(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5871521号
(P5871521)
(45)発行日　平成28年3月1日(2016.3.1)　(24)登録日　平成28年1月22日(2016.1.22)

(51)Int.Cl.　F I
G06F 3/12 (2006.01)　G06F 3/12 377
G06F 3/12 324
G06F 3/12 344
G06F 3/12 303

請求項の数 12　(全 18 頁)

(21)出願番号　特願2011-185044(P2011-185044)
(22)出願日　平成23年8月26日(2011.8.26)
(65)公開番号　特開2013-45422(P2013-45422A)
(43)公開日　平成25年3月4日(2013.3.4)
審査請求日　平成26年8月8日(2014.8.8)

(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　100090273
弁理士　國分　孝悦
(72)発明者　川嵜　敬二
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　田川　泰宏

最終頁に続く

(54)【発明の名称】印刷データ処理方法、印刷データ処理装置及びプログラム

(57)【特許請求の範囲】
【請求項１】
入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第一のフィルターで印刷データを入力し記憶領域に格納する記憶工程と、
前記第一のフィルターの後に位置し、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第二のフィルターで、前記印刷データを構成するパーツを順次、入力し、解析する解析工程と、
前記解析工程における前記解析の結果、前記印刷データより取得された印刷設定情報に従って、前記記憶工程で記憶領域に記憶された印刷データを処理する処理工程と、
前記第二のフィルターで、前記印刷設定情報を取得したあと、以降のパーツの全部又は一部を入力せず処理を終了する終了工程と、
を含む印刷データ処理方法。
【請求項２】
入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第一のフィルターで、印刷データを構成するパーツを順次、入力し、解析する解析工程と、
前記解析工程における前記解析の結果、前記印刷データより取得された印刷設定情報を記憶領域に記憶する記憶工程と、
前記第一のフィルターの後に位置し、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第二のフィルターで、前記印刷データを入力し、前記記憶工程で前記記憶領域に記憶された印刷設定情報に従って処理する処理工程と、

を含む印刷データ処理方法。
【請求項３】
入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第一のフィルターで印刷データを入力し記憶領域に格納する記憶手段と、
前記第一のフィルターの後に位置し、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第二のフィルターで、前記印刷データを構成するパーツを順次、入力し、解析する解析手段と、
前記解析手段における前記解析の結果、前記印刷データより取得された印刷設定情報に従って、前記記憶手段で記憶領域に記憶された印刷データを処理する処理手段と、
前記第二のフィルターで、前記印刷設定情報を取得したあと、以降のパーツの全部又は一部を入力せず処理を終了する終了手段と、
を有する印刷データ処理装置。
【請求項４】
前記処理手段では、前記解析手段における前記解析の結果、前記印刷データより取得された印刷設定情報の保存場所情報に従って、前記記憶手段で記憶領域に記憶された印刷データを保存場所に移動させる請求項３記載の印刷データ処理装置。
【請求項５】
前記処理手段では、前記解析手段における前記解析の結果、前記印刷データより取得された印刷設定情報の印刷プレビュー表示情報に従って、前記記憶手段で記憶領域に記憶された印刷データを印刷プレビュー表示する表示モジュールを起動する請求項３記載の印刷データ処理装置。
【請求項６】
前記処理手段では、前記印刷プレビュー表示情報が印刷プレビューを表示することを示している場合、前記記憶手段で記憶領域に記憶された印刷データを印刷プレビュー表示する表示モジュールを起動させ、
前記終了手段では、前記第二のフィルターで、前記印刷設定情報を取得したあと、前記印刷プレビュー表示情報が印刷プレビューを表示することを示している場合、以降のパーツの全部又は一部を入力せず処理を終了する請求項５記載の印刷データ処理装置。
【請求項７】
入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第一のフィルターで、印刷データを構成するパーツを順次、入力し、解析する解析手段と、
前記解析手段における前記解析の結果、前記印刷データより取得された印刷設定情報を記憶領域に記憶する記憶手段と、
前記第一のフィルターの後に位置し、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第二のフィルターで、前記印刷データを入力し、前記記憶手段で前記記憶領域に記憶された印刷設定情報に従って処理する処理手段と、
を有し、
前記記憶手段では、前記取得された印刷設定情報のうち印刷データの保存場所情報を前記記憶領域に記憶し、
前記処理手段では、前記第二のフィルターで、前記印刷データを入力し、前記記憶領域に記憶された保存場所情報に従って、前記印刷データを保存場所に保存する印刷データ処理装置。
【請求項８】
コンピュータに、
入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第一のフィルターで印刷データを入力し記憶領域に格納する記憶ステップと、
前記第一のフィルターの後に位置し、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第二のフィルターで、前記印刷データを構成するパーツを順次、入力し、解析する解析ステップと、
前記解析ステップにおける前記解析の結果、前記印刷データより取得された印刷設定情

