

(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2009-536396

(P2009-536396A)

(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 11/00	(2006.01)	G06F 9/06	630J		5B176
G06F 13/00	(2006.01)	G06F 13/00	530B		
G06F 9/445	(2006.01)	G06F 9/06	610Q		

審査請求 有 予備審査請求 未請求 (全 64 頁)

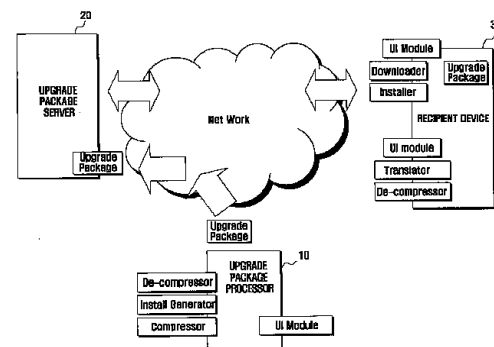
(21) 出願番号	特願2009-509450 (P2009-509450)	(71) 出願人	503447036
(86) (22) 出願日	平成19年6月18日 (2007.6.18)		サムスン エレクトロニクス カンパニー リミテッド
(85) 翻訳文提出日	平成20年11月10日 (2008.11.10)		大韓民国キョンギード, スウォンシー, ヨ ントンーク, マエタンードン 416
(86) 国際出願番号	PCT/KR2007/002947	(74) 代理人	100094145
(87) 国際公開番号	W02007/148900		弁理士 小野 由己男
(87) 国際公開日	平成19年12月27日 (2007.12.27)	(74) 代理人	100121382
(31) 優先権主張番号	10-2006-0054746		弁理士 山下 託嗣
(32) 優先日	平成18年6月19日 (2006.6.19)	(72) 発明者	オ スーンジョ
(33) 優先権主張国	韓国 (KR)		大韓民国, キョンギード, スウォンシー, 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 EA02 EA07 EA08 EB00
			最終頁に続く

(54) 【発明の名称】 OTA (Over-the-air) が可能な携帯装置のためのプログラムアップグレード方法
およびシステム

(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 (Over-the-Air) プログラムメカニズムを用いた移動端末のプログラム(ファームウェア及び応用プログラムの作動を含む)をアップデートすることができる装置及び方法に関する。

【解決手段】 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バージョンプログラムを入力され、二つのバージョンのプログラムの差による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バージョンプログラムを用いて



【特許請求の範囲】

【請求項 1】

ネットワーク上でプログラムバージョンをアップグレードする方法において、

第 1 バージョン及び第 2 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、

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、

前記受信装置が、前記アップグレードパッケージサーバーのアップグレードパッケージをダウンロードし、第 1 メモリに前記アップグレードパッケージをインストールし、アップグレード要求によって第 2 メモリに前記第 2 バージョンプログラムをロードするために、前記アップグレードパッケージと前記第 1 バージョンプログラムとを結合するステップと、

を備えることを特徴とするプログラムアップグレード方法。

【請求項 2】

前記アップグレードパッケージは、前記アップグレードパッケージを前記第 1 バージョンプログラムと結合するためのインストールデータと、前記第 1 バージョンプログラムに結合されるアップグレードデータと、を含むことを特徴とする請求項 1 に記載のプログラムアップグレード方法。

【請求項 3】

前記インストールデータは、前記第 1 バージョンプログラム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 バージョンプログラムに結合するためのマップ情報を含むマップデータと、を有することを特徴とする請求項 2 に記載のプログラム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、

前記第 1 及び第 2 バージョンプログラムを、設定されたサイズのブロックデータに分割して比較するステップと、

前記比較結果により、前記第 2 バージョンプログラムのブロックデータを前記第 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含むインストールデータを発生するステップと、

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として生成するステップと、

を有することを特徴とする請求項 1 に記載のプログラムアップグレード方法。

【請求項 5】

前記第 1 バージョンプログラムと、少なくとも一つの前記アップグレードパッケージデータとは、第 1 メモリに各々格納されることを特徴とする請求項 4 に記載のプログラムアップグレード方法。

【請求項 6】

前記アップグレード要求は、システムが初期化された時、又は、ユーザーの要求がある時に、バージョンを選択することによって発生することを特徴とする請求項 5 に記載のプログラムアップグレード方法。

【請求項 7】

前記マップデータは、前記第 2 バージョンプログラムのブロックデータを各々前記第 1 バージョンプログラムのブロックデータに結合するためのコマンド及びブロックインデックスを含むことを特徴とする請求項 4 に記載のプログラムアップグレード方法。

【請求項 8】

前記マップデータは、[Cmd] [start block No][number of block] [flag] [distance]を含み、Cmdは、コピー、変更、及びシフト中の一つのコマンドであり、start block Noは、当該コマンドの開始ブロック番号を意味し、number of blockは、当該コマンドに対応するブロック

10

20

30

40

50

の数を意味し、`flag`は、前記アップグレードデータの状態を表す識別情報であり、`distance`は、前記第2バージョンプログラムのブロックが位置する前記第1バージョンプログラムのブロックインデックスであることを特徴とする請求項7に記載のプログラムアップグレード方法。

【請求項9】

前記「変更」コマンドは、前記第1バージョンプログラムにおいて新しいブロックの追加、又は、ブロックの変更が行われることを意味することを特徴とする請求項8に記載のプログラムアップグレード方法。

【請求項10】

前記マップデータは、`[Cmd] [start block No, number of block] [flag] [distance] [difference]`を含み、`Cmd`は、コピー、変更、及びシフト中の一つのコマンドであり、`start block No`は、当該コマンドの開始ブロック番号を意味し、`number of block`は、当該コマンドに対応するブロックの数を意味し、`flag`は、前記アップグレードデータの状態を表す識別情報であり、`distance`は、前記第2バージョンプログラムのブロックが位置する前記第1バージョンプログラムのブロックインデックスであり、`difference`は、前記第1及び第2バージョンプログラムのブロックデータの差を意味することを特徴とする請求項7に記載のプログラムアップグレード方法。

【請求項11】

前記`[difference]`フィールドに、前記第1及び第2バージョンプログラムのブロックデータの差データがあると、当該ブロックの前記アップグレードデータは生成されないことを特徴とする請求項10に記載のプログラムアップグレード方法。

【請求項12】

前記第1バージョンプログラムは、前記受信装置の製造時にインストールされるプログラムであり、前記第2バージョンプログラムは、前記第1バージョンプログラムをアップグレードしたプログラムであり、前記第2バージョンプログラムは、複数個生成され、前記受信装置にインストールされることを特徴とする請求項11に記載のプログラムアップグレード方法。

【請求項13】

システムのバージョンをアップグレードするためのアップグレードパッケージを生成する方法において、

前記第1及び第2バージョンプログラムを設定されたサイズのブロックデータに分割して比較するステップと、

前記比較結果によって前記第2バージョンプログラムのブロックデータを前記第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含むアップグレードデータを生成するステップと、

前記第1バージョンプログラムとの結合情報を有するヒストリーデータ及び前記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として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を少なくとも一つの前記受信装置にダウンロードし、不揮発性メモリに前記ダウンロードしたアップグレードパッケージをインストールし、アップグレード要請信号により前記アップグレードパッケージと前記第1バージョンプログラムとを結合して第2バージョンプログラムを生成し、前記第2バージョンプログラムを用いて前記受信装置を運用するステップと、
を備えることを特徴とするプログラムアップグレード方法。

【請求項14】

前記第1バージョンプログラムと少なくとも一つの前記アップグレードパッケージとは、不揮発性メモリに各々格納されることを特徴とする請求項13に記載のプログラムアップグレード方法。

【請求項15】

10

20

30

40

50

前記アップグレード要請信号は、システム初期化又はユーザーの要求によって生成され、前記アップグレードパッケージは、複数のアップグレードパッケージから選択可能であることを特徴とする請求項 14 に記載のプログラムアップグレード方法。

【請求項 16】

前記アップグレードパッケージは、前記アップグレードパッケージと前記第 1 バージョンプログラムとの関係を表すヒストリーデータを含むインストールデータを更に有することを特徴とする請求項 15 に記載のプログラムアップグレード方法。

【請求項 17】

前記アップグレードデータは、コピー、変更、及びシフトコマンド中の一つを備え、前記アップグレードデータのコピーブロックデータは、[start block No][number of block]を含み、変更ブロックデータは、[start block No][number of block][data]を含み、シフトブロックデータは、[start block No][number of block][previous version position]を含み、ここで、前記[start block No]は、開始するブロックのインデックスであり、[number of block]は、当該コマンドのブロック数を意味し、[data]は、変更ブロックのデータであり、[previous version position]は、シフトブロックの開始ブロックインデックスを意味することを特徴とする請求項 16 に記載のプログラムアップグレード方法。

【請求項 18】

前記シフトブロックデータの[previous version position]フィールドは、「第 2 バージョンプログラムの[block No]±ブロック数」、又は、「第 2 バージョンプログラムの[block No]-第 1 バージョンプログラムの[block No]」中のいずれか一つで表現されることを特徴とする請求項 17 に記載のプログラムアップグレード方法。

【請求項 19】

システムのバージョン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が、第 1 バージョン及び第 2 バージョンプログラムを受信し、前記二つのバージョンプログラムの差による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、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をダウンロードし、前記第 1 バージョンプログラムを格納する第 1 メモリのアップグレードパッケージ領域に、前記ダウンロードされたアップグレードパッケージをインストールし、アップグレード要求時に、前記第 1 バージョンプログラムに、前記アップグレードパッケージの中から選択された前記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、前記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ステップと、

を備えることを特徴とするプログラムアップグレード方法。

【請求項 20】

前記アップグレードパッケージを生成するステップが、

前記第 1 及び第 2 バージョンプログラムを設定されたサイズのブロックデータに分割して比較するステップと、

前記比較結果により、前記第 2 バージョンプログラムのブロックデータを前記第 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含む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として生成するステップと、を有することを特徴とする請求項 19 に記載のプログラムアップグレード方法。

【請求項 21】

前記アップグレードパッケージをインストールする第1メモリは、
前記第1バージョンプログラムを格納する第1バージョン格納領域と、
前記インストールデータ及びアップグレードデータを格納するアップグレードパッケージ格納領域とを含み、

前記アップグレードパッケージ格納領域は、少なくとも二つ備えられることを特徴とする請求項20に記載のバージョンアップグレード方法。

【請求項22】

前記アップグレードパッケージのインストールデータは、前記第1バージョンプログラムの結合情報であるヒストリーデータと、前記第1バージョンプログラムとブロックマップのためのマップデータとを含み、

前記アップグレードパッケージを前記第1メモリにインストールするステップは、
前記アップグレードパッケージから前記ヒストリーデータを抽出し、当該アップグレードパッケージ格納領域のヒストリーデータ領域にインストールするステップと、

前記アップグレードパッケージから前記マップデータを抽出して当該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ステップと、

前記アップグレードパッケージのアップグレードデータを、当該アップグレードパッケージ格納領域のアップグレードデータ領域にインストールするステップと、を有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方法。

【請求項23】

前記第2バージョンプログラムを生成する過程が、

アップグレード要求時に、前記第1メモリにインストールされた最新バージョンのアップグレードパッケージをロードするステップと、

前記ロードされたアップグレードパッケージのマップデータによって前記第1バージョンプログラムに前記アップグレードデータを結合して前記第2バージョンプログラムを生成し、前記第2バージョンプログラムを前記第2メモリに格納するステップと、
を有することを特徴とする請求項22に記載のプログラムアップグレード方法。

【請求項24】

前記第1メモリは、不揮発性メモリであり、前記第2メモリは、揮発性メモリであることを特徴とする請求項23に記載のプログラムアップグレード方法。

【請求項25】

前記アップグレード要求は、システム初期化の実行後、生成されることを特徴とする請求項24に記載の前記プログラムアップグレード方法。

【請求項26】

システムのプログラムをアップグレードするためのアップグレードパッケージを生成する方法において、

前記第1及び第2バージョンプログラムを設定されたサイズのブロックデータに分割して比較するステップと、

前記比較結果により前記第2バージョンプログラムのブロックデータを前記第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含む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ステップと、
を備え、

受信側システムが、前記アップグレードパッケージをダウンロードして前記格納している第1バージョンプログラムに前記アップグレードパッケージを結合してプログラムをアップグレードすることを特徴とするアップグレードパッケージの生成方法。

【請求項27】

前記マップデータは、[Cmd] [start block No] [number of block] [flag] [distance]を含み、Cmdは、コピー、変更、及びシフト中の一つのコマンドであり、start block Noは、当該コマンドの開始ブ

10

20

30

40

50

ロックの番号を意味し、`number of block`は、当該コマンドに対応するブロックの数を意味し、`flag`は、前記アップグレードデータの状態を表す識別情報であり、`distance`は、前記第2バージョンプログラムのブロックが位置する前記第1バージョンプログラムのブロックインデックスであることを特徴とする請求項26に記載のプログラムアップグレード方法。

【請求項28】

前記「変更」コマンドは、前記第1バージョンプログラムにおいて新たなブロックの挿入またはブロックの変更を行うことを意味することを特徴とする請求項27に記載のプログラムアップグレード方法。

【請求項29】

前記マップデータは、`[Cmd] [start block No] [number of block] [flag] [distance] [difference]`を含み、`Cmd`は、コピー、変更、及びシフト中の一つのコマンドであり、`start block No`は、当該コマンドの開始ブロックの番号を意味し、`number of block`は、当該コマンドに対応するブロックの数を意味し、`flag`は、前記アップグレードデータの状態を表す識別情報であり、`distance`は、前記第2バージョンプログラムのブロックが位置する前記第1バージョンプログラムのブロックインデックスであり、`difference`は、前記第1及び第2バージョンプログラムのブロックデータの差を意味することを特徴とする請求項26に記載のプログラムアップグレード方法。

【請求項30】

前記`difference`に、第1及び第2バージョンプログラムのブロックデータの差データがあると、当該ブロックのアップグレードデータは生成されないことを特徴とする請求項29に記載のプログラムアップグレード方法。

【請求項31】

第1バージョン及び第2バージョンプログラムを入力され、前記二つのバージョンプログラムの差による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の生成を受信装置に通知及びダウンロード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バージョンプログラムを格納する第1メモリのアップグレードパッケージ領域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インストールし、アップグレード要求時に、前記第1バージョンプログラムに前記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前記第2バージョンプログラムを第2メモリに格納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。

【請求項32】

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バージョンプログラムを設定されたサイズのブロックデータに分割して比較する比較器と、

前記比較結果により前記第2バージョンプログラムのブロックデータを前記第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含む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、

前記インストールデータ及び前記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パッケージ発生器と、を有することを特徴とする請求項31に記載のプログラムアップグレードシステム。

【請求項33】

前記受信装置は、

前記ダウンロードされた前記アップグレードパッケージを前記第1メモリの対応するアップグレードパッケージ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、前記第1メモリにインストールされた前記第1バージョンプロ

10

20

30

40

50

グラム、及び前記アップグレードパッケージをロードし、前記マップデータにより前記アップグレードデータを前記第1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前記第2バージョンプログラムを生成した後、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32に記載のプログラムアップグレードシステム。

【請求項34】

前記アップグレード要求は、システム初期化又はユーザーの要求によって生成され、前記アップグレードパッケージのバージョンは選択可能であることを特徴とする請求項33に記載のプログラムアップグレードシステム。

【請求項35】

前記インストールデータは、前記アップグレードパッケージを前記第1バージョンプログラムと結合するためのヒストリーデータを更に備えることを特徴とする請求項34に記載のシステムのプログラムアップグレードシステム。

【請求項36】

前記マップデータは、前記第2バージョンプログラムのブロックデータを各々前記第1バージョンプログラムのブロックデータに結合するためのコマンド及びブロックインデックスを含むことを特徴とする請求項35に記載のバージョンアップグレードシステム。

【請求項37】

前記マップデータは、[Cmd][start block No][number of block][flag][distance]を含み、Cmdは、コピー、変更、及びシフト中の一つのコマンドであり、start block Noは、当該コマンドの開始ブロックの番号を意味し、number of blockは、当該コマンドに対応するブロックの数を意味し、flagは、前記アップグレードデータの状態を表す識別情報であり、distanceは、前記第2バージョンプログラムのブロックが位置する前記第1バージョンプログラムのブロックインデックスであることを特徴とする請求項35に記載のプログラムアップグレードシステム。

【請求項38】

前記「変更」コマンドは、前記第1バージョンプログラムにおいて新たなブロックの挿入またはブロックの変更を行うことを意味することを特徴とする請求項36に記載のプログラムアップグレードシステム。

【請求項39】

前記マップデータは、[Cmd][start block No][number of block][flag][distance][difference]を含み、Cmdは、コピー、変更、及びシフト中の一つのコマンドであり、start block Noは、当該コマンドの開始ブロックの番号を意味し、number of blockは、当該コマンドに対応するブロックの数を意味し、flagは、前記アップグレードデータの状態を表す識別情報であり、distanceは、前記第2バージョンプログラムのブロックが位置する前記第1バージョンプログラムのブロックインデックスであり、differenceは、前記第1及び第2バージョンブロックデータの差を意味することを特徴とする請求項35に記載のプログラムアップグレードシステム。

【請求項40】

前記differenceに、前記第1及び第2バージョンプログラムのブロックデータの差データがあると、当該ブロックの前記アップグレードデータは生成されないことを特徴とする請求項39に記載の前記プログラムアップグレードシステム。

【請求項41】

前記第1バージョンプログラムは、前記受信装置の製造時にインストールされるプログラムであり、前記第2バージョンプログラムは、前記第1バージョンプログラムをアップグレードしたプログラムであり、前記第2バージョンプログラムは、複数個生成され、前記受信装置にインストールされることを特徴とする請求項40に記載のプログラムアップグレードシステム。

10

20

30

40

50

【請求項 4 2】

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 バージョンプログラムを設定されたサイズのブロックデータに分割して比較する比較器と、

前記第 1 バージョンプログラムの結合情報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、

前記比較結果により前記第 2 バージョンプログラムのブロックデータを前記第 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含むアップグレードデータを生成し、前記第 1 バージョンプログラムとの結合情報を有するヒストリーデータ及び前記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パッケージ発生器と、
を有することを特徴とする請求項 3 1 に記載のプログラムアップグレードシステム。

