含むが、一方混合要素(ME)は十分に構造化されたデータ、非構造化ファイルストリームのどちらも含む。同期化に関して、SEはある種のサイズ制限を満たし、必要に応じてコピーが作成できるほど十分に小さいため、同期化も簡単に行うことができる。一方、MEは通常、アイテムの通常の制限を越えるデータ構造に使用され、そのサイズゆえに簡単にコピーすることはできない。

【0004】

ある種のハードウェア/ソフトウェアインターフェースシステムでは、アイテムの同期化は通常、同期化中にアイテムがコピーできるようにするトランザクション分離レベル(TIL)で実行される。具体的には、同期ピア(synchronization peer)にコピーされている間には他のプロセスによる書き込みができず、その代わりにその他のプロセスによってターゲットアイテムのコピーが作成されて改変され、次いでターゲットアイテムの同期が完了した際に、このコピーでオリジナルを上書きするようにターゲットアイテムの「スナップショット」が取られる。この必要に応じたコピー作成により、同期化システムは、実質的にターゲットアイテムが同期ピアにコピーされている間に改変されることを防ぐことができ(したがって、同期化プロセスの動作中に変化していない完全な「スナップショット」が提供され)、一方、他のプロセスはターゲットアイテムのコピーを改変することが可能になり、このコピーが、ターゲットアイテムの同期化が完了した時点で、最終的にターゲットアイテムに置き換わる。この特定のTILは、スナップショットTIL(STIL)と呼ぶことができる。

【発明の開示】

【発明が解決しようとする課題】

10

20

30

40

50

【 0 0 0 5 】

しかし、S EはスナップショットT I Lを使用して同期化することができるが、ファイルストリーム自体のサイズの故に、また非常に大きいデータオブジェクトはその一時的コピーの作成さえ望ましくないため、多くの場合、M Eのファイルストリームのスナップショットは作成されない。したがって、当技術分野では、M Eを同期化する、特に同期化中に大きな非構造化M Eファイルストリームのスナップショットの作成が可能ではないハードウェア/ソフトウェアインターフェースシステム環境におけるピアコンピュータシステム間でM Eのファイルストリームコンポーネントを同期化するシステムおよび方法が必要とされる。

【課題を解決するための手段】

10

【 0 0 0 6 】

本発明の様々な実施形態は、同期化中に大きな非構造化M Eファイルストリームのスナップショットの作成が可能ではないハードウェア/ソフトウェアインターフェースシステム環境におけるピアコンピュータシステム間でM Eを同期化する（特にM Eのファイルストリームコンポーネントを同期化する）システムおよび方法を対象としている。このような実施形態のいくつかでは、この方法はM Eの同期化を2ステップで行うことを含み、その1つはM Eの構造化データコンポーネント（およびファイルストリームの現バージョン（c v）を含むが、ファイルストリーム自体は含まない）のスナップショットを作成するステップであり、もう1つは、ロックされた後、ファイルストリームがM Eデータコンポーネントのスナップショット作成時から変更されていないと判断された場合、かつその場合に限り、ファイルストリームを、受け取る側の同期ピアにロックし送信するステップである。

20

【 0 0 0 7 】

同期化中、M Eの構造化データコンポーネントはスナップショットT I Lを使用して同期化され、このスナップショットが作成された時点の対応するファイルストリームの現バージョン（c v F）が、第1の値（c v F 0）として記憶される（M Eのファイルストリームはそれ自体の変更ユニットを備え、したがってそれ自体の変更バージョンをもつ）。次いで、同期化中にファイルストリームのコピーを作成する準備が整った場合（必要であれば、同期化の要件に基づいて）、ファイルストリーム自体が（別のプロセスがファイルストリームを変更するのを防ぐために）ロックされ、コピー作成に先立ってc v Fが再びチェックされて、第1の値（c v F 0）と比較される第2の値（c v F 1）が決定される。2つの値が同じである（したがって、ファイルストリームが変更されていないことを示す）場合、ファイルストリームは同期ピアにコピーされる。一方、c v F 1がc v F 0と同じでない場合は、M Eの構造化データのスナップショットが作成されたとき以降にファイルストリームが更新されたことを示し、このケースではファイルストリームはコピーされず、ファイルストリームの変更ユニットは同期化の「例外」を、すなわち同期の失敗が起こったことを反映し、そのことが受け取る側の同期ピアに認識され、当然の成り行きとして、後で2つのピア間での次の同期化処理中にM Eを同期化させる試みが行われる。

