

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7013176号

(P7013176)

(45)発行日 令和4年1月31日(2022.1.31)

(24)登録日 令和4年1月21日(2022.1.21)

(51)国際特許分類

G 0 6 F 8/61 (2018.01)

F I

G 0 6 F 8/61

請求項の数 7 (全17頁)

(21)出願番号 特願2017-171179(P2017-171179)
(22)出願日 平成29年9月6日(2017.9.6)
(65)公開番号 特開2019-46360(P2019-46360A)
(43)公開日 平成31年3月22日(2019.3.22)
審査請求日 令和2年7月31日(2020.7.31)

(73)特許権者 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(74)代理人 100126240
弁理士 阿部 琢磨
(74)代理人 100124442
弁理士 黒岩 創吾
(72)発明者 布施 洋
東京都大田区下丸子3丁目30番2号キ
ヤノン株式会社内
審査官 金田 孝之

最終頁に続く

(54)【発明の名称】 プログラム、情報処理装置、及び制御方法

(57)【特許請求の範囲】

【請求項1】

アプリケーションソフトウェアを情報処理装置にインストールするためのプログラムであって、

前記アプリケーションのベース機能を前記情報処理装置にインストールする第1のインストール処理を実行する第1のインストール手段を有し、

前記第1のインストール手段は、

前記第1のインストール手段によって前記情報処理装置にインストールされた前記アプリケーションをアップデートするための未実行の第2のインストール処理が存在する場合、前記第2のインストール処理を実行する第2のインストール手段に対して前記第2のインストール処理を実行するように指示し、第1のインストール処理が完了していたとしても前記第2のインストール処理が完了するまでは前記アプリケーションのショートカットを作成せず、

未実行の第2のインストール処理が存在しない場合、前記第1のインストール処理が完了すると前記アプリケーションのショートカットを作成することを特徴とするプログラム。

【請求項2】

前記プログラムは、

未実行である前記第2のインストール処理が存在するかどうかを判定する第1の判定手段を更に有し、

前記第1の判定手段は、

前記情報処理装置のOSによって管理されたレジストリ情報に、前記第2のインストール手段を識別する識別情報が存在する場合に、未実行である前記第2のインストール処理が存在すると判定することを特徴とする請求項1に記載のプログラム。

【請求項3】

前記プログラムは、
前記第2のインストール手段によって前記アプリケーションがアップデートされた後、
前記ショートカットが存在するかを判定する第2の判定手段を更に有し、
前記第2の判定手段によって前記ショートカットが存在すると判定された場合は、
前記ショートカットを作成することなく処理を終了することを特徴とする請求項1または2に記載のプログラム。

10

【請求項4】

アプリケーションソフトウェアをインストール可能な情報処理装置であって、
前記アプリケーションのベース機能を前記情報処理装置にインストールする第1のインストール処理を実行する第1のインストール手段を有し、
前記第1のインストール手段は、
前記第1のインストール手段によって前記情報処理装置にインストールされた前記アプリケーションをアップデートするための未実行の第2のインストール処理が存在する場合、
前記第2のインストール処理を実行する第2のインストール手段に対して前記第2のインストール処理を実行するように指示し、第1のインストール処理が完了していたとしても前記第2のインストール処理が完了するまでは前記アプリケーションのショートカットを作成せず、

20

未実行の第2のインストール処理が存在しない場合、前記第1のインストール処理が完了すると前記アプリケーションのショートカットを作成することを特徴とする情報処理装置。

【請求項5】

前記情報処理装置は、
未実行である前記第2のインストール処理が存在するかどうかを判定する第1の判定手段を更に有し、
前記第1の判定手段は、
前記情報処理装置のOSによって管理されたレジストリ情報に、前記第2のインストール手段を識別する識別情報が存在する場合に、未実行である前記第2のインストール処理が存在すると判定することを特徴とする請求項4に記載の情報処理装置。

30

【請求項6】

前記情報処理装置は、
前記第2のインストール手段によって前記アプリケーションがアップデートされた後、
前記ショートカットが存在するかを判定する第2の判定手段を更に有し、
前記第2の判定手段によって前記ショートカットが存在すると判定された場合は、
前記ショートカットを作成することなく処理を終了することを特徴とする請求項4または5に記載のプログラム。

【請求項7】

アプリケーションソフトウェアをインストール可能な情報処理装置の制御方法であって、
前記アプリケーションのベース機能を前記情報処理装置にインストールする第1のインストール処理を実行する第1のインストールステップを有し、
前記第1のインストールステップは、
前記第1のインストールステップによって前記情報処理装置にインストールされた前記アプリケーションをアップデートするための未実行の第2のインストール処理が存在する場合、前記第2のインストール処理を実行する第2のインストールステップに対して前記第2のインストール処理を実行するように指示し、第1のインストール処理が完了していたとしても前記第2のインストール処理が完了するまでは前記アプリケーションのショートカットを作成せず、

40

未実行の第2のインストール処理が存在しない場合、前記第1のインストール処理が完了

50

すると前記アプリケーションのショートカットを作成することを特徴とする情報処理装置の制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、アプリケーションソフトウェアをインストールするためのプログラム、情報処理装置、及び制御方法に関する。

【背景技術】

【0002】

近年、アプリケーションソフトウェアの大規模・複雑化に伴い、1つのアプリケーションソフトウェアを動作させるために複数のインストーラを正しい手順で実行する必要がある製品が増えてきた。しかしながら、ＩＴリテラシーの低いユーザにとって正しい手順でインストーラを実行することは必ずしも容易なことではない。

10

【0003】

そこで特許文献１には、サーバーコンピュータにおいて、複数のソフトウェアをインストールする際のインストール順によるコンフリクトチェック（整合性確認）を実施することが記載されている。特許文献１によれば、ユーザはインストーラの実行順序を意識することなく所望のアプリケーションソフトウェアがインストールできるようになる。

【先行技術文献】

【特許文献】

20

【0004】

【文献】特開２０１２－４８４１０号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、複数のインストーラを連続実行している際にユーザがインストール中のアプリケーションソフトウェアを起動してしまうと、インストール処理が失敗する場合がある。

【0006】

例えば、ユーザがサービスパックを含むアプリケーションソフトウェアを購入した場合、製品にはベース用のインストーラ及びサービスパック用のインストーラが同梱されていることがある。一般的にこのような構成のインストーラでは、ベースのインストール処理が完了した後に自動的にサービスパックのインストール処理が開始されることが多い。この場合、ベースのインストール処理が完了した時点でＯＳ（Operating System）のデスクトップ画面上にショートカットが作成される。そのため、サービスパックのインストール中にユーザがショートカットからアプリケーションソフトウェアを容易に起動することが出来てしまう。

