

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5622590号
(P5622590)

(45) 発行日 平成26年11月12日(2014.11.12)

(24) 登録日 平成26年10月3日(2014.10.3)

(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 3 1 D

請求項の数 21 (全 31 頁)

(21) 出願番号	特願2010-545039 (P2010-545039)	(73) 特許権者	500046438
(86) (22) 出願日	平成21年1月6日(2009.1.6)		マイクロソフト コーポレーション
(65) 公表番号	特表2011-511368 (P2011-511368A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成23年4月7日(2011.4.7)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2009/030205		クロソフト ウェイ
(87) 国際公開番号	W02009/099690	(74) 代理人	100140109
(87) 国際公開日	平成21年8月13日(2009.8.13)		弁理士 小野 新次郎
審査請求日	平成23年12月28日(2011.12.28)	(74) 代理人	100075270
(31) 優先権主張番号	12/024,967		弁理士 小林 泰
(32) 優先日	平成20年2月1日(2008.2.1)	(74) 代理人	100101373
(33) 優先権主張国	米国 (US)		弁理士 竹内 茂雄
前置審査		(74) 代理人	100118902
			弁理士 山本 修
		(74) 代理人	100138759
			弁理士 大房 直樹

最終頁に続く

(54) 【発明の名称】 マルチマスター同期環境のための知識ベースフレームワークにおける質的オブジェクト変更の表現

(57) 【特許請求の範囲】

【請求項 1】

マルチマスター同期環境において1又は複数のネットワークを介して通信可能に結合された複数のノードの中の第1のノードと第2のノードとの間で1組のオブジェクトを同期させる方法であって、

与えられたデータ範囲について、前記第1のノードと前記第2のノードの間で、これら両方のノード上で表現されている1組のオブジェクトについての同期知識を交換するステップであって、前記第1のノードと前記第2のノードとの同期知識は、前記第1のノードと前記第2のノードとにおいてそれぞれ表現される前記1組のオブジェクトの中のオブジェクトの対応するバージョンを表す同期メタデータであって、オブジェクトの品質情報を含む同期メタデータを含み、前記同期知識の表現はデータ・タイプとは独立である、ステップと、

前記同期知識交換に基づいてオブジェクトが所望の質的特徴に変化するかどうかを確認するステップであって、前記所望の質的特徴は前記オブジェクトと関連する複数の潜在的質的特徴から識別された増分品質であり、前記オブジェクトは前記第1のノード又は前記第2のノードのいずれかである受信ノードによって受信可能である、ステップと、

前記受信ノードの前記同期メタデータを更新して、第1の変換が適用された前記オブジェクトの第1のバージョンの前記品質情報についての前記受信ノードによる第1の知識と、第2の変換が適用された前記オブジェクトの第2のバージョンの前記品質情報についての第2の知識とを示す変換メタデータを含むようにするステップと、

10

20

を含み、

前記更新は、前記第 1 のノードと前記第 2 のノードの前記品質情報に係る同期知識を比較して、前記 1 組のオブジェクトについての品質情報の集合的知識を決定するステップと、前記集合的知識に基づいて、同期を取るべきオブジェクトを選択するステップと、を含む、方法。

【請求項 2】

請求項 1 記載の方法であって、

少なくとも 1 つの事前定義された基準に基づいて、前記第 1 の変換が適用された前記オブジェクトが前記受信ノードにとって望ましいかどうかを決定するステップを更に含んでおり、前記決定するステップは第 1 の変換と第 2 の変換とを比較するステップを含む、方法。

10

【請求項 3】

請求項 2 記載の方法であって、

前記第 1 の変換が適用されるオブジェクトが前記受信ノードにとって望ましくない場合には、前記受信ノードによって前記オブジェクトを削除するステップを更に含む、方法。

【請求項 4】

請求項 3 記載の方法であって、

前記第 1 の変換が適用されるオブジェクトが前記受信ノードにとって望ましくない場合には、前記受信ノードの前記同期メタデータを更新することによって前記オブジェクトを無効化し、前記第 1 の変換が適用された前記オブジェクトのバージョンについて前記受信ノードは知っているが、前記マルチマスター同期環境における他のノードとの将来の知識交換の一部として前記バージョンを受信することを望まないことを反映するステップを更に含む、方法。

20

【請求項 5】

請求項 1 記載の方法において、前記同期メタデータを更新する前記ステップは、同じオブジェクトに対する前記第 1 の変換と前記第 2 の変換との間での差異に関する知識を示す変換メタデータを用いて前記同期メタデータを拡大するステップを更に含む、方法。

【請求項 6】

請求項 1 記載の方法において、前記同期メタデータを更新する前記ステップは、異なる第 1 の変換及び第 2 の変換の前記オブジェクトについての知識を示す変換メタデータであって、前記受信ノードが前記第 1 の変換が適用可能な前記オブジェクトを削除したか又は前記第 2 の変換が適用可能な前記オブジェクトを削除したかを示す変換メタデータを用いて前記同期メタデータを拡大するステップを更に含む、方法。

30

【請求項 7】

請求項 6 記載の方法において、前記同期メタデータを更新する前記ステップは、少なくとも部分的に、前記オブジェクトを削除する行為を記述する同期メタデータを維持することによって、そして、前記削除されたオブジェクトと同じ質的特徴を有する将来のオブジェクトとの関係で同期を拒否する規則を定義することによって、前記受信ノードが同一の又は異なるノードから前記削除されたオブジェクトを再び受け取る将来の知識交換を防止するステップを更に含む、方法。

40

【請求項 8】

請求項 6 記載の方法において、前記同期メタデータを拡大する前記ステップは、前記受信ノードが前記第 1 の変換を適用可能な前記オブジェクトを削除したのか又は前記第 2 の変換を適用可能な前記オブジェクトを削除したのかを示すブール値を用いて、前記オブジェクトについての前記同期メタデータを拡大するステップを含む、方法。

【請求項 9】

請求項 1 記載の方法において、前記交換するステップは、ローカル同期知識又は外部同期知識の一部として前記第 1 のノード又は前記第 2 のノードによって学習されたが、前記第 1 のノード又は前記第 2 のノード上で削除されたオブジェクトを考慮しながら、前記デ

50

ータ範囲の中で、前記第 1 のノード上で表現される前記 1 組のオブジェクトに関する前記第 1 のノードのローカル同期知識と前記第 2 のノードの外部同期知識とを比較するステップを含む、方法。

【請求項 10】

請求項 9 記載の方法であって、

前記比較するステップに基づいて、前記第 2 のノード上で表現される前記 1 組のオブジェクトについての前記外部知識に対する変更と、前記第 2 のノード上で表現される前記 1 組のオブジェクトに対する対応する変更とを決定するステップと、

前記外部知識に対する前記変更と前記 1 組のオブジェクトに対する前記対応する変更とを前記第 2 のノードに送信するステップと、
を更に含む、方法。

【請求項 11】

1 又は複数のネットワークを介して接続可能な複数のノードの中のノード・デバイスであって、マルチマスター同期環境において当該ノードと前記複数のノードの中の別のノードとの間で 1 組のオブジェクトを同期させるノード・デバイスにおいて、

当該ノードと前記複数のノードの中の前記別のノードとの間で前記 1 組のオブジェクトを同期させるように構成された同期コンポーネントを備えており、前記同期コンポーネントは、

同期プロトコルを介して前記別のノードとの同期を開始するように構成された同期通信コンポーネントであって、更に、データ・タイプとは独立に前記別のノードと当該ノードとの間での知識交換のためのメタデータ構造を定義し、前記同期プロトコルに基づいて前記別のノードに前記 1 組のオブジェクトとの同期の要求を送信し、前記別のノード上で表されている前記 1 組のオブジェクトに対応する別ノード・オブジェクト第 1 バージョン管理情報と前記別ノード・オブジェクト第 1 バージョン管理情報に対応する別ノード・オブジェクト第 2 バージョン管理情報とを含む前記要求に回答して前記 1 組のオブジェクトに関する外部知識を前記別のノードから受信するように構成されており、前記別ノード・オブジェクト第 2 バージョン管理情報は、前記別のノード・オブジェクト第 1 バージョン管理情報をバージョン管理し、前記別のノード・オブジェクト第 1 バージョン管理情報によって表される前記 1 組のオブジェクトのバージョンに対して異なる忠実度レベルを表す、同期通信コンポーネントと、

前記 1 組のオブジェクトの前記外部知識に基づいて、オブジェクトが前記別のノードにおいて所望の質的特徴に変化したかどうかを確認するように構成された同期分析コンポーネントであって、前記所望の質的特徴は前記オブジェクトと関連する複数の潜在的質的特徴から識別された増分品質であり、更に、前記 1 組のオブジェクトの外部知識が前記オブジェクトと関連する前記所望の質的特徴への変化を含むかどうかに基づいて、前記ノードにおいて表現されている前記 1 組のオブジェクトのローカルな知識と対応するノード・オブジェクト・バージョン管理情報とを更新するように構成されている同期分析コンポーネントと、
を含み、

前記同期コンポーネントは、前記ノードと前記別のノードの、前記オブジェクトの品質情報に係る同期知識を比較して、前記 1 組のオブジェクトについての品質情報の集合的知識を決定するように構成され、更に、前記集合的知識に基づいて、同期を取るべきオブジェクトを選択するように構成されている、
ノード・デバイス。

【請求項 12】

請求項 11 記載のノード・デバイスにおいて、対応するノード・オブジェクト第 1 バージョン管理情報と対応するノード・オブジェクト第 2 バージョン管理情報とを有する更新されたローカルな知識によって表された前記 1 組のオブジェクトの中のそれぞれのオブジェクトに対し、前記同期分析コンポーネントは、前記第 1 バージョン管理情報によって表されたオブジェクトの与えられた第 1 バージョンに対して前記ノードと前記別のノードと

10

20

30

40

50

において表されている第2のバージョン管理情報を比較することにより、前記第2のバージョン管理情報によって表された前記与えられた第1のバージョンの与えられた第2のバージョンが前記ノードと前記別のノードとのいずれか一方に同期するのに望ましいかどうかを判断するように構成されている、ノード・デバイス。

【請求項13】

請求項11記載のノード・デバイスにおいて、前記同期プロトコルは、前記ノードと前記別のノードとの間で同期されつつある実際のデータに関するどのようなスキーマも規定しない、ノード・デバイス。

【請求項14】

マルチマスター同期環境において1又は複数のネットワークを介して通信可能に結合されている複数のノードの中の第1のノードと第2のノードの間で1組のオブジェクトを同期させる方法であって、

前記第1のノードによって前記1組のオブジェクトの中の少なくとも1つのオブジェクトの質的特性を変更するステップと、

前記少なくとも1つのオブジェクトに関する前記第1のノードの同期知識メタデータを更新して、前記質的特性への前記変更を示す品質メタデータを含むようにするステップであって、前記同期知識メタデータの表現はデータ・タイプとは独立である、ステップと、

前記第2のノードに好まれる所望の質的特性に前記少なくとも1つのオブジェクトが変化したかどうかを確認するステップであって、前記所望の質的特性は前記少なくとも1つのオブジェクトと関連する複数の潜在的な質的特徴から識別された増分品質である、ステップと、

前記第1のノードによって前記第2のノードと同期するステップであって、前記第1のノードによって前記第1のノードの更新された同期知識メタデータを前記第2のノードに送信するステップを含んでおり、前記送信するステップは、前記第1のノードで表現された前記1組のオブジェクトのバージョンに対応する前記1組のオブジェクトの中のオブジェクトのバージョン・メタデータと、前記1組のオブジェクトの前記バージョンと関連する任意の対応する品質メタデータとを送信するステップを含む、ステップと、
を含み、

前記同期するステップは、前記第1のノードと前記第2のノードの前記品質メタデータを比較して、前記1組のオブジェクトについての品質情報の集合的知識を決定するステップと、前記集合的知識に基づいて、同期を取るべき少なくとも1つのオブジェクトを選択するステップと、を含む、
方法。

【請求項15】

請求項14記載の方法において、

前記更新するステップは、前記少なくとも1つのオブジェクトの前記質的特質の前記変更に応答して変更するステップを含む、方法。

【請求項16】

請求項14記載の方法であって、

前記変更するステップが前記第1のノードによって定義された少なくとも1つの基準に基づいて前記少なくとも1つのオブジェクトの前記質的特質を変更したかどうかを判断するステップを更に含む、方法。

【請求項17】

請求項16記載の方法において、前記判断するステップは、前記変更するステップが前記少なくとも1つのオブジェクトによって表されるデータを当該データの破壊を生じさせることなく変換したかどうかを判断するステップを含む、方法。

【請求項18】

請求項16記載の方法において、前記判断するステップは、前記変更するステップが前記少なくとも1つのオブジェクトによって表されるデータを当該データの忠実度に変化を生じさせることなく変換したかどうかを判断するステップを含む、方法。

【請求項 19】

請求項 18 記載の方法において、前記判断するステップは、前記変更するステップが前記少なくとも 1 つのオブジェクトによって表されるデータを当該データの忠実度を変換により低下させたかどうかを判断するステップを含む、方法。

【請求項 20】

請求項 18 記載の方法において、前記判断するステップは、前記変更するステップが前記少なくとも 1 つのオブジェクトによって表されるデータを当該データの忠実度を変換により上昇させたかどうかを判断するステップを含む、方法。