30

【 0 0 0 8 】

前述の要約および好ましい実施形態の以下の詳細な説明は、添付の図面と併せて読めば、よりよく理解されよう。本発明を説明するために、図面には本発明の例示的な構成が示されているが、本発明は開示された特定の方法および手段のみに限定されるものではない。

40

【発明を実施するための最良の形態】

【 0 0 0 9 】

主題は、法的要件を満たすために、詳細に説明されている。しかし、その説明自体は、本発明の範囲の限定を意図するものではない。そうではなくて、本発明者らは、請求される主題が、現在のまたは将来の技術と関連して、本明細書に説明されたものと類似の異なるステップまたはステップの組合せを含むように、別のやり方で実施できることを企図している。さらに、「ステップ」という用語は、使用される方法の様々な要素を意味するよ

50

うに本明細書では使用されることがあるが、この用語は、個々のステップの順序が明示的に記載されていない限り、本明細書で開示される様々なステップの中の、またはステップ間の特定の順序を示唆していると解釈してはならない。

【 0 0 1 0 】

コンピュータ環境

本発明の数多くの実施形態は、コンピュータ上で実行することができる。図 1 および以下の議論は、本発明を実施できる適切なコンピューティング環境の簡単な概要を提供するためのものである。必須ではないが、本発明はクライアントワークステーションやサーバなどのコンピュータで実行される、プログラムモジュールなどのコンピュータ実行可能命令という一般的なコンテキストで説明する。一般的に、プログラムモジュールとしては、特定のタスクを実行する、または特定の抽象データ型を実装したルーチン、プログラム、オブジェクト、コンポーネント、データ構造などがある。さらに、本発明はハンドヘルドデバイス、マルチプロセッサシステム、マルチプロセッサベースのまたはプログラム可能な家庭用電化製品、ネットワーク PC、ミニコンピュータ、メインフレームコンピュータなど、他のコンピュータシステム構成で実施され得ることが当業者には理解されよう。本発明はまた、通信ネットワークを介してリンクされたりリモート処理デバイスによってタスクが実行される分散コンピューティング環境で実施することもできる。分散コンピューティング環境では、プログラムモジュールは、ローカルおよびリモートの両方のメモリ記憶装置に置くことができる。

【 0 0 1 1 】

図 1 に示されるように、例示的な汎用コンピューティングシステムは、処理装置 2 1 と、システムメモリ 2 2 と、システムメモリを含むシステムの様々なコンポーネントを処理装置 2 1 に結合するシステムバス 2 3 とを含む従来型のパーソナルコンピュータ 2 0 などを含む。システムバス 2 3 は、メモリバスまたはメモリコントローラ、周辺バス、および様々なバスアーキテクチャのいずれかを使用したローカルバスを含めて、いくつかのタイプのバス構造のうちいずれのものでもよい。システムメモリは、読出し専用メモリ (ROM) 2 4 およびランダムアクセスメモリ (RAM) 2 5 を含む。ROM 2 4 には、例えば起動の際にパーソナルコンピュータ 2 0 内の要素間での情報の転送を助ける基本ルーチンを含む基本入出力システム (BIOS) 2 6 が格納されている。パーソナルコンピュータ 2 0 はさらに、図示されていないハードディスクを読み書きするためのハードディスクドライブ 2 7、取り外し可能な磁気ディスク 2 9 を読み書きするための磁気ディスクドライブ 2 8、CD-ROM や他の光媒体などの取り外し可能光ディスク 3 1 を読み書きするための光ディスクドライブ 3 0 を含むことができる。ハードディスクドライブ 2 7、磁気ディスクドライブ 2 8、および光ディスクドライブ 3 0 はそれぞれ、ハードディスクドライブインターフェース 3 2、磁気ディスクドライブインターフェース 3 3、および光ディスクドライブインターフェース 3 4 によってシステムバス 2 3 に接続されている。これらのドライブとそれぞれに関連するコンピュータ可読媒体は、パーソナルコンピュータ 2 0 のために、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用の不揮発性の記憶を提供する。ここで説明される例示的環境はハードディスク、取り外し可能磁気ディスク 2 9、および取り外し可能光ディスク 3 1 を利用しているが、例示的動作環境では磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌーイカートリッジ、ランダムアクセスメモリ (RAM)、リードオンリーメモリ (ROM) など、コンピュータがアクセスでき、データの格納が可能な他のタイプのコンピュータ可読媒体も使用できることが、当業者は理解すべきである。

