

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2013-541115

(P2013-541115A)

(43) 公表日 平成25年11月7日(2013.11.7)

(51) Int.Cl.
G06F 12/00 (2006.01)F I
G O 6 F 12/00 5 3 3 F

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 24 頁)

(21) 出願番号 特願2013-536673 (P2013-536673)
(86) (22) 出願日 平成23年10月19日 (2011.10.19)
(85) 翻訳文提出日 平成25年6月26日 (2013.6.26)
(86) 国際出願番号 PCT/US2011/056969
(87) 国際公開番号 W02012/058078
(87) 国際公開日 平成24年5月3日 (2012.5.3)
(31) 優先権主張番号 13/015,816
(32) 優先日 平成23年1月28日 (2011.1.28)
(33) 優先権主張国 米国 (US)
(31) 優先権主張番号 61/406,942
(32) 優先日 平成22年10月26日 (2010.10.26)
(33) 優先権主張国 米国 (US)

(71) 出願人 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 100140109
弁理士 小野 新次郎
(74) 代理人 100075270
弁理士 小林 泰
(74) 代理人 100101373
弁理士 竹内 茂雄
(74) 代理人 100118902
弁理士 山本 修
(74) 代理人 100153028
弁理士 上田 忠

最終頁に続く

(54) 【発明の名称】 オンライン文書編集の同期化

(57) 【要約】

オンライン文書サービスは、ユーザによって編集されるアプリケーションサーバによって提供される。文書は、アプリケーションサーバのローカルメモリ内のコンポーネントベースの構造に記憶されて、共同オーサリングおよび衝突解決の精度を提供する。コンポーネントレベルのロックを利用して、同時に起こるユーザ編集ベースの衝突を最小限にし、そして他のユーザの存在も示す。コンポーネントベースの構造修正は、コンポーネント編集をキャプチャするために記憶され、そして文書の保存アクションの後同期する。編集が1または複数のクライアントによってユーザに保存される時、その編集は、文書がクライアント上の現在の表現を生成するのに使用されたバージョンから変更されていても、サーバ文書の現在のバージョンに保存され得る。

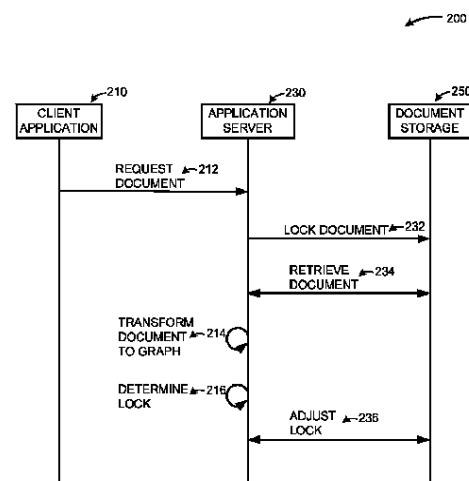


FIG. 2

【特許請求の範囲】**【請求項 1】**

オンライン文書編集を同期するために少なくとも一部はコンピューティングデバイスによって実行される方法であって、前記方法は、

文書の第 1 のセクションと関連付けられる第 1 の共同オーサリングメタデータの指示を受信するステップであって、前記第 1 の共同オーサリングメタデータは、ブラウザベースのクライアントアプリケーションから受信される、受信するステップと、

前記文書の第 2 のセクションと関連付けられる第 2 の共同オーサリングメタデータの指示を受信するステップであって、前記第 2 の共同オーサリングメタデータは、第 2 のクライアントアプリケーションから受信され、前記第 2 のクライアントアプリケーションは、ブラウザで動作しない、受信するステップと、

前記ブラウザベースのクライアントアプリケーションに提供される変換表現に基づいて前記第 1 の共同オーサリングメタデータを解釈するステップと、

前記文書と関連する前記第 1 の共同オーサリングメタデータおよび前記第 2 の共同オーサリングメタデータを記憶するステップとを備えたことを特徴とする方法。

【請求項 2】

前記第 1 の共同オーサリングメタデータは、ユーザ名、ユーザの存在情報、および文書内ロックのセットのうち少なくとも 1 つを含むことを特徴とする請求項 1 に記載の方法。

【請求項 3】

コンポーネントレベルの変更された文書において他の編集と衝突する 1 または複数の編集を認識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。

【請求項 4】

前記衝突する編集が結びの句である場合、通知を用いずに、衝突する編集を同一コンポーネント内で統合することを試みるステップをさらに備えたことを特徴とする請求項 3 に記載の方法。

【請求項 5】

前記文書にアクセスするクライアント間でパラグラフの番号を同期することを確認するための要求を前記クライアントアプリケーションから受信した後、前記文書の各パラグラフを再番号割り当て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。

【請求項 6】

前記ユーザが前記文書の単一の共作成者であるかどうかを判定するための要求を、前記文書と関連付けられる共同オーサリングアプリケーションを実行するサーバに送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。

【請求項 7】

オンライン文書編集を同期するオンライン文書アプリケーションサーバであって、前記サーバは、

メモリと、

前記メモリに接続されたプロセッサであって、前記プロセッサは、前記メモリに記憶された命令と共にアプリケーションを実行し、前記アプリケーションは、

文書の要求をユーザから受信し、

前記文書に対する文書ロックを記憶サーバから要求し、

前記文書を前記記憶サーバから読み出し、

前記文書を、前記文書のコンテンツを含む複数のコンポーネント、ならびにユーザ名およびユーザの存在情報を含む第 1 の共同オーサリングメタデータに変換し、

少なくとも 2 人のユーザによる編集間の衝突を判定し、

前記複数のコンポーネントに対する複数のコンポーネントロックを、前記衝突する編集を評価することによって判定し、

前記文書ロックを解除することによって前記文書ロックを前記複数のコンポーネント

10

20

30

40

50

ロックに調整し、前記複数のコンポーネントロックを、適合するコンポーネントに適用し、

前記グラフを表示用のブラウザベースのクライアントアプリケーションに送信するように構成される、プロセッサと
を備えたことを特徴とするサーバ。

【請求項 8】

前記衝突を判定した後、前記アプリケーションは、

少なくとも 1 つのユーザ選択に対し、少なくとも 1 つの衝突するユーザ編集および少なくとも 1 つの対応するコンポーネントを、前記ブラウザベースのクライアントアプリケーションに送信することであって、前記少なくとも 1 つの選択は、前記衝突を解消する前記
少なくとも 1 つの対応するコンポーネントに記録される

ようにさらに構成されることを特徴とする請求項 7 に記載のアプリケーションサーバ。

【請求項 9】

前記アプリケーションは、

前記複数のコンポーネントロックを、第 2 の共同オーサリングメタデータとして文書グラフに記憶する

ようにさらに構成されることを特徴とする請求項 7 に記載のアプリケーションサーバ。

【請求項 10】

オフラインブラウザベースのクライアントアプリケーションは、オンラインに戻った後、統合するための少なくとも 1 つのユーザ編集を、前記文書グラフに送信することを特徴とする請求項 9 に記載のアプリケーションサーバ。

【請求項 11】

前記アプリケーションは、

編集時間、ユーザの認証情報、およびユーザの役割のセットのうちの少なくとも 1 つを、前記ブラウザベースのクライアントアプリケーションに送信する

ようにさらに構成されることを特徴とする請求項 7 に記載のアプリケーション。

【請求項 12】

オンライン文書編集を同期するための命令を記憶するコンピュータ可読記憶媒体であって、前記命令は、

文書の要求をユーザから受信するステップと、

前記文書に対する文書ロックを記憶サーバから要求するステップと、

前記文書を前記記憶サーバから読み出すステップと、

前記文書を、ユーザ名およびユーザの存在情報を含む第 1 の共同オーサリングメタデータを含む複数のコンポーネントに送信し、かつ前記複数のコンポーネントをグラフに記憶するステップと、

前記複数のコンポーネントに対する複数のコンポーネントロックを、少なくとも 1 つのユーザ編集を評価することによって判定するステップと、

前記文書ロックを解除することによって前記文書ロックを前記複数のコンポーネントロックに調整し、前記複数のコンポーネントロックを、前記グラフ内の適合するコンポーネントに適用し、および前記複数のコンポーネントロックを第 2 の共同オーサリングメタデータとして前記グラフに記憶するステップと、

前記グラフを表示用のブラウザベースのクライアントアプリケーションに送るステップと

を備えたことを特徴とするコンピュータ可読記憶媒体。

【請求項 13】

前記命令は、現在のユーザに、前記複数のコンポーネントロックを通知することをさらに備えたことを特徴とする請求項 12 に記載のコンピュータ可読記憶媒体。

【請求項 14】

前記命令は、

コンポーネントレベル識別子を記憶するステップと、

前記文書の他の部分が変化しても、前記コンポーネントレベル識別子を使用して、ユーザの編集を、適したコンポーネント領域に移動するステップと

をさらに備えたことを特徴とする請求項 1 2 に記載のコンピュータ可読記憶媒体。

【請求項 1 5】

前記命令は、

ユーザに、前記文書を編集する現在の作成者の名前および存在情報を通知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 2 に記載のコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、オンライン文書編集の同期化に関する。