30

【0007】

ここでユーザがインストール中のアプリケーションソフトウェアを起動した場合、ＯＳによってアプリケーションソフトウェア実行に必要なモジュールはロードされ、ロックされる。そのため、サービスパックのインストーラによるサービスパック用モジュールの更新ができなくなるため、サービスパックのインストール処理が失敗するという問題が発生する。

40

【0008】

また、サービスパックのインストーラによる更新ができなかったモジュールを、ソフトウェアの再起動後に更新することも可能である。しかしながら、ユーザがサービスパックを含むアプリケーションソフトウェアを購入したにもかかわらず、そのユーザが「Bug - Fix未適用の製品」や「新機能がまだ存在しないベース機能のみの製品」を一時的にでも使用することは好ましくない。

【0009】

50

そこで、本発明は、既にインストールされたアプリケーションソフトウェアについての追加インストールを行うためのインストーラを起動させる前に、前記インストールされたアプリケーションソフトウェアの起動のための処理を実行させないことを目的とする。

【課題を解決するための手段】

【 0 0 1 0 】

上記課題を解決するために、第 1 のインストール手段によって情報処理装置にインストールされたアプリケーションをアップデートするための未実行の第 2 のインストール処理が存在する場合、第 2 のインストール処理を実行する第 2 のインストール手段に対して第 2 のインストール処理を実行するように指示し、第 1 のインストール処理が完了していたとしても前記第 2 のインストール処理が完了するまでは前記アプリケーションのショートカットを作成せず、未実行の第 2 のインストール処理が存在しない場合、前記第 1 のインストール処理が完了すると前記アプリケーションのショートカットを作成することを特徴とする。

10

【発明の効果】

【 0 0 1 1 】

本発明により、既にインストールされたアプリケーションソフトウェアについての追加インストールを行うためのインストーラを起動させる前に、前記インストールされたアプリケーションソフトウェアの起動のための処理を実行させないようにすることができる。

【図面の簡単な説明】

【 0 0 1 2 】

20

【図 1】ハードウェア構成例を示す図

【図 2】ソフトウェア構成例を示す図

【図 3】インストーラに関する設定情報の一例を示す図

【図 4】実施例 1 の処理を示すフローチャート

【図 5】実施例 2 の処理を示すフローチャート

【図 6】実施例 3 の処理を示すフローチャート

【発明を実施するための形態】

【 0 0 1 3 】

本発明では、ユーザがインストーラ実行中に、アプリケーションソフトウェアの起動を防ぐためのショートカット作成方法を実現する。

30

【 0 0 1 4 】

以下、本発明を実施するための最良の形態について図面を用いて説明する。

【 0 0 1 5 】

[ハードウェア構成]

図 1 は、本発明の一実施形態におけるハードウェア構成例を示す図である。図 1 では、コンピュータである情報処理装置 100 のハードウェア構成例を示している。情報処理装置 100 は、ユーザ操作入力を受信する入力デバイスであるキーボード 105 とポインティングデバイス 106 が設けられている。

【 0 0 1 6 】

キーボード 105 は、情報処理装置 100 に対して文字や数字などの情報を入力する装置である。キーボードは、例えば仮名や英文字などを入力するキーや数字を入力するためのテンキー、各種機能キー、カーソルキー及びその他のキーを備える。

40

【 0 0 1 7 】

ポインティングデバイス 106 は、例えばマウスなどを備える。ユーザが、GUI (Graphical User Interface) などを用いて情報処理装置 100 を操作する場合、表示装置上に表示されたボタンやアイコンなどをマウスでクリックすることにより、所定の情報の入力を行うことができる。例えば、情報処理装置 100 が、インストールされたアプリケーションソフトウェアを表すアイコンを表示部 102 上に表示し、ユーザがアイコンをマウスでクリックすることに応じて、CPU 101 がアプリケーションソフトウェアを起動させる。さらに、ユーザに視覚的な出力情報フィードバックを与える

50

表示部 102 を備える。

【0018】

表示部 102 は、例えば、CRT (Cathode Ray Tube) ディスプレイ、液晶ディスプレイ、又はプラズマディスプレイを備える画面上に各種情報を表示する。表示部 102 は、キーボードやマウスの入力結果や、インストーラやアプリケーションソフトウェアが提供する画面 (例えばユーザインターフェース) 等を表示する。

【0019】

さらに、情報処理装置 100 は、動作を制御するための手段である CPU 101、RAM 103、ROM 104、HDD 107 を備える。本実施形態に特有の動作として、CPU 101 は、情報処理装置 100 上では図 2 に示す複数のインストーラ 200・210 の動作を制御する。CPU 101 は、情報処理装置 100 の各インストーラを実行し、所定のプログラムに従って、各種の演算、情報処理、情報処理装置 100 全体の制御処理を行う。

10

【0020】

RAM 103、ROM 104、HDD 107 は本実施形態における各種プログラムや実行情報を保管する。また、DVD/CD に記録されたデータを読み出すための DVD/CD-ROM ドライバ 108 も設ける。RAM 103 は、CPU 101 により OS (Operating System) や本実施例を実現するための制御プログラムを含むその他の制御プログラムがロードされ実行される。また、制御プログラムを実行するために用いられる各種作業領域、一時待避領域として機能する。本実施形態では、RAM 103 には、CPU 101 がインストーラの起動やシステム内部の処理などを行うためのエリアが確保される。ROM 104 は、CPU 101 の各種制御を実行する各種制御プログラム、データ及びパラメータ等を記憶する記憶装置である。

20

【0021】

HDD 107 は、読み書き可能な記憶媒体と、その記憶媒体に対してプログラムやデータを読み書きするための駆動装置と、を備える。本実施形態では、HDD 107 は、ハードディスクを備えるローカルディスクである。なお、HDD 107 が備える記憶媒体が、光磁気ディスク、磁気ディスク、半導体メモリ等であってもよい。HDD 107 には、図 2 に示す複数のインストーラ 200・210 が記憶されている。また、HDD 107 には、インターフェイスデバイス I/O 109 を制御し、ネットワークを介した通信を維持する通信プログラムが記憶されている。また、HDD 107 には、メモリ管理やファイル入出力管理などのクライアントを動作させるための基本アプリケーションソフトウェアである OS などが記憶されている。

30

【0022】

さらに、外部機器との通信を行うインターフェイスデバイス I/O 109 を備え、プログラム実行を行う CPU 101 を備える。なお、周辺機器との接続形態は有線/無線を問わない。インターフェイスデバイス I/O 109 は、ネットワーク (例えば、LAN やインターネット) を介して他の端末 (Web サーバ、メールサーバなど) に接続する。インターネットを介した通信を行う場合には、例えば、TCP/IP 等のプロトコルが用いられる。