【 0 0 1 2 】

ハードディスク、磁気ディスク 2 9、光ディスク 3 1、ROM 2 4 または RAM 2 5 には、オペレーティングシステム 3 5、1 つまたは複数のアプリケーションプログラム 3 6、他のプログラムモジュール 3 7、およびプログラムデータ 3 8 を含めて多くのプログラムモジュールを格納することができる。ユーザは、キーボード 4 0 およびポインティングデバイス 4 2 などの入力装置を介して、パーソナルコンピュータ 2 0 に命令および情報を

入力することができる。他の入力装置（図示せず）としては、マイクロホン、ジョイスティック、ゲームパッド、パラボラアンテナ、スキャナなどがある。これらおよび他の入力装置は、システムバスに結合されているシリアルポートインターフェース４６を介して処理装置２１に接続されることが多いが、パラレルポート、ゲームポート、ユニバーサルシリアルバス（ＵＳＢ）など他のインターフェースで接続することもできる。モニター４７または他の表示装置も、ビデオアダプタ４８などのインターフェースを介してシステムバス２３に接続されている。モニター４７に加え、パーソナルコンピュータは、通常スピーカやプリンタなど他の周辺出力装置（図示せず）を含む。図１の例示的システムはまた、ホストアダプタ５５と、小型コンピュータ用周辺機器インターフェース（ＳＣＳＩ）バス５６と、ＳＣＳＩバス５６に接続されている外部記憶装置６２をも含む。

10

【００１３】

パーソナルコンピュータ２０は、リモートコンピュータ４９などの１つまたは複数のリモートコンピュータへの論理接続を使用したネットワーク環境で動作することもできる。リモートコンピュータ４９は、他のパーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピアデバイスまたは他の共通ネットワークノードでもよく、図１にはメモリ記憶装置５０しか示されていないが、典型的にはパーソナルコンピュータ２０に関連して説明した上述の多くのまたは全ての要素を含む。図１に示す論理接続は、ローカルエリアネットワーク（ＬＡＮ）５１およびワイドエリアネットワーク（ＷＡＮ）５２を含む。これらのネットワーク環境は、オフィス、事業体規模のコンピュータネットワーク、イントラネットおよびインターネットにおいて一般的となっている。

20

【００１４】

ＬＡＮネットワーク環境で使用する場合、パーソナルコンピュータ２０はネットワークインターフェースまたはアダプタ５３を介してＬＡＮ５１に接続される。ＷＡＮネットワーク環境で使用する場合、パーソナルコンピュータ２０は、通常モデム５４、またはインターネットなどのワイドエリアネットワーク５２上での通信を確立するための他の手段を含む。モデム５４は、内蔵型または外付け型のどちらでもよく、シリアルポートインターフェース４６を介してシステムバス２３に接続されている。ネットワーク環境では、パーソナルコンピュータ２０に関連して示されたプログラムモジュールあるいはその一部を、リモートメモリ記憶装置に格納することもできる。図示されたネットワーク接続は例示的であり、コンピュータ間の通信リンクを確立するための他の手段も使用できることが理解されよう。

30

【００１５】

本発明の多くの実施形態は、コンピュータ化されたシステムに特に適していると考えられるが、本明細書には、本発明をこれらの実施形態に限定することを意図したものはない。逆に、本明細書では「コンピュータシステム」という用語は、電子的、機械的、論理的、またはバーチャルのうちのどの性質のものであるかに関わらず、情報の格納および処理が可能で、かつ／またはデバイス自体の挙動または実行を制御するために格納された情報を使用することができる全てのデバイスを包含するものであることが意図されている。