【背景技術】

【0002】

ウェブアプリケーションは、ネットワークを介して多様なサービスおよびデータをユーザに提供する。データは、集められ、処理され、そして異なる位置に記憶される。ウェブアプリケーションは、そのデータを読み出し、そのデータを提示するのにフォーマットし、そしてクライアントデバイス上でウェブページをレンダリングする閲覧アプリケーションにそのデータを提供する。一部のウェブページには、そのデータが双方向通信しない静的なものもある。他のウェブページは、リンクを介した付加的な情報またはウェブベースのモジュールの起動など、ある程度の双方向性を提供できる。しかしながら、一般に、ウェブページは、ウェブページの作成者によって決められたフォーマットおよび量のデータを提示する。

【0003】

オンライン文書アプリケーションは、文書を編集してビューする能力をユーザに提供するが、その能力は、最近までシッククライアントアプリケーションの分野であった。計算の技術的進歩とネットワークおよびデータ記憶の拡大とによって、オンラインアプリケーションがシッククライアントアプリケーションの文書編集機能を提供することができるようになった。プラットフォームに依存しないさまざまな基本技術を通じてオンラインアプリケーションを使用できる利点により、多数のユーザが文書の生成および管理を共同で行うことができるようになった。しかしながら、複数のユーザによる文書へのアクセスは、非同期のユーザ編集につながる恐れがある。さまざまな技術を介して文書が複数のプラットフォームにアクセスする場合、文書の維持管理および文書の一貫性をさらに複雑にする恐れがある。

【発明の概要】

【0004】

本概要は、以下の発明を実施するための形態でさらに説明される概念の選択を簡略化した形で紹介するために与えられる。本概要は、特許請求される主題の主要な特徴または不可欠な特徴のみを明らかにすることも、特許請求される主題の範囲を決めるのに役立つものとして用いられることも意図しない。

【0005】

実施形態は、文書コンポーネントレベルで修正を制御することによってオンライン文書編集を同期することを対象とする。ある実施形態に従って、文書を文書コンポーネントのグラフに変換でき、および複数のユーザによって提出される変更を管理するためにロックをコンポーネント上でアサートできる。グラフコンポーネントの変更は、グラフの修正を維持管理することによって追跡できる。

【0006】

これらと他の特徴および利点は、以下の詳細な説明を読み、且つ関連する図面を見直すことによって明らかになるであろう。前述の概説と以下の詳細な説明との両方は、説明するためのものであり、特許請求される態様を制限するものではないことを理解されたい。

【図面の簡単な説明】

10

20

30

40

50

【 0 0 0 7 】

【図 1】オンライン文書編集サービスの例示的コンポーネントを示す図である。

【図 2】編集を管理するロックアクションの例示的ステップを示す図である。

【図 3】編集を管理する修正実装の例示的ステップを示す図である。

【図 4 A】ある実施形態に係る例示的シナリオを示す図である。

【図 4 B】ある実施形態に係る例示的シナリオを示す図である。

【図 4 C】ある実施形態に係る例示的シナリオを示す図である。

【図 5】実施形態に係るシステムを実装できるネットワーク化環境である。

【図 6】実施形態を実装できる例示的コンピューティングオペレーティング環境のブロック図である。

10

【図 7】実施形態に係る文書コンポーネントレベルで修正を制御することによってオンライン文書編集を同期するプロセスの論理流れ図である。

【発明を実施するための形態】

【 0 0 0 8 】

上記に簡潔に説明したように、ロックアクションを使用して文書コンポーネントレベルで修正を制御することによってオンライン文書編集を同期できる。文書を文書コンポーネントのグラフに変換できる。複数のユーザによって提出される変更を管理するためにロックをコンポーネント上でアサートできる。各共作成者が編集する文書をコンポーネントレベルで編集することを含めたグラフの修正を維持管理することによってグラフコンポーネントの変更を追跡できる。以下の詳細な説明において、その一部を成す添付図面についての説明が行われ、図解を目的とする実施形態または例によって示される。本開示の精神または範囲から逸脱しない範囲においてこのような態様を組み合わせてもよいし、他の態様を利用してもよいし、構造の変更を行ってもよい。従って、以下の詳細な説明は、限定された意味で解釈されるものではなく、本発明の範囲は、添付の特許請求の範囲およびその等価物によって定義される。

20

【 0 0 0 9 】

実施形態は、コンピューティングデバイスのオペレーティングシステム上で実行するアプリケーションプログラムと共に実行するプログラムモジュールの一般的な状況において説明されるが、態様は、他のプログラムモジュールと共に実装されてもよいことが当業者には認識されよう。

30

【 0 0 1 0 】

一般に、プログラムモジュールは、ルーチン、プログラム、コンポーネント、データ構造、および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する他のタイプの構造を含む。さらに、実施形態は、ハンドヘルドデバイス、マルチプロセッサシステム、マイクロプロセッサベースまたはプログラマブルな家電、ミニコンピュータ、メインフレームコンピュータ、および同等のコンピューティングデバイスを含む、他のコンピュータシステム構成で実施されてよいことが当業者には認識されよう。実施形態を、通信ネットワークにリンクされる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できる。分散コンピューティング環境において、プログラムモジュールをローカルメモリ記憶デバイスとリモートメモリ記憶デバイスとの両方に配置することができる。

40

【 0 0 1 1 】

実施形態を、コンピュータ実装プロセス（方法）、コンピューティングシステムとして、あるいはコンピュータプログラム製品またはコンピュータ可読媒体などの、製造品として実装できる。コンピュータプログラム製品は、コンピュータシステムによって読み取り可能であり、且つコンピュータまたはコンピューティングシステムが例示的プロセス（複数）を実行するようにさせる命令を備えるコンピュータプログラムを符号化する、コンピュータ記憶媒体であってよい。コンピュータ可読記憶媒体を、例えば、揮発性コンピュータメモリ、不揮発性コンピュータメモリ、ハードドライブ、フラッシュドライブ、フロッピー（登録商標）ディスク、またはコンパクトディスク、および同等の記憶媒体のうちの

50

1 つまたは複数によって実装することができる。

【0012】

この明細書全体を通じて、用語「プラットフォーム」は、実施形態を実装できるさまざまな文書タイプまたは同様の環境に共同オーサリングサービスを提供するソフトウェアコンポーネントとハードウェアコンポーネントとの組み合わせであってよい。プラットフォームの例は、複数のサーバ上で実行されるホストサービス、単一のサーバ上で実行されるアプリケーション、および同等のシステムを含むが、これらに限定されない。用語「サーバ」は、一般に、典型的にはネットワーク化環境において 1 または複数のソフトウェアプログラムを実行するコンピューティングデバイスを指す。しかしながら、サーバは、ネットワーク上のサーバと見なされる 1 または複数のコンピューティングデバイス上で実行される仮想サーバ（ソフトウェアプログラム）として実装されてもよい。このような技術および例示的動作について以下により詳細に説明する。

10

【0013】

図 1 は、オンライン文書編集サービスの例示的コンポーネントを示す図である。100 において、サーバ 110 は、1 または複数のオンライン文書編集アプリケーションを実行し、そしてネットワーク 140 経由で情報、特に文書コンテンツを送信できる。ネットワーク 140 は、ローカルネットワークであってよいし、またはインターネットベースのインフラストラクチャなどの、外部エンティティであってよい。そのネットワークは、有線または無線接続性を提供できる。ネットワークノードは、セキュアでないまたはセキュアな接続性を介して互いに接続できる。セキュアな接続性の例として、ネットワークノード間で暗号化された通信を使用して確立される仮想プライベートネットワーク（VPN）がある。

20

【0014】

サーバ 110 は、さまざまなプロトコルを介してクライアントと通信する文書編集アプリケーションを提供できる。その一例として、ハイパーテキストトランスポートプロトコル（HTTP）がある。文書編集アプリケーションは、文書編集サービスをシンククライアントおよびシッククライアントのエンドユーザに提供できる。シンククライアント（またはウェブクライアント）131、134 は、サーバアプリケーションが提供する機能に依存できる。シッククライアント（またはリッチクライアント）137 は、サーバアプリケーションが提供する機能とローカル機能とを組み合わせ、エンドユーザに付加的なユーティリティを与えることができる。リッチクライアント 137 は、同じアプリケーションサーバ 110 に接続する必要がない。最終的に、すべてのクライアントは、同じ文書を文書サーバ 120 上で編集する。いくつかの実装に従って、アプリケーションサーバ 110 は、ウェブ編集能力に適するサーバにすることができ、別のサーバは、リッチクライアント 137 に適していて且つそれによって使用されるサービスを提供できる。アプリケーションサービスの例は、ユーザ編集とユーザ存在情報およびユーザ名とを統合してクライアントデバイス上でユーザ変更を表示できる。さらに、サーバアプリケーションは、複数のユーザが異なるクライアントデバイス（130、133 および 136）を介してサービスにアクセスできるようにする場合がある。例示的シナリオにおいて、ユーザは、文書にアクセスして変更した結果、その同じ文書（132、135 および 138）を異なるバージョンにできる。

