

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5298136号
(P5298136)

(45) 発行日 平成25年9月25日(2013.9.25)

(24) 登録日 平成25年6月21日(2013.6.21)

(51) Int.Cl.

G06F 12/00 (2006.01)

F 1

G06F 12/00 531R
G06F 12/00 531J
G06F 12/00 520E

請求項の数 20 (全 18 頁)

(21) 出願番号	特願2010-542241 (P2010-542241)
(86) (22) 出願日	平成20年12月22日 (2008.12.22)
(65) 公表番号	特表2011-509489 (P2011-509489A)
(43) 公表日	平成23年3月24日 (2011.3.24)
(86) 國際出願番号	PCT/US2008/087925
(87) 國際公開番号	W02009/088727
(87) 國際公開日	平成21年7月16日 (2009.7.16)
審査請求日	平成23年12月7日 (2011.12.7)
(31) 優先権主張番号	11/971,206
(32) 優先日	平成20年1月8日 (2008.1.8)
(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	アンドリュー アール クーネオ 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 インターナショナル パテンツ内

最終頁に続く

(54) 【発明の名称】 ジャバスクリプトグリッドにおける非同期マルチレベル・アンドウのサポート

(57) 【特許請求の範囲】

【請求項 1】

ウェブアプリケーションを介するデータへの非同期の変更をトラッキングするための変更コンポーネントであって、前記非同期のデータ変更は非同期の認証または増加を必要とする変更である、変更コンポーネントと、

前記非同期のデータ変更のマルチレベルのアンドウ/リドウを行なうリバース操作を実行するためのリバースコンポーネントであって、前記リバースコンポーネントは明示的な変更および暗示的な変更を記憶するアンドウスタックを備え、アンドウ命令は明示的な変更が取り消されるまで前記アンドウスタックから変更を取り除きおよび取り消す、リバースコンポーネントと、

前記ウェブアプリケーションに記憶される前記データのテーブル状の表現を提供するためのデータグリッドであって、前記非同期のデータ変更を記録するためのセルを備え、前記非同期のデータ変更の編集および可視化をサポートし、前記データグリッドは前記非同期のデータ変更に関する前記ウェブアプリケーションへ通知を送信し、前記通知は他のデータ変更に関する前記グリッドにおける前記変更の順序づけを実行するオーダキーを含む、データグリッドと、

少なくとも前記変更コンポーネントおよび前記リバースコンポーネントに関連するコンピュータ実行可能命令を実行するプロセッサと

を備えることを特徴とするコンピュータに実装されるアンドウシステム。

【請求項 2】

前記データ変更がウェブページに関連付けられることを特徴とする請求項1に記載のシステム。

【請求項3】

前記ウェブアプリケーションは、ブラウザアプリケーションであることを特徴とする請求項1に記載のシステム。

【請求項4】

前記データ変更は、クライアントベースのデータグリッドを介して行われることを特徴とする請求項1に記載のシステム。

【請求項5】

前記データ変更は、グリッドを介して手動またはプログラムで行われることを特徴とする請求項4に記載のシステム。 10

【請求項6】

前記リバース操作は、少なくとも2つの前の変更に従ってデータを存在した状態に復元することを特徴とする請求項1に記載のシステム。

【請求項7】

前記ウェブアプリケーションは、変更の通知であって、他の変更に関する変更をドキュメントで記録するオーダキーを含む通知、を受け取ることを特徴とする請求項1に記載のシステム。

【請求項8】

前記変更は、サーバドキュメントに関連付けられ、クライアントベースのリバース操作のためのアンドウスタックに入力され、クライアントベースの変更トラッカにおいてトラッキングされることを特徴とする請求項1に記載のシステム。 20

【請求項9】

サーバのウェブアプリケーションのサーバドキュメント内の非同期のデータ変更を、前記非同期のデータ変更を記録するためのテーブル状の表現内のセルを含み前記非同期のデータ変更の編集および可視化をサポートするクライアントベースのグリッドを介して検出するステップと、

オーダキーを変更通知内の前記非同期のデータ変更に割り当てるステップであって、前記オーダキーは前記グリッドにて行なわれる前記非同期のデータ変更の関連する順序づけを示す、ステップと、 30

前記変更通知を非同期の認証のために前記グリッドから前記ウェブアプリケーションおよびサーバに送るステップと、

非同期の認証情報を前記通知に基づきサーバから受け取るステップと、

前記非同期の認証情報をオーダキーに従ってグリッド内に順序づけるステップと、

サーバドキュメントにおけるアンドウ／リドウ操作をオーダキーに基づき管理するステップであって、アンドウスタックは明示的な変更および暗示的な変更を記憶し、アンドウ命令は明示的な変更が取り消されるまで前記アンドウスタックから変更を取り除きおよび取り消す、ステップと、

を備えることを特徴とするデータのリバース操作を提供するためのコンピュータに実装される方法。 40

【請求項10】

前記グリッドにおける変更に基づきアンドウスタックおよび変更トラッカを介して変更をカスケードするステップをさらに備えることを特徴とする請求項9に記載の方法。

【請求項11】

アンドウスタック内の変更を記憶し、アンドウ／リドウ操作の一部として第1の明示的な変更に到達するまで変更を戻すステップをさらに備えることを特徴とする請求項9に記載の方法。

【請求項12】

明示的な変更を変更トラッカデータ構造内に記憶するステップをさらに備えることを特徴とする請求項9に記載の方法。 50

【請求項 13】

全ての默示的な変更を次の明示的な動作の前まで戻し、そして、次の明示的な動作を取り消すステップをさらに備えることを特徴とする請求項 9 に記載の方法。

【請求項 14】

クライアントウェブアプリケーションを介して、サーバのウェブドキュメントに対するデータ変更を開発するステップと、

順序立てられた変更の履歴として前記変更をクライアントアンドウスタック内に記憶し、データに直接影響を与える明示的な変更について変更エントリを別個のクライアント変更トラッカに記憶するステップであって、前記アンドウスタックは明示的な変更および暗示的な変更を記憶する、ステップと、

10

前記変更をサーバにおいて認証するステップと、

認証情報を、非同期にサーバからウェブアプリケーションに受け取るステップと、

前記アンドウスタック内の順序立てられた変更の履歴および前記変更トラッカ内の変更のエントリに基づき、ウェブドキュメントにおけるアンドウ／リドウ操作を管理するステップであって、アンドウ命令は明示的な変更が取り消されるまで前記アンドウスタックから変更を取り除きおよび取り消す、ステップと

を備えることを特徴とするデータのリバース操作を提供するためのコンピュータに実装される方法。

【請求項 15】

前記データ変更を顕在化させ、クライアントのデータグリッドを介して、前記データ変更を可視化するステップをさらに備えることを特徴とする請求項 14 に記載の方法。

20

【請求項 16】

ウェブアプリケーションにおいて実行される保存操作の前に起こった、ウェブドキュメントにおける変更をアンドウするステップをさらに備えることを特徴とする請求項 14 に記載の方法。

【請求項 17】

保存操作に応答して、変更を変更トラッカから読み出し、前記変更をデータソースにコミットし、変更トラッカをクリアするステップをさらに備えることを特徴とする請求項 14 に記載の方法。

