

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6889617号
(P6889617)

(45) 発行日 令和3年6月18日(2021.6.18)

(24) 登録日 令和3年5月25日(2021.5.25)

(51) Int.Cl.	F I				
G06F 9/445 (2018.01)	G06F	9/445	1	3	0
G06F 8/65 (2018.01)	G06F	8/65			
G06F 3/12 (2006.01)	G06F	3/12	3	3	0
B41J 29/38 (2006.01)	B41J	29/38	2	0	2
H04N 1/00 (2006.01)	H04N	1/00			C
請求項の数 10 (全 15 頁) 最終頁に続く					

(21) 出願番号 特願2017-109317 (P2017-109317)
 (22) 出願日 平成29年6月1日(2017.6.1)
 (65) 公開番号 特開2018-205967 (P2018-205967A)
 (43) 公開日 平成30年12月27日(2018.12.27)
 審査請求日 令和2年5月27日(2020.5.27)

(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 110003281
 特許業務法人大塚国際特許事務所
 (72) 発明者 伊藤 万里雄
 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 多賀 実

最終頁に続く

(54) 【発明の名称】 情報処理装置、プログラム管理方法、及びプログラム

(57) 【特許請求の範囲】

【請求項1】

既存プログラムに対する更新版プログラムを前記既存プログラムとは異なるプログラムとしてインストールし、前記更新版プログラムのバージョンを示すバージョン情報を保存する更新手段と、

プログラムを実行する際に、前記バージョン情報を参照し、最新のバージョンのプログラムを実行する実行制御手段と、

前記更新版プログラムの実行を停止する操作に応じて、前記更新版プログラムを停止させ、前記更新版プログラムの前記バージョン情報を削除する削除手段と、
 を有することを特徴とする情報処理装置。

【請求項2】

請求項1に記載の情報処理装置であって、

前記削除手段により前記バージョン情報を削除した後、前記既存プログラムを実行することを特徴とする情報処理装置。

【請求項3】

請求項1又は2に記載の情報処理装置であって、

前記更新版プログラムの実行を開始する操作に応じて、前記更新手段により前記更新版プログラムのバージョンを示す前記バージョン情報を保存し、前記実行制御手段により前記更新版プログラムを実行することを特徴とする情報処理装置。

【請求項4】

請求項 3 に記載の情報処理装置であって、
外部装置に対して画面情報を提供するための Web サーバーをさらに有し、
前記更新版プログラムの実行を停止する操作及び開始する操作は、前記 Web サーバーにより提供される前記画面情報に基づいて前記外部装置で表示される画面上で行われることを特徴とする情報処理装置。

【請求項 5】

請求項 1 乃至 4 のいずれか一項に記載の情報処理装置であって、
前記既存プログラムおよび前記更新版プログラムは Web ブラウザーであることを特徴とする情報処理装置。

【請求項 6】

請求項 1 乃至 5 のいずれか一項に記載の情報処理装置であって、
外部装置からダウンロードしたプログラムのバージョンを示すバージョン情報と、前記プログラムに対応する前記既存プログラムまたは前記更新版プログラムのバージョン情報とを対応付けた対応情報を有し、
前記外部装置から前記プログラムをダウンロードした際には、前記対応情報を参照して、ダウンロードした前記プログラムに対応する前記既存プログラムまたは前記更新版プログラムを実行することを特徴とする情報処理装置。

【請求項 7】

請求項 6 に記載の情報処理装置であって、
前記対応情報は、前記更新版プログラムのインストールの際に更新されることを特徴とする情報処理装置。

【請求項 8】

請求項 6 または 7 に記載の情報処理装置であって、
前記既存プログラムまたは前記更新版プログラムは Web ブラウザーであり、前記外部装置からダウンロードされるプログラムは前記 Web ブラウザーによりダウンロードされるスクリプトであり、
前記スクリプトをダウンロードした際には、前記 Web ブラウザーのバージョンが前記スクリプトに対応していない場合には、対応するバージョンの Web ブラウザーを起動することを特徴とする情報処理装置。

【請求項 9】

既存プログラムに対する更新版プログラムを前記既存プログラムとは異なるプログラムとしてインストールし、前記更新版プログラムのバージョンを示すバージョン情報を保存する更新手段と、
プログラムを実行する際に、前記バージョン情報を参照し、最新のバージョンのプログラムを実行する実行制御手段と、
前記更新版プログラムの実行を停止する操作に応じて、前記更新版プログラムを停止させ、前記更新版プログラムの前記バージョン情報を削除する削除手段としてコンピュータを機能させるためのプログラム。

【請求項 10】

更新手段と実行制御手段と削除手段とを有する情報処理装置によるプログラム管理方法であって、
前記更新手段が、既存プログラムに対する更新版プログラムを前記既存プログラムとは異なるプログラムとしてインストールし、前記更新版プログラムのバージョンを示すバージョン情報を保存し、
前記実行制御手段が、プログラムを実行する際に、前記バージョン情報を参照し、最新のバージョンのプログラムを実行し、
前記削除手段が、前記更新版プログラムの実行を停止する操作に応じて、前記更新版プログラムを停止させ、前記更新版プログラムの前記バージョン情報を削除することを特徴とするプログラム管理方法。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】**【0001】**

本発明は、アプリケーションの更新/追加が可能な情報処理装置およびプログラム管理方法、プログラムに関する。

【背景技術】**【0002】**

PCなどの情報処理装置がネットワーク上のWebサーバーに接続され、Webサーバー上のWebアプリケーションにより提供される操作画面を、情報処理装置が備えるWebブラウザ上に表示することが知られている。