【0023】

なお、本実施例で説明される処理は、ROM 104、または HDD 107 に保存されたプログラムや実行情報、一時データを RAM 103 にロードし、CPU 101 で実行することによって実現される。

40

【0024】

[ソフトウェア構成]

図 2 は、ソフトウェアの構成例を示す図である。

【0025】

情報処理装置 100 は、複数のインストーラ (第一のインストーラ 200、第二のインストーラ 210) を備える。なお、本実施例では、インストーラ 210 はインストーラ 200 と同様の構成であり重複する情報は省略する。

50

【 0 0 2 6 】

ここで、第一のインストーラ 2 0 0 および第二のインストーラ 2 1 0 は、アプリケーションソフトウェアの制作者によって D V D や C D 等に記憶され、或いは、アーカイブファイルに変換され W e b ページ等にアップロードされることが一般的である。また、インストーラの配布方式は、単数のインストーラを別々に配布する単数インストーラ形式や、複数のインストーラをあわせて配布する複数インストーラ形式の 2 パターンが存在する。例えば、アプリケーションソフトウェアのサービスパックがリリースされて、ベースとサービスパックの 2 つのインストーラを 1 つにまとめて配布する場合、複数インストーラ形式となる。なお、ユーザの P C に既にベースのインストールしており、後にリリースされたサービスパックを追加でインストールする場合には、サービスパックのインストーラが単数インストーラ形式で配布されることとなる。図 2 で示すような第一のインストーラ 2 0 0 および第二のインストーラ 2 1 0 は、複数インストーラ形式で配布されたインストーラセットであると言える。

10

【 0 0 2 7 】

情報処理装置 1 0 0 が備える各構成について説明する。

【 0 0 2 8 】

インストーラ 2 0 0 は、アプリケーションソフトウェアのインストール実行するための各種のユーザインターフェースを備え、情報処理装置 1 0 0 のデスクトップ上でユーザからの要求を受け付けて、各機能処理の実行を総括的に管理している。

【 0 0 2 9 】

インストーラ 2 0 0 は、表示部 2 0 1 および制御部 2 0 2 を有する。各インストーラを構成するコンポーネントについて説明する。

20

【 0 0 3 0 】

表示部 2 0 1 は、インストーラ 2 0 0 のユーザインターフェースの構築を行う。また、G U I を介して、ユーザからの各種入力操作の受け付けを処理する。

【 0 0 3 1 】

制御部 2 0 2 は、インストーラ制御に関する処理全般の機能を備える。例えば、ユーザからの各種入力操作の受け付けを基に、図示を省略するアプリケーションソフトウェアのインストール、アンインストール、修復インストールやアプリケーションの設定や構築を行う。また、制御部 2 0 2 は、情報処理装置 1 0 0 のインストーラに関する設定情報を解析する。設定情報とは、例えば、インストーラ情報 3 0 0 や各インストーラが管理するレジストリ情報 3 2 0 等を示す。設定情報の詳細については、図 3 を用いて後述する。

30

【 0 0 3 2 】

さらに、制御部 2 0 2 は、アプリケーションソフトウェアのインストールに際して、既にインストールされたアプリケーションソフトウェアの起動のためのユーザ入力を受け付け可能にする処理を実行する。その処理の一例として、制御部 2 0 2 は、情報処理装置の O S が提供するデスクトップ画面上に、アプリケーションソフトウェアの起動のためのユーザ入力を受け付けるショートカットを作成する。また処理の他の例として、制御部 2 0 2 は、メニュー画面の一覧にアプリケーションソフトウェアを表示させたり、ファイルに対する拡張子の関連付けを行ったりしてもよい。

40

【 0 0 3 3 】

なお、インストーラ 2 0 0 は、アプリケーションソフトウェアの基本機能を情報処理装置にインストールするためのインストーラである。また、表示部 2 1 1 および制御部 2 1 2 を有するインストーラ 2 1 0 は、アプリケーションソフトウェアの追加インストールするためのサービスパックのインストーラである。サービスパックのインストーラにおける追加インストールでは、例えば、ベースのインストーラにおける基本機能についてのバグ修正やセキュリティ強化、ベースの機能をバージョンアップした機能の追加などが行われる。一方、図 5 で後述する実施例 2 におけるインストーラ 2 1 0 は、アプリケーションソフトウェアの追加インストールとして、オプション機能をインストールするためのインストーラとして扱う。オプション機能のインストーラにおいては、ベースの機能には含まれな

50

い付帯機能であり、ユーザがカスタマイズして追加可能な機能がインストールされる。サービスパックのインストーラの制御部 212 は、ベースのインストーラの制御部 202 と同様に、既にインストールされたアプリケーションソフトウェアの起動のためのユーザ入力を受け付け可能にする処理を実行可能である。例えば、制御部 212 はショートカット作成の機能などを有している。一方、オプション機能のインストーラの制御部 212 は、既にインストールされたアプリケーションソフトウェアの起動のためのユーザ入力を受け付け可能にする処理を実行する機能を有していないため、ショートカットの作成などはできない。

【0034】

本実施例で説明される処理は、ROM 104 または HDD 107 に保存されたプログラムが、RAM 103 にロードされて、CPU 101 で実行されることによって実現される。

【0035】

[インストーラに関する設定情報]

図 3 は、インストーラに関する設定情報の一例を示している。

【0036】

インストーラに関する設定情報には、インストーラが保持するインストーラ情報および OS が保持するレジストリ情報が含まれる。これらの情報は各インストーラのインストール処理において、併用して利用される。

【0037】

初めに、図 3 (a) で示すインストーラ情報 300 について説明する。インストーラ情報 300 は、インストーラ 200 のインストール処理に必要な設定情報であり、インストーラ 200 が保持する情報である。設定情報は、例えば `InstallerVersion 301` や `ProductCode 302`、`UpgradeCode` 等を含む。同様に、図 3 (b) で示すインストーラ情報 310 は、インストーラ 210 のインストール処理に必要な設定情報であり、インストーラ 210 が保持する情報である。なお、本実施例では、インストーラ情報 310 はインストーラ情報 300 と同様の構成であり、重複する情報の説明は省略する。

【0038】

インストーラ情報 300 の各項目について説明する。`InstallerVersion 301` は、インストーラ 210 がインストール可能なアプリケーションソフトウェアのバージョンを示す。一般的にアプリケーションソフトウェアのバージョンは、メジャーバージョン、マイナーバージョン、リビジョン（サービスパックのバージョン）の 3 つの値で構成される。例えば、メジャーバージョンの値が「3」、マイナーバージョンの値が「0」、リビジョンの値が「0」であれば、アプリケーションソフトウェアのバージョンは「3.0.0」となる。本実施例における `InstallerVersion 301` の値は「3.0.0」であるため、インストーラ 200 によってバージョン 3.0.0 のアプリケーションソフトウェアがインストールされる。また、インストーラ情報 310 の `InstallerVersion 311` の値は「3.0.1」であるため、インストーラ 210 によってバージョン 3.0.1 のアプリケーションソフトウェアがインストールされる。

