(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-208894
(P2005-208894A)
(43)公開日　平成17年8月4日(2005.8.4)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 9/445	G06F 9/06	610B	5B014
G06F 3/12	G06F 3/12	C	5B021
G06F 13/10	G06F 13/10	320A	5B076

審査請求 未請求　請求項の数 19　ＯＬ　（全 17 頁）

(21)出願番号　特願2004-14263(P2004-14263)
(22)出願日　平成16年1月22日(2004.1.22)

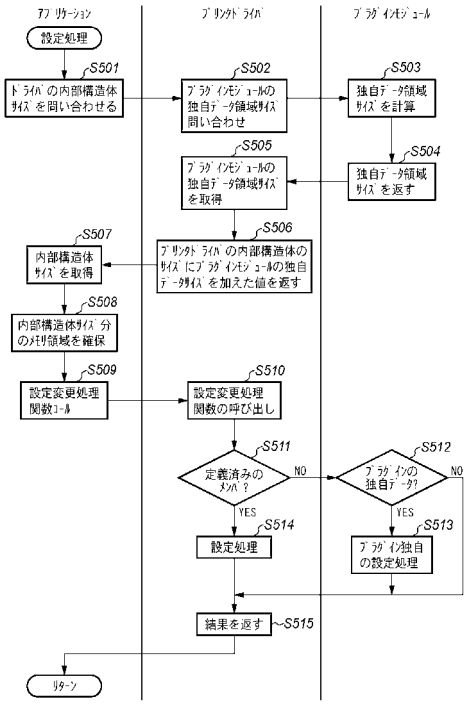
(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(72)発明者　加藤　央
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

Ｆターム(参考)　5B014 FA11

最終頁に続く

(54)【発明の名称】周辺装置制御方法、制御プログラム及びその装置

(57)【要約】
【課題】　デバイスドライバの機能拡張をプラグイン形式によって実現した環境において、アプリケーションからの拡張機能の設定／変更をデバイスドライバのユーザインタフェースを開かずに実現する。
【解決手段】　デバイスドライバの１つであるプリンタドライバと、該プリンタドライバから設定できないパラメータを有する機能拡張モジュールとを使って、アプリケーションから印刷処理を実行する場合に、アプリケーションが、プリンタドライバが管理するパラメータと機能拡張モジュールが管理するパラメータとを含む容量のメモリ領域を確保しS501-S508、アプリケーションから、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、プリンタドライバ又は機能拡張モジュールに渡してS509-S510、前記メモリ領域の前記識別子で特定されるパラメータに前記設定値を設定しS511-S514、前記メモリ領域のパラメータを使用して、プリンタドライバ及び機能拡張モジュールにより印刷処理を行う。
【選択図】　図６



【特許請求の範囲】
【請求項１】
周辺装置制御用のコマンドを生成するデバイスドライバと、該デバイスドライバから設定できないパラメータを有する機能拡張モジュールとを使って、アプリケーションから周辺装置制御処理を実行する周辺装置制御方法であって、
前記アプリケーションが、所定のインタフェースを用いて、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを操作するステップと、
前記操作するステップに応じて、前記デバイスドライバ又は機能拡張モジュールがパラメータを設定するステップと、
前記パラメータを使用して、前記デバイスドライバ及び機能拡張モジュールにより周辺装置制御処理を行うステップとを有することを特徴とする周辺装置制御方法。
【請求項２】
前記所定のインタフェースとは、デバイスドライバ又はオペレーティングシステムが公開しているインタフェースであることを特徴とする請求項１記載の周辺装置制御方法。
【請求項３】
前記アプリケーションが、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを含む容量のメモリ領域を確保するステップをさらに備え、
前記操作するステップは、
前記アプリケーションから、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバ又は機能拡張モジュールに渡すステップと、
前記デバイスドライバ又は機能拡張モジュールが、前記メモリ領域の前記識別子で特定されるパラメータに前記設定値を設定するステップとを実行することにより、
前記機能拡張モジュールが管理するパラメータを操作することを特徴とする請求項１又は２に記載の周辺装置制御方法。
【請求項４】
前記渡すステップでは、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値を、前記所定のインタフェースの引数として、前記アプリケーションから前記デバイスドライバ又は機能拡張モジュールに渡すことを特徴とする請求項３記載の周辺装置制御方法。
【請求項５】
前記設定値を設定するステップでは、前記識別子がデバイスドライバに管理されるパラメータの識別子か否かを判別し、前記識別子がデバイスドライバで管理しているパラメータの識別子ではないと判別された場合に、前記機能拡張モジュールで管理しているパラメータに対して前記設定値の設定を行うことを特徴とする請求項３又は４記載の周辺装置制御方法。
【請求項６】
前記識別子が、前記デバイスドライバで管理しているパラメータの識別子でなく、且つ前記機能拡張モジュールで管理しているパラメータの識別子でない場合に、アプリケーションに対して設定不可の通知をするステップを更に有することを特徴とする請求項３乃至５のいずれか１つに記載の周辺装置制御方法。
【請求項７】
前記周辺装置制御処理が終了した場合に、前記確保されたメモリ領域を解放するステップを更に有することを特徴とする請求項３乃至６のいずれかに記載の周辺装置制御方法。
【請求項８】
周辺装置制御用のコマンドを生成するデバイスドライバと、該デバイスドライバから設定できないパラメータを有する機能拡張モジュールとを使った周辺装置制御処理を含むアプリケーションプログラムであって、