【請求項 18】

30

順序立てられた変更の履歴は、各変更に割り当てられた変更キーに基づくことを特徴とする請求項 14 に記載の方法。

【請求項 19】

默示的な変更情報および明示的な変更情報を、默示的および明示的な変更のアンドウを実行する機能と共に、アンドウスタック内に記憶するステップをさらに備えることを特徴とする請求項 14 に記載の方法。

【請求項 20】

明在化された変更をセルレベルの変更の集合としてエクスポートするステップをさらに備えることを特徴とする請求項 14 に記載の方法。

【発明の詳細な説明】

40

【技術分野】**【0001】**

デスクトップクライアントアプリケーションと「シン」ウェブアプリケーションとの間の主な差別化要因の 1 つに、編集実績の豊富さがある。従来、ウェブアプリケーションでは、ユーザがアプリケーションをナビゲートする際にサーバにデータをコミットする個々のポストバックに従って、サーバとの間でデータが処理される。ユーザの動作をアンドウする（取り消す）ことは、ユーザが一旦ウェブページのナビゲートをやめてしまうと、多くの場合、不可能である。逆に、ユーザは、クライアントアプリケーションとはより流動的に対話可能であるが、ユーザがデータを保存しようとするときに保存できるだけである。さらに、ユーザがクライアントアプリケーションで作業中に編集で間違いをした場合、

50

ユーザは、1回または複数回「アンドゥ」を選択して、保存されているファイルには影響を与えることなく変更を元に戻すことができる。

【背景技術】

【0002】

今日、多くのソフトウェア技術において、ウェブ型のアプリケーションを介して企業のサーバ上でデータと対話する従業員を必要としている。例えば、プロジェクトサーバ内に通常見られる、構造化されたタスクデータの編集（例えば、追加／削除タスク、リソースの割り当て、スケジュールデータの変更等）を、ウェブページを介して行う場合を考える。

【発明の概要】

10

【発明が解決しようとする課題】

【0003】

マルチレベル・アンドゥの能力が無いと、ユーザが動作をアンドゥできないため、これは危険の高いものとなる。ユーザは、頻繁に保存を行うことができるが、そのようなデータセットをサーバに保存することは時間のかかる処理である。従って、効果的な編集性能は、複雑なデータをウェブページにタイプするためのユーザの快適性に比例して減少する。

【課題を解決するための手段】

【0004】

以下では、本明細書に記載されるいくつかの新規な実施形態の、基本的な理解を提供するために、簡素化された要約を示す。この要約は、広範囲な概要ではなく、かつ、重要／決定的な要素を確認することまたはその範囲を明確にすること、を意図していない。その唯一の目的は、後に示されるさらに詳細な記載の前置きとして、いくつかの概念を簡素化された形式で示すことである。

20

【0005】

開示されるマルチレベル・アンドゥアーキテクチャは、変更のトラッキングが、非同期な（および同期的な）シナリオで切れ目なく機能する、制御駆動の変更をカスケードするシステムである。さらに、アンドゥでは、保存動作を超えて、默示的変更と明示的変更との間の違いを理解し、これらの変更を処理し、それに従ってアンドゥのためのコンテキストが提供される。

30

【0006】

クライアントアプリケーションは、アプリケーションのデータを、テーブル状に表現するグリッドオブジェクトに関連付けられ、それによって、編集および可視化をサポートする。アプリケーションは、部分的には、グリッドオブジェクトをインスタンス化し構成する。アプリケーションは、グリッド内のデータに対する変更を、プログラムで開始することが可能であり、および／または、ユーザが、グリッド内のデータを直接編集することが可能である。変更の結果は、アプリケーションへの通知となり、通知にはオーダキーが含まれる。アプリケーションは、通知をコンシューム（消費）して、次に、オーダキーを使用して更新機能を呼び出すことにより、同期または非同期の計算（例えば、スケジューリング）に基づき、新しい変更を付け加えることができる。アプリケーションは、未来の任意の時点においてキーを使用して、この変更にさらなる更新を加えることができる。更新は、アンドゥ／リドゥのために、適切に収集される。

40

【0007】

前述のおよび関連する目的を達成するため、ある例示的態様が以下の記載および添付の図面と関連して本明細書において記載される。これらの態様は、本明細書に開示される原理を採用することが可能である種々の方法のうちのわずかを示し、全てのそのような態様およびその等価物を含むことを意図する。他の利点および新規の特徴は、図面と併せて考観されるときに以下の詳細な説明から明らかになるであろう。

【図面の簡単な説明】

【0008】

50

【図1】コンピュータに実装されるアンドウシステムを例示する図である。

【図2】マルチレベルのデータ変更のリバース操作を提供するクライアント・サーバシステムの実装を例示する図である。

【図3】データへの非同期のアンドウ / リドウ操作の例を例示する図である。

【図4】非同期の順不同変更の処理に対応する変更キーの使用に基づく、最終的なデータ更新を例示する図である。

【図5】変更コンポーネントの一部として変更トラッカ、および、リバースコンポーネントの一部としてアンドウスタック、を例示する図である。

【図6】グリッド内のデータの初期のグリッド状態、アンドウスタックのスタック状態、および変更トラッカのトラッカ状態、を例示する図である。

10

【図7】默示的変更と、スタック状態およびトラッカ状態への影響と、を例示する図である。

【図8】グリッドデータのタスク名の変更と、アンドウスタックおよび変更トラッカへの影響と、を例示する図である。

【図9】アンドウ操作の実行を例示する図である。

【図10】データのリバース操作を提供する、コンピュータに実装される方法を例示する図である。

【図11】明示的 / 默示的変更を処理するための方法を例示する図である。

【図12】データのリバース操作を提供する、コンピュータに実装される方法を例示する図である。

20

【図13】開示されるアーキテクチャに従ってマルチレベル・アンドウを実行すべく作動可能なコンピュータシステムのブロック図である。

【図14】マルチレベル・アンドウ処理のための例示のクライアント・サーバコンピュータ環境の概略ブロック図である。

【発明を実施するための形態】

【0009】

開示されるアーキテクチャでは、グリッドベースアプリケーションにおいてクライアント上でマルチレベル・アンドウをサポートすることにより、「シン」ウェブアプリケーションの編集経験における大きなギャップが埋められる。この特徴は多くのオンライン編集経験にも同様に有用である。例えば、マルチレベル・アンドウにより、ユーザは、一度により多くのデータを編集する際に、より快適に感じることができ、また、アプリケーションの有益な性能をユーザはより認知することができる。さらに、オンラインバージョンのアプリケーションでも、レコード内の複数の変更をアンドウおよびリドウすることができ、ユーザが編集について全体的に制御できる、という利点がある。

30

【0010】

ここで図面を参照すると、全体を通して、同様の参照番号が同様の要素の参照に使用される。以下の記載では、説明の目的で、多数の特定の詳細が、その全体的な理解のために記載される。しかし、新規な実施形態が、これらの特定の詳細が無くとも実践可能であることは明らかである。他の例において、周知の構造およびデバイスが、その説明を容易にするために、ブロック図の形式で示される。

40

【0011】

図1は、コンピュータに実装されるアンドウシステム100を例示する。システム100には、ウェブアプリケーション106を介するデータ104に対する非同期の変更をトラッキングする変更コンポーネント102が含まれる。システム100には、データ104に対するデータ変更を前の状態にするリバース操作（例えば、アンドウ、リドウ）を実行するための、リバースコンポーネント108がさらに含まれる。データ104に対する変更は、データグリッド110を介して行うことができる。データ104に対する変更は、手動またはプログラムで、グリッド110を介して行うことができる。