【0039】

`ProductCode 302` は、アプリケーションソフトウェアの識別などに用いられる固有の番号や文字列である。この値はインストーラ 200 によって、レジストリ情報 320 の `ProductCode 322` へ書き込まれる。また、一般的に、リリース済みのアプリケーションソフトウェアに適用可能なサービスパックをリリースする場合、`ProductCode 302` および `UpgradeCode` は変更しない。即ち、サービスパックをインストールした場合、レジストリ情報 320 に書き込まれる `ProductCode 322` の値および `UpgradeCode` の値は変更されない。なお、第二のインストーラ 210 がオプション機能のインストーラである場合、`ProductCode` や `UpgradeCode` は、第一のインストーラ 200 が保持する値と異なる（不図示）。値が異なる場合、第一のインストーラ 200 と第二のインストーラ 210 によってインス

10

20

30

40

50

トールされるアプリケーションソフトウェアは、それぞれアップグレードの依存関係がないことになる。なお、`ProductVersion323`および`ProductCode324`については、実施例2で説明する。

【0040】

図3を例にまとめると、インストーラ情報300を有するインストーラ200はベースであり、インストーラ情報310を有するインストーラ210は、ベースに対するサービスパックであると言える。

【0041】

次に、図3(c)で示すレジストリ情報320について説明する。レジストリ情報320は、情報処理装置100のOSが保持するレジストリ情報一覧の一例を示す。

10

【0042】

レジストリ情報320は、情報処理装置100にインストールされているアプリケーションソフトウェアの情報であり、ROM104に保存されている。例えば、レジストリ情報320は、アプリケーションソフトウェアの`ProductVersion321`や`ProductCode322`、`UpgradeCode`、インストールパス等を含む。また、レジストリ情報320は複数のアプリケーションソフトウェアを管理する。図3を例にあげると、レジストリ情報320は、ベースのアプリケーションソフトウェアとベースに対するオプション機能のアプリケーションソフトウェアの2つのアプリケーションソフトウェアを含む。具体的には、`ProductVersion321`、`ProductCode322`はベースのアプリケーションソフトウェアのレジストリ情報に関する情報である。また、`ProductVersion323`、`ProductCode324`はベースに対するオプション機能のアプリケーションソフトウェアのレジストリ情報に関する情報である。なお、レジストリ情報320に含まれる2つの`ProductVersion`、`ProductCode`において、重複する情報の説明は省略する。

20

【0043】

レジストリ情報320の各項目について説明する。`ProductVersion321`は、情報処理装置100にインストールされているベースのアプリケーションソフトウェアのバージョンを示す。例えば、`ProductVersion321`の値が「3.0.0」であれば、バージョン3.0.0のアプリケーションソフトウェアがインストールされていることになる。

30

【0044】

ここで、サービスパックがインストールされている場合は、リビジョンの値が「1」以上となる。したがって、図3(c)で示す例では、リビジョンの値が「0」であるため、サービスパックはインストールされていないと判断できる。

【0045】

`ProductCode322`は、アプリケーションソフトウェアの識別などに用いられる固有の番号や文字列である。レジストリ情報320にアプリケーションソフトウェアの`ProductCode302`が存在すれば、アプリケーションソフトウェアがインストールされているということになる。

【0046】

なお、インストーラ情報、および、レジストリ情報が保持する情報は、本実施例に限ったものとしな

40

【0047】

[実施例1における各インストーラの処理]

実施例1では、第一のインストーラ200と第二のインストーラ210が連続実行される場合における課題の解決手法を説明する。なお、実施例1では第一のインストーラ200をベースのインストーラ、第二のインストーラ210をサービスパックのインストーラとして扱う。

【0048】

まず、実施例1における課題を説明する。ベース用とサービスパック用の2つのインスト

50

ーラが連続で実行される場合にサービスパックのインストール処理が失敗するケースがある。具体的には、２つのインストーラを連続実行する場合、まず初めにベースのインストーラによって、ＯＳのデスクトップ画面上にショートカットが作成される。その後サービスパックのインストーラが実行されることとなるが、ベースのインストーラ実行が完了した時点でユーザはＯＳのデスクトップ画面上に作成されたショートカットからアプリケーションソフトウェアを起動するための指示をすることができる。ここで、サービスパックのインストーラ実行中にアプリケーションソフトウェアが起動された場合、サービスパックのインストーラはアプリケーションソフトウェアのモジュールを更新できずにインストール処理が失敗することがある。

【 ０ ０ ４ ９ 】

10

ここでは前述の課題を解決するためのインストーラ構成に応じたショートカット作成のタイミングの動的な変更手法について説明する。

【 ０ ０ ５ ０ 】

[実施例 １ における各インストーラの処理]

図 ４ は、ベースのインストーラ ２ ０ ０ およびサービスパックのインストーラ ２ １ ０ の処理のフローチャートである。図 １ の情報処理装置 １ ０ ０ 上で動作するインストーラ ２ ０ ０ ・ ２ １ ０ を例に処理の流れを説明する。図 ４ に示すフローチャートでは、インストーラにおける動的なショートカット作成処理を実施する手法を示す。また、Ｓ １ ０ ０ １ ～ Ｓ １ ０ １ ０ の各ステップはインストーラ ２ ０ ０ による処理であり、Ｓ １ １ ０ １ ～ Ｓ １ １ ０ ３ の各ステップはインストーラ ２ １ ０ による処理である。

20

【 ０ ０ ５ １ 】

図 ４ (a) は、ベースのインストーラであるインストーラ ２ ０ ０ による処理の流れを示すフローチャートである。

【 ０ ０ ５ ２ 】

インストーラ ２ ０ ０ が起動されると、Ｓ １ ０ ０ １ にて、制御部 ２ ０ ２ はインストーラ ２ ０ ０ でインストール可能なアプリケーションソフトウェアが未インストールかを判定する。はい (未インストールである) の場合、Ｓ １ ０ ０ ２ へ進み、いいえ (未インストールでない) の場合、Ｓ １ ０ ０ ３ へ進む。