10

【請求項 4 3】

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のアップグレードデータを分析して前記マップデータを生成し、前記ヒストリーデータ、前記マップデータ、及び前記アップグレードデータを含むアップグレードパッケージを前記第 1 メモリの対応するアップグレードパッケージ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、前記マップデータにより前記アップグレードデータを前記第 1 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て前記第 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 4 2 に記載のプログラムアップグレードシステム。

20

【請求項 4 4】

前記アップグレード要求は、システム初期化又はユーザーの要求によって発生し、前記パッケージのバージョンは選択可能であることを特徴とする請求項 4 3 に記載の前記バージョンアップグレードシステム。

【請求項 4 5】

前記アップグレードデータは、コピー、変更、及びシフトコマンド中の一つを備え、前記アップグレードデータのコピーブロックデータは、[start block No][number of block]を含み、変更ブロックデータは、[start block No][number of block][data]を含み、シフトブロックデータは、[start block No][number of block][previous version position]を含み、ここで、前記[start block No]は、開始するブロックのインデックスであり、[number of block]は、当該コマンドのブロック数を意味し、[data]は、変更ブロックのデータであり、[previous version position]は、シフトブロックの開始ブロックインデックスを意味することを特徴とする請求項 4 4 に記載のプログラムアップグレードシステム。

30

【請求項 4 6】

前記シフトブロックデータの[previous version position]は、「第 2 バージョンプログラムの[block No]±ブロック数」、又は、「第 2 バージョンプログラムの[block No]－第 1 バージョンプログラムの[block No]」中のいずれか一つで表現されることを特徴とする請求項 4 5 に記載のプログラムアップグレードシステム。

40

【発明の詳細な説明】

【技術分野】

【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、OTA (Over-the-Air) プログラムメカニズムを用いた移動端末のプログラム(ファームウェア及び応用プログラムの作動を含む)をアップデートすることができるシステム及び方法に関する。

【背景技術】

50

【0002】

携帯電話やPDA(Personal digital assistants)のような電子製品は、ファームウェアや応用プログラムを含んでいる。しかし、前記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。ファームウェアや応用ソフトウェアを修理及びアップグレードするために、ユーザーは、端末機を供給する製造会社やサービスキャリア(service carrier)等を訪問することになる。

【0003】

この時、前記端末機が、無線通信機能を備える携帯端末機である場合、前記アップグレードサービスは、OTA(over the air)形態でも可能である。前記OTA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(patch software)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することができるソフトウェアパッケージ(software package)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とが必要である。しかし、ほとんどの従来のOTAが可能な装置は、OTA作動の安全性に限界があった。

【発明の開示】

【発明が解決しようとする課題】

【0004】

本発明は、少なくとも前述した問題点を解決するために考案したものであり、無線通信をすることで伝達されたアップグレードパッケージによってファームウェアをアップデートすることのできるOTAが可能な携帯端末機のプログラムアップグレードシステム及び方法に関する。

【課題を解決するための手段】

【0005】

本発明は、移動端末に設置された旧バージョンのファームウェアと新バージョンのファームウェアとの差をアップグレードパッケージとして生成し、生成されたアップグレードパッケージと移動端末の旧バージョンのファームウェアとを組み合わせ、無線通信を用いて移動端末のファームウェアアップデートを可能にするOTAが可能な移動端末のプログラム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、無線通信をすることで受信されたアップグレードパッケージによってファームウェアのアップデートを可能にするOTAが可能な移動端末のプログラムアップグレードシステム及び方法を提供する。ここで、アップグレードパッケージは、ヒストリーデータと、アップグレードパッケージ及びプログラムのアップグレードターゲットバージョンの関係を表すインデックス情報を含むマップデータと、二つのバージョンのプログラムの差を表すアップグレードデータとを含む。

【0007】

本発明は、アップグレードパッケージの生産を可能にするOTAが可能な移動端末のプログラムアップグレードシステム及び方法を提供する。アップデートパッケージは、新バージョンと旧バージョンとのファームウェアの差によって生成されたアップグレードデータと、アップグレードパッケージとファームウェアプログラムのアップグレードターゲットバージョンとの関係を表すヒストリーデータと、二つのバージョンのブロックをマップするマップデータとを有する。

【0008】

本発明は、移動端末に設置された基準ファームウェアと、無線通信をすることでダウンロードされたアップグレードパッケージとを組み合わせ、移動端末のファームウェアのアップデートを可能にするOTAが可能な移動端末のプログラムアップグレードシステム及び方法を提供する。

【0009】

本発明は、移動端末に設置された基準ファームウェアと、無線通信をすることでダウンロードされた少なくとも一つのアップグレードパッケージとを組み合わせ、移動端末のファームウェアのアップデートを可能にするOTAが可能な移動端末のプログラムアップグレードシステム及び方法を提供する。

【0010】

前記目的を達成するための本発明に係るシステムのバージョン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が第1バージョン及び第2バージョンプログラムを入力され、前記二つのバージョンのプログラムの差による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及びダウンロードするステップと、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、前記第1バージョンプログラムに前記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生成し、前記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ステップと、を備えることを特徴とする。

10

【0011】

ここで、前記アップグレードパッケージを生成する過程は、前記第1及び第2バージョンプログラムを、設定されたサイズのブロックデータに分割して比較するステップと、前記比較結果によって前記第2バージョンのブロックデータを前記第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含むインストールデータを発生するステップと、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アップグレードパッケージとして生成するステップと、を備えることを特徴とする。

20

【0012】

前記受信装置が、前記アップグレードプログラムを生成する過程は、前記ダウンロードされるアップグレードパッケージをメモリのアップグレードパッケージ領域にインストールし、前記メモリは、前記第1バージョンプログラムを格納する領域及び複数のアップグレードパッケージを格納する領域を備えるステップと、アップグレード要求時、前記マップデータによって前記アップグレードデータを前記第1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するステップと、前記生成された第2バージョンプログラムで前記受信装置を運用するステップと、を有することを特徴とする。

30

【0013】

また、前記アップグレードパッケージを生成する過程は、前記第1及び第2バージョンプログラムを設定されたサイズのブロックデータに分割して比較するステップと、前記比較結果によって前記第2バージョンのブロックデータを前記第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含むアップグレードデータを発生するステップと、前記第1バージョンプログラムとの結合情報を有するヒストリー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ステップと、を有することを特徴とする。

40

【0014】

前記受信装置が、アップグレードプログラムを生成する過程は、前記ダウンロードされるアップグレードパッケージのアップグレードデータを分析してマップデータを生成し、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含むアップグレードパッケージをメモリのアップグレードパッケージ領域にインストールし、前記メモリは、前記第1バージョンプログラムを格納する領域及び複数のアップグレードパッケージを格納する領域を備えるステップと、アップグレード要求時、前記マップデータによって前記アップグレードデータを前記第1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するステップと、前記生成された第2バージョンプログラ

50

ラムで前記受信装置を運用するステップと、を有することを特徴とする。

【0015】

また、前記目的を達成するための本発明の実施形態に係るシステムのバージョン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が第1バージョン及び第2バージョンプログラムを受信し、前記二つのバージョンのプログラムの差による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及びダウンロードするステップと、前記受信装置が前記アップグレードパッケージをダウンロードし、前記第1バージョンプログラムを格納する第1メモリのアップグレードパッケージ領域に、前記ダウンロードされたアップグレードパッケージをインストールし、アップグレード要求時、前記第1バージョンプログラムに、前記アップグレードパッケージの中から選択された、前記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、前記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ステップ、とを備えることを特徴とする。

10

【0016】

また、前記目的を達成するための本発明の実施形態に係るシステムのプログラムをアップグレードするためのアップグレードパッケージを生成する方法は、前記第1及び第2バージョンプログラムを、設定されたサイズのブロックデータに分割して比較するステップと、前記比較結果によって前記第2バージョンのブロックデータを、前記第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含むインストールデータを発生するステップと、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アップグレードパッケージとして生成するステップとからなり、受信側のシステムが、前記アップグレードパッケージをダウンロードして、前記格納している第1バージョンプログラムに前記アップグレードパッケージを結合してプログラムをアップグレードすることを特徴とする。

20

【0017】

また、本発明に係るシステムのバージョン情報アップグレード装置は、第1バージョン及び第2バージョンプログラムを受信し、前記二つのバージョンのプログラムの差による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の生成を、受信装置に通知及びダウンロード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バージョンプログラムを格納する第1メモリのアップグレードパッケージ領域に、前記ダウンロードされたアップグレードパッケージをインストールし、アップグレード要求時、前記第1バージョンプログラムに、前記アップグレードパッケージの中から選択された、前記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、前記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受信装置として構成されることを特徴とする。

30

【0018】

ここで、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バージョンプログラムを、設定されたサイズのブロックデータに分割して比較する比較器と、前記比較結果によって前記第2バージョンのブロックデータを前記第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アップグレードパッケージとして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。

40

【0019】

前記受信装置は、前記第1バージョンプログラムを格納する領域、及び複数のアップグレードパッケージを格納する領域を備える第1メモリと、システムプログラムを格納する第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第1メモリの対応するアップグレードパッケージ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、前記第1メモリにインストールされた第1バージョンプログラム及びアップグレ

50

【 0 0 2 0 】

10

20

【 0 0 2 2 】

【 0 0 2 3 】

30

40

50

受信側の装置は、装置の製造時、第1バージョンプログラムがインストールされており、アップグレードプログラムが発生すると、第2バージョンアップグレードパッケージをダウンロードして格納することができ、また、一つ又はそれ以上の第2バージョンを格納することができる。従って、「アップグレードパッケージ」は、第1バージョンプログラムをアップグレードするための情報であり、第1バージョンプログラムから第2バージョンプログラムとして生成するとき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バージョンアップグレードパッケージは、第1バージョンプログラムをコピー(c o p y)、シフト(s h i f t)、及び変更(m o d i f y)のコマンド及びブロック位置データ、そして、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(f i r m w a r e)及びソフトウェア(s o f t w a r e)を含む用語として使われる。

10

【0025】

「第1メモリ」とは、第1バージョン及び第2バージョンを格納するメモリを意味する。「第2メモリ」とは、第1バージョンプログラムに、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プログラム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領域に区分されたメモリとして構成することができ、また、別途の二つ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は、第1及び第2メモリが、物理的に別途の二つ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(n o n - v o l a t i l e m e m o r y)、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(v o l a t i l e m e m o r y)、かつSDRAM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バージョンプログラム、そして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を格納し、第2バージョンアップグレードパッケージは、各バージョンを識別するためのヒストリーデータ(更に、マップデータを含むことができる)、及びアップグレードデータを各々格納する。システムの起動時、又は、ユーザーの命令によってシステムは第1バージョンプログラムに第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プログラムを第2メモリに格納した後、アップグレードされた第2バージョンプログラムによってシステムを運用する。ここで、第1バージョンを基準プログラムにすることができる。第2バージョンは、アップグレードパッケージによって生成することができ、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することができ、インストールデータは、ヒストリーデータ及び/又はマップデータから構成することができ、第1バージョンは、システムのオリジナルプログラム(又は、基準プログラム)にすることができる、第2バージョンは、第1バージョンとの差を有するアップグレードデータ、及び、これをインストールするためのインストールデータを含むことができる。従って、第2メモリに格納されるプログラムは、第1バージョンに第2バージョンを結合して生成されたプログラムにすることができる。

20

30

【0026】

本発明の実施形態に係るシステムアップグレード及び維持補修の方法は、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プログラム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、伝送されるアップグレードパッケージをダウンロードして、システムプログラムをアップグレードするシステムの構成を示す図である。

40

【0027】

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の発生時、第1バージョンプログラムと比較して、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。この時、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との連結方法により、無線(a i r p r o t o c o l s : C D M A、U M T S等)、近

50

距離無線通信(b l u e t o o t h、Z i g b e e等)、近距離有線通信(U S B、U A R T)、有無線インターネット(W i b r o、W i f i、W i m a x等)方式で伝達することができる。又は、アップグレードパッケージサーバー20と、アップグレードパッケージプロセッサ10とは、一体型で構成することもできる。次に、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が発生したことを複数の受信装置30に通知(n o t i f i c a t i o n)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。この時、受信装置のアップグレードパッケージの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の連結方法により、無線(a i r p r o t o c o l s : C D M A、U M T S等)、近距離無線通信(b l u e t o o t h、Z i g b e e等)、近距離有線通信(U S B、U A R T)、有無線インターネット(W i b r o、W i f i、W i m a x等)方式等を使用することができる。

10

【0028】

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メモリから構成することができる。ここで、第1メモリ及び第2メモリは、一つのメモリで構成することができ、また各々別途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(m e r g e)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。

20

【0029】

次に、アップグレードパッケージプロセッサ10の動作を説明する。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、その周辺構成との関係を説明するための図である。

30

【0030】

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、アップグレードされたプログラムであり、受信装置30においては、アップグレードデータの結合(m e r g e、a s s e m b l i n g)により、第1プログラム50から生成されるプログラムである。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バージョンアップグレードパッケージ(u p g r a d e p a c k a g e)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することができる。その場合、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である。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することができ、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り、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、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。更に、アップグレードパッケージは、ヒストリーデータ及びマップデータのみから構成することもできる。その場合、マップデータは、変更されたデータをアップグレードデータとして生成せず、マ

40

50

ップデータに含め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、有線通信方式のいずれかを使用することができる。

【0031】

図 3－図 8 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である。本発明の実施形態に係るアップグレードパッケージは、アップグレードデータ、ヒストリーデータ、及びマップデータから構成されるアップグレードパッケージを生成することができ、また、ヒストリーデータ及びマップデータのみから構成されるアップグレードパッケージを生成することもできる。更に、本発明の実施形態において、第 1 バージョンプログラム(以下、第 1 バージョン又は V1 と称する)を、第 2 バージョンプログラム(以下、第 2 バージョン又は V2 と称する)にアップグレードする場合、アップグレード過程において、シフト動作を減らすために第 1 バージョン V1 のプログラムにギャップ領域を割り当てることができる。

【0032】

図 3－図 8 を参照して、本発明の実施形態に係るアップグレードパッケージの生成方法を説明する。本発明の実施形態において、アップグレードパッケージプロセッサ 10 は、第 1 バージョンプログラム(以下、第 1 バージョン及び/又は V1 と称する)、及び第 2 バージョンプログラム(以下、第 2 バージョン及び/又は V2 と称する)のデータを設定されたサイズのブロック(以下、マクロブロック(macro block:MB)と称する)のサイズで比較し、比較結果により、各ブロックの属性(コピー(C:copy)、変更(M:modify)、シフト(S:shift))を把握し、ブロックデータの属性に応じ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、のように、アップグレードデータ、ヒストリーデータ、及びマップデータから構成するでき、また、ヒストリーデータ及びマップデータのみから構成することもできる。マクロブロックは、データを特定のサイズに分割したものであり、16 インストラクション(instruction)は 16 bit であり、32 インストラクション(instruction)は 32 bit であることを意味する。

【0033】

アップグレードパッケージプロセッサ 10 が、マップデータを含むアップグレードパッケージを生成する場合、マップデータは、C(copy)、M(modify)、S(shift)のコマンド(command)を有し、マップデータの形態(map format)は、下記のように構成することができる。

【0034】

[Cmd][start block No、number of block][flag][distance][difference]

マップデータにおいて、Cmd は、C、M、S のいずれかである。start block No とは、当該コマンドの開始ブロック番号を意味し、number of block とは、当該コマンドに対応するブロックの数を意味する。flag は、0、1、2 から構成することができ、flag 0 は、コピー(just copy(Cmd:C))、及びシフト(just shift(Cmd:S))の場合に使用できるフラグであり、flag 1 は、V1 プログラムからブロックデータを生成することができる場合(generated data from V1(Cmd:M))に使用できるフラグであり、flag 2 は、V1 プログラムからブロックデータを生成することができず、ブロックデータは、デルタパッケージ(delta package)である場合(data is in delta package(not use V1)(Cmd:M))に使用されるフラグである。ここで、デルタパッケージは、アップグレードデータにすることができる。distance とは、V2 プログラムの当該ブロックが位置する V1 プログラムのブロックインデックス情報を意味する。difference とは、V2 プログラムのブロックデータと、V1 プログラムのブロックデータとの差の値を意味する。

【0035】

本発明の実施形態において、アップグレードパッケージプロセッサ10は、V1ブロック及びV2ブロックのデータが相違すると、変更ブロックデータを生成する。この時、V2の変更ブロックデータが確認されると、アップグレードパッケージプロセッサ10は、V1の設定された区間のブロックを探索(s e a r c h i n g)し、V2変更ブロックデータと同一のデータがあるか否かを確認する。

【0036】

図3-図5は、変更ブロックデータの発生時、現在のブロック位置で、設定されたブロック数(以下、探索区間と称する)だけ第1バージョンのブロックをスイングしながら、第2バージョンの変更ブロックデータと同一のブロックデータがあるか否かを検査する。ここで、探索区間は、2、4、8、16、32...のブロックに設定することができ、本発明の実施形態においては、16ブロックと仮定する。アップグレードパッケージプロセッサ10は、第2バージョンの変更ブロックを感知すると、現在の第2バージョンブロックに対応する第1バージョンのブロックインデックスの位置で、探索区間のブロックを検索する。この時、探索区間は、第1バージョンの現在のブロックインデックスの前後に位置する各16個のブロック(すなわち、32個のブロック)である。従って、V2プログラムにおいて、変更ブロックデータを確認すると、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は、V1プログラムの対応するブロック位置で、設定された探索区間内のブロックデータを探索し、同一のブロックデータがあるか否かを確認する。この時、V1プログラムの設定された区間内に同一のブロックデータがあれば、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を生成せず、マップデータに対応するV1プログラムのブロックインデックスを含めてアップグレードパッケージを生成する。