【0012】

アプリケーション106は、グリッド110をインスタンス化し、かつ構成する。グリ

50

ツド110は、アプリケーション106に記憶されるデータをテーブル状に表現したものであり、編集と可視化をサポートする。変更とは、前と後のセルのデータ、加えて、セルの値を変化させるのに要した動作をいう。

【0013】

一実施形態において、ウェブアプリケーション106は、ブラウザアプリケーションであり、これを介して、サーバによりホストされるウェブページに対して変更がなされる。そして、ユーザは、ウェブページドキュメントに対してなされる変更に、マルチレベル・アンドゥ／リドゥのリバース操作を行うことができる。別の実装において、ウェブアプリケーション106は、ユーザがネットワークデータよりむしろローカルデータと対話し、かつ、マルチレベル・アンドゥ／リドゥ等のリバース操作を行うことを可能にする、ブラウザである。10

【0014】

図2は、マルチレベルのデータ変更のリバース操作を提供するクライアント・サーバシステム200の実装を例示する。システム200は、図1のシステム100を示し、ウェブアプリケーション106を介するデータ104に対する非同期の変更をトラッキングする変更コンポーネント102と、データ変更を前の状態にするリバース操作（例えば、アンドゥ、リドゥ）を実行するためのリバースコンポーネント108と、がある。データ変更は、手動および／またはプログラムでデータグリッド110を介して行うことができる。ここで、データ変更が、ウェブサーバ202のウェブドキュメント（例えば、ウェブページ）200に加えられる。グリッド110に編集が通ると、変更通知が非同期の認証のためにアプリケーション106を介してサーバ202に送られる。一度認証されると、変更は、データ104に対する更新として、アプリケーション106を介してグリッド110に戻される。20

【0015】

図3は、データへの非同期のアンドゥ／リドゥ操作の例300を例示する。非同期の変更は、任意の数の非同期の認証または増加を必要とする変更である。エンティティがグリッドデータに対して変更を加えると、この変更の結果がアプリケーション106への通知となる。この通知に含まれるものは、オーダ（または変更）キーである。アプリケーション106では、通知をコンシュームして、次に、そのオーダキーで更新機能を呼び出すことにより、いくつかの同期または非同期の計算（例えば、スケジューリング）に基づき、新しい変更を付け加えることができる。アプリケーション106は、未来の任意の時点においてオーダキーを自由に使用して、この変更にさらなる更新を加える。これらの更新は、アンドゥ／リドゥのために、適切に収集される。30

【0016】

ここで、非同期の認証を必要とする、AおよびBで示される2つの変更が、グリッド110に入力される。編集が行われると、変更はグリッド110により順番にキャプチャされる。各変更は、オーダキーでタグ付けされる。例えば、変更Aは、オーダキーAでタグ付けされ（また、変更キーAでも示される）、第2の変更B（時間的に変更Aよりも後）が変更キーBで示される。データ変更が行われると、グリッド110はこれを検出し、通知をアプリケーション106（例えば、ブラウザ）に送り、アプリケーション106が今度はその通知を認証のためにサーバ202に送る。サーバ202における認証処理は、順不同で発生させることができあり、または一旦認証が完了すると、サーバは前に指示された認証を順不同で送ることができる。従って、オーダキーにより、グリッド110における変更の指示が容易にされる。40

【0017】

本例において、第1の変更通知要求302が、グリッド110からアプリケーション106へ送られて、第2の行の、「期間」とラベル付けされるフィールドを有する列に対して行われた、期間の新しい値が5日間と設定されるデータ変更Aを報告する。それに続くデータ編集では、グリッド110は、第2の変更通知要求304をアプリケーション106に送り、第3の行の、「期間」とラベル付けされるフィールドを有する列に対して行わ50

れた、期間の新しい値が7日間と設定されるデータ変更Bを報告する。変更通知要求(302および304)は、アプリケーション106に順番に送られる(例えば、変更Aの後に変更B)。アプリケーション106は次に、通知要求(302および304)をサーバ202に転送し、サーバ202は、変更要求(302および304)を非同期に認証し、認証をアプリケーション106に戻す。

【0018】

ここで、サーバ202は、第1の変更通知要求302に対する第1の認証処理306を開始する。次に、サーバ202は、第2の変更通知要求304に対する第2の認証処理308を受け取り、開始する。サーバ202は、第1の認証処理306の前に第2の認証処理308を完了する。従って、第2の更新応答310が、関連するデータを更新するために、サーバ202からアプリケーション106を介してグリッド110に送られる。第2の更新応答310には、データ変更が第1のデータ変更Aに対応して行われた順番を示すオーダキーBが含まれる。第2の更新応答310にはまた、変更が、第3の行の、「終了日」とラベル付けされるフィールドで行われ、終了日の新しい値が5/27である、ということも含まれる。これは、第2の変更通知要求304内の新しい値7日間に対応する。

10

【0019】

サーバ202は、次に、第1の認証処理306を完了し、関連するデータを更新するために、第1の更新応答312を、アプリケーション106を介してグリッド110に送る。第1の更新応答312には、データ変更が第2のデータ変更Bに対応して行われた順番を示すオーダキーAが含まれる。第1の更新応答312にはまた、変更が、第2の行の、「終了日」とラベル付けされるフィールドで行われ、終了日の新しい値が5/25である、ということも含まれる。これは、第1の変更通知要求302内の新しい値5日間に対応する。従って、変更は、順不同で認証から戻される(変更Bの後にA)。

20

【0020】

図4は、非同期の順不同変更処理に対処する変更キーに基づく、最終的なデータ更新を例示する。最後の保存操作後に起こったデータ変更への要求の結果400を示す。結果は、サーバ202から戻されると、実際に事象が起こった順番(変更Bの後に変更A)ではなく、変更(またはオーダ)キーが指示する論理的順番(変更Aの後に変更B)で配置され、記憶される。従って、変更Aに関する変更402が、一緒に記憶され、変更Bの前に起こるものとして示される。同様に、変更Bに関する変更404が、一緒に記憶され、変更Aの後に起こるものとして示される。

30

【0021】

アンドゥおよびリドゥのリバース操作は、このように操作もされる。従って、ユーザがアンドゥを選択した場合、変更Bに関連する両方の変更(期間および終了日)は、それが変更キーBにより示されるようにユーザが行った最後の変更であるため、たとえ、変更Aに対する更新が、グリッドが遭遇した最後の事象であったとしても、アンドゥされるであろう。

【0022】

エンティティ(例えば、ユーザまたはシステム)は、グリッドと対話して、明示的または默示的に、グリッド内のデータに対して変更を行う。明示的な変更は、データに直接影響するグリッドに対して行われる変更である(例えば、タスクの開始日を変更する)。默示的な変更は、データに影響を与えないグリッドに対して行われる変更である(例えば、列のサイズを変更する)。

40

【0023】

図5は、変更コンポーネント102の一部として変更トラッカ500、および、リバースコンポーネント108の一部としてアンドゥスタック502、を例示する。エンティティがグリッドデータに対して変更を行うと、各変更の型が記録され、実行およびアンドゥするための機能がアンドゥスタック502に配置される。アンドゥスタック502は、動作の最後の組の順序立てられた履歴が記憶されるデータ構造である。アンドゥ事象が起こると、動作がアンドゥスタック502から取り除かれ、第1の明示的な変更に遭遇するま

