

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5057519号
(P5057519)

(45) 発行日 平成24年10月24日(2012.10.24)

(24) 登録日 平成24年8月10日(2012.8.10)

(51) Int.Cl.		F I			
G06F 11/00	(2006.01)	G06F	9/06	630B	
G06F 9/445	(2006.01)	G06F	9/06	640A	
G06F 12/00	(2006.01)	G06F	12/00	531R	

請求項の数 8 (全 33 頁)

(21) 出願番号	特願2007-514326 (P2007-514326)	(73) 特許権者	501059992
(86) (22) 出願日	平成17年6月1日(2005.6.1)		レッド・ベンド・リミテッド
(65) 公表番号	特表2008-501180 (P2008-501180A)		Red Bend Ltd.
(43) 公表日	平成20年1月17日(2008.1.17)		イスラエル, ホッド ハシャロン 452
(86) 国際出願番号	PCT/IL2005/000568		40, ハナガー 8
(87) 国際公開番号	W02005/119432	(74) 代理人	100107364
(87) 国際公開日	平成17年12月15日(2005.12.15)		弁理士 斉藤 達也
審査請求日	平成20年5月30日(2008.5.30)	(72) 発明者	ペレグ、シャロン
(31) 優先権主張番号	60/575,425		イスラエル、47205 ラマツト ハシ
(32) 優先日	平成16年6月1日(2004.6.1)		ヤロン、ピアリク ストリート 58
(33) 優先権主張国	米国 (US)	(72) 発明者	メラー、エブヤタル
			イスラエル、49732 ベタツハ ティ
			クバ、メナケム ビギン ストリート 5
			8/32

最終頁に続く

(54) 【発明の名称】 記憶装置に記憶されたコンテンツをインプレース更新するための方法およびシステム

(57) 【特許請求の範囲】

【請求項1】

記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、

前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、

更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行う方法であって、

更新装置が、

前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、

前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させ、

前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンド

の特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させること、

を含む方法。

【請求項 2】

前記更新装置が、

前記更新コマンドが、前記記憶装置に記憶された前記元のバージョンのコンテンツ、前記更新されたバージョンのコンテンツ、又は前記中間のバージョンのコンテンツの一部を削除するための削除コマンドである場合、当該コンテンツの一部を削除する前に、当該コンテンツの一部を保持する、

請求項 1 に記載の方法。

【請求項 3】

前記更新装置が、

削除される前に保持された前記コンテンツの一部を用いて、前記削除コマンドを反転させる、

請求項 2 に記載の方法。

【請求項 4】

記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、

前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、

更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行う方法であって、

更新装置が、

前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、

前記順方向更新を行うことが決定された後に、前記元のバージョンのコンテンツに基づいて当該順方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該順方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させ、

前記逆方向更新を行うことが決定された後に、前記更新されたバージョンのコンテンツに基づいて当該逆方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該逆方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記中間のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させること、

を含む方法。

【請求項 5】

記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、

前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョン

10

20

30

40

50

ンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、

更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うシステムであって、

前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定する方向決定ユニットと、

前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる順方向更新処理装置と、

10

前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成するロールバック更新シーケンス生成器と、

前記生成されたロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させるロールバック更新処理装置と、

20

を備えるシステム。

【請求項6】

記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、

前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、

更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うシステムであって、

前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定する方向決定ユニットと、

30

前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる順方向更新処理装置と、

前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成する

40

ロールバック更新シーケンス生成器と、
前記生成されたロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させるロールバック更新処理装置とを備え、

前記順方向更新を行うことが決定された後に、前記元のバージョンのコンテンツに基づいて当該順方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該順方向更新をキャンセルする指示を受けた場合に、前記ロールバック更新シーケンス生成器は、前記ロールバック更新シーケンスを生成し、前記ロールバック更新処理装置は、前記ロールバック更新シーケンスに基づいて、前記中間のバージョンのコンテンツ

50

から、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させ、

前記逆方向更新を行うことが決定された後に、前記更新されたバージョンのコンテンツに基づいて当該逆方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該逆方向更新をキャンセルする指示を受けた場合に、前記順方向更新処理装置は、前記更新パッケージにおける前記複数の更新コマンドの特定の順序に基づいて、前記中間のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる、システム。

【請求項 7】

記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、

前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、

更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うプログラムであって、

前記記憶装置を備えたコンピュータに実行させるためのプログラムにおいて、

前記コンピュータが、

前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、

前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させ、

前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させる、

プログラム。

【請求項 8】

記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、

前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、

更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うプログラムであって、

前記記憶装置を備えたコンピュータに実行させるためのプログラムにおいて、

前記コンピュータが、

前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、

前記順方向更新を行うことが決定された後に、前記元のバージョンのコンテンツに基づいて当該順方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該順方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれ

10

20

30

40

50

る前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させ、

前記逆方向更新を行うことが決定された後に、前記更新されたバージョンのコンテンツに基づいて当該逆方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該逆方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記中間のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる、

プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、記憶装置に記憶されたコンテンツの記憶されたバージョンを更新パッケージを用いて更新することに関する。より詳細には、本発明はインプレース更新 (in-place updating) に関する。

【背景技術】

【0002】

記憶装置に記憶されたコンテンツを更新することが必要となる場合がある。例えば、コンテンツがソフトウェア (実行可能なファイル等) である場合、時としてソフトウェアをアップグレードする必要がある。しかし、注目すべきは、データベース等に記憶されたテキストやデータといったその他のタイプのコンテンツもまた時として更新が必要となるということである。以下、「旧バージョン」または「元のバージョン」という用語は更新前のコンテンツを指し、「新バージョン」または「更新されたバージョン」という用語は更新が行なわれた後のコンテンツを指す。「差分 (difference)」、「差分結果 (difference result)」または「デルタ」と呼ばれることもある「更新パッケージ」は、更新処理のための入力として提供されるデータを含み、更新処理は、更新パッケージに従って旧バージョンを新バージョンに更新する。

【0003】

更新パッケージを生成し、それらをバージョンの更新に用いるいくつかの方法が当該技術分野において知られている。例えば、米国特許第 6,546,552 号 (「内部レファレンスを含むデータ表の 2 つのバージョン間の差分抽出方法 (Difference extraction between two versions of data-tables containing intra-references)」、2003 年公開) には、旧プログラムと新プログラムとの間のコンパクトな差分結果を生成する方法が開示されている。各プログラムは、当該プログラムの他のエントリを参照するレファレンスを格納するレファレンスエントリを含む。米国特許第 6,546,552 号の方法によれば、旧プログラムは走査され、各レファレンスエントリ毎に、レファレンスは区別ラベルマーク (distinct label mark) により置換され、それにより、修正された旧プログラムが生成される。また、米国特許第 6,546,552 号によれば、新プログラムは走査され、各レファレンスエントリ毎に、レファレンスは区別ラベルマークにより置換され、それにより、修正された新プログラムが生成される。よって、修正された旧プログラムおよび修正された新プログラムを直接的または間接的に利用することにより、差分結果が生成される。

【0004】

WO 2004/114130 (「記憶装置に記憶されたコンテンツのバージョンの更新方法およびシステム (Method and system for updating versions of content stored in

10

20

30

40

50

a storage device)」、2004年公開)には、コンテンツの旧バージョンとコンテンツの新バージョンとの間のコンパクトな更新パッケージを生成するシステムおよび方法が開示されている。WO2004/114130のシステムは、旧バージョンおよび新バージョンに対応付けられた変換要素を生成する変換要素生成器を備える。また、修正されたバージョンを生成する修正バージョン生成器、およびコンパクトな更新パッケージを生成する更新パッケージ生成器も備える。コンパクトな更新パッケージは、変換要素、および修正されたバージョンと新バージョンとに基づく修正されたデルタを含む。

【0005】

WO2005/003963(「記憶装置に記憶されたコンテンツのバージョンの更新方法およびシステム(Method and system for updating versions of content stored in a storage device)」、2005年公開)には、記憶装置に記憶されたコンテンツのバージョンを更新するためのシステムおよび方法が開示されている。WO2005/003963のシステムは、変換要素および小さなデルタを取得するための更新モジュールを備える。また、変換要素を旧バージョンのアイテムに適用することにより、変換された旧アイテムを生成する変換済み旧アイテム生成器、修正されたデータエントリと変換された旧アイテムとに基づいてデータエントリを生成するデータエントリ生成器、ならびにコマンドおよびデータエントリを旧バージョンに適用することにより、コンテンツの新バージョンを生成する新バージョン生成器も備える。

【0006】

米国特許第6,832,373号(「情報の更新・配信システムおよび方法(System and method, for updating and distributing information)」、2004年公開)には、ソフトウェア、装置、およびデータによって構成される、デジタル情報シーケンスを更新するための装置、システムおよび方法が開示されている。また、これらのデジタル情報シーケンスは、ファイル、記憶域、および/または内蔵記憶域をはじめとする様々な形態で記憶および使用され得るが、これらに限定されるものではない。さらに、米国特許第6,832,373号に記載の装置、システムおよび方法は、必要に応じて更新情報を生成する能力を当該技術分野の開発者に与え、加えて、エラーが発生しにくく、米国特許第6,832,373号が出願された時点の既存の技術を用いるよりも迅速に実行し得る簡略化された更新パスをユーザが経ることを可能にする。

【0007】

コンテンツはディスクやメモリ等の記憶装置に記憶することができ、記憶装置の中にはブロックに組織化されているものがあることは、当業者に知られている。元のバージョンの一部であるブロックを「旧ブロック」または「元のブロック」と呼び、更新されたバージョンの一部であるブロックを「新ブロック」または「更新されたブロック」と呼ぶ。また、元のバージョンを更新し、それによって更新されたバージョンを形成する際、更新されたバージョンは、元のバージョンのブロックの前に記憶されたコンテンツを使用する場合がある。すなわち、更新されたブロックのコンテンツは、元のブロックのコンテンツに類似している場合がある。

【0008】

さらに、入手可能な記憶装置は、限られたスペース有することがある。よって、記憶装置内のバージョンを更新している間、旧バージョンに代えて新バージョンを記憶し、それによってスペースを節減することが好ましい場合がある。旧バージョンによって前に占められていたスペースの少なくとも一部を新バージョンが占める、このような更新処理は、当該技術分野において「インプレース更新(in-place update)」または「インプレースでの更新(updating in-place)」と呼ばれる。

【0009】

なお、更新処理は、ファイル内のブロックの連続的な順序とは必ずしも対応しない順序で旧バージョンのブロックを処理することがある。旧バージョンのブロックが更新されるこの順序を、「更新シーケンス」または「更新順序」と呼ぶ。

【0010】

10

20

30

40

50

インプレース更新の結果の一つとして、一旦記憶ブロックが更新されると、元のバージョンの一部である、記憶ブロックの少なくとも一部のコンテンツ（すなわち、元のコンテンツの少なくとも一部）は場合によっては失われるということがある。したがって、一旦更新処理が開始し、一部の旧ブロックが修正されて更新されたコンテンツ（更新されたバージョンの一部である更新されたコンテンツ）が保持された後、記憶装置は、部分的にオリジナルであって、部分的に更新されたバージョンであるコンテンツを記憶していることがある。

【 0 0 1 1 】

また、デルタ更新においてそうであるように、更新処理で旧コンテンツが必要となる場合があることが当該技術分野で知られている。したがって、新バージョンの生成が完了する前に、更新処理が実行中に中断されると、記憶されたコンテンツ（または記憶されたバージョン）が部分的に旧バージョンであり、かつ部分的に新バージョンである場合、インプレース更新処理を最初から繰り返すことはできなくなる。