【 ０ ０ ５ ３ 】

Ｓ １ ０ ０ １ における判定のロジックについて具体的に説明する。制御部 ２ ０ ２ は、レジストリ情報 ３ ２ ０ を取得する。そして制御部 ２ ０ ２ は、レジストリ情報 ３ ２ ０ とインストーラ情報 ３ ０ ０ を基に、インストーラ情報 ３ ０ ０ が保持する `Product Code 302` がレジストリ情報 ３ ２ ０ に存在するかを解析する。レジストリ情報 ３ ２ ０ の `Product Code 322` に `Product Code 302` が存在する場合、ベースのアプリケーションソフトウェアは未インストールでないと判定し、Ｓ １ ０ ０ ３ へ進む。一方、レジストリ情報 ３ ２ ０ の `Product Code 322` に `Product Code 302` が存在しない場合、または、レジストリ情報 ３ ２ ０ が存在しない場合、ベースのアプリケーションソフトウェアは未インストールであると判定し、Ｓ １ ０ ０ ２ へ進む。

30

【 ０ ０ ５ ４ 】

次に、Ｓ １ ０ ０ ２ にて、制御部 ２ ０ ２ はベースのアプリケーションソフトウェアのインストール実行フラグを `ON` にする。

40

【 ０ ０ ５ ５ 】

次に、Ｓ １ ０ ０ ３ にて、複数インストーラ形式のインストーラセットに未インストールのサービスパックが存在するかを判定する。はい (存在する) の場合、Ｓ １ ０ ０ ４ へ進み、いいえ (存在しない) の場合、Ｓ １ ０ ０ ５ へ進む。

【 ０ ０ ５ ６ 】

Ｓ １ ０ ０ ３ における判定のロジックについて具体的に説明する。初めに制御部 ２ ０ ２ は、サービスパックのインストーラ情報 ３ １ ０ を取得する。ここでインストーラ情報 ３ １ ０ が取得できない場合、サービスパックのインストーラ ２ １ ０ は存在しないと判定し、Ｓ １ ０ ０ ５ へ進む。次に制御部 ２ ０ ２ は、レジストリ情報 ３ ２ ０ を取得する。そして制御部 ２ ０

50

2 は、サービスパックのインストーラ情報 3 1 0 とレジストリ情報 3 2 0 とを基に、インストーラ情報 3 1 0 が保持する Product Code 3 1 2 がレジストリ情報 3 2 0 に存在するかを解析する。レジストリ情報 3 2 0 の Product Code 3 2 2 に Product Code 3 0 2 が存在しない場合、または、レジストリ情報 3 2 0 が存在しない場合、未インストールのサービスパックが存在すると判定し、S 1 0 0 4 へ進む。一方、レジストリ情報 3 2 0 の Product Code 3 2 2 に Product Code 3 0 2 が存在する場合、さらに Installer Version 3 1 1 と Product Version 3 2 1 とを比較する。ここでは、例えば、それぞれのバージョンの値を比較することとなる。比較した結果、Installer Version 3 1 1 の方が大きい場合、未インストールのサービスパックが存在すると判定し、S 1 0 0 4 へ進む。一方、Installer Version 3 1 1 の方が小さい或いは同じ場合、未インストールのサービスパックが存在しないと判定し、S 1 0 0 5 へ進む。

10

【 0 0 5 7 】

次に、S 1 0 0 4 にて、制御部 2 0 2 はサービスパックのアプリケーションソフトウェアのインストール実行フラグを ON にする。

【 0 0 5 8 】

次に、S 1 0 0 5 にて、制御部 2 0 2 はインストールするアプリケーションソフトウェアが 1 つ以上存在するかを判定する。ここで、S 1 0 0 5 における判定のロジックには「ベースのアプリケーションソフトウェアのインストール実行フラグ」と「サービスパックのアプリケーションソフトウェアのインストール実行フラグ」とを用いる。少なくともいずれかのフラグが ON であれば、何らかのアプリケーションソフトウェアをインストールすると判定する。はい（存在する）の場合、S 1 0 0 6 へ進み、いいえ（存在しない）の場合、本フローチャートの処理を終了する。

20

【 0 0 5 9 】

次に、S 1 0 0 6 にて、制御部 2 0 2 はベースのアプリケーションソフトウェアのインストール実行フラグが ON かを判定する。はい（インストール実行フラグが ON ）の場合、S 1 0 0 7 へ進み、いいえ（インストール実行フラグが OFF ）の場合、S 1 0 0 8 へ進む。

【 0 0 6 0 】

次に、S 1 0 0 7 にて、制御部 2 0 2 はアプリケーションソフトウェアの基本機能をインストールする。S 1 0 0 7 で実行されるインストール処理は、アプリケーションソフトウェアのファイルコピーやレジストリ登録である。なお、S 1 0 0 7 で実行されるインストールの完了後には、制御部 2 0 2 はショートカットを作成しないものとする。後述するサービスパックのインストーラによる追加機能のインストールの完了後に、ショートカットが作成されることとなる。

30

【 0 0 6 1 】

次に、S 1 0 0 8 にて、制御部 2 0 2 はサービスパックのアプリケーションソフトウェアのインストール実行フラグが ON かを判定する。はい（インストール実行フラグが ON ）の場合、S 1 0 0 9 へ進み、いいえ（インストール実行フラグが OFF ）の場合、S 1 0 1 0 へ進む。

40

【 0 0 6 2 】

次に、S 1 0 0 9 にて、制御部 2 0 2 はサービスパックをインストールするためのインストーラ 2 1 0 を起動させ、インストーラ 2 1 0 による処理（図 4（b））を実行する。つまり S 1 0 0 9 では、前記アプリケーションソフトウェアのインストールに際して、該インストールされたアプリケーションソフトウェアの起動のためのユーザ入力を受け付け可能にする処理を実行することなく、制御部 2 0 2 はインストーラ 2 1 0 を起動する。

【 0 0 6 3 】

次に、S 1 0 1 0 にて、制御部 2 0 2 は OS のデスクトップ画面上にアプリケーションソフトウェアのショートカットを作成し、インストーラ 2 0 0 の処理を終了する。

【 0 0 6 4 】

50

図 4 (b) は、サービスパックのインストーラであるインストーラ 2 1 0 による処理の流れを示すフローチャートである。

【 0 0 6 5 】

S 1 0 0 9 でインストーラ 2 1 0 がインストーラ 2 0 0 によって起動されると、S 1 1 0 1 にて、制御部 2 1 2 はアプリケーションソフトウェアの追加機能をインストールする。S 1 1 0 1 で実行されるインストール処理は、アプリケーションソフトウェアのファイルコピーやレジストリ登録の処理であって、ショートカット作成の処理を含まないものとする。

【 0 0 6 6 】

次に、S 1 1 0 2 にて、制御部 2 1 2 は O S のデスクトップ画面上にアプリケーションソフトウェアのショートカットが存在するかを判定する。はい (存在する) の場合、インストーラ 2 1 0 の処理を終了する。いいえ (存在しない) の場合、S 1 1 0 3 へ進む。

【 0 0 6 7 】

次に、S 1 1 0 3 にて、制御部 2 1 2 は O S のデスクトップ画面上にアプリケーションソフトウェアのショートカットを作成し、インストーラ 2 1 0 の処理を終了する。