50

で戻される。第1の明示的な変更は、戻された後の最後の動作である。この時点で、アンドウ操作が、終了する。これに続くアンドウ操作は、同じ方法で処理される。アンドウ命令では、次の明示的な動作の前の全ての默示的な動作を取り消し、そして、明示的な動作を取り消す。

【0024】

このシステムの効果は、アンドウ操作の後、グリッドの視覚的状態が、アンドウされた動作の前まで復元されることである。全ての明示的な変更は、アンドウスタック502から別個の構造（変更トラッカ500）に記憶される。この変更トラッカ500を、セルレベルの変更の集合としてエクスポートすることができる。

【0025】

アンドウスタック502および変更トラッカ500は、独立した構造である。変更が起こると、その変更は、アンドウスタック502に押し入れられ、変更が明示的である場合は、その変更のエントリが変更トラッカ500に作成される。変更が默示的である場合、変更は、変更トラッカ500にログをとられる。

【0026】

アンドウ事象が起こると、変更は、アンドウスタック502および変更トラッカ500から取り除かれ、新しい値が、変更トラッカ500に記録される。変更が変更トラッカ500に存在する場合、その変更は取り除かれる。変更が変更トラッカ500に存在しない場合、変更を取り消す動作（これそのものが変更である）が変更ログに追加される。保存操作が起こると、変更は、変更トラッカ500から読みだされ、データソースにコミットされて、変更トラッカ500がクリアされる。

【0027】

このシステムの効果は、変更をアンドウするのに必要な情報が、アンドウスタック502に記憶されるため、エンティティが、保存事象の前に起こった動作をアンドウすることが可能であることである。同様に、変更トラッカ500は、アンドウスタック502が無い場合に機能することが可能である。

【0028】

図6から9は、グリッドデータ、アンドウスタック、および変更トラッカの状態で起こる変更を例証するための一連の図を例示する。図6は、グリッド内のデータの初期のグリッド状態600、アンドウスタックのスタック状態602、および、変更トラッカのトラッカ状態604を示す。変更は、グリッドデータの行2および3で起こる。

【0029】

エンティティは次に、フェンスのペンキ塗りには、1日ではなく2日かかると決めるこにより明示的な変更を行う。変更グリッド状態606には、期間の列の第2の行を2日間とする、変更が含まれる。プロジェクト管理アプリケーションの一部としてのプロジェクトの場合、例えば、期間の変更は、「終了」とも称する終了日の変更を引き起こす（カスケード変更）。カスケードトランザクションが、両方の変更を包含して作成される。従って、終了の列の第2の行のフィールド情報が、変更後のグリッド状態606に示すように、5/21/x月曜日から5/22/x火曜日に変更される。

【0030】

これらの期間および終了の変更は、スタック状態608に示すように、アンドウスタックに押し入れられる。加えて、トラッカ状態610には、変更が反映される。図7は、默示的変更と、スタック状態およびトラッカ状態への影響と、を示す。ここで、エンティティは、終了の列を非表示にし、これにより、グリッド状態612に変わり、このことで、変更がアンドウスタックに押しいれられ、スタック状態614に変わるが、変更トラッカのトラッカ状態610は変わらない。エンティティは、プロジェクトを保存し、このことで、変更トラッカのトラッカ状態610がクリアされてスタック状態616になり、グリッドのデータはここでサーバデータと一致する。

【0031】

図8は、グリッドデータのタスク名の変更と、アンドウスタックおよび変更トラッカへ

10

20

30

40

50

の影響と、を例示する。エンティティが、行3のタスク名を「道具の後片付け」から「プロジェクトの後片付け」に変更したことに基づき、グリッド状態618を変更する。スタック状態620には、スタックに押し入れられた変更が反映され、トラッカにはトラッカ状態622が反映される。

【0032】

図9は、アンドウ操作の実行を例示する。スタック状態622およびトラッカ状態626（前回からの変更無し）に示すように、アンドウでは、アンドウスタックおよび変更トラッカから変更をポッピングすることにより、最後の明示的な変更を取り消す。エンティティは次に、アンドウをもう一度実行し、これにより默示的変更および残っている明示的変更の両方が取り消される。このカスケード効果により、変更がアンドウスタックから取り除かれてスタック状態628になり、逆の変更が変更トラッカに加えられトラッカ状態630になり、グリッド状態がグリッド状態632に戻されて、図6のグリッド状態600と同じものになる。

10

【0033】

次のものは、開示されるアーキテクチャの新規な態様を実行するための例示の方法論を代表して表す一連のフローチャートである。説明を容易にする目的で、例えば、フローチャートまたはフロー図の形式で本明細書に示される1つまたは複数の方法論が、一連の動作として示され説明されるが、方法論は動作の順番により制限されず、ある動作は、その方法論に従って、本明細書に示され説明されるものとは異なる順番および／または他の動作と一緒に起こる、ことを理解および認識すべきである。例えば、当業者は、あるいは方法論が、状態図におけるもののように、一連の相互に関連する状態および事象として表されることを理解および認識するであろう。さらに、方法論において例示される全ての動作が、新規な実装のために必要であるわけではない。

20

【0034】

図10は、データのリバース操作を提供する、コンピュータに実装される方法を例示する。1000にて、サーバのサーバドキュメントにおけるデータ変更が、クライアントベースのグリッドを介して検出される。1002にて、オーダキーが、変更通知内の変更に割り当てられる。1004にて、変更通知が認証のためにサーバに送られる。1006にて、非同期の認証情報が、通知に基づきサーバから受け取られる。1008にて、認証情報が、オーダキーに従ってグリッド内に順序づけされる。1010にて、サーバドキュメントにおけるアンドウ／リドウ操作が、オーダキーに基づき管理される。

30

【0035】

図11は、明示的／默示的変更を処理する方法を例示する。1100にて、データの変更は、グリッドを介して受け取られる。1102にて、変更の型、および、変更をアンドウ／リドウする機能が、アンドウスタックに記憶される。1104にて、リバース操作（例えば、リドウ、アンドウ）が受け取られる。1106にて、アンドウ操作が、スタックから取り除かれ、第1の明示的な変更（取り消される最後の動作）に遭遇するまで戻される。

【0036】

図12は、データのリバース操作を提供する、コンピュータに実装される方法を例示する。1200にて、データ変更が、クライアントウェブアプリケーションを介してサーバのウェブドキュメントに対して開始される。1202にて、変更は、順序立てられた変更の履歴としてクライアントのアンドウスタックに記憶され、変更のエントリがクライアントの変更トラッカに記憶される。1204にて、変更は、サーバで認証される。1206にて、認証情報は、非同期にサーバからウェブアプリケーションに受け取られる。1208にて、ウェブドキュメントにおけるアンドウ／リドウ操作は、アンドウスタック内の順序立てられた変更の履歴および変更トラッカ内の変更のエントリに基づき、クライアントにて管理される。

40

【0037】

本用途において使用されるとき、用語「コンポーネント」および「システム」は、コン

50

ピュータ関連のエンティティ、すなわち、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、または実行中のソフトウェアのいずれかを参照することを意図する。例えば、コンポーネントは、プロセッサ上で稼働中の処理、プロセッサ、ハードディスクドライブ、(光および/または磁気記憶媒体の)複数の記憶ドライブ、オブジェクト、実行ファイル、実行のスレッド、プログラム、および/または、コンピュータとすることができるが、これに限定されない。例示の目的で、サーバ上で稼働するアプリケーションおよびサーバの両方を、1つのコンポーネントとすることができます。1つまたは複数のコンポーネントは、処理および/または実行のスレッド内に存在させることができ、1コンポーネントは、1つのコンピュータ上に配置すること、および/または、2つまたはそれ以上のコンピュータ間に分散させること、が可能である。