10

【 0 0 1 2 】

上記に鑑み、インプレース更新は、中断に弱い場合があることが理解される。インプレース更新の処理を、それが中断された点（再開位置）から再開しようとする、と中断位置の計算を可能にする情報が必要となる。中断位置を見つけた後、更新処理には、中断の時点で中断された処理に対して入手可能であったその他の情報を復元することも必要となる。

【 0 0 1 3 】

20

現在、当該技術分野においては、中断前に計算された計算結果を記憶するために、以下において「状態レコード」と呼ぶレコードが一般に用いられている。これらの結果は、現在の更新されたブロックのIDと共に、インプレース更新処理の状態を形成する。例えば、更新パッケージが圧縮され、各旧ブロックの更新に必要な部分だけを読み出す更新処理中にオンザフライで（on-the-fly）解凍されている場合、更新パッケージから解凍されたデータ量が更新されたブロックの旧コンテンツに依存していると、この量は、一旦ブロックが実際に更新されると再計算することはできない。

【 0 0 1 4 】

既に上述したように、処理を最初から再開することはできない。したがって、状態レコード内のデータの一部は、最後に更新されたレコードがわかっている場合であっても再計算することができず、他の手段で再現する必要がある、ということが考えられる。先行技術では、上記のような状態レコードを更新処理にアクセス可能な記憶装置に維持し、更新処理中にそれを周期的に更新してその進行を反映させることが一般的に行なわれており、以下において、これを「状態記録」と呼ぶ。状態記録を用いることにより、処理状態を、最後に成功した動作におけるもののように再構築することによってインプレース更新処理の再開を可能にし、完了に向けて処理を継続することが可能となる。

30

【 0 0 1 5 】

場合によっては、状態記録中に記憶された情報は、更新中のコンテンツの一部をなすことはできず、別の記憶領域に分離されなければならない。状態記録は、記憶装置に対する読み書き動作を伴うため、時間を要する。記憶装置への読み書き動作がブロック全体に対してのみ許されるような場合には、処理全体に要する時間は2倍になる（各更新されたブロックには、更新すべき別のブロック、すなわち状態レコードを保持しているブロックが存在する）。

40

【 0 0 1 6 】

中断されたインプレース更新処理を適切に再開するためには、記憶書込動作の中断の結果である破損ブロックに対処することも必要であることも当業者に知られている。ブロックの旧コンテンツが修正されているため、当該ブロック内の情報を復元するのは不可能であるかもしれず、ファイルの他のブロックを検査することだけで旧コンテンツを復元する方法はないかもしれない。この問題は、あらゆるブロックの新コンテンツが最初に記憶されるバックアップバッファを用い、更新すべき目標ブロックにそのコンテンツをコピーす

50

ることにより（以下、「アップデート・ファースト（update-first）」という）先行技術によって解決されるか、あるいは、旧ブロックがバックアップパツファにコピーされた後に、新コンテンツが生成され、その目標ブロックに記憶される（以下、「バックアップ・ファースト（backup-first）」という）。この方法は、当該技術分野において「2相コミット（2-phase commit）方式」としても知られている。

【0017】

携帯電話等の埋め込み式機器のソフトウェアの更新は、インプレース更新手順の一例である。携帯電話の記憶容量は限られていることがあることが理解される。例えば、携帯電話は、そのソフトウェア（ファームウェアとも呼ばれる）の1つのバージョンを保持する分しか記憶容量を有していないことがある。状態記録をファームウェア自体のブロック内で維持することは明らかに非実用的である。なぜならば、これはソフトウェアの生成処理と大きく干渉することになり、したがって、別の記憶領域に記憶されなければならないからである。携帯電話のファームウェアの更新は、使用される記憶装置（フラッシュメモリ）の速度のために、比較的ゆっくりとした処理である。更新処理中、電話は動作不能であり、処理にかかる時間を最小限にする必要が生じる。

10

【0018】

ソフトウェア更新中の埋め込み式機器の動作における中断の短縮を可能にする、より高速で信頼できる更新手順が当該技術分野において必要とされている。

【0019】

記憶装置に記憶されたコンテンツのバージョンを更新するための新たな方法およびシステムを提供することが当該技術において必要とされている。

20

【発明の開示】

【発明が解決しようとする課題】

【0020】

したがって、本発明は、記憶装置に記憶されたコンテンツのバージョンを更新するための新たな方法およびシステムを提供することを目的とする。

【課題を解決するための手段】

【0021】

請求項1に記載の本発明は、記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記憶装置に記憶させる順方向更新、及び/又は、前記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記憶装置に記憶させる逆方向更新を、更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序に含まれる複数の更新コマンドと、に基づいて行う方法であって、更新装置が、前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させ、前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させることを含む。

30

40

【0022】

請求項2に記載の本発明は、請求項1に記載の本発明において、前記更新装置が、前記

50

更新コマンドが、前記記憶装置に記憶された前記元のバージョンのコンテンツ、前記更新されたバージョンのコンテンツ、又は前記中間のバージョンのコンテンツの一部を削除するための削除コマンドである場合、当該コンテンツの一部を削除する前に、当該コンテンツの一部を保持する。

【0023】

請求項3に記載の本発明は、請求項2に記載の本発明において、前記更新装置が、削除される前に保持された前記コンテンツの一部を用いて、前記削除コマンドを反転させる。

【0024】

請求項4に記載の本発明は、記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行う方法であって、更新装置が、前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、前記順方向更新を行うことが決定された後に、前記元のバージョンのコンテンツに基づいて当該順方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該順方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させ、前記逆方向更新を行うことが決定された後に、前記更新されたバージョンのコンテンツに基づいて当該逆方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該逆方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記中間のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる。

【0025】

請求項5に記載の本発明は、記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うシステムであって、前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定する方向決定ユニットと、前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる順方向更新処理装置と、前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成するロールバック更新シーケンス生成器と、前記生成されたロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前

10

20

30

40

50

記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させるロールバック更新処理装置とを備える。

【0026】

請求項6に記載の本発明は、記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うシステムであって、前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定する方向決定ユニットと、前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる順方向更新処理装置と、前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成するロールバック更新シーケンス生成器と、前記生成されたロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させるロールバック更新処理装置とを備え、前記順方向更新を行うことが決定された後に、前記元のバージョンのコンテンツに基づいて当該順方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該順方向更新をキャンセルする指示を受けた場合に、前記ロールバック更新シーケンス生成器は、前記ロールバック更新シーケンスを生成し、前記ロールバック更新処理装置は、前記ロールバック更新シーケンスに基づいて、前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させ、前記逆方向更新を行うことが決定された後に、前記更新されたバージョンのコンテンツに基づいて当該逆方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該逆方向更新をキャンセルする指示を受けた場合に、前記順方向更新処理装置は、前記更新パッケージにおける前記複数の更新コマンドの特定の順序に基づいて、前記中間のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる。

【0027】

請求項7に記載の本発明は、記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うプログラムであって、前記記憶装置を備えたコンピュータに実行させるためのプログラムにおいて、前記コンピュータが、前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、前記順方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記記憶装置に記憶された前記元のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記

10

20

30

40

50

憶装置に記憶させ、前記逆方向更新を行うことが決定された場合、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させる。

【0028】

請求項8に記載の本発明は、記憶装置に記憶された元のバージョンのコンテンツから、更新されたバージョンのコンテンツを生成して前記記憶装置に記憶させる順方向更新、及び/又は、前記記憶装置に記憶された前記更新されたバージョンのコンテンツ又は中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成して前記記憶装置に記憶させる逆方向更新を、更新パッケージに含まれる方向指示と、前記更新パッケージに特定の順序で含まれる複数の更新コマンドと、に基づいて行うプログラムであって、前記記憶装置を備えたコンピュータに実行させるためのプログラムにおいて、前記コンピュータが、前記更新パッケージに含まれる前記方向指示に基づいて、前記順方向更新と前記逆方向更新のいずれの更新を行うのかを決定し、前記順方向更新を行うことが決定された後に、前記元のバージョンのコンテンツに基づいて当該順方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該順方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドに基づいて、複数の逆更新コマンドを含むロールバック更新シーケンスであって、前記複数の逆更新コマンドの各々は当該逆更新コマンドに対応する更新コマンドの逆のプロセスを行うものであり、前記複数の逆更新コマンドは前記複数の更新コマンドの特定の順序の逆の順序で行われるものである、ロールバック更新シーケンスを生成し、当該生成したロールバック更新シーケンスに基づいて、前記中間のバージョンのコンテンツから、前記元のバージョンのコンテンツを生成し、当該生成された元のバージョンのコンテンツを前記記憶装置に記憶させ、前記逆方向更新を行うことが決定された後に、前記更新されたバージョンのコンテンツに基づいて当該逆方向更新を途中まで行うことで中間のバージョンのコンテンツが生成された時点で当該逆方向更新をキャンセルする指示を受けた場合に、前記更新パッケージに含まれる前記複数の更新コマンドの特定の順序に基づいて、前記中間のバージョンのコンテンツから、前記更新されたバージョンのコンテンツを生成し、当該生成された更新されたバージョンのコンテンツを前記記憶装置に記憶させる。

【発明を実施するための最良の形態】

【0029】

本発明を理解し、それが実際にどのように実施され得るかを知るために、添付の図面を参照しながら、好ましい実施形態を非限定的な例としてのみ以下に説明する。

【0030】

以下の説明において、2つ以上の図面に共通する構成要素には、同一の参照符号が付されている。

【0031】

さらに、特に示さない限り、以下において、「更新」という用語はインプレース更新を指すものとして用いられる。

【0032】

図1は、本発明の一実施形態に係る、セルラーネットワークにおいてバージョンを更新するためのシステム101の概略図である。記憶装置103に接続されている、または含まれる携帯電話102は、各自の動作を可能にするプログラムを実行する。プログラムは、通常、ファイルに記憶されている。以下において、携帯電話内で現在実行中のプログラムのバージョンを「旧バージョン」または「元のバージョン」という。

10

20

30

40

50

【 0 0 3 3 】

なお、記憶装置 1 0 3 等の記憶装置は、「記憶装置」または「メモリユニット」とも呼ばれる場合がある。

【 0 0 3 4 】

電話 1 0 2 がその新たなバージョンを実行するためにプログラムの更新が必要となる場合がある。このような更新されたバージョンは、電話内で動作している更新処理によって生成される。更新処理は、携帯電話が受信する更新パッケージ（「デルタファイル」を構成する）に従って動作する。

【 0 0 3 5 】

本発明によれば、更新パッケージは、例えばパーソナルコンピュータ（PC）またはその他のあらゆるタイプのコンピュータ内で動作している更新パッケージ生成器 1 0 4 において生成される。更新パッケージは、更新サーバ 1 0 5 に記憶され、送信機 1 0 6 を介して携帯電話 1 0 2 に送信される。

10

【 0 0 3 6 】

なお、図 1 に示すシステム 1 0 1 は、非限定的な例であり、本発明はプログラムの更新に限定されるものではない。データベースに記憶されたデータや記憶装置に記憶されたファイルといった、記憶装置に記憶されたその他の多くのタイプのコンテンツが更新を必要とする。したがって、以下において、「プログラム」の代わりに、「コンテンツ」という用語を用いる。

【 0 0 3 7 】

20

同様に、本発明は、セルラーネットワークおよび/または携帯電話 1 0 2 に限定されるものではない。携帯電話は、埋め込み式機器と称されるグループに属すると理解される。コンテンツを記憶するために記憶装置に接続された携帯情報端末（PDA）、セットトップボックスおよびそれ以外の民生用電子機器といったその他の埋め込み式機器があり、それらに記憶されたコンテンツの更新が時として必要となる。しかし、PCやその他のコンピュータ等の非埋め込み式機器に接続された記憶装置に記憶されたコンテンツも更新することが可能である。したがって、記憶装置 1 0 3 は、例えば、ハードディスクドライブ、フラッシュメモリ装置またはその他のあらゆる記憶装置でありうる。

