

(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2009-536395

(P2009-536395A)

(43) 公表日 平成21年10月8日(2009.10.8)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 9/445 (2006.01)	G06F 9/06 610Q	5B176
G06F 11/00 (2006.01)	G06F 9/06 630B	

審査請求 有 予備審査請求 未請求 (全 51 頁)

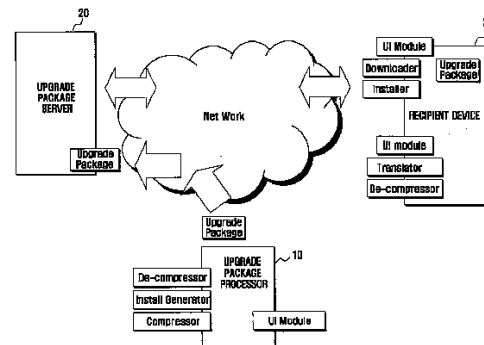
(21) 出願番号 特願2009-509449 (P2009-509449) (86) (22) 出願日 平成19年6月18日 (2007. 6. 18) (85) 翻訳文提出日 平成20年11月10日 (2008. 11. 10) (86) 国際出願番号 PCT/KR2007/002939 (87) 国際公開番号 W02007/148899 (87) 国際公開日 平成19年12月27日 (2007. 12. 27) (31) 優先権主張番号 10-2006-0054746 (32) 優先日 平成18年6月19日 (2006. 6. 19) (33) 優先権主張国 韓国 (KR)	(71) 出願人 503447036 サムスン エレクトロニクス カンパニー リミテッド 大韓民国キョンギード, スウォンシ, ヨ ントンク, マエタンードン 416 (74) 代理人 100094145 弁理士 小野 由己男 (74) 代理人 100121382 弁理士 山下 託嗣 (72) 発明者 オ スーンジョ 大韓民国, キョンギード, スウォンシ, グウォンソング, グウォンソンドン, ウーナム アパートメント, ナンバー 1 14-403, 441-390 Fターム(参考) 5B176 AC04 EA07 EA18 最終頁に続く
---	--

(54) 【発明の名称】 OTAが可能な端末の情報アップグレードシステム及び方法 [INFORMATIONUPGRADE SYSTEM AND METHOD FOR OTA-CAPABLE DEVICE]

(57) 【要約】

【課題】本発明は、システムアップグレード方法に関し、特にOTAプログラムメカニズムを用いた移動端末のプログラム(作動ファームウェア及びアプリケーションソフトウェアを含む)をアップデートする方法及びシステムに関する。

【解決手段】本発明は、OTAプログラムメカニズムを利用した移動端末のプログラムをアップデートするための方法及びシステムを提供する。システムのバージョンアップグレード方法は、アップグレードパッケージプロセッサが第1バージョン及び第2バージョンのプログラムを入力され、二つのバージョンプログラムの差に基づいてアップグレードパッケージを生成するステップと、アップグレードパッケージサーバーがアップグレードパッケージを受信し、アップグレードパッケージの生成を一つ以上の受信装置に通知及びダウンロードするステップと、受信装置がアップグレードパッケージをダウンロードしてアップグレードパッケージを格納するメモリの領域にインストールし、アップグレード要求時、第1バージョンプログラムにアップグレードパッケージの中か



【特許請求の範囲】**【請求項 1】**

ネットワーク上で情報をアップグレードする方法において、

アップグレードパッケージプロセッサで、第 1 バージョン及び第 2 バージョンの差に基づいてアップグレードパッケージを生成するステップと、

前記アップグレードパッケージの生成を少なくとも一つ以上の受信装置に通知するステップと、

前記受信装置が、前記アップグレードパッケージサーバーから前記アップグレードパッケージをダウンロードするステップと、

前記アップグレードパッケージを第 1 メモリに格納するステップと、

10

前記第 1 メモリに既格納された第 1 バージョンと前記アップグレードパッケージとを合成して第 2 バージョンを生成するステップと、

アップグレード要求時、前記第 2 バージョンを第 2 メモリにロードするステップと、を備えることを特徴とする情報アップグレード方法。

【請求項 2】

前記アップグレードパッケージは、前記第 1 バージョンと結合するためのヒストリーデータと、前記アップグレードパッケージを前記第 1 バージョンに結合するためのマップ情報からなるマップデータと、前記マップデータによって前記第 1 バージョンに結合されるアップグレードデータと、を有することを特徴とする請求項 1 に記載の情報アップグレード方法。

20

【請求項 3】

前記マップデータの前記マップ情報は、コマンド及びブロックインデックス情報を含み、前記コマンドは、ブロックインデックス及びデータが同一の場合はコピーコマンドを、第 2 バージョンのブロックデータが挿入又は変更された場合は変更コマンドを、前記第 2 バージョンのブロックデータが一定ブロック移動されて前記第 1 バージョンのブロックデータと同一である場合は移動コマンドをそれぞれ発生し、前記アップグレードデータは、前記変更コマンドのブロックインデックスに対応するデータであることを特徴とする請求項 2 に記載の情報アップグレード方法。

【請求項 4】

前記アップグレードパッケージを生成する過程が、

30

前記入力された第 1 及び第 2 バージョンを予め設定されたサイズのブロックに区分し、前記二つのバージョンのブロックデータを比較するステップと、

前記マップデータにおいて変更コマンドブロックデータをアップグレードデータに生成するステップと、

前記ヒストリーデータ、マップデータ、及びアップグレードデータを結合して前記アップグレードパッケージに生成するステップと、

を有することを特徴とする請求項 3 に記載の情報アップグレード方法。

【請求項 5】

前記第 1 及び第 2 バージョンを比較する過程が、

前記入力された第 1 及び第 2 バージョンを予め設定されたサイズに区分し、前記区分されたブロックデータを圧縮するステップと、

40

前記圧縮された第 1 及び第 2 バージョンブロックデータを比較するステップと、を備え、

前記マップデータにおいて変更コマンドブロックデータをアップグレードデータに生成するステップと、

前記ヒストリーデータ、マップデータ、及びアップグレードデータを結合して前記アップグレードパッケージに生成するステップと、を備えることを特徴とする請求項 3 に記載の情報アップグレード方法。

【請求項 6】

前記アップグレードパッケージを圧縮するステップをさらに備えることを特徴とする請

50

求項 5 に記載の情報アップグレード方法。

【請求項 7】

前記第 2 バージョンを生成する過程が、

前記アップグレードパッケージを前記第 1 バージョンと前記アップグレードパッケージとを格納するための多数の格納領域を含む不揮発性メモリに格納するステップと、

前記アップグレードパッケージと前記第 1 バージョンとをマップして前記第 2 バージョンを生成するステップと、

前記第 2 バージョンを揮発性メモリにロードするステップと、

を有することを特徴とする請求項 3 に記載の情報アップグレード方法。

【請求項 8】

前記第 2 バージョンを生成する過程が、

前記マップデータを分析するステップと、

前記マップデータのコマンドがコピーの場合、前記ブロックインデックスに該当する前記第 1 バージョンのブロックデータを前記第 2 バージョンにコピーするステップと、

前記マップデータのコマンドが移動の場合、前記ブロックインデックスに対応するブロックだけシフトされた前記第 1 バージョンのブロックデータを前記第 2 バージョンに移動して格納するステップと、

前記マップデータが変更の場合、前記ブロックインデックスに対応する第 2 バージョンにアップグレードデータを挿入するステップと、を有することを特徴とする請求項 7 に記載の情報アップグレード方法。

【請求項 9】

前記アップグレード要求は、キー入力や前記受信装置の初期化によって発生することを特徴とする請求項 8 に記載の情報アップグレード方法。

【請求項 10】

前記情報は、プログラムであることを特徴とする請求項 8 に記載の情報アップグレード方法。

【請求項 11】

アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置が前記アップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むネットワーク上でプログラムをアップグレードする方法において、

第 1 バージョンと第 2 バージョンとの差によるアップグレードパッケージをアップグレードパッケージプロセッサで生成するステップと、

前記アップグレードパッケージの生成を少なくとも一つ以上の受信装置に通知するステップと、

前記受信装置が前記アップグレードパッケージサーバーからアップグレードパッケージをダウンロードするステップと、

前記アップグレードパッケージを不揮発性メモリに格納するステップと、

前記アップグレードパッケージと前記不揮発性メモリに既格納された第 1 バージョンを合成して第 2 バージョンを生成するステップと、

アップグレード要求によって前記第 2 バージョンを揮発性メモリにロードするステップと、を備え、

前記アップグレードパッケージを生成する過程は、

前記入力された第 1 及び第 2 バージョンを比較するステップと、

前記比較結果によってアップグレードデータを生成するステップと、

前記比較結果によってアップグレードデータをインストールするためのインストールデータを生成するステップと、

前記インストールデータと前記アップグレードデータとを合併してアップグレードパッケージを生成するステップと、を有することを特徴とするプログラムアップグレード方法。

。

10

20

30

40

50

【請求項 1 2】

アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置がアップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むプログラムアップグレード方法において、

前記アップグレードパッケージサーバーにおいて、第 1 バージョンと第 2 バージョンとの差に基づいてアップグレードパッケージを生成するステップと、

前記アップグレードパッケージの生成を少なくとも一つ以上の受信装置に通知するステップと、

前記受信装置が前記アップグレードパッケージサーバーから前記アップグレードパッケージをダウンロードするステップと、

前記アップグレードパッケージを不揮発性メモリに格納するステップと、

前記不揮発性メモリに既格納された第 1 バージョンと前記アップグレードパッケージとを合成して第 2 バージョンを生成するステップと、

アップグレード要求時、前記第 2 バージョンを揮発性メモリにロードするステップと、を備え、

前記第 2 バージョンを生成する過程は、

第 1 メモリに前記ダウンロードされたアップグレードパッケージを格納するステップと

アップグレード要請に応じて前記第 1 バージョンと前記アップグレードパッケージとをロード及び合成して前記第 2 バージョンを生成するステップと、を有することを特徴とするプログラムアップグレード方法。

【請求項 1 3】

アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置がアップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むプログラムアップグレード方法において、

アップグレードパッケージプロセッサから前記第 1 バージョンと第 2 バージョンとを受信するステップと、

前記第 1 バージョンと第 2 バージョンとを比較するステップと、

前記第 1 バージョンと結合するためのアップグレードデータを生成するステップと、

前記アップグレードデータと第 1 バージョンとを結合するための方法を表すインストールデータを生成するステップと、

前記アップグレードデータとインストールデータとを結合してアップグレードパッケージを生成するステップと、

前記アップグレードパッケージサーバーにおける前記アップグレードパッケージの生成を少なくとも一つの受信装置に通知するステップと、

前記アップグレードパッケージサーバーからのアップグレードパッケージを受信装置にダウンロードするステップと、

前記ダウンロードされたアップグレードパッケージを不揮発性メモリにインストールするステップと、

アップグレード要請によって前記第 1 バージョンと前記アップグレードパッケージとを結合して前記第 2 バージョンを生成するステップと、

前記受信装置を作動するために揮発性メモリに前記第 2 バージョンをロードするステップと、を含むことを特徴とするプログラムアップグレード方法。

【請求項 1 4】

前記第 1 ディスプレイ装置と前記第 2 ディスプレイ装置との自動オンオフ制御を実行し、前記第 1 ディスプレイ装置と前記第 2 ディスプレイ装置との前記自動オンオフ制御に関する情報を設定するプログラムを記憶する記憶装置をさらに備える、

請求項 1 3 に記載のディスプレイのオンオフ制御装置。

【請求項 1 5】

前記アップグレードパッケージプロセッサは、

前記第 1 バージョンと前記アップグレードパッケージとを結合する方法を提供するヒストリーデータと、前記アップグレードパッケージを前記第 1 バージョンにマップする情報を提供するマップデータと、を含むインストールデータを発生するインストールデータ発生器と、

前記マップデータによってアップグレードデータを発生し、前記インストールデータと前記アップグレードデータとを結合して前記アップグレードパッケージを生成するパッケージ発生器と、

を有することを特徴とする請求項 1 4 に記載のプログラムアップグレードシステム。

【請求項 1 6】

前記マップデータの前記マップ情報は、コマンド及びブロックインデックス情報を含み、前記コマンドは、ブロックインデックス及びデータが同一であるとコピー、前記第 2 バージョンのブロックデータが挿入又は変更されると変更、前記第 2 バージョンのブロックデータが一定ブロックだけ移動されて前記第 1 バージョンのブロックデータと同一であると移動コマンドが発生され、前記アップグレードデータは、前記変更コマンドのブロックインデックスに対応するデータであることを特徴とする請求項 1 5 に記載のプログラムアップグレードシステム。

【請求項 1 7】

前記アップグレードパッケージプロセッサは、

前記入力された第 1 及び第 2 バージョンのプログラムを設定されたサイズのブロックに区分し、前記二つのバージョンのブロックデータを比較する比較器を有し、

前記マップデータによって変更コマンドブロックデータからアップグレードデータを生成した後、前記ヒストリーデータ、マップデータ、及びアップグレードデータを結合してアップグレードパッケージを生成することを特徴とする請求項 1 4 に記載のプログラムアップグレードシステム。

【請求項 1 8】

前記アップグレードパッケージプロセッサは、

前記入力された第 1 及び第 2 バージョンを設定されたサイズに区分し、前記区分されたブロックデータを圧縮する圧縮機をさらに有することを特徴とする請求項 1 7 に記載のプログラムアップグレードシステム。

【請求項 1 9】

前記アップグレードパッケージプロセッサは、前記アップグレードパッケージを圧縮符号化する第 2 圧縮機をさらに有することを特徴とする請求項 1 8 に記載のプログラムアップグレードシステム。

【請求項 2 0】

前記受信装置が、

前記第 1 バージョンを格納する領域及び少なくとも一つのアップグレードパッケージ格納領域を備える第 1 メモリと、

アップグレードされた前記第 2 バージョンをロードする第 2 メモリと、

前記ダウンロードされたアップグレードパッケージを前記第 1 メモリのアップグレードパッケージ格納領域に格納するインストーラーと、

アップグレード要求時、前記アップグレードパッケージを前記第 1 バージョンにマップして前記第 2 バージョンを生成し、前記生成された第 2 バージョンを前記第 2 メモリに格納するトランスレータと、を有することを特徴とする請求項 1 8 に記載のプログラムアップグレードシステム。

【請求項 2 1】

前記トランスレータが、

前記マップデータをロードした後、前記マップデータのコマンドがコピーであると、ブロックインデックスに該当する前記第 1 バージョンのプログラムのブロックデータを前記第 2 バージョンにコピーし、前記マップデータのコマンドが移動であると、前記ブロックインデックスに対応するブロックだけシフトされた前記第 1 バージョンのブロックデータ

10

20

30

40

50

を前記第 2 バージョンに移動して格納し、前記マップデータが変更であると、前記ブロックインデックスに対応する第 2 バージョンにアップグレードデータを挿入して前記第 1 バージョンを第 2 バージョンにアップグレードすることを特徴とする請求項 20 に記載のプログラムアップグレードシステム。

【請求項 22】

前記アップグレード要求は、前記アップグレードパッケージをインストールした後、これを表示し、ユーザーの要求によって発生することを特徴とする請求項 21 に記載のプログラムアップグレードシステム。

【請求項 23】

前記アップグレード要求は、電源のオン時、システム初期化の実行後に行なわれることを特徴とする請求項 21 に記載のプログラムアップグレードシステム。

【請求項 24】

システムのバージョンをアップグレードするシステムにおいて、

第 1 バージョン及び第 2 バージョンが入力され、前記二つのバージョンの差に基づいてアップグレードパッケージを生成するアップグレードパッケージプロセッサと、

前記アップグレードプロセッサから前記アップグレードパッケージを受信し、前記アップグレードパッケージの生成を一つ以上の受信装置に通知するアップグレードパッケージサーバーと、

前記アップグレードパッケージをダウンロードし、前記第 1 バージョンを格納する第 1 メモリに前記ダウンロードされたアップグレードパッケージを格納し、アップグレード要求時、前記第 1 バージョンに前記アップグレードパッケージを合成し第 2 バージョンを生成して第 2 メモリに格納し、前記第 2 メモリの第 2 バージョンによってシステムを運用する受信装置と、

を備え、

前記アップグレードパッケージプロセッサは、

前記第 1 及び第 2 バージョンを比較する比較器と、

前記比較結果によってアップグレードデータをインストールするためのインストールデータを生成するインストールデータ発生器と、

前記比較結果によってアップグレードデータを生成し、前記インストールデータ及びアップグレードデータを結合して前記アップグレードパッケージを生成するパッケージ発生器と、

を有することを特徴とするプログラムアップグレードシステム。

【請求項 25】

プログラムアップグレードシステムにおいて、

第 1 バージョン及び第 2 バージョンが入力され、前記二つのバージョンの差に基づいてアップグレードパッケージを生成するアップグレードパッケージプロセッサと、

前記アップグレードプロセッサから前記アップグレードパッケージを受信し、前記アップグレードパッケージの生成を一つ以上の受信装置に通知するアップグレードパッケージサーバーと、

前記アップグレードパッケージをダウンロードし、前記第 1 バージョンを格納する第 1 メモリに前記ダウンロードされたアップグレードパッケージを格納し、アップグレード要求時、前記第 1 バージョンに前記アップグレードパッケージを合成し第 2 バージョンを生成して第 2 メモリに格納し、前記第 2 メモリの第 2 バージョンによってシステムを運用する受信装置と、

を備え、

前記受信装置は、

前記ダウンロードされたアップグレードパッケージを前記第 1 メモリのアップグレードパッケージ格納領域にインストールするインストーラーと、

アップグレード要求時、前記アップグレードパッケージを前記第 1 バージョンにマップして第 2 バージョンを生成し、前記生成された第 2 バージョンを前記第 2 メモリに格納す

10

20

30

40

50

るトランスレータと、

を有することを特徴とするプログラムアップグレードシステム。

【請求項 26】

プログラムアップグレードシステムにおいて、

第 1 及び第 2 バージョンを比較する比較器と、前記比較結果によってアップグレードデータをインストールするためのインストールデータを生成するインストールデータ発生器と、前記比較結果によってアップグレードデータを生成し、前記インストールデータ及びアップグレードデータを結合して前記アップグレードパッケージを生成するパッケージ発生器とを有するアップグレードパッケージプロセッサと、

前記アップグレードパッケージの生成を一つ以上の受信装置に通知するアップグレードパッケージサーバーと、

10

前記第 1 バージョンを格納する領域及び少なくとも一つのアップグレードパッケージ格納領域を備える第 1 メモリと、アップグレードされた第 2 バージョンを格納する第 2 メモリと、前記ダウンロードされるアップグレードパッケージを前記第 1 メモリのアップグレードパッケージ格納領域にインストールするインストーラーと、アップグレード要求時、前記アップグレードパッケージを前記第 1 バージョンにマップして第 2 バージョンを生成し、前記生成された第 2 バージョンを前記第 2 メモリに格納するトランスレータと、を有する受信装置と、を備えることを特徴とするプログラムアップグレードシステム。

【発明の詳細な説明】

【技術分野】

20

【0001】

本発明は、システムアップグレード方法およびシステムに関し、特に O T A プログラムメカニズムを用いた移動端末のプログラム(作動ファームウェア及びアプリケーションソフトウェアを含む)をアップデートする方法及びシステムに関する。

【背景技術】

【0002】

一般的に、端末システムに含まれるソフトウェア(s o f t w a r e)又はファームウェア(f i r m w a r e)等は、多くの検証過程を経て商品化される。しかし、端末システムに採択された全てのソフトウェア又はファームウェアは、プログラムのバグ又はプログラムのバージョン変更等によってプログラムをアップグレードしなければならない。プログラムの変更が、一般的類型の使用中に発生すると、端末システムを供給する製造会社やサービスキャリア(s e r v i c e c a r r i e r)等には、このような問題を解決するためにソフトウェア又はファームウェア等をアップグレードする義務がある。ここで、製造会社及びサービスキャリアのアップグレードサービスは、よりレベルの高いソフトウェア又はファームウェアへの交替も可能である。

30

【0003】

アップグレードサービスは、主にユーザーがサービスセンターを訪問して受けるのが一般的である。ここで、端末機が無線通信機能を備える携帯端末機である場合、アップグレードサービスは、O T A (o v e r t h e a i r) 形態でも可能である。

【発明の開示】

40

【発明が解決しようとする課題】

【0004】

O T A の場合、ユーザーの端末アップグレードのために、パッチソフトウェア(p a t c h s o f t w a r e)を伝送してもらえらるダウンロードモジュールと現在の端末のソフトウェアバージョンをアップグレードできるソフトウェアパッケージ(s o f t w a r e p a c k a g e)を用いてアップグレードするアップグレード処理モジュールが必要である。しかし、従来 O T A が可能なほとんどの装置は、その安定性に問題があった。