【0037】

この時、変更ブロックのマップデータを生成する方法は、一つのブロックを比較して生成することができ、複数のブロックを比較して生成することもできる。一般的に、第2バージョンの変更データは、複数のブロックにして生成することができる。すなわち、プログラムバージョンがアップグレードされる場合、アップグレードプログラム(すなわち、V2プログラム)は、基準となるプログラムにアップグレードされる内容が挿入及び/又は代替される方式の変更ブロックが含まれる形態として生成することができる。その場合、変更ブロックデータは、基準プログラム(ここでは、V1プログラム)と同一又は類似のブロックデータにすることができる。従って、本発明の実施形態においては、変更ブロックデータの確認時、複数のブロックを比較して変更ブロックデータのマップデータを生成する例について説明する。その場合、アップグレードパッケージプロセッサ10は、変更ブロックを含む複数のブロックデータを、V1プログラムの探索区間内のブロックと比較し、変更ブロックのマップデータを生成する。ここで、複数のブロックを比較する場合、そのブロック数は2、4、8、16...等に設定する。これは、アップグレードパッケージにおいて、マップデータを生成する時、2の倍数にて演算を処理し、演算の処理速度を向上させるためである。その場合、V2プログラムのブロックデータをV1プログラムのブロックデータと比較し、コピー又はシフトである場合、当該ブロックのインデックスを含むマップデータを発生する。しかし、変更ブロックデータが発生すると、アップグレードパッケージプロセッサ10は、ブロックデータを比較し、探索区間内のV1プログラムのブロックデータと同一であれば、対応する第2バージョンブロックデータのコマンドを変更と決定し、当該ブロックのインデックスを確認してマップデータとして生成する。この時、複数のブロックの検索方法は、2ブロック単位で検索する場合、0、2、4、6、...等のように、偶数のブロックインデックス位置で検索し、4ブロック単位で検索する場合、0、4、8、...等のように、4番目ごとに、ブロックインデックス位置で検索し、8ブロック単位で検索する場合、0、8、16等のように、8番目ごとに、ブロックインデックス位置で検索をする。これは、アップグレードパッケージプロセッサ10が、この二つのバージョンのブロックデータを演算する時、複雑な演算を簡単にするためである。

【0038】

10

20

30

40

50

下記の説明において、図3－図5は、探索区間が32ブロック(現在のブロックインデックスの前後16ブロック)であり、変更ブロックの検索範囲は、複数のブロックを検索する場合の例を示している。ここで、図3は、V2の変更ブロックデータと同一のブロックデータが、V1の探索区間内に存在する場合の例を示しており、図4は、V2の変更ブロックデータと類似のブロックデータが、V1の探索区間内に存在する場合の例を示しており、図5は、V2の変更ブロックデータと同一又は類似のブロックデータが、V1の探索区間内に存在しない場合の例を示している。

【0039】

図3は、アップグレードパッケージプロセッサ10が、V2プログラムの変更ブロックデータを、設定された区間内で見つけることができる場合のアップグレードパッケージの生成例を説明するための図である。

10

【0040】

図3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、V1及びV2プログラムをマクロブロック単位で比較する。比較過程において、V2プログラムの変更されたマクロブロックデータを、V1プログラムの特定区間内で見つけることができる場合、アップグレードパッケージプロセッサ10は、変更ブロックデータをアップグレードデータとして生成せず、マップデータにV1の対応するブロックインデックス情報を含める。すなわち、アップグレードパッケージプロセッサ10は、V2プログラムの変更ブロックデータと同一のブロックデータを有するV1プログラムのブロックインデックスを見つけた後、これをマップデータに表示する。

20

【0041】

図3は、V2プログラムのブロック16－19が変更され、ブロック20－24がシフトされる例を示している。その場合、アップグレードパッケージプロセッサ10は、V1プログラムの設定区間内において、V2プログラムのブロック16－19のデータと同一の値を有するV1ブロック12、13、8、9のデータを確認することができる。すなわち、アップグレードパッケージプロセッサ10において、V2プログラムのブロック16－17は、V1プログラムのブロック12－13と同一であり、V2プログラムのブロック18－19は、V1プログラムのブロック8－9と同一であることが分かる。図3のようなV1及びV2プログラムを用いて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プロセッサ10は、下記[表1]のようなマップデータを発生することができる。

30

【0042】

【表1】

C:0、15、F:0、null、null

M:16、4 F:1、(16－12)、(17－13)、(18－8)、(19－9)、0、0、0、0

S:20、5、F:0、(20－16)、0

OR

C:0、15、F:0、null、null

M:16、4 F:1、(16－4)、(17－4)、(18－10)、(19－10)、0、0、0、0

S:20、5 F:0、(20－4)、0

40

[表1]のようなマップデータにおいて、V2ブロック0－15は、V1ブロック0－15と同一であり、V2ブロック16－19は、各々V1ブロック12、13、8、9と同一であり、V2ブロック20－24は、V1ブロック16－20と同一であることを意味する。従って、図3のようなV1及びV2プログラムを用いて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プロセッサ10において、特定区間内でV2

50

の変更ブロックを見つけることができる場合は、当該変更ブロックデータをV1の対応するブロックインデックスをマップしてマップデータを生成する。その場合、アップグレードパッケージプロセッサ10は、ヒストリーデータ及び[表1]のようなマップデータを結合し、アップグレードパッケージとして生成する。

【0043】

図4は、アップグレードパッケージプロセッサ10が、V2プログラムの変更ブロックデータを、V1プログラムの設定された区間内で類似のブロックデータを見つけることができる場合において、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である。図4は、V2プログラムのブロック16-19が変更され、ブロック20-24がシフトされる例を示している。

【0044】

図4を参照すると、アップグレードパッケージプロセッサ10において、V2プログラムのブロック16は、V1プログラムの設定区間内のブロック12と同一であるが、ブロック17は、V1プログラムの設定された探索区間内で、一致するブロックを見つけることができず、V2プログラムのブロック18は、V1プログラムの設定された探索区間内で、一致するブロックを見つけることはできないが、ブロック19は、V1プログラムの設定区間内のブロック7と同一であることが分かる。その場合、アップグレードパッケージプロセッサ10において、変更ブロック16及び17は、探索区間内のV1のブロック12及び13と類似のブロックに決定し、変更ブロック18及び19は、探索区間内のV1のブロック6及び7と類似のブロックに決定する。従って、図4のようなV1及びV2プログラムを用いて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プロセッサ10は、下記[表2]のようなマップデータを生成することができる。

【0045】

【表2】

C:0、15、F:0、null、null

M:16、4 F:1、(16-12)、(17-13)、(18-6)、(19-7)、0、code(B、K)、code(B、C)、0

S:20、5、F:0、(20-4)、0

[表2]のようなマップデータにおいて、V2ブロック0-15は、V1ブロック0-15と同一であり、V2ブロック20-24は、V1ブロック16-20と同一であることを意味する。また、マップデータにおいて、V2ブロック16及び19は、各々V1ブロック12及び7と同一であり、V2ブロック17は、V1ブロック13との差の値によってエントロピー符号化(code(B、K))され、V2ブロック18は、V1ブロック6との差の値によってエントロピー符号化(code(B、C))されたことを意味する。従って、図4のようなV1及びV2プログラムを用いて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プロセッサ10において、特定区間内でV2の変更ブロックを見つけることができない場合には、当該変更ブロックデータをV1の対応するブロックインデックスをマップし、マップされたV1及びV2ブロックのデータの差をエントロピー符号化してマップデータを生成する。

【0046】

図5は、アップグレードパッケージプロセッサ10が、V2プログラムの変更ブロックデータを、設定された区間内で見つけることができない場合のアップグレードパッケージの生成例を説明するための図である。

【0047】

アップグレードパッケージプロセッサ10は、V1及びV2プログラムが入力されると、設定区間内において、V1及びV2プログラムを、設定されたサイズのマクロブロック単位で比較する。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

10

20

30

40

50

結果によって各ブロックデータの属性(C、M、S)を把握した後、アップグレードデータを生成する。この時、変更ブロックのデータは、アップグレードデータであり、アップグレードパッケージに含まれるべきデータを意味する。この時、V1及びV2のブロックデータを比較する方法は、V2ブロックデータを、V1の前後の設定区間内で見つけることができるかを検査し、見つけることができなければ、二つのデータの差をそのままエントロピー符号化(entropy coding)してアップグレードデータとして生成する。

【0048】

図5は、ブロック16-19に4つのブロックデータZ、W、P、Xが挿入された場合を仮定している。その場合、アップグレードパッケージプロセッサ10は、設定された区間のブロックデータを検査し、同一の値を有するブロックデータがあるか否かを検査する。この時、設定された区間内に同一の値を有するブロックデータがなければ、当該ブロックの属性をMに設定し、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。すなわち、アップグレードパッケージプロセッサ10において、V2プログラムのブロック16-19は、V1プログラムの設定された区間内、一致するブロックを見つけることができないことが分かる。図5のようなV1及びV2プログラムを用いて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プロセッサ10は、下記[表3]のようなマップデータを発生することができ、Z、W、P、Xをエントロピー符号化してアップグレードデータを発生することができる。

【0049】

【表3】

C:0、15、F:0、null、null
M:16、4、F:2、null、null
S:20、5、F:0、(20-4)、0

[表3]のようなマップデータにおいて、V2ブロック0-15は、V1ブロック0-15と同一であり、V2ブロック20-24は、V1ブロック16-20と同一であることを意味する。更に、マップデータにおいて、V2ブロック16-19は、アップグレードデータのZ、W、P、Xにエントロピー符号化(code(E、C))されたことを意味する。従って、図5のようなV1及びV2プログラムを用いて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プロセッサ10において、特定区間内でV2の変更ブロックを見つけることができない場合は、当該変更ブロックデータのフラグを2(F=2)に設定して対応するV1プログラムのブロックがないことを表示し、別途のアップグレードデータを生成する。その場合、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からなる。

【0050】

また、図5のように、V2の変更ブロックデータが、V1の設定された区間内で見つからない場合、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を生成せず、マップデータのみを用いてアップグレードパッケージを生成することもできる。下記[表4]は、V2の変更ブロックデータをV1の設定された区間内で見つけることができない場合のマップデータの構成例を示している。

【0051】

【表4】

C:0、15、F:0、null、null
M:16、4、F:2、null、Z、W、P、X
S:20、5、F:0、(20-4)、0

10

20

30

40

50

従って、図5のようなV1及びV2プログラムを用いて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プロセッサ10において、特定区間内でV2の変更ブロックを見つけることができない場合は、当該変更ブロックデータをエントロピー符号化してマップデータを生成する。[表4]のように、マップデータを発生する場合、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を生成せず、ヒストリーデータ及び[表4]のようなマップデータを生成してアップグレードパッケージを生成することができる。

【0052】

図6は、V1プログラムが、ギャップ領域を備え、V2プログラムにアップグレードされる時のシフト動作を改善する方法を説明する図である。また、図6及び図7は、V1プログラムの一部のブロックデータが、V2プログラムにおいては、削除される場合のアップグレードパッケージの生成例を説明するための図である。

【0053】

図6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、V1プログラムに基づいてV2プログラムを比較する時、新しいデータが追加又は削除されると、当該領域の次のブロックをシフトさせる。この時、変更ブロックのサイズによって、前へのシフトが発生すると、アップグレードパッケージプロセッサ10は、V1プログラムの設定された区間を探索して変更ブロックと同一のブロックデータがあるか否かを確認し、ある場合には、V1ブロックの番号をV2の変更ブロックにマップさせる。しかし、同一のデータがない場合には、アップグレードパッケージプロセッサ10は、V2の当該ブロックデータをエントロピー符号化してアップグレードデータとして生成する。

【0054】

上記のように、V1プログラムをV2プログラムにアップグレードする場合、V2プログラムは、V1プログラムにおいて、多くのブロックのシフトが発生することがある。従って、上記のようなシフトを考慮して、図6に示すように、V1プログラムにデータを格納しないギャップ(gap)領域を割り当てることができる。ここで、V1のギャップ領域は、V2プログラムのアップグレードデータを考慮して設定することができる。好ましくは、V2プログラムのシフト動作は、図6に示すようなギャップ領域を用いて、その次のコンポーネント(component)に影響を及ぼさないように実行される。例えば、図6において、V2プログラムは、V1プログラムのブロック6-10の5個のブロックデータを必要とせず、ブロック3-4、8-9、13、17-19の8個のブロックが追加され、ブロック15は、同じサイズで代替される(replaced as same size)構造を有する。その場合、V2プログラムは、V1プログラムから5個のブロックが削除され、8個のブロックが追加されるため、図5に示すように、最終ブロックは、3個のブロックのシフトが発生する。この時、最終ブロックのシフトは、V1プログラムのギャップ領域で発生するため、V2プログラムの次のコンポーネントは、シフト動作なしで、最初ブロックからV1プログラムの当該コンポーネントと比較され、アップグレード動作を行うことができる。

【0055】

図7を参照すると、FOTA(Firmware over-the-air)を支援するバイナリは、物理的にギャップ領域を有しており、アップグレード時、各コンポーネント領域間の侵犯を防止する。従って、V1プログラムは、図7のような構造を有することができる。すなわち、V1プログラムは、複数のコンポーネント(図7では、5個のコンポーネントから構成される場合を仮定している)から構成される。従って、図7のようなV1プログラムをアップグレードするV2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、変更されたブロックデータが追加される場合にも、シフトされるブロックをギャップ領域にマップさせることができる。従って、アップグレードパッケージプロセッサ10は、コンポーネント単位でアップグレード過程を実行することができ、従って、各コンポーネント単位で独立的なアップグレードパッケージを生成すること

ができる。

【0056】

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、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から構成されたアップグレードパッケージを生成することができる。その場合、マップデータは、V1及びV2プログラムの差によるブロックの属性(コピー、変更、シフト)及びブロックのインデックス情報を含み、アップグレードデータは、変更ブロックデータにすることができる。また、アップグレードパッケージプロセッサ10は、マップデータ及びヒストリーデータのみからアップグレードパッケージを生成することができる。その場合、アップグレードパッケージプロセッサ10は、マップデータに、V1及びV2プログラムの差によるブロックの属性(コピー、変更、シフト)及びブロックのインデックス情報の他、変更されたデータ(又は、変更ブロックのインデックス情報)を含めることができる。

10

【0057】

図8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である。その場合、アップグレードパッケージプロセッサ10は、マップデータ発生器を備えず、パッケージ発生器130は、図8のようなアップグレードデータを発生する。ここで、アップグレードデータは、V2バージョンの各ブロックのインデックス情報、及びブロックデータ情報を含む構造を有することができる。

20

【0058】

図8を参照すると、V1は、第1バージョンプログラムを示し、V2は、第2バージョンのアップグレードプログラムを表す。V1とV2とを比較すると、V2は、6、7、13、14、15、16、17番のブロックデータが追加され、V1の9、10、11、12番のブロックデータが削除された状態であることがわかる。この時、パッケージ発生器130は、V1及びV2の比較結果を分析し、V2の各ブロックのブロックインデックス(V1のようなブロックマップのためのインデックス)、及びブロックデータの情報を含まなければならない。本発明の実施形態に係るアップグレードデータは、マップデータと類似した構造を有することができる。すなわち、アップグレードデータのコマンドは、C(copy)、M(modify、insert or replace as same size)、S(shift)から構成することができ、各コマンドのフォーマットは、下記のように構成することができる。

30

【0059】

```

コピー(copy format)
[cmd][start block no][number of block]
変更(modify format)
[cmd][start block no][number of block][data]
シフト(shift format)
[cmd][start block no][number of block][previous version position]
```

40

上記のようなフォーマットを有する場合、コピーは、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し、変更は、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し、シフトは、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V1のブロックインデックス等が含まれる。

【0060】

従って、図8の場合、V2のコピーブロックに対するアップグレードデータは、C:0、6にすることができ、変更ブロックに対するアップグレードデータは、M:6、2、X、Y及びM:13、5、A、B、C、D、Eにすることができ、シフトブロックに対する

50

アップグレードデータは、S: 8、3、6、S: 11、2、13、及びS: 18、7、15にすることができる。上記のようなアップグレードデータが伝送される場合、受信側では、第2バージョンであるV2プログラムにアップグレードする時、V2の0番のブロックから6個のブロックは、V1の0番から5番までのブロックをコピーし、6番ブロックから2個のブロックは、X及びYを追加し、13番ブロックから5個のブロックは、順次にA、B、C、D、Eを追加し、8番ブロックから3個のブロックは、V1の6、7、8番のブロックをシフトし、11番ブロックから2個のブロックは、V1の13、14番ブロックをシフトし、18番ブロックから7個のブロックは、各々V1の15、16、17、18、19、20、21番のブロックをシフトすることを命令する意味になる。従って、図8のようなV1及びV2データが入力されると、アップグレードパッケージプロセッサ10は、図8のアップグレードパッケージ(delta package)のアップグレードデータを[表5]のように生成することとなる。

10

【0061】

【表5】

C: 0、6

M: 6、2、X、Y

S: 8、3、6、

S: 11、2、13

M: 13、5、A、B、C、D、E

S: 18、7、15

20

その後、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として生成し、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で生成されるアップグレードパッケージは、圧縮されて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にアップグレードデータを用いてV2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。上記では、圧縮する例を挙げているが、圧縮機を用いずにデータのみからもパッケージの生成が可能である。

30

【0062】

以下、図9～図13の説明においては、アップグレードパッケージプロセッサ10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から構成されるアップグレードパッケージを生成する場合、及びヒストリーデータとアップグレードデータとから構成されるアップグレードパッケージを生成する場合を具体的に説明する。

【0063】

図9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の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生成器の構成を示す図である。

40

【0064】

図10は、図2のアップグレードパッケージプロセッサ10の更に他の構成例を示す図であり、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生成器の構成を示す図である。

【0065】

50

図 11 は、図 9 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 12 は、図 10 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。

【0066】

図 13 は、図 2 のアップグレードパッケージプロセッサ 10 の更に他の構成例を示す図であり、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 の構成を示す図である。ただし、図 10 と類似した方法で、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することによってアップグレードパッケージを生成することもできる。

【0067】

図 9 - 図 12 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 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 9 及び図 10 のように、第 1 圧縮機 160 に入力されて圧縮された形態で入力されるか、又は図 13 のように元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。圧縮された二つのバージョンのプログラムを比較して、二つのブロックデータが同一であるか否かを表示する。ここで、比較器 110 は、排他的論理和 (exclusive OR) 演算回路を用いることができる。二つのバージョンのプログラムの対応するブロックデータが同一でない場合、比較器 110 は、設定された区間内の第 1 バージョンブロックデータを探索し、探索区間で二つのブロックのデータを比較し、一致するか否かを確認する。上記のように、二つのバージョンのブロックデータを探索した後、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。