【 0 0 3 8 】

例えば、PCやその他のコンピュータは、その動作、またはその内部で実行されるプログラムの動作に必要なデータ（当業者に知られる「infoファイル」または「dotファイル」等）を含むファイルを記憶し得る。時として、例えば通信回線を介して（インターネットやその他の通信手段を介して）このデータを更新することが必要となる。

30

【 0 0 3 9 】

これをふまえ、「電話」、「PDA」、「民生用電子機器」、「コンピュータ」、「PC」等の用語を用いる代わりに、以下においては「更新された装置（updated devices）」という用語を用いる。なお、「更新装置（update device）」という用語は、記憶装置に接続され、かつそこに記憶されたコンテンツの更新を可能にするあらゆる装置を意味し得る。

【 0 0 4 0 】

40

記憶装置に記憶されたコンテンツを更新するために、更新パッケージが生成され、更新サーバ 1 0 5 に記憶され、記憶装置またはこれらに接続されたその他の装置（携帯電話 1 0 2 等）に送られることは既に説明した。代わりに、更新パッケージを、最初に更新サーバ 1 0 5 に記憶することなく送ることが可能である。例えば、更新パッケージを、それが生成される更新パッケージ生成器から直接送ることが可能である。このような場合、更新生成器が動作する機器または更新生成器それ自体が更新サーバ 1 0 5 であると考えられる。

【 0 0 4 1 】

さらに、図 1 に示す例において、更新パッケージは、送信機 1 0 6 を介して送られる。これにもまた限らず、更新パッケージを送るために適用可能なその他のあらゆる方法を用

50

いることができる。例えば、更新パッケージをフロッピーディスクやディスクオンキー等の携帯用記憶装置上に記憶し、それによって、更新された装置（電話102等）が更新パッケージをそこから読み出すことにより、これにアクセスすることを可能にすることができる。

【0042】

携帯電話102は、更新パッケージを受信すると、更新パッケージに従って更新処理を動作させることができ、更新処理は、「更新されたバージョン」または「新バージョン」と呼ぶ別のバージョンを生成するために元のバージョンを更新する。なお、携帯電話102は、更新パッケージを受信した直後に更新処理を動作させることができる。あるいは、記憶装置103内というように、不揮発性メモリ内に更新パッケージを記憶し、また後で（例えば、電話が次に再起動した際に）更新処理を動作させることができる。

10

【0043】

なお、記憶装置は、元のバージョンおよび/または更新されたバージョンのコンテンツを記憶することができる。当業者であれば、コンテンツは通常ファイルに記憶され、ファイルまたはそこに記憶されたコンテンツに対して更新処理によって更新が行なわれることを理解するであろう。

【0044】

多くの場合、ファイルは論理的に連続したコンテンツで構成される。例えば、テキストが「123456789」であるテキストを含むファイルにおいて、文字「2」は文字「1」に論理的に連続し、文字「3」は文字「2」に論理的に連続している。しかし、当業者であれば、このファイルを記憶装置に記憶する際、ファイルが断片化し得ること、すなわち、ファイルの異なる部分が記憶装置の異なる領域に保存され得ることを理解するであろう。換言すれば、論理的に連続したコンテンツは、記憶装置において物理的に連続して記憶されとは限らない。

20

【0045】

さらに、当業者は、多くの記憶装置はその内部に個別の領域を含むことを理解するであろう。例えば、ハードドライブは、セクタを有し、フラッシュメモリモジュールもまた同様である。記憶装置内の個別の領域を、「記憶ブロック」、または単に「ブロック」と呼ぶ。しかし、ハードドライブおよびフラッシュメモリモジュールは、記憶装置の2つの例にすぎない。ランダムアクセスメモリ（RAM）等の、それ自体が公知であるその他の記憶装置が存在している。

30

【0046】

論理的に連続したコンテンツは記憶装置に物理的に連続して記憶されとは限らないという点をふまえると、連続したコンテンツはいくつかの記憶ブロックに散在し得ることが理解される。さらに、1つの記憶ブロックは、いくつかの論理的に連続したコンテンツ（いくつかのファイル等）に属するコンテンツを含みうる。上記の例に戻り、テキスト「123456789」が論理的に連続したコンテンツを構成している場合、コンテンツ「1234」は1つの物理ブロックに記憶され得る一方、この例のコンテンツ「56789」は「1234」が記憶されたブロックに物理的に先行する異なる物理ブロックに記憶され得ることが理解される（しかし、コンテンツ「1234」がコンテンツ「56789」に物理的に先行することは明らかである）。

40

【0047】

さらに、論理的に連続したテキスト「123456789」は、元のバージョンであると考えられる。本例によれば、このファイルを、更新されたバージョンを含むように更新することが可能であり、更新されたバージョンに記憶されたテキストは「123756489」である。この更新されたバージョンにおいて、文字「4」と「7」は、元のバージョンと比較すると、その位置が入れ替えられており、言い換えると、更新されたバージョンにおいては、文字「7」は元のバージョンに見られる文字「4」に取って代わり、文字「4」は文字「7」に取って代わる。したがって、この更新されたバージョンを生成するために、元のバージョンをいくつかのセグメントに分割することが可能である（各セグメ

50

ントは「論理ブロック」を構成する)ことが理解される。例えば、第1のセグメントはコンテンツ「123」を含み、第2のセグメントのコンテンツは「4」であり、第3のセグメントは「56」を含み、第4のセグメントは「7」を含み、第5のセグメントは「89」を含む。更新されたバージョンを生成する際、第1、第3および第5のセグメントはそのままにされる一方、第2および第4のセグメントは入れ替えられる。したがって、セグメントは論理的に連続したコンテンツを含む。

【0048】

セグメントは、更新中に一様な動作を行うコンテンツの論理的に連続したセグメントであるとさらに定義することが可能である。本例において、セグメント「123」は論理的に連続しており、更新されたバージョンにおいて、一様に、修正されない状態のままである。同様に、例えば、セグメント「7」もまた、論理的に連続しており(1文字のシーケンスは有効なシーケンスである)、前に「4」によって占められていた位置を占めるように一様に移動させられる。

10

【0049】

各セグメントが論理的に連続したコンテンツを含み、更新中に一様な動作を行う、「123456789」テキストの上記の例および更新されたバージョン「123756489」への更新は簡略化された例である。他の例において、コンテンツを論理的に連続したセグメントに分割し、更新中に一様な動作を行ない、1つ以上の物理ブロックにおける物理的に連続した領域を占めることが可能である。したがって、第1のブロックに「12」が記憶され、第2のブロックに「3」が記憶され、「12」が「3」に連続的に先行しないように「123」が断片化されている場合、本例によれば、「123」は、論理的に連続してはいるが、2つのセグメント(具体的には、「12」および「3」が存在する)に分割される。

20

【0050】

しかし、後者の例において、ブロックが連続(第1のブロックは、第2のブロックに連続的に先行する)していて、かつ文字「1」が第1のブロックが終わる領域に記憶され、文字「2」が第2のブロックが始まる領域に記憶されている限り、セグメント「12」は2つの異なるブロック上に存在することが可能である。

【0051】

後者の例は、非限定的である。例えば、別の実施形態では、(上記例の「1つ以上」とは異なり)1つの物理ブロックにおける物理的に連続した領域を占めるようにセグメントを制限することができる。したがって、「12」が2つの連続したブロックに連続的に記憶されている場合、これを2つのセグメントにさらに分割しなければならない(1つは「1」であり、もう1つは「2」である)。

30

【0052】

更新されたバージョンにおけるセグメント(「更新されたセグメント」)のコンテンツが元のバージョンのセグメント(「元のセグメント」)のコンテンツに対応している場合、これらのセグメントは、「対応するセグメント」または「一致するセグメント」であると考えられる。対応とは、セグメント間のあらゆる論理的なつながりを指し、更新されたセグメントは元のセグメントのコピーであることがあり、それは元のセグメントの修正されたコピーであり得るか(例えば、大きいセグメントにおいては、セグメント全体をコピーし、その後、コピーされたコンテンツの一部を修正することが好ましい場合がある)、あるいは、元のセグメントに記憶されたコンテンツを用いて計算を行うことなどにより受信されたコンテンツを含みうる。

40

【0053】

なお、ある1つの元のセグメントに対応して2つ以上の更新されたセグメントが存在することがある(例えば、元のセグメントのコンテンツが2度コピーされ、それにより、それに対応する2つの更新されたセグメントが生成される)。この逆も言える。1つの更新されたセグメントに対応する2つ以上の元のセグメントが存在する場合がある(例えば、1つの更新されたセグメントのコンテンツを計算するために、2つの別個の元のセグメン

50

トに記憶されたコンテンツを使用することが可能である)。

【0054】

元のバージョンを更新されたバージョンに更新するために、更新パッケージに従って更新処理が動作することは既に述べた。一実施形態によれば、更新パッケージは更新コマンド(簡潔にいうと「コマンド」)を含み、各コマンドは、元のバージョンおよび/または更新されたバージョンにおける1つのセグメントに対応している。更新コマンドは、更新動作を示すコードを含むことができ、コードは、例えば、数字コード、ビットマスク、ストリングコードまたはこの場合に適用可能なその他のあらゆるコードでありうることが理解される。

【0055】

元のバージョンを更新されたバージョンに更新するための異なる更新動作ひいては異なる更新コマンドが存在する。例えば、以下の様なものである。

【0056】

「削除」コマンドは、更新されたバージョンを生成する際にコンテンツが削除された元のバージョンにおける1つのセグメントに対応する(よって、このコマンドは更新されたバージョンにおけるいかなるセグメントにも対応しない)。元のバージョンにおける削除されたセグメントは、更新されたバージョンにおいて対応するセグメントを有しない。削除コマンドにより、記憶装置内のセグメントは開放され、それにより、セグメントに他のコンテンツを書き込むことが可能になる。すなわち、更新処理は、コンテンツの物理的な削除を必要としない。更新処理は、この領域を空きであるとしてマークするだけである。

【0057】

「挿入」コマンドは、更新バージョンにおける1つのセグメントに対応し、このセグメント内には新コンテンツが追加される。更新されたバージョンにおける挿入されたセグメントは、元のバージョンにおいて対応するセグメントを有しない。なお、一部の実施形態においては、挿入されたコンテンツが、空きでない領域(すなわち、以前に、例えば削除コマンドによって解放されなかったコンテンツを含む物理領域)に書き込まれる場合、元のコンテンツは、挿入されたコンテンツによってオーバーライドされ、したがって、削除されたものとされる。これらの実施形態において、このような挿入コマンドは、挿入コマンドを後に伴う削除コマンドと等しいと考えられる。

【0058】

「コピー」コマンドは、元のバージョンにおける1つのセグメント(「元のセグメント」)と、更新されたバージョンにおける1つのセグメント(「更新されたセグメント」)とに対応し、元のセグメントのコンテンツは更新されたセグメントにコピーされる。元のセグメントおよび更新されたセグメントは、対応するセグメントであると考えられる。(コピーコマンドによる)コピー動作を行った後、元のセグメントによって前に占められていた領域は空きであるとされ、よって、他のコンテンツを上書きすることが許される。なお、更新処理は、コンテンツの物理的な削除(解放)を必要としない。更新処理は、この領域を空きであるとしてマークするだけである。また、挿入コマンドと同様に、一部の実施形態では、空きでない領域に書き込まれた挿入されたコンテンツ(すなわち、以前に、例えば削除コマンドによって解放されなかったコンテンツを有する物理領域)が、この解放されていないコンテンツをオーバーライドし、したがって、削除されたものとされる。これらの実施形態において、このようなコピーコマンドは、コピーコマンドを後に伴う削除コマンドと等しいと考えられる。