10

【0038】

ここで図13を参照すると、開示されるアーキテクチャに従ってマルチレベル・アンドウを実行すべく作動可能なコンピュータシステム1300のブロック図が例示される。その種々の態様の追加のコンテキストを提供するために、図13および以下の検討では、種々の態様を実装可能な適切なコンピュータシステム1300の、簡単な、一般的な説明を提供することが意図される。上記の説明は、1つまたは複数のコンピュータ上で稼働することができるコンピュータ実行可能命令の一般的なコンテキストでなされたが、当業者は、新規な実施形態はまた、他のプログラムモジュールの組み合わせで、および/または、ハードウェアとソフトウェアとの組み合わせとして、実装することが可能であることを認識するであろう。

20

【0039】

一般に、プログラムモジュールには、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する、ルーチン、プログラム、コンポーネント、データ構造等が含まれる。さらに、当業者は、理解するであろうが、発明の方法は、シングルプロセッサコンピュータシステムまたはマルチプロセッサコンピュータシステム、ミニコンピュータ、メインフレームコンピュータ、および、パーソナルコンピュータ、ハンドヘルドコンピュータデバイス、マイクロプロセッサベースまたはプログラム可能な家庭用電化製品等の、それぞれが1つまたは複数の関連するデバイスに動作可能に連結することができるもの、を含む、他のコンピュータシステム構成と共に実践可能である。

30

【0040】

例示される態様はまた、特定のタスクが通信ネットワークを介してリンクされるリモート処理装置により実行される、分散コンピュータ環境において実践可能である。分散コンピュータ環境において、プログラムモジュールは、ローカルおよびリモートのメモリ記憶装置の両方に配置することができる。

【0041】

コンピュータには、典型的には、種々なコンピュータ可読媒体が含まれる。コンピュータ可読媒体は、コンピュータによりアクセス可能であり、揮発性媒体および不揮発性媒体、着脱可能媒体および着脱不可能媒体を含む、任意の利用可能な媒体とすることができる。制限ではなく例として、コンピュータ可読媒体には、コンピュータ記憶媒体および通信媒体を含むことができる。コンピュータ記憶媒体には、任意の方法、または、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等の情報を記憶するための技術により実装される、揮発性および不揮発性、着脱可能および着脱不可能な媒体が含まれる。コンピュータ記憶媒体には、RAM、ROM、EEPROM、フラッシュメモリもしくは他のメモリ技術、CD-ROM、DVD(digital Versatile disk)もしくは他の光学ディスク記憶装置、磁気力セット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、または、所望の情報を記憶するのに使用可能な、かつ、コンピュータによりアクセス可能な、任意の他の媒体、が含まれるが、それに限定されない。

40

【0042】

再度図13を参照すると、種々の態様を実装するための例示のコンピュータシステム1

50

300には、プロセシングユニット1304、システムメモリ1306、およびシステムバス1308を有するコンピュータ1302が含まれる。システムバス1308により、システムメモリ1306を含むがこれに限定されないシステムコンポーネントのためのインターフェースがプロセシングユニット1304に提供される。プロセシングユニット1304は、任意の種々の市販のプロセッサとすることができます。デュアルマイクロプロセッサおよび他のマルチプロセッサーアーキテクチャも、プロセシングユニット1304として採用することができる。

【0043】

システムバス1308は、メモリバス（メモリコントローラを用いてまたは用いずに）、周辺機器用バス、および、任意の様々な市販のバスアーキテクチャを使用するローカルバスに、さらに相互接続され得る、任意のいくつかのタイプのバス構造とすることができます。システムメモリ1306には、不揮発性メモリ（N O N - V O L ）1310および/または揮発性メモリ1312（例えば、R A M（r a n d o m a c c e s s m e m o r y））を含むことができる。基本入出力システム（B I O S）は、不揮発性メモリ1310（例えば、R O M、E P R O M、E E P R O M等）に記憶することが可能であり、B I O Sは、起動中などにコンピュータ1302内の要素間の情報の転送を支援する基本ルーチンを記憶する。揮発性メモリ1312にはまた、データをキャッシングするためのスタティックR A M等の高速R A Mを含み事ができる。

【0044】

コンピュータ1302には、内部ハードディスクドライブ（H D D）1314（例えば、E I D E、S A T A）がさらに含まれ、内部H D D1314はまた、適切な筐体、磁気フロッピーディスクドライブ（F D D）1316、（例えば、着脱可能ディスクケット1318への読み書きのため）および光ディスクドライブ1320、（例えば、C D - R O Mディスク1322を読み込み、またはD V D等の他の大容量光媒体への読み書きのため）における、外部使用のために構成することができる。H D D1314，F D D1316、および光ディスクドライブ1320は、それぞれH D Dインターフェース1324、F D Dインターフェース1326、および光ドライブインターフェース1328により、システムバス1308に接続させることができる。外部ドライブ実装のためのH D Dインターフェース1324には、U S B（U n i v e r s a l S e r i a l B u s）およびI E E E1394インターフェース技術の少なくとも一方または両方を含むことができる。

【0045】

ドライブおよび関連するコンピュータ可読媒体は、データの不揮発性記憶、データ構造、コンピュータ実行可能命令等を提供する。コンピュータ1302では、ドライブおよび媒体が適切なデジタルフォーマットで任意のデータの記憶を受け入れる。コンピュータ可読媒体の上記の説明では、H D D、着脱可能磁気ディスクケット（例えば、F D D）、およびC DまたはD V D等の着脱可能な光媒体に言及したが、コンピュータが可読の、Z I Pドライブ、磁気カセット、フラッシュメモリカード、カートリッジ等の他のタイプの媒体もまた例示の操作環境において使用することことができること、およびさらに、任意のそのような媒体に、開示されるアーキテクチャの新規の方法を実行するためのコンピュータ実行可能命令を含むことが可能であることは、当業者には理解されるであろう。

【0046】

オペレーティングシステム1330、1つまたは複数のアプリケーションプログラム1332、他のプログラムモジュール1334、およびプログラムデータ1336を含む、多数のプログラムモジュールを、ドライブおよび揮発性メモリ1312に記憶することができる。1つまたは複数のアプリケーションプログラム1332、他のプログラムモジュール1334、およびプログラムデータ1336には、例えば、変更コンポーネント102、データ104、ウェブアプリケーション106、リバースコンポーネント108、データグリッド110、変更通知（302および304）、更新応答（310および312）、変更トラックカ500、アンドウスタック502、グリッド状態（600、606、612、618、および632）、スタック状態（602、608、614、620、6250）

4および628)、およびトラッカ状態(604、610、616、622、626および630)を含むことができる。

【0047】

オペレーティングシステム、アプリケーション、モジュール、および/またはデータの全てまたは一部はまた、揮発性メモリ1312にキャッシュすることができる。開示されるアーキテクチャは、種々の市販のオペレーティングシステムまたはオペレーティングシステムの組み合わせを用いて実装可能であることは理解されるであろう。