【0068】

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 バージョンの対応するブロックデータと比較し、同一である場合、比較結果値と共に当該ブロックのインデックスをインストールデータ発生器 180 に伝達する。しかし、第 2 バージョンのブロックデータと第 1 バージョンのブロックデータとが相違すると、比較器 110 は、第 1 バージョンプログラムの設定された探索区間 (search window) 内のブロックデータを第 2 バージョンの現在のブロックデータと比較し、同一のブロックデータがあるか否かを検査する。探索区間は、第 1 バージョンプログラムの現在ブロック位置の前後に位置する、設定された数のブロック（図 3 - 図 8 の場合、前後 15 ブロックで、探索区間は 30 ブロックとなる）データであり、探索方法としては、第 2 バージョンのブロックデータを第 1 バージョンの探索区間内でスイングしながら各ブロックデータと比較する方法を用いることができる。その時、同一のブロックデータがあると、比較器 110 は、比較結果値と共に第 1 バージョンのブロックインデックスをインストールデータ発生器 180 に伝達する。ところが、同一のブロックデータがない場合には、比較器 110 は、隣接している第 1 バージョンブロックのインデックス及びこれによる比較結果値を前記インストールデータ発生器 180 に伝達する。

【0069】

図 3 - 図 5 を参照すると、第 2 バージョンのブロック 0 - 15 を比較する場合、比較器 110 は、第 1 バージョンのブロック 0 - 15 と同一の比較結果値を生成し、なお、比較結果値及び比較された第 1 及び第 2 バージョンのブロックインデックス 0 - 15 を、各々インストールデータ発生器 110 に伝達する。また、第 2 バージョンのブロック 16 - 19 を比較する場合、比較器 110 は、第 1 バージョンのブロック 16 - 19 と相違するデータであることを確認し、設定された区間内の第 1 バージョンのブロックデータと比較を行う。設定された区間が 15 ブロックであると、比較器 110 は、まず第 1 バージョンの

ブロック0-15と比較して同一のブロックデータがあるか否かを確認し、その後、第1バージョン17-31のブロックと比較して同一のブロックデータがあるか否かを確認する。図3において、第2バージョンのブロック16-17(18-19)は、第1バージョンのブロック12-13(ブロック8-9)と同一であることを確認し、比較器110は、比較結果値及び第2バージョンのブロック16-17(18-19)、第1バージョンのブロック12-13(8-9)のインデックスをインストールデータ発生器180に伝達する。また、図4において、第2バージョンのブロック16-17(18-19)は、第1バージョンのブロック12-13(ブロック8-9)と類似していることを確認し、比較器110は、比較結果値及び第2バージョンのブロック16-17(18-19)、第1バージョンのブロック12-13(8-9)インデックスをインストールデータ発生器180に伝達する。ここで、比較器110は、ブロック17及び18は、比較結果値(KとBとの差及びCとBとの差の値)及び比較対象の第1バージョンのブロックインデックス13及び6をインストールデータプロセッサ180に伝達する。すなわち、比較器110において、先のブロックは、同一の比較結果値を有するのに対し、次のブロックが異なる比較結果値を有する場合には、上記のように、二つのブロックの比較結果値及び比較された第1バージョンのブロックインデックスをインストールデータ発生器180に伝達する。なお、図5のように、第2バージョンのブロック16-19が、第1バージョンの設定区間内のブロックにおいて一致しない場合、比較器110は、一致しないことを表す第2バージョンのブロック番号16-19をインストールデータ発生器180に伝達する。

【0070】

インストールデータ発生器180は、図9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とから構成されるか、又は図10に示すように、ヒストリーデータ発生器120のみから構成されてもよい。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である場合、第2バージョンはバージョン番号5となり、第1バージョン及びバージョン番号が5である第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。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、及び比較結果値に応じて[Cmd][start block No, number of block][flag][distance][difference]の構造を有するマップデータを生成する。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おいて同一の比較値が発生すると、マップデータ発生器150は、当該ブロックのコマンドをコピーCと決定し、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するか、又は第1バージョンの削除ブロックが発生するまで続けられる。

【0071】

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おいて異なる比較値を生成すると、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と決定することになる。この時、変更Mは2種類の形態で発生される。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。ブロックインデックスを分析し、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であると判断されると、マップデータ発生器150は、コマンドを変更Mとする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、図3のように、設定された探索区間内で同一のブロックデータが存在する場合、図4のように、設定された探索区間内で類似したブロックデータが存在する場合、又は図5のように、設定された探索区間内で同一のブロックデータが存在しない場合に応じて、それぞれ異なる形態の

変更マップデータを生成することができる。また、図5の場合、[表3]のように、マップデータとアップグレードデータとをそれぞれ独立して生成することができ、又は、[表4]のように、マップデータにアップグレードデータを含めて生成することができる。以下の実施形態では、[表3]のように、マップデータ及びアップグレードデータをそれぞれ生成する場合を仮定して説明する。

【0072】

変更ブロックデータが、同じサイズの元のブロックに代替される場合、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と、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。すなわち、第1バージョンと同一の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されなくなる。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。その場合、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にシフトされなければならない。

【0073】

変更ブロックが挿入される場合、その後のブロックデータはシフトされなければならない。そのため、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、入力される比較器110の出力を分析し、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とを分析する。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、また第1バージョン及び第2バージョンのシフトブロック数を含むマップデータを生成する。

【0074】

上記のような過程でマップデータを生成する場合、マップデータ発生器150は、図3のような第2バージョンが発生すると、[表1]のようなマップデータを生成でき、図4のような第2バージョンが発生すると、[表2]のようなマップデータを生成でき、図5のような第2バージョンが発生すると、[表3]又は[表4]のようなマップデータを生成することができる。以下、説明する実施形態では、図5のような第2バージョンである場合に、マップデータ発生器150は、[表3]のようなマップデータを生成するものとする。

【0075】

パッケージ発生器130は、第1圧縮機160から出力される第2バージョンの圧縮されたブロックデータ、及びマップデータ発生器23でマップデータを分析してアップグレードパッケージを発生する。この時、パッケージ発生器130は、マップデータ発生器150から発生するマップデータを分析し、アップグレードデータを発生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て、コピーC及びシフトSのコマンドを有するブロックである場合には、アップグレードデータを発生しない。また、マップデータのコマンドが変更Mである場合、パッケージ発生器130は、マップデータが当該ブロックのデータを含んでいるか、又は当該ブロックのデータをインデックスする第1バージョンのブロックインデックスを含む場合にも、アップグレードデータを発生しない。しかし、マップデータのコマンドが変更Mであり、マップデータが、第2バージョンの当該ブロックデータを変更するための第1バージョンのインデックス又はエントロピーデータを含まない場合、パッケージ発生器130は、圧縮された第2バージョンのブロックデータを用いてアップグレードデータを生成する。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ると、アップグレードデータを生成しない。また、変更Mコマンドである場合には、フラグを分析し、フラグF=1である場合には、アップグレードデータを生成せず、F2フラグの場合にのみアップグレードデータを生成する。

【0076】

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で発生するイン

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として生成する。この時、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である。すなわち、図9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10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には、インストールデータはヒストリーデータとなる。

【0077】

図10を参照すると、インストールデータ発生器180は、図9とは異なり、マップデータ発生器150を備えていない。パッケージ発生器130は、第2バージョンのブロックインデックス値に対応してマップされる第1バージョンのブロックインデックス値、及び、ブロックデータに関する情報を含むアップグレードデータを生成する。この時、アップグレードデータは、マップデータ発生器150と類似したコマンドを備え、コピーC、変更M、及びシフトSコマンドのフォーマットは、それぞれC: [start block no] [number of block]、M: [start block no] [number of block] [data] 及びS: [start block no] [number of block] [previous version position] のような構成にすることができる。すなわち、アップグレードデータにおいて、V2のブロックデータをV1から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、ブロックデータをV1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、V1のブロックデータを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V1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。

【0078】

次いで、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として生成し、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。パッケージ発生器130で生成される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V2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生成速度を向上させることができる。

【0079】

従って、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アップグレードデータによって構成することができる。それにより、パッケージ発生器130で生成されるアップグレードパッケージは、図9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11のように生成され、また、図10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には、図12のように生成される。

【0080】

その後、パッケージ発生器130から出力される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は省略することができる。しかし、送信効率を高めるために、図9に示すように、第2圧縮機140によって圧縮することもできる。なお、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のかを確認するものであり、圧縮時に誤りが発生した場合、第1圧縮機160を制御し、再圧縮するように制御することができる。

【0081】

図9及び図10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、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。その場合、比較器110は、圧縮された二つのバージョンのプログラムを比較することになる。しかし、比較器110は、第1及び第2バージョンを圧縮していないデータで比較することができる。図13は、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。その場合、図7のような構成を有するアップグレードパッケージプロセッサ10の比較器110は、受信される第1及び第2バージョンプログラムを設定されたサイズのブロックデータに分割し、ブロック単位の第1及び第2バージョンのデータを上記のような方法で探索及び比較する。従って、図13において、第1及び第2バージョンを圧縮する構成を除いた残りの構成は、図9と同一であり、その動作も同一である。また、図13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10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。

10

【0082】

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、比較結果に応じ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生じると、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。すると、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沿って前へシフトされるか、又は後へシフトされるという結果を招く。また、第2バージョンの変更されたブロックは、第1バージョンの探索区間内で同一のブロックが存在するのかを探索し、探索結果に応じて同一のブロックデータを有する第1ブロックのブロックインデックスを指定するか、又は当該ブロックのデータをエントロピー符号化することになる。また、上記のように、変更ブロック以後の第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。それにより、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。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との連結方法により、無線、近距離通信、又は有線通信方法を用いることができる。

20

30

【0083】

すると、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、この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の発生を、各受信装置に通知する通知サーバー（notification server）を更に備えることができる。

40

【0084】

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、それに応答を行い、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。図14は、本発明の実施形態に係る受信装置の構成を示す図である。

【0085】

50

図14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の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、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。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。この時、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すると、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、マップデータを生成するか、又はマップデータを生成しない。インストーラー230で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。また、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させて格納する。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。アップグレードパッケージは、少なくとも一つであって、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。

【0086】

なお、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アップグレードが準備できたことを知らせる。システムアップグレードの要求が発生するか、又はシステムの初期化動作を行なった後、受信装置30のトランスレータ(translator)240は、第1メモリ250で第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てから、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バージョンであってもよい)の当該ブロック内容をアップグレードする。また、インストールデータにマップデータが含まれていない場合、トランスレータ240は、ヒストリーデータを分析して、結合する第1バージョンを決定し、アップグレードデータによってアップグレードパッケージを第1バージョンに結合させ、第2バージョンのプログラムを生成することもできる。上記のような動作を行いながら第1バージョンを第2バージョンにアップグレードすると、トランスレータ240は、それを第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。

【0087】

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。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、又はインストールデータ(ヒストリー及びマップデータ)のみから構成されてもよい。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のマップ関連情報であるマップデータとから構成されることができる。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)の比較を通じてCMS(copy、modify、shift

）を探す生成方法であり、迅速なアドレス計算を目的に用いられ、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バージョンプログラムを迅速に生成するためのデータとして用いられる。

【0088】

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で生成されるが、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、又は生成されない場合もある。それにより、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、あるいは含まれない場合もある。——従って、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、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。そのため、インストーラー230は、アップグレードデータを分析してマップデータを生成する。アップグレードデータは、図8に示すような構造を有することができる。インストーラー230が、アップグレードデータからマップデータを生成するのは、トランスレータ240において、第2バージョンのプログラムを生成する速度を向上させるためである。この時、アップグレードデータが、第1バージョンのブロックとマップするための情報を含んでいる場合、アップグレードデータにマップデータを生成してインストールすることができ、アップグレードデータをそのままインストールすることもできる。

【0089】

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てアップグレードするのが一般的だが、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、本発明の実施形態においては、第2バージョンに生成するアップグレードパッケージは、上記のように、最大6個まで格納できると仮定する。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、ユーザーに現在のシステムが保有している再組合せの可能なアップグレードパッケージリストを表示し、ユーザーの選択に応じて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。

【0090】

第1メモリ250は、複数(ここでは、最大6個と仮定する)のアップグレードパッケージを格納できる格納領域を備える。従って、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され、第1メモリ250の格納領域に格納されて、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に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。従って、バージョンの変更によるヒストリーが累積されて格納され、第1バージョンの格納状態をそのまま保持する。従って、第1及び第2バージョンがそのまま保持されるため、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されない場合、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用いることができ、全てのバージョンのアップグレードパッケージロードが失敗するという最悪の場合でも、オリジナルバージョンの第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。

【0091】

図15は、本発明の実施形態に係る受信装置の第1メモリ250の構造を示す図である。図15において、参照番号310は、第1バージョンを格納する第1格納領域であって、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することができる。参照番号320は、少なくとも一つの第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格納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、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。従って、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。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。

【0092】

図16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であり、図17は、図16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である。

【0093】

図16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個に仮定している)のアップグレードパッケージを格納できる格納領域を備え、各アップグレードパッケージの格納領域に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で構成される。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9及び図10を参照)、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よって構成することができる。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を格納することができる。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。また、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、また、その代わりにナルデータ(null data)を入れることもできる。図16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。また、アップグレードパッケージプロセッサ10において、上記のように、ヒストリーデータ及びマップデータから構成されるアップグレードパッケージを生成する場合、第2格納領域320の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及びマップデータを格納する構造に構成することもできる。

【0094】

図17を参照すると、ヒストリーデータの構造は、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)等から構成することができる。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16の場合、#2-#7中のいずれか一つ)にすることができ、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、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)にすることができ、失敗フラグ(fail flag)は、ロード中、失敗(fail)が発生したかを表示するためのフラグである。この時、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V#1が格納された場合、受信装置30は、#5の第2バージョンを#1の第1バージョンに結合(merge)して、アップグレードすることとなる。ダウンロードされるアップグレードパッケージは、図16のような構造で図15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に、第1格納領域310の第1バージョンに結合され、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こととなる。

【0095】

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。

【0096】

図18は、第1メモリ250及び第2メモリ260の構造を示す図である。

【0097】

図18を参照すると、アップグレード要求時、図示していないローダーは、第2格納領域320で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バージョンとマージして第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、アップグレードの要求は、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、システム初期化を行った後、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、アップグレードをするか否か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するとき、上記のような動作を行い、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

【0098】

なお、上記のようなアップグレード動作は、受信装置30の動作を初期化した後、行うことができる。すなわち、図15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る時点で、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。

20

【0099】

図19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であり、図20及び図21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図である。

【0100】

図19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19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、第1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構成することができる。ここで、ヒストリーデータは、上記のように、バージョン領域に $\#1$ が格納され、結合フラグ領域に $\#1$ が格納される。また、マップデータ領域には、[表1]ないし[表3]のような構造を有するマップデータが格納される。

30

【0101】

ここで、アップグレード要求コマンドが発生すると、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19のように、第2バージョンのアップグレードパッケージが存在しない場合、マップデータは、マップデータをnull状態で備えることができ、また、 $\{C:0, n, F:0, null, null\}$ 構造のマップデータを備えることもできる。上記のようなマップデータは、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バージョンを第2メモリ260に、そのままコピーする。すると、受信装置30は、第2メモリ260に、図19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30の動作を制御する。第1メモリ250に格納される第1バージョンは、圧縮されて格納される。従って、第1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、第1圧縮解除機270を介し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を介し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

40

50

240は、まずロードされた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行うことができる。

【0102】

図20及び図21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、基準と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することができる。インストールデータは、上記のように、対応する第2バージョンの番号、及び結合するバージョン(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することができる。そして、第1バージョンは、図20及び図21に示すように、n個のブロック(B#1-B#n)からなる。ここで、ヒストリーデータは、結合フラグ領域に、#0が格納されており、マップデータ領域には、[表1](図20の場合)、又は[表3](図21の場合)のようなマップデータを格納することができる。

10

【0103】

まず、図20を参照すると、第1メモリ250の第1領域310は、第1バージョンのプログラムを格納し、第1メモリ250の第2領域320の特定アップグレードパッケージは、[表1]のようなマップデータを格納していると仮定する。ここで、アップグレードパッケージのヒストリーデータは、第1バージョンのプログラムと結合するためのフラグ情報を格納している。その場合、トランスレータ240は、マップデータによって第1バージョンプログラムをアップグレードして、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。この時、マップデータは、0番ブロックから15番ブロックまでは、第1バージョンのブロックをコピーし、16番目のブロックから19番目のブロックまでの4個のブロックは、各々第1バージョンの12、13、8、9番目のブロックをコピーして挿入し、20番目のブロックから最後のブロック(前の20個のブロックを除く全てのブロック)は、4個ずつシフトするというコマンドを含んでいる。従って、トランスレータ240は、図21の第2メモリ26に示すように、マップデータに応じて第1バージョンプログラムをアップグレードする。

20

【0104】

ここでは、ブロック間にギャップがない場合を説明したが、ギャップがある場合は、ギャップまでのみシフトする。

30

【0105】

次に、図21を参照すると、第1メモリ250の第1領域310は、第1バージョンのプログラムを格納し、第1メモリ250の第2領域320の特定アップグレードパッケージは、[表3]のようなマップデータを格納していると仮定する。ここで、アップグレードパッケージの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フラグ情報を格納している。マップデータは、0番ブロックから15番ブロックまでは、第1バージョンのブロックをコピーし、16番目ブロックから19番目ブロックまでの4個のブロックには、各々アップグレードパッケージのアップグレードデータ{Z, W, P, X}を挿入し、20番目のブロックから最後のブロック(前の20個のブロックを除く残り全てのブロック)は、4個ずつシフトさせるというコマンドを含んでいる。従って、トランスレータ240は、図21の第2メモリ26に示すように、マップデータに応じて第1バージョンプログラムをアップグレードする。

40

【0106】

その後、受信装置30は、第2メモリ260に図20及び図21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、伝送時に圧縮して格納することができる(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を用いて圧縮する場合)。従って

50

、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、圧縮解除機270を介して圧縮された第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、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になる。その場合、トランスレータ240は、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。

【0107】

図22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、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図である。

【0108】

図22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を第1バージョンに結合して第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。従って、受信装置30は、第1メモリ250に格納している第1バージョンのプログラム及び第2バージョン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り繰り返して行なわれる。

【0109】

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めら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メモリに格納し、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メモリに格納した後、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、以下に説明する本発明の実施形態においては、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、迅速な実行のための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において、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。

【0110】

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について説明すると、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、この二つのバージョンを、アップグレードパッケージの生成時に、圧縮するか、又は、圧縮していない状態で比較した後、二つのバージョン間の差異点を見出す。ここで、二つのバージョン間の差異点は、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、マップデータにすることができる)として生成される。また、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として生成する。