【 0 0 6 8 】

なお、S 1 1 0 1 でのサービスパックのインストール処理が完了した後、サービスパックのインストーラではなくベースのインストーラがショートカットを作成しても良い。

【 0 0 6 9 】

また、事前にベースのアプリケーションソフトウェアのみをインストール済みの場合、すでに O S のデスクトップ画面上にアプリケーションソフトウェアのショートカットが作成されている。しかしながら、ユーザが O S のデスクトップ画面上からショートカットを削除していることがある。そのため、サービスパックのインストーラが実行する S 1 1 0 2 、S 1 1 0 3 の処理で、ベースのインストーラが作成していたショートカットを修復する必要があるかを判定し、必要があれば修復するようにしてもよい。

【 0 0 7 0 】

以上、インストーラ構成に応じたショートカット作成タイミングの動的な変更手法について説明した。これにより、ベース用とサービスパック用の 2 つのインストーラが連続で実行される場合、ベースのインストーラでショートカットを作成せずに、サービスパックのインストーラでショートカットを作成できるようになる。従って、インストーラの構成に関わらず、ショートカット作成のタイミングが一律してインストール処理の最後に実施されるようになり、ユーザがインストール中にショートカットからアプリケーションソフトウェアを起動することがなくなる。その結果、インストール中におけるアプリケーションソフトウェアの起動操作を抑止することができるようになる。

【 0 0 7 1 】

[実施例 2 における各インストーラの処理]

実施例 2 では、第一のインストーラ 2 0 0 と第二のインストーラ 2 1 0 が連続実行される場合における課題の解決手法を説明する。なお、実施例 2 では第一のインストーラ 2 0 0 をベースのインストーラ、第二のインストーラ 2 1 0 をオプション機能のインストーラとして扱う。

【 0 0 7 2 】

実施例 2 における課題を説明する。オプション機能のインストーラとはベースのアプリケーションソフトウェアに対し、オプション機能をインストールできるインストーラである。一般的にオプション機能のインストーラは、ベースに対するオプション機能をインストールするだけであり、ベースのアプリケーションソフトウェアのショートカットを作成する処理は存在しない。そのため、実施例 1 の課題と同様に、ベース用とオプション機能用の 2 つのインストーラが連続で実行される場合、オプション機能が有効になる前にユーザがアプリケーションソフトウェアを起動してしまう可能性がある。

【 0 0 7 3 】

実施例 2 では前述の課題を解決するためのインストーラ構成に応じたショートカット作成

10

20

30

40

50

タイミングの動的な変更手法について説明する。

【 0 0 7 4 】

[実施例 2 における各インストーラの処理]

図 5 は、ベースのインストーラ 2 0 0 およびオプション機能のインストーラ 2 1 0 の処理のフロー図である。図 1 の情報処理装置 1 0 0 上で動作するインストーラ 2 0 0 ・ 2 1 0 を例に処理の流れを説明する。本フローでは、インストーラにおける動的なショートカット作成処理を実施する手法を示す。また、S 1 0 0 1 ~ S 1 0 1 0 の各ステップはインストーラ 2 0 0 による処理であり、S 1 1 0 1 ~ S 1 1 0 3 の各ステップはインストーラ 2 1 0 による処理である。なお、図 4 に示したフローチャートのステップと重複する部分についての説明を省略する。

10

【 0 0 7 5 】

図 5 (a) は、ベースのインストーラであるインストーラ 2 0 0 による処理の流れを示すフローチャートである。

【 0 0 7 6 】

S 2 0 0 3 にて、複数インストーラ形式のインストーラセットに未インストールのオプション機能が存在するかを判定する。はい (存在する) の場合、S 2 0 0 4 へ進み、いいえ (存在しない) の場合、S 2 0 0 5 へ進む。

【 0 0 7 7 】

S 2 0 0 3 における判定のロジックについて具体的に説明する。初めに制御部 2 0 2 は、オプション機能のインストーラ情報 3 1 0 を取得する。ここでインストーラ情報 3 1 0 が取得できない場合、オプション機能のインストーラ 2 1 0 は存在しないと判定し、S 2 0 0 5 へ進む。次に制御部 2 0 2 は、レジストリ情報 3 2 0 を取得する。そして制御部 2 0 2 は、オプション機能のインストーラ情報 3 1 0 とレジストリ情報 3 2 0 を基に、インストーラ情報 3 1 0 が保持する `Product Code 3 1 2` がレジストリ情報 3 2 0 に存在するかを解析する。

20

【 0 0 7 8 】

レジストリ情報 3 2 0 の `Product Code 3 2 4` に `Product Code 3 0 2` が存在しない場合、または、レジストリ情報 3 2 0 が存在しない場合、未インストールのオプション機能が存在すると判定し、S 2 0 0 4 へ進む。一方、レジストリ情報 3 2 0 の `Product Code 3 2 4` に `Product Code 3 0 2` が存在する場合、インストーラ情報 3 1 0 の `Installer Version 3 1 1` とレジストリ情報 3 2 0 の `Product Version 3 2 3` とを比較する。ここでは、例えば、それぞれのリビジョンの値を比較することとなる。比較した結果、`Installer Version 3 1 1` の方が大きい場合、未インストールのオプション機能が存在すると判定し、S 2 0 0 4 へ進む。一方、`Installer Version 3 1 1` の方が小さい或いは同じ場合、未インストールのオプション機能が存在しないと判定し、S 2 0 0 5 へ進む。

30

【 0 0 7 9 】

次に、S 2 0 0 4 にて、制御部 2 0 2 はオプション機能のアプリケーションソフトウェアのインストール実行フラグを ON にする。

【 0 0 8 0 】

次に、S 2 0 0 5 にて、制御部 2 0 2 はインストールするアプリケーションソフトウェアが 1 つ以上存在するかを判定する。ここで、S 2 0 0 5 における判定には「ベースのアプリケーションソフトウェアのインストール実行フラグ」と「オプション機能のアプリケーションソフトウェアのインストール実行フラグ」を用いる。何れかのフラグが ON であれば、何らかのアプリケーションソフトウェアをインストールすると判定される。はい (存在する) の場合、S 1 0 0 6 へ進み、いいえ (存在しない) の場合、インストーラ 2 0 0 の処理を終了する。

40

【 0 0 8 1 】

次に、S 2 0 0 8 にて、制御部 2 0 2 はオプション機能のアプリケーションソフトウェアのインストール実行フラグが ON かを判定する。はい (インストール実行フラグが ON)