報に従って、前記記憶ステップで記憶領域に記憶された印刷データを処理する処理ステップと、
前記第二のフィルターで、前記印刷設定情報を取得したあと、以降のパーツの全部又は一部を入力せず処理を終了する終了ステップと、
を実行させるためのプログラム。
【請求項９】
前記処理ステップでは、前記解析ステップにおける前記解析の結果、前記印刷データより取得された印刷設定情報の保存場所情報に従って、前記記憶ステップで記憶領域に記憶された印刷データを保存場所に移動させる請求項８記載のプログラム。
【請求項１０】
前記処理ステップでは、前記解析ステップにおける前記解析の結果、前記印刷データより取得された印刷設定情報の印刷プレビュー表示情報に従って、前記記憶ステップで記憶領域に記憶された印刷データを印刷プレビュー表示する表示モジュールを起動する請求項８記載のプログラム。
【請求項１１】
前記処理ステップでは、前記印刷プレビュー表示情報が印刷プレビューを表示することを示している場合、前記記憶ステップで記憶領域に記憶された印刷データを印刷プレビュー表示する表示モジュールを起動させ、
前記終了ステップでは、前記第二のフィルターで、前記印刷設定情報を取得したあと、前記印刷プレビュー表示情報が印刷プレビューを表示することを示している場合、以降のパーツの全部又は一部を入力せず処理を終了する請求項１０記載のプログラム。
【請求項１２】
コンピュータに、
入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第一のフィルターで、印刷データを構成するパーツを順次、入力し、解析する解析ステップと、
前記解析ステップにおける前記解析の結果、前記印刷データより取得された印刷設定情報を記憶領域に記憶する記憶ステップと、
前記第一のフィルターの後に位置し、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第二のフィルターで、前記印刷データを入力し、前記記憶ステップで前記記憶領域に記憶された印刷設定情報に従って処理する処理ステップと、
を実行させ、
前記記憶ステップでは、前記取得された印刷設定情報のうち印刷データの保存場所情報を前記記憶領域に記憶し、
前記処理ステップでは、前記第二のフィルターで、前記印刷データを入力し、前記記憶領域に記憶された保存場所情報に従って、前記印刷データを保存場所に保存するプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、印刷データ処理方法、印刷データ処理装置及びプログラムに関する。
【背景技術】
【０００２】
従来、仮想的なプリンタドライバ（以後、仮想ドライバ）を利用してアプリケーションで作成した文書を取得する技術が知られる（特許文献１参照）。これによれば、まずアプリケーションから仮想ドライバに印刷することで、仮想ドライバの印刷キューに印刷データがスプールされる。印刷データ中の印刷設定情報には、印刷データの処理内容が記述された出力制御情報が含まれる。仮想ドライバは、この印刷データを、出力制御情報の内容に従って処理する。例えば、仮想ドライバは、複数の出力装置に対して配布印刷する等の処理を行う。
このように、仮想ドライバは、印刷指示を受けることで、あらゆるアプリケーションの

文書を印刷データとして受け取り、処理することができる。また、仮想ドライバに対する処理内容を、印刷データに付加する印刷設定情報に記述する構成とすることで、ユーザーによる指示やその他条件に応じて、仮想ドライバの処理内容を変更することができる。

【先行技術文献】

【特許文献】

【０００３】

【特許文献１】特開平１１－１４３６６１号公報

【発明の概要】

【発明が解決しようとする課題】

【０００４】

仮想ドライバが印刷データ中の印刷設定情報を取得するためには、印刷データの構造を解釈し、展開する必要がある。しかし、印刷データの構造によっては、印刷データの独自の展開処理は技術的に容易でなく、また印刷データフォーマットのバージョンアップに伴う構造の変化に対応できないといった問題を招く可能性がある。