【0048】

ユーザは、コマンドおよび情報を、1つまたは複数の有線/無線の入力デバイス、例えば、キーボード1338およびマウス1340等のポインティングデバイスを介して、コンピュータ1302に入力することができる。他の入力デバイス(図示せず)には、マイクロフォン、IRリモートコントロール、ジョイスティック、ゲームパッド、スタイルスペン、タッチスクリーン等を含むことができる。これらおよび他の入力デバイスは、システムバス1308に連結される入力デバイスインターフェース1342を介してプロセシングユニット1304に接続されることが多いが、パラレルポート、IEEE1394シリアルポート、ゲームポート、USBポート、IRインターフェース等の他のインターフェースにより接続することができる。

10

【0049】

モニタ1344または他のタイプのディスプレイデバイスも、ビデオアダプタ1346等のインターフェースを介してシステムバス1308に接続される。モニタ1344に加えて、コンピュータは典型的には、スピーカ、プリンタ等の他の周辺出力デバイス(図示せず)が含まれる。

20

【0050】

コンピュータ1302は、有線および/または無線の通信を介したリモートコンピュータ(群)1348等の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て、ネットワーク化環境において作動することができる。リモートコンピュータ(群)1348は、ワークステーション、サーバコンピュータ、ルータ、パーソナルコンピュータ、携帯用コンピュータ、マイクロプロセッサベースの娛樂用電化製品、ピアデバイス、または他の共通ネットワークノードとすることが可能であり、かつ、典型的には、コンピュータ1302に関して説明した要素の多くまたは全てを含むが、簡潔にするため、メモリ/記憶装置1350のみが例示される。示される論理接続には、ローカルエリアネットワーク(LAN)1352および/またはより大きなネットワーク、例えば、ワイドエリアネットワーク(WAN)1354への有線/無線接続性が含まれる。そのようなLANおよびWANネットワーク環境は、オフィスおよび企業においては一般的なものであり、インターネット等の企業規模のコンピュータネットワークを容易にするものであるが、それらの全てがグローバル通信ネットワーク、例えば、インターネットに接続することができる。

30

【0051】

LANネットワーク環境において使用されるとき、コンピュータ1302は、有線および/または無線通信ネットワークインターフェースまたはアダプタ1356を介してLAN1352に接続される。アダプタ1356は、LAN1352への有線および/または無線通信を容易にすることが可能であり、LAN1352にはまた、アダプタ1356の無線機能性と通信するための、無線アクセスポイントを配設することができる。

40

【0052】

WANネットワーク環境において使用されるとき、コンピュータ1302は、モデル1358を含むことが可能であり、または、WAN1354上の通信サーバに接続され、または、インターネット等によりWAN1354上の通信を確立するための他の手段を有する。モデル1358は、内部的または外部的であり、および有線および/または無線のデバイスであり、入力デバイスインターフェース1342を介してシステムバス1308に接続される。ネットワーク化環境において、コンピュータ1302に関連して示されるプログラムモジュール、またはその一部分は、リモートメモリ/記憶装置1350に記憶す

50

ることが可能である。示されるネットワーク接続は、例示のものであり、コンピュータ間の通信リンクを確立する他の手段が使用可能であることは、理解されるであろう。

【0053】

コンピュータ1302は、例えば、プリンタ、スキャナ、デスクトップおよび／または携帯用コンピュータ、PDA（携帯情報端末）、通信衛星、無線で検出可能なタグが関連づけられる任意の設備または場所（例えば、売店、新聞売店、トイレ）、および電話、との無線通信（例えば、IEEE802.11無線変調技術）において作動可能に配置される、無線デバイス等の、IEEE802系の規格を使用して、有線および無線デバイスまたはエンティティと通信すべく作動可能である。これには、少なくともWi-Fi（またはワイヤレス・フィディリティ）、ワimax、およびブルートゥース（登録商標）無線技術が含まれる。従って、通信は、従来のネットワークまたは少なくとも2つのデバイス間の単なるアドホック通信を用いるような、所定の構造とすることができます。Wi-Fiネットワークは、IEEE802.11x（a、b、g等）と呼ばれる無線技術を使用して、セキュアな、信頼性のある、速い無線接続性を提供する。Wi-Fiネットワークを使用して、コンピュータをお互いに、インターネットに、および有線ネットワーク（IEEE802.3関連の媒体および機能を使用する）に接続することが可能である。10

【0054】

ここで図14を参照すると、マルチレベル・アンドウ処理のための例示のクライアント・サーバコンピュータ環境1400の概略ブロック図が例示される。環境1400には、1つまたは複数のクライアント（群）1402が含まれる。クライアント（群）1402は、ハードウェアおよび／またはソフトウェア（例えば、スレッド、プロセス、コンピュータデバイス）とすることができます。クライアント（群）1402は、例えば、クッキーおよび／または関連する状況的情報を収容することができます。20

【0055】

環境1400にはまた、1つまたは複数のサーバ（群）1404が含まれる。サーバ（群）1404はまた、ハードウェアおよび／またはソフトウェア（例えば、スレッド、プロセス、コンピュータデバイス）とすることができます。サーバ1404は、スレッドを収容して、例えば、アーキテクチャを採用して変換を行うことができる。クライアント1402とサーバ1404との間の1つの可能な通信は、2つまたはそれ以上のコンピュータプロセス間で伝送されるよう適合されるデータパケットの形式とすることができます。データパケットは、例えば、クッキーおよび／または関連する状況的情情報を収容することができます。環境1400には、クライアント（群）1402とサーバ（群）1404との間の通信を容易にするために採用することが可能な通信フレームワーク1406（例えば、インターネット等のグローバル通信ネットワーク）が含まれる。30

【0056】

通信は、有線（光ファイバを含む）および／または無線技術を介して容易にすることが可能である。クライアント（群）1402は、クライアント（群）1402にローカルな情報（例えば、クッキーおよび／または関連する状況的情報）を記憶するよう採用され得る1つまたは複数のクライアントデータストア（群）1408に作動可能に接続される。同様に、サーバ（群）1404は、サーバ1404にローカルな情報を記憶するよう採用され得る1つまたは複数のサーバデータストア1410に作動可能に接続される。40

【0057】

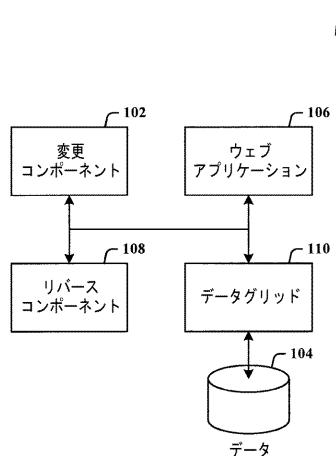
クライアント（群）1402には、ウェブアプリケーション106を含むことが可能であり、クライアントデータストア（群）1408には、データ104を含むことが可能である。サーバ（群）1404には、サーバ202および非同期の認証処理（306および308）を含むことが可能であり、サーバデータストア1410には、ドキュメント200を含むことが可能である。