30

40

【0015】

実施形態において、文書サーバ 120 は、文書記憶サービスであってよい。文書は、テキスト、描画、イメージ、ビデオ、およびオーディオを含む、文書のさまざまなタイプおよびフォーマットを記憶できるが、これらに限定されない。例示的システムにおいて、文書サーバは、アプリケーションサーバによって提供されるオンライン編集アプリケーションを介して複数のユーザによって編集されるテキスト文書を記憶できる。別の例示的システムにおいて、文書サーバは、アプリケーションサーバによって提供されるオンラインイメージ編集アプリケーションを介して複数のユーザによってアクセスおよび編集されるイメージ文書を記憶できる。しかし、他の例において、文書記憶サーバは、ハイブリッド文

50

書タイプのオンラインアプリケーションサービスを介してアクセスおよび編集を同時に行う複数のファイルタイプおよびフォーマットを複数のユーザに提供できる。

【 0 0 1 6 】

例示的シナリオにおいて、ユーザは、アプリケーションサーバによって提供される文書アプリケーションを介して編集するために既存の文書にアクセスできる。ユーザ要求の後、アプリケーションサーバは、記憶サーバ内の文書を読み出してロックできる。アプリケーションサーバは、文書を、その文書のコンポーネントをカプセル化するグラフに変換できる。アプリケーションサーバは、そのグラフを修正番号に割り当てることができる。アプリケーションは、ユーザ変更を評価し、そして文書ロックを、変更されたコンポーネントを包含するグラフコンポーネントを隠すコンポーネントロック（複数）に変えることができる。アプリケーションサーバは、グラフコンポーネントの変更を書き込んで、グラフの修正番号を変更でき、そしてファイルサーバに書き込むことによってそのグラフ変更を同期できる。さらに、オフラインクライアントは、オンラインに戻った後に統合する編集をグラフに送信できる。

10

【 0 0 1 7 】

ユーザが文書を共同オーサリングできるようにさせるアプリケーションサーバは、クライアントからのある通信を予測することができる。その通信は、2種類の要求、即ち、記憶要求およびサーバアクセス要求の形で入ってくる。要求動作を、2つの層を使用して実装できる。第1に、サーバは、各要求に対してシンプルオブジェクトアクセスプロトコル（SOAP）インタフェースを示すことができる。要求は、その後、その要求にサービスするサーバのフロントエンドのコンポーネントに渡される。

20

【 0 0 1 8 】

別の例示的实施形態において、ウェブブラウザベースのクライアントアプリケーションは、ブラウザで実行するスクリプトベースのコードと、フロントエンドで実行するC#コードなどの、実装に特化したコードとの2つの部分から構成され得る。フロントエンドは、要求を受信してサービスできる。その結果、アプリケーションは、ウェブブラウザベースのクライアントが要求できる方法を実装する際のオプションを有することができる。ブラウザで実行するコードは、1)サーバ上で示されたSOAPインタフェースに直接要求できるか、または2)サーバ上の単一のエン트리ポイントに直接すべて要求して、SOAP要求を受信すると、フロントエンドが、そのフロントエンド内から直接要求にサービスできるコンポーネントを呼び出すようにさせることができる。

30

【 0 0 1 9 】

サーバ上の単一のエン트리ポイントに直接すべて要求することは、事前処理、事後処理、管理容易性、移植性、および一貫性において有利になり得る。事前処理について、非同期JavaScript（登録商標）（AJAX）または同様の要求は、サーバによって要求される動作の詳細を包含できるバイナリストリームを包含できる。バイナリストリームの生成は、スクリプト言語では面倒になる恐れがある。さらに、サーバは、より単純な命令からバイナリ要求を生成できるユーティリティにすでにアクセスした可能性がある。

【 0 0 2 0 】

40

事後処理について、生応答をバイナリストリーム以外でブラウザがより解釈しやすいフォーマットに変換するのに望ましい処理もあり得る。その応答は、文書を参照できるデータを包含できる。管理容易性について、単一のエン트리ポイントを利用するので、システムは、フロントエンド上で真の負荷を考慮するためのスロットリングなどの動作を実装できる。アプリケーションは、関連する複数の要求が同時に到達する時、より良い再使用状態にすることもできる。移植性について、ブラウザがさまざまなエンドポイントに発呼するようにさせるので、システムは、他のインプリメンタに実装から実装までのエンドポイント名を保持するように強制することによって、そのインプリメンタに負荷を加えることができる。一貫性について、ブラウザベースの文書編集は、ほとんどの通信を同じエンドポイントに集めることができる。

50

【 0 0 2 1 】

別の実施形態において、記憶要求を使用して、データを記憶または読み出すことができる。このような記憶要求は、基本文書に結合された 1 または複数のパーティションされたデータセルに要求できる。ウェブブラウザなどのクライアントアプリケーションからサーバのフロントエンドに記憶要求を行うことは、異なる機構を通じて実現できる。しかしながら、文書を共同オーサリングするユーザは、新しい要求をシステムに課すことができる、共同オーサリングに関連するメタデータを記憶して読み出す付加的な要求を要求できる。

【 0 0 2 2 】

例において、ウェブブラウザは、ページのコンテンツを読み出す動作を実行できる。ウェブブラウザは、文書に対してウェブブラウザベースのサービスを発呼できる。ウェブブラウザは、どの文書およびどのセルをターゲットにして要求を発行したいかについての情報をインスタンス化し、そしてその応答オブジェクトに書き込むことができる。

【 0 0 2 3 】

サーバがこの要求を受信すると、サーバアプリケーションは、(スロットリング、バッティングの後で) タイプに基づいて変換し、そしてそのタイプに基づいて適切なハンドラにデスパッチできる。サーバアプリケーションは、実装技術に基づいてエンドポイントが受け入れ可能なフォーマットに入力を変換できる。プロトコルは、その要求の入力をバイナリストリームでのみ受け取ることもあるし、または任意の入力を(そのようなデータを記憶する効率を最大にするために) バイナリデータに変換することもある。機能は、ストリー

【 0 0 2 4 】

別の例示的实施形態において、アプリケーションプロバイダ(110)のコンポーネントは、引数からバイナリストリームを構築できるオブジェクトを包含できる。ラッパーを実装して、そのような技術の実装において相互運用することが賢明であろう。

【 0 0 2 5 】

ラッパーは、ネイティブ機能をラップし、そしてその機能を実装コードに示すことができる。ラッパー実装オブジェクトは、ブラウザベースのサービス(特定のセルのデータを読み出すなど)に出す各タイプの要求ごとに存在し得る。ラッパー実装オブジェクトを、ブラウザベースのサービス要求に使用できる同じ引数でインスタンス化できる。ラッパー実装オブジェクトは、要求を実行するネイティブ機能の実装メソッドを呼び出す方法を認識する `AddToNativeRequest` メソッドを実装できる。最後に、実行メソッドは、要求を実行し、そしてその結果(即ち、ストリーム)を戻すインタフェースで呼び出されることもある。その結果は、ラッパーコード内のオブジェクト指向構造に変換して戻される。セルデータを記憶するまたは読み出す要求は、文書コンテンツのラッパーを使用するブラウザベースのアプリケーションとしてサービスされてもよい。

【 0 0 2 6 】

アプリケーションは、共同オーサリング用のメタデータを記憶して読み出す際に使用されるラッパーインタラクションを適用できる。しかしながら、メタデータは、セルオブジェクトの観点から表すことができない。アプリケーションによって記憶されるメタデータは、拡張マークアップ言語(XML)文書などのデータのオパークプロブになり得る。メタデータをセルオブジェクトのグラフに入れ込むことで、セルの観点から記憶できる。読み出した後、データは、ストリームに再構成され得るセルオブジェクトのグラフとして再生できる。

【 0 0 2 7 】

アプリケーションサーバ上のアプリケーションコードは、データをストリームとして記憶または読み出すことによってメタデータ記憶要求に対処できる。ストリームを、適切なスキーマに従って操作のXML文書オブジェクトに供給できる。

【 0 0 2 8 】

さらに別の実施形態において、サーバアクセス要求は、共同オーサリングセッションに参加する / 離脱するなどの、機能性を実行する要求のセットに限定されてもよい。機能性は、現在認証されているユーザに関する情報（ユーザ名、電子メールアドレス、および他のユーザ情報など）を求めることができる。そのような要求は、共同オーサリング要求と名付けられてもよい。サーバ要求は、記憶要求と同じパターンに従う。オブジェクトは、扱い易い引数を使用して共同オーサリング要求を生成する能力がある。サーバ要求のパラメータおよび出力をより単純にして、記憶要求が必要とするバイナリストリームを生成するコストの負荷よりもかなり少なくできる。

【0029】

例示的实施形態を、本明細書では特定のプロトコル、コマンド、メッセージ、およびシステムを用いて説明する。しかしながら、それらは、実施形態に対する制限と見なされるべきではない。本開示の異なる態様は、本明細書で説明される発明の原理を用いた他のプログラミング言語、プロトコル、システム、およびコンポーネントで実装され得る。

【0030】

図2は、文書へのアクセスを管理するロックアクションの例示的ステップを示す。図200は、実施形態に従って編集を管理するロックアクションのいくつかの例示的ステップを示す。ウェブブラウザなどのクライアントアプリケーション210は、アプリケーションサーバ230から文書を要求する(212)ことができる。要求が受信されると、アプリケーションサーバは、要求された文書上で文書ロックを生成するために文書記憶サーバ250にロック要求を送信する(232)ことができる。文書ロックを生成すると、アプリケーションサーバは、記憶サーバから文書を読み出す(234)ことができる。