【0111】

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30にダウンロードされる。ここで、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した後、迅速な実行のための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のために使われる。すなわち、第1

10

20

30

40

50

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。

【0112】

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について説明すると、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、迅速なアドレス計算のために使われ、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 で結合することを助ける役割として使用される。

【0113】

上記のインストール方法は、マップデータが存在するか否かによって異なる形態になる。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る場合、受信装置 30 は、アップグレードパッケージにおいて、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分類した後、これらを各々第 1 メモリ 250 のアップグレードパッケージ領域の割り当てられた格納領域に格納する。

【0114】

しかし、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。この時、アップグレードデータは、図 8 のように、マップ情報を含むことができる。その場合、受信装置 30 は、インストール段階において、アップグレードデータからマップ情報を抽出してマップデータを生成した後、これをアップグレードパッケージのマップデータ領域にインストールし、アップグレードデータを抽出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領域にインストールする。その場合、受信装置 30 は、アップグレードパッケージにマップデータが含まれた場合と同じ形態でアップグレードパッケージをインストールすることができる。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。ここで、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 250 に保管することを仮定して説明しており、特定アップグレードパッケージについて、第 2 メモリ 260 において結合失敗が発生した場合、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。

【0115】

アップグレードパッケージの保管及び処理方法について説明すると、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の格納領域に保管され、格納領域は複数備えられる。従って、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ても、以前のアップグレードパッケージは削除されない。また、特定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、ユーザーが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。最悪の場合、全てのアップグレードパッケージのロードが失敗した場合には、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。

【0116】

図 23 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 23 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。

【0117】

図 23 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、ステップ 411 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

10

20

30

40

50

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た場合に生成され、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(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。ここで、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することのできる基準バージョンとなる。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となる。また、第2バージョンのアップグレードパッケージは、第1バージョンと結合されてアップグレードされる第2バージョンのプログラムを生成する情報であって、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となる。従って、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、ステップ413で二つのバージョンの差を分析し、ステップ415で二つのバージョンの差によるアップグレードパッケージを生成する。ここで、アップグレードパッケージは、二つのバージョンの差に応じて、第1バージョンにアップグレードする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。また、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。この時、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、その場合、マップデータは、受信装置30において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。従って、インストールデータは、プログラムのアップグレード時、アップグレードバージョンプログラムを速かに生成するための情報として使われる。

10

20

【0118】

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。また、アップグレードパッケージの発生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。ここで、そのダウンロード方法は、受信装置30の通信方法によって決定される。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、無線通信方式に応じてCDMA、UMTS、GSM等のような無線通信方式でダウンロードされ、また、インターネットに接続されている場合には、USB、ブルートゥース、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ができる。

30

【0119】

受信装置30は、ステップ451でアップグレードパッケージをダウンロードした後、ステップ453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する第2領域320から構成される。第2領域320は、複数の格納領域から構成され、複数の第2バージョン情報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ことと仮定している。また、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段階において、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抽出することができる。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、ステップ455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

40

50

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、その場合、第2メモリ260は、揮発性メモリに具現することができる。従って、受信装置30は、システム初期化段階において、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る第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システムを運用する。また、受信装置30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的に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30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実行することができる。また、システムの初期化時、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、ユーザーに選択されたバージョンのプログラムをアップグレードすることもできる。

10

【0120】

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。

【0121】

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。

【0122】

図24-図26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。

20

【0123】

図24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、ステップ501で第1及び第2バージョンをロードする。その後、アップグレードパッケージプロセッサ10は、ステップ503で第1及び第2バージョンのプログラムを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。また、アップグレードパッケージプロセッサ10は、ステップ505及びステップ507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。ここで、アップグレードパッケージプロセッサ10は、各々パッケージ発生器130及びインストール発生器180を各々備える場合、図24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。また、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で生成することができ、また、インストールデータ-アップグレードデータの順で生成することもできる。

30

【0124】

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)するために用いるデータであって、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。ここで、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)である。ここで、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で生成することができ、また、受信側でも生成することができる。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、又は、含めないこともできる。

40

【0125】

従って、アップグレードパッケージプロセッサ10は、図25のように、インストールデータの生成時、マップデータを生成することも、又は、生成しないこともできる。図2

50

6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)する時に、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25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、ステップ521でヒストリーデータを生成した後、ステップ523でマップデータ生成が必要であるかを検査する。ここで、マップデータ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、ステップ525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24にリターンする。また、図26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、ステップ531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、マップデータも一緒にマージするかを検査する。この時、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、ステップ533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する。また、マップデータをマージしない場合には、ステップ535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。ここで、マップデータは、図3-図5、及び[表1]-[表4]に示すような方法で生成することができる。

10

【0126】

また、図24において、ステップ501でロードされた第1及び第2バージョン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を介して行なうことができる)、ステップ503へ進行することができ、また、ステップ509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を介して行なうことができる)。ここで、第1及び第2バージョンプログラムの圧縮符号化動作を行なう場合、ステップ503で比較するバージョンのサイズが小さくなってデータ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。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されたかを確認し、正常に圧縮符号化された場合にのみ次の段階へ進行するように制御することができる。

20

【0127】

図27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り、図28は、図27における第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29は、図27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であり、図30は、図27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である。

30

【0128】

図27-図30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、ステップ551で構成ファイル(configuration file)を入力する。この時、構成ファイルは、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を含む。ここで、フラグ中、C_FLAGは、圧縮の有無を設定するフラグ(compression flag)であり、M_FLAGは、マップデータの生成の有無を決定するフラグ(mapping 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、ステップ553で第1及び第2バージョンをロードする。ここで、第1バージョンは、上記のように、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にすることができ、第2バージョンは、最近のアップグレードバージョンにすることができる。

40

【0129】

二つのバージョンをロードした後、アップグレードパッケージプロセッサ10は、ステップ555で二つのバージョンの圧縮符号化が必要であるか(C_FLAG=1)を検査する。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、ステッ

50

ブ 5 6 1 で第 1 及び第 2 バージョン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、ステップ 5 6 3 へ進行して二つのバージョンの比較を行なう。しかし、ステップ 5 5 5 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 1 0 は、ステップ 5 5 7 で第 1 圧縮機 1 6 0 を実行した後、ステップ 5 5 9 で第 1 及び第 2 バージョンプログラムを第 1 圧縮機 1 6 0 を介して圧縮符号化した後、ステップ 5 6 3 で圧縮された二つのバージョンのプログラムを比較器 1 1 0 を介して比較する。

【0 1 3 0】

ステップ 5 5 9 において、二つのバージョンを圧縮する過程は、図 2 8 のような過程で行なわれる。ここで、圧縮フラグが設定されている場合には、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_F L A G)がセットされると、アップグ

10

【0 1 3 1】

図 2 8 を参照すると、アップグレードパッケージプロセッサ 1 0 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、ステップ 6 0 1 で検証が必要なのか(V_F L A G = 1)を検査する。ここで、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 1 0 は、ステップ 6 0 3 で第 1 圧縮解除機 1 6 5 を実行させて圧縮された第 1 及び第 2 バージョンの圧縮を解除した後、ステップ 6 0 5 で圧縮前のバージョンと圧縮解除した後のバージョンとを比較し、ステップ 6 0 7 で比較の結果、圧縮前及び圧縮解除後のバージョンが同一の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 1 0 は、ステップ 6 0 7 でこれを確認し、ステップ 6 0 9 で圧縮されたバージョンを比較器 1 1 0 に送ると共に、正常状態であることを検証する通知を生成する。しかし、ステップ 6 0 7 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 1 0 は、ステップ 6 1 1 でこれをエラー処理した後、リターンする。その場合、アップグレードパッケージプロセッサ 1 0 は、第 1 及び第 2 バージョンプログラムを圧縮する過程を再実行することができる。

20

【0 1 3 2】

図 2 7 に戻り、上記のように、圧縮された、又は、圧縮されていない第 1 バージョン及び第 2 バージョンが発生すると、アップグレードパッケージプロセッサ 1 0 は、ステップ 5 6 3 で比較器 1 1 0 を実行させた後、ステップ 5 6 5 でインストールデータ発生器 1 8 0 を実行させる。この時、インストールデータ発生器 1 8 0 は、マップデータを生成することも、又は、生成しないこともできる。これは、M_F L A G 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 1 0 は、ステップ 5 6 7 でマップデータを生成するべきか(M_F L A G = 1)を確認する。マップデータを生成する場合(M_F L A G = 1)、アップグレードパッケージプロセッサ 1 0 は、ステップ 5 6 7 でこれを感知して、ステップ 5 6 9 で第 1 バージョンと第 2 バージョンとを比較器 1 1 0 を介して比較する。この時、第 1 及び第 2 バージョンの比較は、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断する。すなわち、比較器 1 1 0 は、図 3 - 図 5 に示すように、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。

30

40

【0 1 3 3】

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、又は、圧縮されていないバージョンである。また、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 1 0 は、ステップ 5 7 1 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生成器 1 8 0 に伝達すると共に、ステップ 5 7 7 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良い場合(M_F L A G = 0)、アップグレ

50

ードパッケージプロセッサ10は、ステップ567でこれを感じ知し、ステップ575で第1バージョンと第2バージョンとを比較器110を介して比較する。ここで、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、ステップ577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、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生成器180に伝達すると共に、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することができるようにし、マップデータを生成しない場合、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することができるようにする。 10

【0134】

その後、アップグレードパッケージプロセッサ10は、ステップ579でインストールデータ生成器180を駆動させてインストールデータを生成する。ここで、インストールデータの生成は、図29のような過程で進行される。

【0135】

図29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、ステップ651でインストールデータの生成を活性化した後、ステップ653で二つのバージョンのヒストリー情報を検査する。その後、アップグレードパッケージプロセッサ10は、ステップ655で、ヒストリーデータ発生器120を駆動し、ステップ657でヒストリーデータを生成する。この時、ヒストリーデータは、図示のように、ヘッダー(header)、第1バージョンの情報(first input version info)、第2バージョンの情報(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バージョンのプログラムは、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。従って、ヒストリーデータは、アップグレード時、基準となるバージョン(ここでは、第1バージョンかつオリジナルバージョンである)と基準バージョンで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)の識別情報とが、ヒストリーデータに含まれる。 20 30

【0136】

その後、アップグレードパッケージプロセッサ10は、ステップ659でマップデータを生成するべきか(M_FLAG=1)を確認する。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、ステップ603でマップデータ発生器150を駆動した後、ステップ665でマップデータを生成する。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等であり、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応じて生成されるが、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、二つのバージョンのブロックデータ値が異なっているか、あるいは、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っている場合には、同じ位置までの移動Sに設定する。従って、図3-図6に示すように、マップデータ発生過程は、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値を分析し、第1バージョンと差のあるブロックのインデックス及びデータをマップデータとして生成する。この時、マップデータは、上記のように、[Cmd][start block No, number of block][flag][distance][difference]の形態で生成することができ、第1バージョン及び第2バージョンが、図3-図5のような場合には、[表1]-[表3]のように、マップデータを生成することができる。そのため、上記のようなマップデータを生成することとなり、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 40 50

として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、ステップ667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として生成する。

【0137】

ステップ661まで実行すると、アップグレードパッケージプロセッサ10は、M_F L A Gの設定の有無に応じてインストールデータを生成することになるが、ここで、生成す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することができる。

【0138】

図27に戻り、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、ステップ581でパッケージ発生器を実行させ、ステップ583でパッケージ発生器によってアップグレードパッケージを生成する。図30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。

【0139】

図30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、ステップ621で比較結果によるアップグレードデータを発生する。ここで、パッケージ発生器130は、第1バージョンと第2バージョンとにおいて、比較結果値が、設定された探索区間内から同一のブロックデータを見つけることができなかった場合、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。また、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。本発明の実施形態では、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から構成された場合を仮定して説明しているが、上記のように、ヒストリーデータ及びマップデータのみから構成することもできる。インストールデータのみからアップグレードパッケージを構成する場合、アップグレードパッケージプロセッサ10は、マップデータに、第1バージョンと差のある第2バージョンの各ブロックのマップ情報、及び差のあるブロックデータを含めて発生することができる。

【0140】

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。その場合、アップグレードデータは、図8に示すように、コピーC、変更M、及びシフトSのコマンドフォーマットとして生成することができる。すなわち、第1バージョンと同じブロック番号のデータをコピーする場合、アップグレードデータは、コピーする第1バージョンの開始ブロック番号及びコピーするブロック数を含むコマンドとして生成される。また、第1バージョンにブロックデータを追加する場合、アップグレードデータは、追加するブロック番号及び追加変更するブロック数、及び各ブロックのデータを含むコマンドが生成される。最後に、第1バージョンにシフトされるブロックデータの場合、アップグレードデータは、開始ブロックの番号、シフトするブロックの数、及びシフトされる一番目のブロック番号にマップされる第1バージョンのブロック番号等が含まれる。上記のような構造のアップグレードデータを生成すると、アップグレードデータは、マップ情報を含むようになる。

【0141】

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。従って、アップグレードパッケージプロセッサ10は、ステップ623で圧縮機を駆動した後、ステップ625でアップグレードデータを圧縮し、それから、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、ステップ627及びステップ629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、検証する。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、ステップ631でこれを感知し、ステップ633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、ステップ635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、ステップ 631 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、ステップ 637 で、エラー処理を行う。

【0142】

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。

10

【0143】

図 31 は、本発明の実施形態におけ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過程を示すフロー図である。

【0144】

図 31 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、ステップ 711 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 間の通信方法は、受信装置 30 とアップグレードパッケージサーバー 20 間の連結方法によって決定される。すなわち、受信装置 30 が、携帯端末機の場合、受信装置 30 は、空中無線通信(例えば、CDMA, 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、あるい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なうことができる。ここで、携帯端末機が、無線通信機能を備える場合、インターネットは、無線インターネット(wibro, wimax, wifi, 無線LAN等)を使用することもできる。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、ステップ 713 でこれを受信した後、ステップ 723 でダウンロードの許可を通知する。その後、受信装置 30 が、これに対して応答すると、アップグレードパッケージサーバー 20 は、ステップ 717 でこれを受信し、ステップ 719 で管理情報(management info)を当該受信装置 30 に伝送する。その後、受信装置 30 が、これに対して応答すると、アップグレードパッケージサーバー 20 は、ステップ 723 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サーバー 20 は、ステップ 725 で受信装置 30 からダウンロード完了データを受信すると、ステップ 727 で伝送の終了情報を受信装置 30 に伝送し、ステップ 729 で応答メッセージを受信すると、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。

20

30

【0145】

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 から伝達されるアップグレードパッケージを複数の受信装置 30 に通知し、通知に応じ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。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。

40

【0146】

図 32 は、本発明の実施形態における受信装置 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。

50

【0147】

図32を参照すると、受信装置30は、ステップ801で既に第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、システムの製造時に、第1メモリ250にインストールすることができ、また、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情報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31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ことになる。従って、受信装置30は、ステップ803ーステップ807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、一時格納する。また、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にインストールされるが、ここで、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実行することができ、また、受信装置30の正常な動作モードを実行した後にインストールすることもできる。従って、アップグレードパッケージのダウンロード後に、インストールコマンドが発生しない場合、受信装置30は、ステップ809でこれを感知し、ステップ811で受信装置30の正常モードへ戻る。上記のような状態で、正常モードを実行した後、インストールコマンドが発生すると、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ことになる。

10

【0148】

従って、ステップ809でインストールコマンドを感知すると、受信装置30は、ステップ813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。ここで、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に、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は、図15、図16ー図17のようなバージョンを格納する第1及び第2領域で割り当てることができる。従って、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、ステップ813で該当するアップグレードパッケージを格納するバッファ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は別に独立的に格納されるため、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、従って、バージョンのアップグレードヒストリーを全て格納している状態になる。

20

30

【0149】

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、システム初期化コマンドが発生しないと、受信装置30は、ステップ823でこれを感知し、ステップ817で正常モードを実行する。その場合、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実行されていない状態であるため、旧バージョンで実行されることになる。

【0150】

しかし、ステップ815でシステム初期化コマンドが発生すると、受信装置30は、ステップ821でシステム初期化動作を実行し、初期化動作の実行後、受信されたアップ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、ステップ823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。ここで、受信装置30は、ステップ825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。アップグレードパッケージの状態に異状があると、受信装置30は、ステップ833でアップグレードバージョンの旧バージョンを第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

40

50

アップグレードパッケージの状態が正常であると、受信装置 30 は、ステップ 825 でこれを感じし、ステップ 827 で第 1 メモリ 250 にインストールされた該当するアップグレードパッケージをロードし、ステップ 829 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、ステップ 831 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。

【0151】

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 33 は、本発明の実施形態における受信装置 30 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 250 にインストールする過程を示すフロー図であり、図 34 は、本発明の実施形態における受信装置 30 で第 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、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。

【0152】

図 33 を参照すると、ダウンロードの必要時、受信装置 30 は、ステップ 841 でダウンローダー(down loader)を実行させ、ステップ 843 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。この時、アップグレードパッケージのダウンロード方法は、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場合、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である場合に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、当該有線通信方法、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。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、ステップ 845 でこれを感じし、ステップ 847 及びステップ 849 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実行する。

【0153】

ステップ 845 で正常にダウンロードが完了した場合、受信装置 30 は、ステップ 851 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、ステップ 853 からステップ 857 から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 の当該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。

【0154】

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かを検査する。この時、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、ステップ 859 でこれを感じし、ステップ 875 及びステップ 877 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、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、ステップ 879 で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。従って、ステップ 881 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バッファ内に、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。

【0155】

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、ステップ859でこれを感じし、ステップ861及びステップ863で圧縮解除機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、ステップ865〜ステップ869を実行し、圧縮解除されたアップグレードデータをパージングした後、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、ステップ871及びステップ873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バッファ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。また、ステップ863で圧縮解除したアップグレードデータがマップデータの情報を含む場合、インストーラ230は、マップデータを生成しないこともある。すなわち、アップグレードデータが、図8のように、マップデータを含む場合、アップグレードデータからマップデータを生成することができる。また、マップデータを生成する過程を省略することができる。図8の場合、アップグレードデータは、第2バージョンプログラムの各ブロックに対するコマンド(コピー、変更、シフト)及び当該ブロックのインデックス情報を含んでおり、従って、アップグレードデータを用いてマップデータ及びアップグレードデータを分離して生成することができる。上記のように、アップグレードデータでマップデータを生成すると、受信装置30は、インストールデータ及びアップグレードデータを用いてアップグレードプログラムを生成することになる。また、アップグレードデータがマップデータを含んでいるため、マップデータを生成しないこともある。その場合、ステップ861〜ステップ873の動作は、省略することができ、アップグレードプログラムを生成する時、トランスレータは、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データを第1バージョンのプログラムに結合することができる。本発明の実施形態では、アップグレードデータがマップ情報を含む場合、インストーラは、アップグレードデータからマップデータを生成してインストールすると仮定する。その場合、アップグレードパッケージは、同一の形態の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、アップグレードパッケージ領域にインストールされ、トランスレータは、同一のプログラムでアップグレードプログラムを生成することができる。