前記デバイスドライバ及び機能拡張モジュールに、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを含む容量を問い合わせて、該容量のメモリ領域を確保するステップと、
少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバ又は機能拡張モジュールに渡すステップと、
前記識別子で特定されるパラメータに前記設定値を設定した前記メモリ領域のパラメータを使用して、前記デバイスドライバ及び機能拡張モジュールに周辺装置制御処理を行うよう指示するステップとを有することを特徴とするアプリケーションプログラム。
【請求項９】
前記周辺装置制御処理が終了した場合に、前記確保されたメモリ領域を解放するステップを更に有することを特徴とする請求項８記載のアプリケーションプログラム。
【請求項１０】
アプリケーションの周辺装置制御処理に応答して周辺装置制御用のコマンドを生成するデバイスドライバプログラムであって、
前記アプリケーションからの問い合わせに応答して、前記デバイスドライバプログラムが管理するパラメータと、前記デバイスドライバプログラムの機能拡張用の機能拡張モジュールが管理する前記デバイスドライバプログラムから設定できないパラメータとを含む容量を返すステップと、
少なくとも、前記アプリケーションにより確保された前記容量のメモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記アプリケーションから受け取るステップと、
前記識別子が前記デバイスドライバプログラムに管理されるパラメータの識別子か否かを判別するステップと、
前記識別子が前記デバイスドライバプログラムで管理しているパラメータの識別子であると判別された場合に、前記識別子で特定されるパラメータに前記設定値を設定し、前記識別子が前記デバイスドライバプログラムで管理しているパラメータの識別子ではないと判別された場合に、少なくとも、前記アプリケーションにより確保された前記容量のメモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記機能拡張モジュールに渡すステップと、
前記設定値を設定した前記メモリ領域のパラメータを使用して、周辺装置制御処理を行うステップとを有することを特徴とするデバイスドライバプログラム。
【請求項１１】
アプリケーションの周辺装置制御処理に応答して周辺装置制御用のコマンドを生成するデバイスドライバの機能拡張用の機能拡張モジュールプログラムであって、
前記デバイスドライバからの問い合わせに応答して、前記機能拡張モジュールプログラムが管理する前記デバイスドライバから設定できないパラメータの容量を返すステップと、
少なくとも、前記アプリケーションにより確保された前記容量のメモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバから受け取るステップと、
前記識別子が前記機能拡張モジュールプログラムに管理されるパラメータの識別子か否かを判別するステップと、
前記識別子が前記機能拡張モジュールプログラムで管理しているパラメータの識別子であると判別された場合に、前記識別子で特定されるパラメータに前記設定値を設定し、前記識別子が前記機能拡張モジュールプログラムで管理しているパラメータの識別子ではないと判別された場合に、設定不可の通知を前記デバイスドライバに返すステップと、
前記設定値を設定した前記メモリ領域のパラメータを使用して、周辺装置制御の機能拡張処理を行うステップとを有することを特徴とする機能拡張モジュールプログラム。
【請求項１２】
請求項８乃至１１のいずれか１つに記載のプログラムをコンピュータ読み出し可能に記

憶する記憶媒体。
【請求項１３】
周辺装置制御用のコマンドを生成するデバイスドライバと、該デバイスドライバから設定できないパラメータを有する機能拡張モジュールとを使って、アプリケーションから周辺装置制御処理を実行する周辺装置制御装置であって、
前記アプリケーションが、所定のインタフェースを用いて、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを操作する手段と、
前記操作する手段の操作に応じて、前記デバイスドライバ又は機能拡張モジュールがパラメータを設定する手段と、
前記パラメータを使用して、前記デバイスドライバ及び機能拡張モジュールにより周辺装置制御処理を行う手段とを有することを特徴とする周辺装置制御装置。
【請求項１４】
前記所定のインタフェースとは、デバイスドライバ又はオペレーティングシステムが公開しているインタフェースであることを特徴とする請求項１３記載の周辺装置制御装置。
【請求項１５】
前記アプリケーションが、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを含む容量のメモリ領域を確保する手段をさらに備え、
前記操作する手段は、
前記アプリケーションから、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバ又は機能拡張モジュールに渡す手段と、
前記デバイスドライバ又は機能拡張モジュールが、前記メモリ領域の前記識別子で特定されるパラメータに前記設定値を設定する手段とを有し、
前記機能拡張モジュールが管理するパラメータを操作することを特徴とする請求項１３又は１４に記載の周辺装置制御装置。
【請求項１６】
前記渡す手段は、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値を、前記所定のインタフェースの引数として、前記アプリケーションから前記デバイスドライバ又は機能拡張モジュールに渡すことを特徴とする請求項１５記載の周辺装置制御装置。
【請求項１７】
前記設定値を設定する手段は、前記識別子がデバイスドライバに管理されるパラメータの識別子か否かを判別し、前記識別子がデバイスドライバで管理しているパラメータの識別子ではないと判別された場合に、前記機能拡張モジュールで管理しているパラメータに対して前記設定値の設定を行うことを特徴とする請求項１５又は１６記載の周辺装置制御装置。
【請求項１８】
前記識別子が、前記デバイスドライバで管理しているパラメータの識別子でなく、且つ前記機能拡張モジュールで管理しているパラメータの識別子でない場合に、アプリケーションに対して設定不可の通知をする手段を更に有することを特徴とする請求項１５乃至１７のいずれか１つに記載の周辺装置制御装置。
【請求項１９】
前記周辺装置制御処理が終了した場合に、前記確保されたメモリ領域を解放する手段を更に有することを特徴とする請求項１５乃至１８のいずれかに記載の周辺装置制御装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、デバイスドライバに拡張機能を提供するための周辺装置制御方法、制御プログラム及びその装置に関するものである。

【背景技術】
【０００２】
従来、コンピュータ上で動作するアプリケーションプログラム（以下、アプリケーションと称す）が周辺装置の制御を行う場合、制御開始命令と共に、制御に必要な各種設定をデバイスドライバ等の周辺装置制御関連処理プログラムにパラメータとして引き渡すのが一般的な方法である。このように周辺装置制御に必要な各種設定をアプリケーションよりデバイスドライバにパラメータとして渡すことで、アプリケーションが意図した周辺装置の動作が得られるようになっている。例えば、コンピュータ上で動作するアプリケーションが印刷を実行する場合、印刷開始命令と共に、印刷に必要な各種設定をプリンタドライバ等の印刷関連処理プログラムにパラメータとして引き渡すことで、アプリケーションが意図した印刷出力を得られるようになっている（特許文献１参照）。
【０００３】
尚、以下の背景技術及び課題の説明では、技術内容や問題点をより明瞭に把握するため、本発明の一適応技術であるアプリケーションからの印刷処理について具体的に説明するが、その問題点や課題は、アプリケーションから周辺装置を制御する場合に共通のものであり、印刷処理に限定されるものではない。
【０００４】
上述のアプリケーションからの印刷処理では、アプリケーションが意図した印刷出力を得るために指定可能な設定値の種類の範囲は、どのプリンタでもサポートが可能と思われる用紙の給紙口の指定や両面/片面印刷の指定等の非常に限られた範囲のものであり、プリンタ又はプリンタドライバが提供していてもアプリケーションより直接指定のできないその他の有用な機能については、プリンタドライバのユーザインタフェースを開いてそこで指定をする必要があった。このため、企業の各種システムソフトウェアからの印刷出力に際して、プリンタドライバのユーザインタフェースを表示したがらないケースでは、プリンタ又はプリンタドライバの持つ機能が十分に利用できないという問題点があった。
【０００５】
そこで、昨今ではプリンタ又はプリンタドライバをフルに生かせるように、プリンタやプリンタドライバへの機種に依存する設定についても、アプリケーションから利用可能な特別なインタフェースを設けて、このインタフェースを利用した場合には、プリンタドライバのユーザインタフェースで設定する場合と同等な機能の設定が可能となるようなプリンタドライバが出現してきている。
【特許文献１】特開２００３－９１３８９
【発明の開示】
【発明が解決しようとする課題】
【０００６】
しかしながら、このようなインタフェースを、プラグイン形式のプリンタドライバの機能拡張を実現している環境で使用すると、以下のような問題が出てくる。プラグイン形式のプリンタドライバの機能拡張とは、プリンタベンダーから配布されているプリンタドライバのインストールセットとは別に、差分モジュール、つまりプラグインのみを提供して機能追加を実現する方法である。このプラグイン形式で機能拡張した場合には、拡張された機能に対応した設定項目が存在するが、プリンタドライバにとってこの設定項目が未知となる構造となっている。アプリケーションからプラグイン形式で追加した機能の設定項目を上記の特別なインタフェースを用いて変更しようとした場合、プリンタドライバは不明な設定項目への設定／変更要求と認識し、サポートしてない旨の結果をアプリケーションに対して返す。
【０００７】
従って、アプリケーションからプラグイン機能を設定／変更するには、プリンタドライバがプラグインの機能を考慮する必要が生じている。すなわち、アプリケーションから周辺装置制御におけるプラグイン機能を設定／変更するには、周辺装置制御用のデバイスドライバがプラグインの機能を考慮する必要が生じていることになる。