50

の場合、S 2 0 0 9 へ進み、いいえ（インストール実行フラグがOFF）の場合、S 2 0 1 0 へ進む。

【0082】

次に、S 2 0 0 9 にて、制御部 2 0 2 はオプション機能のアプリケーションソフトウェアをインストールするためのインストーラ 2 1 0 を起動させ、インストーラ 2 1 0 による処理（図 5（b））を実行する。

【0083】

次に、S 2 0 1 0 にて、制御部 2 0 2 はOSのデスクトップ画面上にアプリケーションソフトウェアのショートカットを作成し、インストーラ 2 0 0 の処理を終了する。

【0084】

図 5（b）は、オプション機能のインストーラであるインストーラ 2 1 0 による処理の流れを示すフローチャートである。

【0085】

S 2 0 0 9 でインストーラ 2 1 0 がインストーラ 2 0 0 によって起動されると、S 2 1 0 1 にて、制御部 2 1 2 はベースに対して適用可能なオプション機能をインストールし、インストーラ 2 1 0 の処理を終了する。S 2 1 0 1 で実行されるインストール処理は、アプリケーションソフトウェアのファイルコピーやレジストリ登録の処理であって、ショートカット作成の処理を含まないものとする。S 2 1 0 1 でのオプション機能のインストール処理が完了した後、S 2 0 1 0 でインストーラ 2 0 0 によってショートカットが作成される。

【0086】

以上、インストーラ構成に応じたショートカット作成タイミングの動的な変更手法について説明した。これにより、ベース用とオプション機能用の2つのインストーラが連続で実行される場合、オプション機能のインストーラが終了した後にベースのインストーラでショートカットを作成できるようになる。従って、インストーラの構成に関わらず、ショートカット作成のタイミングが一律してインストール処理の最後の実施されるようになり、ユーザがインストール中にショートカットからアプリケーションソフトウェアを起動することがなくなる。その結果、インストール中におけるアプリケーションソフトウェアの起動操作を抑止することができるようになる。

【0087】

〔実施例3におけるインストーラの処理〕

実施例3では、第二のインストーラ 2 1 0 が単体実行される場合における課題の解決手法を説明する。なお、実施例3では第二のインストーラ 2 1 0 をサービスパックのインストーラとして扱う。

【0088】

実施例3における課題を説明する。事前にベースのアプリケーションソフトウェアがインストールされている状態では、OSのデスクトップ画面上にショートカットが作成済みになっている。この状態でサービスパックのインストールのみを単体で実行した場合、ユーザはサービスパックのインストーラ実行中にOSのデスクトップ画面上に存在するショートカットから容易にアプリケーションソフトウェアが起動できる。そのため、実施例1や実施例2の課題と同様にサービスパックのインストーラを単体で実行する場合、サービスパックのインストールが完了する前にユーザがアプリケーションソフトウェアを起動してしまう可能性がある。

【0089】

実施例3では、前述の課題を解決するためのショートカット存在の有無に応じたショートカット作成手法について説明する。

【0090】

〔実施例3における各インストーラの処理〕

図6は、サービスパックのインストーラ 2 1 0 の処理の流れを示すフローチャートである。図1の情報処理装置 1 0 0 上で動作するインストーラ 2 1 0 を例に処理の流れを説明す

10

20

30

40

50

る。本フローでは、インストーラにおけるショートカット存在の有無に応じたショートカット作成処理を実施する手法を示す。

【 0 0 9 1 】

インストーラ 2 1 0 は起動処理が開始されると、S 3 0 0 1 にて、制御部 2 1 2 は O S のデスクトップ画面上にアプリケーションソフトウェアのショートカットが存在するかを判定する。はい（存在する）の場合、S 3 0 0 2 へ進み、いいえ（存在しない）の場合、S 3 0 0 3 へ進む。

【 0 0 9 2 】

次に、S 3 0 0 2 にて、制御部 2 1 2 は O S のデスクトップ画面上に存在するアプリケーションソフトウェアのショートカットを削除する。

10

【 0 0 9 3 】

次に、S 3 0 0 3 にて、制御部 2 1 2 はベースに対して適用可能なサービスパックのアプリケーションソフトウェアをインストールする。S 3 0 0 3 で実行されるインストール処理は、アプリケーションソフトウェアのファイルコピーやレジストリ登録の処理であり、ショートカット作成の処理を含まないものとする。

【 0 0 9 4 】

次に、S 3 0 0 4 にて、制御部 2 1 2 は O S のデスクトップ画面上にアプリケーションソフトウェアのショートカットを作成し、インストーラ 2 1 0 の処理を終了する。

【 0 0 9 5 】

なお、S 3 0 0 2 でショートカットを削除するのではなく、アプリケーションを起動できないことを示すようなショートカットの表示に一時的に変更しても良い。

20

【 0 0 9 6 】

また、実施例 3 におけるインストーラ 2 1 0 は、サービスパックのインストーラに限ったものとしなない。また、実施例 3 におけるソフトウェア構成は、情報処理装置 1 0 0 には第一のインストーラ 2 0 0 を含まずに、第二のインストーラ 2 1 0 のみを含んでもよい（不図示）。

【 0 0 9 7 】

以上、ショートカット存在の有無に応じたショートカット作成手法について説明した。これにより、サービスパックのインストーラが単体で実行された場合、サービスパックのインストール中はショートカットが削除され、インストール処理の最後にショートカットが作成されるようになる。そのため、ユーザがサービスパックのインストール中にショートカットからアプリケーションソフトウェアを起動することがなくなる。その結果、インストール中におけるアプリケーションソフトウェアの起動操作を抑止することができるようになる。

30

【 0 0 9 8 】

（他の実施例）

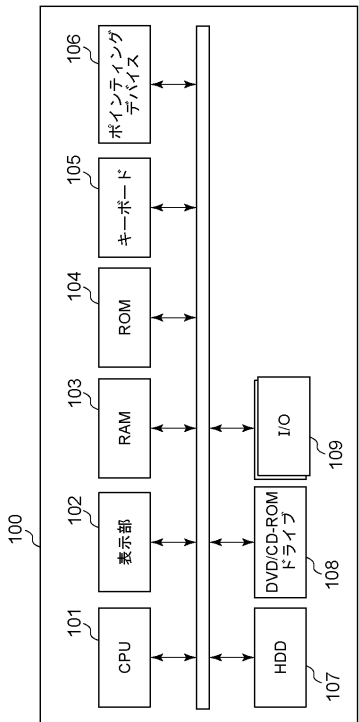
本発明は、上述した実施形態を適宜組み合わせることにより構成された装置あるいはシステムやその方法も含まれるものとする。