【課題を解決するための手段】

【0005】

本発明は、アップグレードパッケージを用いてファームウェアをアップグレードできる

50

OTAが可能な装置のファームウェアアップグレードシステム及び方法を提供する。

【0006】

本発明は、オリジナルバージョン及び少なくとも一つのアップグレードバージョンを第1メモリに格納し、第2メモリにオリジナルバージョンの情報をアップグレードバージョンの情報にアップグレードする情報を格納し、システムが第2メモリの情報を用いてシステムを運用することができるシステム及び方法を提供する。

【0007】

本発明は、オリジナルバージョン情報と少なくとも一つのアップグレードバージョン情報とを格納しており、アップグレードバージョン情報は、インストールデータ及びアップグレードデータから構成されるシステムにおいて、システム起動又はユーザーの初期化命令によってオリジナルバージョンの情報をアップグレードバージョンの情報にアップグレードし、システムを運用するシステム及び方法を提供する。

【0008】

本発明は、オリジナルバージョンとアップグレードバージョンの情報を比較してアップグレードデータを生成し、アップグレードデータをアップグレードバージョンの履歴情報とマージしてアップグレードパッケージを生成した後、情報をアップグレードするシステムに伝送することができるシステム及び方法を提供する。

【0009】

本発明は、アップグレードデータ及びアップグレードバージョンの履歴情報がマージされたアップグレードパッケージを受信するシステムにおいてアップグレードパッケージ受信時、アップグレードパッケージのインストールデータを抽出した後、これをアップグレードパッケージと共に第1メモリに格納し、アップグレード命令時、第2メモリに、オリジナルバージョンにインストールデータによってアップグレードバージョンのデータをアップグレードして格納し、第2メモリの内容によってシステムを運用することができるシステム及び方法を提供する。

【0010】

本発明の一局面では、アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置が前記アップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むネットワーク上でプログラムをアップグレードする方法によって上記目的が達成される。当該プログラムをアップグレードする方法は、アップグレードパッケージプロセッサで、第1バージョン及び第2バージョンの差に基づいてアップグレードパッケージを生成するステップと、前記アップグレードパッケージの生成を少なくとも一つ以上の受信装置に通知するステップと、前記受信装置が、前記アップグレードパッケージサーバーから前記アップグレードパッケージをダウンロードするステップと、前記アップグレードパッケージを第1メモリに格納するステップと、前記第1メモリに既格納された第1バージョンと前記アップグレードパッケージとを合成して第2バージョンを生成するステップと、アップグレード要求時、前記第2バージョンを第2メモリにロードするステップと、を備える。

【0011】

本発明の他の局面では、アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置が前記アップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むネットワーク上でプログラムをアップグレードする方法によって上記目的が達成される。当該プログラムをアップグレードする方法は、第1バージョンと第2バージョンとの差によるアップグレードパッケージをアップグレードパッケージプロセッサで生成するステップと、前記アップグレードパッケージの生成を少なくとも一つ以上の受信装置に通知するステップと、前記受信装置が前記アップグレードパッケージサーバーからアップグレードパッケージをダウンロードするステップと、前記アップグレードパッケージを不揮発性メモリに格納するステップと、前記アップグレードパッケージと前記不揮発性メモリに既格納された第1バージョンを合成して第2バージョンを生成するステップと、アップグレード要求によって前記第2バージョン

ンを揮発性メモリにロードするステップと、を備える。前記アップグレードパッケージを生成する過程は、前記入力された第1及び第2バージョンを比較するステップと、前記比較結果によってアップグレードデータを生成するステップと、前記比較結果によってアップグレードデータをインストールするためのインストールデータを生成するステップと、前記インストールデータと前記アップグレードデータとを合併してアップグレードパッケージを生成するステップと、を有する。

【0012】

本発明の他の局面では、アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置がアップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むプログラムアップグレード方法によって上記目的が達成される。当該プログラムアップグレード方法は、において、前記アップグレードパッケージサーバーにおいて、第1バージョンと第2バージョンとの差に基づいてアップグレードパッケージを生成するステップと、前記アップグレードパッケージの生成を少なくとも一つ以上の受信装置に通知するステップと、前記受信装置が前記アップグレードパッケージサーバーから前記アップグレードパッケージをダウンロードするステップと、前記アップグレードパッケージを不揮発性メモリに格納するステップと、前記不揮発性メモリに既格納された第1バージョンと前記アップグレードパッケージとを合成して第2バージョンを生成するステップと、アップグレード要求時、前記第2バージョンを揮発性メモリにロードするステップと、を備える。前記第2バージョンを生成する過程は、第1メモリに前記ダウンロードされたアップグレードパッケージを格納するステップと、アップグレード要請に応じて前記第1バージョンと前記アップグレードパッケージとをロード及び合成して前記第2バージョンを生成するステップと、を有する。

【0013】

本発明の他の局面では、アップグレードパッケージを生成するためのアップグレードパッケージプロセッサと、受信装置がアップグレードパッケージをダウンロードできるようにするアップグレードパッケージサーバーとを含むネットワークにおけるプログラムアップグレード方法によって上記目的が達成される。当該プログラムアップグレード方法は、アップグレードパッケージプロセッサから前記第1バージョンと第2バージョンとを受信するステップと、前記第1バージョンと第2バージョンとを比較するステップと、前記第1バージョンと結合するためのアップグレードデータを生成するステップと、前記アップグレードデータと第1バージョンとを結合するための方法を表すインストールデータを生成するステップと、前記アップグレードデータとインストールデータとを結合してアップグレードパッケージを生成するステップと、前記アップグレードパッケージサーバーにおける前記アップグレードパッケージの生成を少なくとも一つの受信装置に通知するステップと、前記アップグレードパッケージサーバーからのアップグレードパッケージを受信装置にダウンロードするステップと、前記ダウンロードされたアップグレードパッケージを不揮発性メモリにインストールするステップと、アップグレード要請によって前記第1バージョンと前記アップグレードパッケージとを結合して前記第2バージョンを生成するステップと、前記受信装置を作動するために揮発性メモリに前記第2バージョンをロードするステップと、を含む。

【0014】

本発明の他の局面では、プログラムアップグレードシステムによって上記目的が達成される。当該プログラムアップグレードシステムは、第1バージョン及び第2バージョンが入力され、前記二つのバージョンの差に基づいてアップグレードパッケージを生成するアップグレードパッケージプロセッサと、前記アップグレードパッケージプロセッサから前記アップグレードパッケージを受信し、前記アップグレードパッケージの生成を通知するアップグレードパッケージサーバーと、前記アップグレードパッケージをダウンロードし、前記第1バージョンを格納する第1メモリに前記ダウンロードされたアップグレードパッケージを格納し、アップグレード要求時、前記第1バージョンにアップグレードパッケージを合成し第2バージョンを生成して第2メモリに格納し、前記第2メモリの第2バー

ジョンを介してシステムを運用する受信装置と、を備える。

【0015】

本発明の他の局面では、システムのバージョンをアップグレードするシステムによって上記目的が達成される。当該システムのバージョンをアップグレードするシステムは、第1バージョン及び第2バージョンが入力され、前記二つのバージョンの差に基づいてアップグレードパッケージを生成するアップグレードパッケージプロセッサと、前記アップグレードプロセッサから前記アップグレードパッケージを受信し、前記アップグレードパッケージの生成を一つ以上の受信装置に通知するアップグレードパッケージサーバーと、前記アップグレードパッケージをダウンロードし、前記第1バージョンを格納する第1メモリに前記ダウンロードされたアップグレードパッケージを格納し、アップグレード要求時、前記第1バージョンに前記アップグレードパッケージを合成し第2バージョンを生成して第2メモリに格納し、前記第2メモリの第2バージョンによってシステムを運用する受信装置と、を備える。前記アップグレードパッケージプロセッサは、前記第1及び第2バージョンを比較する比較器と、前記比較結果によってアップグレードデータをインストールするためのインストールデータを生成するインストールデータ発生器と、前記比較結果によってアップグレードデータを生成し、前記インストールデータ及びアップグレードデータを結合して前記アップグレードパッケージを生成するパッケージ発生器と、を有する。

10

【0016】

本発明の他の局面では、プログラムアップグレードシステムによって上記目的が達成される。当該プログラムアップグレードシステムは、第1バージョン及び第2バージョンが入力され、前記二つのバージョンの差に基づいてアップグレードパッケージを生成するアップグレードパッケージプロセッサと、前記アップグレードプロセッサから前記アップグレードパッケージを受信し、前記アップグレードパッケージの生成を一つ以上の受信装置に通知するアップグレードパッケージサーバーと、前記アップグレードパッケージをダウンロードし、前記第1バージョンを格納する第1メモリに前記ダウンロードされたアップグレードパッケージを格納し、アップグレード要求時、前記第1バージョンに前記アップグレードパッケージを合成し第2バージョンを生成して第2メモリに格納し、前記第2メモリの第2バージョンによってシステムを運用する受信装置と、を備える。前記受信装置は、前記ダウンロードされたアップグレードパッケージを前記第1メモリのアップグレードパッケージ格納領域にインストールするインストーラーと、アップグレード要求時、前記アップグレードパッケージを前記第1バージョンにマップして第2バージョンを生成し、前記生成された第2バージョンを前記第2メモリに格納するトランスレータと、を有する。

20

30

【0017】

本発明の他の局面では、プログラムアップグレードシステムによって上記目的が達成される。当該プログラムアップグレードシステムは、アップグレードパッケージプロセッサと、アップグレードパッケージサーバーと、受信装置と、を備える。アップグレードパッケージプロセッサは、第1及び第2バージョンを比較する比較器と、前記比較結果によってアップグレードデータをインストールするためのインストールデータを生成するインストールデータ発生器と、前記比較結果によってアップグレードデータを生成し、前記インストールデータ及びアップグレードデータを結合して前記アップグレードパッケージを生成するパッケージ発生器とを有する。アップグレードパッケージサーバーは、前記アップグレードパッケージの生成を一つ以上の受信装置に通知する。受信装置は、前記第1バージョンを格納する領域及び少なくとも一つのアップグレードパッケージ格納領域を備える第1メモリと、アップグレードされた第2バージョンを格納する第2メモリと、前記ダウンロードされるアップグレードパッケージを前記第1メモリのアップグレードパッケージ格納領域にインストールするインストーラーと、アップグレード要求時、前記アップグレードパッケージを前記第1バージョンにマップして第2バージョンを生成し、前記生成された第2バージョンを前記第2メモリに格納するトランスレータと、を有する。

40

50

【発明を実施するための最良の形態】

【0018】

以下、本発明の好ましい実施形態の詳細な説明が添付された図面を参照しながら説明する。図面において、同一構成には、同一符号を付して表している。

【0019】

下記の説明において、アップグレードバージョンブロックの数、マクロブロックのサイズ等のような特定の詳細が、本発明をより理解しやすくするために表されている。これらの特定の詳細がなくても、また、これらの変形によっても本発明を容易に実施できるということは、この技術分野における通常の知識を有する者にとって自明なことである。

【0020】

本発明の実施形態において、「アップグレード (upgrade)」という用語は、システムの情報をアップグレードする動作を意味する。また、「情報 (information)」という用語は、アップグレード対象であり、上記のようにファームウェア及びソフトウェア等のプログラム及び不揮発性データを意味する。ここで、情報におけるプログラムとは、ファームウェア (firmware) 及びソフトウェア (software) 等を意味し、不揮発性データは、受信装置で使用されるフォント (font)、システムイメージ、ユーザーデータ等である。不揮発性データのアップグレード例について説明すると、受信装置が携帯端末機の場合、ユーザーはコンピュータ等を介して携帯端末機で使用するためのデータ (例えば、加入者情報、マルチメディアデータ等) を作成又は購入した後、これをサーバーにアップロードすることができ、このような場合、サーバーは、以前の不揮発性データと比較してアップグレードパッケージを生成し、生成されたアップグレードパッケージを携帯端末機にダウンロードすることができる。以下の説明においては、情報がプログラムである場合を仮定して説明する。ここで、プログラムは、ファームウェア及びソフトウェアにすることができ、本発明の実施形態では、情報アップグレードとして、プログラムのアップグレードパッケージを生成して処理する過程を具体的に説明する。「アップグレードパッケージ」とは、アップグレードデータ及びインストールデータを含む情報を意味する。「アップグレードデータ (upgrade data)」とは、プログラムアップグレード時、旧バージョン情報に代わる新バージョンの情報を生成するためのデータを意味する。「インストールデータ」とは、情報アップグレード時、旧バージョンを用いてアップグレードバージョンを生成する時のインストールデータを意味し、インストールデータは、ヒストリーデータ及びマップ (map) データから構成される。マップデータは、発明の実施形態において、新バージョンのデータを生成するためにコピー (copy)、移動 (shift)、及び変更 (modify) 等のコマンド及びこれらのコマンドを実行するためのブロック位置データを含む。「第1バージョン」は、システムのオリジナルバージョン情報、又はシステムの基準となるバージョンの情報にすることができる。「第2バージョン」は、アップグレードバージョンであり、第1バージョン情報をアップグレードした情報となる。この時、受信側は、第1バージョン情報をダウンロードして格納し、一つ又は二つ以上の第2バージョン情報をアップグレードパッケージとして格納することができる。ここで、「アップグレードパッケージ」は、第1バージョンをアップグレードするための情報であり、第1バージョン情報から第2バージョン情報を生成する時に必要なインストールデータ及びアップグレードデータを含むことができ、本発明の実施形態において、第2バージョンのアップグレードパッケージは、第1バージョン情報をコピー (copy)、シフト (shift)、及び変更 (modify) のコマンド、並びにブロック位置データ、また、これに関するアップグレードデータを含むことができる。「プログラム」は、システムのファームウェア (firmware) 及びソフトウェア (software) を含む用語として使用される。「第1メモリ」は、第1バージョン及び第2バージョンのアップグレードパッケージを格納するメモリを意味する。「第2メモリ」は、第1バージョン情報に第2バージョンアップグレードパッケージを用いて生成された、アップグレードされた情報を格納するメモリを意味する。ここで、第1メモリ及び第2メモリは、一つのメモリ上において第1領域及び第2領域に区分されるメモ

10

20

30

40

50

リとして構成することもでき、又は、2個の別のメモリとして構成することもできる。本発明の実施形態において、第1及び第2メモリが、物理的に別の2個のメモリであると仮定し、ここで、各々「第1メモリ」は、不揮発性メモリ（non-volatile memory）かつフラッシュメモリであり、「第2メモリ」は、揮発性メモリ（volatile memory）かつSRAMであると仮定する。本発明の実施形態において、第1メモリは、第1バージョン情報と少なくとも一つの第2バージョンアップグレードパッケージとを格納し、第2バージョンのアップグレードパッケージは、各バージョン情報を識別するためのヒストリーデータ（マップデータを含む）及びアップグレードデータを各々格納する。システム起動時又はユーザーの命令により、システムは、第1バージョン情報に第2バージョンアップグレードパッケージを用いて生成されたアップグレード情報を第2メモリに格納した後、アップグレードされた第2バージョン情報によってシステムを運用する。ここで、アップグレードパッケージは、インストールデータ及びアップグレードデータから構成される。第1バージョン情報は、システムのオリジナルプログラム、又は基準プログラムにすることができ、第2メモリに格納される第2バージョンのプログラムは、第1バージョンプログラムに第2バージョンのアップグレードパッケージを結合して生成されたプログラムにすることができる。

10

【0021】

本発明の実施形態に係るプログラムアップグレードシステムは、アップグレードパッケージを生成して伝送する送信システム、及びアップグレードパッケージを受信してシステム情報をアップグレードする受信装置等から構成される。図1は、本発明の実施形態により、アップグレードパッケージを生成して伝送し、伝送されたアップグレードパッケージをダウンロードしてシステム情報をアップグレードするシステムの構成を示す図面である。以下の説明では、上記のように、情報がプログラムである場合を仮定して説明する。

20

【0022】

図1を参照すると、アップグレードパッケージプロセッサ10は、第2バージョンプログラム情報の発生時、第1バージョンプログラム情報と比較してその差によるアップグレードパッケージを生成した後、これをアップグレードパッケージサーバー20に伝送する。ここで、アップグレードパッケージプロセッサ10は、アップグレードパッケージサーバー20との連結方法によって無線（air protocols：CDMA，UMTS等）、近距離無線通信（bluetooth，Zigbee等）、近距離有線通信（USB，UART）、有無線インターネット（Wibro，Wifi，Wimax等）方式で伝達される。アップグレードパッケージサーバー20は、アップグレードパッケージが発生したことを一つ以上の受信装置30に通知（notification）し、応答する受信装置30にアップグレードパッケージをダウンロードする。受信装置のアップグレードパッケージダウンロード方法は、上記のように、アップグレードパッケージサーバー20及び受信装置30の連結方法によって無線（air protocols：CDMA，UMTS等）、近距離無線通信（bluetooth，Zigbee等）、近距離有線通信（USB，UART）、有無線インターネット（Wibro，Wifi，Wimax等）方式等を用いることができる。本発明の実施形態では、アップグレードパッケージプロセッサ10とアップグレードパッケージサーバー20とが独立して構成された例を説明しているが、アップグレードパッケージプロセッサ10とアップグレードパッケージサーバー20とは一体型に構成することもできる。

30

40

【0023】

受信装置30は、ダウンロードされるアップグレードパッケージを第2バージョンプログラムを生成するための情報としてメモリに格納する。この時、受信装置30のメモリは、第1メモリ及び第2メモリから構成される。ここで、第1メモリ及び第2メモリは、一つのメモリ上に領域を分割して区分されたメモリに構成されるか、又は、二つの別のメモリにすることもできる。ここで、第1メモリは、ダウンロードされる第1バージョンプログラム及びアップグレードパッケージを格納するメモリであり、第2メモリは、第1バージョンプログラムをアップグレードパッケージによりアップグレードして生成された第2

50

バージョンプログラムを格納するメモリである。すなわち、受信装置 30 は、ダウンロードされるアップグレードパッケージを第 1 メモリに第 2 バージョンプログラムのアップグレード情報として格納する。その後、アップグレード要求時、受信装置は、第 1 バージョンプログラムにダウンロードされたアップグレードパッケージを結合 (merge) して第 2 バージョンプログラムを生成し、生成された第 2 バージョンプログラムを第 2 メモリに格納する。その後、受信装置 30 は、第 2 メモリに格納された、アップグレードされた第 2 バージョンプログラムによってシステムを運用する。

【0024】

次に、アップグレードパッケージプロセッサ 10 の動作について説明する。図 2 は、本発明の実施形態に係るアップグレードパッケージプロセッサ 10 と周辺構成との関係を説明するための図面である。

【0025】

図 2 を参照すると、第 1 バージョンプログラム 50 と第 2 バージョンプログラム 55 とは、アップグレードパッケージプロセッサ 10 に入力される。ここで、第 1 バージョンプログラム 50 は、オリジナルバージョンのプログラムにすることができ、第 2 バージョンプログラム 55 は、アップグレードデータの結合 (merge) によって第 1 プログラム 50 から生成されるプログラムにすることができる。すると、アップグレードパッケージプロセッサ 10 は、まず第 1 バージョンから現在の第 2 バージョンへと移動するための第 2 バージョンのアップグレードパッケージ (upgrade package) を生成する。第 1 バージョン及び第 2 バージョンプログラムを比較し、比較結果によってアップグレードパッケージを生成した後、アップグレードパッケージサーバー 20 に伝達する。アップグレードパッケージは、アップグレードデータ及びインストールデータから構成される。この場合、アップグレードデータは、第 1 及び第 2 バージョンプログラムの差によるデータにすることができる。インストールデータは、ヒストリーデータ及びマップデータから構成され、ヒストリーデータは、第 1 バージョンプログラムと結合するためのデータにすることができ、マップデータは、第 1 及び第 2 バージョンプログラムの比較結果によって第 1 バージョンプログラムに第 2 バージョンプログラムをマップするためのコピー、変更、及びシフトのコマンド及びコマンドのインデックス情報を含むことができる。アップグレードパッケージをアップグレードパッケージサーバー 20 に伝送する方法は、その連結方法によって無線、近距離通信、無線インターネット、有線通信方式中の一つの方法を用いることができる。

【0026】

図 3 は、アップグレードパッケージプロセッサ 10 において、アップグレードパッケージを生成する方法を説明するための図面である。

【0027】

ここで、図 3 a は、第 1 バージョンプログラムの例を示しており、図 3 b は、第 2 バージョンプログラムの例を示している。ここで、第 2 バージョンプログラムは、第 1 バージョンプログラムをそのままコピー (copy) 又は修正 (modify) することができ、場合によって、修正されたデータの量によりその情報の位置がシフト (shift) されることもある。