【0003】

この場合、まず情報処理装置のWebブラウザが、Webサーバー上のWebアプリケーションに対して操作画面を要求(リクエスト)する。そして、Webアプリケーションが情報処理装置からの要求に応じて、Webブラウザに操作画面を表示させるためのHTMLデータを情報処理装置に回答(レスポンス)する。情報処理装置のWebブラウザは、受信したHTMLデータを解析し、HTMLデータの記述に基づいた操作画面を表示する。

【0004】

更に、Webブラウザに表示された操作画面を介してユーザーが指示を入力すると、入力された指示をWebブラウザがWebサーバーに対して通知する。そして、この通知を受けたWebサーバー上のWebアプリケーションは、入力された指示に従って処理

【0005】

近年、スキャナやプリンタを備えたMFP(Multi Function Peripheral)と呼ばれる画像処理装置の中にも、上述したようなWebブラウザを備えているものがある。

【0006】

ところで、HTML5やECMAScript6、CSS3に代表されるWeb技術は、新しい仕様や新しい技術が常に検討されており、日々進化している。Webブラウザはそれらの新しい仕様に追随し、数多くの更新が行われており、それらの仕様変更や仕様追加に合わせて、Webアプリケーションの更新も活発である。一方で、業務用アプリなど、様々な要因で頻繁に更新することができないWebアプリケーションが存在する。それらのWebアプリケーションを利用する場合、Webブラウザは更新せずに古いままにしておくことがある。

【先行技術文献】**【特許文献】****【0007】**

【特許文献1】特開2016-103159号公報

【発明の概要】**【発明が解決しようとする課題】****【0008】**

しかしながら、日々進化するWebサイトを閲覧するには、Webサイトに対応した新しいWebブラウザが必要になる。そこで、Webアプリケーションのために古いWebブラウザを残しつつ、最新のWebサイトを閲覧する際は最新のWebブラウザを利用したい、という要望がある。しかし、最新のWebブラウザの更新プログラムをインストールしてしまうと、古いWebブラウザのプログラムは消えてしまう。特許文献1では、更新プログラムをインストールする際に、更新の信頼性を高めるために以前のプログラムをバックアップする例が記載されているが、更新が完了するとバックアップは削除される。再度古いWebブラウザを利用するには、古いWebブラウザをインストールし直す必要がある。このように、古いWebブラウザと新しいWebブラウザを共存させることができず、容易にそれら二つを切り替えることができないという課題があ

10

20

30

40

50

る。

【 0 0 0 9 】

本発明は上記従来例に鑑みて成されたもので、異なるバージョンのプログラムを簡単な操作で使い分けることができ、ユーザビリティを向上させる情報処理装置、プログラム管理方法及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【 0 0 1 0 】

上記目的を達成するために本発明は以下の構成を有する。すなわち、情報処理装置であって、

既存プログラムに対する更新版プログラムを前記既存プログラムとは異なるプログラムとしてインストールし、前記更新版プログラムのバージョンを示すバージョン情報を保存する更新手段と、

プログラムを実行する際に、前記バージョン情報を参照し、最新のバージョンのプログラムを実行する実行制御手段と、

前記更新版プログラムの実行を停止する操作に応じて、前記更新版プログラムを停止させ、前記更新版プログラムの前記バージョン情報を削除する削除手段とを有することを特徴とする。

【発明の効果】

【 0 0 1 1 】

本発明によれば、異なるバージョンのプログラムを簡単な操作で使い分けることができ、ユーザビリティを向上させることが可能になる。

【図面の簡単な説明】

【 0 0 1 2 】

【図 1】本発明の実施形態における M F P の接続されるシステムの全体図である。

【図 2】本発明の実施形態における M F P のハードウェア構成を示すブロック図である。

【図 3】本発明の実施形態における更新情報および追加アプリ保存部のソフトウェア構成を示す図である。

【図 4】本発明の実施形態における初期状態および更新済みの更新情報テーブルそれぞれを示す図である。

【図 5】本発明の実施形態における更新プログラムインストール時のシーケンスを示す図である。

【図 6】本発明の実施形態におけるプログラム起動部のプログラム起動のフローを示す図である。

【図 7】本発明の実施形態における追加アプリ管理画面を示す図である。

【図 8】本発明の実施形態における J a v a S c r i p t と W e b ブラウザーの対応表を示す図である。

【図 9】本発明の実施形態における W e b ブラウザーの自動切換えのシーケンスを示す図である。

【図 1 0】本発明の実施形態における W e b ブラウザーの自動切換えの判断に利用する J a v a S c r i p t ファイルの例である。

【図 1 1】実施形態 1 における停止指示および開始指示に応じた処理フローを示す図である。

【発明を実施するための形態】

【 0 0 1 3 】

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。

[実施形態 1]

<システム構成>

図 1 は、本実施形態の画像処理装置を用いたシステムの構成例である。本システムはネットワーク 1 0 0 を介して接続される画像処理装置である M F P 1 0 1 と、M F P 1 0 1 の外部装置である W e b サーバ 1 0 2 とを含むよう構成される。M F P 1 0 1 は、その

10

20

30

40

50

情報処理機能に着目すれば情報処理装置と呼ぶこともできる。ネットワーク100は各装置間で通信を行うための基盤であって、インターネットに接続されていても良い。MFP101はWebブラウザ機能を有する装置であれば、MFP(Multiple Function Peripheral)に限定されるものではない。Webサーバー102は、ネットワーク経由でMFP101と接続されており、Webコンテンツを、ネットワークを介して配信する機能を持っており、MFP101からのリクエストに応じてWebコンテンツを提供する。各装置はローカルネットワーク上に存在しても良いし、インターネットを越えて接続されていても良い。また、Webサーバー102はMFP101の機能としてMFP101の中に存在してもよい。