【０００５】

本発明はこのような問題点に鑑みなされたもので、簡易、かつ、堅牢性の高い構成で、印刷データを、印刷データ中の印刷設定情報に従って処理することを目的とする。

【課題を解決するための手段】

【０００６】

そこで、本発明の印刷データ処理方法は、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第一のフィルターで印刷データを入力し記憶領域に格納する記憶工程と、前記第一のフィルターの後に位置し、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第二のフィルターで、前記印刷データを構成するパーツを順次、入力し、解析する解析工程と、前記解析工程における前記解析の結果、前記印刷データより取得された印刷設定情報に従って、前記記憶工程で記憶領域に記憶された印刷データを処理する処理工程と、前記第二のフィルターで、前記印刷設定情報を取得したあと、以降のパーツの全部又は一部を入力せず処理を終了する終了工程と、を含む。

【発明の効果】

【０００７】

本発明によれば、簡易、かつ、堅牢性の高い構成で、印刷データを、印刷データ中の印刷設定情報に従って処理することができる。

【図面の簡単な説明】

【０００８】

【図１】印刷装置２と印刷装置２に接続された情報処理装置（パソコン１）とによって構成される印刷システムの全体構成を概念的に表した一例を示す図である。

【図２】パソコン１のソフトウェア構成等の一例を示す図である。

【図３】実施形態１の印刷プログラム中の仮想ドライバ２０３の全体構成を概念的に表した一例を示す図である。

【図４】実施形態１のフィルター構成例とそのときの動作を示す概念図である。

【図５】チケットフィルター３０７の印刷データの保存場所情報の取得処理の一例を示すフローチャートである。

【図６】データフィルター３０８の印刷データ保存処理の一例を示すフローチャートである。

【図７】実施形態２の印刷プログラム中の仮想ドライバ２０３の全体構成を概念的に表した一例を示す図である。

【図８】実施形態２のフィルター構成例とそのときの動作を示す概念図である。

【図９】データフィルター３０８の印刷データ保存処理の一例を示すフローチャートである。

【図１０】チケットフィルター３０７の印刷データの保存場所情報の取得処理の一例を示

すフローチャートである。
【図１１】実施形態３のパソコン１上で実行される印刷プログラムにおいて、プリンタドライバ１１０７を中心に全体構成を概念的に表した一例を示す図である。
【図１２】プレビューフィルター１１０８の印刷プレビュー表示モジュール起動処理の一例を示すフローチャートである。
【図１３】レンダーフィルター１１０９の処理の一例を示すフローチャートである。
【発明を実施するための形態】
【０００９】
以下、本発明の実施形態について図面に基づいて説明する。
【００１０】
＜実施形態１＞
図１は、印刷装置２と印刷装置２に接続された情報処理装置（パソコン１）とによって構成される印刷システムの全体構成を概念的に表した一例を示す図である。
パソコン１は、入力インターフェース１１とＣＰＵ１２、ＲＯＭ１３、ＲＡＭ１４、外部記憶装置１５、出力インターフェース１６、表示部１７、キーボード１０、マウス１８、入出力インターフェース１９を有する。ＲＯＭ１３には初期化プログラムが入っており、外部記憶装置１５にはアプリケーションプログラム群とＯＳ（Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）、プリンタドライバやその他各種のデータが保存されている。ＲＡＭ１４は、外部記憶装置１５にストアされる各種プログラムがワークメモリとして使用する。
ＣＰＵ１２がプログラム等に基づいて処理を実行することによって後述するソフトウェア構成が実現されたり、後述するフローチャートの処理が実現されたりする。
印刷装置２は、入出力インターフェース２１とＲＡＭ２２、プリントエンジン２３、ＲＯＭ２４、ＣＰＵ２５を有する。入出力インターフェース２１は、パソコン１の入出力インターフェース１９に接続してある。ＲＡＭ２２は、ＣＰＵ２５の主メモリとワークメモリとして用いられ、受信した印刷ジョブを一旦保存するための受信バッファや各種のデータを保存する。プリントエンジン２３は、ＲＡＭ２２に保存されたデータに基づき印刷を行う。ＲＯＭ２４には各種の制御プログラムが使用するデータが入っており、ＣＰＵ２５はこれらの制御プログラムに従って印刷装置２の各部を制御する。
ここでは、例としてパソコン１と印刷装置２との処理分担を上記のように示したが、特にこの分担形態に限らず他の形態であってもよい。
なお、パソコン１は、印刷データ処理装置（コンピュータ）の一例である。
【００１１】
図２は、パソコン１のソフトウェア構成等の一例を示す図である。
ブラウザー２０１は、Ｗｅｂページを表示するためのアプリケーションである。ブラウザー２０１は、不図示のインターネットを介して不図示のＷＷＷサーバから受信した構造化文書を外部記憶装置１５へダウンロードし、それに基づいて表示部１７へ表示させる。Ｗｅｂページは、ＨＴＭＬやＸＨＴＭＬ等の構造化文書ファイルであり、テキストや画像等の要素がタグを用いて記述されている。
アドオン２０２は、ブラウザー２０１から呼び出されるプラグインソフトウェアである。アドオン２０２は、Ｗｅｂページの印刷やプレビュー、一部領域のクリップ等の処理の実行をユーザーから受け付けるユーザーインターフェースを持つ。アドオン２０２は、ユーザーからの入力に応じて、ブラウザー２０１に対し現在表示されているＷｅｂページを仮想ドライバ２０３に対して印刷するよう指示する。仮想ドライバ２０３は、特定の印刷装置に接続されず、スプールされた印刷データを外部記憶装置１５やＲＡＭ１４に保存するよう構成される仮想的なプリンタドライバである。アドオン２０２は、印刷を指示する際に指定する印刷設定情報の予め決められた場所に、印刷データの保存場所情報を含める。アドオン２０２が、仮想ドライバ２０３に対して、印刷データの保存場所情報を印刷設定情報に付加するよう依頼し、仮想ドライバ２０３が保存場所情報を含めるような構成としてもよい。ここで、印刷データの保存場所情報とは、例えばファイル名を含むフルパス等である。