【0028】

図 4 は、図 2 のアップグレードパッケージプロセッサ 10 の構成例を示す図面であり、第 1 及び第 2 バージョンプログラムを圧縮し、圧縮された二つのバージョンプログラムを比較分析してマップデータ及びアップグレードデータを生成し、マップデータを含むインストールデータ及びアップグレードデータを合成してアップグレードパッケージを生成するアップグレードパッケージプロセッサの構成を示す図面である。

【0029】

また、図 5 は、図 2 のアップグレードパッケージプロセッサ 10 のさらに他の構成例を示す図面であり、第 1 及び第 2 バージョンを圧縮し、圧縮された二つのバージョンを比較分析してアップグレードデータを生成し、マップデータを含まないインストールデータ及

10

20

30

40

50

びアップグレードデータを合成してアップグレードパッケージを生成するアップグレードパッケージプロセッサの構成を示す図面である。

【0030】

図6は、図4のようなアップグレードパッケージプロセッサ10で生成されるアップグレードパッケージの構成を示しており、図7は、図5のようなアップグレードパッケージプロセッサ10で生成されるアップグレードパッケージの構成を示している。

【0031】

図8は、図2のアップグレードパッケージプロセッサ10の構成例を示すさらに他の図面であり、圧縮していない第1及び第2バージョンを比較分析してアップグレードデータを生成し、マップデータを含むインストールデータ及びアップグレードデータを合成してアップグレードパッケージを生成するアップグレードパッケージプロセッサ10の構成を示す図面である。しかし、図5と類似の方法で、アップグレードパッケージプロセッサ10は、マップデータを含まないインストールデータを生成した後、これをアップグレードデータに合成してアップグレードパッケージを生成することもできる。

【0032】

まず、図3-図7を参照すると、第1バージョンプログラム50と第2バージョンプログラム55とは、アップグレードパッケージプロセッサ10に入力されるが、図4及び図5のように、第1圧縮機160に圧縮された形態で入力されるか、図8のように、本来の情報そのまま入力される。第1圧縮機160は、第1バージョンプログラムを設定されたサイズ(以下、マクロブロック(macro block)と称する)に分割して圧縮し、また、第2バージョンプログラムもマクロブロック単位で分割して圧縮する。第1圧縮機160が設定されたブロックのサイズに圧縮する理由は、後段の比較器110が二つのバージョンをブロック単位で比較できるようにするためである。比較器110は、圧縮された二つのバージョンの情報を比較して二つのブロックデータが同一であるか否かを表示する。この時、二つのバージョンの対応するブロックデータが同一でないと、比較器110は、第1バージョン及び第2バージョンのブロックデータが設定された区間内の第1バージョンブロックデータと比較する。上記のように、二つのバージョンのブロックデータを比較した後、比較器110は、比較結果値及び対応する第1バージョンのブロックインデックス値をインストールデータ発生器180に伝達する。

【0033】

インストールデータ発生器180は、図4に示すように、ヒストリーデータ発生器120及びマップデータ発生器150から構成されるか、又は、図5に示すように、ヒストリーデータ発生器120のみから構成される。

【0034】

まず、図4を参照すると、ヒストリーデータ発生器120は、第2バージョンのバージョン番号(version number)を含む。すなわち、バージョン番号が5の場合、第2バージョンは、バージョン番号5になり、第1バージョン及びバージョン番号5の第2バージョンを結合してアップグレードするという意味を有する。マップデータ発生器150は、比較器110から出力されるブロックデータを分析してアップグレードパッケージをインストールするためのコマンド(command)を決定するマップデータを生成する。本発明の実施形態において、マップデータに含まれたコマンドは、コピー(copy:C)、変更(modify:M)、及び移動(shift:S)と仮定する。ここで、コピーCは、第1及び第2バージョンのブロックインデックスが同一である場合にマップされるデータを意味し、変更Mは、第1及び第2バージョンの当該ブロックデータのサイズが異なる場合(大きいか、又は小さい場合)、又は、当該ブロックのデータ値が異なる場合にマップされるデータを意味し、移動Sは、第1及び第2バージョンのブロックデータは同一であるが、当該ブロックの位置値が異なる場合にマップされるデータを意味する。上記のようなマップデータは、各ブロックの数に対応するようにマップされ、移動Sの場合にはシフト値を含み、変更Mの場合には該当ブロックの第2バージョンデータをアップグレードデータとして有する。マップデータは、第1及び第2バージョンのブロッ

クデータが同一であるとコピーC、ブロックデータが異なると(ブロックデータの追加、又は内容が異なる場合)変更M、ブロックデータは同一であるが位置が異なると移動Sのマップデータを当該ブロックにマッチして設定する。注意すべき点は、第1バージョンプログラム及びコマンドを用いて第2バージョンプログラムを生成する時、実際にブロックデータが必要なのは変更Mの場合であり、コピーC及び移動Sの場合には、コピー又は移動される第1バージョンのブロック及び位置(ブロックインデックス)情報だけで充分という点である。アップグレードデータは、上記のように、変更Mに該当するブロックのデータにすることができ、コピーC及び移動Sの場合には、ブロック、又は、ブロック及び位置に関する情報がインストールデータとして構成される。

【0035】

上記のように、マップデータ発生器150は、第1及び第2バージョンのブロックインデックス値及び比較結果値を分析してアップグレードパッケージを生成するためのマップデータを生成する。この時、マップデータ発生器150は、上記のように、比較器110から出力される第1バージョン及び第2バージョンのブロックインデックス及び比較結果値によってマップデータを生成する。ここで、第1及び第2バージョンの同一のブロックインデックスで同一の比較値が発生すると、マップデータ発生器150は、該当ブロックのコマンドをコピーCに決定し、このようなブロック番号を累積する。上記のような動作は、第2バージョンの変更ブロックが発生されるか、又は第1バージョンの削除ブロックが発生するまで続く。

【0036】

また、比較器110が、第1及び第2バージョンの同一のブロックインデックスに他の比較値が発生すると、マップデータ発生器150は、第2バージョンの当該ブロックのコマンドを変更M又はシフトSに決定するようにする。ここで、変更Mは、2種類の形態に発生することができる。その一つは挿入であり、もう一つは代替である。マップデータ発生器150は、比較器110から出力される比較結果値及びブロックインデックスを連続して分析する。ここで、ブロックインデックスを分析して第1バージョンに第2バージョンのデータを挿入する変更と判断されると、マップデータ発生器150は、コマンドを変更Mにするマップデータを生成する。また、変更ブロックデータが同じサイズに代替される場合、マップデータ発生器150は、二つのブロックの差データをエントロピー符号化してマップデータに含めることができる。変更ブロックが同じサイズのブロックに代替される場合、続く第2バージョンの後のブロックは、シフト動作が発生しないこともある。すなわち、第1バージョンと同じブロックインデックスを有する第2バージョンのブロックを同じサイズに変更する場合、続く第2バージョンの次のブロックは、シフトしなくなる。また、第2バージョンが第1バージョンの一部のブロックを削除するプログラムである場合、マップデータ発生器150は、続く第2バージョンの次のブロックをシフトさせるマップデータを生成する。このような場合、シフト動作は、第1バージョンの削除されたブロックインデックスでシフトされるべきである。

【0037】

上記のように、変更ブロックが挿入される場合、後のブロックデータをシフトしなければならない。マップデータ発生器150は、変更ブロックを挿入するマップデータを生成した後、その後に入力される比較器110の出力を分析して第1バージョンのブロックインデックスと第2バージョンのブロックインデックスを分析する。そして、第2バージョンのブロックインデックス、シフトするブロックの数、及び第1バージョンと第2バージョンとのシフトブロック数を含むマップデータを生成する。

【0038】

上記のような動作の実行中、アップグレードパッケージ発生器100は、マップデータを発生しないこともある。これは、受信側で第1バージョンプログラムを用いてアップグレードデータをインストールするためのマップデータを生成することができるからである。そのため、インストールデータ発生器180からマップデータ発生器150を省略してもよい。

10

20

30

40

50

【0039】

パッケージ発生器130は、第1圧縮機160において圧縮された第2バージョンプログラム、及び、マップデータ発生器150で生成されたマップデータを分析してアップグレードパッケージを生成する。この時、パッケージ発生器130は、マップデータ発生器150において生成されるマップデータを分析してアップグレードデータを生成するか否かを決定する。すなわち、パッケージ発生器130は、マップデータのコマンドを分析し、コピーC及びシフトSのコマンドを有するブロックの場合には、アップグレードデータを生成しない。しかし、マップデータのコマンドが変更Mの場合、パッケージ発生器130は、比較器110で圧縮された第2バージョンのブロックデータを入力してアップグレードデータに設定する。すなわち、パッケージ発生器130は、マップデータを分析し、コピーC及びシフトSコマンドを有するマップデータであれば、アップグレードデータを生成せず、変更Mコマンドであれば、アップグレードデータを生成する。

10

【0040】

その後、パッケージ発生器130は、インストールデータ発生器180で生成されるインストールデータにアップグレードデータをマージ（merge）してアップグレードパッケージを生成する。ここで、インストールデータは、ヒストリーデータ、又は、ヒストリーデータ及びマップデータにすることができる。すなわち、図4のように、インストールデータ発生器180が、ヒストリーデータ発生器120及びマップデータ発生器150から構成された場合、インストールデータは、ヒストリーデータ及びマップデータとなり、図5のように、インストールデータ発生器180が、ヒストリーデータ発生器120のみから構成された場合、インストールデータは、ヒストリーデータとなる。

20

【0041】

次に、図5を参照すると、インストールデータ発生器180はマップデータ発生器150を備えない。このような場合、パッケージ発生器130は、第2バージョンの各ブロックが、第1バージョンの各ブロックにマップされるブロックインデックス値及びブロックデータに情報を含めるアップグレードデータを生成する。アップグレードデータは、マップデータ発生器150と類似のコマンドを含むことができる。すなわち、アップグレードデータは、第2バージョンプログラムのブロックデータを第1バージョンプログラムでコピーするコマンドは、開始ブロックの番号及びコピーするブロックの数を含む情報を有するように構成し、第2バージョンプログラムのブロックデータを第1バージョンプログラムに追加又は変更するコマンドは、開始ブロックの番号、変更ブロックの数、及び各変更ブロックのデータが順次に連結される情報を有するように構成し、第2バージョンプログラムのブロックデータを、第1バージョンプログラムブロックにシフトするコマンドは、開始ブロックの番号、シフトブロックの数、及び最初シフトブロックに対応する第1バージョンプログラムのブロックインデックスの情報を有するように構成する。

30

【0042】

その後、パッケージ発生器130は、アップグレードデータとヒストリーデータとを結合してアップグレードパッケージを生成し、アップグレードパッケージはアップグレードパッケージサーバー20に伝達される。この時、パッケージ発生器130から生成されたアップグレードパッケージは、第2圧縮機140によって圧縮され、アップグレードパッケージサーバー20に伝達されることもできる。上記のように、マップデータを生成せずアップグレードデータを用いて第2バージョンプログラムのアップグレードパッケージを生成する場合、アップグレードパッケージの生成速度を向上させることができる。

40

【0043】

アップグレードパッケージは、ヒストリーデータ、マップデータ、及びアップグレードデータによって構成することができ、又は、ヒストリーデータ及びアップグレードデータによって構成することができる。パッケージ発生器130から生成されるアップグレードパッケージは、図4のように、ヒストリーデータ及びマップデータから構成されるインストールデータが生成される場合、図6のように生成され、図5のように、ヒストリーデータのみから構成されるインストールデータが生成される場合、図7のように生成される。

50

【0044】

その後、パッケージ発生器130から出力されたアップグレードパッケージは、第2圧縮機140で圧縮され、アップグレードパッケージサーバー20に伝達される。ここで、第2圧縮機140は省略されてもよい。これは、アップグレードパッケージが、すでに第1圧縮機160によって圧縮符号化された状態であるためである。しかし、伝送効率を高めるために、図4に示すように、第2圧縮機140によって再び圧縮することもできる。また、第1圧縮機160で第1及び第2バージョンプログラムを圧縮する場合、第1圧縮解除機165は、圧縮された第1及び第2バージョンプログラムを圧縮解除する。これは、第1圧縮機160で圧縮された第1及び第2バージョンプログラムが正常に圧縮されたかを確認するためのものであり、圧縮時に誤りが発生した場合、第1圧縮機160を制御して再圧縮するように制御することができる。

10

【0045】

図4及び図5は、アップグレードパッケージプロセッサ10が、第1及び第2バージョンを圧縮した後、二つのバージョンを比較分析してアップグレードパッケージを生成する構造を示している。このような場合、比較器110は、圧縮された二つのバージョンプログラムを比較するようになる。しかし、比較器110は、第1及び第2バージョンプログラムを圧縮していないデータで比較することができる。図8は、上記のように、第1及び第2バージョンプログラムを圧縮していないデータで比較してアップグレードパッケージを生成する構造を示している。図8において、第1及び第2バージョンプログラムを圧縮する構成を除く残りの構成は、図4と同一であり、その動作も同一である。また、図8は、インストールデータ発生器180が、ヒストリーデータ及びマップデータを生成する例を示しているが、図5のように、インストールデータ発生器180が、ヒストリーデータ発生器120のみを備えることもできる。

20

【0046】

上記のように、アップグレードパッケージプロセッサ10は、第1バージョンに基づいて第2バージョンを比較し、比較結果によってインストールデータを含むアップグレードパッケージを生成する。この時、比較過程において、第2バージョンが第1バージョンに新しいデータを追加又は削除する変更が発生すると、第2バージョンのブロックは、第1バージョンの当該領域に続く次のブロックに比べてシフトされる結果になる。このような場合、第2バージョンのブロックは、第1バージョンで変更されたブロックによって前へシフトされるか、又は後にシフトされるという結果になる。また、第2バージョンの変更されたブロックは、当該ブロックのデータをエントロピー符号化するようになる。そして、上記のように、変更ブロックの後の第2バージョンのブロックデータは、第1バージョンのブロックデータにおいて、挿入又は削除されたブロック数だけシフトされる。アップグレードパッケージプロセッサ10は、第1バージョン及び第2バージョンを比較した後、コマンドコピーC、変更M、及びシフトSからなるマップデータ(マップデータを用いない場合には、アップグレードデータにこれを含めて生成することができる)を生成し、マップデータ、ヒストリーデータ、及びアップグレードデータ(マップデータを生成しない場合には、ヒストリーデータ及びアップグレードデータ)からなる第2バージョンのアップグレードパッケージを生成し、アップグレードパッケージサーバー20に伝送する。ここで、アップグレードパッケージプロセッサ10でアップグレードパッケージを伝送する方法は、アップグレードパッケージサーバー20との連結方法に応じて無線通信、近距離通信、インターネット通信、又は有線通信方法を用いることができる。

30

40

【0047】

アップグレードパッケージサーバー20は、アップグレードパッケージをダウンロードできる受信装置30に新しいアップグレードパッケージの発生を通知する。そして、通知結果によって応答する受信装置30にアップグレードパッケージをダウンロードする。アップグレードパッケージサーバー20は、アップグレードパッケージの発生を各受信装置に通知する通知サーバー(notification server)をさらに備えることができる。

50

【0048】

受信装置30は、ダウンロードするアップグレードパッケージの発生を通知されると、これに対する応答をしてアップグレードパッケージサーバー20からアップグレードパッケージをダウンロードする。図9は、本発明の実施形態に係る受信装置の構成を示す図面である。

【0049】

図9を参照すると、ダウンローダー(down-loader)220は、アップグレードパッケージサーバー20からアップグレードパッケージをダウンロードし、インストーラー(installer)230は、ダウンロードされたアップグレードパッケージにおけるインストールデータ及びアップグレードデータを分析して第1メモリ250に格納する。ここで、インストールデータは、ヒストリーデータ及びマップデータから構成されるか、又は、ヒストリーデータのみから構成される。ここで、インストールデータがマップデータを含まない場合、第2バージョンの第1バージョンマップ情報はアップグレードデータに含まれる構造を有することができる。ここで、インストーラー230は、マップデータが含まれていないインストールデータを受信した場合、第1バージョンとアップグレードデータとを比較分析してマップデータを生成するか、又は、生成しない。ここで、インストーラー230においてマップデータを生成しない場合、トランスレータ240でアップグレードデータのマップ情報を用いてアップグレードパッケージを直接第1バージョンに結合することもできる。本発明の実施形態では、マップデータを生成すると仮定して説明する。そして、インストーラー230は、第1メモリ250の第2バージョンのアップグレードパッケージを格納する領域でヒストリーデータ、マップデータ、及びアップグレードデータをマッチして格納する。ここで、第1メモリ250には、オリジナルバージョンである第1バージョン及び第1バージョンを第2バージョンに生成するための少なくとも一つのアップグレードパッケージが格納される。ここで、アップグレードパッケージは、少なくとも一つであり、最大N個のアップグレードパッケージを格納することができる。本発明の実施形態では、N=6個と仮定する。

【0050】

また、上記のように、新バージョンのアップグレードパッケージをダウンロードすると、システムは、ユーザーにプログラムのバージョンアップグレード時点であることを表示する。この時、システムアップグレード要求が発生するか、又はシステム初期化の動作を行なった後、受信装置30のトランスレータ(translator)240は、第1メモリ250から第1バージョン及びダウンロードされた第2バージョンのアップグレードパッケージをロードした後、第1バージョンにインストールされたアップグレードパッケージを結合して第2バージョンを生成した後、これを第2メモリ260に格納する。この時、トランスレータ240は、アップグレードパッケージのインストールデータを分析してバージョン番号及び結合するバージョンを決定し、マップデータを分析してアップグレードパッケージのアップグレードデータを結合するバージョン(ここでは、第1バージョンにすることができる)の当該ブロック内容をアップグレードする。上記のような動作を行ない第2バージョンのプログラムにアップグレードすると、トランスレータ240は、これを第2メモリ260に格納し、その後、受信装置30は、第2メモリ260に格納された第2バージョンプログラムを用いてシステムの動作を制御する。

【0051】

上記のように、第1メモリ250には、第1バージョンプログラム及び第2バージョンプログラムを生成するための少なくとも一つのアップグレードパッケージが格納される。ここで、アップグレードパッケージは、インストールデータ(ヒストリー及びマップデータ)及びアップグレードデータ、又は、ヒストリーデータ及びアップグレードデータから構成される。そして、インストールデータは、第1バージョンにアップグレードデータをアップグレードする時に用いるデータであり、アップグレードパッケージのヒストリーデータとアップグレードデータとのマップ関連情報であるマップデータから構成される。ここで、マップデータは、二つのバージョン間の関係を表すものであり、上記のように、三

つの形態 (copy, modify, shift) で表示することができる。マップデータは、特定データ(ここでは、第1バージョン)の比較によってCMS (copy, modify, shift) を見つける生成方法であり、迅速なアドレス計算を目的に使用され、第1メモリ250に含まれている第1バージョンに基づき第2メモリ260で第2バージョンのプログラムを速かに生成するためのデータとして使用される。

【0052】

この時、ダウンロードされるアップグレードパッケージは、アップグレードパッケージプロセッサ10から生成されるが、インストールデータのマップデータは、生成されることもあり、又は、生成されないこともある。受信装置にダウンロードされるアップグレードパッケージにマップデータが含まれることもあり、又は、含まれないこともある。マップデータが含まれていないアップグレードパッケージをダウンロードした場合、インストーラー230は、逆に第1メモリ250に格納された第1バージョンとダウンロードされたアップグレードパッケージとを比較分析してアップグレードデータを第1バージョンにマップするためのマップデータを生成しなければならない。

【0053】

また、トランスレータ240は、第1バージョンプログラムと最も最近ダウンロードされた第2バージョンのアップグレードパッケージとを組み合わせるアップグレードするのが一般的であるが、ユーザーが選択する特定バージョンにアップグレードすることもできる。すなわち、少なくとも一つのバージョンを格納するが、本発明の実施形態では、第2バージョンを生成するためのアップグレードパッケージはN個(本発明の実施形態では、最大6個と仮定している)まで格納することができる。このような場合、特定のアップグレードパッケージが、第2メモリ260で再組合せに失敗した場合、ユーザーに現在システムが保有している再組合せ可能なアップグレードパッケージリストを表示し、ユーザーの選択によるアップグレードパッケージを選択して第2バージョンプログラムを生成することもできる。