【0014】

図2は本実施例のMFP101、Webサーバー102のハードウェア構成の一例である。この構成例には、装置全体を制御するCPU201、CPU201のワークエリアを提供するRAM202、本実施形態に係る発明を実現するためのプログラムや様々な設定を記憶する記憶装置203(HDDやNVRAM等でもよい)が含まれる。さらに、ユーザーがコマンドの入力を行うユーザー入力装置204、画面表示を行うUI表示装置205、他機器とネットワークによる通信を行うネットワーク装置206、メインバス200が含まれるよう構成されている。尚、本実施形態では特に断らない限り、MFP101、Webサーバー102では、CPU201がメインバス200を介してRAM202、記憶装置203、ユーザー入力装置204、UI表示装置205、ネットワーク装置206を制御する。また、タッチパネルディスプレイのようにUI表示装置205がユーザー入力装置204を兼ねても良い。ただし、Webサーバー102は、ユーザー入力装置204とUI表示装置205を有していなくても良い。また、MFP101はその他にも画像読取装置や画像印刷装置がメインバス200に接続されていても良い。

【0015】

ソフトウェア構成

図3(a)は本実施例におけるMFP101のソフトウェア構成を説明するための図である。図3(a)に示す各機能部は、MFP101が有しているCPU201が制御プログラムを実行することにより実現される。MFP101は、Webブラウザ301、プログラム起動部302、追加アプリ管理部303、追加アプリ保存部304、更新情報記憶部305を有する。これら各機能部によりMFP101によるプログラム管理方法が実現される。

【0016】

Webブラウザ301は、任意のサーバーと通信を行うことによってWebコンテンツを取得し、取得したWebコンテンツを画面に表示するといった機能を有するものであればよい。

【0017】

プログラム起動部302は、MFP101を起動した際に一緒に起動されるプログラムで、起動後、Webブラウザ301を起動する役割を担う。本実施例では、Webブラウザを起動する例を示すが、汎用的に他のプログラムを起動する役割を持っていても良い。プログラム起動部302は、MFP101の起動時、または他のプログラムからWebブラウザ301を起動するように依頼された場合に、更新情報記憶部305に記憶されている更新情報テーブル(図4(a))を参照する。更新情報テーブル(図4(a))は、アプリケーションごとに、アプリ名401、既存アプリパス402、更新アプリパス403、既存アプリバージョン404、更新アプリバージョン405の情報を有するテーブルである。アプリ名はアプリケーションの識別情報であり、アプリパスはアプリケーションの所在、アプリバージョンはアプリケーションの版(バージョン)を識別(あるいは特定)するための情報である。プログラム起動部302は、アプリ名401がWebブラウザを示す「browser」となっている行の更新アプリパス403を参照し、値が格納されているかどうか判断する。値が格納されている場合は(図4(b)参照)、その更新アプリパス403の値が指し示す場所に最新版アプリケーションが存在することを意味する

10

20

30

40

50

。このとき、プログラム起動部302は既存アプリバージョン404と更新アプリバージョン405とを比較し、更新アプリバージョン405のほうが新しい場合は、プログラム起動部302は更新アプリパス403が指し示す場所に存在するWebブラウザを起動する。一方更新アプリバージョン405のほうが古い場合は、更新アプリパス403ではなく、既存アプリパス402のWebブラウザを起動する。これは、更新版アプリケーションが既存アプリケーションより古い可能性があるためである。一方で、更新アプリパス403に値が格納されていない場合は(図4(a))、更新プログラムがインストールされていないことを意味し、プログラム起動部302は、既存アプリパス402の場所に存在するWebブラウザを起動する。

【0018】

追加アプリ管理部303は、MFP101に対して追加アプリをインストール/アンインストールしたり、開始/停止を制御したりするプログラムである。追加アプリとは、MFP101のファームウェアに最初から含まれないアプリケーションで、あとから追加可能なプログラムのことを指す。たとえば、MFP101がJava(登録商標)で記述したアプリケーションを実行するためのプラットフォームを有する場合、追加アプリはJava(登録商標)アプリとして作られ、追加アプリ管理部303はJavaVM及びJavaアプリの管理モジュールであることが考えられる。追加アプリ管理部303は、アプリ管理画面(図7)をUI表示装置205に表示でき、利用者はアプリ管理画面(図7)を通じて、追加アプリのインストール/アンインストールや、開始/停止の制御が可能である。本実施例では、MFP101はWebサーバーも有しており、アプリ管理画面(図7)は、MFP101にWebアプリケーションとして実装されていることを想定している。管理者あるいはユーザーは、ネットワーク100に接続されているPC(不図示)上のWebブラウザか、もしくはデバイス内部のWebブラウザ301からMFP101のWebサーバーにアクセスして追加アプリ管理部303を利用する。

【0019】

アプリ管理画面(図7)には、インストールされている追加アプリごとにアプリ状態701とアプリ制御ボタン702とが含まれており、さらに、追加アプリをインストールするために追加アプリを参照する参照ボタン703が含まれる。アプリ状態701は、インストールされている追加アプリが動作中か停止中かを表示する。アプリ制御ボタン702は追加アプリの開始/停止を切り替えるボタンである。参照ボタン703は、追加アプリをインストールする際に利用するボタンで、押下することでPC(不図示)に保存されている追加アプリを選択し、インストールすることができる。追加アプリ管理部303は、アプリ管理画面(図7)の参照ボタン703を通じて追加アプリのインストールを受け付けると、受け付けた追加アプリを追加アプリ保存部304に保存する。また、追加アプリ管理部303は、追加アプリのインストールと同時に追加アプリを開始する。