【０００８】
本発明は、上記従来の問題点に鑑みてなされたもので、デバイスドライバの機能拡張をプラグイン形式によって実現した環境において、アプリケーションからの拡張機能の設定／変更を、デバイスドライバのユーザインタフェースを開かずに実現できることを目的としている。
【課題を解決するための手段】
【０００９】
上記目的を達成するために、本発明は次のような構成からなる。
【００１０】
すなわち、周辺装置制御用のコマンドを生成するデバイスドライバと、該デバイスドライバから設定できないパラメータを有する機能拡張モジュールとを使って、アプリケーションから周辺装置制御処理を実行する周辺装置制御方法であって、前記アプリケーションが、所定のインタフェースを用いて、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを操作するステップと、前記操作するステップに応じて、前記デバイスドライバ又は機能拡張モジュールがパラメータを設定するステップと、前記パラメータを使用して、前記デバイスドライバ及び機能拡張モジュールにより周辺装置制御処理を行うステップとを有することを特徴とする。
【００１１】
ここで、前記所定のインタフェースとは、デバイスドライバ又はオペレーティングシステムが公開しているインタフェースである。また、前記アプリケーションが、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを含む容量のメモリ領域を確保するステップをさらに備え、前記操作するステップは、前記アプリケーションから、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバ又は機能拡張モジュールに渡すステップと、前記デバイスドライバ又は機能拡張モジュールが、前記メモリ領域の前記識別子で特定されるパラメータに前記設定値を設定するステップとを実行することにより、前記機能拡張モジュールが管理するパラメータを操作する。また、前記渡すステップでは、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値を、前記所定のインタフェースの引数として、前記アプリケーションから前記デバイスドライバ又は機能拡張モジュールに渡す。また、前記設定値を設定するステップでは、前記識別子がデバイスドライバに管理されるパラメータの識別子か否かを判別し、前記識別子がデバイスドライバで管理しているパラメータの識別子ではないと判別された場合に、前記機能拡張モジュールで管理しているパラメータに対して前記設定値の設定を行う。また、前記識別子が、前記デバイスドライバで管理しているパラメータの識別子でなく、且つ前記機能拡張モジュールで管理しているパラメータの識別子でない場合に、アプリケーションに対して設定不可の通知をするステップを更に有する。また、前記周辺装置制御処理が終了した場合に、前記確保されたメモリ領域を解放するステップを更に有する。
【００１２】
又、周辺装置制御用のコマンドを生成するデバイスドライバと、該デバイスドライバから設定できないパラメータを有する機能拡張モジュールとを使った周辺装置制御処理を含むアプリケーションプログラムであって、前記デバイスドライバ及び機能拡張モジュールに、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを含む容量を問い合わせて、該容量のメモリ領域を確保するステップと、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバ又は機能拡張モジュールに渡すステップと、前記識別子で特定されるパラメータに前記設定値を設定した前記メモリ領域のパラメータを使用して、前記デバイスドライバ及び機能拡張モジュールに周辺装置制御処理を行うよう指示するステップとを有することを特徴とする。ここで、前記周辺装置制御処理が終了した場合に、前記確保されたメモリ領域を解放するステップを更に有する。

【００１３】
又、アプリケーションの周辺装置制御処理に応答して周辺装置制御用のコマンドを生成するデバイスドライバプログラムであって、前記アプリケーションからの問い合わせに応答して、前記デバイスドライバプログラムが管理するパラメータと、前記デバイスドライバプログラムの機能拡張用の機能拡張モジュールが管理する前記デバイスドライバプログラムから設定できないパラメータとを含む容量を返すステップと、少なくとも、前記アプリケーションにより確保された前記容量のメモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記アプリケーションから受け取るステップと、前記識別子が前記デバイスドライバプログラムに管理されるパラメータの識別子か否かを判別するステップと、前記識別子が前記デバイスドライバプログラムで管理しているパラメータの識別子であると判別された場合に、前記識別子で特定されるパラメータに前記設定値を設定し、前記識別子が前記デバイスドライバプログラムで管理しているパラメータの識別子ではないと判別された場合に、少なくとも、前記アプリケーションにより確保された前記容量のメモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記機能拡張モジュールに渡すステップと、前記設定値を設定した前記メモリ領域のパラメータを使用して、周辺装置制御処理を行うステップとを有することを特徴とする。
【００１４】
又、アプリケーションの周辺装置制御処理に応答して周辺装置制御用のコマンドを生成するデバイスドライバの機能拡張用の機能拡張モジュールプログラムであって、前記デバイスドライバからの問い合わせに応答して、前記機能拡張モジュールプログラムが管理する前記デバイスドライバから設定できないパラメータの容量を返すステップと、少なくとも、前記アプリケーションにより確保された前記容量のメモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバから受け取るステップと、前記識別子が前記機能拡張モジュールプログラムに管理されるパラメータの識別子か否かを判別するステップと、前記識別子が前記機能拡張モジュールプログラムで管理しているパラメータの識別子であると判別された場合に、前記識別子で特定されるパラメータに前記設定値を設定し、前記識別子が前記機能拡張モジュールプログラムで管理しているパラメータの識別子ではないと判別された場合に、設定不可の通知を前記デバイスドライバに返すステップと、前記設定値を設定した前記メモリ領域のパラメータを使用して、周辺装置制御の機能拡張処理を行うステップとを有することを特徴とする。
【００１５】
上記各プログラムをコンピュータ読み出し可能に記憶する記憶媒体をも提供する。
【００１６】
又、周辺装置制御用のコマンドを生成するデバイスドライバと、該デバイスドライバから設定できないパラメータを有する機能拡張モジュールとを使って、アプリケーションから周辺装置制御処理を実行する周辺装置制御装置であって、前記アプリケーションが、所定のインタフェースを用いて、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを操作する手段と、前記操作する手段の操作に応じて、前記デバイスドライバ又は機能拡張モジュールがパラメータを設定する手段と、前記パラメータを使用して、前記デバイスドライバ及び機能拡張モジュールにより周辺装置制御処理を行う手段とを有することを特徴とする。
【００１７】
ここで、前記所定のインタフェースとは、デバイスドライバ又はオペレーティングシステムが公開しているインタフェースである。また、前記アプリケーションが、前記デバイスドライバが管理するパラメータと前記機能拡張モジュールが管理するパラメータとを含む容量のメモリ領域を確保する手段をさらに備え、前記操作する手段は、前記アプリケーションから、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値とを、前記デバイスドライバ又は機能拡張モジュールに渡す手段と、前記デバイスドライバ又は機能拡張モジュールが、前記メモリ領域の前記識別子で特定さ