【請求項 21】

請求項 16 記載の方法において、前記第 1 のノードの前記同期知識メタデータを更新して前記質的特性に対する前記変更を示すようにするステップは、前記少なくとも 1 つのオブジェクトの現在のバージョン・メタデータを変更されないままに維持するステップと、前記少なくとも 1 つのオブジェクトの同じバージョンに対して品質的な差異を区別する追加的な品質バージョン・メタデータを前記現在のバージョン・メタデータと共に保存するステップとを含む、方法。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、マルチマスター同期環境 (multi-master synchronization environment) のための知識ベース同期フレームワーク (knowledge based synchronization framework) における、同期が取られたオブジェクト (複数可) に対する変更についての知識を記述するための拡張メタデータに関し、拡張メタデータは、変更についての質的情報 (qualitative information) を含む。

【背景技術】

【0002】

モバイルコンピューティングおよび通信デバイスの人気は、いつでもユーザーが望むときに情報を送受信できるようにしたいという対応する願望を生み出した。簡単に言えば、ユーザーは、いつでもどこにいても、またデバイス個々の能力がどうであれ、情報およびアプリケーションへの様々なデバイスからのユビキタスアクセスを望んでおり、加えて、ユーザーは、そのような情報に直に (on the fly) アクセスし、更新できることを望んでおり、データが可能な限り正確で最新であることの保証を望んでいる。

【0003】

デバイスおよびオブジェクトに互いにデータのレプリカを共有させることを試みる様々な分散データシステムが存在する。例えば、音楽共有システムは、PC、セルフォン、ゲーム機、および MP3 プレーヤーの間で音楽の同期を取ることができる。電子メールデータは、ワークサーバー、クライアント PC、およびポータブル電子メールデバイスの間で同期を取ることができる。しかし、今日、そのようなデバイスが互いに 1 組の共通情報の同期を取る限りにおいて、同期は一般に、静的設定に従ってデバイス間で取られる。しかし、これらのデバイスが頻繁にまたは断続的に切断される場合、すなわち、相手との通信から切断され得るようにデバイスが疎結合されている場合、例えば、セルフォンがトンネル内に入った場合、または同期が取られるデバイスの数が動的である場合、デバイスが、互いに再接続するとき、またはネットワークに参加するときに、他の各デバイスがどのような変更を必要とするかを決定するための、トポロジー独立な方法を有することが望ましくなる。

【0004】

図 1 に示されるように、今日では、電子メールサーバーが電子メールクライアントと同期を取る場合など、マスターノード 100 がクライアントノード 110 と専用方式で同期を取る様々な例が存在する。2 つのデバイス間の専用同期であるため、2 つのデバイス間で同期を取るために必要とされる情報 102 は、マスターノード 100 によって追跡する

10

20

30

40

50

ことができる。そのような情報 102 は、任意選択的に、クライアントノード 110 によっても同様に追跡することができるが、マスターノード 100 とクライアントノード 110 の間の接続が時々切断される場合、または同期デバイスの数が突然増加もしくは減少し得る場合、それらのデバイスのすべてにおいて各デバイスが必要とする共通な情報の必要な情報を追跡することは、難しい問題となる。

【0005】

現在の解決策は、しばしば、同期セマンティクス (synchronization semantics) を、任意のノードではなく、もっぱら特定のノード (例えば、電子メールサーバー) のクロックまたは論理透かし (logical watermark) に基づかせる。これらのシステムは、単一の接続ノードまたはマスターの場合には良好に機能し得る。しかし、ノード接続のトポロジーマたはパターンが予測不能に変化し得る場合、それらのシステムは、問題に打ち当たる。

10

【0006】

他のシステムは、問題に対処するために、特定の種類のデータオブジェクトのための独自仕様の同期モデルを構築し、すべてのデバイスにおいて、データフォーマットに固有な膨大な量のプリミティブメタデータを追跡する。例えば、特定の Word プロセッシング文書フォーマットのオブジェクトの同期を取るために、文書およびその基本プリミティブを表現することに、それらが時間とともに変化するにつれて、多くのオーバーヘッドおよび複雑さがもたらされ、共通の 1 組の Word プロセッシング文書に従って同期をとることを望む他のデバイスに対してその情報を効率的に表現することにも、多くのオーバーヘッドおよび複雑さがもたらされる。そのようなシステムは、構築が複雑になって多くのコストがかかり、それが基づいているカスタムデータフォーマットのせいで拡張可能性がない (non-extendible) のに加えて、生成し、分析し、追跡しなければならない大量のメタデータのせいで本質的に拡張可能性がない (unscaleable)。

20

【0007】

加えて、そのような解決策は、1つの特定の領域、例えば Word プロセッシング文書にのみ適用される。例えば、ピクチャ、ビデオ、電子メール、文書、データベースストアなど、すべての種類の同期オブジェクトが考慮される場合、マルチマスター環境内のすべてのデバイスにおいてそのようなオブジェクトの進化を追跡するために、各オブジェクトタイプに基づいたカスタム同期解決策を実施することは、現在のところ機能し得ないことが分かる。したがって、そのような解決策は、同期セマンティクスをデータセマンティクス (data semantics) と密接に結び付ける。

30

【0008】

したがって、トポロジ内のコンピューターが互いに接続する方法を変更した場合、またはコンピューターの数が増加するにつれて、ノード独立の同期知識が必要とされている。例えば、メディアプレーヤーの場合、多数のコンピューター間および多数のウェブサイト間で同期を取るのが望ましいことがある。たいていの場合、ほとんどのアプリケーションは、少数のよく知られたエンドポイント (ホーム PC およびメディアプレーヤー) の間でデータの同期を取ることができるにすぎない。しかし、メディアプレーヤーアプリケーションのユーザーにとって、デバイスコミュニティ (device community) は時間とともに進化するので、デバイスによって利用される音楽ライブラリーのためのデータ同期の柔軟性がますます必要となり、それによって、より堅牢なシステムの必要性が生み出される。

40

【0009】

その必要性は、起こり得る変更のタイプの観点から多くのコンピューティングオブジェクトが一枚岩的でないことを考えた場合、より一層複雑になる。今日の複雑なコンピューティング環境では、オブジェクト変更は、単なる「オブジェクトへのデータの追加」または「オブジェクトからのデータの除去」を越えた多様な変更を含む。オブジェクトのデータのいくつかに対する変更は、変更されるデータの除去と、変更を反映したデータの追加と考えることもできる。したがって、一枚岩的な見方の下では、オブジェクトの進化の観

50

点からオブジェクトに何が起こり得るかについて、限られた見方が、すなわち、オブジェクトによって表されるデータは、いくつかの新しいデータを用いて拡大でき、オブジェクトによって表されるデータの一部は、残りを同じに留めたまま変更でき、および/またはオブジェクトによって表されるデータの一部は、削除できる、という見方があるにすぎない。

【 0 0 1 0 】

これに関連して、オブジェクトに関する多数の属性のすべてでも変更することができる。例えば、オブジェクトの名前、オブジェクトが最後に閲覧されたとき、オブジェクトのフィールドの許容可能な長さ、オブジェクトのスキーマ（例えば、連絡先スキーマ）、オブジェクトの忠実度（例えば、オーディオ）、オブジェクトに適用される圧縮方式、オブジェクトの解像度（例えば、画像もしくはビデオ）、オブジェクトのファイルフォーマット、またはオブジェクトに対して定義できる他の任意の機能はすべて、オブジェクトのデータの追加または除去という単純な表現によっては適切に反映されない、オブジェクトの変更可能な属性の例である。

10

【 0 0 1 1 】

これに関連して、異なるデバイスは、同じオブジェクトの表現に関して、異なる要件または制限を有する。例えば、PC上でのオブジェクトの表現は一般に、メモリーが限定されたモバイル電話上でのオブジェクトの表現とは異なる要件に関連する。一例として、画像オブジェクトが与えられた場合、PC上では10Mbの原画像を表現する方がより適しており、一方、ディスプレイおよびメモリーがより小さなモバイル電話に対しては、同じ与えられた画像オブジェクトを300Kbの画像として表現する方がより適している。

20

【 0 0 1 2 】

いくつかの従来のシステムは、第1の品質から第2の品質への品質変化をこうむるオブジェクトを別個のオブジェクトとして扱い、すなわち、第1の品質に留まる第1のオブジェクトを維持するのに加えて、オブジェクトがこうむった変更を第2の品質に反映したまったく新しい第2のオブジェクトを生成する。結果として、両方のオブジェクトは、同期が取られて進行する。異なるオブジェクトの数、デバイスの数、および/または異なる品質のバージョンの数が少なからぬものになった場合、そのようなシステムが、禁止せざるを得ない、非常に無駄の多いものになることが分かる。

【 0 0 1 3 】

したがって、オブジェクトは複雑な方法で変更でき、特に、異なる能力の異なるデバイスにわたって同期が取られるので、同期メタデータ内においてそのような変更を効率的に表し、同時に、上で説明されたような疎結合されたマルチマスター同期環境において同期を扱うための方法が、現在望まれている。簡潔に言えば、オブジェクトが品質の変更をこうむった場合に、可能であるならば、デバイスの複雑なマルチマスターネットワークポロジにおける知識交換(knowledge exchange)において、オブジェクトの品質の減少または増加を表す能力が、無数の同期シナリオのために望まれる。

30

【 0 0 1 4 】

今日の同期モデルの上で説明された難点は、従来のシステムの問題のいくつかについての概要を提供することを意図したものにすぎず、網羅的であることは意図していない。従来のシステムが抱える他の問題、および本明細書で説明される様々な非限定的な実施形態の対応する利点は、以下の説明を検討することでより明らかとなるであろう。

40

【 発明の概要 】

【 0 0 1 5 】

本明細書では、より詳細な説明および添付の図面における以下の例示的で非限定的な実施形態の様々な態様の基本的または概略的な理解を可能にする助けとなるように、簡略化された要約が提供される。しかし、この要約は、広範なまたは網羅的な概要として意図されていない。代わりに、この要約の唯一の目的は、以下の様々な実施形態のより詳細な説明に対する前置きとして、いくつかの例示的で非限定的な実施形態に関連するいくつかの概念を簡略化された形式で提示することである。

50

【 0 0 1 6 】

本明細書では、オブジェクト品質の概念を含むように知識ベース同期フレームワークを拡張する、マルチマスター同期環境における複数のネットワークノード間の同期を提供する様々な実施形態が説明される。有利には、同期フレームワークによれば、エンドポイントは、知識交換の目的で、同期データの1つまたは複数のオブジェクトの品質の定義を可能にする方法で、データの同期を取ることができる。

【 0 0 1 7 】

一実施形態では、オブジェクトの与えられたバージョンについての知識ベクトル (knowledge vector) に、オブジェクトのそのバージョンについての増分品質情報を表す追加的な次元 (複数可) が置かれ、オブジェクトの品質のどのタイプが同期プロセスの一部としてノードに伝えられるべきかをノードが決定することを可能にするために、その次元が同期プロセス中に考慮される。例えば、ノードにとって望ましくない品質を有するオブジェクトは、マルチマスター同期環境において他のノードによって維持されるそのようなオブジェクトに関する将来の同期を回避するために無効化 (tombstone) することができ、他の実施形態では、オブジェクトに対する変更が非破壊的 (non-destructive) である (とノードによって決定された) 場合、オブジェクトの現在のバージョンは、現在のバージョンの別のバージョン、すなわち、変更されたオブジェクトの品質レベルを表すバージョンを反映するように拡張される。変更が破壊的 (destructive) と見なされた場合、同期メタデータによって、新しいバージョンが反映される。

【 0 0 1 8 】

上記および他の実施形態が、以下でより詳細に説明される。

【 0 0 1 9 】

様々な非限定的な実施形態が、添付の図面を参照してさらに説明される。

【 図面の簡単な説明 】

【 0 0 2 0 】

【 図 1 】 システムの 2 つの適切に定義されたエンドポイントの間で同期を提供する専用同期システムの図である。

【 図 2 】 同期オブジェクトについての品質情報を含む同期メタデータを含むマルチマスター同期のためのインフラストラクチャーの高水準ブロック図である。

【 図 3 】 ネットワークに接続するノードおよびネットワークから切断するノードの存在下で、品質同期メタデータに基づいて同期を取るための例示的で非限定的なプロセスを示すフロー図である。

【 図 4 】 品質同期メタデータに基づいて同期を取るための例示的で非限定的なプロセスを示す別のフロー図である。

【 図 5 】 ノードが疎結合されたネットワークのうちの 4 つのノードの間の例示的で非限定的な知識交換を示す図である。

【 図 6 】 デバイスのいくつかが互いに切断された場合の、ノードが疎結合されたネットワークのうちの 4 つのノードの間の例示的で非限定的な知識交換を示す図である。