【００１６】

基本的な同期化

ピアトゥピアの同期環境およびその基礎を成すハードウェア／ソフトウェアインターフェースシステムがある。ここでは、とりわけ、格納可能な情報の自律ユニット（アイテム）が、同期化のための基本データ変更要素を構成するいくつかのコンポーネント変更ユニット（ＣＵ）を含む、ハードウェア／ソフトウェアインターフェースシステム（例えばオペレーティングシステムがあるが、ただしそれだけには限定されない）について説明する。各ＣＵは、同期化システムが、ピアコンピュータシステム上で更新される必要がある変更はどれなのかを追跡し、他のピアコンピュータシステムから受け取ったどの変更が適用されるべきかを追跡できるようにする、関連するメタデータを有する。

40

【００１７】

50

C U は、本質的に、常にユニットとして同期化される 1 組のプロパティである。例えば、C o n t a c t I t e m の F i r s t N a m e および L a s t N a m e プロパティは、一緒になって単一の変更ユニットを構成することができる。さらに、各変更ユニットには、その変更ユニットに特定の変更を行った最後のパートナーを表すための「バージョン」、およびその更新が行われた時点に対応するタイムスタンプが関連付けられる。したがって、本明細書では、説明を簡単にするために、パートナーは文字で表され、バージョンは逓増する整数で表され、それによって、単にパートナー（例えば「A」）およびタイムスタンプ（例えば「1」）を対にするだけで、ある特定の時点で各パートナーが各変更ユニットに行った最終の変更を反映する一意的な識別番号、例えば「A 1」がそのユニットに関連付けられる。

10

【0018】

同期化のために変更ユニットおよびメタデータ（または同様の構造）を使用するシステムにおける基本的な同期化の例示的な形態は、一般に 2 つのピアコンピュータシステム上に存在するアイテムのための単一の変更ユニットに関して説明することができる。この単一のチェンジユニットに関して、第 1 のピアであるパートナー A（あるいは、単に「A」）は、変更ユニットの A のレプリカ（C U a）に関連する A の現在のメタデータ（M a）を第 2 のピアであるパートナー B（あるいは、単に「B」）に送信することによって、B に変更を要求する。M a（C U a に対応）を受領後、パートナー B はこのメタデータを同じ変更ユニットの B のレプリカ（C U b）のための B 自体のメタデータ（M b）と比較する。次いで、M a に反映されていない C U b への変更を M b が反映している場合は、B は C U b と M b をいずれも A に送信し、そうでない場合は、C U a は既に C U b に加えられた変更の全てを反映しており、したがって C U b を B から A へコピーする必要がないため、B はそれ自体の M b のみを A に送信する（あるいは「更新なし」インジケータを A に送信する）。

20

【0019】

A は、それ自体が行った同期化要求への B の応答に基づき、B から M b のみ（あるいは「更新なし」インジケータ）を受信し、C U b を受信しない場合は、C U a が C U b と同じか、または C U b よりも新しく更新されていることを知る。しかし、A が B から M b と C U b を共に受信した場合は、A は M a を M b と比較することにより、C U b が C U a よりも新しく更新されているかどうか、または C U a と C U b とが競合する変更を含むかどうかを判断する必要がある。C U b が C U a よりも新しく更新されている場合、次いで A は C U a を C U b で更新し、M a を M b で更新する。一方、C U a と C U b とが競合する変更をもつ場合、A はそれ自体のローカル競合解消手順に従ってその競合を解消する。例えば、この場合の A の競合解消手順が、リモートソリューションよりもローカルなソリューションを選ぶことであった場合、A は引き続き C U a を保持（C U b の変更を拒否）し、M a および M b（M a と M b との和集合）に既に反映されている全ての更新を含むように M a を更新する。その後、B が次いで同じ手順を使用して A に同期化する場合、（介在する追加の変更がないと仮定して）A は C U a が C U b よりも新しく更新されていると判断し、C U b を C U a で更新する。

30

【0020】

この基本的な同期化スキームについて、各変更ユニットのメタデータは、変更ユニットの現在の「更新」状況を示す現バージョン（c v）と、ピアコミュニティにおける各パートナー（または少なくとも何らかの変更を加えたパートナー）の変更ユニットに行われた更新に対応するアウェアネスベクタ（a w a r e n e s s v e c t o r）（a v）（「バージョンの配列」の別名でも知られている）という 2 つのコンポーネントを含む。変更ユニットメタデータの典型的なフォーマットは＜変更バージョン、アウェアネスベクタ＞、またはより簡単には＜c v、[a v]＞である。これらの情報を一緒に使用して、例えば＜A 1 3、[A 1 3、B 1 1]＞のメタデータ（M a）をもつパートナー A 上の変更ユニット（C U a）が、＜B 1 1、[A 1 2、B 1 1]＞のメタデータ（M b）をもつパートナー B 上の変更ユニット（C U b）よりも新しく更新されていると容易に判断すること