【0059】

「置換」コマンドは、元のセグメントのコンテンツを更新バージョンにおける新コンテンツで置換する。よって、これは、1つの元のセグメントと、1つの更新されたセグメントとに対応する。元のセグメントのコンテンツは削除され、補足データ(以下の図4の、置換補足データ部分419に記憶されたデータ等)が更新されたセグメントに書き込まれる。しかし、当業者であれば、データを別のものに反転可能に変換するビット毎のXOR(排他的論理和)といった変換関数が存在することを理解するであろう。例えば、元のセ

10

20

30

40

50

グメントのコンテンツと、更新されたセグメントのコンテンツを生じさせるデータ（「変換データ」とのビット毎の排他的論理和を取ることが可能である。当業者であれば、更新されたコンテンツと変換データとのビット毎の排他的論理和を取り、元のコンテンツを再び得ることが後に可能となることを理解するであろう（ビット毎の排他的論理和は、数学的に逆戻り可能な（revertible）関数である。すなわち、「 $A \text{ XOR } B = C$ 」である場合、「 $C \text{ XOR } B = A$ 」）である。コンテンツの置換に変換関数を用いる実施形態においては、補足データは変換データを含みうる。

【0060】

なお、上記のコマンド例は、非限定的なものであり、追加のタイプの更新コマンド（「バックアップ」等）または更新パッケージにおいてより少数のタイプの更新コマンドが存在し得る。例えば、「置換」コマンドの代わりに、上記に説明したような「削除」と「挿入」の組み合わせを用いることが可能であることが理解される。別の例によると、例えば、更新処理に対して、1つのセグメントを削除することだけを命令する単純な更新パッケージ（よって、このパッケージは挿入コマンド、コピーコマンドおよび置換コマンドを含まない）が存在し得る。

【0061】

1度につき、1つのブロックに記憶されたコンテンツへの（読み出しおよび/または書き込みのための）アクセスを許可する記憶装置103が設けられている。このような記憶装置用に適合された更新パッケージにおいては、単一のブロックに対応するコマンドは、グループに組織化されているほうが良く、各グループにおけるコマンドは、1つのブロックに対応し、間に交じった他のブロックに対応するコマンドを含まない。この更新処理を行うことにより、元のブロックから要求された全て元のコンテンツに1回の読み出し動作でアクセスすること、あるいは、更新されたブロックから要求された全ての更新されたコンテンツに1回の書き込み動作でアクセスすることが可能である。このような記憶装置に適合された更新パッケージにおける異なるブロックに対応する、交じり合うコマンドは、更新処理がコマンドに従って動作を行う度にブロック全体にアクセスするが、実際にはそのセグメントへのアクセスを必要とする。

【0062】

図2に移る前に、記憶装置に現在記憶されているバージョンを指すのに「記憶されたバージョン」という用語が用いられていることに注目されたい。記憶されたバージョンは、例えば元のバージョンである可能性があり、例えば以下の図2を参照して示す中間バージョン、あるいは、更新されたバージョンである可能性がある。

【0063】

図2は、ブロック全体へのアクセスを許可する記憶装置における、元のバージョン201の更新されたバージョン202への更新の簡略化された段階例を示している。元のバージョン201は4つのブロックを含み、これらは具体的には203、204、205および206である。本例によると、更新されたバージョン202もまた、4つのブロック、すなわち207、208、209および210を含み、ブロック207はブロック203に取って代わり、ブロック208はブロック204に取って代わり、ブロック209はブロック205に取って代わり、ブロック210はブロック206に取って代わる。本例によれば、更新処理は、最初にブロック203をブロック207で置換する。したがって、記憶されたバージョン211は、元のバージョン201と更新されたバージョン202との組み合わせを含み、第1のブロックは207（これは更新されたバージョンに属する）であり、その他のブロック（すなわち、204、205および206）は、依然として元のバージョン201に属する。次に、本例によれば、更新処理は、ブロック204をブロック208で置換し、それにより、記憶されたバージョンは212となる。そして、記憶されたバージョン213が生成され、最後に、記憶されたバージョンは更新されたバージョン202となる。部分的にオリジナルであって、部分的に更新されたバージョン（例えば、バージョン211、212、213）を、「中間バージョン」と呼ぶ。

【0064】

10

20

30

40

50

図2に示す更新処理は、正常にあるいはそうでないかに拘らず、どの段階においても終了し得ることが理解される。したがって、記憶装置に記憶されたバージョンは、バージョン201、211、212、213または202のいずれでもありうる。さらに、更新されたブロック内のコンテンツを記憶している間に中断が起こりうるため、中断点で最後に更新されたブロックは破損され得る。すなわち、当業者に公知のように未定義のコンテンツを含みうる。

【0065】

なお、図2を参照して示した例は決して限定的なものではない。元のバージョン、中間バージョンおよび/または更新されたバージョンは、いかなる数のブロックを含む可能性があり、1つのブロックは、適用可能ないかなる数のビットを含む可能性があり、更新処理は、ブロックの置換とは別に、またはこれに加えて、この場合に適応可能なあらゆる動作（ブロックの削除、ブロックの挿入、ブロックの移動等）を行うことができる。

【0066】

また、ブロックの一部へのアクセスを許可する記憶装置においても、元のバージョン、更新されたバージョンおよび中間バージョンが存在し得る。しかし、これらの場合、図2はブロックではなくセグメントを示すことになる。

【0067】

図3は、元のバージョン301の例および対応する更新されたバージョン302の例を示している。元のバージョンは、「ソースバージョン」とも呼ぶ場合があり、更新されたバージョンは、「ターゲットバージョン」とも呼ぶ場合がある。セグメント303、304、305、306、307および308は、共に元のバージョン301を構成し、セグメント309、310、311、312、313、314および315は、共に更新されたバージョン302を構成している。なお、更新されたバージョンと比べて元のバージョンのセグメントの数は少ない（更新されたバージョンが7つのセグメントを有しているのに対して、元のバージョンにおいては6つのセグメント）。したがって、更新処理は、更新されたバージョンに新コンテンツ（本例においては、セグメント311および313）を挿入したことになる。また、元のバージョンにおけるセグメント303、305、306および308のコンテンツは、更新されたバージョンにおけるセグメント309、312、315および314のコンテンツとそれぞれ同一であることに注目されたい。すなわち、これらのセグメントの元のコンテンツは、更新されたバージョンにおける対応するセグメント内にコピーされている。さらに、元のバージョンのセグメント307のコンテンツは削除され、よって、新バージョンには現れず、セグメント310の更新されたコンテンツがセグメント304の元のコンテンツに取って代わる。

【0068】

図3の例において、元のバージョン301は、記憶装置内の1つのブロック、具体的にはブロック316に記憶されている。更新処理はインプレース更新処理である。すなわち、更新されたバージョンが元のバージョンをオーバーライドするので、更新バージョン302もまたブロック316に記憶されている。しかし、これには限らず、バージョン（元のバージョンおよび/または更新されたバージョン）は、上記「123456789」の例を参照して既に説明したように、2つ以上のブロックに記憶され得る。また、301および302を、より大きい元のバージョンおよび更新されたバージョンの一部として考えることもできる。したがって、「元のバージョン301」および「更新されたバージョン302」という用語は、全体としてのバージョン、またはそこからの1つのブロックを指す。

【0069】

同図を見ると、元のバージョン301の両側を埋める2つの空きセグメントが存在することが示されている。具体的には、これらはセグメント318および319である。更新されたバージョン302を埋める2つの空きセグメント320および321も存在する。なお、空きセグメント318、319、320および321は、元のバージョンまたは更新されたバージョンの一部ではなく、これらは、元のバージョンまたは更新されたバージョン

10

20

30

40

50

ョンによって占められた際の、ブロック316における未使用領域である。これらの空きセグメントは、その他のファイルの一部であるコンテンツを記憶するために用いられる場合がある。本例によれば、空きセグメント320は空きセグメント318よりも小さく、空きセグメント321は空きセグメント319よりも小さいことにさらに注目されたい。すなわち、元のバージョン301によって占められるスペースは、更新されたバージョン302によって占められる大きさよりも小さい。

【0070】

セグメント303および309は対応するセグメントである。304と310、305と312、306と315、および、308と314も同様である。

【0071】

図4は、図3の元のバージョン301をその更新されたバージョン302に更新するように構成された別の更新パッケージ401および402を示している。更新パッケージ401は、バージョンを更新するための更新コマンドを含む。コマンド403は、更新処理に対し、セグメント303の元のコンテンツを更新されたセグメント309にコピーするように命令する。更新コマンド404は、更新処理に対し、セグメント304の元のコンテンツを「置換されたデータ」で置換するように命令し、それにより、更新されたセグメント310の更新されたコンテンツが得られる。なお、上記に説明したように、データを置換するためにXOR等の変換関数を用いることも可能である。このような実施形態において、「置換されたデータ」は変換データを記憶することになる。

【0072】

コマンド405は、セグメント307の元のコンテンツを削除する役割を担い、コマンド306は、セグメント308の元のコンテンツを更新されたセグメント314にコピーし、コマンド407は、セグメント306の元のコンテンツを更新されたセグメント315にコピーし、コマンド408は、更新処理に対し、「挿入されたデータ」を更新されたセグメント313に挿入するよう命令し、コマンド409は、更新処理に対し、セグメント305の元のコンテンツを更新されたセグメント312にコピーするよう命令し、コマンド410は、挿入されたデータをセグメント311の更新されたコンテンツとして挿入する役割を担う。なお、挿入されたデータおよび/または置換されたデータ等の更新処理に供されるデータは、更新パッケージ401の更新コマンドと交じり合っている。データが更新コマンドと交じり合っているこのような更新パッケージを、「データ非区別 (indistinct) 更新パッケージ」または「データ混交更新パッケージ」と呼ぶ。

【0073】

なお、更新コマンドが更新パッケージに現れる順序は、元のバージョンおよび/または更新されたバージョンの各セグメントの順序と等しいとは限らない。また、301および302が元のバージョンまたは更新されたバージョンの一部である場合、更新パッケージは、さらなるブロックのさらなるセグメントを更新するためのさらなるコマンドを含む。この場合、401は更新パッケージの一部である。

【0074】

なお、さらに、一部の更新コマンドに必要な「挿入されたデータ」および「置換されたデータ」等のデータを、通常、「補足データ」と呼ぶ。補足データを必要とする更新コマンドを、「データ消費更新コマンド」と呼ぶ。

【0075】

更新パッケージ402は、元のバージョン301から更新されたバージョン302を生成する別の更新パッケージ(またはその一部)である。更新パッケージ402における更新コマンドは、更新パッケージ401の更新コマンドに相当し、そこに現れる順序と同様に順序づけられ、コマンド411はコマンド403に相当し、コマンド412はコマンド404に相当し、コマンド413はコマンド405に相当し、コマンド414はコマンド406に相当し、コマンド415はコマンド407に相当し、コマンド416はコマンド408に相当し、コマンド417はコマンド409に相当し、コマンド418はコマンド409に相当する。しかし、更新パッケージ402においては、更新パッケージ401と

10

20

30

40

50

は異なり、挿入コマンドおよび置換コマンドに必要な全ての補足データは、更新パッケージの終わりに現れる。更新パッケージの部分 4 1 9 は、置換コマンドに必要な補足データを記憶するために使用され（よって、これを「置換補足データ部分」と呼ぶ）、更新パッケージの部分 4 2 0 は、挿入コマンドに必要なデータを記憶するために使用される（よって、これを「挿入補足データ部分」と呼ぶ）。補足データが補足データ部分に記憶されるこのような更新パッケージを、「データ区別 (distinct) 更新パッケージ」と呼ぶ。