【 図 7 】 ネットワークのノード間で多数のオブジェクトが共有される状況における例示的な知識交換を示す図である。

【 図 8 】 ネットワークのノード間で多数のオブジェクトが共有される状況における例示的な知識交換を示す図である。

【 図 9 】 ネットワークのノード間で多数のオブジェクトが共有される状況における例示的な知識交換を示す図である。

【 図 1 0 】 ネットワークのノード間で多数のオブジェクトが共有される状況における知識交換のためのプロセスを示す例示的で非限定的なフロー図である。

【 図 1 1 】 知識に基づいて変更を要求し、伝えるためのフレームワークを示す全体的なアーキテクチャーの図である。

【 図 1 2 】 質的情報を同期メタデータに含めるための第 1 の実施形態を示す全体的なフロ

10

20

30

40

50

一図である。

【図 1 3】無効化する実施形態が例示的な方式で示された第 1 の非限定的なシナリオを示す図である。

【図 1 4】質的情報を同期メタデータに含めるための代替的な実施形態を示す全体的なフロー図である。

【図 1 5】バージョンをバージョン管理する実施形態が例示的な方式で示された第 2 の非限定的なシナリオの図である。

【図 1 6】同期メタデータの一部として品質情報を伝達できる共通の 1 組の A P I を介して別のノードとの知識交換を実行するためのデバイスの例示的で非限定的な実施のブロック図である。

10

【図 1 7】本明細書で説明される様々な実施形態が実施できる例示的で非限定的なネットワーク環境を表すブロック図である。

【図 1 8】本明細書で説明される様々な実施形態の 1 つまたは複数の態様が実施できる例示的で非限定的なコンピューティングシステムまたは動作環境を表すブロック図である。

【発明を実施するための形態】

【 0 0 2 1 】

概要

背景技術において説明されたように、とりわけ、従来のシステムは、オブジェクトに生じたあらゆる変更毎にオブジェクトの異なるバージョンを生成するが、そのようなシステムは、無損失変換、トランスコーディング、トランスクリプション、または忠実度の変更など、オブジェクトに対する変更についての質的情報を獲得しない。例えば、デバイスが、望む質的バージョンのオブジェクトとのみ同期を取ることができ、オブジェクトによって拒否される質的バージョンを有するオブジェクトとは同期を取らないように、オブジェクトまたはオブジェクトの特定のバージョンの品質の変更についての情報を、同期言語 (`synchronization language`) 自体の一部として獲得することが望ましい。

20

【 0 0 2 2 】

したがって、様々な非限定的な実施形態では、マルチマスター同期環境内の異なるノードが、知識を比較でき、どのオブジェクトの同期を取り、どのオブジェクトは同期を取るのが望ましくないかを決定できるように、異なる忠実度、変換、符号化の、すなわち、異なる質的バージョンのオブジェクトは、マルチマスター同期環境において、オブジェクトの品質についての情報を獲得する方法で同期が取られる。

30

【 0 0 2 3 】

以降のロードマップについて一言すると、まず、本明細書で説明されるいくつかの実施形態の概要が提示される。その後、マルチマスターデータ同期システムにおいて知識を効率的に表現するための一般的な機構についてのいくつかの補足的なコンテキストが与えられる。次に、補足的なコンテキストについての例示的で非限定的な実施形態および特徴がより詳細に説明され、オブジェクトに対する変更についての質的情報を表現するためのマルチマスターデータ同期システムの代替実施形態が理解され、続いて、そのような実施形態を実施できる代表的なネットワークおよびコンピューティング環境が説明される。

40

【 0 0 2 4 】

これに関連して、様々な実施形態は、本明細書で知識と呼ばれる概念に基づいた同期のためのモデルを定義する、マルチマスター同期環境のノードのためのデータ同期フレームワークに関する。簡潔に言えば、知識は、同期が取られるオブジェクトから論理的に分離されて維持されるメタデータであり、デバイスが、最初に知識を交換し、次にオブジェクトに対するどの変更を互いに受け渡すべきかを効率的に列挙できるように、レプリカの状態ベース同期についての効率的な要約を定義する。有利には、各ノードは、オブジェクトのどの品質は同期を取るのが望ましく、オブジェクトのどの品質は同期を取るのが望ましくないかに関する独自のルールを定義することができる。例えば、デバイスは、どのバージョンが望ましいかを決定するためのテストを指定することができ、または異なる品質を

50

好ましさの順序にランク付けして並べることができる。

【 0 0 2 5 】

これに関連して、多くの場合、マルチマスター同期トポロジーの各デバイスが、同期が取られるオブジェクトの品質レベルについて何を知っているか（例えば、低解像度か、それとも高解像度か）、またはより一般的に、各デバイスが、デバイスによって維持されるオブジェクトに適用される任意の1つもしくは複数の変換について何を知っているかを効率的に表現する方法で、データの同期を取ることが有益である。この点において、個々のエンドポイントが、より低い品質のオブジェクト（またはオブジェクトの第1の変換）と同期を取るべきか、それともより高い品質のオブジェクト（またはオブジェクトの第2のより適切な変換）を有するデバイスを探し出すべきかを決定できるように、デバイスは、

10

【 0 0 2 6 】

一実施形態では、エンドポイントは、許容できない変換、符号化、忠実度などのバージョンを受信した場合は常に、その許容できないオブジェクトバージョンを無効化し、実質的にそのアイテムを同期プロセスから削除することができるが、将来の同期がその許容できないバージョンを再び受け入れることを回避するように、エンドポイントは、許容できないバージョンについてのメタデータを維持する。しかし、時間とともに、そのような無効化メタデータが、許容し得ないほど増加することがあり得る。要するに、デバイスは、特に機能が限定されており、おそらくは多くを拒否するデバイスの場合は、本当のところ、それまでに対話したが好まなかったものすべての記録を維持する必要はない。

20

【 0 0 2 7 】

このタイプの知識の効率的な表現を利用する他の実施形態では、ノード間での質的情報の共有の達成は、データのバージョンをバージョン管理することによって達成される。そのような実施形態は、2層の場合はバージョンのバージョンをバージョン管理し、3層の場合はバージョンのバージョンのバージョンをバージョン管理し、以降も同様にするによって、オブジェクトに適用される変換または品質レベルを任意の数のレベルまたは層にまで拡張することもできる。個々のエンドポイントは、オブジェクトに対する特定の変更が、バージョンのバージョン管理を必要としない一般的な変更（ティックカウント（tick count）の通常の増加、または新しいバージョンの適用）を表すか、それとも代わりに、現在のティックカウントによって表されるバージョンをバージョン管理することによって表される、忠実度または変換の変更を表すかを決定することが可能である。

30

【 0 0 2 8 】

どのオブジェクトが変化したかについての情報をバージョンを介して効率的に獲得し、表現する能力と、それに加えて、それらのオブジェクトがどのように変化したかについての情報を獲得し、表現する能力は、実質的に、エンドポイントに、オブジェクトの特定のバージョンがエンドポイントにとってどれだけ有益かを告げる。したがって、知識表現にバージョン情報のバージョン管理を含めることによって知識を拡張することは、オブジェクトに対する変換についての質的情報を伝える目的を達成するための別の方法である。

【 0 0 2 9 】

言及されたように、無効化する実施形態は、オブジェクトが望まないアイテムを受信した（例えば、デバイスのメモリーに対して解像度が高すぎる）ときに、デバイスが、そのアイテムを再び受信しないようにそのアイテムを無効化することができるが、アイテムについて知らされたが、そのアイテムを削除したという知識を伝えることができるように実施することもできる。したがって、様々な実施形態は、オブジェクトの異なる品質についての情報が同期知識の一部として定義されるように、オブジェクトの変換またはトランスコーディングの品質の概念を同期自体の言語に組み込む。

40

【 0 0 3 0 】

図2は、マルチマスター同期環境において同期を取るオブジェクトの概念を全体として図説するブロック図であり、オブジェクトは、オブジェクトのために定義された同期メタデータに従って異なる品質にある。示されるように、デバイス200およびデバイス21

50

0 は、ネットワーク 220（複数可）と、それぞれ同期コンポーネント 202、212 とを介して互いに接続し、同期を取るものとして示されている。各同期コンポーネント 202、212 は、ストレージ 204、214 内にオブジェクトを保存するばかりでなく、以下でより詳細に説明されるように、それぞれそれらのオブジェクトの同期知識 206、216 も維持する。これに関連して、データタイプおよびネットワークプロトコルとは独立に同期を取るために使用される同期知識 206、216 は、オブジェクトの品質を説明するメタデータを含むように拡大することができる。

【0031】

オブジェクトの品質を説明するメタデータの場合、これは、オブジェクトの同期のために、オブジェクトの異なる変換についての質的情報を獲得できることを意味する。例えば、図示されるように、同じオブジェクト X は、デバイス 200 およびデバイス 210 上で表現することができ、それぞれオブジェクト 230 および 230' として表現される。オブジェクト 230 は、品質 Q1 を有し、オブジェクト 230' は、品質 Q1 と異なる品質 Q2 を有する。したがって、様々な実施形態に関連して説明される知識に基づいたマルチマスター同期プロトコルによれば、同期知識 206、216 は、オブジェクトに対する変更のバージョン管理に加えて、オブジェクトの同じバージョンの異なる品質を説明する情報を含む。

【0032】

例えば、オブジェクト X 230 は、WMA フォーマットの歌とすることができ、一方、オブジェクト X 230' は、同じ歌とすることができ、AAC フォーマットとして符号化される。2つのオブジェクトをその歌の異なるバージョンとして表現する代わりに、本明細書の様々な実施形態は、それが同じバージョンであるが、そのバージョンに関連付けられた異なる品質が存在することを示す、知識表現を可能にする。したがって、ノードは同期メタデータ内に含まれる品質情報に基づいて、与えられた品質バージョンの同期を取るべきかどうかを決定することができる。

【0033】

歌が例として使用されたが、オブジェクトのタイプは関係がない。オブジェクトのバージョンが異なる質的特性を有すると考え得ることを宣言するのが有益である場合はいつでも、同じ技法を適用することができる。例えば、PC 上の Outlook 内の連絡先アイテムは、モバイル電話上の連絡先アイテムよりも多くのフィールドを含むことができる。前者は、後者よりも品質が高い連絡先のバージョンであると、ノードによって見なされ得る。そのような場合、本質的に、それは、連絡先アイテムによって表されるデータの量についての忠実度または解像度の変化である。したがって、実質的に、任意のアイテムは、例えば、異なる符号化または同じプログラムの異なるプログラミング言語トランスコーディングなど、同じオブジェクトの変換に関連する変化に分類することができる。したがって、本明細書の実施形態のいくつかは、概念の簡素化のため、オーディオまたはビデオの文脈において説明されるが、任意のタイプのオブジェクトが、品質変換をこうむると見なすことができ、各エンドポイントは、その目標のため、同期の観点からそれが何を意味するかを定義することができる。

【0034】

図 3 は、マルチマスター同期環境において様々なノード間で同期を取ることを目的とした、オブジェクトのための「品質」情報の使用を説明する、全体的なフロー図である。300 において、いくつかのポイントにおいて、異なる品質を有するオブジェクトのバージョンに対して、同期メタデータが定義される。310 において、ノードが、マルチマスター同期環境において、任意のネットワークプロトコルに従って構成された 1 つまたは複数のネットワークを介して、別のノードに接続する。320 において、ノードは、同期メタデータを学習すること、すなわち、別のノードから受信すること、もしくは別のノードに要求し、受信することができ、またはノードは、同期メタデータを別のノードに送信することができ、メタデータは、同期が取られる 1 組のオブジェクトのためのバージョン管理情報を説明し、オブジェクトに適用される変換の質的特性についての品質情報を含む。

【 0 0 3 5 】

3 3 0において、オブジェクトについての品質情報の集合的知識を決定するために、品質情報を含む2つのノードの同期メタデータが比較される。3 4 0において、1組のオブジェクトについての品質情報の集合的知識に基づいて、ノードは、どのオブジェクトの同期を取るべきかを選択することができる。3 5 0において、比較の後、任意選択的に、ノードは、ノードのために定義された品質基準を満たさない、1組のオブジェクトのうちのオブジェクトにアクションを取る（例えば、削除する）ことができるが、他のノードとの将来の効率的な知識交換のために、取られたアクションについて説明する同期メタデータを維持することもできる。

【 0 0 3 6 】

例えば、ノードは、オブジェクトの最高品質のバージョンのみを保持することを選択することができる。画像の場合、例えば、オブジェクトが5 M bバージョンを有する場合、3 0 0 K bバージョンは、ノード上での両バージョンのサポートを仮定したより低い解像度であるので、削除することができる。別のノードの場合、3 0 0 K bは、そのノードによってサポートされる最高の解像度であることがあり、そのため、ノードは、3 0 0 K bよりも高いオブジェクトの品質バージョンを拒否するように定義することができる。同様に、同じデータの異なる符号化は、互換性の観点から、異なるノードでは異なるように適用することができ、したがって、有利には、同期メタデータへの品質メタデータの包含は、取得できる異なる種類のオブジェクトについての各ノードがもち得る視力（*vision*）を改善し、同時に、どの品質バージョンが望ましくないかをノードが表現する仕組みを可能にする。