40

50

ができる。というのは、各バージョン番号のタイムスタンプ部分が時間と共に徐々に増えることを考えれば、M aはM bに反映されていない変更（すなわちA 1 3）を含むからである。

【 0 0 2 1 】

スナップショットおよびファイルストリーム

ある種のハードウェア/ソフトウェアインターフェースシステムでは、格納可能な情報の自律ユニット（例えばアイテム）の一部は、2種類のもの、つまり構造化されたもの、または混合されたものとすることができる。構造化要素（S E）は、十分に構造化されたデータだけを含むが、混合要素（M E）は十分に構造化されたデータ、非構造化ファイルストリームのどちらも含む。図2 Aは、複数のプロパティ2 0 4を有するS E 2 0 2を示すブロック図である。このS Eは、構造化データだけを含み、したがって構造化データコンポーネント2 0 6を含む。図2 Bは、複数のプロパティ2 0 4を有するM E 2 1 2を示すブロック図であり、プロパティ2 0 4のうちの1つはM Eのファイルストリーム2 0 8へのリンク2 0 4'である。M E 2 1 2の構造化データコンポーネント2 0 6は、M Eの非構造化データコンポーネントを表す非構造化ファイルストリーム2 0 8と区別することができる。同期化に関して、S Eはある種のサイズ制限を満たし、必要に応じてコピーが作成できるほど十分に小さいため、同期化も簡単に行うことができる。一方、M Eは通常、アイテムの通常の限界を越えるデータ構造に使用され、そのサイズゆえに、当業者には自明な理由により、簡単にコピー、同期化などを行うことはできない。

【 0 0 2 2 】

ある種のハードウェア/ソフトウェアインターフェースシステムでは、アイテムの同期化は通常、同期化中にターゲットアイテムが変更されることを防ぎ、その代わりにターゲットアイテムの変更を望む他のプロセスにコピーを提供し、ターゲットアイテムの同期が完了した際に、このコピーでターゲットアイテムを更新する、トランザクション分離レベル（T I L）で実行される。このスナップショットT I L（S T I L）により、本質的に、ターゲットアイテムの同期化中に、ターゲットアイテムが他の同期ピアにコピーされる間にターゲットアイテム自体が変更されないように、同期化システムが、他のプロセスによる修正のために、必要に応じてターゲットアイテムのコピー（「スナップショット」）を作成することが可能になる。

【 0 0 2 3 】

図3は、S E（および、本明細書で後ほど議論される、M Eの構造化部分）を同期化するために、本発明のいくつかの実施形態で使用される方法を示したプロセスフロー図である。この図で、最初のステップ3 0 2で、変更ユニット（このケースではターゲットアイテム）の「スナップショット」を作成し、ステップ3 0 4で、同期化ピアへの変更ユニットのコピーが完了されつつあるいずれかの時点で他のいずれかのプロセスがこの変更ユニットに変更を加えようと試みる場合に、変更ユニットのコピーを作成する。コピー作成が完了すると、ステップ3 0 6でターゲットアイテムのコピーを使用してターゲットアイテムが更新（例えば、置換え）される。

【 0 0 2 4 】

スナップショットなしの同期化

S EはスナップショットT I Lを使用して同期化することができるが、多くの場合、いくつかの理由でM Eのファイルストリームのスナップショットは作成されない。最も明らかな理由は、ファイルストリーム自体のサイズの故に、また非常に大きいデータオブジェクトはその一時的コピーの作成が望ましくないことである。

【 0 0 2 5 】

本発明の様々な実施形態は、同期化中に大きな非構造化M Eファイルストリームのスナップショットの作成が可能ではないハードウェア/ソフトウェアインターフェースシステム環境におけるピアコンピュータシステム間でM Eを同期化する（特にM Eのファイルストリームコンポーネントを同期化する）システムおよび方法を対象としている。このような実施形態のいくつかでは、この方法はM Eの同期化を2ステップで行うことを含み、そ