【0156】

図33に示すように、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。従って、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は、第1メモリ250に格納し、基準バージョンである第1バージョン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[バッファ]格納領域にインストールして格納する。

【0157】

図34は、受信装置30から図33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。

【0158】

図34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、システムの初期化時、ハードウェア等の初期化を実行した後、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)を組み合わせるアップ

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できる全ての情報を含む。

【0159】

システム初期化動作が実行されると、受信装置30は、ステップ881ーステップ883を実行し、システム初期化を行なった後、第1メモリ250で第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、ステップ884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。ここで、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、ステップ884でこれを感知し、ステップ885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、ステップ886で第1バージョンのセキュリティー検査を行なう。その後、受信装置30は、ステップ887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、ステップ888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、ステップ889で第1バージョンを第2メモリに格納し、ステップ890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、ステップ887で第1バージョンが圧縮されていないものと判定されると、受信装置30は、ステップ889及びステップ890を実行し、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。

【0160】

しかし、ステップ884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、ステップ891でトランスレータ240をロードした後、ステップ892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、ステップ893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、ステップ894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バージョンであるか否かを確認し、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、ステップ895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否かを検査し、圧縮された状態であると、ステップ896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、ステップ897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、ステップ890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。ここで、ステップ897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、アップグレードする時、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、ステップ895で第1バージョンが圧縮されていないと判定されると、受信装置30は、ステップ897及びステップ890を実行し、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。

【0161】

図35ー図38は、図34と同様に、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。

【0162】

図35を参照すると、受信装置30は、ステップ901ーステップ905を実行し、シ

システムの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、ステップ907及びステップ909を実行し、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否かを検査する。ここで、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と、受信装置30は、ステップ911へ進行し、トランスレータ240をロードした後、図36のステップ921ーステップ929を実行し、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。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、ステップ922でこれを感知し、ステップ923ーステップ927を実行し、第1バージョンV1の最後のデータ(EOD: end of data)までトランスレートし、第2メモリ260に格納する。第1バージョンV1が圧縮されたデータでない場合には、受信装置30は、ステップ926及びステップ927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、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、ステップ928で第2メモリ260に格納されたバージョン(ここでは、システム全体のソフトウェア情報にすることができる)を検証した後、ステップ929でこれを用いてシステムの全体的動作を実行させる。

【0163】

しかし、図35のステップ909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、ステップ913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し、ステップ915でヒストリーデータ内の失敗フラグ(fail flag)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し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(fail flag = true)されたという意味は、当該アップグレードパッケージに異状があったことを表している。従って、受信装置30は、ステップ917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37のステップ931ーステップ946を実行しながらシステムアップグレード動作を行ない、失敗フラグがセットされた状態であると、図38のステップ951ーステップ969を実行し、システムアップグレード動作を行なう。

【0164】

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37を参照して説明すると、受信装置30は、ステップ931ーステップ937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ない、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。第1バージョンV1が圧縮された場合、受信装置30は、ステップ939及びステップ941を実行しながら第1バージョンV1プログラムの圧縮を解除し、ステップ942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をアップグレードし、アップグレードされるバージョンを第2メモリ260に格納する。上記のような動作では、ステップ943及びステップ944を繰り返し行ないながら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。また、ステップ938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、ステップ938でこれを感知し、ステップ940、ステップ943、及びステップ944を行いながら第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させ、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、ステップ945でこれを検証した後、ステップ946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を用いてシステムを駆動する。

【0165】

ここで、受信装置30が、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させて、アップグレードされたプログラムを生成する過程を説明すると、アップグレードパッケージがマップデータを含む場合、マップデータは、[表1]～[表4]のような構造を有することができる。その場合、受信装置30のトランスレータ240は、[表1]、[表2]、及び[表4]のようなマップデータである場合、第1バージョンV1にアップグレードパッケージのマップデータを結合して第2バージョンV2のプログラムを生成する。すなわち、[表1]のようなマップデータ{C:0, 15、F:0、null、null}、{M:16, 4 F:1、(16-12)、(17-13)、(18-8)、(19-9)、0、0、0、0}、{S:20, 5、F:0、(20-4)、0}であると、C、M、Sコマンドによってマップデータに含まれたブロックインデックスに対応する第1バージョンV1のブロックデータを、対応する第2バージョンV2のブロックデータとして格納する。その場合、受信装置30は、第2バージョンV2の0番から15番ブロックまでは、第1バージョンV1のブロック0番から15番までのブロックデータを格納し、ブロック16から19番ブロックまでは、第1バージョンV1のブロック12, 13, 8, 9番のブロックデータを各々格納し、ブロック20番から24番までのブロックは、第1バージョンV1のブロック16番から20番までのブロックデータを格納する。

【0166】

[表2]のようなマップデータ{C:0, 15、F:0、null、null}、{M:16, 4 F:1、(16-12)、(17-13)、(18-8)、(19-9)、0、code(B, K)、code(E, C)、0}、{S:20, 5、F:0、(20-4)、0}である場合、受信装置30は、C及びSコマンドに係る第2バージョンV2のブロックは、マップデータに対応するブロックインデックスの第1バージョンV1のブロックデータに対応する第2バージョンのブロックデータとして格納する。また、Mコマンドに係る第2バージョンのブロックは、マップデータに含まれたブロックインデックス及びエントロピー符号化されたデータを用いてブロックデータを生成する。その場合、受信装置30は、第2バージョンV2の0番から15番までのブロックは、第1バージョンV1のブロック0番から15番までのブロックデータを格納し、ブロック16から19番までのブロックは、第1バージョンV1のブロック12, code(B, K), code(E, C), 9番ブロックデータを各々格納し、ブロック20番から24番までのブロックは、第1バージョンV1のブロック16番から20番までのブロックデータを格納する。ここで、コード(B, K)は、第1バージョンV1の13番ブロックデータと第2バージョンV2の17番ブロックデータとの差をエントロピー符号化したデータを意味し、コード(E, C)は、第1バージョンの8番のブロックデータと第2バージョンV2の18番ブロックデータとの差をエントロピー符号化したデータを意味する。

【0167】

また、[表3]のようなマップデータ{C:0, 15、F:0、null、null}、{M:16, 4、F:2、null、null}、{S:20, 5、F:0、(20-4)、0}である場合、受信装置30は、C及びSコマンドに係る第2バージョンV2のブロックは、マップデータに対応するブロックインデックスの第1バージョンV1のブロックデータに対応する第2バージョンのブロックデータとして格納する。Mコマンドに係る第2バージョンのブロックは、マップデータに含まれたブロックインデックス値及びアップグレードデータの数を確認し、これをアップグレードデータでアクセスしてブロックデータを生成する。その場合、受信装置30は、第2バージョンV2の0番から15番までのブロックは、第1バージョンV1のブロック0番から15番までのブロックデータを格納し、ブロック16から19番までのブロックは、アップグレードデータで4個のブロックデータを順次にアクセスして各々対応するブロックインデックスのブロックデータで格納する。その場合、第2バージョンV2のブロック16, 17, 18, 19のブロックデータは、Z, W, P, Xにすることができる。

【0168】

10

20

30

40

50

また、マップデータがなく、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アップグレードパッケージが受信された場合、受信装置 30 は、アップグレードデータ及び第 1 バージョン V 1 のプログラムを比較分析してマップデータを生成することができ、また、マップデータの生成を省略することもできる。本発明の実施形態では、インストーラーが、アップグレードデータのマップ情報を用いてマップデータを生成すると仮定する。例えば、アップグレードデータが、[表 5]のように、第 1 バージョン V 1 のブロックインデックス情報{C: 0, 6}, {M: 6, 2, X, Y}, {S: 8, 3, 6, } {S: 11, 2, 13}, {M: 13, 5, A, B, C, D, E}, {S: 18, 7, 15}を含んでいる場合、インストーラー 230 は、コピー及びシフトブロックのマップ情報は、のような形態のマップデータに生成し、変更ブロックのマップ情報は、設定された探索区間を確認してマップ情報及び/又はアップグレードデータを生成する。また、上記のように生成されるアップグレード及びマップデータを、対応するアップグレードパッケージ領域にマップデータ及びアップグレードデータとしてインストールする。

10

【0169】

また、図 35 のステップ 917 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、図 38 において、受信装置 30 は、ステップ 951 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。ここで、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ある場合、受信装置 30 は、ステップ 952 でトランスレータをロードした後、図 36 のステップ 921 以後の動作過程を行ないながら第 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状が発生した場合、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、システムの製造時に、第 1 メモリ 250 にロードされたプログラムにすることができる。

20

【0170】

しかし、図 38 のステップ 951 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はない場合、受信装置 30 は、ステップ 953 及びステップ 954 において、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っ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、ステップ 955 でこれを感知し、ステップ 956 ーステップ 960 を実行しながら選択されたアップグレードパッケージの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てセキュリティー検査を行なった後、アップグレードデータが圧縮されていたらアップグレードデータの圧縮を解除する。その後、ステップ 961 ーステップ 969 を実行しながら第 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プさせて第 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。ステップ 961 ーステップ 969 の動作は、ステップ 938 ーステップ 946 と同一方法で実行することができる。

30

【0171】

上記のように、本発明の実施形態では、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、比較結果に応じて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プさせて第 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。

40

【0172】

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。

【0173】

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

50

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、システムの製造時に、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、複数個発生され、最も最近生成された第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である。

【0174】

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`)した後に、比較することができ、また、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。

10

【0175】

第2に、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。

【0176】

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、ヒストリーデータを含む。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、ロード失敗の有無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をするためのインデックス情報とに区分される。また、第2バージョンにブロックデータが挿入されて変更が発生する場合、変更されるブロックデータが、第1バージョンV1の設定された探索区間のブロックデータと同一あるいは類似の場合、これをアップグレードデータに生成せずマップデータに含めることもできる。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。

20

【0177】

第3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。

30

【0178】

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、比較の結果、ブロックのサイズ(大きいか、小さい場合、又はブロックが挿入される場合)に変化があるか、又は、ブロックのデータが互いに異なるデータ値を有するブロックを選択してアップグレードデータに設定する。ここで、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。そして、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として生成する。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。その場合、受信側で生成してインストールすることができる。また、アップグレードデータがマップ情報を含む場合には、マップデータの生成は省略することもできる。

40

【0179】

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。その場合、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。

【0180】

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、伝達されるアップグレー

50

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。

【0181】

上記のようなアップグレードパッケージを受信する受信装置30は、第1メモリ250及び第2メモリ260を備え、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、インストールされたアップグレードパッケージを第2メモリ260でアップグレードしてシステムの駆動を制御する。

【産業上の利用可能性】

【0182】

本発明は、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を有し、プログラムアップグレード方法およびシステムとして有用である。

【図面の簡単な説明】

【0183】

【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を説明する図面。

【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作動例を説明する図面。

【図3】図2のアップグレードパッケージプロセッサにおけるアップグレードパッケージデータフォーマットを説明する図面。

【図4】図2のアップグレードパッケージプロセッサにおけるアップグレードパッケージデータフォーマットを説明する図面。

【図5】図2のアップグレードパッケージプロセッサにおけるアップグレードパッケージデータフォーマットを説明する図面。

【図6】図2のアップグレードパッケージプロセッサにおけるアップグレードパッケージデータフォーマットを説明する図面。

【図7】図2のアップグレードパッケージプロセッサにおけるアップグレードパッケージデータフォーマットを説明する図面。

【図8】図2のアップグレードパッケージプロセッサにおけるアップグレードパッケージデータフォーマットを説明する図面。

【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10】アップグレードパッケージプロセッサの他の構成例を示す図面。

【図11】図9のアップグレードパッケージプロセッサにおいて発生するアップグレードパッケージの構成を示す図面。

【図12】図10のアップグレードパッケージプロセッサにおいて発生するアップグレードパッケージの構成を示す図面。

【図13】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の更に他の構成例を示す図面。

【図14】本発明の実施形態に係る受信装置の構成を示す図面。

【図15】図14に示された受信装置の第1メモリの構造を示す図面。

【図16】図15に示された第1メモリに格納される第2格納領域の構造を示す図面。

【図17】図16のような第2格納領域に格納される各々のアップグレードパッケージのヒストリーデータのフォーマットを示す図面。

【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システムのアップグレード動作を示す図面。

【図19】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説明する図面。

【図20】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説明する図面。

【図21】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説明する図面。

10

20

30

40

50

【図 2 2】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におけるバージョンアップグレード方法を説明する図面。

【図 2 3】本発明の実施形態に係るアップグレードパッケージの生成及びアップグレード過程を説明するフロー図。

【図 2 4】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 2 5】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 2 6】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 2 7】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 2 8】図 2 7における圧縮及び検証する過程を示すフロー図。

10

【図 2 9】図 2 7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 3 0】図 2 7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 3 1】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 3 2】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 3 3】本発明の実施形態に係る受信装置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 メモリにインストールする過程を示すフロー図。

20

【図 3 4】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におけるアップグレード過程を示すフロー図。

【図 3 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方法におけるアップグレード過程を示すフロー図。

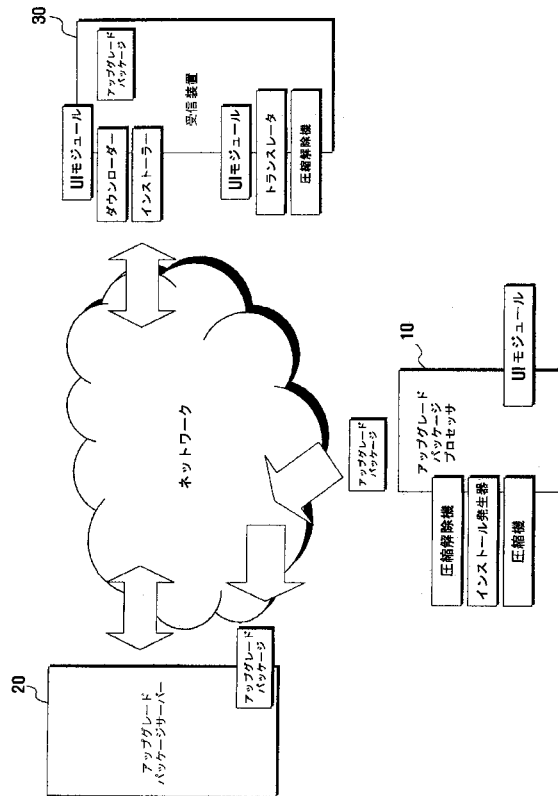
【図 3 6】本発明の他の実施形態に係るプログラムアップグレード方法におけるアップグレード過程を示すフロー図。

【図 3 7】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方法におけるアップグレード過程を示すフロー図。

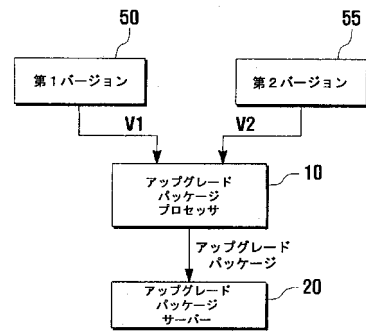
【図 3 8】本発明の他の実施形態に係るプログラムアップグレード方法におけるアップグレード過程を示すフロー図。