【 0 0 3 7 】

3 6 0において、任意選択的に、ステップ3 5 0に従って削除されたいくつかのオブジェクトは、同期メタデータ内で無効化することができ、それは、ローカルデバイス上でオブジェクトが削除され、ノードは過去にその品質を有するアイテムについて知らされたことがあるが、ノードは将来再びそのアイテムについて知らされたくないことを示す。一実施形態では、ノードは、拒否された品質を有するアイテムについての情報を維持することによって、アイテムを有することについて実質的に嘘をつくが、そのアイテムが将来望まれないことも通知する。例えば、そのようなオブジェクトについては、将来の知識交換の最中に同期知識を比較する目的で、オブジェクトがかつて学習されて拒否されたことを示すようにブールフラグ（*Boolean flag*）を設定することができる。

【 0 0 3 8 】

図4は、マルチマスター環境において同期を取るための知識交換フレームワークにおける、同期メタデータ内の品質情報の代表的な実施を示すフローチャート図である。示されるように、デバイス3 0 0とデバイス4 1 0は接続され、同期を取ることを望む。例を簡潔にするため、図4は、デバイス4 0 0とデバイス4 1 0の間で同期が取られる1つのオブジェクト4 0 2および4 1 2をそれぞれ示しているが、当該技法は、任意の1組のオブジェクトのために拡張することができる。言及されたように、本明細書で説明される実施形態によれば、同期知識4 0 4および4 1 4が、オブジェクト4 0 2および4 1 2のために、デバイス4 0 0および4 1 0によってそれぞれ維持される。これに関連して、同期知識4 0 4は、品質Q 1がオブジェクト4 0 2に適用されることを表し、同期知識4 1 4は、品質Q 2がオブジェクト4 1 2に適用されることを表す。

【 0 0 3 9 】

4 5 0に示されるように、ノードが、マルチマスター同期環境において、任意のネットワークトポロジーに従って構成された1つまたは複数のネットワークを介して、他のノードに接続した後、デバイス4 1 0は、デバイス4 1 0から同期知識4 1 4を受信することができる。4 5 2において、デバイス4 0 0は、以下でより詳細に説明される知識交換原理に従って、知識4 0 4を知識4 1 4と比較する。

【 0 0 4 0 】

これに関連して、4 5 4において、デバイス4 0 0は、同期知識4 1 4内で表されるバ

10

20

30

40

50

ージョンQ2が望ましいかどうかを決定することができる。望ましい場合、456において、ローカル知識404は、知識414の品質情報Q2を用いて更新される。次に458において、デバイス400は、デバイス400が関心のあるオブジェクトの更新を要求し、更新された知識404が、デバイス410に送信される。今度は、460において、デバイス410が、更新された知識404を知識414と比較し、462において、品質Q1を有するデバイス400のバージョンが望ましいかどうかを決定する。464において、知識414は、更新された知識414を含むように更新され、知識交換を完了させる。466において、更新された知識404が、Q2がデバイス400にとって望ましいことを示している場合、468において、デバイス400は、そのような更新を受信し、470において、それがデバイス400によって保存される。同様に、デバイス410は、望む変更をデバイス400から受信する。このようにして、同期知識交換は、デバイスが、同期中に受信したバージョンの品質について選り好みすることを可能にする。

10

効率的な知識表現および交換

様々な非限定的な実施形態による、マルチマスター同期環境における、知識として表現された同期メタデータを介するオブジェクトの品質の表現を説明する前置きとして、このセクションでは、データ同期フレームワークにおける、知識を効率的に表現するための全体的な機構の概要が提示される。

【0041】

全体的な同期機構は、(1)第1のノードによって必要とされる最低限のデータのみを送信するように第2のノードに要求することによる、接続デバイス間での知識の効率的な交換、(2)第1のノードと第2のノードの間でのデータの状態の不一致すなわち競合を効率的かつ正確に認識する能力、(3)任意の数のノードを同期させる能力、および(4)任意のノードを任意の他のノードを介して同期させる能力、すなわち、ピアツーピアのマルチマスター同期環境において動作する能力を含む。

20

【0042】

全体的な機構を用いて、2つのデバイス間で共有されるいくつかの情報に対して、任意の数の変更を行うことができる。2つのデバイスは接続されるといつでも、互いに知識を交換することによって、デバイス間での変更を容易にするための、互いが何を知っており、何を知らないかを再構成するために必要とされる少なくとも最低限の量の情報を知るようになる。3つ以上のデバイスが含まれる場合、共有される情報のより大きなベース(base)についての知識に関して、知識は不完全になり得るが、多数のデバイス間でより多くの知識が共有されるので、時間とともにデバイスが他のデバイスに接続するにつれ、集合的な知識がデバイスによって蓄積され続けることに留意されたい。

30

【0043】

有利には、様々な非限定的な実施形態では、同期は、1組のデバイスまたは1組のデバイスの一部に対して実行され、そのすべてが1組のオブジェクトの最新バージョンを維持することに関心があるが、同期は、そのようなデバイスが、コンピューティングエコシステム(computing ecosystem)においてデバイスが同期を取ることができる異なるオブジェクトについて質的結論を下すことも可能にする。デバイスが1つまたは複数のネットワークを介して1組のデバイスのうちの他のデバイス(複数可)と再び接続する場合はいつも、そのデバイスは、他のデバイス(複数可)がそれらの集合的知識を用いて表すのと同じくらい最新の集合的知識を再獲得する。このようにして、疎接続されたデバイスであっても、1組のデバイスと接触し、接触を断つことができ、その後、最新の1組の集合的知識を所有する任意の1組のデバイスと接触することによって、獲得し損ねたすべての知識を再学習することができる。

40

【0044】

図5は、知識交換が、任意の数のデバイスに一般化または拡張可能であることを示している。示されるように、非限定的な数のデバイスとして、4つのデバイス500、510、520、530が、デバイス間で共有される1組の共通情報について各デバイスが何を知っており、何を知らないかをそれぞれ示す知識表現502、512、522、532と

50

ともに示されている。

【0045】

有利には、図6に示されるように、ネットワーク内の接続が切断された場合でさえも、それに関わらず、他のデバイスへの少なくとも1つの接続が直接的または間接的に存在する限り、デバイス500、510、520、530のすべてによって、完全な1組の知識が獲得できる。例えば、示されるように、デバイス530の知識532は、デバイス520との知識交換、次にデバイス520とデバイス510の間の知識交換、最後にデバイス510とデバイス500の間の知識交換を介して、依然として、デバイス500に到達する。

【0046】

より多くのデバイスが、共有される共通情報についての知識を共有する場合、様々な限定的な実施形態による知識交換（複数可）は、集合的知識がどのデバイスから来るかについて関知しないので、デバイスのすべてが利益を得る。デバイスの各々は、それが接続された他のデバイスのいずれかから、デバイス間で共有される情報についての多くの知識を獲得しようと試みて独立に動作する。

【0047】

例示的で非限定的な詳細では、2つのノードが会話に関わり、会話の最後に、関心のあるデータセットについて同等の知識を有するための方法が、さらに詳細に説明される。この方法は、ピアツーピアネットワーク/マルチマスター環境に入って来た新しい各デバイスに対して知識交換能力を授けることによって、2つのノードを超えて拡張可能である。

【0048】

したがって、図7に示されるように、任意の数のノードを有するピアツーピアネットワークのノード700は、ノード710とデータを交換することを望む。ノードAは、ノード710に変更を要求することによって開始し、それを行うために、ノード700は、示されるように、(K_{N700} として表された)その知識をノード710に送信する。

【0049】

デバイスまたはノードの知識は、デバイス間で共有される各オブジェクトを文字識別子を用いてラベル付けすることによって表され、次に、後に付けられた数字が、このオブジェクトの最新バージョンを表す。例えば、図7に示されるように K_{N700} は、オブジェクトA、B、C、Dを含み、その各々は、ノード700とノード710の間で同期が取られ、各オブジェクトの後に付けられた数字は、デバイスにおいて知られているオブジェクトの最新バージョンを表す。例えば、時間 $t = 1$ における知識 K_{N700} は、図7においてA4、B3、C6、D0と表記される、Aの第5バージョン、Bの第4バージョン、Cの第7バージョン、Dの第1バージョンを含む。対照的に、時間 $t = 1$ におけるノード710の知識 K_{N710} は、図7においてA3、B6、C6、D2と表記される、Aの第4バージョン、Bの第7バージョン、Cの第7バージョン、Dの第3バージョンを含むことができる。

【0050】

図8に示されるように、時間 $T = 2$ において、ノード710は、ノード700から受信した知識 K_{N700} を自らの知識 K_{N710} と比較し、何がノード700に送信される必要があるかを決定する。この例では、ノード700の知識のB3、D0は、ノード710の知識のB6、D2よりも遅れているので、結果として、ノード710は、BおよびDに関する変更をノード700に送信する。ノード710は、B6とB3の間の変更およびD2とD0の間の変更をノード700に送信する場合、(ノード710上で最新の変更が行われたときがいつであろうとそれを反映した)ノード710が有する知識 K_{N710} の最新バージョンとともに送信する。

【0051】

時間 $t = 3$ を表す図9に示されるように、ノード700への知識 K_{N710} の送信は、同じバージョンであったにも関わらず、ノード700とノード710の両方がオブジェクトに対する変更を行ったことが後で判明した場合、ノード700が、競合を検出する(例

10

20

30

40

50

えば、後の解決のためにそれらを保存する)ことを可能にする。これは、自律的な更新、効率的な列挙を可能にするが、ノードが変更に出会い、それを交換した場合に、正しい競合検出も可能にする。例えば、例において、C 6 が知識 K_{N710} と知識 K_{N710} の双方で同じオブジェクトではない場合、例えば、双方が C 5 から C 6 に独立に進化した場合、どちらの C 6 が正しい C 6 であるかは、例えば、同期シナリオおよび含まれるデバイスに適した事前に定められた方針による解決に従った競合解決に委ねるために保留にすることができる。

【0052】

上で説明された全体的な機構を使用する、分散マルチマスター同期環境の任意の2つのノード間の例示的な知識交換プロセスが、図10のフロー図に示されている。1000において、ノードAは、ノードBとの同期を要求し、それによって、ノードAが知らない変更をノードBに求める。ノードBに授けるため、1010において、ノードAは、その知識をノードBに送信する。1020において、ノードBは、ノードAから受信した知識を自らの知識と比較して、ノードBが知っているどの変更をノードAに送信すべきかを決定する。1030において、ノードBは、そのような変更をノードAに送信し、加えて、ノードBは、ノードAが1040において同様の知識比較を実行できるように、その知識をノードAに送信する。本明細書で説明される実施形態と一貫性を保って、デバイスにとって望ましくないメタデータによる質的特性を有するオブジェクトは、デバイスの制限のせいで、またはそのような質的特性はデバイスが目標を促進する助けとならないので、同期が取られない。

【0053】

1050において、ノードAは、ノードAおよびノードB上でバージョンの独立した進化が発生している場合、ノードBの知識に反映された最新バージョンとノードAの知識に反映された最新バージョンの間の潜在的な競合を検出する。任意選択的に、競合の場合にどちらのノードが他方のノードに優先されるかを決定するために、いずれかの競合解決方針を適用することができる。1060において、ノードBによって所有されない最新の変更が、ノードAからノードBに送信される。競合解決方針が、ノード間で共通情報を維持するために、変更がノードBからノードAに送信されるか、それともノードAからノードBに送信されるかを追加的に指図する。独立したバージョン管理がOKまたは望ましい場合、競合解決は別の選択肢にならない。本明細書で説明される実施形態と一貫性を保って、デバイスにとって望ましくないメタデータによる質的特性を有するオブジェクトは、デバイスの制限のせいで、またはそのような質的特性はデバイスが目標を促進する助けとならないので、同期が取られない。

【0054】

図11は、フィルタリングされた知識が可能である場合に、すなわち、ノードの知識の一部が1つまたは複数の他のノードと同期が取られる場合に、知識を交換するための一般化された機構を示している。示されるように、各レプリカAおよびBは、同期プロバイダPAおよびプロバイダPBをそれぞれ有する。これに関連して、各レプリカAおよびBは、知識 K_A および K_B をそれぞれ維持し、潜在的には、フィルタリングされた知識 F_A および F_B も維持する。部分化(subsetting)のない場合と同様に、レプリカのいずれかが、別のレプリカの変更1100を要求することができ、変更を伝える他のレプリカに応答して、変更1110を受信することができる。図示されるように、レプリカAは、1100において、所定の範囲の1組のオブジェクトについての変更を要求することができ、組内のオブジェクトの品質についての情報を含むその知識を送信する。同様に、1110において、知識 K_A および K_B の分析に基づいて、1110において、レプリカBは知っているが、レプリカAは知らない変更が、品質の範囲内にあるオブジェクトについて、レプリカAに送信される。フィルタリングされた知識 F_A とフィルタリングされた知識 F_B が同じ範囲(scope)内にある場合、一般化された知識交換に関して、

$$K_A = K_A \quad K_B$$

となる。

【 0 0 5 5 】

フィルタリングされた知識 F_A とフィルタリングされた知識 F_B が同じ範囲内にない場合、

代わりに、知識は、以下のように、それぞれのフィルター F_A と F_B の共通部分に射影された他のレプリカの知識を既存の知識に加える関数である。

【 0 0 5 6 】

$$K_A = K_A \cup (K_B \cap (F_A \cup F_B))$$

アプリケーションの中でもとりわけ、これらのタイプのフィルターのための例示的で非限定的なアプリケーションは、カラム (column)、または同期フレームワークの任意の変更ユニット (change unit) をフィルタリングするためのものである。システムにおいてカラム変更は移動操作の影響を受ける可能性が低いので、これは特に適用可能である。フィルター表現 (filter representation) および知識統合 (knowledge consolidation) という、留意するに値する2つの検討対象が、このシナリオに関して存在する。