【0020】

追加アプリ保存部304は、追加アプリ管理部302を通して追加アプリがインストールされる際に、追加アプリを保存する領域であり、Webブラウザ301が保存されている領域とは異なる領域である。従って、追加アプリとしてWebブラウザを更新するための更新プログラムがインストールされた場合、更新プログラム310は追加アプリ保存部304に保存されて既存のWebブラウザ301を上書きはしない。一方で、Webブラウザ機能を含む或いは利用する追加アプリが複数インストールされた場合、新しいバージョンの追加アプリによって古いバージョンの追加アプリを上書きしても良いし、並列に保存した後、排他動作させても良い。

【0021】

追加アプリとしてWebブラウザの更新プログラムがインストールされた場合の、追加アプリ保存部304の構成を図3(b)に示す。追加アプリ保存部304には、更新プログラム310が保存されている。更新プログラム310は、更新対象のプログラムを更新するための更新制御モジュール311と、更新対象のプログラムの新たなバージョンである更新版プログラム(本例では更新版Webブラウザ312)で構成されている。追

10

20

30

40

50

加アプリ管理部 303 を通じて、更新プログラム 310 が開始されると更新制御モジュール 311 が実行される。更新制御モジュール 311 は、更新版 Web ブラウザー 312 の配置されているパスとバージョンの情報とを更新情報記憶部 305 の更新情報テーブルに記憶する。更新版 Web ブラウザー 312 の配置されているパスは更新アプリパス 403 に、更新版 Web ブラウザー 312 のバージョン情報は更新アプリバージョン 405 に格納される。これは、アプリ管理画面 (図 7) のアプリ制御ボタン 702 (「開始する」ボタン) によって追加アプリが開始された場合も同様である。すなわち、停止状態にある更新プログラム 310 を対象プログラムとして「開始する」ボタンが押下されると、更新プログラム 310 の更新制御モジュール 311 が実行されて上述したように更新版 Web ブラウザー 312 の配置されているパスは更新アプリパス 403 に、更新版 Web ブラウザー 312 のバージョン情報は更新アプリバージョン 405 に格納される。

10

【0022】

一方で、アプリ管理画面 (図 7) のアプリ制御ボタン 702 (「停止する」ボタン) によって追加アプリが停止された場合は、更新制御モジュール 311 は、更新情報記憶部 305 の更新情報テーブルから、停止されたアプリの更新アプリパス 403、更新アプリバージョン 405 の情報を削除する。図 4 (a) は、更新版 Web ブラウザー 312 をはじめ追加アプリがインストールされていないか、あるいは停止されている状態の更新情報テーブルを示す。一方図 4 (b) は、バージョン 2.5.5 の更新版 Web ブラウザー 312 がインストールされ、かつ動作している状態の更新情報テーブルを示す。

【0023】

なお、同じプログラム例えば Web ブラウザーについて、複数のバージョンの更新プログラム 310 がインストールされている場合にも、上記手順で実行するバージョンを選択できる。すなわち、バージョンごとの更新プログラムがそれぞれ別々にインストールされるので、利用したいバージョンの更新プログラムをアプリ管理画面から選択して「開始する」を選択すればよい。ただしこの場合には、いずれかのバージョンを排他的に実行させるために、「開始する」が選択されたならそのバージョンを除く更新プログラムについて停止させ、その後で選択されたバージョンの更新プログラムを実行する。こうすることで、実行開始するものとして選択されていないバージョンの更新アプリパスおよび更新アプリバージョンを更新情報テーブルからまず削除し、その後で、実行するバージョンの更新アプリパスおよび更新アプリバージョンを登録できる。そして登録したバージョンの更新版プログラムたとえば Web ブラウザーが実行開始される。

20

30

【0024】

なお更新版でないアプリケーション (すなわち初めてインストールするアプリケーション) については、更新情報テーブルのアプリ名 401 および既存アプリパス 402 および既存アプリバージョン 404 がそのインストール時に登録される。

【0025】

更新情報記憶部 305 は、更新情報テーブル (図 4 (a)、図 4 (b)) を保存する記憶領域である。前述の通り、追加アプリ保存部 304 に保存されている更新プログラム 310 の更新制御モジュール 311 によって書き換えられる。

【0026】

更新プログラムのインストールのシーケンス

図 5 は本実施例における更新プログラムインストール時のシーケンスであり、アプリ管理画面 (図 7) を通じて追加アプリ管理部 303 に対して更新プログラム 310 のインストールが指示された場合の処理を示す。

ステップ S501 では、追加アプリ管理部 303 は、更新プログラム 310 を追加アプリ保存部 304 に保存し、更新プログラム 310 を開始する。

ステップ S502 では、追加アプリ保存部 304 に保存されている更新プログラム 310 の更新制御モジュール 311 が、更新情報記憶部 305 に、更新プログラム 310 のパスとバージョンとを更新情報テーブルに保存する。このとき更新プログラム 310 は、更新対象のアプリ名 (本例では「browser」という名前) を用いて、そのアプリ名と一

40

50

致するアプリ名の行を、更新情報テーブル(図4)から対象となるアプリを示すレコードとして決定している。該当するレコードがあれば、更新プログラム310は、そのレコードの更新アプリパス403と更新アプリバージョン405とに、更新版プログラムのパスとバージョンとを記録する。該当するレコードがなければ、更新プログラム310は、たとえばそのレコードの既存アプリパス402と既存アプリバージョン404とに、更新版プログラムのパスとバージョンとを記録してよい。これにより既存のプログラムがない状態でインストールされた更新版プログラムは、既存の(すなわちインストール済みの)プログラムとして更新情報テーブルに記録される。あるいは該当するレコードが無ければその旨をUI表示装置205に表示してユーザーに通知してもよい。