【0058】

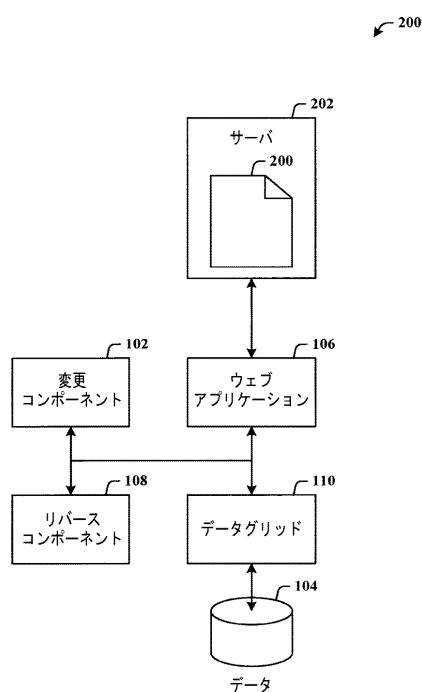
上記で説明したものには、開示されるアーキテクチャの例が含まれる。もちろん、コンポーネントおよび／または方法論の全ての考え得る組み合わせを説明することはできない50

が、当業者は、多くのさらなる組み合わせおよび置換が可能であることを認識するであろう。従って、新規なアーキテクチャが、添付の請求項の精神および範囲にある全てのそのような代替、修正、および変動を包含することが意図される。さらに、用語「含む」が、発明を実施するための形態または特許請求の範囲において使用される限りにおいて、そのような用語は、用語「備える」が特許請求の範囲において遷移語として採用される際に解釈されるように、「備える」と同様に包括的であることが意図される。