【 0 0 5 7 】

フィルター表現に関して、移動のないフィルターの場合のフィルター表現は、以下ようになる。各フィルターは、フィルター内に含まれる変更ユニットのリストとして表現される。この表現は、表現の便利な手段を提供するばかりでなく、必要な場合にフィルターを組み合わせる能力も提供する。フィルターを組み合わせる能力は、知識を統合するために有益である。

【 0 0 5 8 】

知識統合に関して、知識を最も簡潔な形式に保つため、知識を統合する能力が維持されなければならない。これに関連して、知識を最もコンパクトな形式に維持できるように、フィルタリングされた知識の断片 (fragment) を統合することができる。

【 0 0 5 9 】

フィルターを組み合わせる能力について検討すると、フィルターは1組の変更ユニットとして表現できるので、両方のフィルターに存在する変更ユニットの組を分離することによって、フィルター内の重複を照合調整することができる。

【 0 0 6 0 】

また、フィルターのためのベクトルは、フィルター内の個々の変更ユニットの各々に適用されるので、フィルターの組み合わせは、両方のフィルター内の各変更ユニットについて、変更ユニットのための合成ベクトルを見つけることによって、実行することができる。その後、ベクトルのすべてが知られると、共通のベクトルを有する変更ユニットが、新しいフィルター内に再度組み合わせられる。

【 0 0 6 1 】

フィルターは、デバイス間でどの情報の同期を取るかを制御するための1つの方法であるが、フィルターは一般に、以下でより詳細に説明される実施において提供されるように、アイテムの品質の特性を同期メタデータの一部として扱わない。

【 0 0 6 2 】

したがって、知識の概念は、マルチマスター同期ネットワークの多数のノード間での知識交換のため、データを効率的に表現するために使用することができ、ネットワークのいずれのノードも、ノード間で同期が取られる共通情報または共通情報の一部を独立して進化させることができる。上で説明された知識ベースフレームワークは、マルチマスター同期環境のために実施することができ、以下でより詳細に説明されるように、フレームワークは、効率的な同期メタデータを介したオブジェクトのバージョンの品質についての概念を含むように拡張可能である。

オブジェクトの品質の知識ベース表現

様々な実施形態では、マルチマスター同期トポロジにおけるデバイスによるデータの効率的な同期が提供され、デバイスは、同期が取られるオブジェクトの品質レベル (例えば、第1のオーディオフォーマットか、それとも第2のオーディオフォーマットか、第1

10

20

30

40

50

の解像度が、それとも第2の解像度が、もしくは第1のスキーマに準拠するのか、それとも第2のスキーマに準拠するのか)について知っており、またはより一般的に、各デバイスは、オブジェクトのバージョンに適用されるいずれか1つもしくは複数の変換について知っている。個々のエンドポイントが、第1の品質のオブジェクトのバージョン(もしくはオブジェクトのバージョンの第1の変換)と同期を取るべきかどうかを決定でき、または異なる品質を有するオブジェクトのバージョン(もしくはオブジェクトのバージョンの第2のより適切な変換)を有するデバイスを探し出せるように、デバイスは、知識交換の一部としてそのような情報を効率的に伝達することができる。

【0063】

どのオブジェクトが変化したかについての情報をバージョンを介して効率的に獲得および表現する能力と、それらのオブジェクトが質的にどのように変換されたかについての情報を獲得および表現する能力は、実質的に、オブジェクトの特定のバージョンがエンドポイントにとってどれだけ有益であるかをエンドポイントに告げ、オブジェクトが「変化した」または「異なる」という単なる事実よりも多くの情報が利用可能である。

【0064】

例えば、iPodは、すべての種類のオーディオファイルを消費することはできず、オーディオファイルの独自仕様フォーマットを再生するのに最もよく適している。対照的に、他のほとんどのMP3プレーヤーは、オーディオファイルのiPod独自仕様フォーマットを再生することができない。したがって、同期言語自体の一部として、異なるオーディオファイル符号化など、同じオブジェクトの異なる品質バージョンについて、知識交換の一部として同期を取ることが望ましいかを、デバイスに決定させることが望ましい。

【0065】

実施形態の第1の組では、無効化が実施され、それによって、オブジェクトが望まないアイテム(例えば、デバイスのメモリーに対して高すぎる解像度、またはデバイスによってサポートされない異なるスキーマに準拠するオブジェクトのバージョン、またはデバイスが符号化を復号できないなど)を受信した場合、デバイスは、デバイスがそのアイテムを再び受信することを望まないことを同期メタデータが反映するように、そのアイテムを無効化することができる。エンドポイントは、許容できない変換、符号化、忠実度などのバージョンを受信した場合、許容できないオブジェクトバージョンを無効化することができ、実質的に、そのデバイスとの同期プロセスからそのアイテムを削除する。

【0066】

アイテムは望まれていないので、デバイスによって削除することができ、または他の適切なアクションをオブジェクトに対して取ることができる。別の適切なアクションの例は、オブジェクトのバージョンがデバイスにとって不適切な符号化であるのに、デバイスがそのオブジェクトの復号器を有さない場合に、それにも関わらず、デバイスが、不適切な符号化をデバイスにとって理解可能な符号化にトランスコードするトランスコーダーを有することがある場合とすることができる。そのような場合、オブジェクトを削除する代わりに、デバイスは、オブジェクトを別の符号化に変換し、同期メタデータ内でその変換または新しい品質レベルを表すことができる。

【0067】

これに関連して、エンドポイントは、将来の同期が許容できないバージョンを再び受け入れることを回避するように、許容できないバージョンについてのメタデータを維持し続ける。しかし、時間とともに、そのような無効化メタデータが、許容し得ないほど増加することがあり得る。要するに、デバイスがそれまでに対話したが好まなかったものすべての記録を維持することは、特に機能が限定されており、多くのオブジェクトを拒否するデバイスの場合、またはめったに遭遇しない品質のオブジェクトの場合、効率的ではない。

【0068】

しかし、あるコンピューティング同期エコシステムの場合、そのような1組の実施形態は、例えば、デバイスが一般に互換性をもち、類似のアプリケーションならびに類似のメモリーおよび処理能力を有し、デバイスの各々が他のデバイスが知っているすべてを知り

10

20

30

40

50

たがる可能性が高いため、少数の拒否しかもたらされない可能性が高い同期環境において、きわめて良好に機能することがある。デバイスがハードウェアおよび/またはソフトウェア特徴においてより一層異質になるにつれて、デバイスが他のデバイスが知っているすべてを知りたがらない可能性がより高くなる。その状況は、異なる言語を話す人々と類似している。互いに異なる言語を話す人々の間では、人が他人が理解できる言語を話し始めるまで、多くのコミュニケーションの拒否が発生する。

【 0 0 6 9 】

この点において、オブジェクトのバージョンの品質の概念を、オブジェクト自体のソースであるデバイスに拡張することができる。例えば、時間とともに、デバイスは、特定のノードからの通信のほとんどまたはすべてが低いまたは非互換の品質にあり、したがって、多くが拒否されることを学習することができる。そのような場合、デバイスは、時間とともに、低い品質の別のデバイスと同期を取らず、そのデバイスにとってより関係のある知識を有する、より高い品質のデバイスを待つことを選択することができる。例えば、MP3プレーヤーは、iPodがiPod独自仕様フォーマットの歌のみを有することを学習した後は、iPodと同期を取らないことを選択することができ、デバイスが将来、iPodとの知識交換および比較を実行しなくて済むようにする。

【 0 0 7 0 】

言及されたように、オブジェクトのバージョンについての品質情報を表現する1つの方法は、デバイスに適さないオブジェクトのバージョンにデバイスが気づき、それを拒否した場合に、オブジェクトを無効化することである。多くの拒否が発生した場合、オブジェクトの数が劇的に増加するにつれて、各デバイスが将来何を受信したくないかについてのデバイス当たりに保存される情報の量が、問題を提示し得る。

【 0 0 7 1 】

無効化の実施形態が、全体として、図12のフロー図によって示されている。1200において、様々なノードが、多様なマルチマスター同期環境において、任意のトポロジカルフレームワークに従って、互いに接続することができる。1210において、2つ以上のノードが、知識交換を介して同期を開始する。220において、他のノードから受信したオブジェクトの各バージョンについて、各ノードは、オブジェクトのバージョンによって表される品質がノードにとって望ましいかどうかを決定する。1230において、望ましいかどうかに関わらず、遭遇したバージョンの知識を含むように、同期メタデータが更新される。1240において、バージョンが望ましい場合、そのバージョンは同期を取り続け、すなわち、オブジェクトのそのバージョンに関する同期は、将来の知識交換において変わることなく取られ続ける。1250において、オブジェクトが望ましくない場合、ノードが決定した場合、バージョンの知識は（将来の知識交換においてノードに対してそのオブジェクトの同期を取らないようにするために）維持されるが、1260において、オブジェクトの望ましくないバージョンは削除することができ、またはオブジェクトのそのバージョンに対して、変換などの他のアクションを取ることができる。

【 0 0 7 2 】

無効化の実施形態の動作が、全体として、図13に示されている。示されるように、最初、1302において、ノードA1300とノードC1320は、同期を取っており、両方のノードがサポートする高精細度オブジェクトO1を同期させている。しかし、1304において、ノードA100とノードB1310が同期を取るとき、1314において、ノードB1310は、オブジェクトの高精細度バージョンを保存できないと決定し、そのため、オブジェクトが削除またはより低い精細度バージョンに変換される。例では、簡略にするため、オブジェクトがノードB1310によって削除される。1312において、ノードBは、ノードB1310がオブジェクトのそのバージョンと再び同期を取ることが望まないことを反映するように、同期知識を拡大する。

【 0 0 7 3 】

有利には、その後、1322において、ノードC1320がノードB1310と同期を取るとき、高精細度オブジェクトO1は、ノードB1310がそのバージョンの非所望を

10

20

30

40

50

反映する知識を有するので、同期が取られない。しかし、1332において、後でノードD1330がノードB1310と同期を取るとき、低精細度バージョンには除外が存在しないので、ノードD1330は、オブジェクトO1の低精細度バージョンをノードB1310に同期させる。このようにして、各ノードは、時間とともに、異なる品質のデータの様々なバージョンに対する嫌忌を反映することができ、サポートされないアイテムに対する努力を繰り返して、不必要な同期トラフィックを生じさせることなく、より知的に同期を取ることができる。

【0074】

代替的な1組の実施形態では、新しい独立したバージョンに値する破壊的変更とは対照的に、品質は、データのバージョンが品質変換をこうむったことが決定されるデータのバージョンのバージョン管理によって、同期メタデータ内で表される。オブジェクトのバージョンに適用される変換または品質レベルの1つの層を表すためのバージョンのバージョン管理に加えて、そのような実施形態は、任意の数のレベルまたは層に、例えば、2層の場合のバージョンのバージョンのバージョン管理、3層の場合のバージョンのバージョンのバージョンのバージョン管理などに拡張することもできる。

【0075】

有利には、個々のエンドポイントは、オブジェクトに対する特定の変更が、新しいバージョンが生成される、例えば、ティックカウントが増加されるなど、バージョンのバージョン管理をトリガーしない一般的な変更を表すか、それとも代わりに、例えば、現在のティックカウントによって表されるオブジェクトのバージョンをバージョン管理するなど、オブジェクトの現在のバージョンをバージョン管理することによって表されるタイプの、忠実度または変換変更を表すかを決定することが可能である。したがって、バージョン情報のバージョン管理を知識表現に含めることによる知識の拡張は、オブジェクトに対する変換についての質的情報を伝える目的を達成する別の方法である。

【0076】

バージョン情報をバージョン管理するための例示的な実施形態が、図14のフロー図に示されている。1400において、1組のオブジェクトの一部として同期が取られるオブジェクトのバージョンの質的特性が変更される。1410において、同期メタデータは、例えば質的変更のためにバージョンがバージョン管理される、オブジェクトのバージョンに対する質的変更を反映した品質情報を含むように拡大される。品質情報が2層である場合、バージョンのバージョンがバージョン管理され、さらに層が増えた場合も同様である。1420において、ノードが、マルチマスター同期環境内の別のノードに接続する。

【0077】

1430において、ノードは、同期が取られる1組のオブジェクトのうちのオブジェクトのバージョンについての品質バージョン情報の交換を含む知識交換を介して同期を取る。1440において、各ノードは、知識交換に従って更新を列挙し、受信する。その後、オブジェクトの各バージョンについて、各ノードは、品質情報と、ノードが異なる質的特性を有するバージョンの様々なバージョンを望むかどうかを検討する。