【 0 0 7 6 】

なお、各補足データ部分（部分 4 1 9 および 4 2 0 等）は、2 つ以上のデータ消費更新コマンドに対応する補足データアイテムを含みうる。図 4 の例においては、挿入補足データ部分 4 2 0 は、更新コマンド 4 1 6 および 4 1 8 の補足データを含む。

10

【 0 0 7 7 】

一実施形態によれば、データ消費更新コマンドは、更新パッケージに記憶された各補足データアイテムに対するポインタ、例えば、部分 4 1 9 および 4 2 0 のうち一方に含まれる補足データアイテムの始まりに対するポインタを含みうる。別の実施形態によれば、補足データを必要とする各更新コマンドは、各データのサイズを含むことができ、それにより、各補足データアイテムの開始位置を演算で求めることが可能となる。しかし、これには限らず、その他の実施形態もまた可能である。例えば、補足データ部分は、各補足データアイテムをこのアイテムのサイズと対応付けることができる。補足データアイテムが、更新パッケージにおけるデータ消費更新コマンドと同順序で補足データ部分に記憶されている場合、当業者であれば、この実施形態により、補足データアイテムの消費更新コマンドごとの開始位置を演算で求めることができることを理解するであろう。

20

【 0 0 7 8 】

元のバージョンが更新され、更新されたバージョンが生成されることが当業者には理解される。このような（または元のバージョンから更新されたバージョンへの）更新を、「順方向更新」と呼ぶ。また、当業者であれば、バージョンの順方向更新の間、更新パッケージは、その始まりから終わりに向かって処理されることを理解するであろう。更新パッケージをその始まりから終わりに向かって処理することを、更新パッケージの「順方向処理」と呼ぶ。

【 0 0 7 9 】

本発明によれば、更新の方向を反転させて、それにより、元のバージョンに復帰することが必要となることがある。このような反転は、例えば、順方向更新処理が途中で失敗し、それにより、中間バージョンである記憶されたバージョンが生成される場合に望まれる場合がある。このような失敗が起こった場合、例えば、携帯電話の操作者は、前のバージョン（元のバージョン）に復帰することを望む場合がある。

30

【 0 0 8 0 】

別の例によれば、更新処理は、開始時に、「携帯端末はソフトウェアを更新中です。更新は 3 0 分間継続すると見込まれ、その間、電話は動作不能となります」といったメッセージをユーザに提示することができる。ユーザが、次の 3 0 分以内に電話を使用しなければならない場合、ユーザには、更新処理をキャンセルする（したがって、これを後に延期する）オプションが与えられる。しかし、更新処理は、ユーザがキャンセルを押す前に更新タスクの一部を行なうかもしれず、これにより、中間バージョンが生じる。したがって、ユーザが「キャンセル」を押すと、更新処理は反転し、元のバージョンに復帰しなくてはならない。

40

【 0 0 8 1 】

記憶されたバージョンからその対応する元のバージョンに復帰することを、「ロールバック (roll-back)」と呼び、ロールバックを行う更新処理を、「ロールバック (rolling-back) 更新処理」と呼ぶ。本発明の一実施形態によれば、ロールバック更新処理は、この元のバージョンの順方向更新に用いられる更新パッケージに従って動作し得るが、詳しく後述するように、逆方向に（すなわち、その終わりから始まりに向かって）動作する。

【 0 0 8 2 】

50

示されている例は、決して限定的なものではなく、ロールバックのその他の例および理由も存在し得る。そのようなさらなる例の一つは、携帯電話の操作者が元のバージョンを更新されたバージョンに更新することに成功するが、更新されたバージョンが満足のいかないものであることが分かり、したがって、操作者は前に動作していた元のバージョンへの復帰（ロールバック）を望む場合である。すなわち、ロールバックは、記憶されたバージョンが必ずしも中間バージョンである必要はなく、記憶されたバージョンが更新されたバージョンである場合にも起こりうる。

【 0 0 8 3 】

図 4 の更新パッケージ 4 0 2 に戻ると、4 2 1 は「方向指示」であり、「更新属性ビット」を構成している。方向指示は、更新処理が動作すべき方向および/またはこの動作の理由を示すために用いられる。例えば、異なる方向および/または動作理由には、異なるコードが存在し得る。

10

【 0 0 8 4 】

しかし、これには限らず、その他の実施形態では、必ずしも更新パッケージの一部ではないその他の記憶領域に方向指示を記憶することができる。例えば、記憶装置 1 0 3 内に「方向指示バッファ」を設けることができる。

【 0 0 8 5 】

ある実施形態によれば、更新方向は、2 つの方向のうち一方である。具体的には、この方向は、順方向およびロールバックのうち一方である。しかし、これには限らず、その他の実施形態は、この場合に適用可能なあらゆる方向を有し得る。

20

【 0 0 8 6 】

本発明の一実施形態に従って動作する更新処理は、動作を開始する際、動作方向および開始位置を認識する必要がある。なお、順方向更新が最初に開始すると、記憶されたバージョンは、通常、元のバージョンであり、開始位置は、元のバージョンの始まりであり、かつ更新パッケージの始まりであり、方向は（既に示したように）順方向である。しかし、記憶されたバージョンが中間バージョンである場合、開始位置は、中間バージョンのどこかであり、かつ更新パッケージのどこかであることになる。同様に、方向は、順方向またはロールバックのいずれでもありうる。

【 0 0 8 7 】

図 3 に戻り、更新処理中、いくつかのセグメントが修正された（したがって、「置換されたブロック」または「挿入されたブロック」を構成している）ことは既に述べた。修正されたセグメントのコンテンツは更新処理により更新パッケージを備えていることも図 4 を参照して説明した。したがって、更新されたセグメントのコンテンツがどのようなものであるかを更新パッケージに基づいて予測することが可能である。

30

【 0 0 8 8 】

したがって、一実施形態によれば、更新セグメントの修正された部分に対応するシグニチャを更新パッケージに含めることが可能である。シグニチャは、例えば、特定のブロック用に演算で求められた、それ自体が公知のチェックサムでありうる。シグニチャは、その対応するブロックと共に記憶され得る。しかし、シグニチャを、更新パッケージの始まりまたは終わりにおいて、異なるテーブル等に記憶するといった別の実施形態も可能である。

40

【 0 0 8 9 】

一部のセグメントのコンテンツは、元のバージョンから更新されたバージョンにコピーされることも（例えば、図 3 および図 4 を参照して）説明した。したがって、元のバージョンおよび更新パッケージが与えられるならば、更新されたバージョンにおける全セグメントに対するシグニチャを演算で求めることが可能である。さらに、更新されたバージョンの一部であるブロック内の全てのセグメントに対応する 1 つのシグニチャを演算で求めることが可能である。このようなシグニチャを、「ブロックの更新されたシグニチャ」と呼ぶ。

【 0 0 9 0 】

50

本実施形態によれば、「ブロックの元のシグニチャ」、すなわち、元のバージョンの一部であるブロック内の全てのセグメントに対応するシグニチャを計算することもさらに可能である。

【0091】

このことを理解し、また、記憶されたバージョンが、元のバージョン、中間バージョンまたは更新されたバージョンのいずれでもありうる（例えば、図2を参照）という説明をふまえると、当業者であれば、記憶されたバージョンにおけるブロックのシグニチャを計算し、それらをブロックの元のシグニチャおよびブロックの更新されたシグニチャと比較することにより、記憶されたバージョンのIDを決定することが可能であることを理解するであろう。さらに、記憶されたバージョンが中間バージョンであると判断された場合、前に動作していた更新処理の中断位置を決定することが可能である。そして、中断位置に対応する更新パッケージにおける更新コマンドを見つけることが可能である。

10

【0092】

ブロックの一部を占めるコンテンツへのアクセスを許可する記憶装置においては、前に動作していた更新処理の中断が起きたセグメントを決定するために、セグメントのシグニチャを比較することがさらに可能である。

【0093】

「状態レコード」と呼ぶ更新パッケージレコードと対応付けることがさらに可能である。一実施形態によれば、状態レコードを各ブロックに対応するコマンドと対応付けることが可能であり、状態レコードは、対応するブロックの更新が完了した後にあるはずなので、更新処理の状態を反映している。例えば、状態レコードは、それ自体が公知である局所変数等の処理を維持する変数を保持するために更新処理によって用いられるブロックのコピーをRAM内に含みうる。

20

【0094】

記憶されたバージョンのID、前に動作していた更新処理の中断位置および状態レコードを決定した後、本発明の一実施形態によれば、更新処理を反転（すなわち逆戻り）させ、それにより、前に動作していた更新処理の実行開始前に存在していた元のバージョンに復帰することが可能である。

【0095】

図5は、本発明の一実施形態に係る、更新処理を反転させる前に実行される動作を示すフローチャートである。501において、更新パッケージに従って、更新シーケンスが決定され、502において、更新処理は、更新シーケンスに従って、記憶されたバージョンの最初のブロックを読み出す。503において、読み出されたブロックに対応するシグニチャ（例えば、これを更新パッケージから読み出すことにより）へのアクセスが行われ、502で読み出されたコンテンツに基づいて対応するシグニチャが504において計算される。505において、更新処理が2つのシグニチャが一致することを確認し、かつ未検査のブロック（506を参照）が他にある場合、507において、更新シーケンスに従って、次のブロックに対応するシグニチャへのアクセスが行なわれ、504において対応するシグニチャを計算するために、508において当該ブロックが読み出される。未検査のブロックが他に存在しないことが506で分かった場合、これは記憶されたバージョンが更新されたバージョンであることを意味し、更新処理は反転することなく終了する。しかし、これには限らず、その他の実施形態では、更新されたバージョンである記憶されたバージョンであっても反転させることができる。

30

40

【0096】

505において2つのシグニチャが一致しない場合、510での処理状態を復元するために、509で行なわれたように、更新処理は、読み出されたブロックに対応する状態レコードへのアクセスを行うことができる。一旦状態が復元されると、元のバージョンに復帰するために、511において処理は反転する。

【0097】

なお、一例によれば、ブロックを更新している間、更新処理は、その元のコンテンツを

50

バックアップバッファに記憶し、ブロックの更新が完了する前に更新処理が失敗した場合にそのコンテンツの復元を可能にする（「バックアップ・ファースト」）。別の例では、ブロック自体の変化を引き起こすことなく、元のコンテンツの代わりに、更新されたブロックのコンテンツをバックアップバッファに記憶し、（「アップデート・ファースト」）、その後、更新されたコンテンツをバックアップバッファから当該ブロックにコピーすることができる。バックアップ・ファーストの場合、中断点を見つける際、このブロックのコンテンツが破損している可能性がある。この場合、バックアップバッファからコンテンツを回復することができる。あるいは、「アップデート・ファースト」の場合、新コンテンツは既にバックアップバッファに存在している。

【0098】

10

なお、図5に限らないことに注意されたい。別の実施形態においては、更新パッケージにおけるシグニチャは、更新されたバージョンではなく、元のバージョン内のブロックに対応しているかもしれない。当業者は、更新パッケージの対応するシグニチャにシグニチャが一致しない記憶されたバージョンにおける最初のブロックを見ることによって、中断点を見つけるためにシグニチャを比較する代わりに、この別の実施形態においては、更新パッケージの対応するシグニチャにシグニチャが一致する記憶されたバージョンにおける最初のブロックを探すことが要求され、中断点は、更新シーケンスによる記憶されたバージョンの前のブロックであることを理解するであろう。

【0099】

ロールバックは、ユーザ（携帯端末の所持者）の選択等により行なわれる場合があることは既に説明した。所持者によって開始されたロールバックの場合、開始位置の探索は必ずしも必要ではない。本発明の一実施形態によれば、更新処理をキャンセルする際、開始位置の指示を記憶することが可能である（指示は、例えば、方向指示と対応付けて記憶され得る）。