の1つはMEの構造化データコンポーネント（および、ファイルストリーム自体ではなく、ファイルストリームのcv）のスナップショットを作成するステップであり、もう1つは、ファイルストリームをロックし転送するステップであって、ファイルストリームがロックされた後に、ファイルストリームがMEデータコンポーネントのスナップショット作成時以降改変されていないと判断された場合に、かつその場合に限り、ファイルストリームを、受け取る側の同期ピアに送信するステップである。

【0026】

図4は、MEを同期化する方法のステップを示すプロセスフロー図である。ステップ402で、特定の変更ユニット（MEアイテム）の同期化の始めに、スナップショットTIL（すなわち、同期化中はターゲットアイテムではなくそのコピーに変更が加えられ、次いで同期化完了後に変更が適用される）を使用してMEの構造化データコンポーネント（SDC）のスナップショットが作成され、ステップ404でこのスナップショットが取られた時点での対応するファイルストリーム現バージョン（cvF）（例えば、cvF値はMEの構造化データのスナップショット中のファイルストリームプロパティから直接取得することができる）が第1の値（cvF0）として記憶される。（MEのファイルストリームはそれ自体の変更ユニットを備え、したがってそれ自体の変更バージョンをもつ。この値はMEの構造化データ部分のファイルストリームプロパティの一部としても記憶してもよい。）次いで、ステップ406でファイルストリームがロックされ、ステップ408でcvFが再びチェックされて、第2の値（cvF1）が決定され、ステップ410でそれが第1の値（cvF0）と比較され、2つの値が同じである（すなわち、cvF1がcvF0よりも大きくなく、したがってファイルストリームが変更されていないことを示す）場合は、ステップ412でファイルストリームが同期ピアにコピーされ、コピーが完了すると、ステップ416でファイルストリームがアンロックされる。一方、cvF1がcvF0と同じでない（すなわち、cvF1がcvF0よりも大きい）場合は、MEの構造化データのスナップショットが作成されてそれ以降にファイルストリームが更新されたことを示し、この場合にはファイルストリームはコピーされず、ステップ414でファイルストリームのための変更ユニットは同期化の「例外」を、すなわち同期の失敗が起こったことを反映し、そのことが受け取る側の同期ピアに認識され、当然の成り行きとして、後で2つのピア間での次の同期化処理中にMEを同期化させる試みが行われ、ステップ416でファイルストリームが再びアンロックされる。

【0027】

前記に加えて、再び図4を参照すると、ステップ406でファイルストリームのロックが行えない場合、ステップ414で再び例外が起きるが、ステップ416でのアンロックは不要である（このロック失敗例外パスは図示されていない）。また、ロックは、読み込みのターゲットを生得的にロックする、例えばトランザクションリード動作などの一連の標準的関数呼出しを通じて取得することができる点も留意されたい。

【0028】

より大きな同期化の観点から、また本明細書に記載する実施形態のいくつかを利用すると、この方法を使用する同期化アルゴリズムは以下のように進行する。同期化システムはまず、通常の変更列挙プロセス（change enumeration process）を使用して同期レプリカのどのアイテム（つまり、変更ユニット）が変更されたかを識別する。この変更列挙はまた、アイテムがMEである場合、SDC（ただし、ファイルストリームではない）のスナップショットの作成と併せてファイルストリームの変更バージョンも読み出す。次いで、MEに対して、同期化システムはスナップショットトランザクションからのステップアウト後のある時点でファイルストリームをロックし、ロックが得られた後にファイルストリームの変更バージョンを読み出す。ファイルストリームの変更バージョンが、SDCのスナップショットが取られた時点から変化していない場合、ロックされているために読み出し/コピーを行ってもファイルストリームが修正されないのので、同期化システムがファイルストリームの内容を同期ピアに読み出すことを進めることは安全である。一方、ファイルストリームのcvの2回目の読み出しによってストリームの新

たな変更ユニットバージョンが取り出される場合、これは、プロセスがスナップショットランザクションからステップアウトした時点から、プロセスがファイルストリームをロックした時点までの間にファイルストリームが修正されたことを示す。このようなケースでは、この要素が同期化されず、後続の同期化処理で同期化される必要があることを同期パートナーに示す、「例外」を作成する。すなわち、ファイルストリームはこの特定の同期化パスでは送られないが、それが送られなかったことについて、後続の同期化プロセスの間にこのMEを同期化する別の試みが行われるのに十分な情報が維持される。