【0078】

同じオブジェクトの異なる質的バージョンに対応するオブジェクトのバージョンの異なるバージョンを表す効率性が、図15のブロック図にさらに示されている。図15では、最初に、ノードA1500は、オブジェクトOの同じバージョンの異なる質的バージョンを有する。示されるように、O1Q1は、オブジェクトOの第1のバージョンを表すメタデータを示す表記であり、Q1は、オブジェクトO1のバージョンの第1の品質バージョンであることを示す。したがって、ノードA1500は、オブジェクトO1の3つの異なる質的バージョンQ1、Q2、Q3を含む。その後、T1において、T1におけるノードA1500とノードB1510の間の知識交換の後、ノードB1510は、バージョンQ1およびQ2のみを望むことを指定し、その後、それらの同期が取られる。その後、T2およびT3におけるさらなる同期において、オブジェクトC1520は、品質Q1を有するバージョンのみを望み、一方、オブジェクトD1530は、品質Q2を有するバージョンのみを望み、一方、オブジェクトE1540は、品質Q3を有するバージョンのみを望む。

ンのみを望む。したがって、異なる品質の表現は、オブジェクトの多くの不必要で無駄な同期を回避するために、マルチマスター同期環境においてどのバージョンの同期を取るべきかを決定する機会を各デバイスに与える。

【 0 0 7 9 】

図 1 6 は、1 組の A P I を介して完全または部分知識交換を実行するためのデバイス 1 6 0 0 の例示的で非限定的な実施のブロック図である。示されるように、デバイス 1 6 0 0 は、非限定的な実施形態による、別のデバイスと 1 組のオブジェクト 1 6 3 0 の同期を取るための知識交換技法を実行する同期モジュール 1 6 2 0 を含む。1 組のオブジェクト 1 6 3 0 は、効率的な動作のためにキャッシュ（図示されず）内に保存することもでき、その場合、1 組のオブジェクト 1 6 3 0 は、オフラインアプリケーションによって後で更新することができる。同期モジュール 1 6 2 0 は、本明細書で説明されたように、一般に知識交換技法に従ってデータを他のノードに送信し、他のノードから受信するための同期通信モジュール 1 6 2 2 を含むことができる。

【 0 0 8 0 】

同期通信モジュール 1 6 2 2 は、例えばオプションの承認モジュール 1 6 4 0 を介して承認された場合に、第 2 のデバイスとの同期を開始し、第 2 のデバイスに接続することができる同期開始モジュール 1 6 2 4 を含むこともできる。同期モジュール 1 6 2 2 は、同期の開始に応答して、例えば、知識を取得もしくは送信するため、または変更を取得もしくは送信するために、1 組のオブジェクト 1 6 3 0 についての完全および/または部分知識 1 6 0 2 を A P I を介して第 2 のデバイスに送信する I / O モジュール 1 6 2 6 を含むこともできる。同様に、I / O モジュール 1 6 2 6 は、第 2 のデバイスの要求した知識または変更 1 6 1 2、および第 2 のデバイスが起源の 1 組のオブジェクト 1 6 3 0 に施された変更を受信することができる。次に、同期分析モジュール 1 6 2 8 は、1 組のオブジェクト 1 6 3 0 に対して行われた任意の変更を適用するように、またローカルに行われた変更を決定するために、第 2 のデバイスから受信した知識 1 6 1 2 を第 1 のデバイスの知識 1 6 0 2 と比較するように、またはデバイス間の同期を完了するために第 2 のデバイスに送信するように動作する。

【 0 0 8 1 】

本明細書の実施形態によれば、図 5 から図 1 1 に関連して説明されたようなバージョン管理知識 1 6 0 3 などの、1 組のオブジェクト 1 6 3 0 についてのノードによって所有される知識 1 6 0 2 は、知識ベースフレームワーク内のオブジェクトに適用される変換についての質的情報を定義する品質知識 1 6 0 4 を含むように拡大される。

【 0 0 8 2 】

言及されたように、その概要が上で提供された知識フレームワーク内に含まれるメタデータを拡大することによる、オブジェクトのバージョンの知識ベース品質の様々な実施形態が、本明細書で提供された。疑念を避けるために言えば、本明細書で使用される「品質」という用語は、広い意味をもち、データに関する質的特性に影響を与える 1 つまたは複数の変換または機能に従ってデータを変換できる任意の方法を指す。したがって、品質という用語は、しばしば、文脈固有またはアプリケーション固有である。この点について、上で説明されたように、知識交換の一部として交換される同期メタデータ内にオブジェクトの異なる質的バージョンの概念を含めることによって、各エンドポイントは、コンピューティングシステム内で同期を取るようになった場合に、異なるバージョンにどう対処するかを決定することができる。

【 0 0 8 3 】

同様に、「変換」という用語および「トランスコーディング」という用語は、データを変更するための任意の機能を指す。加えて、「解像度」または「忠実度」の変更という用語は、広く、細部をより多くまたは少なく含むように任意のオブジェクトを変更することを指す。マルチマスター同期環境において複数のネットワークノード間の同期を提供する様々な実施形態が本明細書で説明され、それらは、同期オブジェクト（複数可）の質的変換の概念を含むように知識ベース同期フレームワークを拡張する。

【 0 0 8 4 】

したがって、様々な実施形態の拡張メタデータ表現は、エンドポイントが、オブジェクトの同じバージョンの異なる質的バージョンについてノードと他のノードの間で同期を取ることが望ましいかどうかをノードが決定できるように、オブジェクトに適用される変更がオブジェクトを質的に変換するかどうかを含むオブジェクトに関する知識を効率的に学習し、伝えることを可能にする。その場合、知識交換は、関与するノードにとってオブジェクトのバージョンのどの品質が最も望ましいかについての分析を含むことができ、オブジェクトの望ましくない品質バージョンを、オブジェクトのそれらの品質バージョンを受信することを望まない他のノードに送信する不必要な反復を防止することができる。

例示的なネットワークおよび分散環境

10

本明細書で説明された同期インフラストラクチャーの様々な実施形態は、コンピューターネットワークの一部としてまたは分散コンピューティング環境内に展開でき、任意の種類のデータストアに接続できる、任意のコンピューターまたは他のクライアントもしくははサーバーデバイスとの関連で実施できることを当業者であれば理解できよう。これに関連して、本明細書で説明された様々な実施形態は、任意の数のメモリーまたはストレージユニット、ならびに任意の数のストレージユニットにわたって発生する任意の数のアプリケーションおよびプロセスを有する、任意のコンピューターシステムまたは環境において実施することができる。これは、リモートまたはローカルストレージを有する、ネットワーク環境または分散コンピューティング環境内に展開されるサーバーコンピューターおよびクライアントコンピューターを有する環境を含むが、それに限定されない。

20

【 0 0 8 5 】

分散コンピューティングは、コンピューティングデバイスおよびシステム間の通信交換によって、コンピューターリソースおよびサービスの共用を提供する。これらのリソースおよびサービスは、情報の交換、ファイルなどのオブジェクトのキャッシュストレージおよびディスクストレージを含む。これらのリソースおよびサービスは、負荷バランシング、リソースの拡張、および処理の専門化などのための、多数の処理ユニットにわたる処理能力の共用も含む。分散コンピューティングは、ネットワーク接続性を利用し、クライアントが、それらの集団的能力を利用して、企業全体を益することを可能にする。これに関連して、様々なデバイスは、本開示の様々な実施形態のために説明されたような同期インフラストラクチャーを使用できる、アプリケーション、オブジェクト、またはリソースを有することができる。

30

【 0 0 8 6 】

図 17 は、例示的なネットワークまたは分散コンピューティング環境の概略図を提供している。分散コンピューティング環境は、アプリケーション 1730、1732、1734、1736、1738 によって表されるような、プログラム、メソッド、データストア、プログラマブルロジックなどを含むことができる、コンピューティングオブジェクト 1710、1712 など、およびコンピューティングオブジェクトまたはデバイス 1720、1722、1724、1726、1728 などを含む。オブジェクト 1710、1712 など、およびコンピューティングオブジェクトまたはデバイス 1720、1722、1724、1726、1728 などが、PDA、オーディオ/ビデオデバイス、モバイル電話、MP3 プレーヤー、パーソナルコンピューター、ラップトップなど、異なるデバイスを含み得ることは理解できよう。

40

【 0 0 8 7 】

各オブジェクト 1710、1712 など、およびコンピューティングオブジェクトまたはデバイス 1720、1722、1724、1726、1728 などは、通信ネットワーク 1740 によって直接的または間接的に、1 つまたは複数の他のオブジェクト 1710、1712 など、およびコンピューティングオブジェクトまたはデバイス 1720、1722、1724、1726、1728 などと通信することができる。図 17 では単一の要素として示されているが、ネットワーク 1740 は、図 17 のシステムにサービスを提供する他のコンピューティングオブジェクトおよびコンピューティングデバイスを含むこと

50

ができ、ならびに／または図示されていない多数の相互接続ネットワークを表すことができる。各オブジェクト1710、1712など、または1720、1722、1724、1726、1728などは、本開示の様々な実施形態に従って提供される同期インフラストラクチャーとの通信または同期インフラストラクチャーの実施に適した、API、または他のオブジェクト、ソフトウェア、ファームウェア、および／もしくはハードウェアを利用できる、アプリケーション1730、1732、1734、1736、1738などのアプリケーションも含むことができる。

【0088】

分散コンピューティング環境をサポートする様々なシステム、コンポーネント、およびネットワーク構成が存在する。例えば、コンピューティングシステムは、有線または無線システムによって、ローカルネットワークまたは広域分散ネットワークによって、互いに接続することができる。現在、多くのネットワークが、広域分散コンピューティングのためのインフラストラクチャーを提供し、多くの異なるネットワークを包含する、インターネットに結合されるが、任意のネットワークインフラストラクチャーが、様々な実施形態で説明されたような同期インフラストラクチャーに付随する例示的な通信のために使用できる。

【0089】

したがって、クライアント／サーバー、ピアツーピア、またはハイブリッドアーキテクチャーなど、多数のネットワークトポロジーおよびネットワークインフラストラクチャーが利用できる。「クライアント」は、それが関係しない別のクラスまたはグループのサービスを使用する、クラスまたはグループのメンバーである。クライアントは、別のプログラムまたはプロセスによって提供されるサービスを要求するプロセス、すなわち、大雑把に言って1組の命令またはタスクとすることができる。クライアントプロセスは、他のプログラムまたはサービス自体についてのいかなる作業詳細も「知る」必要なしに、要求したサービスを利用する。

【0090】

クライアント／サーバーアーキテクチャー、特にネットワークシステムでは、クライアントは通常、別のコンピューター、例えばサーバーによって提供される共用ネットワークリソースにアクセスするコンピューターである。図17の図では、非限定的な例として、コンピューター1720、1722、1724、1726、1728などが、クライアントと考えることができ、コンピューター1710、1712などが、サーバーと考えることができ、サーバー1710、1712などは、クライアントコンピューター1720、1722、1724、1726、1728などからのデータの受信、データの保存、データの処理、クライアントコンピューター1720、1722、1724、1726、1728などへのデータの送信などの、データサービスを提供するが、状況に応じて、任意のコンピューターをクライアント、サーバー、またはその両方と見なすことができる。これらのコンピューティングデバイスのいずれかは、1つまたは複数の実施形態のために本明細書で説明されたような同期インフラストラクチャーに関連し得る、処理データ、同期または要求サービスまたはタスクとすることができる。

【0091】

サーバーは一般に、インターネットまたはワイヤレスネットワークインフラストラクチャーなどのリモートまたはローカルネットワークを介してアクセス可能なリモートコンピューターシステムである。クライアントプロセスは、第1のコンピューターシステムにおいてアクティブとすることができ、サーバープロセスは、第2のコンピューターシステムにおいてアクティブとすることができ、それらは通信媒体を介して互いに通信し、したがって、分散機能を提供し、多数のクライアントがサーバーの情報収集能力を利用することを可能にする。同期インフラストラクチャーに従って利用される任意のソフトウェアオブジェクトは、スタンドアロンで提供することができ、または多数のコンピューティングデバイスもしくはオブジェクトにわたって分散することができる。

【0092】

通信ネットワーク/バス 1740 がインターネットであるネットワーク環境では、例えば、サーバー 1710、1712 などは、ハイパーテキスト伝送プロトコル (HTTP) など、多くの知られたプロトコルのいずれかを介して、クライアント 1720、1722、1724、1726、1728 などが通信するウェブサーバーとすることができる。サーバー 1710、1712 などは、分散コンピューティング環境に特徴的なように、クライアント 1720、1722、1724、1726、1728 などとして機能することもできる。

例示的なコンピューティングデバイス

言及されたように、有利には、本明細書で説明された技法は、コンピューティングシステムにおいて他のオブジェクトと同期を取るのが望ましい任意のデバイスに適用することができる。したがって、ハンドヘルド、ポータブル、および他のコンピューティングデバイス、ならびにすべての種類のコンピューティングオブジェクトが、様々な実施形態と関連して、すなわち、デバイスが同期を取ることができるどこでも、使用されることが企図されていることを理解されたい。したがって、以下図 18 で説明される以下の汎用リモートコンピューターは、コンピューティングデバイスの単なる一例にすぎない。

【0093】

必ずしも必要ではないが、実施形態は、デバイスもしくはオブジェクト用のサービスのデベロッパーによって使用するために、オペレーティングシステムを介して部分的に実施することができ、および/または本明細書で説明された様々な実施形態の 1 つもしくは複数の機能態様を実行するように動作するアプリケーションソフトウェア内に含まれることができる。ソフトウェアは、クライアントワークステーション、サーバー、または他のデバイスなど、1 つまたは複数のコンピューターによって実行される、プログラムモジュールなどのコンピューター実行可能命令という一般的な文脈において説明することができる。コンピューターシステムが、データを伝達するために使用できる様々な構成およびプロトコルを有し、したがって、特定の構成またはプロトコルを限定的なものに見なすべきではないことを当業者であれば理解されよう。