【0054】

第1メモリ250は、多数個(ここでは、最大6個と仮定している)のアップグレードパッケージを格納できる領域を備える。アップグレードパッケージは、必要な場合にダウンロードして第1メモリ250の格納領域に格納され、次のアップグレードパッケージがダウンロードされても以前ダウンロードされたアップグレードパッケージは削除しない。そのため、バージョンの変更によるヒストリーデータが累積格納され、第1バージョンの格納状態をそのまま維持する。第1バージョンプログラム及び第2バージョンのアップグレードパッケージがそのまま維持されるため、アップグレードの信頼性 (fault-tolerance) が高い。また、最新の第2バージョンに生成するためのアップグレードパッケージがロードできない場合、ユーザーの選択によって特定バージョンのアップグレードパッケージを利用することができ、全てのバージョンのアップグレードパッケージロードに失敗する最悪の場合においても、オリジナルバージョンである第1バージョンプログラムをロードして使用することができる。

【0055】

図10は、本発明の実施形態に係る受信装置の第1メモリ250の構造を示す図面である。図10において、参照番号310は、第1バージョンを格納する第1格納領域であり、第1バージョンは圧縮された状態、又は圧縮されていない状態で格納される。参照番号320は、少なくとも一つの第2バージョンプログラムを生成するためのアップグレードパッケージを格納する第2格納領域であり、対応する第2バージョンのアップグレードパッケージを格納するための領域にすることができる。すなわち、第2格納領域320に格納されるアップグレードパッケージは、各々対応するバージョンのアップグレードデータ及びインストールデータにすることができ、この時、アップグレードデータは、第1バージョンで変更されるデータ(コマンド、実際のデータ等)及びブロックインデックスにすることができる。第2格納領域320のサイズは、アップグレードパッケージの数によって決定される。例えば、図10に示されたように、一つのアップグレードパッケージ格納領

域のサイズが 300kbyteであり、6個のアップグレードパッケージを格納する場合、第2格納領域320のサイズは1.8Mbyteのサイズに割り当てられる。参照番号330は、ユーザー領域でありファイルシステム(file system)等が格納される。

【0056】

図11は、第1メモリ250の第2格納領域320に格納されるアップグレードパッケージの構造を示す図面であり、図12は、図11のようなアップグレードパッケージにおけるヒストリーデータの構造を示す図面である。

【0057】

図11を参照すると、第2格納領域320は、設定された数(ここでは、6個と仮定している)のアップグレードパッケージを格納できる領域を備え、各アップグレードパッケージの格納領域は、対応するバージョンのヒストリーデータ、マップデータ、及びアップグレードデータを格納できる構造に構成される。この時、アップグレードパッケージは、インストールデータ及びアップグレードデータから構成され、インストールデータは、上記のように(図4及び図5参照)、ヒストリーデータ、又は、ヒストリーデータ及びマップデータにすることができる。また、第2格納領域320には、アップグレードデータのヒストリーデータ及びマップデータが格納される。ここで、アップグレードデータのヒストリーデータを格納する理由は、第1格納領域310に格納された第1バージョンと連結するためである。第1バージョンに関するマップデータ及びアップグレードデータは省略することができる、又は、代わりにナルデータ(null data)を入れることもできる。図11は、アップグレードパッケージが、ヒストリーデータ、マップデータ、及びアップグレードデータ等から構成された例を示している。

【0058】

図12を参照すると、ヒストリーデータ構造は、ヒストリーデータのバージョン(version)、サイズ(size)、結合フラグ(combined flag)、失敗フラグ(fail flag)等から構成される。ここで、バージョン(version)は、アップグレードパッケージのバージョン番号(図11の場合、#2-#7中の一つ)にすることができる、サイズ(size)は、当該バージョンのヒストリーデータのサイズにすることができる、結合フラグ(combined flag)は、当該バージョンのアップグレードパッケージを結合してアップグレードするバージョン番号(ここでは、第1バージョンの番号#1にすることができる)にすることができる、失敗フラグ(fail flag)は、ロード中に失敗(fail)の発生状況を表示するためのフラグにすることができる。このような場合、第1バージョンのヒストリーデータは、バージョン(version)領域に#1が格納され、結合フラグ(combined flag)領域に#1が連結される。また、例えば、任意のアップグレードパッケージ領域のヒストリーデータにおいてバージョン(version)領域に#5が格納され、結合フラグ(combined flag)領域に#1が格納された場合、受信装置30は、#5番号の第2バージョンを#1番号の第1バージョンに結合(merge)してアップグレードするようになる。ダウンロードされるアップグレードパッケージは、図11のような構造で図10のような第1メモリの第2格納領域320に格納される。この時、第2格納領域320にダウンロードされるアップグレードパッケージは、アップグレード要求時、第1格納領域310の第1バージョンに結合されてアップグレードされた第2バージョンプログラムを生成するようになる。

【0059】

以下の説明では、アップグレードパッケージが、マップデータ、ヒストリーデータ、及びアップグレードデータから構成された場合を仮定して説明する。

【0060】

図13は、第1メモリ250及び第2メモリ260の構造を示す図面である。

【0061】

図13を参照すると、アップグレード要求時、図示しないローダーは、第2格納領域3

10

20

30

40

50

20からアップグレードするバージョンのアップグレードパッケージをロードし、トランスレータ240は、ロードされたアップグレードパッケージを第1格納領域310の第1バージョンにマージして第2バージョンプログラムを生成した後、第2メモリ260に格納する。この時、アップグレード要求は、受信装置30のユーザー応答によって決定される。すなわち、受信装置30は、アップグレードパッケージをダウンロードする時点、又は、システム初期化の実行後、アップグレードされていないバージョンのアップグレードパッケージが存在する場合、アップグレード状況を案内するメッセージを表示し、ユーザーがアップグレードを要求した時に、のような動作を行ないバージョンをアップグレードした後、アップグレードされたバージョンのプログラムを第2メモリ260に格納する。その後、受信装置30は、第2メモリ260に格納された第2バージョンプログラムを用いて受信装置30を運用する。

10

【0062】

また、上記のようなアップグレード動作は、受信装置の動作を初期化した後に実行することができる。すなわち、図10に示すように、第1及び第2バージョンプログラムは、互いに結合されていない状態で第1メモリ250に格納されており、アップグレードが要求された時点で第1バージョンプログラムに第2バージョンのアップグレードパッケージを結合して第2バージョンプログラムを生成し、この第2バージョンプログラムを第2メモリ260に格納して使用する。

【0063】

図14は、第2バージョンのアップグレードパッケージが第1メモリ250に格納されていない場合の動作を説明するための図面であり、図15は、第2バージョンのアップグレードパッケージが第1メモリ250の第2領域320に格納された場合の動作を説明するための図面である。

20

【0064】

図14を参照すると、第1メモリ250は、第1バージョンを格納しており、ここで、第1バージョンは、初期バージョン(initial version)のプログラムにすることができる。そして、第1バージョンは、図14に示すように、n個のブロック(B#1-B#n)からなると仮定している。ここで、第2バージョンのインストールデータは、ヒストリーデータとマップデータとからなる。ここで、ヒストリーデータは、上記のように、第1バージョンの結合情報が格納される。図14のような場合、マップデータ領域にはC:B#1-B#nが格納される。

30

【0065】

この時、アップグレード要求命令が発生すると、トランスレータ240は、インストールデータを分析する。この時、図14のような場合、第1メモリ250の第1バージョンをそのまま第2メモリ260にコピーするという命令であるため、トランスレータ240は、第1メモリ250の第1バージョン#1を第2メモリ260にそのままコピーする。受信装置30は、第2メモリ260に、図14のようにコピーされた第1バージョンによって受信装置の動作を制御する。この時、第1メモリ250に格納されるバージョンは、圧縮されて格納される。バージョンが圧縮されて格納された場合、トランスレータ240は、圧縮解除機270によって圧縮されたバージョンの圧縮を解除した後、第2メモリ260にコピーする。また、アップグレードパッケージプロセッサ10が第2圧縮機140によってアップグレードパッケージを圧縮する場合、トランスレータ240は、先にロードされるアップグレードパッケージの圧縮を解除した後、上記のようなトランスレート動作を実行することができる。

40

【0066】

図15を参照すると、第1メモリ250は、第1格納領域310に第1バージョン#1を格納しており、第2格納領域320に第2バージョンのアップグレードパッケージを格納している。ここで、第1バージョンは、初期バージョン(initial version)又は基準に設定されたバージョンにすることができ、第2バージョンのアップグレードパッケージは、アップグレードデータ及びインストールデータから構成される。イン

50

ストールデータは、上記のように、対応する第2バージョン番号、及び結合すべきバージョン(ここでは、第1バージョン)の番号等を格納するヒストリーデータ及びマップデータから構成される。第1バージョンは、図15に示すように、 n 個のブロック($B\#1-B\#n$)からなることを仮定している。ここで、ヒストリーデータは、結合フラグ領域に $\#1$ が格納された場合を仮定しており、マップデータ領域には、 $C:B\#1-B\#2$ 、 $M:B\#3$ 、 $S:[B\#4-B\#n+1][B\#3-B\#n]$ が格納された場合を仮定している。このような場合、マップデータは、第2バージョンの $B\#1-B\#2$ は第1バージョンの $B\#1-B\#2$ をコピー(copy)し、第2バージョンの $B\#3$ はアップグレードデータであり、第2バージョンの $B\#4-B\#n+1$ はシフトされたブロックであり、第1バージョンの $B\#3-B\#n$ と同一であることを意味する。

10

【0067】

アップグレード要求命令が発生すると、トランスレータ240は、インストールデータを分析する。この時、図15のような場合、第1メモリ250の第1バージョンプログラムにアップグレードデータUP#2をアップグレードして第2メモリ260にコピーしなければならない。この時、トランスレータ240は、第1メモリ250の第1バージョンで $B\#1-B\#2$ のデータをコピーし、 $B\#3$ のデータをUP#2のデータに変更し、 $B\#4$ 以後のデータは、第1バージョンの $B\#3$ 以後のデータを移動させてから、これを第2メモリ260にコピーする。このような場合、第2メモリ260には、図15のように、 $B\#3$ のデータはUP#2のアップグレードデータに変更され、 $B\#4$ のデータは設定されたブロック数だけ(ここでは、1ブロック)シフトされた状態になる。その後、受信装置30は、第2メモリ260に、図15のようにアップグレードされた第2バージョンによって受信装置30の動作を制御する。ここで、第1メモリ250に格納される第1バージョン及び第2バージョンのアップグレードパッケージは、伝送時に圧縮されて格納される(アップグレードパッケージプロセッサ10で生成されるアップグレードパッケージを第2圧縮機140を用いて伝送する場合)。第1バージョン及びアップグレードパッケージが圧縮されて格納された場合、トランスレータ240は、第1圧縮解除機270によって圧縮された第1バージョン及びアップグレードパッケージの圧縮を解除することができる。また、第1バージョン及び第2バージョンのプログラムを圧縮した後(アップグレードパッケージプロセッサ10の第1圧縮機160を用いて圧縮する場合)、圧縮された第1バージョン及び第2バージョンのブロックを比較してアップグレードパッケージを生成した場合、トランスレータ240でトランスレートした第2バージョンのブロックデータは圧縮されたデータとなる。このような場合、トランスレータ240は、第2圧縮解除機275を用いて圧縮を解除した後、第2メモリ260に格納する。

20

30

【0068】

図16は、受信装置30が第1バージョンプログラムにアップグレードパッケージを結合して第2バージョンプログラムを生成する過程を総合的に説明するための図面である。

【0069】

図16を参照すると、第1メモリ250は、上記のように、第1バージョンプログラム及び第2バージョンのアップグレードパッケージを格納しており、アップグレード要求時、トランスレータ240は、アップグレードパッケージ210を第1バージョンプログラム250に結合して第2バージョンプログラムを生成した後、これを第2メモリ260に格納する。そのため、受信システム30は、第1メモリ250に格納している第1バージョンのプログラム及び第2バージョンのアップグレードパッケージを第2メモリ260で結合(merge)して使用し、上記のようなアップグレードは、システム初期化の実行後又はアップグレード要求によって繰り返し行なわれる。

40

【0070】

前述のように、本発明の実施形態に係るシステムのプログラムアップグレード方法は、アップグレードパッケージを生成し、生成されたアップグレードパッケージを決定されたネットワーク通信方法によってダウンロードし、ダウンロードされたアップグレードパッケージをインストールして第1格納し、第1格納されたアップグレードパッケージを用い

50

てシステムのアップグレードプログラムを生成して第2格納した後、第2格納されたアップグレードプログラムを用いてシステムを運用する。そのために、本発明の実施形態において、まず、迅速なアップグレードのためにアップグレードパッケージ生成方法を提案し、第2に、アップグレードパッケージをダウンロードした後、迅速な実行のためにインストールデータ処理方法を提案し、第3に、アップグレードパッケージをダウンロードするシステムからダウンロードされたアップグレードパッケージを保管及び処理する方法を提案し、第4に、保管しているアップグレードパッケージを用いてシステムアップグレードを行なう方法を提案する。

【0071】

まず、迅速なアップグレードのためのアップグレードパッケージの生成方法を説明すると、第1バージョンのプログラムと第2バージョンのプログラムとをアップグレードパッケージの生成のために入力し、二つのバージョンは、アップグレードパッケージ生成時、圧縮した状態、又は圧縮されていない状態で比較された後、二つのバージョン間の差を見出す。この時、二つのバージョン間の差は、これを受信するシステムにおいて、第1バージョンにアップグレードパッケージを再組合せするためのインストールデータ (install data: 特に、map dataにすることができる) に生成される。そして、インストールデータと共にアップグレードデータを生成した後、これをアップグレードパッケージに生成する。

【0072】

次に、上記のように生成されたアップグレードパッケージは、受信装置にダウンロードされる。受信装置は、アップグレードパッケージをダウンロードした後、迅速な実行のためにアップグレードパッケージ処理過程を行なう。アップグレードパッケージに含まれるインストールデータは、基準となるプログラム(ここでは、第1バージョンプログラム)との比較から得られ、迅速なアドレス計算を目的に使用される。すなわち、第1メモリ250に格納された第1バージョンとダウンロードされるアップグレードパッケージとを、第2メモリ260で結合する時、インストールデータを用いて迅速にブロック別に処理することができる。

【0073】

次に、アップグレードパッケージをダウンロードした後、迅速な実行のためのインストール処理方法を説明すると、インストールデータは、基準プログラムとの比較から得られるマップデータを見出す生成方法で、迅速なアドレス計算を目的に使用され、第1メモリ250に含まれている第1バージョンとアップグレードパッケージとを、第2メモリ260で再組立することを助ける役割として使用される。この時、マップデータが、アップグレードパッケージに含まれていない場合、受信装置30は、逆に第1メモリ250に含まれた第1バージョンプログラムとアップグレードデータとを比較し、当該マップ機能のためのデータを生成することができる。また、インストールデータは、アップグレードパッケージのヒストリーデータを含んでいる。ヒストリーデータは、第1メモリ250に格納されるアップグレードパッケージ及びアップグレードパッケージが結合されるバージョンを表わすデータを含んでいる。本発明の実施形態では、上記のように、6個のアップグレードパッケージを第1メモリに保管することを仮定して説明しており、この時、特定のアップグレードパッケージが、第2メモリ260で結合に失敗した場合、受信装置30が保管している他のアップグレードパッケージのリストを表示し、ユーザーの選択によるアップグレードパッケージにアップグレードすることができる。

【0074】

次に、アップグレードパッケージの保管及び処理方法を説明すると、アップグレードパッケージは、第1メモリ250上に構成される特定格納領域に保管され、格納領域は複数個備えられる。そのため、アップグレードパッケージはダウンロードされて第1メモリ250に格納され、その次のアップグレードパッケージがダウンロードされた場合にも、以前のアップグレードパッケージは削除されない。特定のアップグレードパッケージがロードされない場合、受信装置30は、ユーザーに保管しているアップグレードパッケージの

リストを提供し、特定バージョンを再組合せするための選択を支援する。そして、最悪の場合、全てのアップグレードパッケージのロードに失敗した場合には、オリジナルバージョンである第1バージョンをロードして使用することができる。

【0075】

図17は、本発明の実施形態に係るアップグレードパッケージ生成及びアップグレード過程を説明するフロー図である。図17の動作は、図1においてアップグレードパッケージプロセッサ10及び受信装置30の動作を中心に示されている。

【0076】

図17を参照すると、アップグレードパッケージプロセッサ10は、411ステップで、アップグレードパッケージを生成するためのバージョンを受信する。ここで、アップグレードパッケージは、新バージョンのプログラムが生成される場合に生成され、生成されるアップグレードパッケージは、基準となるバージョンプログラムと比較されて生成されるデータにすることができる。本発明の実施形態において、基準となるバージョンは、第1バージョンと称しており、新バージョンは、第2バージョンと称している。この時、第1バージョンは、オリジナルバージョン又は他のバージョンと結合される基準バージョンにすることができる。すなわち、第1バージョンは、アップグレードされるバージョンと結合される基準バージョンにすることができる。第2バージョンのアップグレードパッケージは、第1バージョンと結合してアップグレードされた第2バージョンのプログラムを生成するための情報であり、第2バージョンのアップグレードパッケージは、少なくとも一つ以上になる。上記のように、第1及び第2バージョンが受信されると、アップグレードパッケージプロセッサ10は、413ステップで二つのバージョンの差を分析し、415ステップで二つのバージョンの差に基づいてアップグレードパッケージを生成する。この時、アップグレードパッケージは、二つのバージョンの差によって第1バージョンにアップグレードすべきアップグレードデータと、アップグレードデータを第1バージョンに結合するための情報であるインストールデータ等から構成される。インストールデータは、第2バージョンのヒストリーを表示するヒストリーデータと、アップグレードデータを第1バージョンにアップグレードする時のマップ情報を表すマップデータ等から構成される。ここで、マップデータはアップグレードパッケージから省略することができ、このような場合、マップデータは、受信装置30でアップグレードパッケージをインストールする時に生成することができる。

【0077】

上記のように生成されるアップグレードパッケージは、アップグレードパッケージサーバー20に伝達され、アップグレードパッケージサーバー20は、受信装置30にアップグレードパッケージの生成を通知する。アップグレードパッケージの生成を感知した受信装置30が、アップグレードパッケージサーバー20にダウンロードを要求すると、アップグレードパッケージサーバー20は、対応する受信装置30にアップグレードパッケージをダウンロードする。ダウンロード方法は、受信装置30の通信方法に応じて決定される。すなわち、受信装置30が無線通信機能を備える携帯端末機であると、無線通信方式によってCDMA、UMTS、GSM等の無線通信方式でダウンロードされる。また、USB、Bluetooth、インターネットに接続されている場合には、ワイプロ、無線LAN等のような通信方法でアップグレードパッケージをダウンロードすることもできる。

【0078】

受信装置30は、451ステップでアップグレードパッケージをダウンロードした後、453ステップでダウンロードされるアップグレードパッケージを第1メモリ250に格納する。ここで、第1メモリ250は、第1バージョンのプログラムを格納する第1領域310及びアップグレードパッケージを格納する格納領域から構成される第2領域320から構成される。第2領域320は、多数の格納領域から構成され複数のアップグレードパッケージを格納することができる。本発明の実施形態では、第2領域320が6個の格納領域から構成されると仮定している。アップグレードパッケージを格納する格納領域は、インストールデータであるヒストリーデータ及びマップデータを格納する領域とアップ

グレードデータを格納する領域等から構成される。アップグレードパッケージをインストールする過程において、アップグレードパッケージにマップデータが含まれていない場合、受信装置30のインストーラー230は、第1バージョンプログラムにアップグレードデータをマップしてアップグレードするためのマップデータを生成し、第1メモリ250に格納することができる。上記のように、アップグレードパッケージが第1メモリ250に格納されると、受信装置30は、アップグレード要求時、455ステップで第1バージョンプログラムにアップグレードパッケージをマージして第2バージョンのプログラムを生成した後、これを第2メモリ260に格納し、格納された第2バージョンプログラムによってシステムを運用する。ここで、第2メモリ260は、ワークメモリ（work memory）にすることができ、このような場合、第2メモリ260は揮発性メモリに具現することができる。受信装置30は、システム初期化過程の動作が実行されるか、又はアップグレードパッケージがダウンロードされた後、ユーザーによってアップグレード要求命令が生成された場合、第1メモリ250に格納された第1バージョン及び最も最近ダウンロードされた第2バージョンのアップグレードパッケージを結合して生成された第2バージョンのプログラムを第2メモリ260に格納し、システムは、第2メモリ260に格納された第2バージョンを用いて運用する。また、受信装置300は、プログラムのアップグレードが失敗した場合、旧バージョンプログラムを用いて自動でプログラムアップグレードを試みることができる。また、受信装置300は、プログラムアップグレードが失敗した場合、旧バージョンのリストを表示し、ユーザーによって選択されたバージョンを用いて手動でプログラムアップグレードを行なうことができる。また、システム初期化時、ユーザーは必要とするプログラムのバージョンを選択することができ、受信装置30は、ユーザーによって選択されたバージョンのプログラムをシステムのプログラムにアップグレードすることもできる。