ステップS503では、追加アプリ保存部304に保存されている更新プログラム310の更新制御モジュール311が、プログラム起動部302に対して、Webブラウザの再起動を依頼する。

ステップS504では、プログラム起動部302は、Webブラウザ301を停止する。

ステップS505では、プログラム起動部302は、更新情報記憶部305に記憶されているWebブラウザの更新情報テーブル(図4)を参照する。

ステップS506では、プログラム起動部303は、ステップS505で参照した更新情報テーブルを元に、既存プログラムと更新プログラムのどちらを起動するかを判断する。ここでは更新情報テーブルに登録されたアプリの最新バージョンを実行するアプリとして決定する。本例では、Webブラウザの更新アプリパス403が存在し、更新アプリバージョン405が既存アプリバージョンより新しいため、更新アプリパス403に記載されているプログラムを起動することを決める。

ステップS507では、プログラム起動部302は、ステップS506の判断により決定したアプリ、本例では追加アプリ保存部304に保存されている更新プログラム310の更新版Webブラウザ312を起動する。

【0027】

以上のように、更新プログラムのインストール時に更新プログラムが起動する。また、MFP101の起動時には、更新プログラムの起動の流れはS505～S508と同じである。

【0028】

アプリ管理画面(図7)の「開始する」ボタンあるいは「停止する」ボタンに応じた手順については図11を参照して改めて説明する。

【0029】

起動対象アプリケーションの決定

図6は、プログラム起動部302における、既存アプリケーションと更新版アプリケーションのどちらを起動するか判断処理、すなわち起動対象のバージョンの決定処理(あるいは実行制御)を示すフローである。この手順は、ソフトウェア上はプログラム起動部302により、ハードウェア上はCPU201により実行される。この手順は図5のステップS505～S507に相当し、プログラム起動部302により実行される。

ステップS601では、プログラム起動部302は、更新情報記憶部305に記憶されている対象プログラムの更新情報を参照する。

ステップS602では、ステップS601で参照した更新情報に、更新アプリパス403が記載されているかどうかを確認する。更新アプリパスが記載されている場合は、ステップS603に進む。記載されていない場合はステップS605に進む。

ステップS603では、ステップS601で参照した更新情報において、更新アプリバージョンが、既存アプリバージョンより新しいかどうかを確認する。新しい場合はステップS604に移動する。そうでない場合は、ステップS605に移動する。

ステップS604では、プログラム起動部302は、ステップS601で参照した更新情報に含まれる更新アプリパス403に記載されているプログラムを起動する。

ステップS605では、プログラム起動部302は、ステップS601で参照した更新情

10

20

30

40

50

報に含まれる既存アプリパス402に記載されているプログラムを起動する。

【0030】

アプリケーションの開始および停止手順

図11に、図7のアプリ管理画面において「停止する」ボタン（以下、停止指示とも呼ぶ）が押下された場合の処理手順および、「開始する」ボタン（以下、開始指示とも呼ぶ）が押下された場合のMFP101により処理手順それぞれを示す。Webブラウザにより表示されたアプリ管理画面の停止指示に応じてMFP101のWebサーバーがその停止指示を受け付ける。停止指示はたとえば追加アプリ管理部303に引き渡されて、停止指示の対象プログラムを停止させる（S1101）。このとき対象プログラムが更新プログラム310である場合、そこに含まれる更新版プログラム（たとえば更新版Webブラウザ312）も停止する。続いて、対象プログラムの名称をアプリ名として持つ更新情報テーブルの更新アプリパス402および更新アプリバージョン405を削除する（S1102）。この後、図6のステップS601へ分岐し、更新前の既存プログラムを実行する。これにより、更新版プログラムは停止し、更新されていない、既存プログラムが実行される。

10

【0031】

一方、アプリ管理画面で開始指示が入力された場合、MFP101では、Webサーバーがその開始指示を受け付ける。開始指示に応じてたとえば追加アプリ管理部303は、開始指示の対象プログラムを実行させる。そのために、図5のステップS502から実行させる。ステップS502からの処理では、開始指示の対象プログラムが通常の追加アプリケーションであればそれを実行する。一方、対象プログラムが更新プログラム310であれば、更新プログラム310が実行されて、所在およびバージョンを、更新情報テーブルの更新アプリパス403及び更新アプリバージョン405へと登録し（S502）、プログラム起動部302により既存アプリを停止させる。その後プログラム起動部302すなわち図6の手順がステップS601から実行される。

20

【0032】

これにより、更新情報テーブルの更新アプリパス403および更新アプリバージョン405には更新プログラム310のパスおよびバージョンが登録されて、図6の手順が実行され、更新版プログラム（更新版Webブラウザ）が実行される。

【0033】

以上の構成及び手順により、Webブラウザに表示された画面上で更新アプリケーションの停止および開始を指示でき、既存バージョンと更新バージョンとを容易に切り替えることができる。

30

【0034】

[実施形態2]

ここまでに記載した方法では、ユーザーが、アプリ管理画面（図7）を用いて、更新プログラムを利用するか、既存プログラムを利用するかを設定する。そのため、ユーザーが必要に応じて、例えば図7のアプリ管理画面から随時切り替え操作をすることになる。本実施形態では、Webコンテンツに含まれるスクリプト例えばJavaScript（登録商標）ライブラリの違いによって、自動でアプリが対応するWebブラウザを判断して切り替える。なお本実施形態でも、実施形態1のように新たにインストールした更新プログラムについては、更新情報テーブル（図4）に登録され、また、停止または開始に応じて、更新プログラムの更新アプリパスおよび更新アプリバージョンが削除され、または登録されて、実行されているWebブラウザのバージョンを特定できる。