【0031】

ある実施形態に従って、サーバは、最初に文書のロックを要求し、その後、その要求が共同オーサリングに適しているかを確認するために検査し(216)、そしてひとたびその判定が行われると、サーバは、そのロックを複数のクライアントが文書を開くことができる1つのロックに調整する(236)ことができる。ロックが適していないと判定される場合、サーバは、排他的ロックに戻ることができる。これによって、サーバが、文書が複雑なために共同オーサリングに適していない文書、ならびに文書コンテンツとの同期から外れたまたは同期が誤っている恐れのある文書のコンテンツに対する事前の情報を持たなくても、そのようなアクションに適した文書を保持できるようにさせる。

【0032】

例示的实施形態をXML文書ホスティングテキストにすることができる。アプリケーションサーバは、XML文書をそのスキーマおよびパラグラフコンポーネントにパースし、そしてグラフに修正番号を与える間にグラフのコンポーネントを記憶できる。記憶要求の応答は、ストリーム形式であってよい。ストリームをXML文書か、あるいはXMLの単純なAPIにパースしてメモリの負荷を回避する必要があるかもしれない。プロパティを特定するXML要素は、メンバ変数によって構成部分となり得る。サーバアクセス要求からの応答は、特化した構造になる場合もあり、ブラウザが扱い易い表現に直接変換され得る。

【0033】

代替的实施形態において、より複雑な事後処理のステップは、サーバに仕事をさせる代償の結果、ウェブブラウザなどのシンクライアントアプリケーション上でかなりの性能利得が得られる。二次的メタデータは、ロックの記述、および各ロックに対してロックによって隠されたパラグラフのパラグラフ識別子のリストを包含できる。ブラウザは、グラフをトラバースして、各ロックを隠されたパラグラフに適用するために、識別子がロックによって特定されるパラグラフを探す必要があるかもしれない。

【0034】

代替として、ロックの仕様は、対応するパラグラフ識別子の代わりに、パラグラフオブジェクトのオブジェクト識別子を包含できる。グラフ内のオブジェクトの識別子によってオブジェクトを求めるには、ランダムアクセスが効率的であり、アプリケーションは、全

10

20

30

40

50

てトラバースすることを回避できる。

【0035】

二次的メタデータの要求に応答するサーバのフロントエンドは、パラグラフ識別子の代わりに、隠れたパラグラフのオブジェクト識別子を特定するロックに応答するのに十分な情報を有することができる。二次的メタデータを読み出すと、サーバのフロントエンドは、ユーザがロックするのをパースし、その後、アプリケーションサーバ上の記憶装置からグラフの最新の修正を読み出すことができる。サーバのフロントエンドは続いて、グラフ内のすべてのパラグラフオブジェクトを見つけて、パラグラフ識別子からパラグラフオブジェクトのオブジェクト識別子までの逆マップを構築できる。サーバのフロントエンドは、パラグラフ識別子に代わりに、パラグラフオブジェクト識別子を包含する応答を構築できる。

10

【0036】

ブラウザ側では、共同オーサリングメタデータを解釈してそれに基づいて動作する共通のアクタ/エディタ/マネージャのパターンと、メタデータを移動するレプリケータとを有する以下のオブジェクトを含むことができる。

【0037】

1、エディタ表

エディタ表を定期的にダウンロードおよびアップロードする（セッション中2回）プロセス。

エディタ表のスキーマの情報をカプセル化するエディタ。

20

文書内の現在の共作成者を表すメモリ内構造を維持管理するマネージャ。マネージャは、エディタがデータを解釈する時にそのエディタによって呼び出される追加および削除メソッドを示すことができる。マネージャは、相手のGUID識別子によってユーザを見つける検索メソッドも示すことができる。

インスタントメッセージなどの他の機能性を提供することもできる任意のUIに対処するアクタ。

【0038】

2、二次的メタデータ

二次的メタデータを定期的にダウンロードおよびアップロードするプロセス。

【0039】

二次的メタデータは、いくつかの情報を包含でき、主に文書内ロックに関する情報を包含する。二次的メタデータの、文書内ロック以外の情報をレプリケータで管理できる。ロックについて、アクタ/エディタ/マネージャのパターンを再度展開できる。

30

【0040】

1、ロックを追加および削除して、グラフのコンポーネントを提供するメソッドを有するアクタ。アクタは、ロックに対して「ここで入力できますか？」という問いに答えるメソッドも有することができる。

【0041】

2、プレースホルダー、エフェメラル、自動削除などの、さまざまなリストに入れることができる現在周知のロックのセットを維持管理するマネージャ。すべてのエフェメラルロックを除去して、それらをプレースホルダーロックに変えるなどの、グローバルロック動作。プレースホルダー、エフェメラル、および自動削除ロックをパラグラフなどの文書コンポーネントと関連付けることができる。

40

【0042】

3、設計は、ロック編集オブジェクトを必要としないかもしれない。ロックオブジェクトは、ひとたび生成されると、効率的に不変になり得るので、実行される実編集動作が必要ないかもしれない。

【0043】

スキーマロックをパラグラフなどの文書コンポーネントと関連付けることもできる。代替として、ウェブブラウザベースのクライアントは、共同オーサリングを許可しない特徴

50

の広範なセットを有することができる。ブラウザのクライアントがスキーマロックを用いてすでに開いている文書（即ち、別のクライアントがロックを用いた）を見つけることができたとしても、ブラウザのクライアントは、全文書をスキャンする必要がある、問題となる特徴が見つからなければ、ユーザに編集のみを許可することもある。これは、サーバアプリケーションがほとんどすべての文書コンテンツを読み取って、そのコンテンツをグラフに変換することができ、そしていつでも動作を停止できるので、ウェブブラウザベースのクライアントに対する付加的な負荷とはならない。

【 0 0 4 4 】

別の実施形態において、シード同期を、文書内の各パラグラフ識別子に再度番号を割り当てる要求として使用できる。要求は、要素を二次的メタデータまたは主メタデータ内の同じ名前に一致させることによって発行され得る。再番号割り当ては、文書識別子から開始される単一の増分カウントとして実行され、および事前定義された順序で文書内のパラグラフをウォークスルーすることによって実行され得る。二次的メタデータの後続するすべての要求および情報は、元の文書に記憶されたパラグラフ識別子ではなく、再番号割り当てから生じるパラグラフ識別子を参照できる。

【 0 0 4 5 】

シード同期化要求の実装は、再番号割り当てがヘッダ/フッタ、フットノートおよびエンドノート、テキストボックス内のテキストなどの、ブラウザでは（現在）見えない文書の部分に依存する場合があるので、ブラウザでは不可能かもしれない。シード同期化をサーバのフロントエンドに実装できる。

【 0 0 4 6 】

シード同期要求を実装する単純な手法は、文書のすべてのパラグラフのプロパティ（パラグラフ識別子が記憶されているプロパティ）を変更できる新しい修正を生成することである。第2の手法は、フロントエンド内の特定の要求メソッドを実装することである。そのメソッドは、古いパラグラフ識別子を新しいパラグラフ識別子にマップするディクショナリの戻り値を有する。ブラウザは、二次的メタデータのシード同期要素を受信する時に要求を発行して、結果として生じたマップを使用してメタデータのリマインダを解釈できる。

【 0 0 4 7 】

代替的实施形態において、サーバの負荷を最小に保つために、ユーザが文書の単一の共作成者であるかどうかを判定する負荷の軽い要求を実装できる。ウェブブラウザベースのクライアントもサーバの負荷を最小限にする「自分1人ですか？」という要求を実装できる。

【 0 0 4 8 】

例示的实施形態として全シーケンスを以下に示す。以下のステップにおいて、（B）は、「ブラウザ」であり、（S）は、（フロントエンドアプリケーションの）サーバである。

【 0 0 4 9 】

1、（B）ファイルのコンテンツを入手するウェブブラウザベースのサービス要求を行う。

【 0 0 5 0 】

2、（S）スキーマロックを用いてファイルをロックして、セッションに参加することを試みる。

a、（S）スキーマロックによってファイルが既にロックされていて、継続する場合、ユーザは、もう単独ではない。

b、（S）ファイルがその他のロックによってロックされた場合、セッションに失敗する。

c、（S）現在のユーザの認証情報およびサーバ時間を入手する要求を発行する。

【 0 0 5 1 】

3、（S）ストアからファイルを（ストリームとして）読み出す。

10

20

30

40

50

【 0 0 5 2 】

4、(S) ストリームを (ファイル内の各 x m l 要素をウォークスルーする) グラフに変換する。

a、(S) 共同オーサリングのセッションにおいて許可されていない要素と遭遇した場合に記憶する。

【 0 0 5 3 】

5、(S) ロックを調整する。

a、(S) すでにスキーマロックを用いてロックされた場合、

i、ステップ 4 a から発行がない場合、継続する。あるいは、排他的ロックに切り替えるように試みる。

b、(S) アプリケーションがスキーマロックか、または排他的ロックで終了したかどうかを記録する。さらに、ユーザが文書内ですでに単独ではない場合、2 a から記憶する。