30

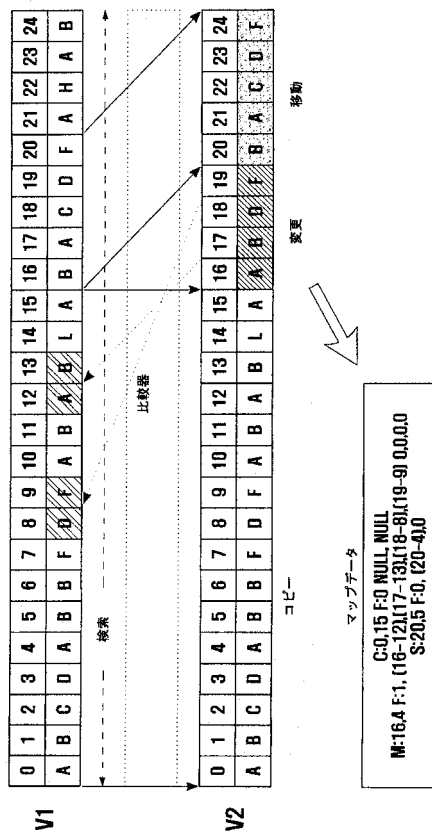
【 図 1 】



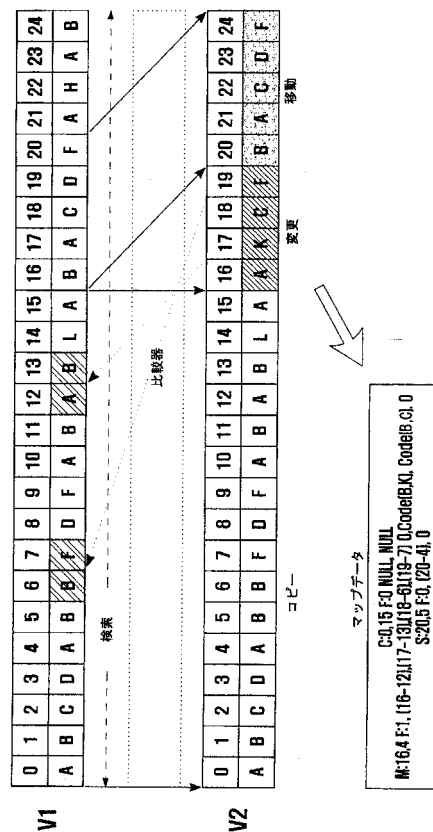
【圖 2】



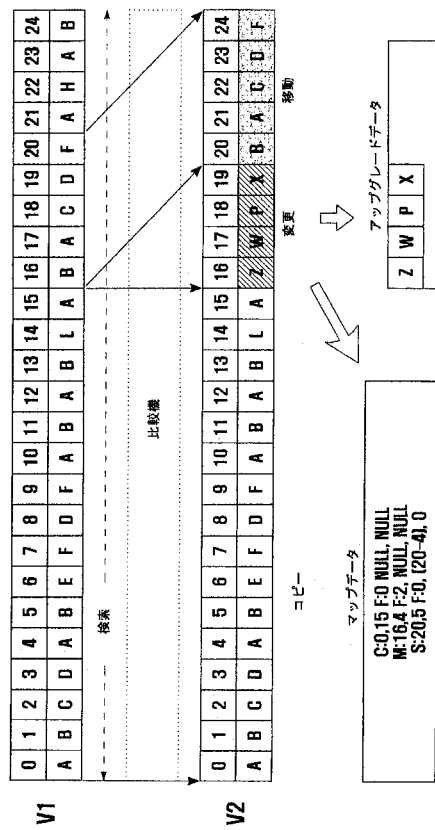
【 図 3 】



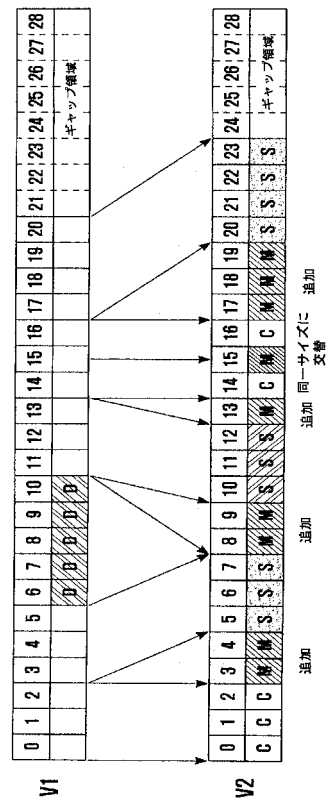
【 図 4 】



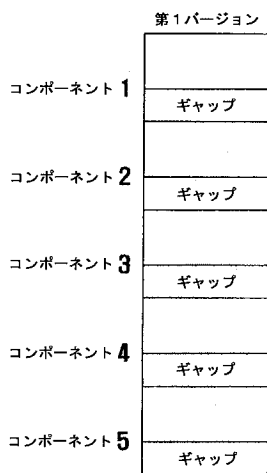
【図 5】



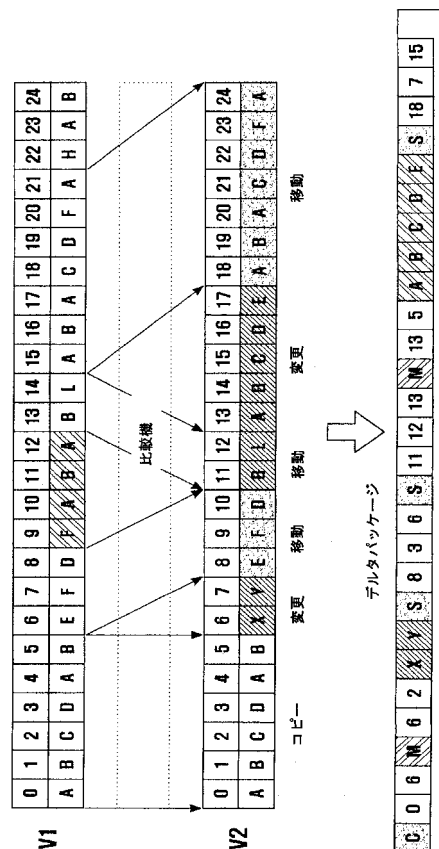
【図 6】



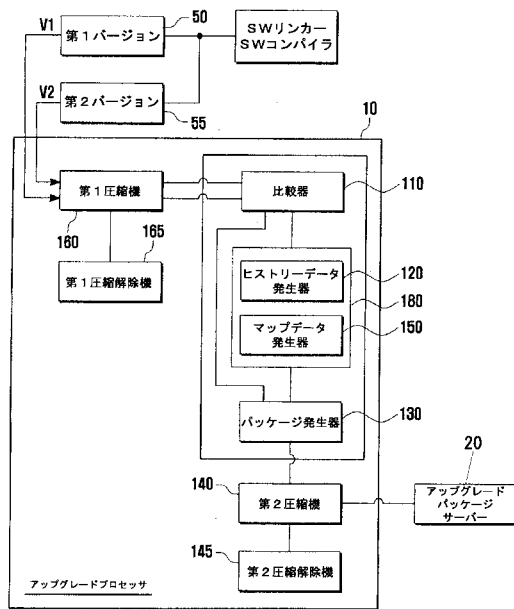
【図 7】



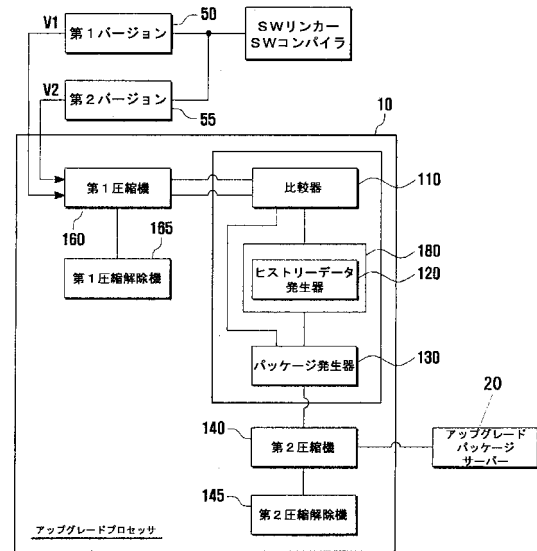
【図 8】



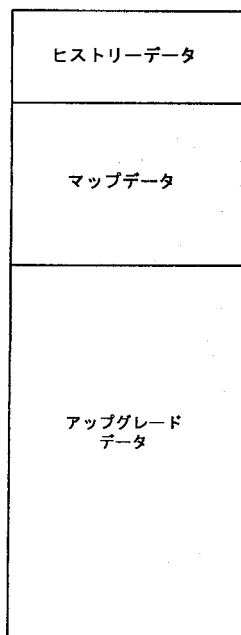
【図 9】



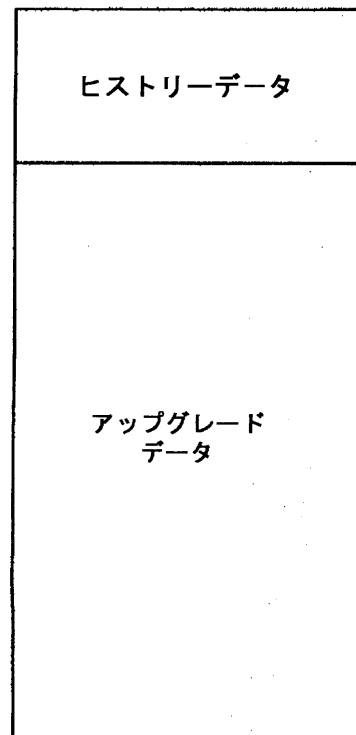
【図 10】



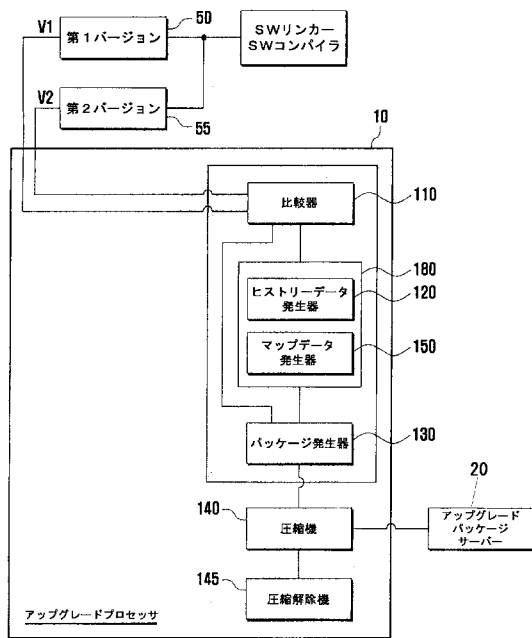
【図 11】



【図 12】



【図 13】



【図 14】

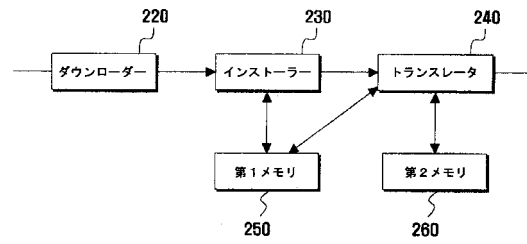
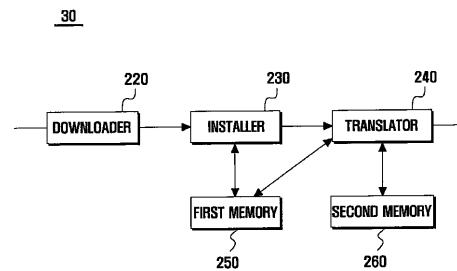
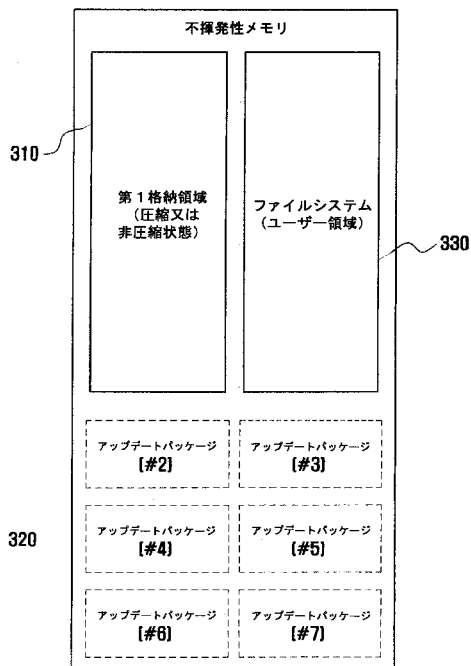


FIG. 14



【図 15】



【図 16】

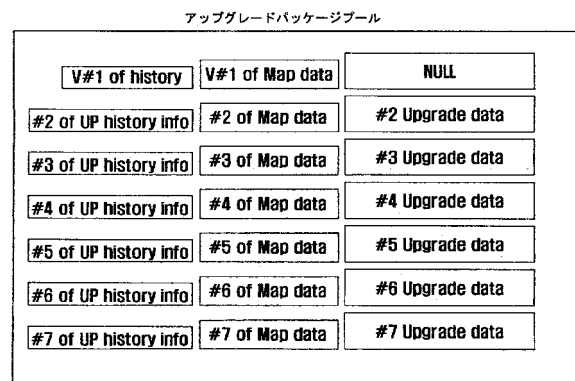
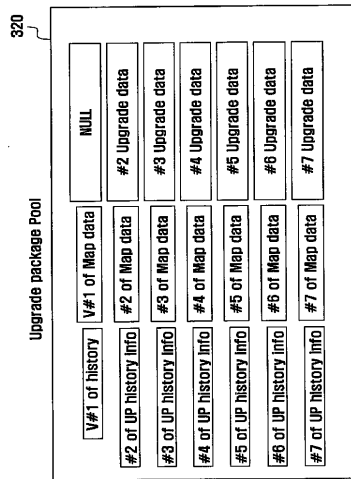
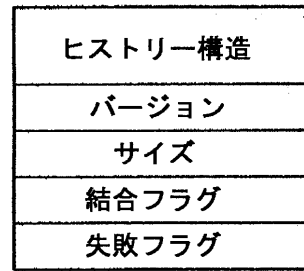