【0029】

結論

本明細書に記載した様々なシステム、方法および技術は、ハードウェアまたはソフトウェアで、あるいは適切な場合には、両者の組合せで実施することができる。したがって、本発明の方法および装置、またはそのいくつかの態様または部分は、フロッピー（登録商標）ディスク、CD-ROM、ハードドライブ、または他の任意の機械可読記録媒体などの有形媒体において実施され、それがコンピュータなどの機械にロードされて実行されるとき、その機械が本発明を実施するための装置となる、プログラムコード（すなわち、命令）の形態をとることができる。プログラム可能なコンピュータ上でプログラムコードを実行する場合、コンピュータは、一般にプロセッサ、プロセッサ可読の記録媒体（揮発性および不揮発性のメモリおよび/または記憶要素を含む）、少なくとも1つの入力装置、および少なくとも1つの出力装置を含む。好ましくは、1つまたは複数のプログラムが、コンピュータシステムと通信可能となるように、高水準の手続き型プログラミング言語またはオブジェクト指向プログラミング言語で実施される。しかし、必要に応じて、この（1つまたは複数の）プログラムは、アセンブリ言語または機械語で実施することができる。いずれの場合も、言語はコンパイラ型言語でもインタープリタ型言語でもよく、ハードウェア実装と組み合わせることができる。

【0030】

本発明の方法および装置はまた、電気配線またはケーブル、光ファイバあるいは他の任意の送信形態などなんらかの通信媒体を介して送信され、EPROM、ゲートアレイ、プログラム可能論理回路（PLD）、クライアントコンピュータ、ビデオレコーダなどの機械によって受け取られ、ロードされて実行されるとき、その機械が本発明を実施するための装置となる、プログラムコードの形態をとることができる。汎用プロセッサ上で実施される場合、プログラムコードはプロセッサと組み合わされて、本発明のインデックス作成機能（indexing functionality）を実行するように動作する固有の装置を提供する。

【0031】

本発明を様々な図に示した好ましい実施形態に関連して説明したが、本発明から逸脱することなく、本発明の同様の機能を実行するのに他の類似の実施形態を使用してもよく、あるいは説明した実施形態に改変または追加を行ってもよいことを理解されたい。例えば、本発明の例示的实施形態は、パーソナルコンピュータの機能をエミュレートするデジタル装置のコンテキストで説明してきたが、本明細書で説明したようなデジタル装置だけに限定されず、有線または無線に関わらずゲーム用コンソール、ハンドヘルドコンピュータ、ポータブルコンピュータなどの既存の、または将来の任意の数のコンピューティングデバイスまたは環境に適用することができ、通信ネットワークを介して接続され、ネットワークを介して対話する任意の数のそうしたコンピューティングデバイスに適用することができることが当業者には理解されよう。さらに、本明細書では、特にワイヤレスネットワーク接続された装置の数が増え続ける中で、ハンドヘルドデバイスオペレーティングシステムや他のアプリケーション固有のハードウェア/ソフトウェアインターフェースシステムを含む様々なコンピュータプラットフォームが企図されていることも強調しておきたい。したがって、本発明はどの単一の実施形態にも限定されるべきではなく、添付の特許請求の範囲による広さおよび範囲の中で解釈すべきである。

【図面の簡単な説明】

【 0 0 3 2 】

【図 1】本発明の諸態様を組み込むことができるコンピュータシステムを表すブロック図である。

【図 2 A】複数のプロパティを有する構造化要素（ S E ）を表すブロック図である。

【図 2 B】構造化データコンポーネント、非構造化ファイルストリームのいずれも含む混合要素（ M E ）を表すブロック図である。

【図 3】 S E （および、本明細書で前に議論された、 M E の構造化部分）を同期化するために、本発明のいくつかの実施形態と併せて使用される方法を示したプロセスフロー図である。