【 0 0 5 4 】

6、(S)、このグラフを記憶装置に記憶し、そしてグラフをユーザ専用ルートに提供する。この方法において、複数のユーザは、すべての異なるグラフコンテンツを記憶装置の同じパーティション内に有して、互いに衝突しないようにすることができる。2人のユーザが同じ文書から開始するが、異なる変更を行う場合を例にする。グラフを記憶するために、グラフの変更は、保存動作を呼び出して、グラフの変更が1つずつコミットされるまで、別個のままにしなければならない。同じユーザが別のコンピュータまたは別のブラウザのインスタンスから (クラッシュの後でも) アプリケーションを起動する場合に、システムが、グラフのどの部分がそのユーザに属しているかを特定することができるように、このルートは、ユーザに固有である (しかし、ユーザのブラウザまたはコンピュータに固有ではない) 。

【 0 0 5 5 】

7、(S) このグラフをブラウザに送る。

【 0 0 5 6 】

8、(B) コンテンツを非直列化して表示する。

【 0 0 5 7 】

9、(B) ステップ 5 c において、アプリケーションがスキーマロックを有する場合、

a、(B) (ステップ 2 a から) 現時点で単独の場合、他の作成者が参加する時 / 参加するかどうかを判定するために A m I A l o n e に頻繁に尋ねるプロセスを開始する。

b、(B) ユーザが編集する時、(ユーザが単独または単独でないかに関わらず) ロックを生成する。

c、(ステップ 2 a から) 単独でない場合または (上記の 9 a から) 単独でない場合、

i、(B) このセッション (ステップ 9 b) の間に生成されるロックの表示を開始する。

i i、(B) 現在のユーザをエディタ表に付加する要求を発行する。

i i i、(B) エディタ表および二次的メタデータに対する反復プロセスを開始する。

【 0 0 5 8 】

付加的な作成者またはロックに関する新しい情報を適切な情報としてグラフに適用できる。ひとたびユーザがエディタを離脱すると、アプリケーションは、エディタ表から現在のユーザを除去する要求および共同オーサリングセッションを離脱する要求を発行できる。

【 0 0 5 9 】

さらに別の実施形態において、実装は、「編集不可能な」領域をサポートできる。その完全な機能は、編集可能な領域と編集不可能な領域との両方、ならびにテキスト以外のオブジェクト、またはテキストと他のオブジェクトとの組み合わせを包含する範囲にわたる

10

20

30

40

50

選択肢から成る。ユーザは、選択を行い、そして相手のコンテンツを編集不可能な領域に置いて、非ブラウザベースのクライアントに一貫した体験を提供することができる。

【0060】

図3は、ある実施形態に従った例示的シナリオを示す。図3に示すように、アプリケーションサーバ330は、文書編集サービスを提供できる。文書は、テキスト、描画、イメージ、オーディオ、およびビデオを含むが、これらに限定されないさまざまなフォーマットから構成されてよい。例示的実装は、複数のユーザの複数のクライアントアプリケーション310によって共同オーサリングされた文書を維持管理するアプリケーションサーバである。代替として、文書は、単一のフォーマットから構成されてよいし、またはテキスト、オーディオおよびビデオのコンテンツを組み合わせた文書などの、タイプの組み合わせを包含してもよい。

10

【0061】

図300に示すように、クライアントアプリケーションは、ユーザによって編集された文書を保存する要求を行う(312)ことができる。アプリケーションサーバは、文書記憶サーバ350に記憶された文書を読み出す(332)ことができる。ステップ314において、アプリケーションサーバは、ローカルに記憶された文書の基本グラフを読み出すまたは生成できる。ステップ316において、アプリケーションサーバは、ユーザ編集を包含するグラフの修正を読み出すことができる。その後、読み出された文書をローカルに記憶された基本修正に変換する(318)ことができる。基本修正と記憶された文書との間の変更は、アプリケーションサーバによって同期されることができる。ステップ320において、アプリケーションサーバは、最近再記憶された基本グラフの修正と、ユーザ編集を包含する任意の修正とを比較できる。基本グラフに入れられる編集は、ステップ322において文書に入力する準備ができてい。これは、322において読み出す文書が、グラフ修正316を生成するのに使用された文書と同じ文書でなくても、起こり得ることに留意されたい。従って、共同オーサリングアプリケーションは、「元」文書のローカルコピーを保持する必要がなく、(アプリケーションは、サーバ上の最新のコピーに対して働くので)スケールアウトする能力が向上する。文書の変更は、記憶サーバに保存できる(334)。文書の変更を保存した後、アプリケーションサーバは、その変更を包含する修正を基本グラフに付加できる。ステップ326において、アプリケーショングラフは、変更されたグラフをローカルに記憶できる。ステップ336において、どのロックホール

20

30

【0062】

実施形態において、どの変更も、非ブラウザベースのクライアントとブラウザベースのクライアントとの両方における現在のセッション中に文書に付加できる。しかしながら、非ブラウザベースのクライアントなどのシッククライアントは、保存アクション中にサーバから文書の最新バージョンを入手できる。ユーザは、衝突する変更または編集を解消するように求められる場合がある。ユーザ選択は、衝突を解消するためにグラフ内の対応するコンポーネントに記録され得る。ブラウザベースのアプリケーションに送られる衝突解消情報は、編集時間、ユーザに認証情報、ユーザの存在情報、およびユーザの役割を含むことができる。結果として生じたコンテンツは、新しい最新のバージョンとしてサーバに戻って保存され得る。

40

【0063】

別の実施形態において、保存中に統合するためのハイレベル実装は、以下を含むことができる。

【0064】

1、他のユーザ(または作成者)によって行われた変更のセットを入手する。

【0065】

2、ユーザによって行われた変更を他の作成者によって行われた変更と比較する。サーバは、「衝突」として双方によって操作されるすべてのオブジェクトを収集できる。

【0066】

50

3、衝突すると見なされるすべてのオブジェクトに対して、サーバは、アプリケーションが、ユーザが介入せずに衝突を解消できるかどうかを判定できる。衝突の解消は、望ましい変更の統合セットを表す現在のユーザの修正を操作することによって実現され得る。オブジェクトが解消されないままである場合、サーバは、統合動作を停止して、ブラウザに信号を送って衝突を解消するUIをユーザに提示するようにさせる。

【0067】

4、変更を文書に適用して、記憶サーバに保存できる。

【0068】

5、アプリケーションサーバは、変更のセットを他の作成者からブラウザに送ることができる。さらに、衝突を解消するために、ユーザ自身の変更に対して変更を任意に行うことができ、その変更を送ることもできる。

【0069】

他のユーザによって行われる変更のセットの入手は、なおも課題が存在し得る。他のユーザの変更を管理するための入力は、不完全に見えるかもしれない。文書の現在の状態は、利用できるであろうが、元の状態は、利用できない。しかしながら、アプリケーションサーバは、元グラフを通じて文書の元の状態について把握している。ユーザによって行われる変更は、一部グラフが元に変換されなかった文書において、他の作成者によって行われた変更と恐らく衝突しないであろう。サーバアプリケーションは、文書の現在の状態を変換し、そして結果として生じたグラフを元のグラフと比較できる。比較は、文書の他の共作成者によって行われた変更のセットにおいて生じ得る。変更のセットは、グラフの2つの状態の違いとなり得るので、修正として最も自然に表すことができる。従って、サーバアプリケーションは、衝突を検出するのに必要なすべての情報を入手し、そして統合動作の最終において他の共作成者の変更をブラウザに送ることができるようになる。

【0070】

現在の文書の状態から生じるグラフと元グラフとの比較は、それ自体に課題が存在する。パラグラフおよび他のオブジェクトが付加または除去されたか、あるいは任意の方法で移動された恐れがあるので、グラフ内のノードを1つずつ比較することを単に用いることができない。ある種の固有の識別子がアタッチされる文書（パラグラフまたは表の行を例とする）の構成では、これらの識別子を使用して、すぐに対応しなければならない元グラフからノードを選び出し、そして変化があったかどうかを確かめるためにプロパティを比較することができる。例示的实施形態において、システムは、最初に元グラフ全体にわたって読み込み、そしてパラグラフの識別子によってそのオブジェクト（元状態）を表したグラフノードにマップすることを構築する。その後、文書の現在の状態に対応するグラフを構築する。この新しいグラフのパラグラフに対応するノードが構成されると、元ノードは、パラグラフの識別子を使用して検索される。そのようなパラグラフが、元グラフに基づいて構成されたマップ内で見つからない場合、そのパラグラフは、他の作成者によって新しく付加されたことになる。ノードがマップ内で見つかった場合、現在のノードのプロパティは、元ノードのプロパティと比較される。プロパティが任意に付加または除去された場合、またはプロパティの値が元ノードの同じプロパティの値とは異なる場合、そのプロパティは、変更されたことになる。

【0071】

文書内に固有の識別子が適用される構成もあり得る。現在の構成と元の構成とを比較するために、固有の識別子が関連するそのような構成の近くまたは構成内に包含する他のオブジェクトに依存することができる。例えば、セル表は、それ自体の固有の識別子を有していない。しかしながら、この実施形態において、任意のセル表に包含する最終パラグラフと関連付けられる識別子は、その表（ならびに、そのパラグラフ自体）（セル表およびパラグラフノードのオブジェクトは、異なったタイプから成るので、両方の識別子として同じ値を使用する際の曖昧さが無い）の識別子として使用される。これは、それらの識別子に変更しないように、そのオブジェクトを変更する時のある法則を守るために編集システムの他の部分が必要になる。セル表の例の場合、セルに対するどの編集も、最終パラグ