20

【0100】

反転がどのように行なわれうるかについて説明を始める前に、ロールバック更新を反転させ、それにより、順方向更新を達成することも可能であることに注目されたい。例えば、所持者が順方向更新をキャンセルする（それにより、ロールバックを開始する）場合、所持者は、考え直し、（ロールバック更新処理をキャンセルすることにより）更新処理を再び反転させて順方向更新に復帰し、更新されたバージョンを得るかもしれない。すなわち、一実施形態によれば、更新処理をキャンセルする際、順方向方向指示とロールバック方向指示との間で「切り替えを行う」ことが可能である。

30

【0101】

なお、「反転」という用語は、順方向更新からロールバック更新への反転およびその逆を示しているため、反転の前に動作している更新処理を反転前の更新処理と呼び、反転の後に動作している更新処理を、反転後の更新処理と呼ぶ。反転前の更新処理が順方向更新を行う場合、反転後の更新処理はロールバックを行うことが理解される。この反対もまた正しく、反転前の更新処理がロールバックを行う場合、反転後の更新処理は順方向更新を行う。

【0102】

40

同様に、更新処理（順方向またはロールバック）が動作を開始する前に存在する記憶されたバージョンを、通常、ソースバージョンと呼び、更新処理後の記憶されたバージョンを、通常、「ターゲットバージョン」と呼ぶ。ソースバージョンは、元のバージョン、更新されたバージョンまたは中間バージョンでありうる。ターゲットバージョンは、元のバージョンまたは更新されたバージョンのいずれでもありうる。

【0103】

本発明の一実施形態によれば、反転（順方向更新からロールバックへ、およびその逆）を可能にするために、反転前の更新処理中に削除されたあらゆるコンテンツをバックアップすることが必要である。なぜならば、このコンテンツをターゲットバージョン内に再導入するために必要となるからである。図6は、本発明の一実施形態に係る、反転前の更新

50

処理中に削除されたコンテンツの記憶の例を示している。同図においては、本例におけるソースバージョンである元のバージョン301（図3を参照）の1つのセグメント307が示されている。図4の更新パッケージ402の一部も示されており、図示された部分は、セグメント307の削除を担う削除更新コマンド413を含む。この部分は、置換補足データ部分419、挿入補足データ部分420および方向指示421も含む。

【0104】

また、「コンテンツ保持バッファ」601が示されている。「保管（save）バッファ」を構成するコンテンツ保持バッファは、更新処理にアクセス可能な記憶装置103内の領域であり、削除されたコンテンツの保持を可能にするために特別に割当てられている。一実施形態によれば、コンテンツ保持バッファ601を、更新処理の期間のみ一時的に割当て、後に任意の別のタスクのために再利用することができる。更新処理中にコンテンツを削除する前に、そのコピーをコンテンツ保持バッファに保持（602）することが可能である。あるいは、より大きい更新パッケージ（小さい更新パッケージと比べて、より広い帯域幅および/またはより多くの時間を伝送に要する）にして、コンテンツ保持バッファ用の専用スペースを更新パッケージ内に設け、それにより、記憶装置103内にこのバッファに必要なスペースを確保してもよい。

10

【0105】

なお、図6は削除動作中のコンテンツの保持を示している。しかし、これには限らず、実施形態の中には、進行中の更新処理の反転を後に可能にするために、更新処理が「置換」等のコンテンツをオーバーライドする動作を行うたびに保持が起きるものもある。なお、さらに、「置換」動作を行っている間に変換関数を演算する実施形態では、変換データを記憶されたコンテンツと同一のコンテンツに再変換するために、変換データを置換用データに再び適用することができるので、記憶されたコンテンツを置換コンテンツでオーバーライドする前に保持しなくてもよい。

20

【0106】

図6の実施形態は、非限定的なものであり、別の実施形態も可能である。例えば、更新パッケージに削除されたデータのコピーを含めることが可能である。このようにすると更新パッケージ（これはさらに大きくなる）の伝送により多くのネットワークリソースが必要となるが、保持動作が必要でないため更新処理は高速化することになる。

【0107】

図7は、本発明の一実施形態に係る、コンテンツ保持バッファの使用の軽減の例を示している。なお、反転前の更新処理における挿入コマンドおよび置換コマンドに対して用いられる補足データアイテムは、それらの更新コマンドの実行後、それらの補足データは既にターゲットバージョンに書き込まれているため、不要である。しかし、進行中の更新処理中に削除されたコンテンツは、例えばコンテンツ保持バッファに保持されることは既に説明した。したがって、コンテンツ保持バッファに記憶されているコンテンツを、既に使用された補足データ420の代わりに更新パッケージ内にコピーし（701）、それによりコンテンツ保持バッファを解放（および/または再利用）することができることが理解される。

30

【0108】

図7の後者の例によって示されるように、更新パッケージそれ自体をインプレース更新することができ、そのためには、更新処理が中断された場合にその記憶領域を修正するための信頼できる修正方法が必要となりうる。更新パッケージの信頼できるインプレース修正には、上記に説明したようなバックアップバッファ、パッケージングニチャおよび状態レコードの使用といったあらゆる信頼性できる技術を用いることができる。

40

【0109】

本発明の一実施形態によれば、補足データアイテムが「空き」になるたびに（すなわち、これらのそれぞれを使用したほぼ直後に）これらを置換することが可能である。別の実施形態によれば、補足データアイテムを、後で、例えば更新処理の終了時に置換することが可能である。

50

【 0 1 1 0 】

コンテンツ（またはそれらのコンテンツの少なくとも一部）が類似しているか、あるいは共有されている、2つ以上の削除されたセグメントが時として存在することが理解される。別の実施形態によれば、2つ以上のセグメントがコンテンツを共有している場合、このコンテンツは一度だけ保持されることができ、それ以外の場所では、保持されたコンテンツが記憶されている場所、あるいはそれをどのように検索するかについての指示を保持してもよい。

【 0 1 1 1 】

反転前の更新処理が終了すると、先行する更新処理（このような更新処理が開始される場合）が動作すべき方向を示すように方向指示 4 2 1 が設定される。反転前の更新処理が失敗するかあるいは途中で停止した場合にも方向指示 4 2 1 が設定され、先行する更新処理（開始された場合）が順方向モードまたはロールバックモードで動作すべきかが示される（図 5、5 1 1 のように）。

10

【 0 1 1 2 】

本発明によれば、順方向更新処理の反転は、順方向更新処理によって実行される各更新コマンド取り消すことにより行なわれるが、順方向更新処理によって実行された最後の更新コマンドから始まり、更新パッケージにおける最初の更新コマンドで終わるという逆の順序で行なわれる。同様に、ロールバック更新処理の反転は、ロールバック更新シーケンスによって実行された最後に反転された更新コマンドから反転コマンドがロールバック更新処理によって実行された各更新コマンドを再実行することにより行なわれる。

20

【 0 1 1 3 】

図 3 および図 4 において上記に説明した例を参照して反転についてさらに説明する。本例によれば、順方向更新処理の動作が成功し、更新パッケージ 4 0 2 に従って元のバージョン 3 0 1 を更新することにより、更新されたバージョン 3 0 2 が得られたと仮定されている。なお、この成功した順方向更新に続いて、方向指示部 4 2 1 は、更新処理方向を反転すべきであることを指示する（すなわち、先行する更新処理（開始された場合）はロールバックで動作することになる）。

【 0 1 1 4 】

本実施例によれば、ロールバック更新処理は、その先行する順方向更新処理によって実行された更新動作を取り消すが、進行中の処理の終了前に実行された最後の更新コマンドから始まり、最初の更新コマンド 4 1 1 で終わるという逆の順序でこれを行う。本例においては、ロールバック中に取り消されるべき最初の更新コマンドは 4 1 8 である。なお、更新処理を反転している間、補足データ部分（同図においては 4 2 0 および 4 2 1）へのアクセスは逆の順序で行なわれる。

30

【 0 1 1 5 】

本実施例によれば、補足データ部分は、順方向更新中にコンテンツを削除している間にそこに書き込まれた反転されたコンテンツを含む。ロールバック更新は順方向更新と同様（反転された進行方向およびコマンドの反転を除いて）に動作することをふまえると、ロールバック更新処理が記憶されたバージョンからコンテンツを削除するたびに、削除されたコンテンツのコピーが補足データ部分（またはコンテンツ保持バッファ）に書き込まれることが理解される。

40

【 0 1 1 6 】

以下の例は、元の動作の結果を反転させる適切な動作（「反転更新コマンド」または「逆戻し動作」）を行うことにより、各更新コマンド例（コピー、削除、挿入および置換）がどのように取り消されるかを説明するものである。

【 0 1 1 7 】

「削除」動作は、挿入動作により取り消す（ロールバックする、反転する）ことができる。前に削除コマンドによって削除されたコンテンツは、挿入コマンド用の補足データである。例えば、3 0 7 を削除というコマンド（例えばコマンド 4 1 3 を参照）によって行なわれた削除動作は、3 0 7 補足データを挿入によって反転させられる（補足データは、

50

補足データ部分またはコンテンツ保持バッファに記憶され得る)。

【0118】

「挿入」コマンドに従って行なわれた「挿入」動作は、削除動作により取り消すことができる。例えば、コマンド418(311を挿入)は、311を削除を実行することにより取り消すことができる。なお、反転コマンドを実行している間にコンテンツを削除する際、削除されたコンテンツは、例えば、コンテンツ保持バッファまたは挿入補足データ部分内に記憶するとよい。

【0119】

「コピー」動作は、更新されたセグメントをその元の位置にコピーしなおすことにより取り消すことができる。例えば、コマンド417(305を312にコピー)は、312を305にコピーというコマンドにより反転させることができる。

10

【0120】

「置換」動作は、反転前の更新処理によって挿入されたコンテンツを削除し、置換補足データ部分419またはコンテンツ保持バッファ内に記憶されたデータをその代わりに書込むことにより取り消すことができる。換言すると、置換動作は、別の反転された置換動作により取り消すことができる。しかし、反転前の更新処理における「挿入」にビット毎のXORといった変換関数が用いられる場合、反転前のセグメントに記憶されたコンテンツを再び得るために、反転前の挿入動作によって補足データ(変換データ)を用いて前に書き込まれたコンテンツに対して、同じ変換関数を演算することにより挿入動作を反転することができることを当業者であれば理解するであろう。

20

【0121】

したがって、反転された更新コマンド(またはロールバック更新コマンド)は、更新コマンドに逆に対応することが示される。

【0122】

図8は、本発明の一実施形態に係る、反転可能な更新処理においてソースバージョンからターゲットバージョンへの更新の間に行われる主な手順を示すフローチャートである。なお、801において、方向指示部419の値に従って更新処理の方向が決定される。指示部が、更新処理は順方向更新であるべきであると示している場合、802において、更新処理は、更新パッケージに従って順方向に動作することになる。終了(正常終了または失敗)時に、803において、更新処理は前の更新処理(開始された場合)はロールバックを実行すべきであると指示するように方向指示部を設定する。しかし、801において、更新処理が、更新処理はロールバックを実行するように動作すべきであると決定する場合は、804において、更新処理は、更新パッケージに従ってロールバック方向に動作する。終了(正常終了または失敗)時、805において、更新処理は、前の更新処理(開始された場合)は順方向更新を実行すべきであると指示するように方向指示部を設定する。