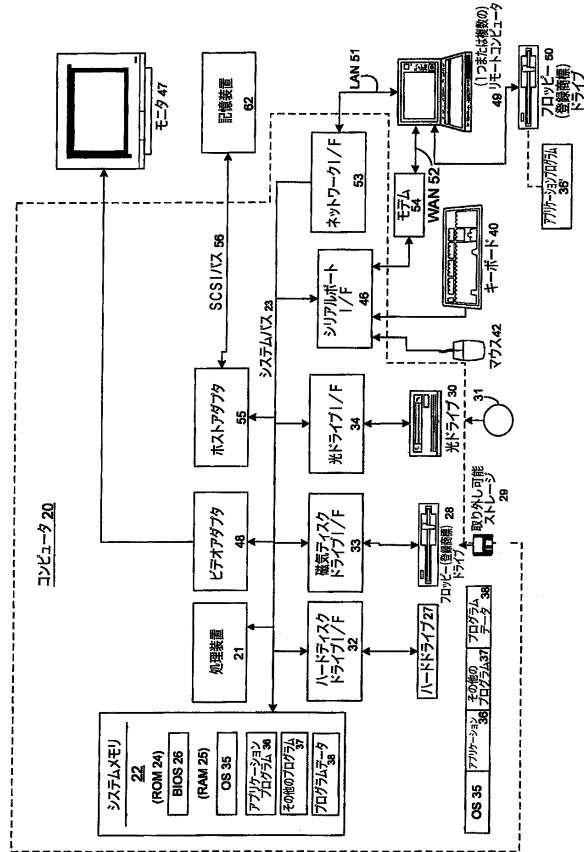
【図 4】本発明のいくつかの実施形態に利用される方法を表す、 M E を同期化する方法のステップを示したプロセスフロー図である。 10

【符号の説明】

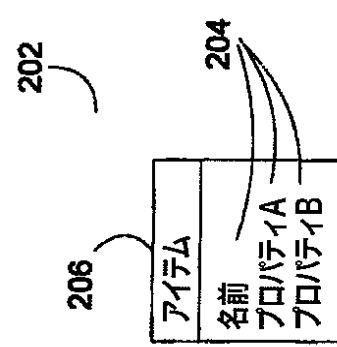
【 0 0 3 3 】

2 0	コンピュータ	
2 1	処理装置	
2 2	システムメモリ	
2 3	システムバス	
2 7	ハードドライブ	
2 8	フロッピー（登録商標）ドライブ	
2 9	取り外し可能ストレージ	20
3 0	光ドライブ	
3 2	ハードディスクドライブ I / F	
3 3	磁気ディスクドライブ I / F	
3 4	光ドライブ I / F	
3 6	アプリケーションプログラム	
3 6	アプリケーション	
3 7	その他のプログラム	
3 8	プログラムデータ	
4 0	キーボード	
4 2	マウス	30
4 6	シリアルポート I / F	
4 7	モニタ	
4 8	ビデオアダプタ	
4 9	（ 1 つまたは複数の ）リモートコンピュータ	
5 0	フロッピー（登録商標）ドライブ	
5 3	ネットワーク I / F	
5 4	モデム	
5 5	ホストアダプタ	
5 6	S C S I バス	
6 2	記憶装置	40

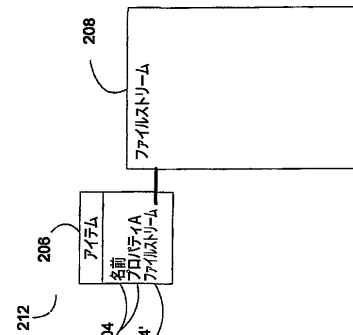
【図 1】



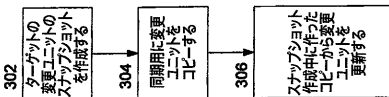
【図 2 A】



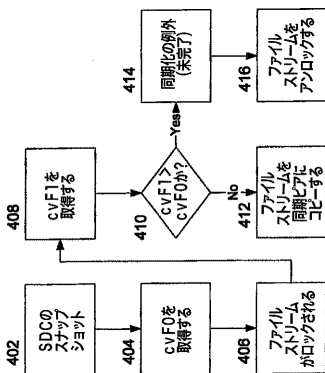
【図 2 B】



【図 3】



【図 4】



フロントページの続き

- (72)発明者 レフ ノビック
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 ラジェシュ エム・ラオ
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 サミート エイチ・アガーワル
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 コーポレーション内
- (72)発明者 ビベック ジャワヒー ジャベリ
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 桜井 茂行

- (56)参考文献 特開平6 - 214845 (JP, A)
特開2004 - 38977 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 12/00