10

20

30

40

50

ラフの Paragraph 識別子を変更できない。

【0072】

代替として、衝突を自動的に解消できない場合、他の作成者によって行われた変更を包含する修正は、衝突するオブジェクトとマークされてブラウザに送られ得る。さらに、同じ変更をブラウザに送って、ユーザの文書を最新の状態にすることができる。最後に、ユーザに、衝突を解消するためにUIを提示できる。ユーザは、すべての衝突が解消されると、再度保存を試みることができる。

【0073】

保存動作中に統合を実行する例示的シーケンスを詳述する。以下のステップにおいて、(B)は、「ブラウザ」であり、(S)は、(フロントエンドアプリケーションの)サーバである。

10

【0074】

1、(B)文書を保存するウェブブラウザベースのサービス要求を送る。

【0075】

2、(S)記憶サーバから文書の現在の状態を(I S t r e a mとして)読み出す。

【0076】

3、(S)記憶装置に記憶されていた元グラフをロード中に読み出す。

【0077】

4、(S)ユーザが行った編集としてサーバに蓄積された修正を読み出す。

【0078】

20

5、(S)ステップ3の元グラフを使用して、他のユーザによって行われた変更の修正を蓄積するベースラインとして文書を変換する。

【0079】

6、(S)ステップ4の修正をステップ5の修正と比較する。

【0080】

a、(S)両方のステップにおいてオブジェクトが見つかった場合、ステップ3または4による修正を変更することによって解消を試みる。

【0081】

b、どのオブジェクトも解消することができない場合、保存を停止し、ステップ5の修正に戻る。

30

【0082】

7、(S)文書の保存およびステップによって変更された修正を実行する。

【0083】

8、(S)結果として生じた文書を記憶サーバに戻して記憶する。

【0084】

9、(S)ステップ5による修正を基本グラフに付加して、その結果を新しいベースとしてマークする。

【0085】

10、(S)ステップ6の(恐らく変更された)修正を記憶装置に付加する。

【0086】

40

11、(S)ファイルのロックをリフレッシュする。

【0087】

12、(S)ステップ5の修正と(変更された場合)ステップ6の修正との両方を戻す。

【0088】

図4Aから図4Cまでは、ある実施形態に従った例示的シナリオを示す。例示の実施形態において、表の構造的変更の統合は、変更が互いに直交する場合に複雑になり得る。図410において、ユーザは、2×2表の編集を開始できる。ユーザが中央の列を付加して、そして他のユーザが中央の行を付加する場合、予測される結果は、図420ようになる。予測される結果を達成するには、サーバアプリケーションは、付加される行(表が2つ

50

の行を有する場合に付加される)が、統合中にその表に付加される第3のセルを有する必要があるかを検出する必要がある。あるいは、他のユーザによって付加された列が、セルが欠けている図430の表につながる恐れがある。このタスクは、列が第1クラスのオブジェクトとして文書でもグラフ表記でも表されないという事実によって困難になる。この複雑な状況は、1人のユーザが列(その列の数が変更していないように見える)を検出して列の同じ番号を付加する時か、またはすべての共作成者のアクションの総数が全表を除去する数に達する時に生じ得る。

【0089】

複雑な統合解決の代替的实施形態において、ブラウザベースのアプリケーションは、同じパラグラフを編集する2人の作成者に対処しなければならない場合がある。ブラウザベースのアプリケーションは、キャラクターレベルで変更を組み合わせるのに望ましくないもので、ユーザ入力無しではこの衝突の解消を試みることができない。アプリケーションは、ワードの複雑さのために無意味なワードを生成する恐れがある。分類する次の論理的単位は、ワード処理の境界に置かれるであろう。ブラウザベースのアプリケーションは、2つの付加または削除が合理的に独立したものと理解されている表のパラグラフまたは行と違って、パラグラフからワードまたは文を付加または除去するには、コンテンツをかなり変更するため、ユーザ入力も要求できる。

【0090】

複雑な統合解決の別の实施形態において、ブラウザベースのアプリケーションは、リストおよび再番号割り当てを用いて衝突を解消しなければならない場合がある。実装は、ワード処理アプリケーションなどのブラウザベースのアプリケーションが、リスト項目を記録し、そしてその項目に番号を割り当てることができる方法に関するコードの中で最高のレベルである。リスト項目の番号を計算し、そしてグラフ自体に(永続的でないプロパティとして)記憶できる。リスト項目として更新される番号を付加および削除するか、または上げるおよび下げることができる。さらに、ワード処理の場合、そのアプリケーションは、ブラウザのそのようなすべての項目メンバの値を再計算できない。しかしながら、サーバアプリケーションは、ロード動作中に値を計算して、ユーザ編集アクションの後、追加的にのみ変更できる。ユーザと他の共作成者との両方が衝突しない変更を行うことができる場合(しかし、結果として変更するリスト内の項目の番号割り当てが生じる)、サーバアプリケーションは、統合動作の部分として番号を更新する必要があるかもしれない。

【0091】

上記のオンライン文書編集を同期するシステムおよび実装は、図解を目的とし、実施形態に対して制限するものではない。文書は、テキスト、描画、イメージ、オーディオ、およびビデオを含むが、これらに限定されないさまざまなタイプから構成されてよい。文書は、タイプの組み合わせによって構成され得る。ユーザ編集を、本明細書で説明される発明の原理を使用した他のモジュール、プロセス、および構成を用いて同期できる。

【0092】

図5は、実施形態を実装できる例示的ネットワーク化環境である。ユーザ編集の同期化を管理するサーバアプリケーションを、ホストサーバなどの、1または複数のサーバ514または単一のサーバ(例えば、ウェブサーバ)516上で実行されるソフトウェア経由で実装できる。プラットフォームは、ネットワーク510を介してスマートフォン513、ラップトップコンピュータ512、またはデスクトップコンピュータ511(「クライアントデバイス」)などの、個々のコンピューティングデバイス上のクライアントアプリケーションと通信できる。

【0093】

上記のように、文書アプリケーションサーバは、文書記憶サーバに記憶された文書のユーザ編集を同期するアルゴリズムを実行できる。ユーザが文書のコンポーネントを編集する場合、アプリケーションサーバは、ユーザ編集集中にロックされたコンポーネントに関する情報を、クライアントデバイス511、512、513上で文書を編集する他の共作成者に送信できる。

10

20

30

40

50

【 0 0 9 4 】

クライアントデバイス 5 1 1、5 1 2、5 1 3 は、以前に説明したように、リモートサーバ（複数）（例えば、サーバ 5 1 4 のうちの 1 つ）上で実行されるアプリケーションにアクセスできるようにさせる。サーバ（複数）は、直接またはデータベースサーバ 5 1 8 を介して関連データをデータストア（複数）5 1 9 から読み出すまたはそのデータストアに記憶できる。

【 0 0 9 5 】

ネットワーク（複数）5 1 0 は、サーバ、クライアント、インターネットサービスプロバイダ、および通信媒体のうちの任意のトポロジを備えることができる。実施形態に従ったシステムは、静的または動的トポロジを有することができる。ネットワーク（複数）5 1 0 は、企業ネットワークなどのセキュアなネットワーク、無線オープンネットワークなどのセキュアでないネットワーク、またはインターネットを含むことができる。ネットワーク（複数）5 1 0 は、公衆交換電話網（PSTN）またはセルラーネットワークなどの、他のネットワークを介して通信を調整することもできる。さらに、ネットワーク（複数）5 1 0 は、Bluetooth（登録商標）またはこれに順ずる、短距離無線ネットワークを含むことができる。ネットワーク（複数）5 1 0 は、本明細書で説明されるノード間の通信を提供する。ネットワーク（複数）5 1 0 は、例として、音響、RF、赤外線および他の無線媒体などの、無線媒体を含むが、これらに限定されない。

【 0 0 9 6 】

コンピューティングデバイス、アプリケーション、データソース、およびデータ分散システムによる他の多くの構成を用いて、オンライン文書編集を同期できる。さらに、図 5 において論じたネットワーク化環境は、単に図解を目的とするものである。実施形態は、例示的アプリケーション、モジュール、またはプロセスに限定されない。

【 0 0 9 7 】

図 6 および関連する論述は、実施形態を実装できることに適したコンピューティング環境の簡潔で一般的な説明を与えることを意図する。図 6 を参照し、実施形態に従って、コンピューティングデバイス 6 0 0 などのアプリケーションの例示的コンピューティングオペレーティング環境を説明する。基本構成において、コンピューティングデバイス 6 0 0 は、オンライン文書のユーザ編集を同期するオンラインアプリケーションサーバであり、少なくとも 1 つの処理ユニット 6 0 2 およびシステムメモリ 6 0 4 を含むことができる。コンピューティングデバイス 6 0 0 は、プログラムを実行する際に協働する複数の処理ユニットも含むことができる。コンピューティングデバイスの正確な構成およびタイプに応じて、システムメモリ 6 0 4 は、揮発性（RAM など）、不揮発性（ROM、フラッシュメモリ など）またはその 2 つのある組み合わせであってよい。システムメモリ 6 0 4 は、典型的には、ワシントン州レッドモンドにあるマイクロソフト社製 WINDOWS（登録商標）オペレーティングシステムなどの、プラットフォームの動作を制御するのに適したオペレーティングシステム 6 0 5 を含む。システムメモリ 6 0 4 は、プログラムモジュール 6 0 6、文書サービス 6 2 2、および同期化モジュール 6 2 4 などの、1 または複数のソフトウェアアプリケーションも含むことができる。