【0094】

したがって、図 18 は、本明細書で説明された実施形態の 1 つまたは複数の態様を実施できる適切なコンピューティングシステム環境 1800 の一例を示しているが、上で明確にしたように、コンピューティングシステム環境 1800 は、適切なコンピューティング環境の一例にすぎず、使用または機能の範囲に関していかなる限定も暗示しようとは意図していない。コンピューティング環境 1800 は、例示的な動作環境 1800 内に示されたコンポーネントのいずれか 1 つまたは組み合わせに関して、何らかの依存性または必要性を有すると解釈されるべきではない。

【0095】

図 18 を参照すると、1 つまたは複数の実施形態を実施するための例示的なリモートデバイスは、コンピューター 1810 の形態をとる汎用コンピューティングデバイスを含む。コンピューター 1810 のコンポーネントは、限定することなく、処理ユニット 1820、システムメモリー 1830、およびシステムメモリーを含む様々なシステムコンポーネントを処理ユニット 1820 に結合するシステムバス 1822 を含むことができる。

【0096】

コンピューター 1810 は一般に、コンピューター 1810 によってアクセスできる任意の利用可能媒体とすることができる様々なコンピューター可読媒体を含む。システムメモリー 1830 は、リードオンリーメモリー (ROM) および/またはランダムアクセスメモリー (RAM) などの揮発性および/または不揮発性メモリーの形態をとるコンピューター記憶媒体を含むことができる。限定することなく例を挙げると、メモリー 1830 は、オペレーティングシステム、アプリケーションプログラム、他のプログラムモジュール、およびプログラムデータも含むことができる。

【0097】

ユーザーは、入力デバイス 1840 を介してコンピューター 1810 にコマンドおよび

10

20

30

40

50

情報を入力することができる。モニターまたは他のタイプの表示デバイスも、出力インタフェース 1850 などのインタフェースを介してシステムバス 1822 に接続される。モニターに加えて、コンピューターは、出力インタフェース 1850 を介して接続できる、スピーカーおよびプリンターなどの他の周辺出力デバイスも含むことができる。

【0098】

コンピューター 1810 は、リモートコンピューター 1870 などの 1 つまたは複数の他のリモートコンピューターへの論理接続を使用して、ネットワークまたは分散環境において動作することができる。リモートコンピューター 1870 は、パーソナルコンピューター、サーバー、ルーター、ネットワーク PC、ピアデバイスもしくは他の共通ネットワークノード、または他の任意のリモート媒体消費もしくは伝送デバイスとすることができ、コンピューター 1810 に関して上で説明された要素のいずれかまたはすべてを含むことができる。図 18 に示される論理接続は、ローカルエリアネットワーク (LAN) またはワイドエリアネットワーク (WAN) などのネットワーク 1872 を含むが、他のネットワーク/バスを含むこともできる。そのようなネットワーク環境は、家庭、オフィス、企業規模のコンピューターネットワーク、イントラネット、およびインターネットにおいて一般的である。

10

【0099】

上で言及されたように、例示的な実施形態が様々なコンピューティングデバイスおよびネットワークアーキテクチャーに関連して説明されたが、基礎をなす概念は、任意のネットワークシステム、および同期を取ることが望ましい任意のコンピューティングデバイスまたはシステムに適用することができる。

20

【0100】

また、アプリケーションおよびサービスが同期インフラストラクチャーを使用することを可能にする、例えば、適切な API、ツールキット、ドライバーコード、オペレーティングシステム、コントロール、スタンドアロンまたはダウンロード可能ソフトウェアオブジェクトなど、同一または類似の機能を実施するための多数の方法が存在する。したがって、本明細書の実施形態は、API (または他のソフトウェアオブジェクト) の観点からばかりでなく、同期機能を提供するソフトウェアまたはハードウェアオブジェクトからも企図される。したがって、本明細書で説明された様々な実施形態は、全部がハードウェアである態様、部分的にハードウェアで、部分的にソフトウェアである態様、およびソフトウェアの態様を有することができる。

30

【0101】

「例示的 (exemplary)」という語は、本明細書では、例、例証、または実例として役立つことを意味するために使用される。疑念を避けるために言えば、本明細書で開示される主題は、そのような例によって限定されない。加えて、「例示的」として本明細書で説明された任意の態様または設計は、他の態様または設計よりも好ましいまたは有利であると解釈される必要はなく、当業者に知られる等価の例示的な構造および技法を排除することを意味するものでもない。さらに、「含む (includes)」、「有する (has)」、「含む (contains)」という用語、および他の類似の語が詳細な説明または特許請求の範囲において使用される限り、疑念を避けるために言えば、そのような用語は、任意の追加要素または他の要素を排除しない開かれた移行語としての「含む (comprising)」という用語と類似の仕方では包含的であることを意図している。

40

【0102】

「限定された品質」という用語は、オブジェクトの存在の開始および/または終了が制限されるような、同期システムにおけるオブジェクトの存在に対する制限を指す。

【0103】

言及されたように、本明細書で説明された様々な技法は、ハードウェアもしくはソフトウェアに関連して、または適切な場合には両方の組み合わせに関連して実施することができる。本明細書で使用される「コンポーネント」および「システム」などの用語も同様に

50

、ハードウェア、ハードウェアとソフトウェアの組み合わせ、ソフトウェア、または実行中ソフトウェアといったコンピューター関連エンティティを指すことを意図している。例えば、コンポーネントは、限定することなく、プロセッサ上で実行中のプロセス、プロセッサ、オブジェクト、実行可能ファイル、実行スレッド、プログラム、および/またはコンピューターとすることができる。例えば、コンピューター上で実行中のアプリケーションおよびコンピューターは、コンポーネントとすることができる。1つまたは複数のコンポーネントが、プロセスおよび/または実行スレッド内に存在することができ、コンポーネントは、1つのコンピューター上に局在すること、および/または2つ以上のコンピューター間に分散することができる。

【0104】

10

上述のシステムは、いくつかのコンポーネント間の対話に関して説明された。そのようなシステムおよびコンポーネントが、それらのコンポーネントもしくは特定のサブコンポーネント、特定のコンポーネントもしくはサブコンポーネントのいくつか、および/または追加のコンポーネントを、それらの様々な順列および組み合わせに従って含むことができることが理解できよう。サブコンポーネントは、親コンポーネント内に含まれる(階層的である)代わりに、他のコンポーネントに通信可能に結合されたコンポーネントとして実施することもできる。加えて、1つまたは複数のコンポーネントは、集約機能を提供する単一のコンポーネントに組み合わせること、またはいくつかの別個のサブコンポーネントに分割することができ、統合機能を提供するために、そのようなサブコンポーネントに通信可能に結合する、管理層などの任意の1つまたは複数の中間層を提供することに留意されたい。本明細書で説明されたコンポーネントはいずれも、本明細書で特に説明されないが、当業者に一般に知られている、1つまたは複数の他のコンポーネントと対話することもできる。

20

【0105】

上で説明された例示的なシステムに鑑みて、説明された主題に従って実施できる方法は、様々な図のフローチャートを参照することでより良く理解されよう。説明を簡素にする目的で、方法は一連のブロックとして示され、説明されたが、いくつかのブロックは本明細書で叙述および説明されたのとは異なる順序ならびに/または他のブロックと同時に生じることができるので、特許請求される主題はブロックの順序によって限定されないことを理解および認識されたい。非順次的または分岐するフローがフローチャートを介して示された場合、同一または類似の結果を達成する他の様々な分岐、フロー経路、およびブロックの順序が実施できることが理解できよう。さらに、これ以降で説明される方法を実施するために、図示されたブロックのすべてが必ずしも必要とされるわけではない。

30

【0106】

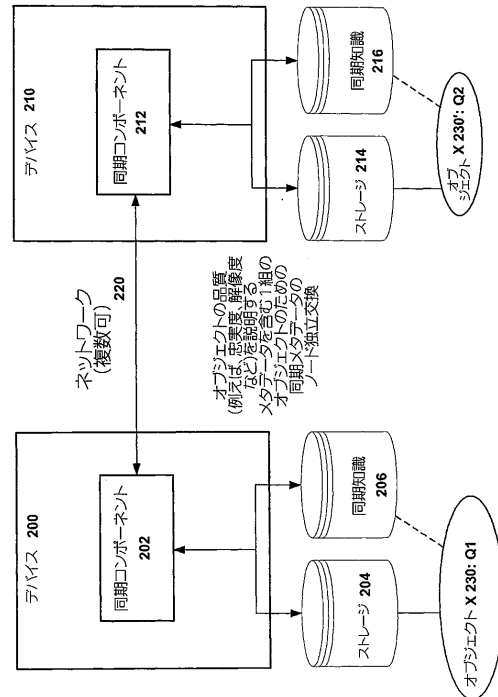
本明細書で説明された様々な実施形態に加えて、対応する実施形態から逸脱することなく、それらと同一または等価の機能を実行するために、他の類似の実施形態が使用でき、または説明された実施形態に変更および追加を施し得ることを理解されたい。またさらに、多数の処理チップまたは多数のデバイスは、本明細書で説明された1つまたは複数の機能の実行を分担することができ、同様に、ストレージは、複数のデバイスにわたって達成することができる。したがって、本発明は、任意の単一の実施形態または1組の実施形態に限定されず、むしろ添付の特許請求の範囲に従った広さ、主旨、および範囲において解釈されるべきである。