40

【0035】

スクリプトを用いた実行対象バージョンの決定

図8は、コンテンツに合わせて既存アプリケーションと更新版アプリケーションのうちのどれを起動するか判断する場合に利用する、JavaScript（登録商標）とWebブラウザのバージョンを対応付けたバージョン対応表あるいは対応情報である。バージョン対応表は、更新情報記憶部305に保存されており、更新プログラム310

50

のインストール時に、更新制御モジュール311によって更新される。バージョン対応表には、スクリプト名801、スクリプトバージョン802、対応ブラウザバージョン803が含まれる。対応ブラウザバージョン803は、スクリプト名801およびスクリプトバージョン802で特定されるスクリプトに対応するブラウザである。Webブラウザ301、または更新版Webブラウザ312がサーバー102のWebコンテンツに含まれるJavaScript(登録商標)をダウンロードした際に、ダウンロードしたスクリプトのスクリプト名およびスクリプトバージョンを取得する。プログラム起動部302はスクリプト名801、スクリプトバージョン802から、対応ブラウザバージョン803を参照する。そして、プログラム起動部302は、参照したブラウザバージョン803と現在動作しているWebブラウザのバージョンとを比較し、異なる

10

判断した場合は、現在動作しているWebブラウザを停止し、ブラウザバージョン803のWebブラウザを起動する。現在動作しているWebブラウザは、たとえば実施形態1の更新情報テーブル(図4)などを参照することで特定できる。またWebブラウザの起動の際には、更新情報テーブル(図4)を参照して、既存アプリバージョン404または更新アプリバージョン405にブラウザバージョン803と一致するものがあるか判定する。一致するものがあれば該当するバージョンの所在を一致したアプリバージョン対応するアプリパスで特定し、そのアプリ(Webブラウザ)を起動する。一致するものが無ければたとえば最新のバージョンを起動してよい。

【0036】

Webコンテンツに含まれるJavaScript(登録商標)の例を図10に示す。Webコンテンツに含まれるJavaScript(登録商標)ライブラリには、JavaScript(登録商標)ライブラリ名1001と、JavaScript(登録商標)ライブラリバージョン1002が含まれる。Webブラウザ301、または更新版Webブラウザ312がこれらを参照することによって、プログラム起動部302は、前記バージョン対応表から、対応するWebブラウザのバージョンを決定することができる。

20

【0037】

Webブラウザの切り替え

プログラム起動部302が、JavaScriptライブラリのバージョンによって、利用するWebブラウザを切り替える際のシーケンスを図9に示す。

30

ステップS901では、Webブラウザ301は、Webアプリケーションの画面情報を取得するためにWebサーバー102に対してHTTプリクエストを送信する

ステップS902では、Webサーバー102は、ステップS901で受信したリクエストに対するレスポンスとしてHTMLデータを送信する。

ステップS903では、Webブラウザ301は、ステップS902で受信したHTMLデータを解釈/実行することでJavaScript(登録商標)ライブラリを取得するためのHTTプリクエストを送信する。

ステップS904では、Webサーバー102は、ステップS903で受信したリクエストに対するレスポンスとしてJavaScript(登録商標)ライブラリを送信する。

40

ステップS905では、Webブラウザ301は、前記ステップS904で受信したJavaScript(登録商標)ライブラリから、ライブラリ名とライブラリのバージョンを取得する。

ステップS906では、Webブラウザ301は、プログラム起動部302に対してステップS905で取得したライブラリ名とライブラリのバージョン、およびステップS901でリクエストを送信したURLを通知する。

【0038】

ステップS907では、プログラム起動部302は、更新情報記憶部305に保存されているバージョン対応表から、ステップS906で通知されたライブラリ名とライブラリのバージョンを基に、対応ブラウザバージョンを取得する。本例では、更新版Webブラウザ312に対応するバージョンが得られたとして説明する。なおここでいう対応と

50

たとえば、プログラム間（たとえばスクリプトとWebブラウザとの間）のインターフェースが整合しており、正しく動作することなどをいう。

ステップS908では、プログラム起動部302は、ステップS907で取得した対応ブラウザバージョンを基に、どのWebブラウザを起動するかを決める。具体的には、現在起動しているWebブラウザは、未更新版のWebブラウザ301であるため、Webブラウザ301を停止し、更新版Webブラウザ312を起動することを決める。このとき、スクリプトに対応するバージョンのWebブラウザが存在しない場合は、その旨を通知するダイアログを表示しても良い。

ステップS909では、プログラム起動部302は、Webブラウザ301を停止する。

10

ステップS910では、プログラム起動部302は、スクリプトに対応するバージョンのWebブラウザ、本例では追加アプリ保存部304に保存されている更新版Webブラウザ312を起動する。このとき、プログラム起動部302は、ステップS906で受信したURLを更新版Webブラウザ312に通知する。

ステップS911では、更新版Webブラウザ312は、ステップS910で受信したURLに従って、Webサーバー102にリクエストを送信する。

【0039】

ステップS912では、Webサーバー102は、更新版Webブラウザ312にHTMLデータを送信する。

ステップS913では、更新版Webブラウザ312は、ステップS912で受信したHTMLデータを解釈/実行することでJavaScript（登録商標）ライブラリを取得するためのHTTPリクエストを送信する。

20

ステップS914では、Webサーバー102は、ステップS913で受信したリクエストに対するレスポンスとしてJavaScript（登録商標）ライブラリを送信する。

ステップS915では、更新版Webブラウザ312は、ステップS914で受信したJavaScript（登録商標）ライブラリを実行する。

【0040】

本実施例ではJavaScript（登録商標）の名前とバージョンから対応するWebブラウザのバージョンを特定しているが、方法はこれに限らない。たとえば、JavaScript（登録商標）ファイルのハッシュ値で特定しても良いし、JavaScript（登録商標）ファイルに含まれるJavaScript（登録商標）APIの種類やバージョンによって判断しても良い。この様にして本実施形態では、ダウンロードしたスクリプトに対応するWebブラウザを、ユーザーが切り替えの操作を行わなくとも実行できる。これにより一層操作性が向上する。