【 0 0 9 8 】

文書サービス 6 2 2 は、編集用のオンライン文書を提供するサービスの一部であってよい。同期化モジュール 6 2 4 は、ユーザ編集を記憶された文書に同期して、共同オーサリング編集によって生じる衝突を解消することができる。文書は、コンポーネントに細分化され、そしてコンポーネントは、コンポーネントレベルのロックをパラグラフなどの文書の部分的編集に実装するグラフに記憶され得る。この基本構成は、図 6 の破線 6 0 8 内のコンポーネントによって示される。

【 0 0 9 9 】

コンピューティングデバイス 6 0 0 は、付加的な特徴または機能性を有することができる。例えば、磁気ディスク、光ディスクまたは磁気テープ、光テープなどの、付加的なデータ記憶デバイス（リムーバブルおよび / またはノンリムーバブル）を含むこともできる

10

20

30

40

50

。そのような付加的記憶装置は、図 6 のリムーバブル記憶装置 609 およびノンリムーバブル記憶装置 610 によって示される。コンピュータ可読記憶媒体は、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、または他のデータなどの、情報を記憶する任意の方法または技術で実装される、揮発性および不揮発性、リムーバブルおよびノンリムーバブル媒体を含むことができる。システムメモリ 604、リムーバブル記憶装置 609 およびノンリムーバブル記憶装置 610 は、すべてコンピュータ可読記憶媒体の例である。コンピュータ可読記憶媒体は、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)または他の光記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶デバイス、あるいは所望の情報を記憶するのに使用することができ、且つコンピューティングデバイス 600 によってアクセスすることができるその他の媒体を含むが、これらに限定されない。このような任意のコンピュータ可読記憶媒体は、コンピューティングデバイス 600 の一部であってよい。コンピューティングデバイス 600 は、キーボード、マウス、ペン、音声入力デバイス、タッチ入力デバイス、および同様の入力デバイスなどの、入力デバイス(複数) 612 を有することもできる。ディスプレイ、スピーカ、プリンタ、および他のタイプの出力デバイスなどの、出力デバイス(複数) 614 も含むことができる。このようなデバイスは当業者には周知であり、ここで詳細に論じる必要はない。

10

【0100】

コンピューティングデバイス 600 は、デバイスが、分散コンピューティング環境における無線ネットワーク、衛星リンク、セルラーリンク、および同様の機構などを介して、他のデバイス 618 との通信を行うようにさせる通信接続 616 を包含することもできる。他のデバイス 618 は、通信アプリケーション、記憶サーバ、および同様のデバイスを実行するコンピュータデバイス(複数)を含むことができる。通信接続(複数) 616 は、通信媒体の一例である。通信媒体は、その中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、または搬送波または他の移送機構などの、変調データ信号の他のデータを含むことができ、そして任意の情報配信媒体を含む。用語「変調データ信号」は、その特性セットのうちの 1 または複数を有するか、または信号の情報を符号化することに関してそのような方法で変更される信号を意味する。通信媒体は、例として、有線ネットワークまたは直接有線接続などの、有線媒体と、音響、RF、赤外線および他の無線媒体などの、無線媒体とを含むが、これらに限定されない。

20

30

【0101】

実施形態の例も方法を含む。その方法を、本明細書で説明される構造を含む、いくつかの方法で実装することができる。そのような 1 つの方法は、本明細書で説明されるタイプのデバイスをマシン動作によって行うことである。

【0102】

別の選択的方法は、一部を実行する 1 または複数の人間のオペレータと共に実行される方法の個々の動作のうちの 1 または複数に関する。この人間のオペレータは、互いに共同配置される必要はないが、それぞれは、唯一、プログラムの一部を実行するマシンと共に配置されなければならない。

【0103】

図 7 は、実施形態に従って、修正を文書コンポーネントレベルで制御することによってオンライン文書編集を同期するプロセスである、プロセス 700 の論理流れ図を示す。プロセス 700 を、オンライン文書サービスをクライアントに提供するアプリケーションサーバによって実装できる。

40

【0104】

プロセス 700 は、動作 710 から開始し、そこでオンライン文書アプリケーションサーバ(例えば、ウェブサーバのフロントエンド)は、ユーザから文書の要求を受信する。文書は、さまざまなフォーマットから構成されてよい。ユーザ要求を受信すると、アプリケーションサーバは、動作 720 において文書が記憶サーバでロックされるようにする要求ができる。文書をロックした後、アプリケーションサーバは、動作 730 において記憶

50

サーバから文書を読み出すことができる。記憶サーバは、文書をストリームとしてアプリケーションサーバに送信できる。アプリケーションサーバは、動作 740 において、ストリームを文書コンポーネントに変換し、そしてコンポーネントをグラフにロードできる。サーバは、読み出された文書を包含するグラフ修正を、ローカルに記憶された基本グラフと比較し、そしてコンポーネントの変更を任意に同期できる。

【0105】

動作 750 において、アプリケーションサーバは、どの文書コンポーネントがユーザによって変更されたかを判定できる。動作 760 において、保存動作は、ユーザが保存動作を呼び出した後で実行され得るし、アプリケーションサーバは、ユーザ変更に基づいてグラフのどのコンポーネントをロックするかを判定し、そして後続の動作 770 においてグラフの新しい修正を生成する間に、対応するコンポーネントの変更を記録できる。アプリケーションサーバは、動作 780 において、現在働いているコンポーネント上でユーザ編集が同時にならないように、文書ロックをローカルグラフ上のコンポーネントベースのロックに変更できる。クライアントアプリケーションは、文書のどのコンポーネントがユーザによってロックされかまたは動かされるかを示す通知を指示することができる。提供される情報は、ユーザ存在情報、およびコンポーネントの現在の作成者を示す作成者名も包含できる。さらに、文書レベルのロックもユーザに表示することができる。さらに、共作成者の存在情報は、共作成者が最後に編集したグラフのコンポーネントに記憶され得る。共作成者の存在情報は、編集されたコンポーネントを有するクライアントアプリケーションによって表示され得る。

10

20

【0106】

文書内の他の作成者の情報を示すメタデータを受信すると、現在のユーザは、ロックされたコンポーネントをさらに変更することができない。他の作成者が相手の変更をコミットするために保存する場合/時、それによって保持されたロックを任意に除去し、代わりに「リフレッシュ要求された」ロックに変えることができる。このロックは、異なる体裁を有するので、もはや他の作成者と関連しない。

【0107】

逆に、現在のユーザが編集アクションを実行しない場合、自分達の現在の作成者の存在情報は、(必要に応じて)文書内ロックの存在情報を記述するメタデータに付加するよことによって他の作成者に通信され得る。現在のユーザが自身の変更を保存する場合、現在のユーザによって保持されたロックを任意に除去する要求をサーバに送って、代わりに他の人に対する「リフレッシュ要求された」ロックに変えることができる。

30

【0108】

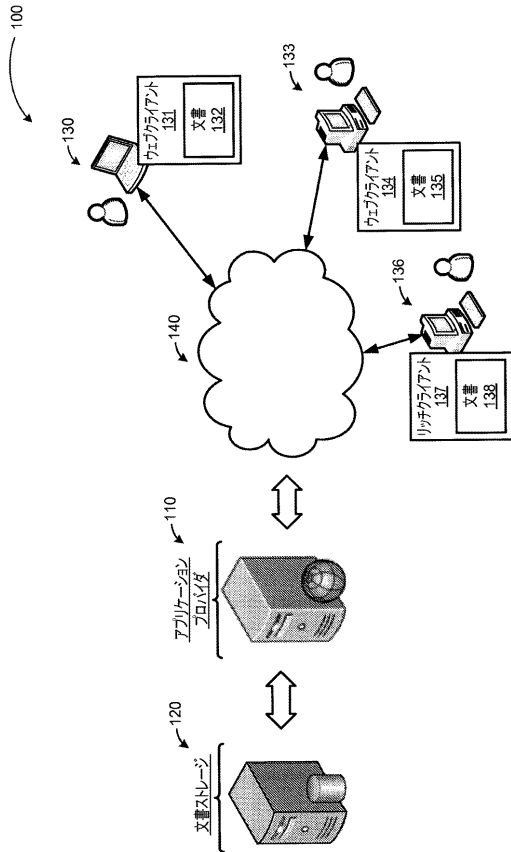
プロセス 700 に含まれる動作は、図解を目的とするものである。実施形態に従ったオンライン文書編集の同期は、本明細書で説明される発明の原理を使用して、ステップを少なくしたまたは付加した同様のプロセス、ならびに動作の異なる順序によって実装されてよい。