40

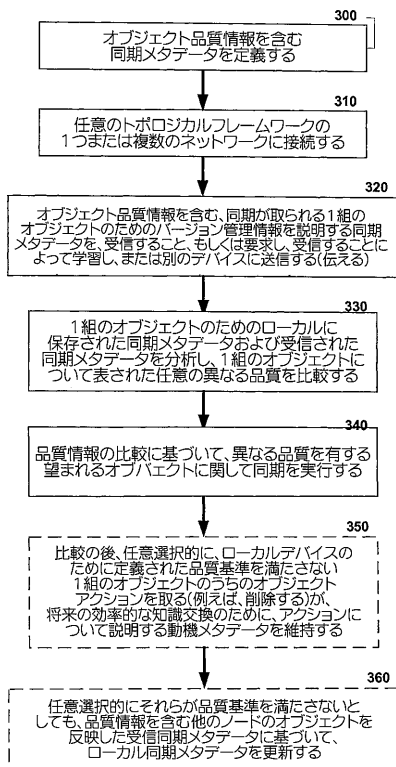
【図 1】



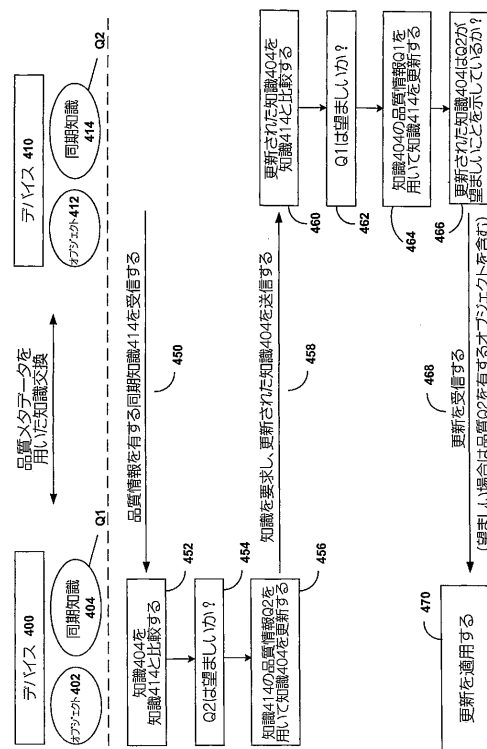
【図 2】



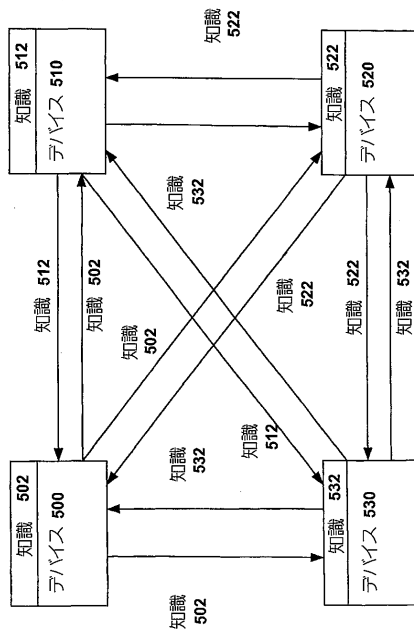
【図 3】



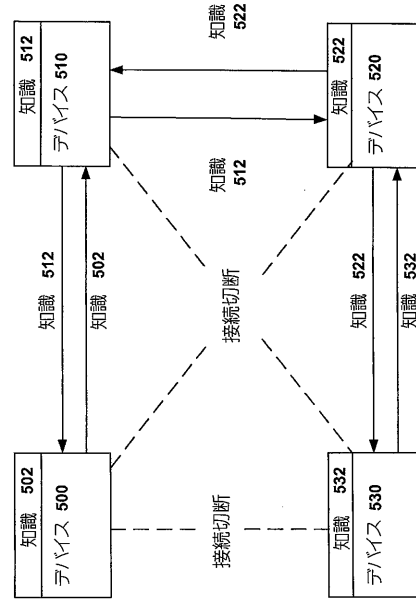
【図 4】



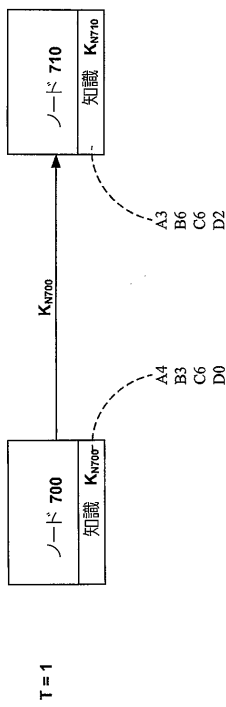
【図 5】



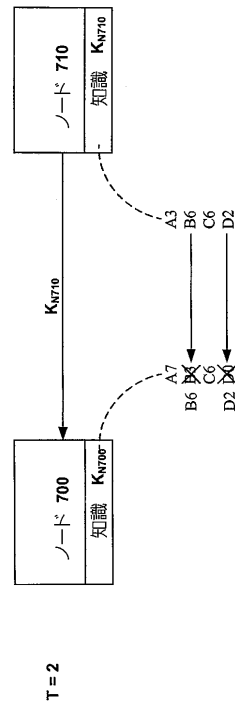
【図 6】



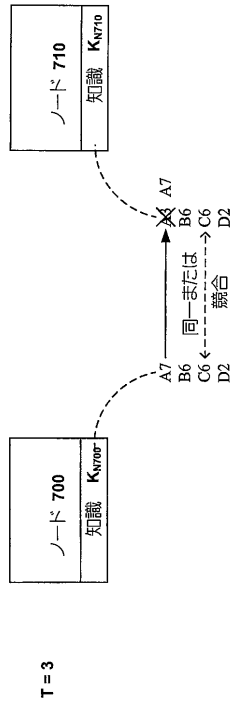
【図 7】



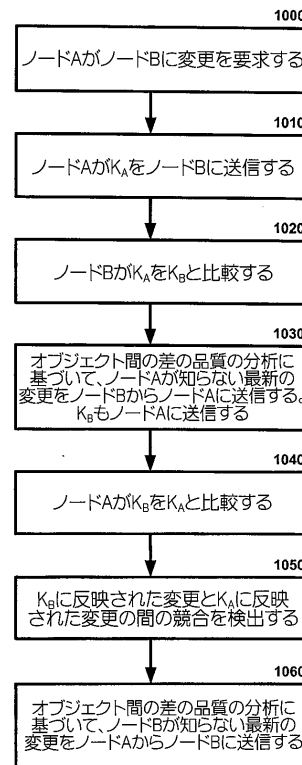
【図 8】



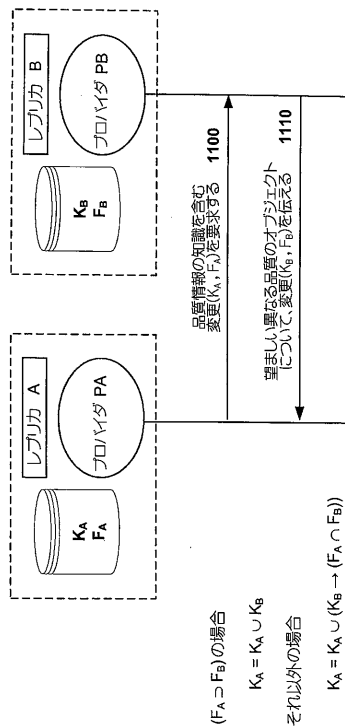
【図 9】



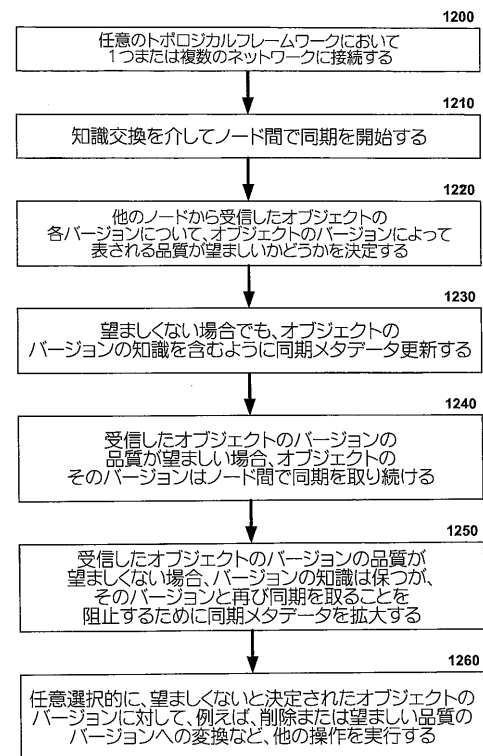
【図 10】



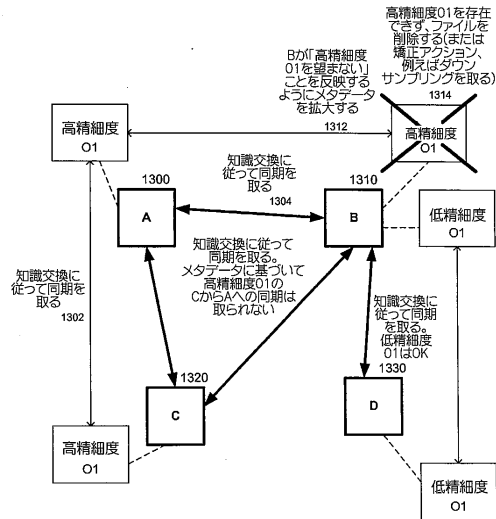
【図 11】



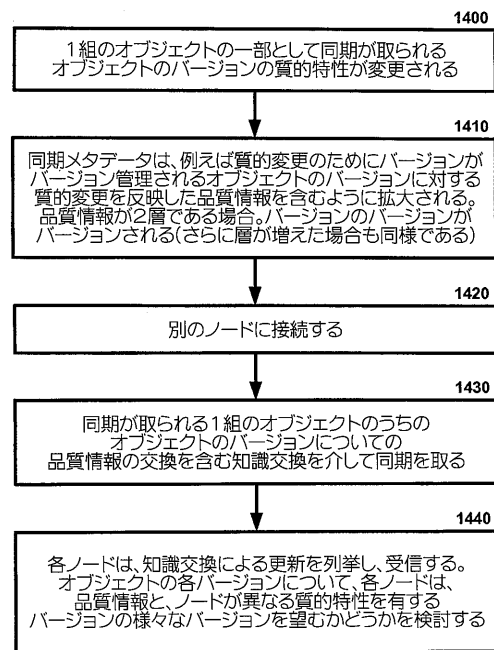
【図 12】



【 図 1 3 】



【 図 1 4 】



【 図 1 5 】

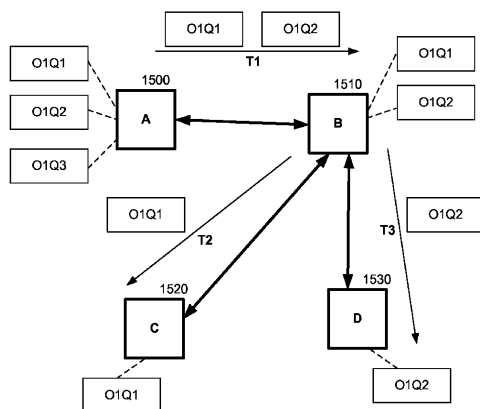
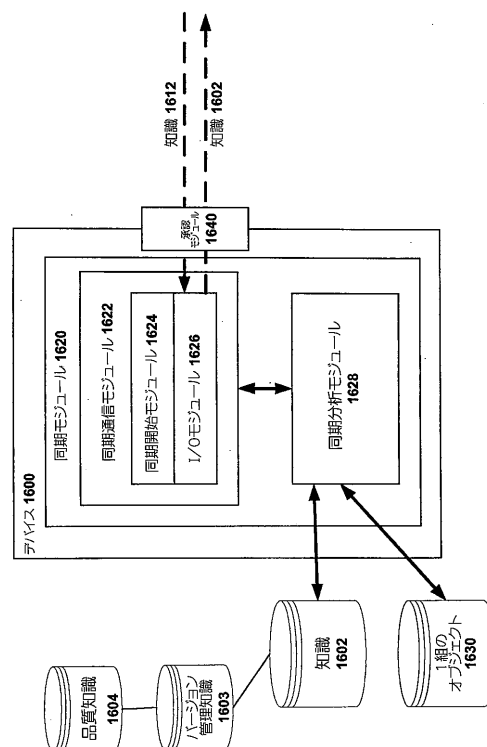
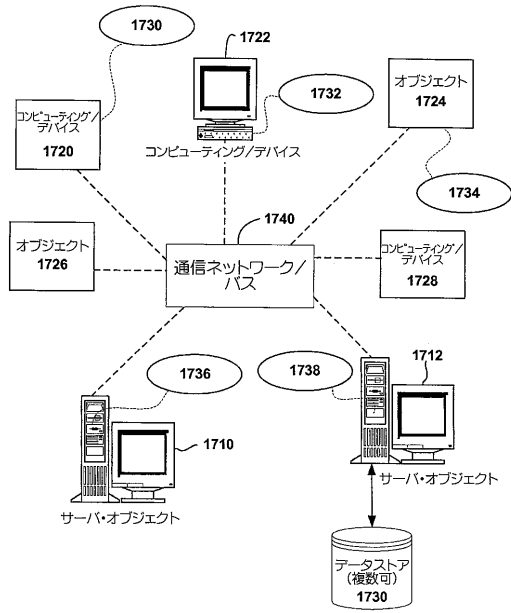


FIG. 15

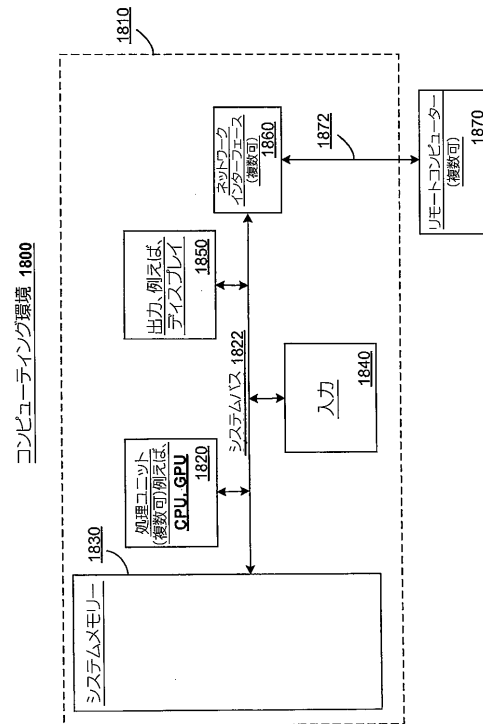
【 図 1 6 】



【図 17】



【図 18】



フロントページの続き

- (72)発明者 クラーク, マイケル・レイ
アメリカ合衆国ワシントン州 9 8 0 5 2, 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
- (72)発明者 ノヴィク, レヴ
アメリカ合衆国ワシントン州 9 8 0 5 2, 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
- (72)発明者 コースラヴィ, モエ
アメリカ合衆国ワシントン州 9 8 0 5 2, 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
- (72)発明者 リー, オリヴァー
アメリカ合衆国ワシントン州 9 8 0 5 2, 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

審査官 池田 聡史

- (56)参考文献 特開 2 0 0 7 - 0 5 8 2 7 5 (J P , A)
特開平 1 1 - 1 4 9 4 4 8 (J P , A)
特開 2 0 0 0 - 0 5 7 0 3 2 (J P , A)
特開 2 0 0 0 - 2 2 2 2 6 8 (J P , A)
特開 2 0 0 0 - 3 3 9 2 1 1 (J P , A)
米国特許出願公開第 2 0 0 3 / 0 1 4 5 0 2 0 (U S , A 1)
荒木拓也ほか 2 名, マスターレスな DB 同期システムとシームレスな動画視聴システムへの応用, 情報処理学会研究報告, 日本, 社団法人情報処理学会, 2 0 0 6 年 2 月 1 7 日, 第 2006 巻 第 14 号, pp.299 ~ 304
藤田将幸, 効きめバツグン! Active Directory 実践テクニック 最終回 複製の競合に関するトラブル, Windows 2 0 0 0 World, 日本, 株式会社 IDG ジャパン, 2 0 0 3 年 6 月 1 日, 第 8 巻 第 6 号, pp.171 ~ 173
MSDN Selection(2) Microsoft Windows NT Active Directory 技術概要, アスキー NT, 日本, 株式会社アスキー, 1 9 9 8 年 7 月 1 日, 第 3 巻 第 7 号, pp.268 ~ 276

- (58)調査した分野(Int.Cl., DB 名)
G 0 6 F 1 2 / 0 0
J S T P l u s (J D r e a m I I I)