30

【0123】

図9は、本発明の一実施形態に係る順方向更新を詳細に示すフローチャートである。開始時、901において、更新処理は、記憶されたバージョンにおいて更新を開始すべき位置を見つける必要がある。記憶されたバージョンが元のバージョンである場合、開始位置はバージョンの始まりであり、記憶されたバージョンが更新されたバージョンである場合、開始位置はバージョンの終わりであり(したがって、順方向更新処理は、更新動作を行うことなく終了することになる)、記憶されたバージョンが中間バージョンである場合、開始位置は前に動作していた更新処理の終了点であることが理解される(例えば図5を参照)。

40

【0124】

902において、更新処理は、その処理の状態を開始する。記憶されたバージョンが中間バージョンである場合、状態は状態レコードに従って開始される。しかし、記憶されたバージョンが元のバージョンまたは更新されたバージョンである場合、状態は新しいものとして開始される。また、この段階で、更新処理は、読み出しを開始すべき更新コマンド

50

を見つけ、置換補足データ部分 4 1 9、挿入補足データ部分 4 2 0 およびコンテンツ保持バッファ 6 0 1 内などにおけるポインタ位置を決定し、それにより、そこに記憶された正しいデータアイテム（読み取りおよび/または書き込みのために）へのアクセスを行うことが可能となる。

【 0 1 2 5 】

記憶されたバージョンが更新されたバージョンである場合、更新処理が順方向更新処理であるので、更新シーケンスによる更新コマンドがもはや存在せず、したがって、更新処理は、更新コマンドを実行することなく終了することが 9 0 3 において示されている。しかし、記憶されたバージョンが元のバージョンまたは中間バージョンである場合、更新処理は、更新シーケンスに従って更新パッケージから次の更新コマンドを読み出し、それに
10
10 応じて動作する。更新コマンドが「コピー」コマンド（9 0 5）である場合、9 0 6 において、元のセグメントのコンテンツが更新されたセグメント内にコピーされる。なお、ターゲットセグメントが空きでない場合、実施形態によっては、そこに記憶されたコンテンツを、コピーコマンドの前に削除コマンドが存在するようにコンテンツ保持バッファ内（または、挿入補足データ部分 4 2 0 内）に保持することが必要である。

【 0 1 2 6 】

コマンドが「挿入」コマンド（9 0 7）である場合、9 0 8 において、データは挿入補足データ部分 4 2 0 から更新されたセグメント内にコピーされ、9 0 9 において、挿入補足データ部分内のポインタが次のデータアイテムに進められる。ここでもまた、実施形態によっては、補足データを更新されたセグメントに挿入する前に、更新されたセグメントが
20
20 空きでない場合、そこに記憶されたコンテンツは、コピーコマンドの前に削除コマンドが存在するようにコンテンツ保持バッファ内（または挿入補足データ部分 4 2 0 内）に保持され得る。

【 0 1 2 7 】

コマンドが「削除」コマンド（9 1 0）である場合、9 1 1 において、元のセグメントに記憶されたコンテンツはコンテンツ保持バッファ内（または挿入補足データ部分 4 2 0 内）にコピーされ、9 1 2 において、コンテンツ保持バッファ内のポインタが次の利用可能なスペースに進められる。

【 0 1 2 8 】

本例（「コピー」コマンド、「挿入」コマンド、「削除」コマンドおよび「削除」コマンドが存在する）によれば、当業者であれば、更新コマンドが、「コピー」でも、「挿入」でも、「削除」でもない場合、それは「置換」にちがいないことを理解するであろう。したがって、9 1 3 において、元のセグメントのコンテンツと変換データとのビット毎の排他的論理和が取られ、その結果が更新されたセグメントに記憶され（9 1 4 を参照）、置換補足データ部分 4 1 9 内の次のデータアイテムに対するポインタは次のデータアイテムに進められる。
30
30

【 0 1 2 9 】

図 1 0 は、本発明の一実施形態に係る、記憶されたバージョンのロールバックを詳細に示すフローチャートである。開始時、1 0 0 1 において、更新処理は、記憶されたバージョンにおいて更新を開始すべき位置を見つける必要がある。記憶されたバージョンが元のバージョンである場合、開始位置はバージョンの始まりであり（したがって、ロールバック更新処理は、更新動作を行うことなく終了することになる）、記憶されたバージョンが更新されたバージョンである場合、開始位置はバージョンの終わりであり、記憶されたバージョンが中間バージョンである場合、開始位置は前に動作していた更新処理の終了点であることが理解される（例えば図 5 を参照）。
40
40

【 0 1 3 0 】

1 0 0 2 において、更新処理は、その処理の状態を開始する。記憶されたバージョンが中間バージョンである場合、状態は状態レコードに従って開始される。しかし、記憶されたバージョンが元のバージョンまたは更新されたバージョンである場合、状態は新しいものとして開始される。また、この段階で、更新処理は、読み出しを開始すべき更新コマン
50
50

ドを見つけ、置換補足データ部分 4 1 9、挿入補足データ部分 4 2 0 およびコンテンツ保持バッファ 6 0 1 内などにおけるポインタ位置を決定し、それにより、そこに記憶された正しいデータアイテム（読み取りおよび/または書き込みのために）へのアクセスを行うことが可能となる。

【 0 1 3 1 】

ロールバック更新処理は、更新パッケージの更新シーケンスと逆であるロールバック更新シーケンスに従って動作する。記憶されたバージョンが元のバージョンである場合、更新処理がロールバック更新処理であるので、ロールバック更新シーケンスによる更新コマンドはもはや存在せず、したがって、更新処理は更新コマンドを実行することなく終了することが 1 0 0 3 において示されている。しかし、記憶されたバージョンが更新されたバージョンまたは中間バージョンである場合、ロールバック更新処理は、反転された更新シーケンスに従って更新パッケージから次の更新コマンド（言い換えると、更新シーケンスによる前の更新コマンド）を読み出し、更新コマンドに逆に対応する更新動作を実行する。

10

【 0 1 3 2 】

更新コマンドが「コピー」コマンド（1 0 0 5）である場合、1 0 0 6 において、更新されたセグメントのコンテンツが元のセグメント内にコピーされ、これは、「挿入」コマンドに逆に対応する更新コマンドである。なお、元のセグメントが空きでない場合、一部の実施形態では、そこに記憶されたコンテンツを、コピーコマンドの前に削除コマンドが存在するかのようにコンテンツ保持バッファ内（または挿入補足データ部分 4 2 0 内）に

20

【 0 1 3 3 】

コマンドが「挿入」コマンド（1 0 0 7）である場合、1 0 0 8 において、データは記憶されたバージョンからコンテンツ保持バッファ内（または挿入補足データ部分 4 2 0 ）にコピーされ、これは、「挿入」コマンドに逆に対応する更新コマンドであり、1 0 0 9 において、コンテンツ保持バッファに対するポインタが次の利用可能なスペースに進められる（または、挿入補足データ部分に対するポインタが前のデータアイテムに後退させられる）。

【 0 1 3 4 】

コマンドが「削除」コマンド（1 0 1 0）である場合、1 0 1 1 において、データは挿入補足データ部分 4 2 0 から（またはコンテンツ保持バッファから）元のセグメント内にコピーされ、1 0 1 2 において、挿入補足データ部分内のポインタが前のデータアイテムに後退させられる。ここでもまた、補足データを元のセグメント内にコピーする前に、元のセグメントが空きでない場合、実施形態によっては、そこに記憶されたコンテンツは、削除コマンドの前に挿入コマンドが存在するかのようにコンテンツ保持バッファ内（または挿入補足データ部分 4 2 0 内）にロールバック更新シーケンスに従って保持され得る。

30

【 0 1 3 5 】

本例（「コピー」コマンド、「挿入」コマンド、「削除」コマンドおよび「削除」コマンドが存在する）によれば、当業者であれば、更新コマンドが、「コピー」でも、「挿入」でも、「削除」でもない場合、それは「置換」にちがいないことを理解するであろう。したがって、1 0 1 3 において、更新されたセグメントのコンテンツと変換データとのビット毎の排他的論理和が取られ、その結果が元のセグメントに記憶される（1 0 1 4 を参照）。本実施形態によれば、変換データは置換補足データ部分 4 1 9 に記憶される。したがって、1 0 1 5 において、置換補足データ部分 4 1 9 内の次のデータアイテムに対するポインタは前のデータアイテムに後退させられる。

40

【 0 1 3 6 】

本発明は、記憶されたバージョンの更新処理を実現するための方法を提供する。この方法は、前記記憶されたバージョンを第 1 のバージョンに更新するための順方向更新を実現するステップと、前記記憶されたバージョンを第 2 のバージョンに更新するためのロールバック更新を実現するステップと、前記順方向更新および前記ロールバック更新のうちの

50

いずれかである第1の更新動作を適用するステップと、前記順方向更新および前記ロールバック更新のうちのいずれかであって、前記第1の更新動作以外の第2の更新動作を適用するステップとを含む。

【0137】

ある実施形態によれば、第1のバージョンは、元のバージョンおよび更新されたバージョンを含むいくつかのバージョンのうちの1つでありうるということが理解される。しかし、これには限らず、その他の実施形態では、中間バージョン等の別のバージョンが第1のバージョンでありうる。

【0138】

同様に、ある実施形態によれば、第2のバージョンは、元のバージョンおよび更新されたバージョンを含むいくつかのバージョンのうちの1つでありうるということが理解される。しかし、これにもまた限らず、その他の実施形態では、中間バージョン等の別のバージョンが第2のバージョンでありうる。

10

【0139】

また、ある実施形態によれば、第1の更新動作は、順方向更新動作およびロールバック更新動作等のいくつかの更新動作のうちの1つでありうるということが理解される。これには限らず、その他の実施形態では、異なる更新動作が第1の更新動作でありうる。

【0140】

ある実施形態によれば、第2の更新動作は、順方向更新動作およびロールバック更新動作等のいくつかの更新動作のうちの1つでありうることも理解される。これには限らず、その他の実施形態では、異なる更新動作が第2の更新動作でありうる。

20

【0141】

図11は、本発明の一実施形態に係る、記憶装置に記憶されたコンテンツの記憶されたバージョンを更新するためのシステム1101を概略的に示すブロック図である。システム1101は、例えば方向指示421に従って更新方向を決定するように構成された方向決定ユニット1102を含む。

【0142】

システム1101は、方向決定ユニット1102に接続された順方向更新処理装置1103も含む。方向決定ユニット1102が更新方向は順方向を示していると判断すると、順方向更新処理装置1103は、更新シーケンスに従い、記憶されたバージョンを更新されたバージョンに順方向に更新する。これは、例えば、図9のフローチャートに従って行なわれる。

30

【0143】

方向決定ユニット1102が更新方向はロールバックを示していると判断すると、方向決定ユニット1102に接続されたロールバックシーケンス生成器1104が、更新シーケンスとは逆のロールバック更新シーケンスを生成する。ロールバックシーケンス生成器1104は、一度に1コマンドずつ、逆の順序で更新シーケンスに従うことができる。あるいは、全て一度に更新シーケンスに従い、それにより、ロールバック更新シーケンスを生成し、記憶装置内に、例えばリストとして記憶する。

【0144】

システム1101は、ロールバックシーケンス生成器1104に接続されたロールバック更新処理装置1105も含む。ロールバック更新処理装置1105は、ロールバック更新に従って、記憶されたバージョンを元のバージョンにロールバックする。これは、例えば、図10のフローチャートに従って行なわれる。

40

【0145】

図12は、本発明の一実施形態に係る、更新処理を反転させるためのシステムを概略的に示すブロック図である。システム1201は、更新コマンド反転ユニット1202を含む。更新コマンド反転ユニット1202は、更新中に前に実行された更新パッケージ内の更新コマンドを反転させる。例えば、図10の1006、1008、1011および1013を参照されたい。