30

【0041】

以上に示す構成により、あるプログラムに対して複数の異なるバージョン、すなわち既存プログラムと更新プログラムとを共存させることができる。また、更新プログラムの開始、停止を容易に変更できるようになる。すなわち、あるプログラムの複数のバージョンのうちから選択したバージョンを実行できるようになる。あるいは、ダウンロードしたスクリプトに対応したバージョンを自動的に実行できる。これにより、一度インストールすれば、インストール/アンインストールを実施することなく、異なるバージョンのWebブラウザを使い分けることが可能になり、ユーザビリティが向上する。

40

【0042】

なお実施形態1および実施形態2では、既存プログラム及び更新版プログラムとしてWebブラウザを例としたが、Webブラウザ以外のプログラムについても適用できる。たとえばアプリケーションプログラムによりWebブラウザ機能を提供するために、WebViewというクラスが例えばJava（登録商標）などでは提供されている。このWebViewについても、たとえばWebViewのバージョンや、WebViewを含むライブラリのバージョンなどを用いて、上記実施形態を適用することもできる。

【0043】

50

[その他の実施例]

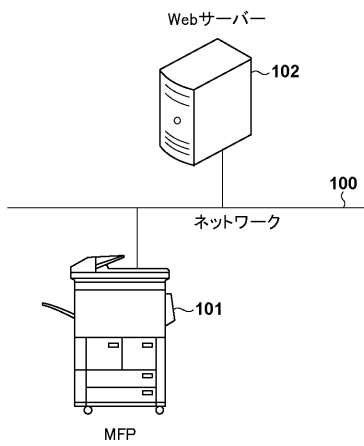
本発明は、上述の実施形態の1以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける1つ以上のプロセッサがプログラムを読み出し実行する処理でも実現可能である。また、1以上の機能を実現する回路(例えば、ASIC)によっても実現可能である。

【符号の説明】

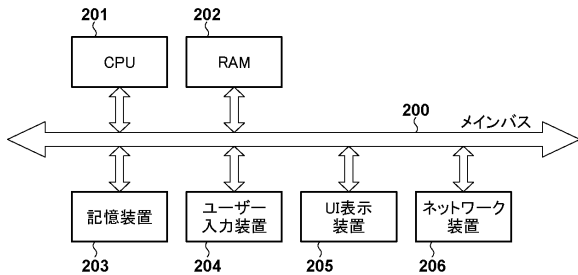
【0044】

301 Webブラウザ、302 プログラム起動部、303 追加アプリ管理部、304 追加アプリ保存部、305 更新情報記憶部

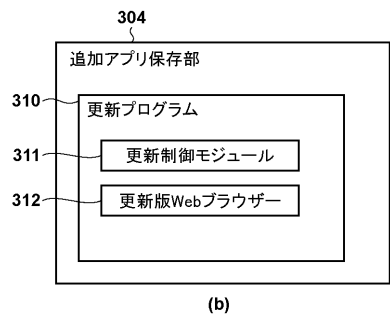
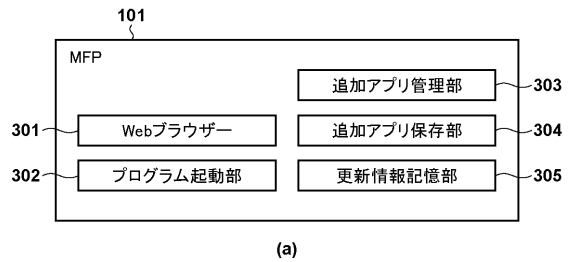
【図1】



【図2】



【図3】



【 図 4 】

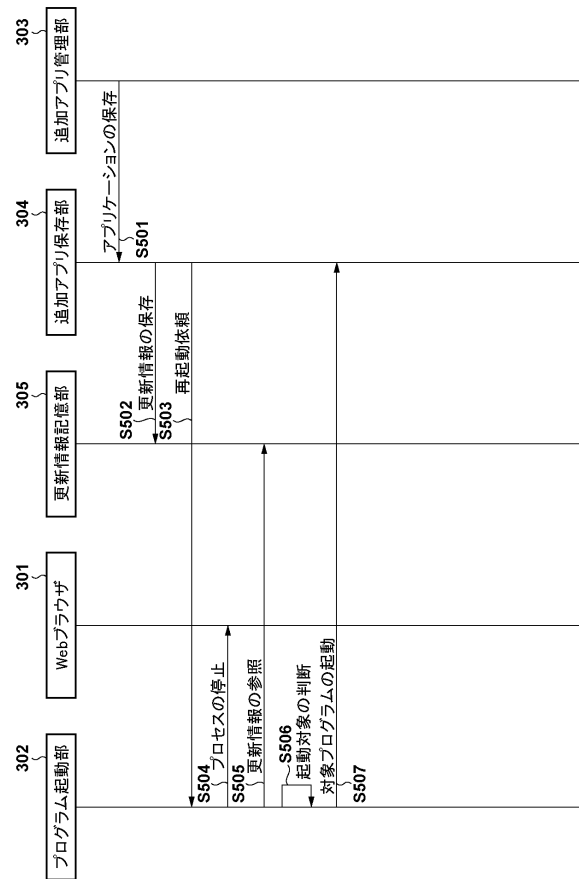
401	402	403	404	405
アプリ名	既存アプリパス	更新アプリパス	既存アプリバージョン	更新アプリバージョン
browser	/firmware/browser	-	2.1.1	-
appx	/firmware/appx	-	1.0.0	-