10

20

【0079】

以下の説明において、アップグレードパッケージプロセッサ10、アップグレードパッケージサーバー20、及び受信装置30の動作過程を具体的に説明する。

【0080】

まず、アップグレードパッケージを生成する過程を詳細に説明する。

【0081】

図18－図20は、アップグレードパッケージを生成する過程を示すフロー図である。

30

【0082】

図18を参照すると、まず、アップグレードパッケージプロセッサ10は、501ステップで第1及び第2バージョンのプログラムを入力する。その後、アップグレードパッケージプロセッサ10は、503ステップで第1及び第2バージョンのプログラムを設定されたサイズのブロック単位で比較して二つのバージョン間の差を確認した後、これを比較データとして生成する。アップグレードパッケージプロセッサ10は、505ステップ及び507ステップで比較データを用いてアップグレードデータ及びインストールデータを生成する。この時、アップグレードパッケージプロセッサ10は、パッケージ発生器130及びインストール発生器180を各々備える場合、図18に示すように、アップグレードデータ及びインストールデータを並列に生成することができる。しかし、パッケージ発生器130及びインストール発生器180を備えない場合、アップグレードパッケージプロセッサ10は、アップグレードデータ及びインストールデータを順次に生成することができる。この時、アップグレードデータ及びインストールデータの生成順序は、アップグレードデータ－インストールデータの順に生成するか、又はインストールデータ－アップグレードデータの順に生成する。

40

【0083】

ここで、インストールデータは、受信側で第1バージョンにアップグレードパッケージを結合（merge）するために使用するデータであり、上記のように、ヒストリーデータとマップデータとがある。ヒストリーデータは、上記のように、該当するアップグレードパッケージのバージョン、アップグレードパッケージが結合されるバージョン（ここで

50

は、第1バージョンにすることができる)、サイズ等からなり、マップデータは、第1バージョンプログラムが各ブロックデータと第2バージョンプログラムの各ブロックデータとを結合するためのマップ情報(mapping information)にすることができる。インストールデータにおいて、マップデータは、アップグレードパッケージプロセッサ10で生成するか、又は受信側でも生成する。従って、マップデータは、アップグレードパッケージに含めることもでき、又は、含めないこともできる。

【0084】

アップグレードパッケージプロセッサ10は、図19のように、インストールデータ生成時、マップデータを生成することもあり、又は生成しないこともある。図20のように、インストールデータをアップグレードパッケージに結合(merge)する時、マップデータを含めて合成することもでき、又は、マップデータを省略して結合することもできる。図19を参照すると、アップグレードパッケージプロセッサ10は、521ステップでヒストリーデータを生成した後、523ステップでマップデータの生成が必要であるか否かをチェックする。マップデータの生成が必要である場合、アップグレードパッケージプロセッサ10は、525ステップで比較データを用いてアップグレードデータをマップするためのマップデータを生成した後、図18にリターンする。図20を参照すると、アップグレードパッケージプロセッサ10は、531ステップでインストールデータをアップグレードパッケージにマージする時にマップデータも共にマージするかをチェックする。アップグレードパッケージプロセッサ10は、マップデータをマージする場合、533ステップでヒストリー及びマップデータから構成されるインストールデータを、アップグレードパッケージに合成してアップグレードパッケージを生成し、マップデータをマージしない場合、535ステップでヒストリーデータのみから構成されたインストールデータをアップグレードパッケージに結合してアップグレードパッケージを生成する。

【0085】

また、図18において、501ステップでロードされる第1及び第2バージョンのプログラムを圧縮した後(例えば、アップグレードパッケージプロセッサ10の第1圧縮機160によって実行できる)、503ステップへ進行することができ、また、509ステップを実行して生成されたアップグレードパッケージを圧縮することができる(例えば、アップグレードパッケージプロセッサ10の第2圧縮機140によって実行することができる)。この時、圧縮符号化動作を実行する場合、503ステップで比較するバージョンのサイズが縮小されデータの処理時間を短縮することができ、また、アップグレードパッケージを圧縮符号化する場合には、伝送される情報の量を減らすことができる。この時、圧縮符号化動作を実行する場合、アップグレードパッケージプロセッサ10は、これを再び圧縮解除して正常に圧縮符号化が行なわれたかを確認し、正常に圧縮符号化が行なわれた場合にのみ、次のステップへ進行するように制御することができる。

【0086】

図21は、本発明の実施形態に係るアップグレードパッケージプロセッサ10でアップグレードパッケージを生成する過程を示すフロー図であり、図22は、図21において第1及び第2バージョンを圧縮及び検証する過程を示すフロー図であり、図23は、図21においてインストールデータを生成する過程を示すフロー図であり、図24は、図21においてアップグレードパッケージを生成する過程を示すフロー図である。

【0087】

図21-図24を参照すると、アップグレードパッケージプロセッサ10は、551ステップでコンフィギュアファイル(configuration file)を入力する。この時、コンフィギュアファイルには、アップグレードパッケージプロセッサ10の動作を規定するフラグの設定状況を確認することができる。ここで、フラグ中、C_FLAGは、圧縮するか否かを設定するフラグ(compression flag)であり、M_FLAGは、マップデータを生成するか否かを決定するフラグ(map generation flag)であり、V_FLAGは、圧縮された情報を圧縮解除して検討することによって、正常に圧縮されたかを検証するフラグ(verify flag)である。その

後、アップグレードパッケージプロセッサ 10 は、553 ステップで第 1 及び第 2 バージョンのプログラムをロードする。ここで、第 1 バージョンプログラムは、上記のように、携帯端末機のオリジナルバージョン又はバージョンアップグレードのための基準バージョンプログラムにすることができ、第 2 バージョンプログラムは最近のアップグレードバージョンのプログラムにすることができる。

【0088】

二つのバージョンのプログラムをロードした後、アップグレードパッケージプロセッサ 10 は、555 ステップで二つのバージョンプログラムの圧縮符号化が必要であるか否か (C__FLAG=1) を検査する。この時、圧縮符号化が必要でない場合、アップグレードパッケージプロセッサ 10 は、561 ステップで第 1 及び第 2 バージョンプログラムを各々コンフィギュアバージョンにフォーマットした後、563 ステップへ進行して二つのバージョンプログラムの比較を行なう。しかし、555 ステップで、圧縮符号化が必要な場合、アップグレードパッケージプロセッサ 10 は、557 ステップで第 1 圧縮機 160 を実行した後、559 ステップで第 1 及び第 2 バージョンプログラムを第 1 圧縮機 160 によって圧縮符号化した後、563 ステップで圧縮された二つのバージョンプログラムを比較器 110 によって比較する。

【0089】

この時、559 ステップで二つのバージョンプログラムを圧縮する過程は、図 22 のような過程で行なわれる。この時、圧縮フラグが設定された場合には、検証フラグも共に設定されるのが一般的である。この時、検証フラグ (V__FLAG) がセットされると、アップグレードパッケージプロセッサ 10 は、圧縮データの圧縮を解除した後、圧縮解除されたデータを圧縮前のデータと比較して正常に圧縮が行なわれたかを検証するようになる。

【0090】

図 22 を参照すると、アップグレードパッケージプロセッサ 10 は、第 1 及び第 2 バージョンプログラムを各々圧縮した後、601 ステップで検証が必要であるか (V__FLAG=1) を検査する。検証が必要な場合、アップグレードパッケージプロセッサ 10 は、603 ステップで圧縮解除機 165 を実行させ、圧縮された第 1 及び第 2 バージョンプログラムの圧縮を解除した後、605 ステップで圧縮前のバージョンと圧縮解除した後のバージョンプログラムとを比較し、607 ステップで、比較の結果により、圧縮前及び圧縮解除後のバージョンが同じ値を有するかを検査する。この時、圧縮前及び圧縮解除後のバージョンが同一であると、アップグレードパッケージプロセッサ 10 は 607 ステップでこれを確認し、609 ステップで圧縮されたバージョンを比較器 110 に印加すると同時に、正常状態であることを検証する通知を発生する。しかし、607 ステップで圧縮前及び圧縮解除した後のデータが同一でない場合、アップグレードパッケージプロセッサ 10 は、611 ステップでこれをエラー処理した後、アップグレードパッケージ処理前にリターンする。

【0091】

上記のように、圧縮されているか又は圧縮されていない第 1 バージョンプログラム及び第 2 バージョンプログラムが生成されると、アップグレードパッケージプロセッサ 10 は、563 ステップで比較器 110 を実行させた後、565 ステップでインストールデータ発生器 180 を実行させる。この時、インストールデータ発生器 180 は、マップデータを生成することもあり、又は、マップデータを生成しないこともある。これは、M__FLAG によって決定される。従って、アップグレードパッケージプロセッサ 10 は、567 ステップでマップデータを生成するべきか (M__FLAG=1) を確認する。この時、マップデータを生成する場合 (M__FLAG=1)、アップグレードパッケージプロセッサ 10 は、567 ステップでこれを感知し、569 ステップで第 1 バージョンと第 2 バージョンとのプログラムを比較器 110 によって比較する。この時、第 1 及び第 2 バージョンプログラムの比較は、ブロックのサイズで比較することができ、第 1 バージョン及び第 2 バージョンプログラムのブロックデータが同一でない場合には、現在の第 2 バージョン

プログラムのブロックデータを、第1バージョンプログラムの設定された探索区間内のブロックデータと比較して一致するか否かを判断する。すなわち、比較器110は、現在の第2バージョンブロックデータを、第1バージョンの設定された探索区間内のブロックデータと比較して一致するか否かを判定する。

【0092】

この時、第1バージョン及び第2バージョンは、圧縮されたバージョン情報、又は、圧縮されていないバージョンである。第1及び第2バージョンの比較を完了すると、アップグレードパッケージプロセッサ10は、571ステップでマップデータを生成するために、比較データをインストールデータプロセッサ180に伝達すると同時に、577ステップへ進行してマップデータ及びアップグレードデータを生成するために、比較結果値を格納領域に格納する。しかし、マップデータを生成しなくてもよい場合（M_FLAG=0）、アップグレードパッケージプロセッサ10は、567ステップでこれを感じし、575ステップで第1バージョンプログラムと第2バージョンプログラムとを比較器110によって比較する。この時、第1バージョン及び第2バージョンは、上記のように、圧縮されたバージョン、又は圧縮されていないバージョンである。第1及び第2バージョンの比較を完了すると、アップグレードパッケージプロセッサ10は、577ステップでアップグレードデータを生成するために、比較結果値を格納領域に格納する。上記のように、マップデータを生成する場合に、アップグレードパッケージプロセッサ10は、二つのバージョンの比較結果値をインストールデータプロセッサ180に伝達すると同時に、格納領域に格納してマップデータ及びアップグレードデータを生成できるようにし、マップデータを生成しない場合に、アップグレードパッケージプロセッサ10は、二つのバージョンの比較結果値を格納領域に格納してアップグレードデータを生成できるようにする。

【0093】

その後、アップグレードパッケージプロセッサ10は、579ステップでインストールデータプロセッサ180を駆動させてインストールデータを生成し、インストールデータの生成は、図23のような過程で行なわれる。

【0094】

図23を参照すると、インストールデータ生成時、アップグレードパッケージプロセッサ10は、651ステップでインストールデータの生成を活性化した後、653ステップで二つのバージョンのヒストリーデータを検査する。その後、アップグレードパッケージプロセッサ10は、655ステップでヒストリーデータ発生器120を駆動し、657ステップでヒストリーデータを生成する。ここで、ヒストリーデータは、図示のように、ヘッダー（header）、第1バージョンプログラムの情報（first input version info）、第2バージョンプログラムの情報（second input version info）、及びメモリ情報（memory info）等から構成される。第1及び第2バージョンプログラムの情報は、各々バージョン識別情報（SW VER; 0x2AB, 0x2AC）、タイムスタンプ（SW time stamp）、チェックサム情報（SW checksum）等からなる。ヒストリーデータは、アップグレード時に基準となるバージョン（ここでは、第1バージョンでありオリジナルバージョン）と、基準バージョンからアップグレードされるべきデータを提供するアップグレードバージョン（ここでは、第2バージョン）の識別情報とを含む。

【0095】

その後、アップグレードパッケージプロセッサ10は、659ステップでマップデータを生成するか（M_FLAG=1）を確認する。この時、マップデータを生成する場合、アップグレードパッケージプロセッサ10は、603ステップでマップデータ発生器150を駆動した後、665ステップでマップデータを生成する。ここで、マップデータは、アップグレードデータを基準バージョンである第1バージョンに含めてアップグレードする場合、アップグレードデータの位置情報及びアップグレード形態等を指定するマップ情報にすることができる。ここで、マップデータは、上記のように、コピーC、変更M、及び移動S、並びに関連したブロックインデックス情報等にすることができ、マップデータ

は、所定サイズを有するマクロブロックに分割されて設定される。この時、マップデータは、二つのバージョンの比較結果値によって生成されるが、二つのバージョンのブロックデータが同一であると、当該ブロックのマップデータはコピーCに設定し、二つのバージョン情報のブロックデータ値が異なるか、又は、ブロックデータが追加された場合には、当該ブロックのマップデータを変更Mに設定し、二つのバージョンのブロックデータ値は同一であるが、そのブロックのインデックスが異なる場合には、同じ位置までの移動Sに設定する。図23に示すように、マップデータが設定された場合、第2バージョンプログラムのブロック番号は、#0000-#1F00であり、ここで、第1バージョン情報と第2バージョン情報との比較結果、#0000-#0010、#0024-#0220のブロックは同一であり、#0011-#0023、#0221ブロックは、変更されたブロックであり、#0222-#1F00は、ブロック内の位置のみが異なり移動すれば同じ値を有するブロックである。上記のような場合、第2バージョンの#0011-#001F、#0020-#0023ブロックは、第1バージョンの#0011-#001F、#0020-#0023ブロックを代替するという変更が発生した場合であり、第2バージョンの#0221ブロックは、第1バージョンの#0220及び0221の間に挿入される変更ブロックが発生された例である。マップデータ発生器150は、C：B#0000-B#0010、M：B#00011-B#0023、C：B#0024-B#0220、M：B#0221、S：[B#0222-B#1F00][B#0221-1E FF]のマップデータを生成するようになり、この時、アップグレードデータは、変更Mに関するブロックのデータに生成することができる。上記のように、マップデータを生成した後、アップグレードパッケージプロセッサ10は、667ステップでヒストリーデータにマップデータをマージしてインストールデータを生成する。

【0096】

661ステップまで行なってから、アップグレードパッケージプロセッサ10は、M__FLAGの設定有無によってインストールデータを生成するようになるが、この時に生成されるインストールデータは、ヒストリーデータのみから構成されるか、又は、ヒストリーデータ及びマップデータがマージされた形態のデータになる。

【0097】

上記のように、インストールデータを生成した後、アップグレードパッケージプロセッサ10は、581ステップでパッケージ発生器130を実行させ、583ステップでパッケージ発生器130によってアップグレードパッケージを生成する。図24は、アップグレードパッケージを生成する過程を示すフロー図である。

【0098】

図24を参照すると、アップグレードパッケージプロセッサ10は、621ステップで比較結果に基づいてアップグレードデータを生成する。ここで、パッケージ発生器130は、第1バージョンと第2バージョンとの比較結果値が、設定された探索区間で同じブロックデータを見つけられなかった場合、当該ブロックのデータをアップグレードデータに設定することができる。アップグレードデータのマップ位置は、マップデータによって決定される。すなわち、第2バージョンのブロックデータが、第1バージョンの探索区間において、第1バージョンのブロックデータのサイズと異なるか、又は、別のデータである場合、該当するブロックのデータをアップグレードデータに設定し、アップグレードデータは、マップデータにおいて変更Mに対応するブロックとなる。

【0099】

また、パッケージ発生器130は、インストールデータがマップデータを含まない場合、アップグレードデータに第1バージョンと結合するための第2バージョンの各ブロックのブロックインデックス値を含めてアップグレードデータを生成することもできる。このような場合、アップグレードデータは、コピーC、変更M、及びシフトSのコマンドフォーマットに生成される。

【0100】

アップグレードデータは、圧縮してダウンロードするのが好ましい。アップグレードパ

10

20

30

40

50

ッケージプロセッサ10は、623ステップで第2圧縮機140を駆動した後、625ステップでアップグレードデータを圧縮し、またアップグレードデータの圧縮が正常に行なわれたかを検証するために、627ステップ及び629ステップで圧縮されたアップグレードデータの圧縮を解除した後、圧縮前及び圧縮解除後のデータを比較して検証する。この時、検証が正常に行なわれると、アップグレードパッケージプロセッサ10は、631ステップでこれを感知し、633ステップでアップグレードデータにインストールデータを合成してアップグレードパッケージを生成し、635ステップでアップグレードパッケージをアップグレードパッケージサーバー20に伝達する。しかし、631ステップで検証が失敗した場合、アップグレードパッケージプロセッサ10は、637ステップでエラー処理する。

10

【0101】

上記のように、アップグレードパッケージプロセッサ10でアップグレードパッケージが生成されると、アップグレードパッケージは、アップグレードパッケージサーバー20に伝達され、これを必要とする受信装置30にダウンロードされる。この時、アップグレードパッケージは、基準バージョンである第1バージョンと、アップグレードバージョンである第2バージョンとの差を有するアップグレードデータ及びこのようなアップグレードデータをインストールするためのデータから構成される。

【0102】

図25は、本発明の実施形態に係るアップグレードパッケージを伝達されるアップグレードパッケージサーバー20が、アップグレードパッケージを受信装置30にダウンロードするステップを示すフロー図である。ここで、アップグレードパッケージサーバー20は、アップグレードパッケージプロセッサ10と一体型に構成されるか、又は、分離して構成される。

20

【0103】

図25を参照すると、アップグレードパッケージを伝達されると、アップグレードパッケージサーバー20は、711ステップでアップグレードパッケージが存在することを各受信装置30に通知する。この時、アップグレードパッケージサーバー20と受信装置30との通信方法は、受信装置30とアップグレードパッケージサーバー20との連結方法によって決定される。すなわち、受信装置30が携帯用端末機である場合、受信装置30は、空中無線通信(例えば、CDMA, GSM, GSM, GPRS等)方法で通信機能を行なう。また、受信装置30が、コンピュータ又はコンピュータを介して連結された携帯端末機である場合、アップグレードパッケージの通信をインターネットを介して行うことができる。ここで、携帯端末機が、無線通信機能を備える場合、無線インターネット(wibro, wimax, wifi, 無線LAN等)を用いることもできる。その後、受信装置30がこれに対する応答をすると、アップグレードパッケージサーバー20は、713ステップでこれを受信した後、715ステップでダウンロードの許容を通知する。その後、受信装置30がこれに対する応答をすると、アップグレードパッケージサーバー20は、717ステップでこれを受信し、719ステップで管理情報(management info)を該当受信装置30に伝送する。その後、受信装置30がこれに対する応答をすれば、アップグレードパッケージサーバー20は、723ステップでアップグレードパッケージを当該受信装置30にダウンロードする。アップグレードパッケージをダウンロードするアップグレードパッケージサーバー20は、725ステップで、受信装置30でダウンロード完了データを受信するようになり、727ステップで送信終了情報を受信装置30に伝送し、729ステップで応答メッセージ受信時、アップグレードパッケージのダウンロード過程を終了する。

30

40

【0104】

上記のように、アップグレードパッケージサーバー20は、アップグレードパッケージプロセッサ10から伝達するアップグレードパッケージを多数の受信装置30に通知し、通知に応答する受信装置30にアップグレードパッケージをダウンロードする。この時、受信装置30は、アップグレードパッケージサーバー20からダウンロードされるアップ

50

グレードパッケージを、第1メモリ250にインストールする過程（installing）及びアップグレード要求時、第1メモリ250にインストールされたアップグレードデータを用いて第2バージョンプログラムを生成し、これを第2メモリ260に格納して運用する過程（translating）を実行するようになる。

【0105】

図26は、本発明の実施形態に係る受信装置30が、アップグレードパッケージをダウンロードしてインストールする過程、及びインストールされたアップグレードパッケージを用いて第2バージョンプログラムを生成した後、これを用いてシステムを運用する過程等を示すフロー図である。