50

【0146】

図13は、記憶装置に記憶されたコンテンツの記憶されたバージョンを更新するためのシステム1301を概略的に示すブロック図である。システム1301は、方向決定ユニット1302を含む。一実施形態によれば、方向決定ユニット1302は、方向を決定するのに方向指示を用いる。

【0147】

方向決定ユニット1302には、順方向更新処理装置1303が接続されている。方向が順方向を示す場合、順方向更新処理装置1303は、例えば図9のフローチャートに従い、記憶されたバージョンを更新されたバージョンに順方向に更新する。

【0148】

システム1301は、ロールバック更新処理装置1304も含む。方向決定ユニット1302が方向はロールバックであると判断すると、ロールバック更新処理装置1304は、例えば図10のフローチャートに従い、記憶されたバージョンを元のバージョンにロールバックする

本発明によるシステムは、適切にプログラムされたコンピュータでありうることも理解される。また、本発明は、本発明の方法を実施するための、コンピュータによって読み取り可能なコンピュータプログラムを想定している。本発明は、さらに、本発明の方法を実施するための、機械によって実行可能な命令プログラムを具体的に実施する機械可読メモリを想定している。

【0149】

本発明をある程度詳細に説明したが、以下の請求項の範囲を逸脱することなく、様々な変更と修正がなされうることを当業者は容易に理解するであろう。

【図面の簡単な説明】

【0150】

【図1】図1は、本発明の一実施形態に係る、セルラーネットワークにおいてバージョンを更新するためのシステムの概略図である。

【図2】図2は、元のバージョンから更新されたバージョンへの更新における簡略化された段階例を示す。

【図3】図3は、元のバージョン例および対応する更新されたバージョン例を示す。

【図4】図4は、図3の元のバージョンから、その更新されたバージョンに更新するように構成された更新パッケージ例を示す。

【図5】図5は、本発明の一実施形態に係る、更新処理を反転させる前に実行される動作を示すフローチャートである。

【図6】図6は、本発明の一実施形態に係る削除されたコンテンツの記憶の例を示す。

【図7】図7は、本発明の一実施形態に係る、コンテンツ保持パuffァの使用の軽減を示す。

【図8】図8は、本発明の一実施形態に係る、反転可能な(reversible)更新処理において元のバージョンから更新されたバージョンへの更新の間に実行される主な手順を示すフローチャートである。

【図9】図9は、本発明の一実施形態に係る、記憶されたバージョンの順方向更新を詳細に示すフローチャートである。

【図10】図10は、本発明の一実施形態に係る、記憶されたバージョンのロールバックを詳細に示すフローチャートである。

【図11】図11は、本発明の一実施形態に係る、記憶装置に記憶されたコンテンツの記憶されたバージョンを更新するためのシステムを概略的に示すブロック図である。

【図12】図12は、本発明の一実施形態に係る、更新処理を反転させるためのシステムを概略的に示すブロック図である。

【図13】図13は、本発明の一実施形態に係る、記憶装置に記憶されたコンテンツの記憶されたバージョンを更新するためのシステムを概略的に示すブロック図である。

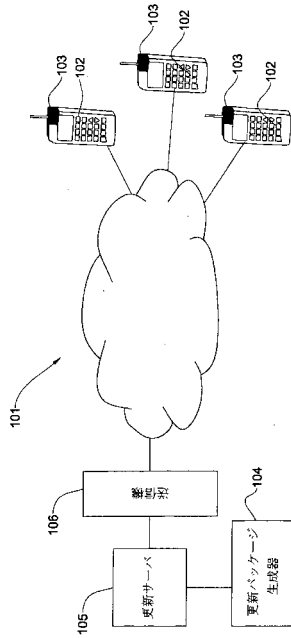
10

20

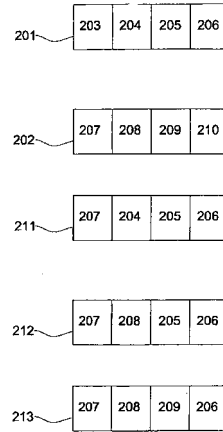
30

40

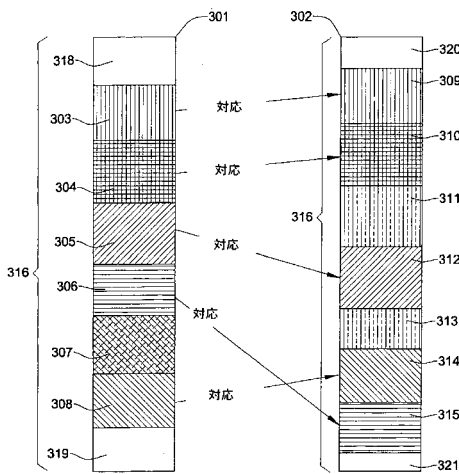
【図1】



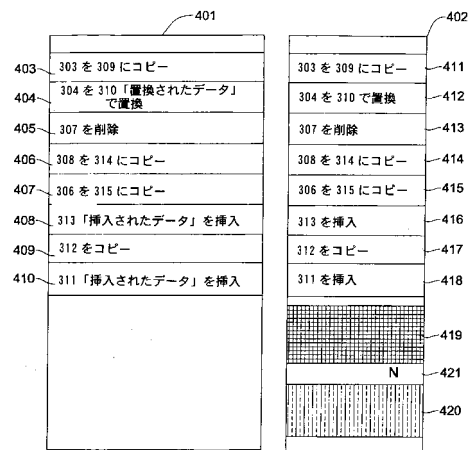
【図2】



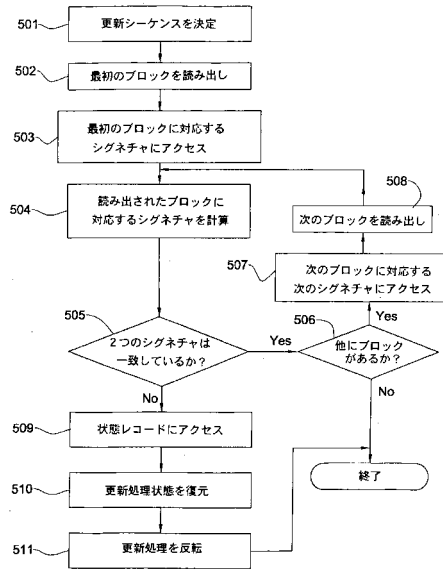
【図3】



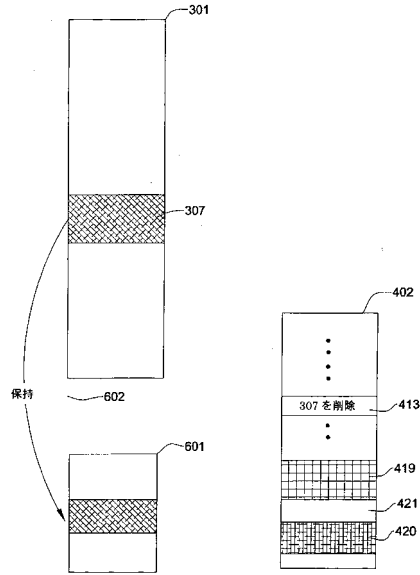
【図4】



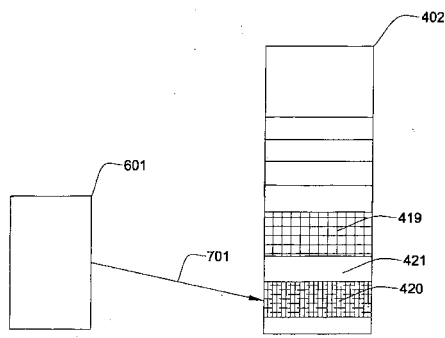
【図5】



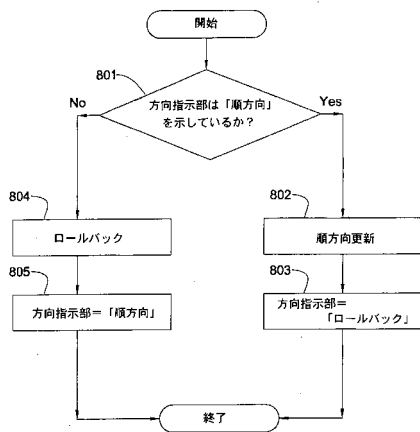
【図6】



【図7】



【図8】



【図9】

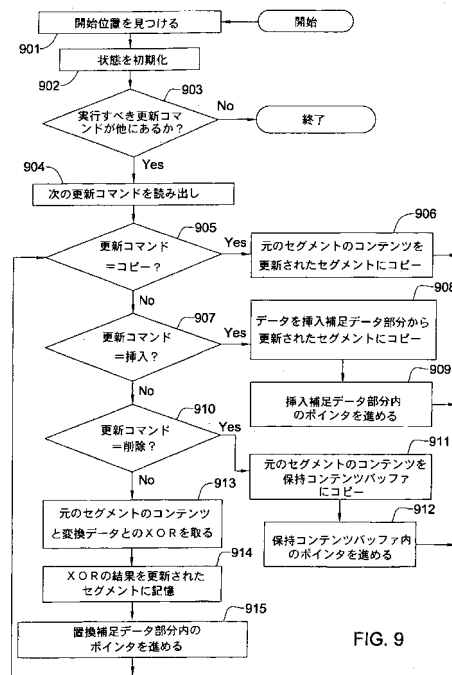


FIG. 9

【図10】

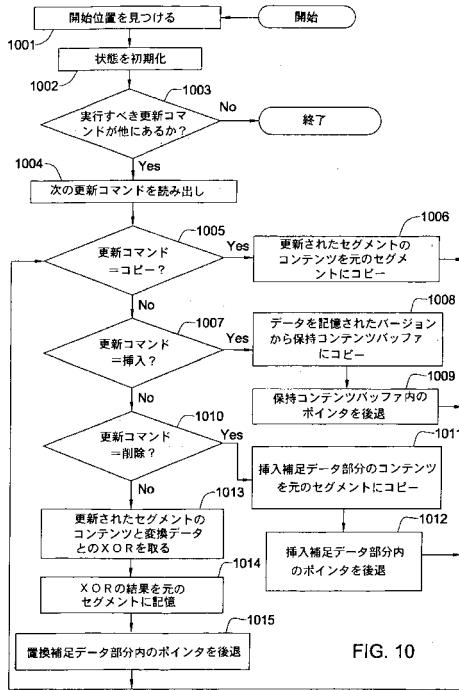
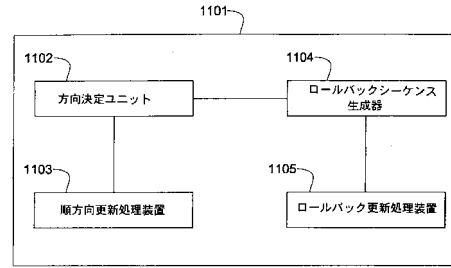
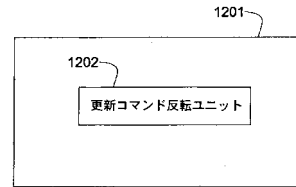


FIG. 10

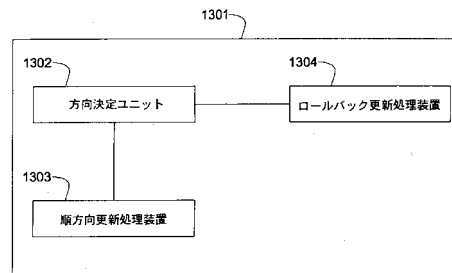
【図11】



【図12】



【図13】



フロントページの続き

審査官 林 毅

(56)参考文献 米国特許第06438749(US, B1)

特開2003-345747(JP, A)

特開平02-214936(JP, A)

特開2003-208316(JP, A)

特開平08-263349(JP, A)

特開平04-052755(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 11/00

G06F 9/445

G06F 12/00