れるパラメータに前記設定値を設定する手段とを有し、前記機能拡張モジュールが管理するパラメータを操作する。また、前記渡す手段は、少なくとも、前記メモリ領域の位置と、パラメータを特定する識別子と、該パラメータの設定値を、前記所定のインタフェースの引数として、前記アプリケーションから前記デバイスドライバ又は機能拡張モジュールに渡す。また、前記設定値を設定する手段は、前記識別子がデバイスドライバに管理されるパラメータの識別子か否かを判別し、前記識別子がデバイスドライバで管理しているパラメータの識別子ではないと判別された場合に、前記機能拡張モジュールで管理しているパラメータに対して前記設定値の設定を行う。また、前記識別子が、前記デバイスドライバで管理しているパラメータの識別子でなく、且つ前記機能拡張モジュールで管理しているパラメータの識別子でない場合に、アプリケーションに対して設定不可の通知をする手段を更に有する。また、前記周辺装置制御処理が終了した場合に、前記確保されたメモリ領域を解放する手段を更に有する。

【発明の効果】

【００１８】

以上説明したように、本発明によれば、デバイスドライバに新しい機能をプラグイン形式で追加した場合にも、新しい機能の設定値をアプリケーションからデバイスドライバのユーザインタフェースを開くことなく変更できるという効果がある。

【発明を実施するための最良の形態】

【００１９】

以下、本発明を適用するのに好適である実施形態について説明を行う。尚、背景技術の説明でも指摘したが、本発明の適用例として、以下の実施形態ではアプリケーションからプリンタドライバを介して印刷処理を行う例を説明するが、この技術はアプリケーションから周辺装置制御用のデバイスドライバを介して周辺装置の制御処理を行う場合に共通に適用可能であり、この技術も本発明に含まれる。すなわち、実施形態中のプリンタは周辺装置に、プリンタドライバはデバイスドライバに、グラフィックスドライバは他の周辺装置の機能実現用ドライバに読み替えられる。例えば、周辺装置がディスプレイであれば、デバイスドライバはディスプレイドライバであり、周辺装置がスキャナであれば、デバイスドライバはスキャナドライバであり、例えばプラグインモジュールは両面スキャンや領域分割などの拡張機能、などの対応が出来る。

【００２０】

＜本実施形態の印刷処理システムの構成例＞

図１は、本実施形態の印刷処理システムの構成を説明するブロック図である。なお、特に断らない限り、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、ＬＡＮ,ＷＡＮ等のネットワークを介して接続が為され処理が行われるシステムであっても、本発明を適用できることは言うまでもない。

【００２１】

（ホストコンピュータの構成例）

図１において、３００はホストコンピュータであり、ホストコンピュータ３００はコンピュータ制御部本体２００を含む。

【００２２】

ホストコンピュータ３００は、ＲＯＭ３のプログラム用ＲＯＭあるいは外部メモリ１１に記憶された文書処理プログラム等に基づいて、図形、イメージ、文字、表（表計算等を含む）等が混在した文書処理を実行するＣＰＵ１を備え、システムバス４に接続される各デバイスをＣＰＵ１が総括的に制御する。また、このＲＯＭ３のプログラム用ＲＯＭあるいは外部メモリ１１には、ＣＰＵ１の制御プログラムであるオペレーティングシステムプログラム（以下ＯＳ）等を記憶し、ＲＯＭ３のフォント用ＲＯＭあるいは外部メモリ１１には、上記文書処理の際に使用するフォントデータ等を記憶し、ＲＯＭ３のデータ用ＲＯＭあるいは外部メモリ１１には、上記文書処理等を行う際に使用する各種データを記憶する。２はＲＡＭで、ＣＰＵ１の主メモリ、ワークエリア等として機能する。

【００２３】

５はキーボードコントローラ（ＫＢＣ）で、キーボード９や不図示のポインティングデバイスからのキー入力を制御する。６はＣＲＴコントローラ（ＣＲＴＣ）で、ＣＲＴディスプレイ（ＣＲＴ）１０の表示を制御する。７はディスクコントローラ（ＤＫＣ）で、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、デバイスドライバの一例であるプリンタ制御コマンド生成プログラム（プリンタドライバ）等を記憶するハードディスク（ＨＤ）、フロッピー（登録商標）ディスク（ＦＤ）等の外部メモリ１１とのアクセスを制御する。８はプリンタコントローラ（ＰＲＴＣ）で、所定の双方向性インタフェース２１を介してデジタルカメラ、プリンタ、ファクシミリ、及びこれらの複合機などの各種画像形成装置を含む周辺装置の一例であるプリンタ１５０に接続されて、プリンタ１５０との通信制御処理を実行する。

【００２４】

なお、ＣＰＵ１は、例えばＲＡＭ２上に設定された表示情報ＲＡＭへのアウトラインフォントの展開（ラスタライズ）処理を実行し、ＣＲＴ１０上でのＷＹＳＩＷＹＧを可能としている。また、ＣＰＵ１は、ＣＲＴ１０上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウインドウを開き、種々のデータ処理を実行する。ユーザは印刷を実行する際、印刷の設定に関するウインドウを開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバに対する印刷処理方法の設定を行える。

【００２５】

（プリンタの構成例）

図１において、１５０はプリンタであり、プリンタ制御部本体１００を含む。

【００２６】

プリンタ１５０において、１２はプリンタＣＰＵで、ＲＯＭ１３のプログラム用ＲＯＭに記憶された制御プログラム等あるいは外部メモリ１４に記憶された制御プログラム等に基づいて、システムバス１５に接続される印刷部（プリンタエンジン）１７に出力情報としての画像信号を出力する。また、このＲＯＭ１３のプログラムＲＯＭには、ＣＰＵ１２の制御プログラム等を記憶する。ＲＯＭ１３のフォント用ＲＯＭには、上記出力情報を生成する際に使用するフォントデータ等を記憶し、ＲＯＭ１３のデータ用ＲＯＭには、ハードディスク等の外部メモリ１４がないプリンタの場合には、ホストコンピュータ上で利用される情報等を記憶している。