【0106】

図26を参照すると、受信装置30は、801ステップですでに第1バージョンプログラムを第1メモリ250に格納している状態である。この時、第1バージョンは、システムの製造時、第1メモリ250にインストールでき、又はシステムの製造後、基準となる新バージョンをダウンロードしてこれを基準バージョンのプログラムに設定することもできる。上記のような状態でアップグレードパッケージのダウンロードを通知されると、受信装置30は、図25のような過程を行ないアップグレードパッケージサーバー20からアップグレードパッケージをダウンロードするようになる。受信装置30は、803ステップー807ステップを行ない、アップグレードパッケージサーバー20からダウンロードされるアップグレードパッケージを受信して一時格納する。一時格納されたアップグレードパッケージは、第1メモリ250にインストールするが、この時、インストール動作は、アップグレードパッケージをダウンロードしてからすぐにも実行することもでき、又は、受信装置30の正常な動作モードを実行した後インストールすることもできる。アップグレードパッケージのダウンロード後、インストール命令が発生されないと、受信装置30は、809ステップでこれを感知し、811ステップで受信装置30の正常動作モードに戻る。のような状態で正常モードを実行した後、インストール命令が発生すると、受信装置30は、一時格納中のアップグレードパッケージを第1メモリ250にインストールする動作を行なうようになる。

【0107】

809ステップでインストール命令を感知すると、受信装置30は、813ステップで第1メモリ250にダウンロードされたアップグレードパッケージをインストールする。第1メモリ250は、不揮発性メモリにすることができ、バージョンアップグレード時、基準となる第1バージョンを格納する領域と、複数のアップグレードパッケージを格納する領域とから構成される。すなわち、第1メモリ250には、図10、図11ー図12のように、バージョンを格納する第1及び第2領域が割り当てられる。アップグレードパッケージのダウンロード時、受信装置30は、813ステップで該当するアップグレードパッケージを格納する格納領域にダウンロードされたアップグレードパッケージをインストールする。この時、アップグレードパッケージは、以前のアップグレードパッケージと独立して格納されるため、第1メモリ250には、第2バージョンのアップグレードパッケージが生成された順に格納されており、バージョンのアップグレード履歴を全て格納している状態になる。

【0108】

アップグレードパッケージを第1メモリ250にインストールした後、受信装置30は、システム初期化（system reboot）動作を実行するかを表示する。この時、システム初期化命令が発生されないと、受信装置30は、815ステップでこれを感知し、817ステップで正常動作モードを実行する。このような場合、動作モードは、現在ダウンロードされたアップグレードパッケージによるバージョンアップグレードが行われていない状態であるため、旧バージョンで動作モードを実行することになる。

【0109】

しかし、815ステップでシステム初期化命令が発生されると、受信装置30は、821ステップでシステム初期化動作を実行し、初期化動作を行なった後、受信されたアップ

10

20

30

40

50

グレードパッケージから第2バージョンプログラムを生成するために、823ステップでトランスレータ240を駆動する。この時、トランスレータ240は、第1メモリ250にインストールされたアップグレードパッケージを第1バージョンにマージし第2バージョンのプログラムを生成して第2メモリ260に格納し、その後、受信装置30は、第2メモリ260に格納された第2バージョンのプログラムに基づいて動作モードを制御するようにする。受信装置30は、825ステップでアップグレードパッケージのインストールデータを検査してアップグレードパッケージの状態を確認する。この時、アップグレードパッケージに異常があれば、受信装置30は、833ステップでアップグレードバージョンの前のバージョンを第2メモリ260に格納してシステムの動作を制御する。しかし、アップグレードパッケージが正常であれば、受信装置30は、825ステップでこれを
10
感知し、827ステップで第1メモリ250にインストールされた当該アップグレードパッケージをロードし、829ステップで第1メモリ250に格納している第1バージョンに、ロードしたアップグレードパッケージを組み合わせて第2バージョンのプログラムを生成し、これを第2メモリ260に格納する。その後、受信装置30は、831ステップで第2メモリ260に格納されたアップグレードバージョンのプログラムを用いてシステムの動作を制御する。

【0110】

上記のように、受信装置30は、アップグレードパッケージをダウンロードすると、インストール過程及びアップグレード過程を実行する。図27は、本発明の実施形態に係る受信装置30がダウンロードしたアップグレードパッケージを第1メモリ250にインストールする過程を示すフロー図であり、図28は、本発明の実施形態に係る受信装置30において第1メモリ250にインストールされたアップグレードパッケージを第1バージョンのプログラムにアップグレードして第2メモリ260に格納した後、システムを運用する過程を示すフロー図である。
20

【0111】

図27を参照すると、ダウンロード必要時、受信装置30は、841ステップでダウンローダー(down loader)を実行させ、843ステップでネットワークを介してアップグレードパッケージサーバー20から伝送されるアップグレードパッケージをダウンロードする。ここで、アップグレードパッケージのダウンロードは受信装置30によって変わる。すなわち、受信装置30が移動通信端末機であると、受信装置30は、設定される無線リンク(例えば、CDMA, UMTS, GSM, GPRS等)を介してアップグレードパッケージをダウンロードすることができ、インターネットを介して連結される端末機又は無線インターネット(wibro, wifi, wimax, 無線LAN等)を介してアップグレードパッケージをダウンロードすることができる。また、USBのような有線通信リンク、又は、ブルートゥース等のような近距離通信リンクを介してシステムに連結される端末機の場合には、該当有線通信方法、又は、近距離無線通信方法によってアップグレードパッケージをダウンロードすることができる。この時、ダウンロードは、アップグレードパッケージ全体がダウンロードされるまで続き、ダウンロード中にエラーが発生すると、受信装置30は、845ステップでこれを感知し、847ステップ及び849ステップを行ないアップグレードパッケージの再ダウンロードを行なう。
30
40

【0112】

この時、845ステップで正常にダウンロードを完了した場合、受信装置30は、851ステップでインストーラー(installer)230をロードし、アップグレードパッケージを第1メモリ250にインストールする準備をする。アップグレードパッケージを第1メモリ250にインストールする過程を説明すると、受信装置30は、853ステップー857ステップでアップグレードパッケージのヒストリーデータを抽出し、抽出されたヒストリーデータを確認した後、第1メモリ250の該当アップグレードパッケージ格納領域内のヒストリーテーブルに格納する。

【0113】

その後、受信装置30は、アップグレードパッケージ内にマップデータが含まれている
50

かを検査する。アップグレードパッケージ内にマップデータが含まれている場合、受信装置 30 は、859 ステップでこれを感じ知し、875 ステップ及び 877 ステップでアップグレードパッケージ内からマップデータを抽出して第 1 メモリ 250 の当該アップグレードパッケージ格納領域内にマップデータを格納し、879 ステップでアップグレードパッケージ格納領域内のアップグレードデータ格納領域にアップグレードデータを格納する。881 ステップになると、受信装置 30 は、受信されたアップグレードパッケージからヒストリーデータ、マップデータ、及びアップグレードデータを各々抽出した後、第 1 メモリ 250 の該当するアップグレードパッケージ格納領域内にヒストリーデータ、マップデータ、及びアップグレードデータをインストールして格納した状態になる。

【0114】

また、アップグレードパッケージ内にマップデータが含まれていない場合、受信装置 30 は、859 ステップでこれを感じ知し、861 ステップ及び 863 ステップで圧縮解除機 270 を駆動してアップグレードパッケージからアップグレードデータを抽出して圧縮を解除する。その後、受信装置 30 は、865 ステップ〜869 ステップを行ない圧縮解除されたアップグレードデータをパーシングした後、これを第 1 メモリ 250 に格納している第 1 バージョンプログラムと比較して差データを求め、これに基づいてアップグレードデータを第 1 バージョンプログラムにマップするためのマップデータを生成する。その後、受信装置 30 は、871 ステップ及び 873 ステップで第 1 メモリ 250 に割り当てられたアップグレードパッケージの格納領域に生成されたマップデータ及びアップグレードデータを各々格納する。

【0115】

図 27 に示されたように、受信装置 30 は、アップグレードパッケージのダウンロードが通知されると、アップグレードパッケージをダウンロードし、ダウンロードされたアップグレードパッケージを分析して第 1 メモリ 250 の対応する領域に各々ヒストリーデータ、マップデータ、及びアップグレードデータをインストールする。ここで、マップデータは、アップグレードパッケージに含まれることもあり、含まれないこともある。アップグレードパッケージにマップデータが含まれていない場合、受信装置 30 は、第 1 メモリ 250 に格納し、基準バージョン情報である第 1 バージョン情報にダウンロードされたアップグレードデータを比較してマップデータを生成し、生成されたデータをアップグレードパッケージ格納領域にインストールして格納する。

【0116】

図 28 は、受信装置 30 において、図 27 のような過程でダウンロードするアップグレードパッケージを第 2 メモリ上でアップグレードする過程を示すフロー図である。

【0117】

図 28 を参照すると、アップグレードパッケージを用いて受信装置 30 のバージョンをアップグレードする過程は、システム電源をオンにするか又はユーザーのアップグレード要求によって実行される。すなわち、本発明の実施形態に係る受信装置 30 は、システム初期化時、ハードウェア等の初期化を行なった後、第 1 メモリ 250 に格納された第 1 バージョンプログラム及び最も最近アップグレードされたアップグレードパッケージ(又は、ユーザーによって設定されたアップグレードパッケージ)を組み合わせてアップグレードされた第 2 バージョンプログラムを第 2 メモリ 260 に格納し、これを用いてシステムの動作を制御する。ここで、第 2 メモリ 260 に格納される情報は、プログラムとして不揮発性メモリにインストールされる全ての情報を含む。

【0118】

システム初期化動作が実行されると、受信装置 30 は、881 ステップ〜883 ステップを行いシステム初期化を実行した後、第 1 メモリ 250 から第 1 バージョンプログラム及びアップグレードパッケージをロードするためのローダーを実行する。その後、受信装置 30 は、884 ステップで第 1 メモリ 250 内のアップグレードパッケージ格納領域を分析し、アップグレードパッケージが一つ以上格納されているかを検査する。この時、アップグレードパッケージが格納されていない場合、受信装置 30 は、884 ステップでこ

れを感知し、８８５ステップでアップグレード過程を実行するためのトランスレータ２４０をロードし、８８６ステップでダウンロードしたアップグレードパッケージが受信側に適しているか否かを確認してウイルスチェック等のセキュリティー検査を行なう。その後、受信装置３０は、８８７ステップで第１メモリ２５０に格納されている第１バージョンプログラムが圧縮された状態であるかを検査し、圧縮された状態であると、８８８ステップで圧縮解除機２７０を駆動して第１バージョンの圧縮を解除した後、８８９ステップで第１バージョンを第２メモリに格納し、８９０ステップで第２メモリ２６０に格納された第１バージョンを用いてシステムを運用する。しかし、８８７ステップで第１バージョンが圧縮されていないと判定されれば、受信装置３０は、８８９ステップ及び８９０ステップを行い第１バージョンを第２メモリ２６０に格納し、これを用いてシステムを運用する。

10

【０１１９】

しかし、８８４ステップで一つ以上のアップグレードパッケージが第１メモリ２５０に格納されていると、受信装置３０は、８９１ステップでトランスレータ２４０をロードした後、８９２ステップで最も最近アップグレードされたアップグレードパッケージをロードする。この時、第１メモリ２５０にインストールされたアップグレードパッケージは、ヒストリーデータ、マップデータ、及びアップグレードデータ、又は、ヒストリーデータ及びマップデータ、又は、ヒストリーデータ及びアップグレードデータ等から構成される。その後、受信装置３０は、８９３ステップで圧縮解除機２７０を駆動してロードされたアップグレードパッケージ(ここで、アップグレードデータのみ圧縮されていることもある)の圧縮を解除し、８９４ステップでセキュリティー検査を行なう。その後、受信装置３０は、８９５ステップで第１メモリ２５０に格納されている第１バージョンが圧縮された状態であるかを検査し、圧縮された状態であると、８９６ステップで圧縮解除機を駆動して第１バージョンの圧縮を解除した後、８９７ステップで第１バージョンにアップグレードパッケージを結合してアップグレードされた第２バージョンプログラムを第２メモリに格納し、８９０ステップで第２メモリ２６０にアップグレードされた第２バージョンプログラムを用いてシステムを運用する。この時、８９７ステップでアップグレードパッケージのアップグレードデータを第１バージョンに結合してアップグレードする時にマップデータを用いてアップグレードデータを第１バージョンにマップする。また、８９５ステップで第１バージョンが圧縮されていないと判定されれば、受信装置３０は、８９７ステップ及び８９０ステップを行い第１バージョンにアップグレードパッケージを組み合わせてアップグレードバージョンのプログラムを生成した後、これを第２メモリに格納し、これを用いてシステムを運用する。

20

30

【０１２０】

図２９－図３２は、図２８のように、第１メモリ２５０にインストールされたアップグレードパッケージを基準バージョンプログラムに結合してアップグレードされた第２バージョンプログラムを生成した後、第２メモリ２６０に格納する過程を詳細に示すフロー図である。

【０１２１】

図２９－図３２を参照すると、受信装置３０は、９０１ステップー９０５ステップを行いシステム初期化を実行した後、バージョンをアップグレードするためにローダーを実行させ、９０７ステップ及び９０９ステップを行いアップグレードパッケージが有効であるかを決定した後、第１メモリ２５０に少なくとも一つのアップグレードパッケージが格納されているかを検査する。この時、第１メモリ２５０に有効なアップグレードパッケージが一つも格納されていないければ、受信装置３０は、９１１ステップへ進行してトランスレータ２４０をロードした後、図３０の９２１ステップー９２９ステップを行い第１メモリ２５０に格納している第１バージョンをトランスレートして第２メモリ２６０に格納し、これを用いてシステムを運用する。この時、第１バージョンＶ１が圧縮された状態であると、受信装置３０は、９２２ステップでこれを感知し、９２３ステップー９２７ステップを行い第１バージョンＶ１の最後のデータまでトランスレートして第２メモリ２６０に格

40

50

納する。第1バージョンV1が圧縮されたデータではない場合、受信装置30は、926ステップ及び927ステップを繰り返し行い、第1バージョンV1の最後のデータまでトランスレートして第2メモリ260に格納する。上記のように、第1バージョンV1を格納した後、受信装置30は、928ステップで第2メモリ260に格納されたバージョン(ここでは、システムの全体ソフトウェア情報にすることができる)を検証した後、929ステップでこれを用いてシステムの全体的動作を実行させる。

【0122】

しかし、909ステップで少なくとも一つのアップグレードパッケージが第1メモリ250に格納されていることを感知すると、受信装置30は、913ステップでアップグレードパッケージがヒストリーデータを備えているかを検査する。この時、ヒストリーデータを含んでいるアップグレードパッケージであると、受信装置30は、915ステップでヒストリーデータ内の失敗フラグ(f a i l f l a g)を検査する。ここで、失敗フラグは、当該アップグレードパッケージのロード失敗が発生されたことがあることを表示するフラグであり、失敗フラグがセット(f a i l f l a g = t r u e)されたということは、当該アップグレードパッケージに異常があったことを表す。受信装置30は、917ステップでヒストリーデータの失敗フラグがセットされたかを検査した後、失敗フラグがセットされていない状態であると、図31の931ステップー946ステップを行いシステムアップグレード動作を実行し、失敗フラグがセットされた状態であると、図32の951ステップー969ステップを行いシステムアップグレード動作を実行する。

【0123】

まず、アップグレードパッケージのヒストリーデータが正常である場合の動作を図31を参照して説明すると、受信装置30は、931ステップー937ステップで第1メモリ250に格納している最も最近のアップグレードパッケージのヒストリーデータを探索して失敗フラグを確認し、正常であると、マップデータ及びアップグレードデータをロードし、トランスレータをロードした後、セキュリティー検査を行い、アップグレードデータが圧縮された場合には、アップグレードデータの圧縮を解除する。その後、第1バージョンV1が圧縮された場合、受信装置30は、939ステップ及び941ステップを行い第1バージョンV1プログラムの圧縮を解除し、942ステップで第1バージョンV1にアップグレードパッケージのアップグレードデータをマップしてバージョン情報をアップグレードし、アップグレードされるバージョン情報を第2メモリ260に格納する。そして、943ステップ及び944ステップを繰り返さないアップグレードデータの最後のデータまでアップグレードする動作を実行する。938ステップで第1バージョンV1が圧縮されていない場合、受信装置30は、938ステップでこれを感知し、940ステップ、943ステップ、及び944ステップを行ない第1バージョンV1にアップグレードパッケージのアップグレードデータをマップしてアップグレードバージョンV2のプログラムを生成した後、これを第2メモリ260に格納する。上記のように、第2メモリ260にアップグレードバージョンのプログラムを格納した後、受信装置30は、945ステップでこれを検証した後、946ステップで第2メモリ260に格納されたアップグレードバージョン情報を用いてシステムを駆動する。

【0124】

また、917ステップでアップグレードパッケージに失敗フラグがセットされた場合、受信装置30は、951ステップで全てのアップグレードパッケージの失敗フラグがセットされたかを検査する。この時、全てのアップグレードパッケージの失敗フラグがセットされた状態であると、受信装置30は、952ステップでトランスレータをロードした後、921ステップ以後の動作過程を行ない第1バージョンを第2メモリ260に格納し、これを用いてシステムの駆動を制御する。すなわち、インストールされた全てのアップグレードパッケージに異常が発生した場合、受信装置30は、第1メモリ250に格納している第1バージョンをロードして第2メモリ260に格納し、これを用いてシステムの動作を制御する。この時、第1バージョンは、システムの製造時、第1メモリ250にロードされるプログラムにすることができる。

【0125】

しかし、951ステップで全てのアップグレードパッケージの失敗フラグがセットされた状態でない場合、受信装置30は、953ステップ及び954ステップで失敗フラグがセットされたアップグレードパッケージを検査した後、使用可能なアップグレードパッケージのリストを作成してユーザーに表示する。その後、ユーザーがリストの中から必要とするアップグレードパッケージを選択すると、受信装置30は、955ステップでこれを感知し、956ステップー960ステップを行い選択されたアップグレードパッケージのマッピングデータ及びアップグレードデータをロードし、トランスレータをロードしてセキュリティ検査を行なった後、アップグレードデータが圧縮された状態であれば、アップグレードデータの圧縮を解除する。その後、961ステップー969ステップを行い第1メモリ250から第1バージョンをロードした後、アップグレードデータをマッピングして第2バージョンを生成し第2メモリ260に格納した後、これを用いてシステムの駆動を制御する。961ステップー969ステップの動作は、938ステップー946ステップと同じ方法で行なうことができる。

10

【0126】

上記のように、本発明の実施形態において、アップグレードパッケージを生成する送信側が、基準バージョンとアップグレードバージョンとの差を比較し、比較結果によるアップグレードパッケージを生成して伝送し、アップグレードパッケージをダウンロードする受信側は、ダウンロードされるアップグレードパッケージを第1メモリ250にインストールし、その後、第1メモリ250のアップグレードパッケージを第1バージョンのプログラムにマッピングして第2バージョンのプログラムを生成し第2メモリ260に格納した後、これを用いてシステムの駆動を制御する。

20

【0127】

まず、アップグレードパッケージプロセッサ10において、アップグレードパッケージを生成する方法(upgrade module generation; delta generation)を説明すると、まず動画像信号を符号化する方法と同じ方法で圧縮されていない初期バージョン(第1バージョンV1)と、圧縮されていない現在バージョン(第2バージョンV2)との差を符号化するためにコード第1圧縮機(code compressor)160を使用する。この時、第1圧縮機160は、初期バージョンV1と現在バージョンV2との差を符号化して圧縮する(domain: V1(previous frame), range: V2(current frame))。

30

【0128】

アップグレードパッケージ生成モジュールの特性は、高圧縮率(high compression ratio: 平均99%を越える)、迅速なアップグレード時間(very fast updating time: file loading時間より短い)、及び生成時間(generation time)も短いという特性を有する。

【0129】

本発明の実施形態に係るアップグレードパッケージ生成の特徴は、次のとおりである。

【0130】

まず、第1バージョン及び第2バージョンを受信し、二つのバージョンを比較してその差を含む比較データを生成する。ここで、第1バージョンは、基準バージョンであり、システムの製造時、メモリにインストールされるプログラム、又はその後決定された基準プログラムにすることができる。第2バージョンは、第1バージョンのアップグレードバージョンであり、システムの製造後、システムにダウンロードされるバージョンを意味する。ここで、第2バージョンは、複数個発生することもでき、最も最近生成される第2バージョンが最も最近アップグレードされたバージョンとなる。

40

【0131】

二つのバージョンを比較する時、バージョンを各々圧縮(compress_1)した後比較することもでき、又は、圧縮していないバージョンを圧縮していない状態で比較することもできる。バージョンを圧縮して比較する場合、バージョンが正常に圧縮されたか