【 0 0 9 9 】

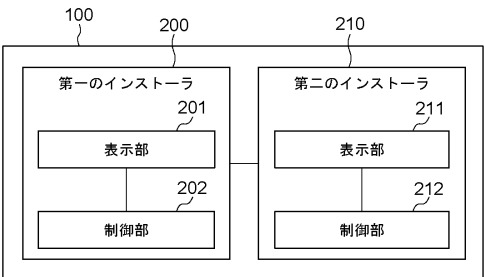
ここで、本発明は、上述した実施形態の機能を実現する 1 つ以上のソフトウェア（プログラム）を実行する主体となる装置あるいはシステムである。また、その装置あるいはシステムで実行される上述した実施形態を実現するための方法も本発明の 1 つである。また、そのプログラムは、ネットワークまたは各種記憶媒体を介してシステムあるいは装置に供給され、そのシステムあるいは装置の 1 つ以上のコンピューター（C P U や M P U 等）によりそのプログラムが読み出され、実行される。つまり、本発明の 1 つとして、さらにそのプログラム自体、あるいは当該プログラムを格納したコンピューターにより読み取り可能な各種記憶媒体も含むものとする。また、上述した実施形態の機能を実現する回路（例えば、A S I C）によっても、本発明は実現可能である。

40

【図面】
【図 1】



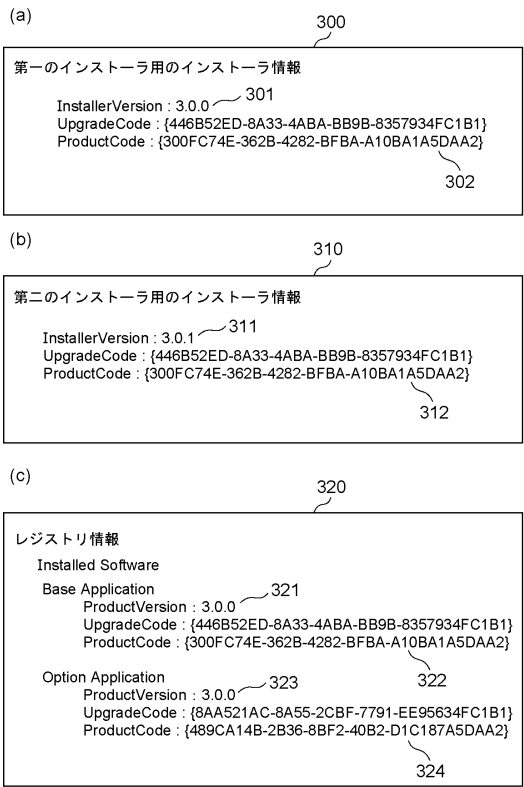
【図 2】



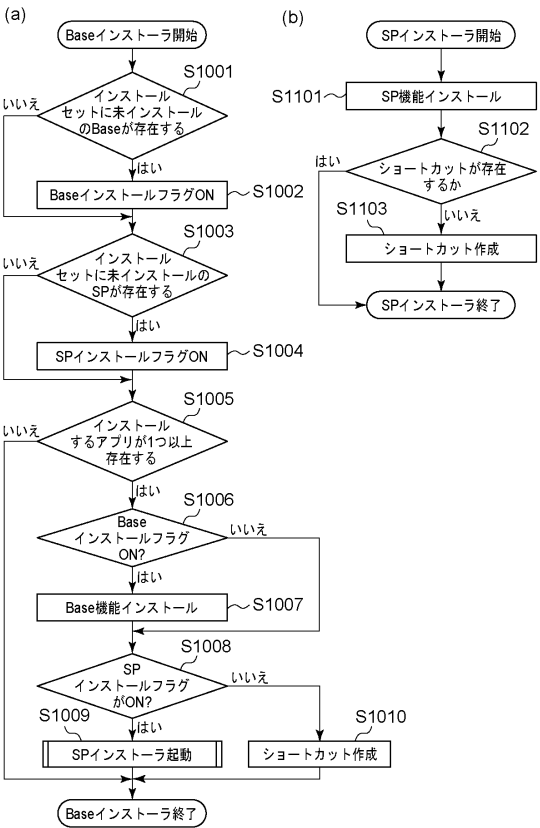
10

20

【図 3】



【図 4】

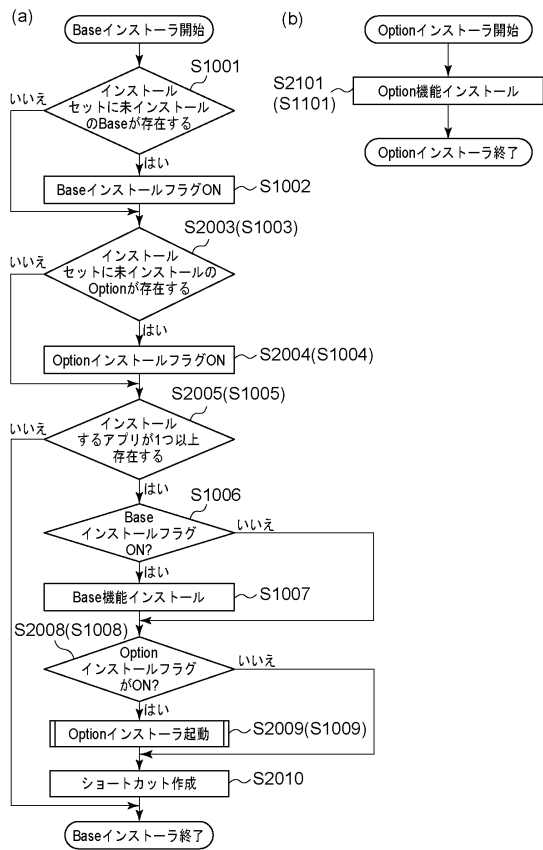


30

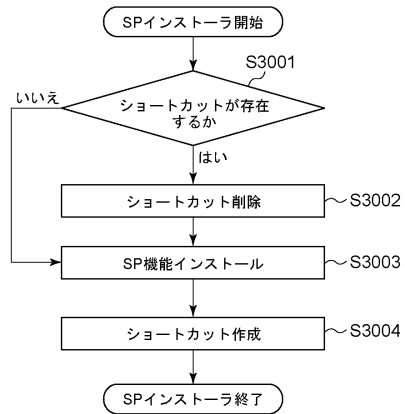
40

50

【図 5】



【図 6】



10

20

30

40

50

フロントページの続き

- (56)参考文献 特開 2 0 1 3 - 2 1 4 2 4 7 (J P , A)
米国特許第 0 8 9 1 0 1 4 0 (U S , B 1)
- (58)調査した分野 (Int.Cl. , D B 名)
- | | | |
|---------|-----------|-------------|
| G 0 6 F | 8 / 0 0 | - 8 / 3 8 |
| G 0 6 F | 8 / 6 0 | - 8 / 7 7 |
| G 0 6 F | 9 / 4 4 | - 9 / 4 4 5 |
| G 0 6 F | 9 / 4 5 1 | |