【００２７】

ＣＰＵ１２は、入力部１８を介してホストコンピュータ３００との通信処理が可能となっており、プリンタ１５０内の情報等をホストコンピュータ３００に通知可能に構成されている。１９はＣＰＵ１２の主メモリ、ワークエリア等として機能するＲＡＭで、図示しない増設ポートに接続されるオプションＲＡＭによりメモリ容量を拡張することができるように構成されている。なお、ＲＡＭ１９は、出力情報展開領域、環境データ格納領域、ＮＶＲＡＭ等に用いられる。前述したハードディスク（ＨＤ）、ＩＣカード等の外部メモリ１４は、メモリコントローラ（ＭＣ）２０によりアクセスを制御される。外部メモリ１４は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、１８は前述した操作パネルで操作のためのスイッチおよびＬＥＤ表示器等が配されている。また、前述した外部メモリは１個に限らず、少なくとも１個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていてもよい。さらに、図示しないＮＶＲＡＭを有し、操作パネル２２からのプリンタモード設定情報を記憶するようにしてもよい。

【００２８】

（本実施形態のプリンタドライバの拡張システム例）

本実施形態におけるプリンタドライバの拡張システムは、図３に示すようにプリンタドライバにプラグインモジュールを追加する構成になっている。プラグインモジュールは、図３に示すとおり、UI（ユーザインタフェース）ドライバ３０３の拡張機能を有するUIプラグイン３０４と、グラフィックスドライバ３０５の拡張機能を有するグラフィックスプ

ラグイン３０６からなる。図３を用いてそれぞれのプラグインモジュールについて説明する。

【００２９】

UIプラグイン３０４は、ドライバのUIに新たなシートを加える場合や、UIドライバ３０３で処理している、インストール時の処理などのUIドライバ３０３に実装されている様々なイベント処理をカスタマイズする場合に用いる。UIドライバ３０３は、システムに対してエクスポートしているDDI（Device Driver Interface）がコールされた際に、UIプラグイン３０４が公開しているインタフェースを取得する。UIドライバ３０３が取得したUIプラグイン３０４のインタフェースを用いて、UIプラグイン３０４と通信することで、UIドライバ３０３に実装されたカスタマイズ処理が実行される。

【００３０】

一方、グラフィックスプラグイン３０６は、GDI（Graphics Device Interface）とグラフィックスドライバ３０５とのインタフェースであるDDIの処理を横取り、又は特定のタイミングで処理を追加する場合や、印刷データのスプール処理を横取りする場合に用いる。グラフィックスドライバ３０５がエクスポートしているDDIは、ジョブを生成するためにGDIから適宜コールされるが、その初期化のタイミングで、グラフィックスドライバ３０５は、グラフィックスプラグイン３０６が公開しているインタフェースを取得し、RAM２に保持する。ここで取得したインタフェースを用いて、DDIがコールされるタイミングにそのDDIに対応したグラフィックスプラグイン３０６の処理を呼び出す。呼び出されたグラフィックスプラグイン３０６は、プラグイン内に実装された処理を実行し、処理を再びグラフィックスドライバ３０５に返す。ここで、グラフィックスプラグイン３０５が処理を横取りしている場合、DDIはそこで処理を終了し、追加処理を実装している場合には、そのまま後続する処理を続ける。グラフィックスプラグイン３０６がグラフィックスドライバ３０５の印刷データのスプール処理を横取りする場合は、グラフィックスドライバ３０５内の印刷データをスプールする直前にグラフィックスプラグイン３０６のインタフェースをコールする。コールされたグラフィックスプラグイン３０６は、スプールされるデータを取得し、特定の処理を実行した後、再度グラフィックスドライバ３０５のスプール処理関数をコールすることで、グラフィックスドライバ３０５に特定の処理を施したデータを渡す。データを受け取ったグラフィックスドライバ３０５は、実際にシステムスプーラ３０７にスプールする処理を実行する。

【００３１】

このようにプラグインモジュールに拡張機能を実装することで、プリンタドライバの拡張システムを実現している。

【００３２】

（本実施形態の印刷処理時のメモリマップ）

図２は、本実施形態の印刷処理時のRAM２のメモリマップである。このメモリマップは、本実施形態でのアプリケーションによる設定処理が実行される時点の状態を示している。

【００３３】

図２の左図で、２０１は印刷処理を含むアプリケーションの領域、２０２は空き領域、２０３は印刷処理に使用される印刷処理関連データの領域、２０４はプリンタドライバとプラグインモジュールを含む印刷処理関連プログラム領域、２０５はＯＳ領域、２０６はＢＩＯＳの領域である。なお、図２では各領域をまとまった領域として示しているが、ＯＳやＢＩＯＳ等の常駐プログラム以外は通常RAM２の領域確保／解放制御により任意のアドレスに分散している。

【００３４】

図２の右図は、印刷処理関連データ２０３及び印刷処理関連プログラム２０４の一構成例を示している。プリンタドライバのプログラム２０４ａと関連データ２０３ｂ、プラグインモジュールのプログラム２０４ｂ，２０４ｃと関連データ２０３ｂ，２０３ｃ、プラグインモジュールのインタフェース２０４ｄ、パブリックパラメータ２０３ｄが、本実施

形態のアプリケーションによる設定処理前に準備されている。

【００３５】

＜本実施形態の印刷処理システムの動作手順例＞

上記構成における、本実施形態の印刷処理システムの動作手順例を示す。

【００３６】

図５は、アプリケーションの印刷処理の手順の一例を示すフローチャートである。

【００３７】

（印刷処理の初期化例：ステップS51）

アプリケーションで印刷処理を指示すると、ステップS51で印刷処理の初期化を要求する。この要求はOS２０５の管理下でプリンタドライバにより実行される。

【００３８】

本実施形態に示す印刷処理関連プログラム（プリンタドライバを含む）２０４による印刷処理は、OS２０５管理の下で動作するアプリケーション２０１よりプリンタドライバの備える初期化処理が呼び出されると、OS２０５の管理の下、RAM２に印刷処理関連プログラム２０４がロードされる。印刷処理関連プログラム２０４がRAM２にロードされると、プリンタドライバの管理部に備えられたプリンタドライバの初期化処理部が呼び出され、初期化処理が行われる。

【００３９】

図４は、プリンタドライバ２０４に追加機能のプラグインモジュール４０３がインストールされている状態の各モジュール間の構成を示す図である。

【００４０】

４０２はプリンタドライバであり、内部構造体４０１は、プリンタドライバ４０２で設定可能な設定項目に対応する変数であるメンバなどを格納したものである。プラグインモジュール４０３はプリンタドライバ４０２に追加された機能が実装されたモジュールである。図４のようにプラグインモジュール４０３がインストールされている場合には、プリンタドライバ４０２は初期化処理の際に、外部メモリ１に格納されているプラグインモジュール４０３をRAM２にロードし、プリンタドライバ４０２はプラグインモジュール４０３のインタフェースを取得する。プリンタドライバ４０２は、ここで取得したインタフェースを用いて、プラグインモジュール４０３と通信することが可能となる。