(a)

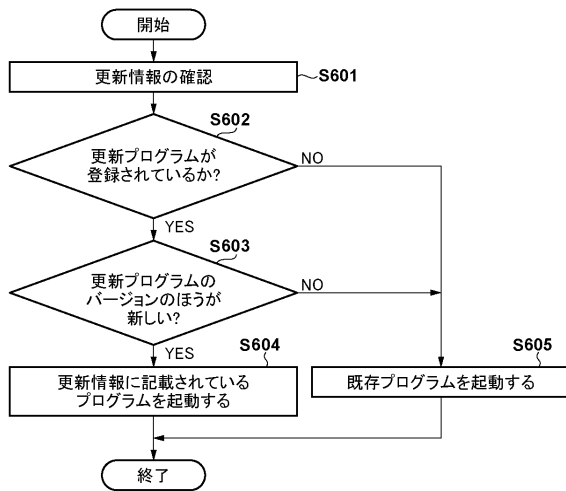
アプリ名	既存アプリパス	更新アプリパス	既存アプリバージョン	更新アプリバージョン
browser	/firmware/browser	/java/app1/browser	2.1.1	2.5.5
appx	/firmware/appx	-	1.0.0	-

(b)

【 図 5 】



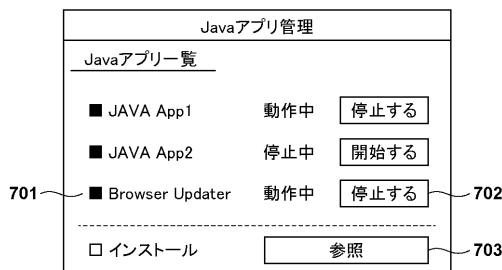
【 図 6 】



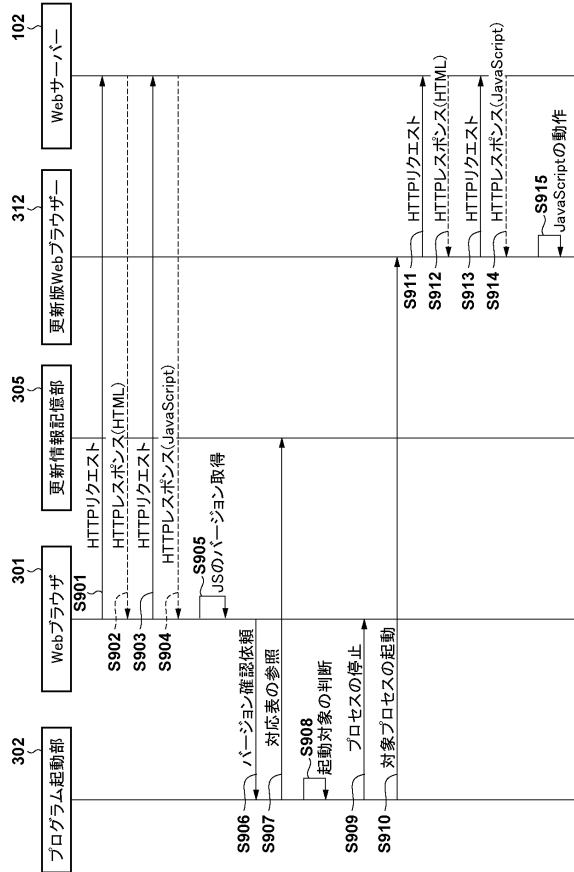
【 図 8 】

801	802	803
スクリプト名	スクリプトバージョン	対応ブラウザバージョン
cononbrws.js	1.2.0	2.0.0
cononbrws.js	1.3.0	3.0.0
Getinfo.js	2.1.0	2.2.0

【 図 7 】



【 図 9 】

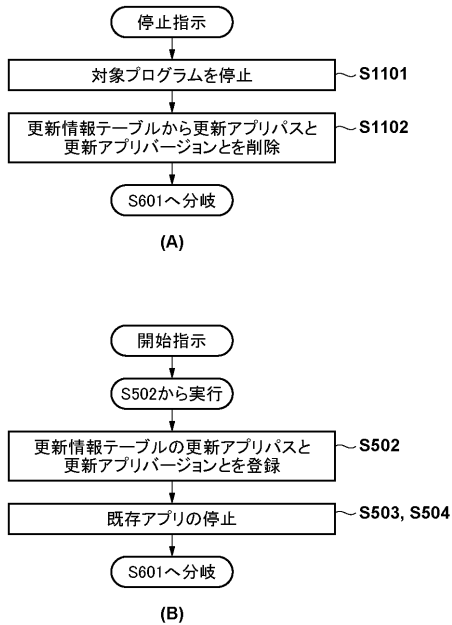


【 図 10 】

```

1001 /*
      *cononbrws.js
1002 *
      *@version: 1.3.0
      *-----
      */
      function getDeviceInfo( param ){
          //javascript logic
          .....
      }
  
```

【 図 11 】



フロントページの続き

(51)Int.Cl. F I
H 0 4 N 1/00 1 2 7 Z

(56)参考文献 特開2012-020546(JP,A)
特開2001-211488(JP,A)
米国特許出願公開第2013/0104060(US,A1)
特開昭64-88631(JP,A)
特表2012-530322(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 8 / 0 0 - 8 / 7 7
G 0 6 F 9 / 4 4 - 9 / 4 5 5
G 0 6 F 3 / 1 2
B 4 1 J 2 9 / 3 8
H 0 4 N 1 / 0 0