【００１２】
ブラウザー２０１が作成した印刷データは、仮想ドライバ２０３によって外部記憶装置１５やＲＡＭ１４に保存される。このとき、仮想ドライバ２０３は、印刷設定情報中に含まれる印刷データの保存場所情報に従った場所に印刷データ２０４を保存する。
アドオン２０２は、仮想ドライバ２０３に印刷を指示した後、発行した印刷ジョブが残存するか否かを定期的に後述するスプーラに問い合わせる。印刷ジョブが残存しない場合、アドオン２０２は、編集アプリケーション２０５を起動し、印刷データ２０４と、アドオン２０２のユーザーインターフェースを経由してユーザーから入力された指示と、を編集アプリケーション２０５に渡す。アドオン２０２は、指定した印刷データの保存場所に印刷データが存在するかどうかを定期的に確認することで、印刷データの保存完了を判定してもよい。
編集アプリケーション２０５は、受け取った指示に応じて、印刷データの編集や表示を行い、最終的にプリンタドライバ２０６に対して印刷要求を発行する。プリンタドライバ２０６は、編集アプリケーション２０５によって編集された印刷データを受け取り、印刷装置２が解釈可能な印刷コマンドに変換する。プリンタドライバ２０６は、印刷コマンドを印刷装置２に送信する。印刷装置２は、受け取った印刷コマンドを元に印刷処理を実行する。
【００１３】
以上のように、本実施形態の印刷プログラムは、ブラウザー２０１に表示されたＷｅｂページを適宜編集して印刷することができる。ここで、仮想ドライバ２０３は、Ｗｅｂページを元に作成された中間データを取得することを目的としている。中間データは、その後編集される等の理由から、特定解像度で展開されたラスターデータではなく、テキストやグラフィックスの形式を維持したベクターデータであることが望ましい。ブラウザー２０１からの印刷により作成される印刷データは、通常、ベクターデータであることから、本印刷プログラムでは、中間データとしてその印刷データを用いる。このような構成により、アドオン２０２は、ブラウザー２０１からＷｅｂページを元に作成された中間データを直接取得できないような場合にも、ベクターデータである印刷データを中間データとして取り出すことができる。
また、本実施形態の印刷プログラムは、アドオン２０２が印刷データの保存場所を仮想ドライバ２０３に対して指定する構成とすることで、仮想ドライバ２０３とアドオン２０２や編集アプリケーション２０５との間で通信が発生しない。プリンタドライバである仮想ドライバ２０３と、ブラウザー２０１のプラグインソフトウェアであるアドオン２０２という特殊な条件下で動作する２つのプログラム同士の通信は避けることが好ましい。更に、アドオン２０２は各種制御を一元的に行うことができ、ユーザーからの各種要求のハンドリングを容易に行うことができる。
【００１４】
図３は、実施形態１の印刷プログラム中の仮想ドライバ２０３の全体構成を概念的に表した一例を示す図である。以下、一般的に普及しているパソコンにＭｉｃｒｏｓｏｆｔ社のＷｉｎｄｏｗｓ（登録商標）をＯＳとして使用する印刷システムについて説明する。また、スプールデータフォーマットとしてＭｉｃｒｏｓｏｆｔ社が開発したオープン規格の電子文書フォーマットの一つであるＸＭＬ Ｐａｐｅｒ Ｓｐｅｃｉｆｉｃａｔｉｏｎ（以下、ＸＰＳ）を用いる印刷システムについて説明する。以後、このような印刷システム上で動作するプリンタドライバをＸＰＳドライバと呼ぶ。
ブラウザー２０１等のアプリケーション３０１が作成した印刷データは、ＯＳの印刷サポート機能３０２を介しスプールデータ３０５としてスプーラ３０３のプリントキュー３０４に一時的に蓄えられる。アプリケーション３０１の作成する印刷データの種類によっては、印刷データはＯＳの印刷サポート機能３０２は経由することなく、そのままスプールデータ３０５として蓄えられる。スプールデータ３０５は、名称等を付加した印刷ジョブとしてスプーラ３０３により管理される。印刷ジョブは、順次、仮想ドライバ２０３によって処理される。アプリケーション３０１は、印刷指示を行う際に、ユーザーインター