【００４１】

前記図２のメモリマップは、この時点の状態である。

【００４２】

（パラメータの設定処理例：ステップS52-S55）

アプリケーションは、ステップS52-S55で、印刷処理に係る所望のパラメータの設定／変更を行う。この場合、メモリの空き領域２０２に該アプリケーション独自の内部構造体を構築して、この独自の内部構造体上で所望のパラメータの設定／変更を行う。プリンタドライバ及びプラグインモジュールは、このアプリケーション独自の内部構造体のパラメータを使って、アプリケーション独自の印刷処理を実現する。

【００４３】

まず、ステップS52で、プリンタドライバ及びプラグインモジュールのパラメータ量をそれぞれから得て、内部構造体のサイズを取得する。次にステップS53で、取得したサイズの領域を空き領域２０２に確保する。この時に確保された内部構造体の構成は図７のようになっており、この構造は以下で詳細に説明する。ステップS54及びS55で、プリンタドライバの設定変更処理関数の引数として、内部構造体の各パラメータの名あるいはＩＤ、設定情報、アドレスをアプリケーションからプリンタドライバ又はプラグインモジュールに渡すことによって、パラメータの設定／変更を、設定が完了するまで繰り返す。

【００４４】

（パラメータの設定処理の詳細手順例）

図６は、プリンタドライバ４０２からエクスポートされている図８に示す設定変更処理関数を用いて、図４に示すアプリケーション４０４からプリンタドライバ４０２の設定値

、すなわち図４の内部構造体４０１を設定／変更する際のフローチャートを示したものである。なお、図６では、図５のステップS54及びS55における繰り返しについては、煩雑さをさけるため省いている。

【００４５】

まず、アプリケーション４０４はプリンタドライバの内部構造体４０1を格納するためのメモリ領域のサイズをプリンタドライバ４０２に問い合わせる（ステップS501）。プリンタドライバ４０２は、プラグインモジュール４０３に対して、プラグインモジュールのインタフェースを介して、プラグインモジュール４０３に必要な内部構造体のデータ領域のサイズを問い合わせる（ステップS502）。プラグインモジュール４０３は、必要な内部構造体４０１の領域サイズを計算し（ステップS503）、計算された領域のサイズをプリンタドライバ４０２に返す（ステップS504）。プリンタドライバ４０２はプラグインモジュール４０３に必要な内部構造体４０１のサイズを取得した（ステップS505）後、プリンタドライバ４０２に必要な内部構造体４０１のサイズに前記ステップS505で取得したプラグインモジュール４０３のサイズを加えたデータサイズを、アプリケーション４０４に返す（ステップS506）。

【００４６】

アプリケーション４０４は、プリンタドライバ４０２から返された値を受け取り（ステップS507）、取得したサイズ分のメモリ領域を空き領域２０２上に確保する（ステップS508）。

【００４７】

この時の内部構造体は、図７のような構成になっている。

【００４８】

図７の領域６０１は、アプリケーション４０４からも直接参照可能なPublic領域で、領域６０２は、プリンタドライバ４０２から参照可能なDriver Private 領域で、領域６０３は、プラグインモジュール４０３から参照可能なPlugin Private 領域となっている。プリンタドライバ４０２はPlugin Private 領域６０３の内部構造の情報を持たないため、Plugin Private 領域６０３の内容を直接参照又は変更を行うことができない。図７の各領域は、Public領域６０１が図２のパブリックパラメータ２０３ｄに、Driver Private 領域６０２がプリンタドライバの定義済みのメンバ２０３ａに、Plugin Private 領域６０３がプラグインの独自データ２０３ｂ，２０３ｃに対応するが、図７の内部構造体は各アプリケーション独自に空き領域２０２に確保されたものであり、各アプリケーション独自の設定が可能である。各アプリケーションの印刷処理は空き領域２０２に確保されたこの内部構造体に基づいて実行される。

【００４９】

前記ステップS508で確保したメモリ領域のアドレスと設定するパラメータＩＤ、設定内容を、図８に一例を示す設定変更処理関数に渡して、設定／変更処理を実行する（ステップS509）。

【００５０】

この図８の設定変更処理関数で用いられている引数の中で、特に設定変更処理に関わるパラメータは、pDevModeInOutputとpItemParam，dwItemIDである。それぞれのパラメータには以下のような情報がセットされる。

dwItemID：変更を行いたいのはプリンタドライバ内部構造体のどのメンバなのかを指定する（パラメータＩＤ）。

pItemParam：dwItemIDで示したメンバのパラメータを指定する（設定内容）。

pDevModeInOutput：プリンタドライバ内部構造体へのポインタを指定する（内部構造体の先頭アドレス、又はDriver Private 領域６０３の先頭アドレス）。

【００５１】

例えば、アプリケーションからの設定変更要求がPrint Styleを1-Sided PrintingからBooklet Printingに変更するものであった場合を例にとると、上記の各々のパラメータには、

dwItemID＝DM_ITEM_LAYOUT
pItemParam ＝DM_PARAM_BOOKLET
pDevModeInOutput＝ｐDevMode
がセットされた上で、DocumentPropertiesEx( )が呼び出される。

【００５２】

プリンタドライバ４０２内でエクスポートしている設定変更処理関数が呼び出されると（ステップS510）、プリンタドライバ４０２は、設定変更処理関数によって渡されてくる、内部構造体のどのメンバの変更をするのかを表す "deItemID" の値を評価し、その値がプリンタドライバ４０２で定義されているIDであるかどうかを判定する（ステップS511）。"dwItemID" がプリンタドライバ４０２内で定義されていると判断されると、プリンタドライバがアプリケーションから渡された内部構造体へのポインタからパラメータのアドレスを特定して設定変更処理を実行し（ステップS514）、アプリケーション４０１に対して、処理結果を通知する（ステップS515）。なお、パラメータのアドレスは、内部構造体６００の先頭アドレスが与えられた場合は、Public領域６０１の容量を加えてDriver Private 領域６０２の先頭アドレスを算出する。

【００５３】

前記ステップS511で、"dwItemID" プリンタドライバ４０２内で定義されていないと判断されるとプラグインモジュール４０３からエクスポートされているインタフェースをコールする。この際、設定変更処理関数から渡されてきたパラメータを渡す。この時に、例えば、プリンタドライバ４０２で定義されているメンバを保持するためのDriver Private 領域６０２の容量を渡せば、プラグインモジュール４０３はPlugin Private 領域６０３の先頭アドレスを知ることができる。

【００５４】

コールされたプラグインモジュール４０３内では、"dwItemID" の値を評価して、その値がプラグインモジュール４０３内で定義されているIDであるかどうかを判定する（ステップS512）。そこでプラグインモジュール４０３内で定義されている独自データであると判断されれば、その独自データのアドレスを特定して変更処理を実行し、処理結果をプリンタドライバ４０２に返す（ステップS513）。このプラグインモジュール４０３の独自データへの変更は、図６のPlugin Private 領域６０３内のデータに適用されることになる。