【0109】

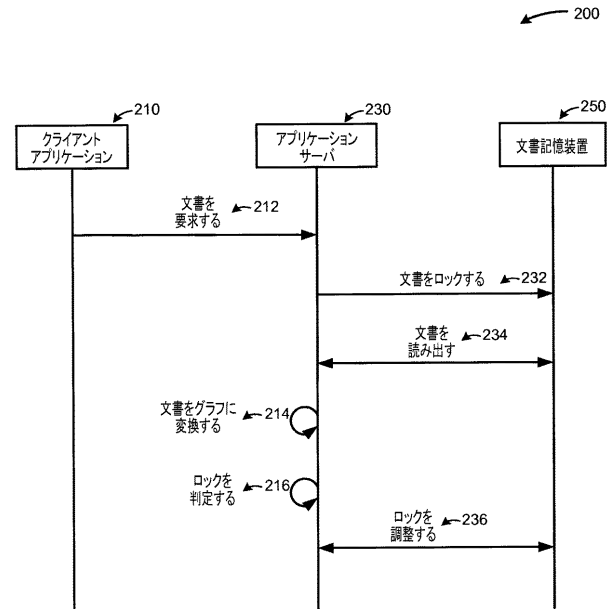
上記の明細書において、例およびデータは、製品の完全な説明および実施形態の構成の使用を提供する。発明の主題が構造的特徴および/または方法論的動作に固有の言語で説明されているが、添付の特許請求の範囲で定義される発明の主題は、必ずしも上記の固有の特徴または動作に限定されるものではないことを理解されたい。むしろ、上記の固有の言語および動作は、特許請求の範囲および実施形態を実装する例示的形態として開示される。

40

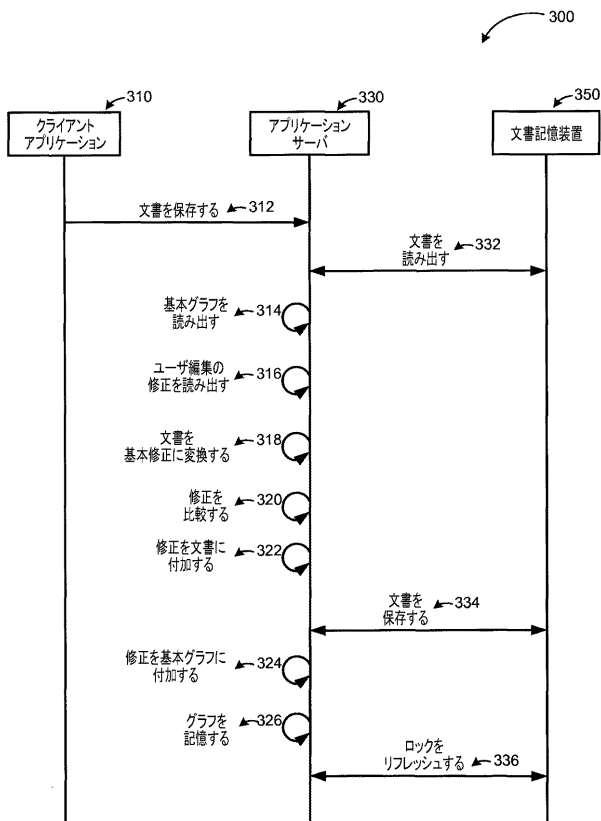
【図 1】



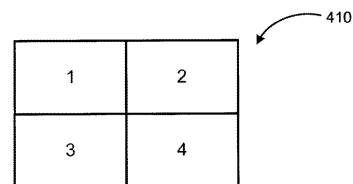
【図 2】



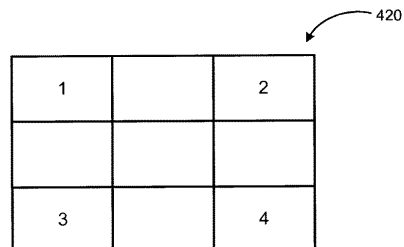
【図 3】



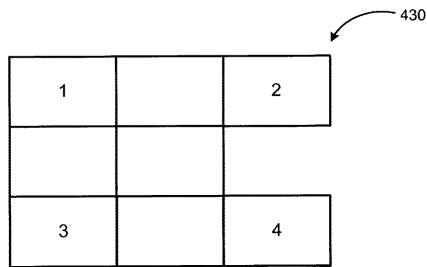
【図 4 A】



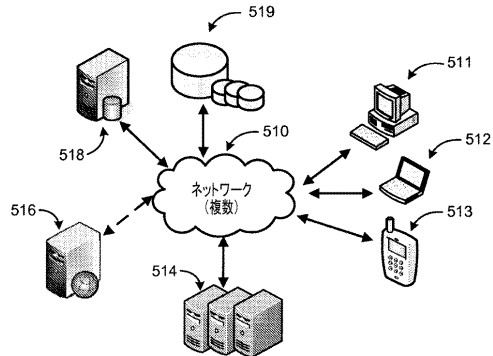
【図 4 B】



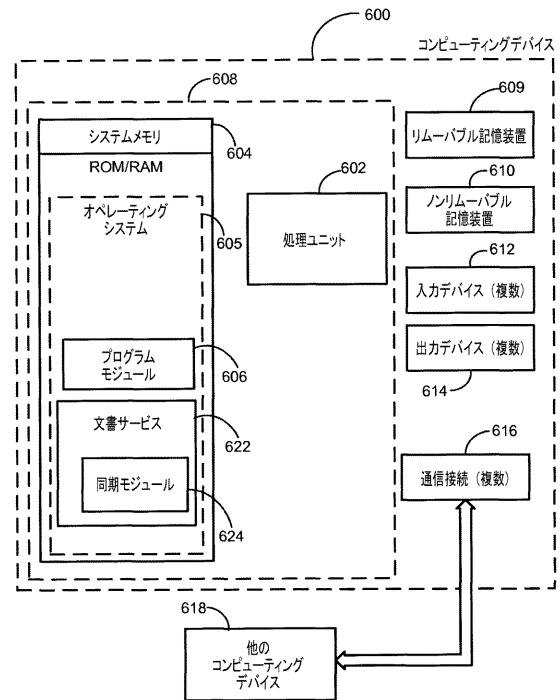
【図 4 C】



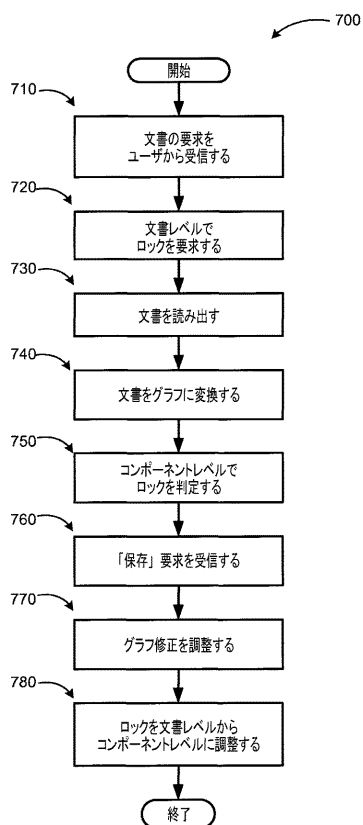
【図 5】





【図 6】



【図 7】



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2011/056969
A. CLASSIFICATION OF SUBJECT MATTER		
G06F 17/21(2006.01)i, G06F 15/16(2006.01)i, G06F 9/44(2006.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F 17/21; G06F 15/16; G06F 7/00; G09G 5/00		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords:online document, edit, coauthor<or>collaborate<or>cowork, sever<or>web<or>browser		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007-0143398 A1 (JEAN GRAHAM) 21 June 2007 See paragraph 51 - paragraph 65; paragraph 78 - paragraph 79;	1
A	paragraph 82 - paragraph 84; and figures 3a-6.	2-15
A	US 2008-0301228 A1 (FLAVIN ROBERT A.) 04 December 2008 See paragraph 24 - paragraph 30; paragraph 41 - paragraph 129; and figures 1-6.	1-15
A	US 2007-0130177 A1 (TINA SCHNEIDER et al.) 07 June 2007 See paragraph 35 - paragraph 49; paragraph 58 - paragraph 60; and figures 1,2,8.	1-15
A	US 2004-0021699 A1 (ULF FILDEBRANDT) 05 February 2004 See paragraph 14 - paragraph 31; and figures 1-3.	1-15
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 10 MAY 2012 (10.05.2012)		Date of mailing of the international search report 14 MAY 2012 (14.05.2012)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office Government Complex-Daejeon, 189 Cheongsu-ro, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer PARK, SANG HYUN Telephone No. 82-42-481-8263 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/056969

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0143398 A1	21.06.2007	US 7761591 B2	20.07.2010
US 2008-0301228 A1	04.12.2008	None	
US 2007-0130177 A1	07.06.2007	US 7996436 B2	09.08.2011
US 2004-0021699 A1	05.02.2004	US 7461346 B2	02.12.2008

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN

(74)代理人 100120112

弁理士 中西 基晴

(74)代理人 100147991

弁理士 鳥居 健一

(74)代理人 100119781

弁理士 中村 彰吾

(74)代理人 100162846

弁理士 大牧 綾子

(74)代理人 100173565

弁理士 末松 亮太

(74)代理人 100138759

弁理士 大房 直樹

(74)代理人 100091063

弁理士 田中 英夫

(72)発明者 アリ ターレガーニー

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ
マイクロソフト コーポレーション エルシーエー - インターナショナル パテント内

(72)発明者 トリスタン デイビス

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 エルシーエー - インターナショナル パテント内