50

を検証するために、圧縮バージョンの圧縮を解除した後、圧縮解除されたバージョンを各々圧縮前のバージョンと比較して圧縮が正常に行なわれたかを検証することができる。

【0132】

そして、二つの比較データを用いてインストールデータを生成する。

【0133】

インストールデータは、第1バージョンにアップグレードデータをマップする時に使用するためのインストールデータを意味する。インストールデータは、少なくともヒストリーデータを含まなければならない。ヒストリーデータは、アップグレードパッケージのバージョン識別情報、結合されるバージョン識別情報等を含んでおり、ロードの失敗状況を表示するフラグを備えることができる。また、インストールデータは、ヒストリーデータ以外にマップデータを含むことができる。マップデータは、アップグレードデータを第1バージョンにマップするためのデータであり、コピー、変更、及び移動等のコマンド及び第1バージョンとブロックマップのためのインデックス情報に区分される。ここで、インストールデータがマップデータを含まない場合には、受信側でマップデータを生成することができる。

10

【0134】

次に、比較データを用いてアップグレードデータを生成し、アップグレードデータにインストールデータを合成してアップグレードパッケージを生成する。

【0135】

アップグレードデータは、第1バージョンと第2バージョンとをブロック単位で比較し、比較結果、ブロックデータが互いに異なるか又は挿入されたブロックデータである場合、該当するブロックデータをアップグレードデータに設定する。この時、アップグレードデータに対応するマップデータは、変更(modify)Mデータにすることができる。この時、アップグレードデータは、アップグレードデータにインストールデータを合成してアップグレードパッケージに生成する。この時、インストールデータには、マップデータが含まれないこともある。このような場合、受信側に生成してインストールさせる。

20

【0136】

また、アップグレードデータ又はアップグレードパッケージは、圧縮してダウンロードすることができる。このような場合、アップグレードデータ(又は、アップグレードパッケージ)を圧縮し、圧縮されたアップグレードデータ(又は、アップグレードパッケージ)の圧縮を解除し、圧縮前のアップグレードデータ(又は、アップグレードパッケージ)と圧縮解除したアップグレードデータ(又は、アップグレードパッケージ)とを比較検討して、圧縮が正常に行なわれたかを検証する。

30

【0137】

上記のように生成されたアップグレードパッケージは、アップグレードパッケージサーバー20に伝達され、アップグレードパッケージサーバーは、伝達されたアップグレードパッケージを受信装置に通知し、通知に応答する受信装置にアップグレードパッケージをダウンロードする。

【0138】

本発明の実施形態に係る受信装置の特徴は、次のとおりである。

40

【0139】

第1に、第1メモリ250に第1バージョンのプログラムを格納し、その後にダウンロードされる少なくとも一つのアップグレードパッケージを格納する。ここで、第1バージョンは、オリジナルバージョンのプログラム又は設定された基準プログラムを意味する。アップグレードパッケージは、アップグレードデータ及びマップデータからなり、アップグレードデータは、旧バージョンパッケージと現バージョンパッケージとの差データ、又は、第2バージョンV2のデータであり、マップデータは、コマンド(ここでは、copy, shift, modify)及びコマンドが実行されるアドレス(ブロックインデックス)を含む。アップグレードパッケージは、第1メモリ250に多数個(本発明の実施形態では、最大6個)を格納することができ、超過する場合には、オーバーライト(over

50

w r i t e) する。マップ (m a p) データは、サーバーからアップグレードパッケージに含めて伝送することができ、携帯端末で生成する方法を使用することもできる。

【0140】

第2に、システム起動時、第1メモリ250の第1バージョンのパッケージ及びアップグレードパッケージを第2メモリ260にロードしてプログラムをアップグレードすることができる。

【0141】

ここで、第1メモリ250は、不揮発性メモリを、第2メモリ260は、揮発性メモリを使用することができる。第1バージョンのパッケージを第2メモリ260にコピーするか、第1バージョンのパッケージをアップグレードパッケージにより変更することにより変更パッケージが生成される。後者の場合、まず、第1バージョンのパッケージを第2メモリ260にコピーし、前記アップグレードパッケージによって前記第1バージョンのパッケージをアップグレードして変更パッケージを生成する。この時、変更方法は、上記のように、c o p y, s h i f t, m o d i f y等に行うことができる。変更の全般的動作は、ローダー (l o a d e r) によって管理され、ローダーは、再起動 (r e - b o o t) が発生される時、アセンブラ及び圧縮解除機の駆動、U I 提供等の機能を行なう。ローダーに含まれたマップ管理者 (m a p m a n a g e r) は、ヒストリー管理及びマッププロセッサ (m a p g e n e r a t o r) を実行するかを決定する。この時、マップデータがあれば、マッププロセッサを実行しない。

【0142】

第3に、第2メモリ260の変更されたパッケージを用いて携帯端末機の動作を行う。

【0143】

受信装置のユーザーがアップグレードパッケージを選択し、選択されたアップグレードパッケージによってシステムのバージョンをアップグレードすることができる。すなわち、受信装置は、アップグレード可能なアップグレードパッケージのリストを表示し、システムユーザーによって選択されたアップグレードパッケージを第1バージョンにアップグレードしてシステムを運用することができる。また、上記のように、アップグレードパッケージダウンロード時、アップグレードパッケージを各々割り当てられた特定格納領域に格納して格納領域を設定された数だけ備えるため、特定アップグレードパッケージに異常が発生しても、他のバージョンのアップグレードパッケージを用いてアップグレードバージョンのプログラムを生成することができる。

【産業上の利用可能性】

【0144】

前述のように、システムのアップグレードバージョンを生成する時、基準バージョンとアップグレードバージョンとの差によるアップグレードパッケージを生成することができ、これによって、アップグレードバージョンのプログラムを迅速に生成することができる。また、前記アップグレードバージョンをダウンロードする時、受信装置は、第1メモリに前記基準バージョンのプログラム及びアップグレードバージョンのアップグレードパッケージを独立にインストールして保管し、前記保管された基準バージョン及びアップグレードバージョンを第2メモリでマージしてアップグレードされたプログラムを生成することによって、安定的にシステムのプログラムをアップグレードできるという利点がある。また、多数のアップグレードパッケージを保管していることによって、システムのアップグレードヒストリーを保持することができ、ユーザーは、必要時に必要とするバージョンのアップグレードプログラムにシステムプログラムを変更できるという利点がある。また、前記第1バージョンプログラム自体のアップグレードは、第1メモリで実行されないため、アップグレード失敗時、コントロールする制御機能であるフォールトトレラント制御 (f a u l t t o l e r a n t c o n t r o l) がなくても、フォールトトレラント制御 (f a u l t t o l e r a n t c o n t r o l) 機能を有する。すなわち、本発明は、アップグレードが失敗しても、第1バージョンプログラムが第1メモリに有効であり、アップグレードに失敗してもいつでも再び実行することができる。アップグレード過程を実

行する途中にどんなフォールトが発生しても、再び実行できるという利点がある。

【図面の簡単な説明】

【0145】

【図1】本発明の実施形態に係るプログラムアップグレードシステムの構成を説明するための図面。

【図2】図1のプログラムアップグレードシステムのアップグレードパッケージプロセッサの動作を示す図面。

【図3】図2のアップグレードパッケージプロセッサ10のアップグレードパッケージを生成するための第1バージョンと第2バージョンとの関係を説明するための図面。

【図4】本発明の実施形態に係るプログラムアップグレードシステムにおけるアップグレードパッケージプロセッサの構成例を示す図面。

【図5】本発明の他の実施形態に係るプログラムアップグレードシステムのアップグレードパッケージプロセッサの構成を示す図面。

【図6】図4のようなアップグレードパッケージプロセッサで発生されたアップグレードパッケージのデータフォーマットを示す図面。

【図7】図5のようなアップグレードプロセッサで発生されたアップグレードパッケージのデータフォーマットを示す図面。

【図8】本発明の他の実施形態に係るプログラムアップグレードシステムのアップグレードパッケージプロセッサの構成を示す図面。

【図9】本発明の実施形態に係るプログラムアップグレードシステムの受信装置の構成を示す図面。

【図10】図9の受信装置の第1メモリの構成を示す図面。

【図11】図10の第1メモリの第2格納領域の構造を示す図面。

【図12】図11の第2格納領域に格納されたアップグレードパッケージのヒストリーデータフォーマットを示す図面。

【図13】本発明の実施形態に係るプログラムアップグレードシステムのアップグレード動作を示す図面。

【図14】本発明の他の実施形態に係るプログラムアップグレードシステムのアップグレード動作を示す図面。

【図15】本発明の他の実施形態に係るプログラムアップグレードシステムのアップグレード動作を示す図面。

【図16】本発明の実施形態に係るプログラムアップグレードシステムの受信装置のアップグレード動作を示す図面。

【図17】本発明の実施形態に係るプログラムアップグレード方法を示すフロー図。

【図18】本発明の実施形態に係るプログラムアップグレード方法のアップグレードパッケージの生成過程を示すフロー図。

【図19】本発明の実施形態に係るプログラムアップグレード方法のアップグレードパッケージの生成過程を示すフロー図。

【図20】本発明の実施形態に係るプログラムアップグレード方法のアップグレードパッケージの生成過程を示すフロー図。

【図21】本発明の実施形態に係るプログラムアップグレード方法のアップグレードパッケージ生成過程を示すフロー図。

【図22】図21における第1及び第2バージョン情報を圧縮及び検証する過程を示すフロー図。

【図23】図21におけるインストールデータを生成する過程を示すフロー図。

【図24】図21におけるアップグレードパッケージを生成する過程を示すフロー図。

【図25】本発明の実施形態に係るアップグレードパッケージを伝達されるアップグレードパッケージサーバーが、アップグレードパッケージを受信装置にダウンロードする過程を示すフロー図。

【図26】本発明の実施形態に係る受信装置が、アップグレードパッケージをダウンロー

10

20

30

40

50

ドしてインストールする過程、及びインストールされたアップグレードパッケージをアップグレードして運用する過程等を示すフロー図。

【図 27】本発明の実施形態に係る受信装置からダウンロードされるアップグレードパッケージを第 1 メモリにインストールする過程を示すフロー図。

【図 28】本発明の実施形態に係る受信装置における第 1 メモリにインストールされたアップグレードパッケージを、第 1 バージョンプログラムにアップグレードして第 2 メモリに格納した後、システムを運用する過程を示すフロー図。

【図 29】図 24 のように第 1 メモリにインストールされたアップグレードパッケージを基準バージョン情報にアップグレードして第 2 メモリに格納する過程を詳細に示すフロー図。

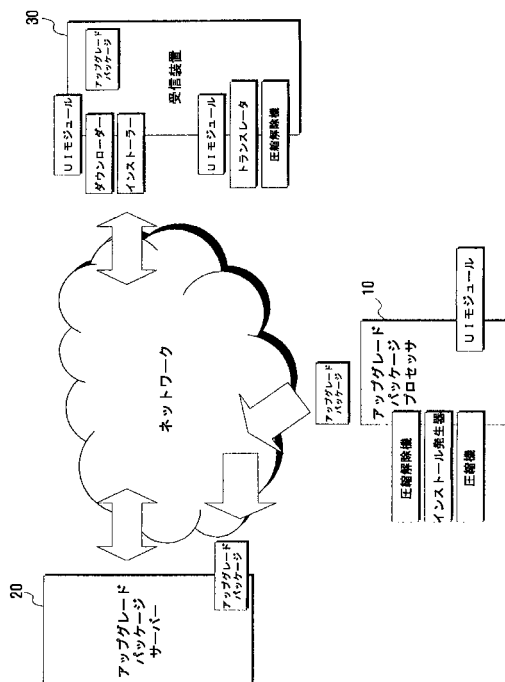
10

【図 30】図 24 のように第 1 メモリにインストールされたアップグレードパッケージを基準バージョン情報にアップグレードして第 2 メモリに格納する過程を詳細に示すフロー図。

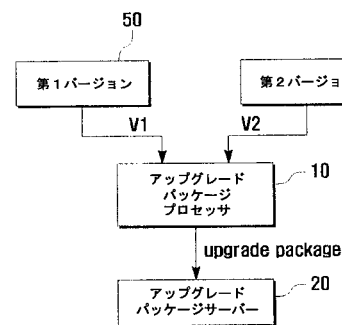
【図 31】図 24 のように第 1 メモリにインストールされたアップグレードパッケージを基準バージョン情報にアップグレードして第 2 メモリに格納する過程を詳細に示すフロー図。

【図 32】図 24 のように第 1 メモリにインストールされたアップグレードパッケージを基準バージョン情報にアップグレードして第 2 メモリに格納する過程を詳細に示すフロー図。

【図 1】



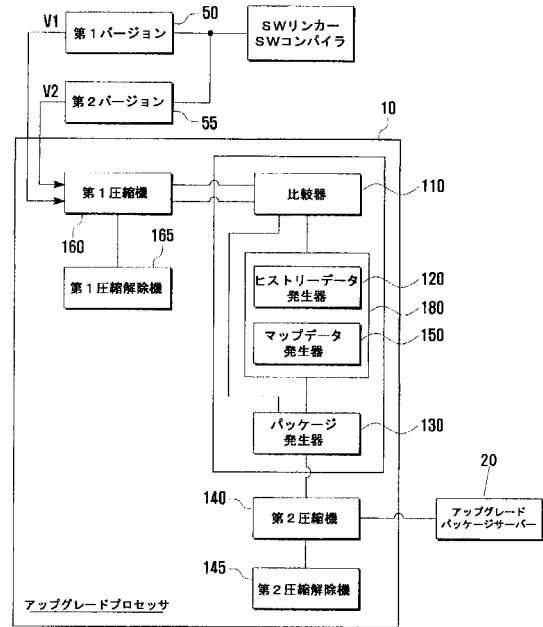
【図 2】



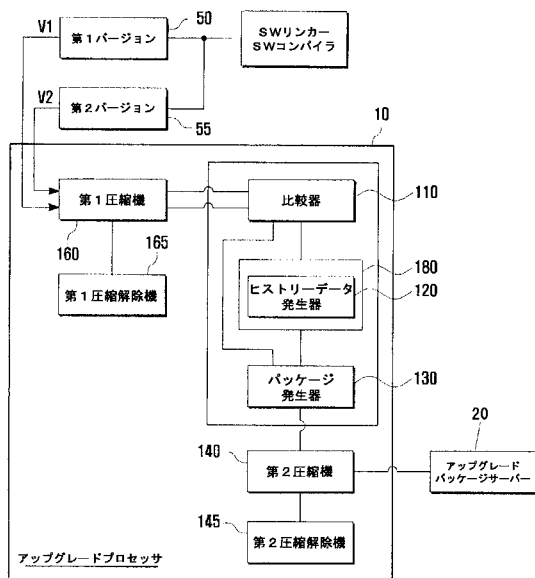
【図 3】



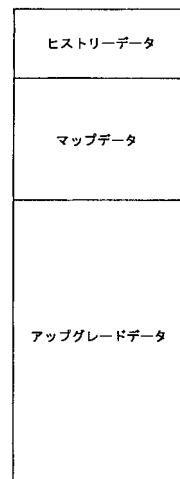
【図 4】



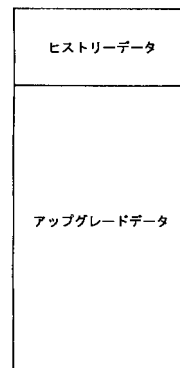
【図 5】



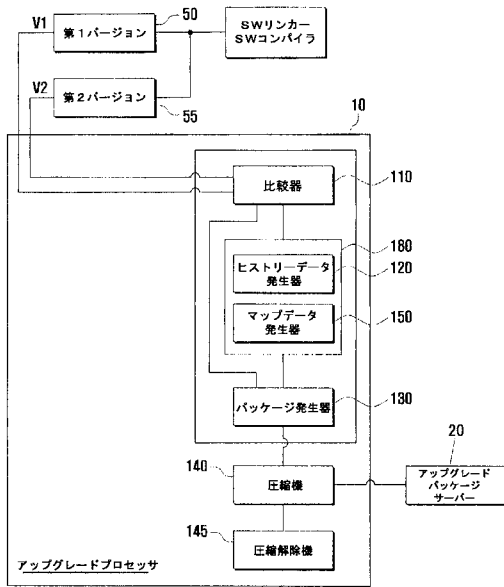
【図 6】



【図 7】



【図 8】



【図 9】

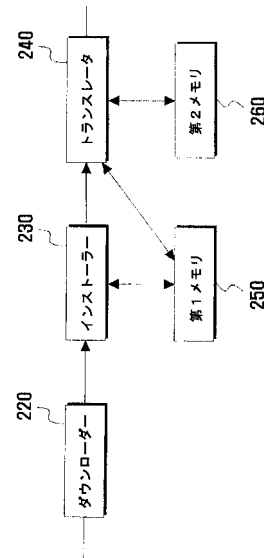
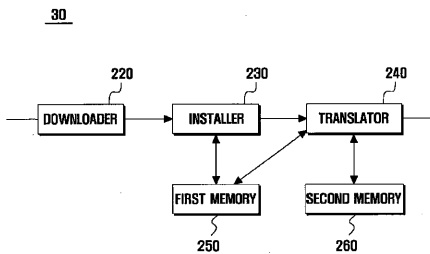
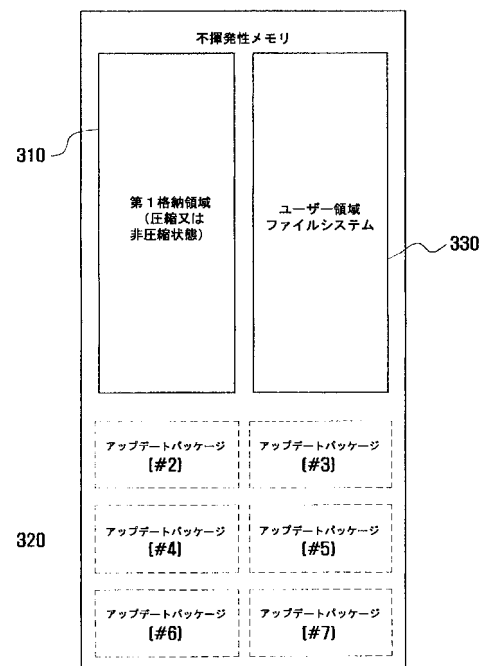