【００５５】

一方、前記ステップS512において、プラグインモジュール４０３内で定義されていないと判断された場合には、サポートしていない旨の通知をプリンタドライバ４０２に返す。

【００５６】

以上のステップS509からS515の一連の処理は、アプリケーションによってプリンタドライバの管理部に備えられた終了処理が呼び出されるまで、繰り返し実行される。

【００５７】

以上の処理によって、アプリケーションからプラグインモジュールの設定値の変更をプリンタドライバのUIを開くことなく実行することができる。

【００５８】

（プリント処理S56及びS57）

アプリケーションによるパラメータの設定／変更処理が終了すると、ステップS56及びS57でプリント処理が開始される。このプリント処理で、プリンタドライバ及びプラグインモジュールがそれぞれ使用するパラメータは、空き領域２０２に確保されたアプリケーション独自の内部構造体であり、アプリケーション独自のプリント処理が実現される。

【００５９】

（終了処理例：S58）

アプリケーション４０１よりプリント終了のためにプリンタドライバ４０２の終了処理が呼び出されると、空メモリ２０２に確保したメモリ領域を解放する（ステップS58)などの終了処理が実行される（図９のステップS801）。これにより処理は全て終了し、本実施

形態におけるアプリケーションからの印刷処理関連プログラムによる印刷処理も終了し、RAM２からはＯＳ４０５の機能により印刷処理関連データ及びプログラムが消去される。

【００６０】

なお、本実施形態においては、印刷処理関連プログラムを記憶する媒体を外部メモリとしているが、外部メモリとしては、FD、HDドライブ、CD-ROMやICメモリカード等であってもよい。更に、本印刷プログラム単独、もしくはOSその他のホストコンピュータ上で動作するプログラムと共にROM３に記録しておき、これをメモリマップの一部となすように構成し、直接CPU１で実行することも可能である。

【００６１】

本実施形態の始めにも指摘したが、本発明の適用例として、上記実施形態ではアプリケーションからプリンタドライバを介して印刷処理を行う例を説明したが、この技術はアプリケーションから周辺装置制御用のデバイスドライバを介して周辺装置の制御処理を行う場合に主要な変更なしに共通に適用可能であり、本実施形態と同等の効果を奏するものである。従って、これらの範囲の技術も本発明に含まれるものである。すなわち、上記実施形態中のプリンタは周辺装置に、プリンタドライバはデバイスドライバに、グラフィックスドライバは他の周辺装置の機能実現用ドライバに読み替えられる。例えば、周辺装置がディスプレイであれば、デバイスドライバはディスプレイドライバであり、周辺装置がスキャナであれば、デバイスドライバはスキャナドライバである、などの対応が出来ることは明らかである。

【００６２】

更に、一般に、ドライバプログラムとは、アプリケーションの指示を、コンピュータ本体の外部にある装置（他のコンピュータでもよい）を制御するためのコマンドに変換するプログラムであるが、本発明は、ドライバとの名称に関りなく、また制御の対象が所謂入出力装置であるとの制限もなく、アプリケーションが走っているコンピュータから外部装置を制御するプログラムに適用が可能である。従って、本明細書の周辺装置あるいはデバイスとの文言も、アプリケーションが走っているコンピュータから見た外部装置全般を含む概念であり、これも本発明に含まれる。

【図面の簡単な説明】

【００６３】

【図１】本実施形態の印刷処理システムの構成例を説明するブロック図である。

【図２】本実施形態の印刷関連処理関連プログラムがホストコンピュータ３００のＲＡＭ２上にロードされ、実行可能となった状態のメモリマップを示す図である。

【図３】本実施形態のプリンタドライバ拡張システムの例を示す図である。

【図４】本実施形態の印刷処理システムの各モジュール間の関連例を示す図である。

【図５】本実施形態のアプリケーションの印刷処理の手順例を示すフローチャートである。

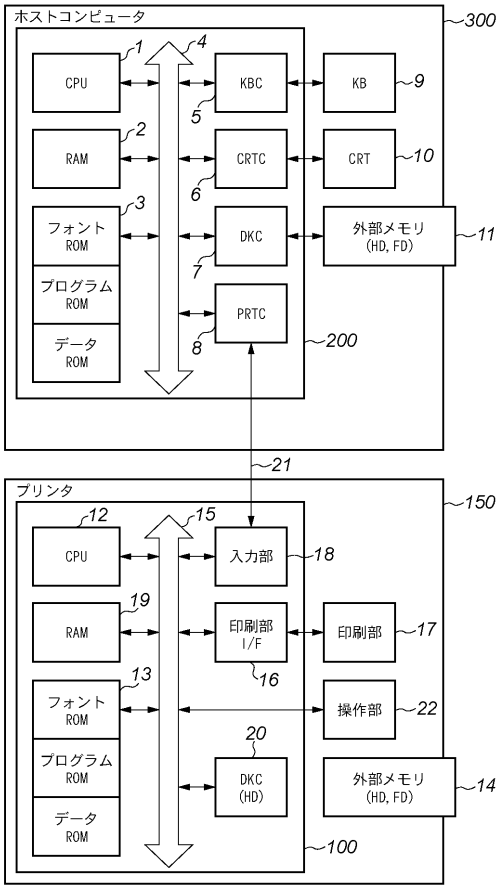
【図６】本実施形態の設定変更処理関数からの設定変更処理の手順例を示すフローチャートである。

【図７】本実施形態の内部構造体の構成例を示す図である。

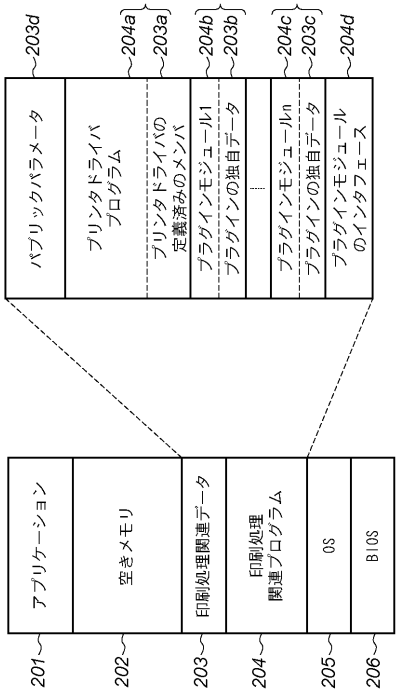
【図８】本実施形態の設定変更関数の一例を示した図である。

【図９】本実施形態の終了処理例を示すフローチャートである。

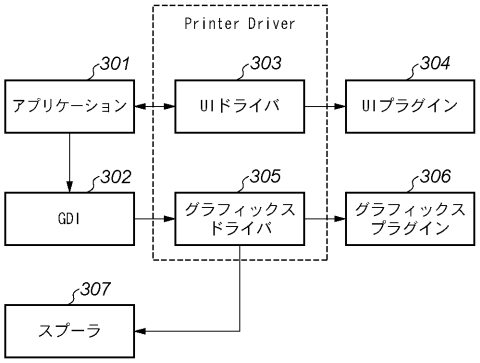
【図１】


ホストコンピュータ
300
CPU
RAM
フォント ROM
プログラム ROM
データ ROM
KBC
CRTC
DKC
PRTC
KB
CRT
外部メモリ (HD, FD)
200
21
プリンタ
150
CPU
RAM
フォント ROM
プログラム ROM
データ ROM
入力部
印刷部 I/F
印刷部
操作部
DKC (HD)
外部メモリ (HD, FD)
100