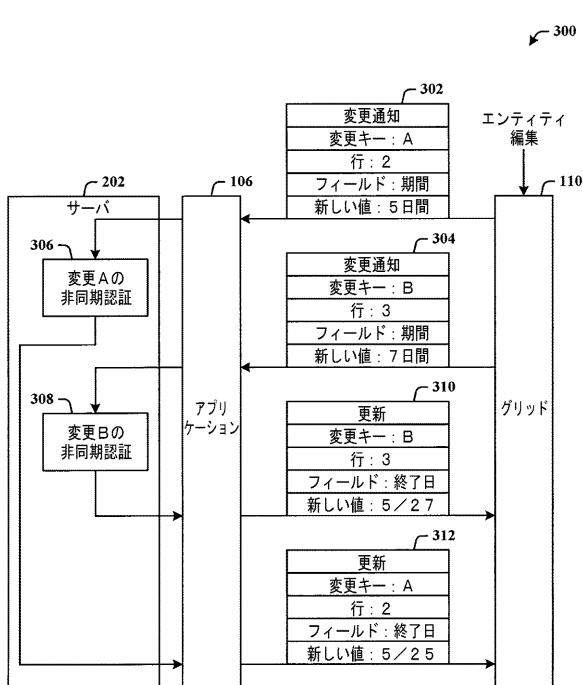
【図1】



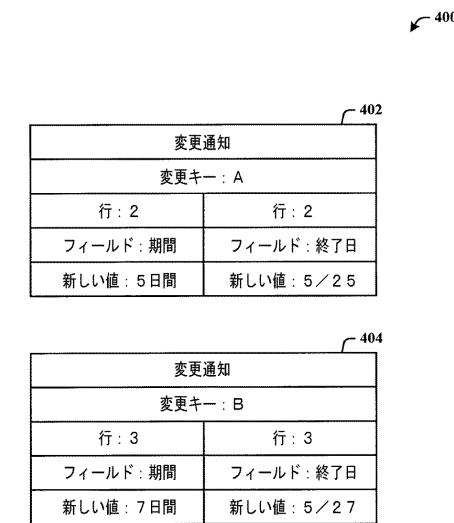
【図2】



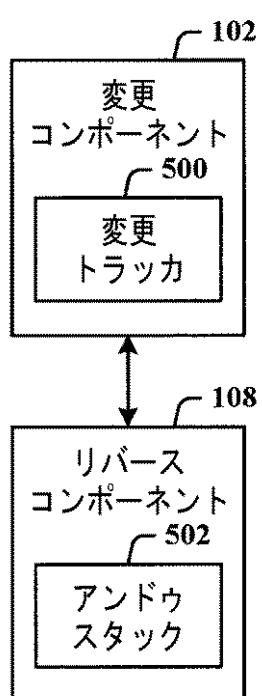
【図3】



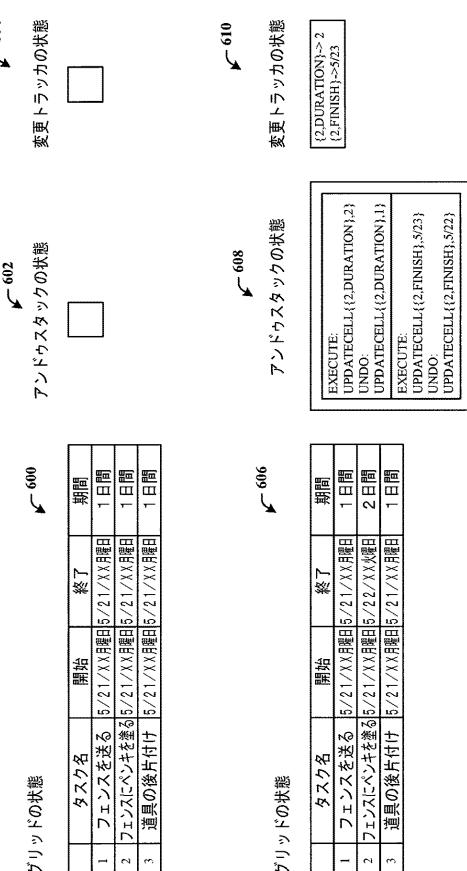
【図4】



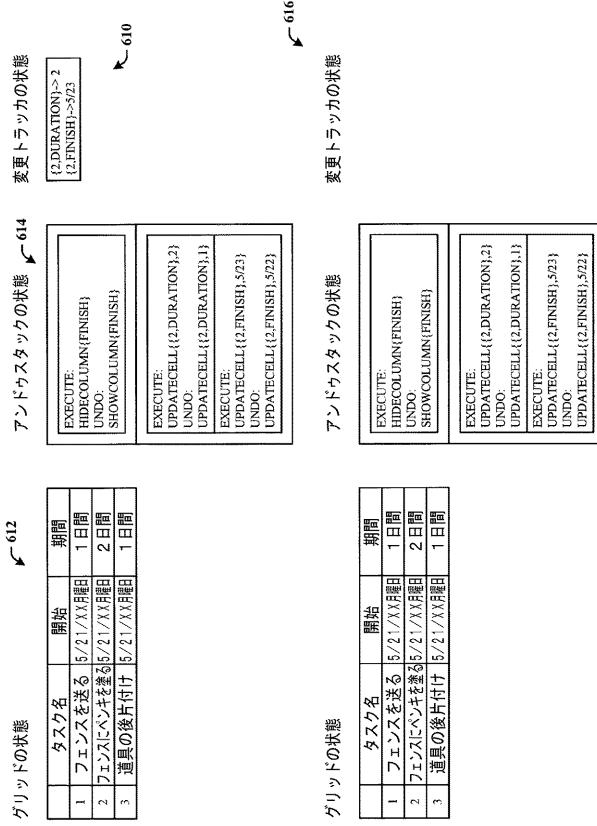
【図5】



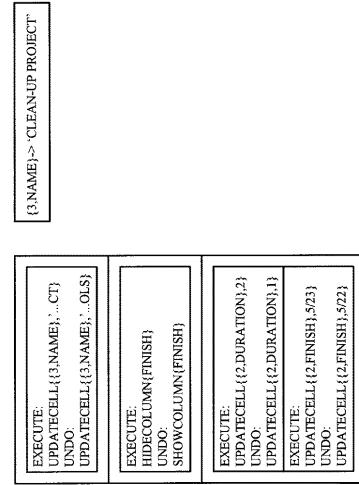
【図6】



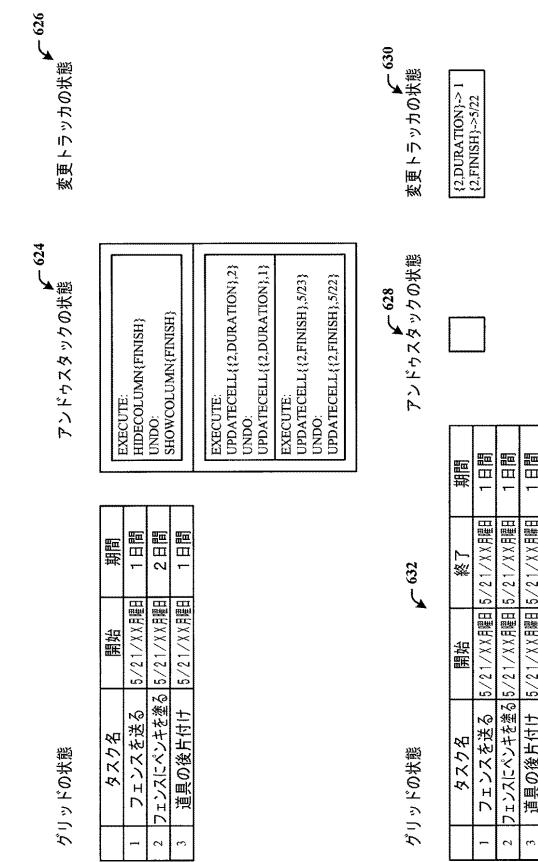
【図7】



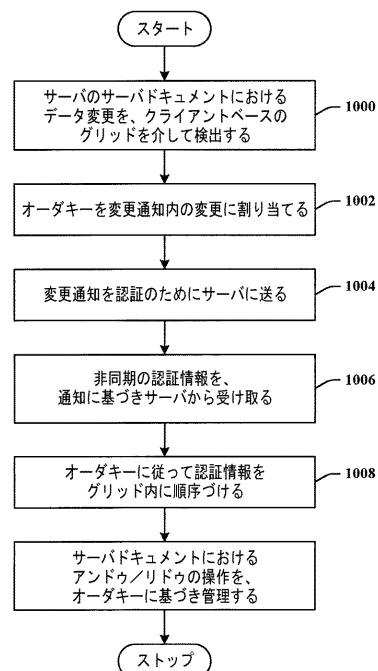
【図8】



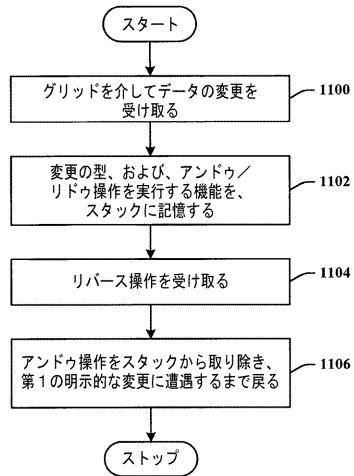
【図9】



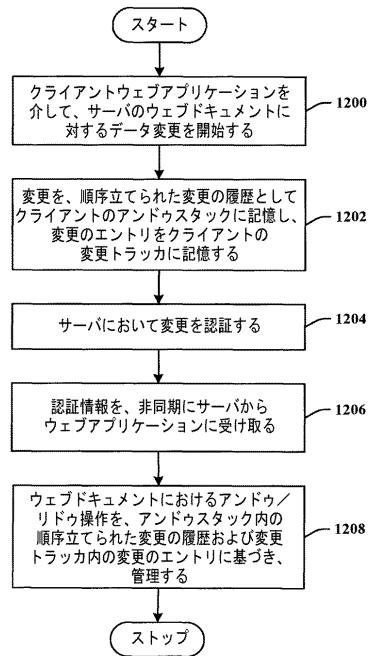
【図10】



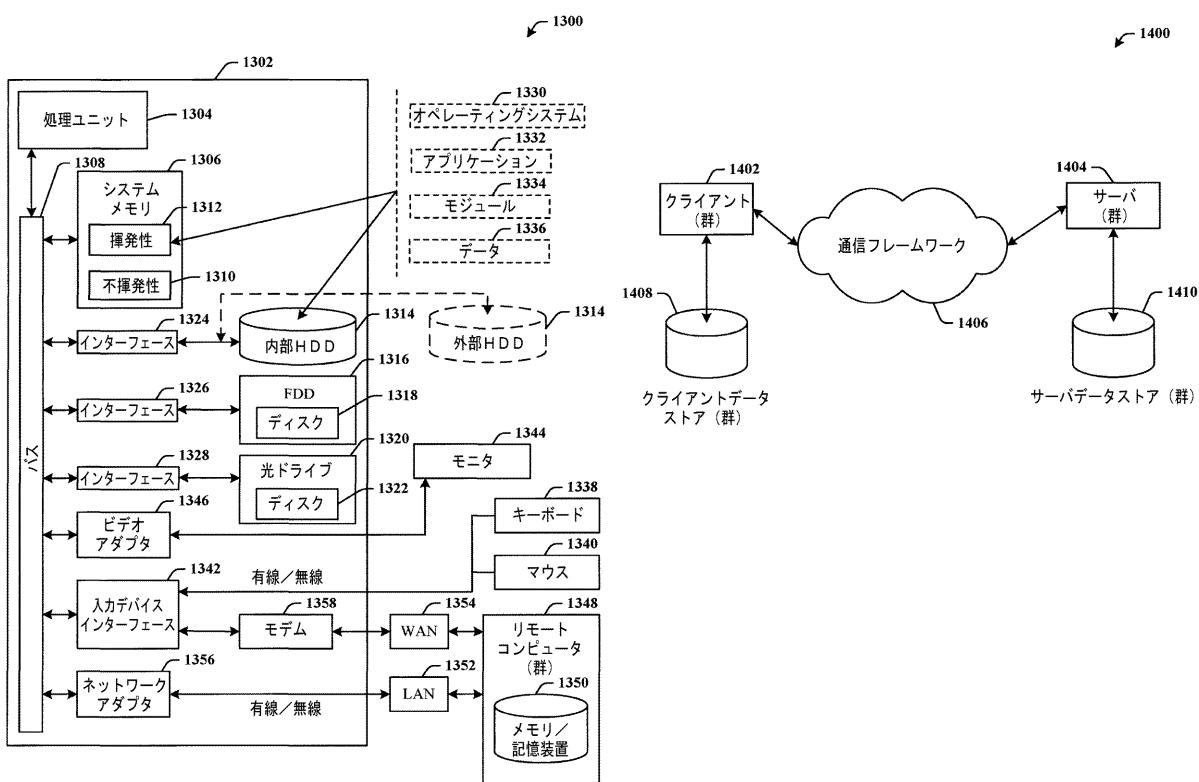
【図11】



【図12】



【図13】



フロントページの続き

(72)発明者 ベン ウオーリン

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

(72)発明者 エリック エム.ゼンツ

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

審査官 加内 慎也

(56)参考文献 特開2005-018774(JP,A)

特開2003-242010(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/00