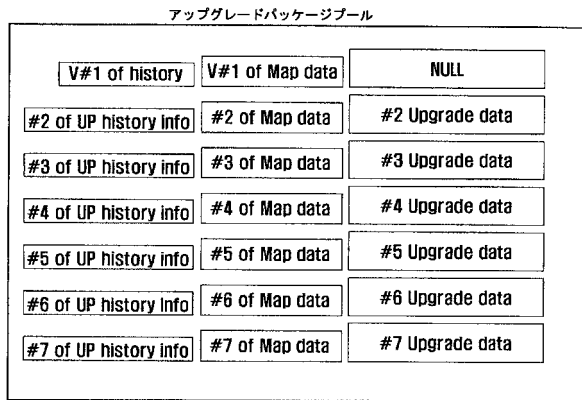
FIG. 9



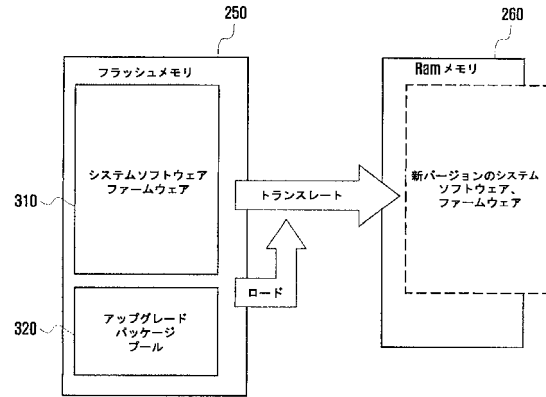
【図 10】



【図 1 1】



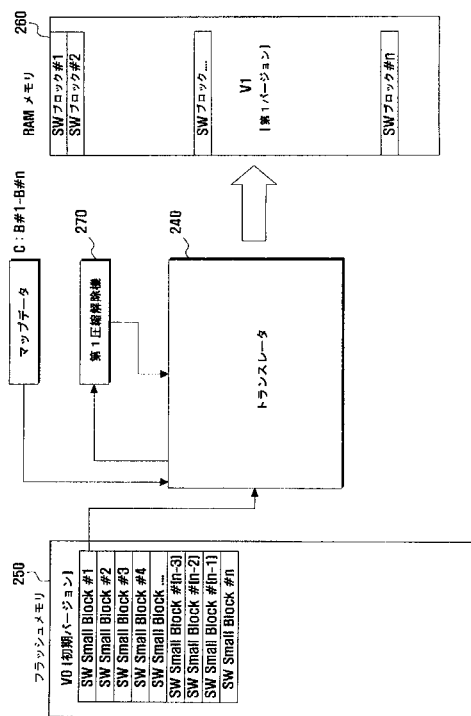
【図 1 3】



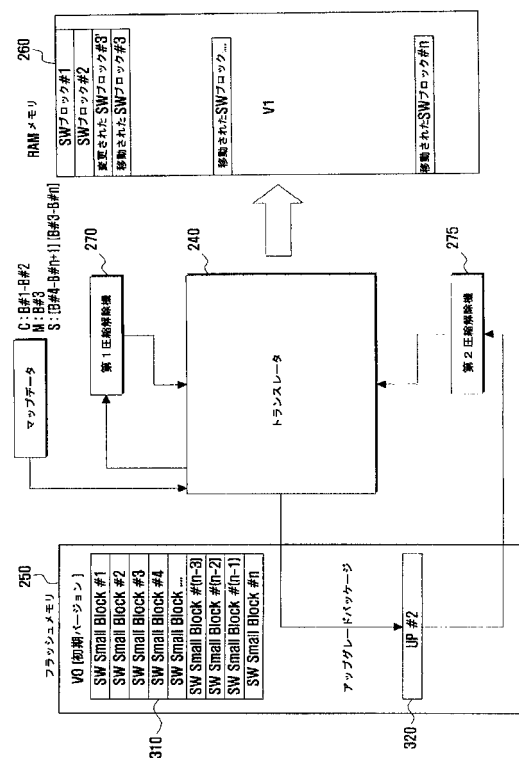
【図 1 2】

履歴構造
バージョン
サイズ
結合フラグ
失敗フラグ

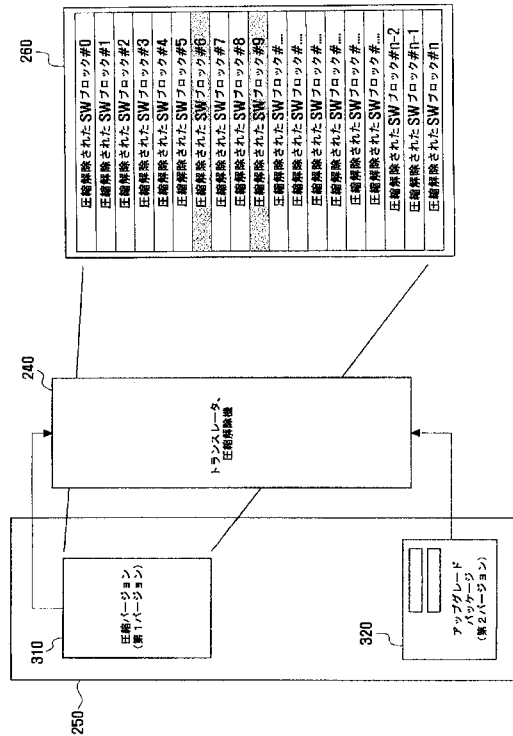
【図 1 4】



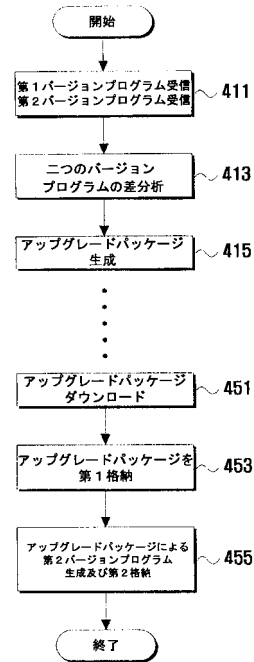
【図 1 5】



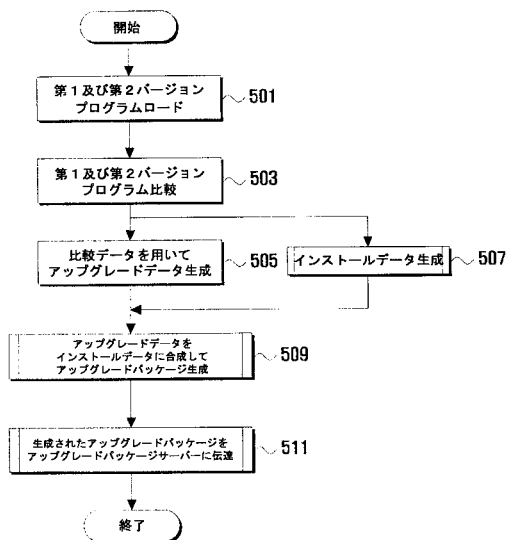
【図 16】



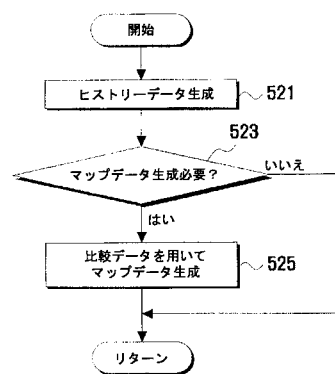
【図 17】



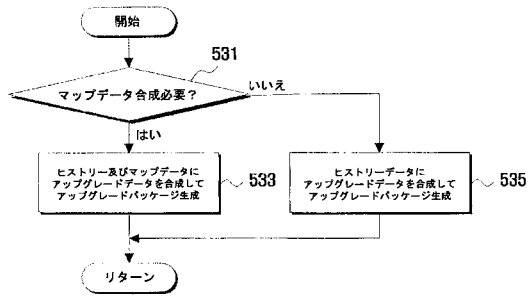
【図 18】



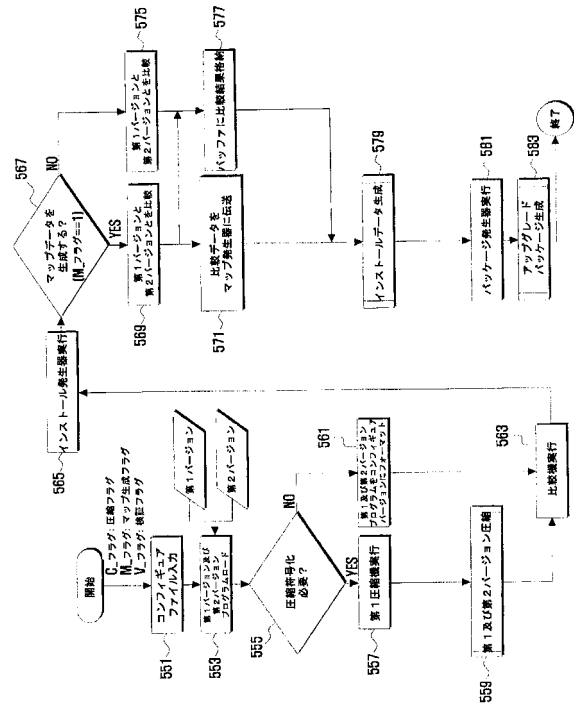
【図 19】



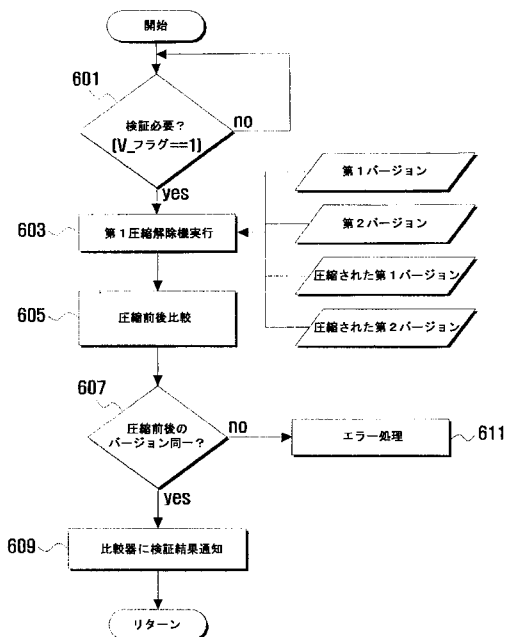
【図 20】



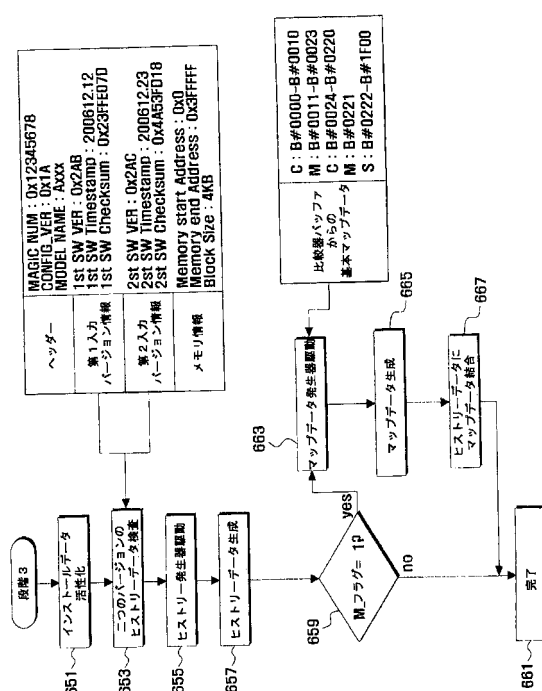
【図 21】



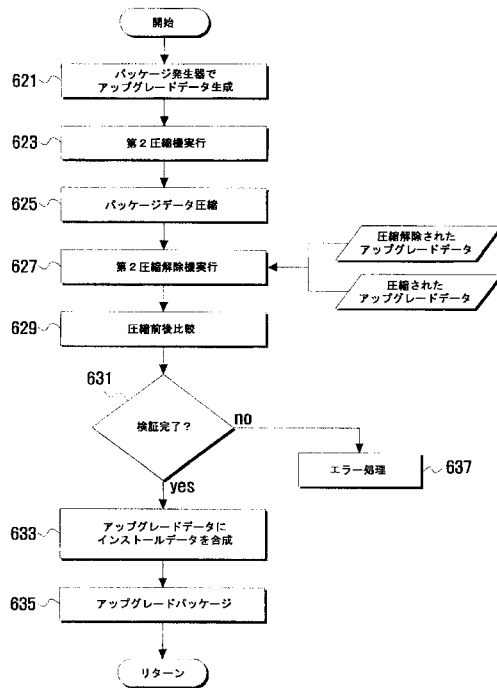
【図 22】



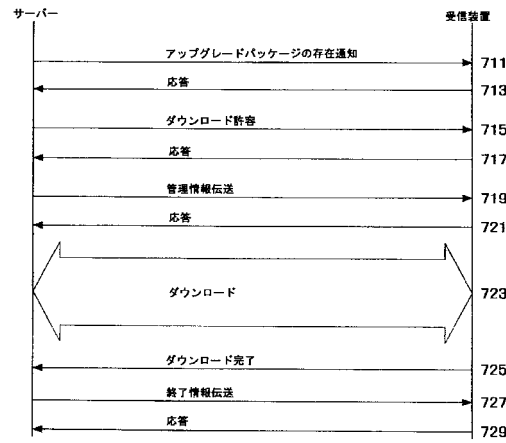
【図 23】



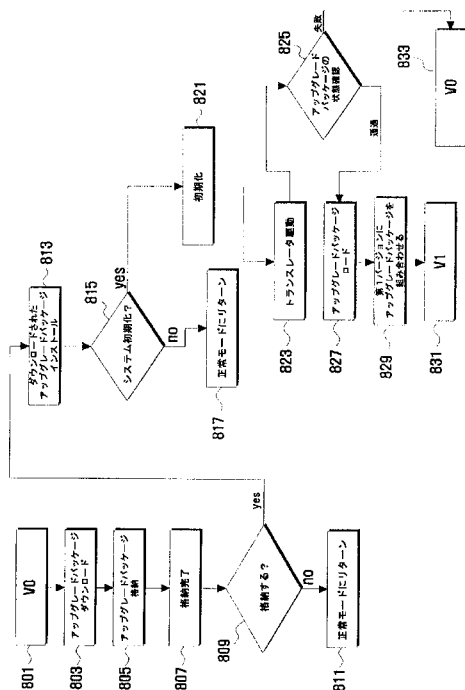
【図 24】



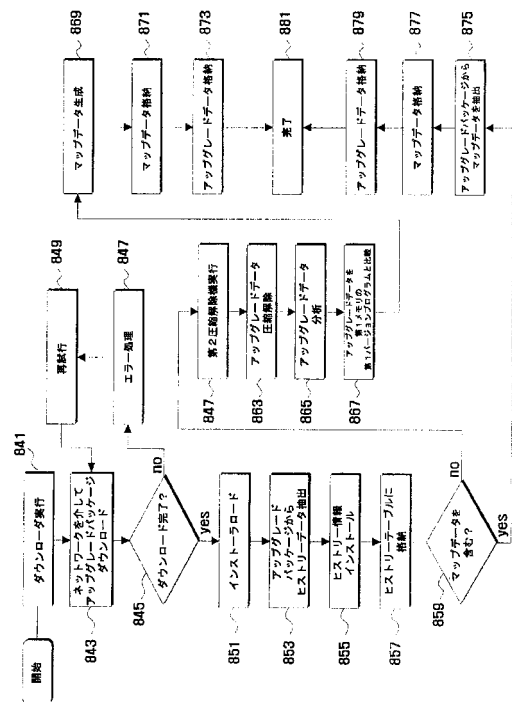
【図 25】



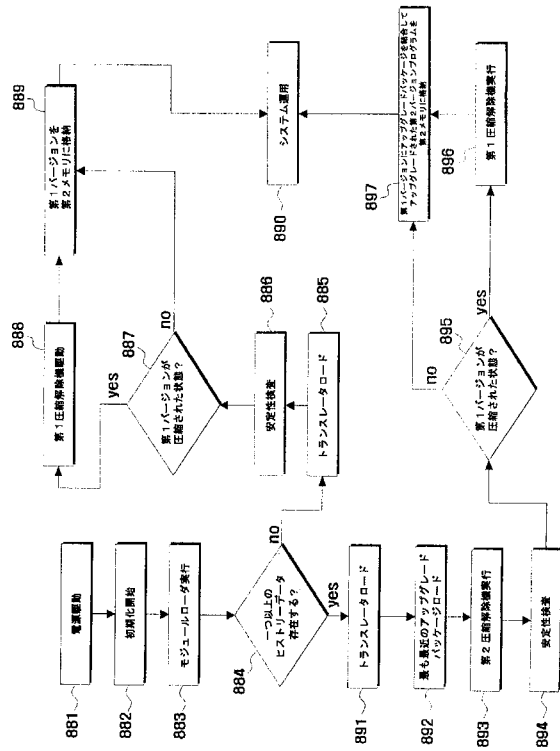
【図 26】



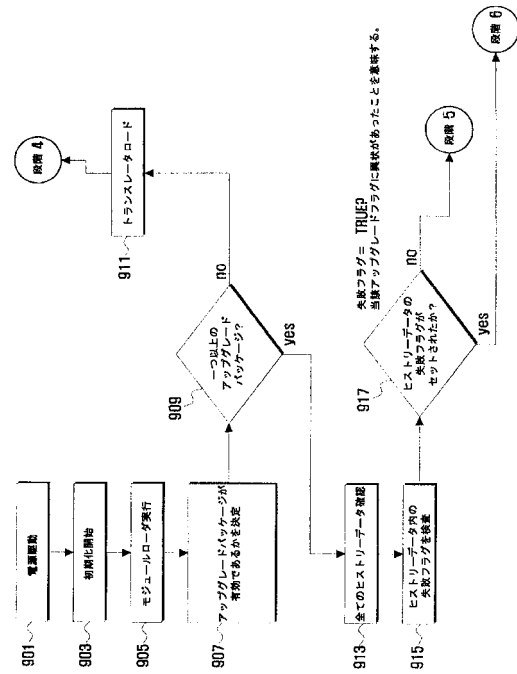
【図 27】



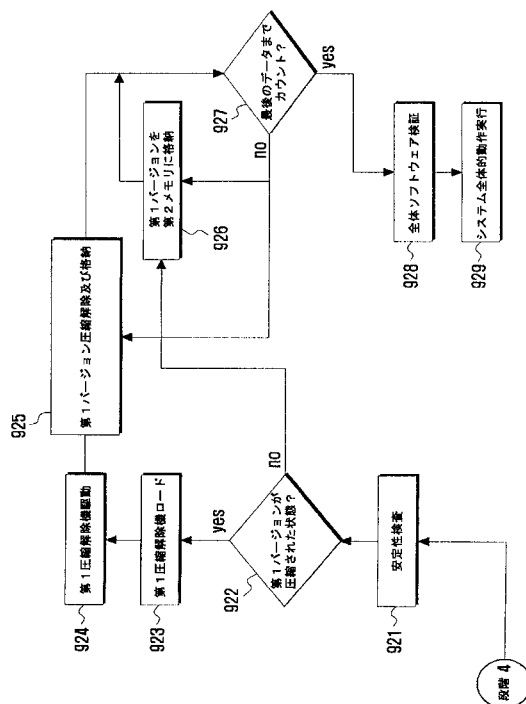
【図 28】



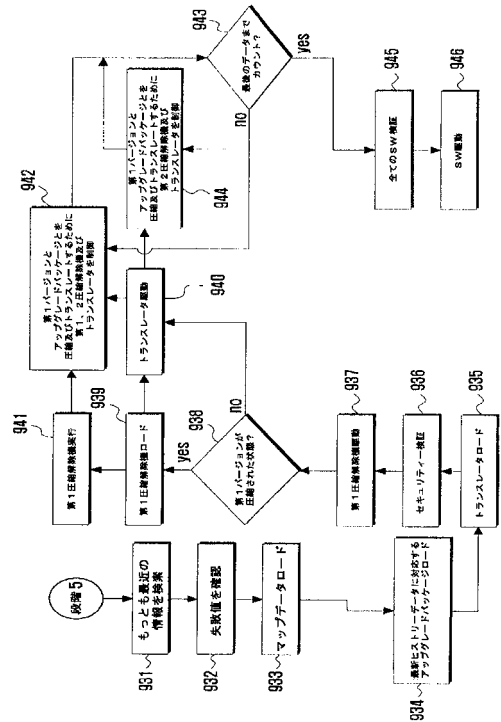
【図 29】



【図 30】



【図 31】



前記アップグレードパッケージをダウンロードし、前記第1バージョンを格納する第1メモリに前記ダウンロードされたアップグレードパッケージを格納し、アップグレード要求時、前記第1バージョンにアップグレードパッケージを合成し第2バージョンを生成して第2メモリに格納し、前記第2メモリの第2バージョンを介してシステムを運用する受信装置と、を備えることを特徴とするプログラムアップグレードシステム。

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/KR2007/002939
A. CLASSIFICATION OF SUBJECT MATTER		
<i>G06F 15/00(2006.01)i</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) IPC8 G06F		
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 since 1975 Japanese Utility models and applications for Utility models since 1975		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKIPASS(KIPO internal) & Keyword: package, version and upgrade		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 17-11209 A (SONY CORP.) 13 January 2005 See Paragraph [0017]-[0031], [0041]-[0045] & Figs. 1, 3	1-26
A	US 2004/0267833 A1 (EVYATAR MELLER et al.) 30 December 2004 See Paragraph [0194]-[0196] & Fig. 18	1-26
A	US 5,909,581 A (PARK SEONG KAB) 1 June 1999 See Column 2, Line 42-Column 3, Line 6 & Fig. 1	1-26
A	US 6,604,194 B1 (NAM CHA HEE) 5 August 2003 See abstract & Fig. 3	1-26
☐ Further documents are listed in the continuation of Box C. ☒ 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 28 SEPTEMBER 2007 (28.09.2007)		Date of mailing of the international search report 28 SEPTEMBER 2007 (28.09.2007)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 920 Dunsan-dong, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer YEO, Won Hyeon Telephone No. 82-42-481-5696 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2007/002939

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 17-011209A	13.01.2005	CN1809809A KR1020060021332A US2006200812A1 W02004114126A1	26.07.2006 07.03.2006 07.09.2006 29.12.2004
US 2004/0267833A1	30.12.2004	None	
US 5,909,581A	01.06.1999	KR100286006B1	10.01.2001
US 6,604,194B1	05.08.2003	KR1020000040430A	05.07.2000

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(特許庁注：以下のものは登録商標)

1. Bluetooth
2. ZIGBEE
3. GSM

【要約の続き】

ら選択されたアップグレードパッケージを結合してアップグレードされた第2バージョンのプログラムを生成し、アップグレードされた第2バージョンのプログラムを利用してシステムを運用するステップと、からなる。

【選択図】 図1