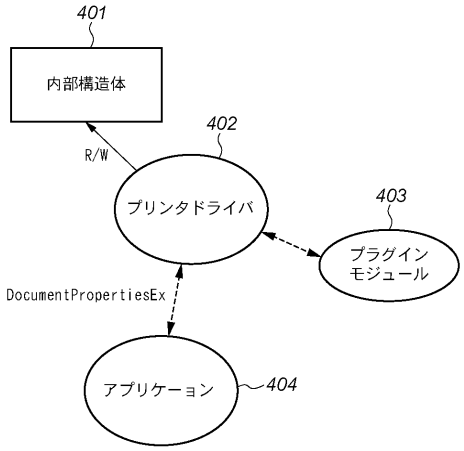
【図２】


パブリックパラメータ
プリンタドライバ プログラム
プリンタドライバの 定義済みのメンバ
プラグインモジュール1
プラグインの独自データ
プラグインモジュールn
プラグインの独自データ
プラグインモジュール のインタフェース
アプリケーション
空きメモリ
印刷処理関連データ
印刷処理 関連プログラム
OS
BIOS

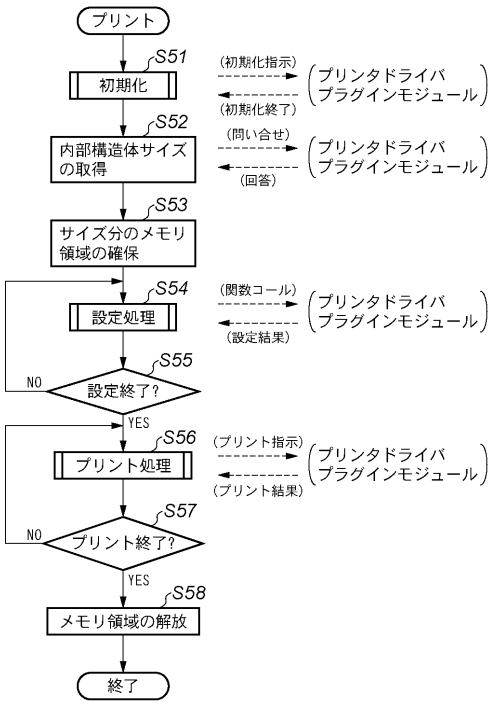
【図３】


Printer Driver
アプリケーション
UIドライバ
UIプラグイン
GDI
グラフィックス ドライバ
グラフィックス プラグイン
スプーラ

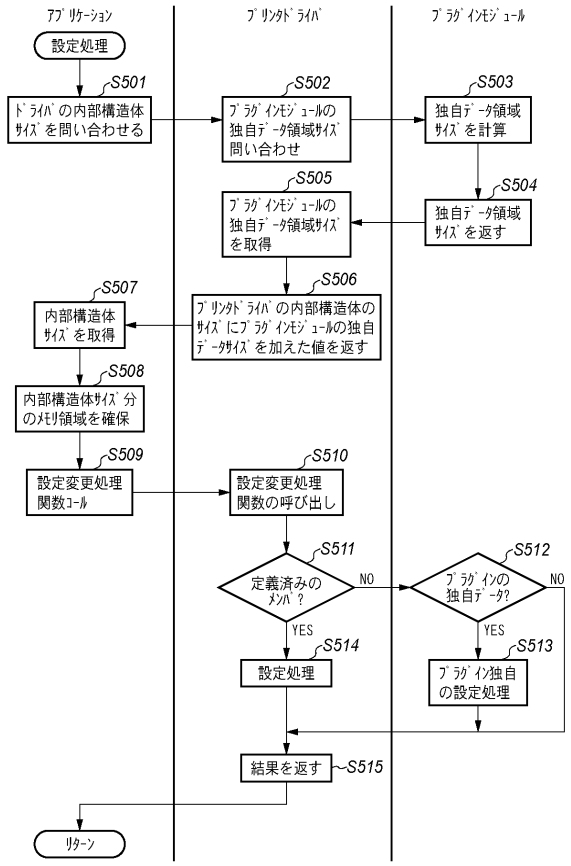
【図４】


内部構造体
R/W
プリンタドライバ
プラグイン モジュール
DocumentPropertiesEx
アプリケーション

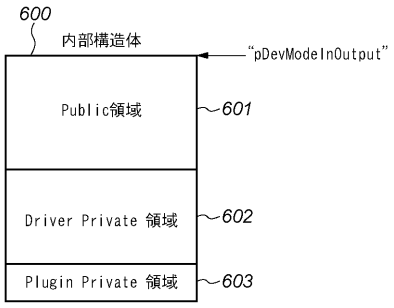
【図５】


プリント
初期化
(初期化指示)
(初期化終了)
プリンタドライバ プラグインモジュール
内部構造体サイズの取得
(問い合せ)
(回答)
サイズ分のメモリ領域の確保
設定処理
(関数コール)
(設定結果)
設定終了?
NO
YES
プリント処理
(プリント指示)
(プリント結果)
プリント終了?
メモリ領域の解放
終了

【図６】



【図７】

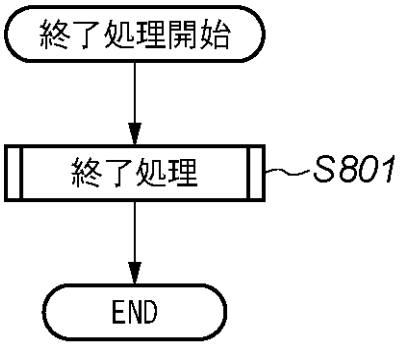


【図８】

設定変更処理関数

```
LONG DocumentPropertiesEx(
        HWND hWnd,                  // handle to parent window
        HANDLE hPrinter,            // handle to printer object
        LPTSTR pDeviceName,         // device name
        PDEVMODE pDevMdeInOutput,   // original device mode &
                                    // modified device mode
        DWORD dwItemID,             // indicate ItemID
        LPVOID pItemParam           // Parameter indicated by ItemID
);
```

【図９】



フロントページの続き

Ｆターム(参考) 5B021 AA01 CC06
5B076 AA03 AB20