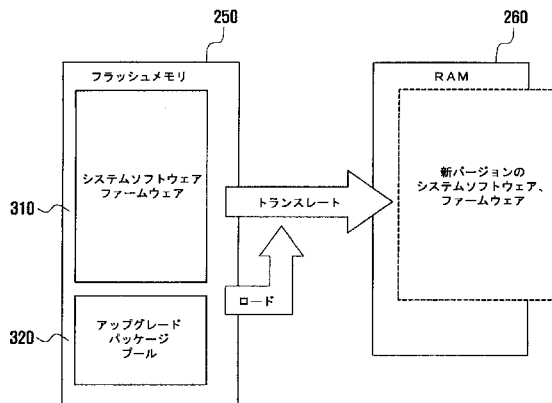
FIG. 16



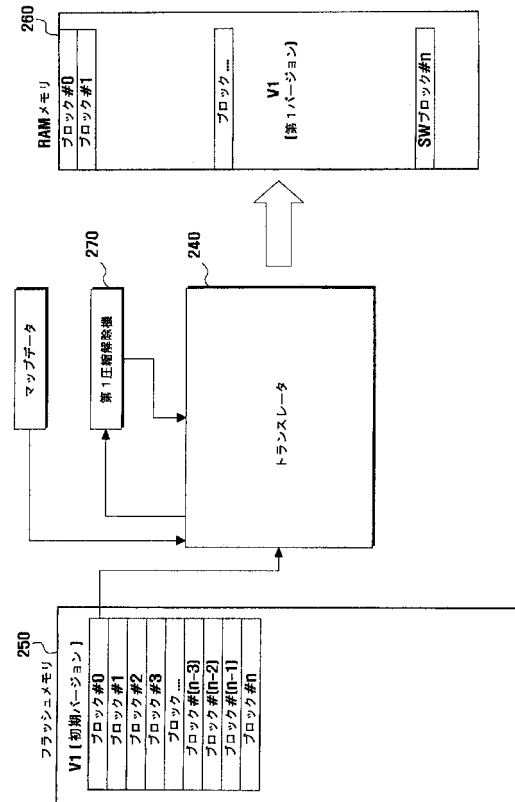
【図 17】



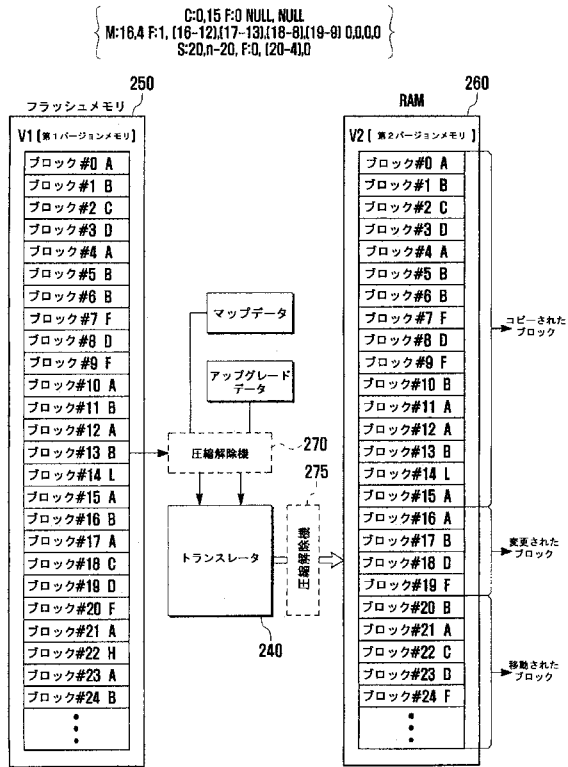
【図 18】



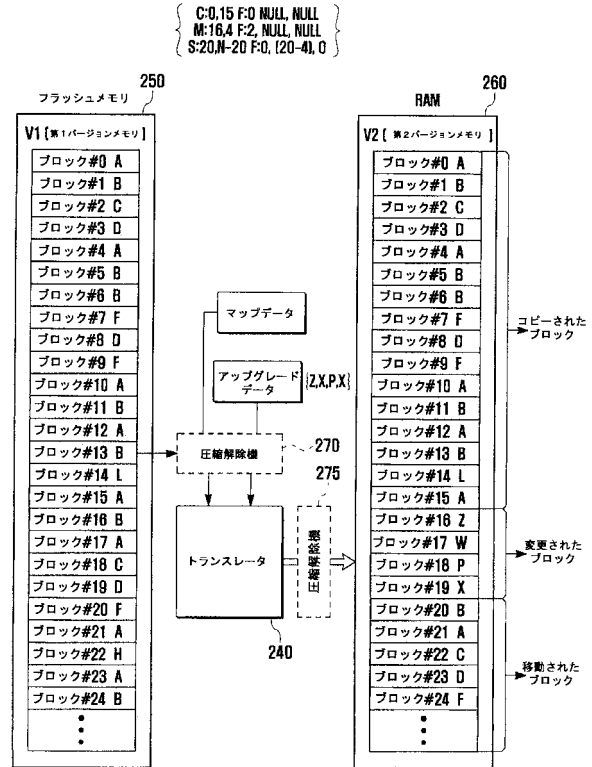
【図 19】



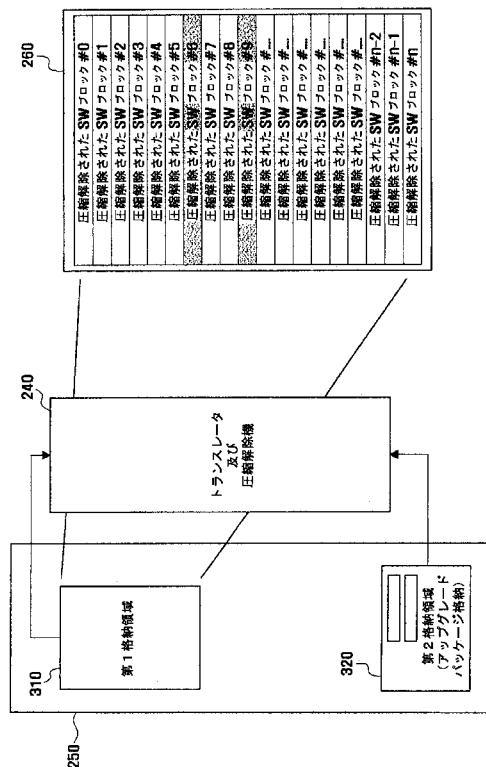
【図 20】



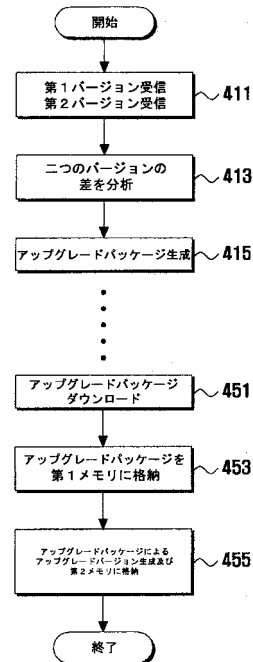
【図 21】



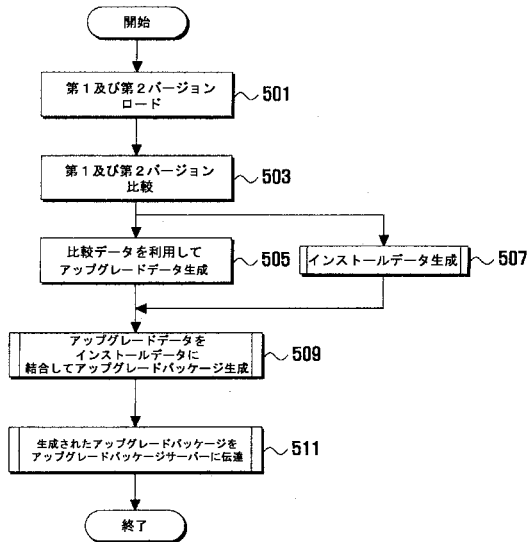
【図 22】



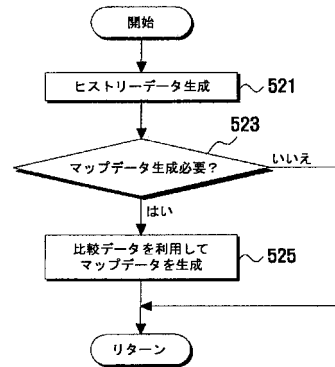
【図 23】



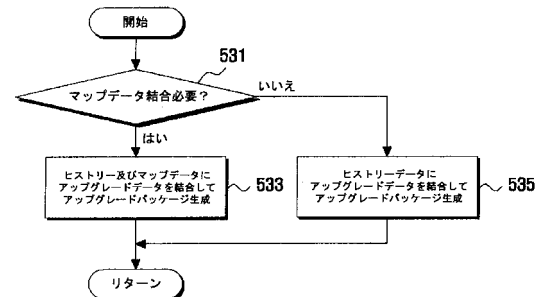
【図 24】



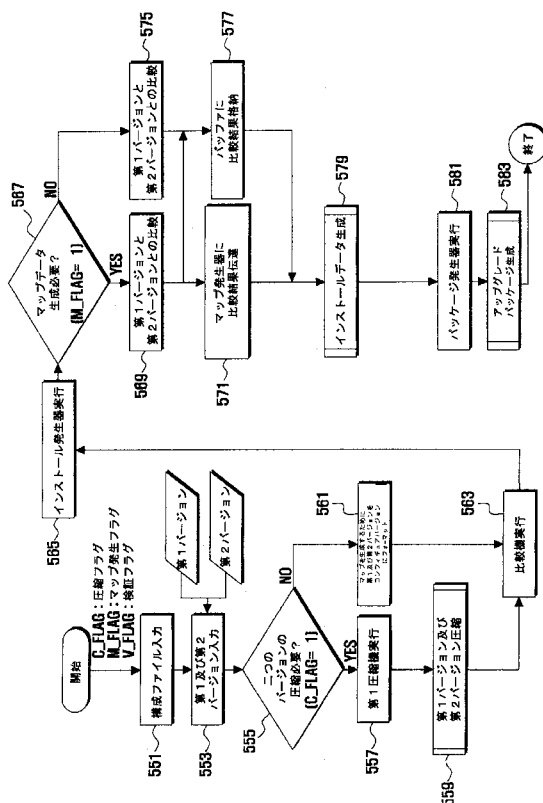
【図 25】



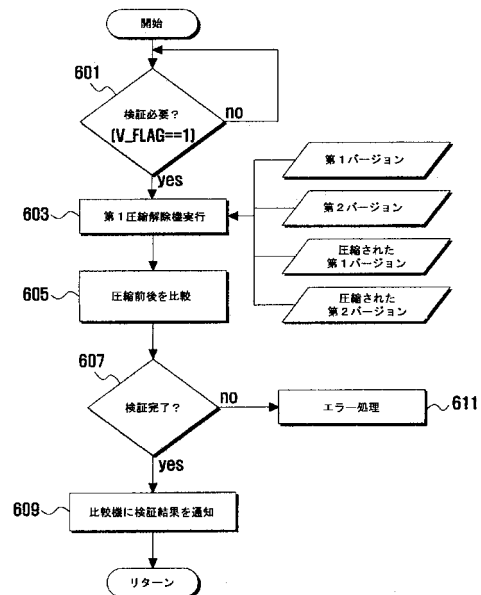
【図 26】



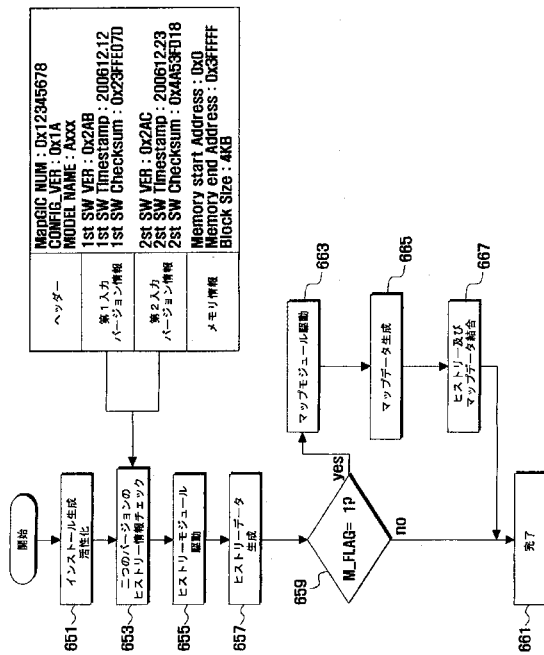
【図 27】



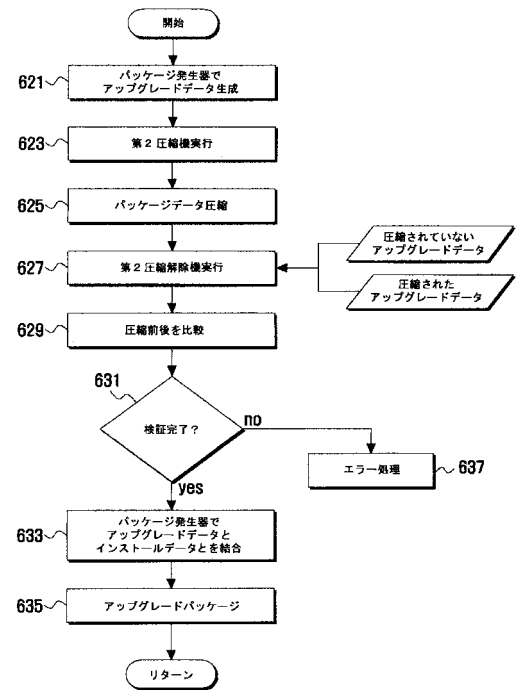
【図 28】



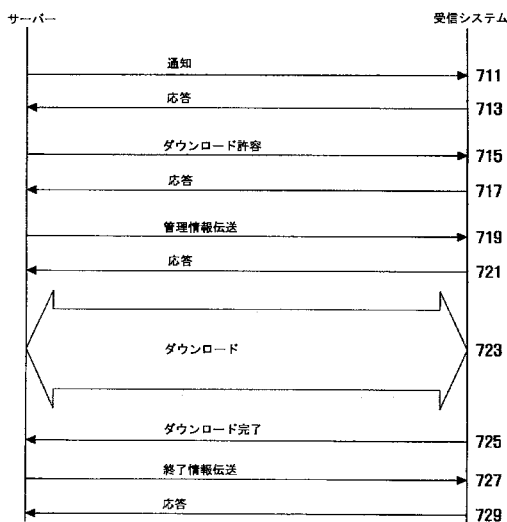
【図 29】



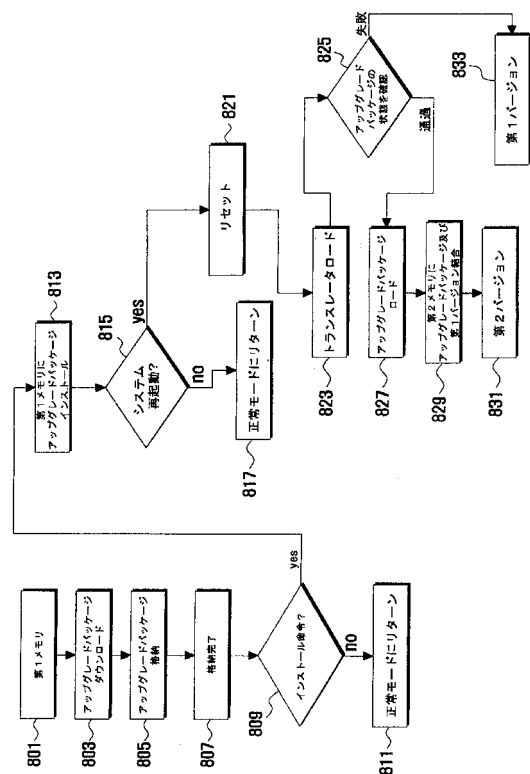
【図 30】



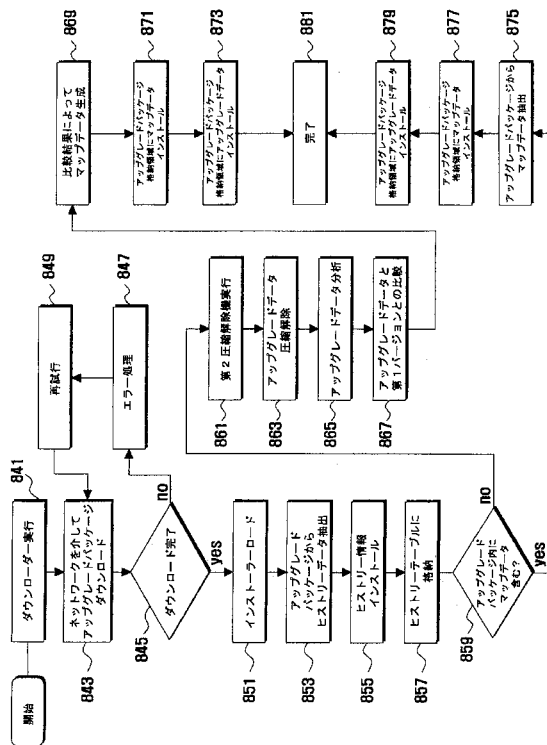
【図 31】



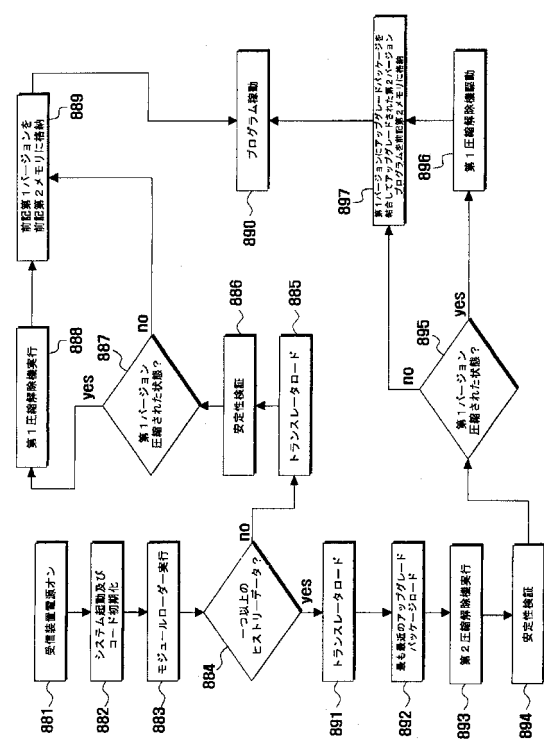
【図 32】



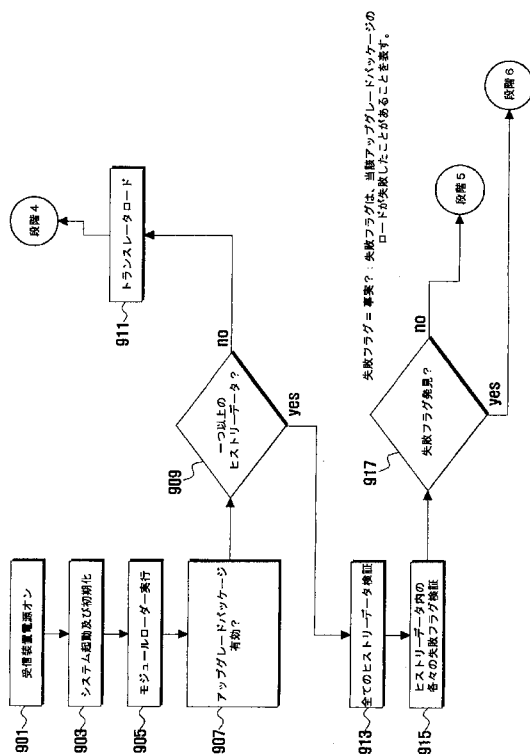
【図 3 3】



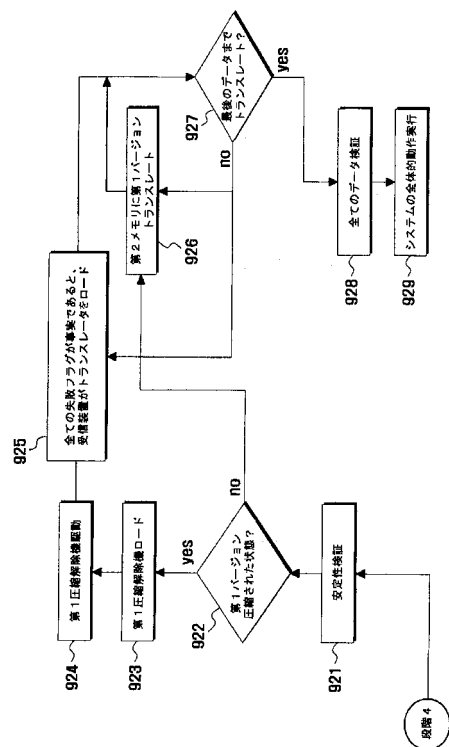
【図 3 4】



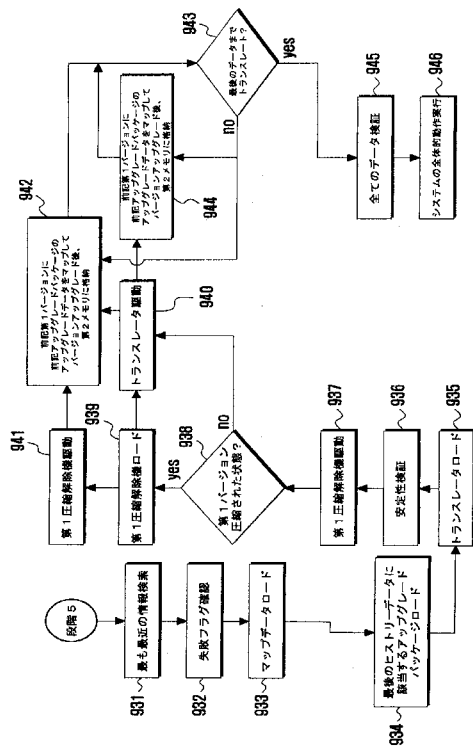
【図 3 5】



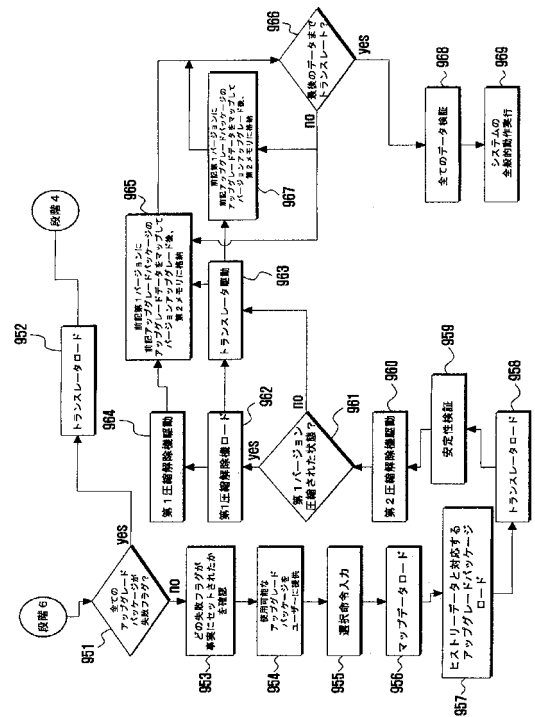
【图 3 6】



【図 37】



【図 38】



【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/KR2007/002947
A. CLASSIFICATION OF SUBJECT MATTER		
G06F 15/00(2006.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) 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 : software, upgrade, network and similar terms"		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 17-11209A (SONY CORP.) 13 January 2005 See Paragraph [0017]-[[0031], [0041]-[0045] & Figs. 1, 3	1-46
A	US 5,909,581A (SEONG KAB, PARK) 1 June 1999 See Column 3, Line 25 - Column 4, Line 36 & Fig. 3	1-46
A	US 2005/0193386 A1 (JED MCCALED et al.) 1 September 2005 See Paragraph [0023]-[0047] & Figs. 2, 3	1-46
A	US 2004/0237081 A1 (DANIEL P. HOMILLER) 25 November 2004 See abstract & Fig. 1	1-46
☐ 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 20 SEPTEMBER 2007 (20.09.2007)		Date of mailing of the international search report 20 SEPTEMBER 2007 (20.09.2007)
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/KR2007/002947

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 17-11209A	13.01.2005	CN1809809A	26.07.2006
		KR1020060021332A	07.03.2006
		US2006200812A1	07.09.2006
		W02004114126A1	29.12.2004
US 5,909,581A	01.06.1999	KR100286008B1	10.01.2001
US 2005/0193386A1	01.09.2005	None	
US 2004/0237081A1	25.11.2004	CN1791859A	21.06.2006
		EP1625496A2	15.02.2006
		JP2007503654T2	22.02.2007
		US2004215386A1	28.10.2004
		US2006247843A1	02.11.2006
		US7010416B2	07.03.2006
		US7231292B2	12.06.2007
		W02004102382A2	25.11.2004
		W02004102382A3	20.01.2005

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. B l u e t o o t h
2. Z I G B E E
3. G S M

【要約の続き】

システムを運用するステップと、からなる。

【選択図】 図 1