フェースモジュール３０６から返却された印刷設定情報を印刷ジョブに付加する。印刷設定情報は、アプリケーション３０１からの印刷開始指示を行う前に予め設定された情報である。

【００１５】

仮想ドライバ２０３に渡された印刷データは、先ずチケットフィルター３０７によって処理される。ここでフィルターとは、入力されたデータをもとに、加工、変換、無変換、生成等の工程を介して、何らかのデータを出力する機能を有するプログラムのことで、後述するＯＳのフィルター管理機能から呼び出される。ＸＰＳドライバは、その目的や機能に応じて、任意の個数のフィルターを任意の順番で構成することが許容されている。印刷データ中のＰｒｉｎｔＴｉｃｋｅｔ（ＰＴ。ＸＰＳドキュメントにおける印刷設定情報）を参照し、印刷データを保存する場所等の情報を取得する。データフィルター３０８は、入力した印刷データを、前述の印刷データを保存する場所に保存する。

通常、最後のフィルターは、入力した印刷データを印刷装置２が解釈可能な印刷コマンドに変換し、出力する。出力した印刷コマンドは、プリンタドライバに関連付けられたポートに対応するポートドライバーを経由して印刷装置２に到達する。本実施形態における仮想ドライバ２０３の出力ポートは、ＮＵＬポートとしてある。ＮＵＬポートは、入力したデータを全て無視するだけの論理的なポートである。仮想ドライバ２０３の役割は印刷データを外部記憶装置１５に格納することであり、それはデータフィルター３０８によって実行される。したがって、データフィルター３０８は、あらゆるデータの出力を行わない。そのため、仮想ドライバ２０３の出力ポートは、ＮＵＬポート以外であっても問題無い。また、データフィルター３０８は、印刷データと印刷データの保存場所情報とを出力し、ポートドライバーが印刷データを保存するように構成してもよい。

図３の例では、データフィルター３０８は、チケットフィルター３０７の後に位置している。

【００１６】

前述したように、ＸＰＳドライバは任意の個数のフィルターを任意の順番で構成することが許容されている。フィルター間でデータの受け渡しを行うインターフェースには、ＸＰＳ　ｓｔｒｅａｍ形式と、ＸＰＳ　ｄｏｃｕｍｅｎｔ形式の２種類が存在する。

ＸＰＳ　ｓｔｒｅａｍ形式のインターフェースでは、印刷データはバイトストリームとして扱われる。ＸＰＳ　ｄｏｃｕｍｅｎｔ形式のインターフェースでは、印刷データであるＸＰＳドキュメントがＸＰＳパート単位で扱われる。ここでいうＸＰＳパートとはＦｉｘｅｄＤｏｃｕｍｅｎｔＳｅｑｕｅｎｃｅ（ＦＤＳ）、ＦｉｘｅｄＤｏｃｕｍｅｎｔ（ＦＤ）、ＦｉｘｅｄＰａｇｅ（ＦＰ）とそれらに付随するＰＴや、画像、フォント等のリソースのことを指す。各フィルターは、入力と出力それぞれに対して、どちらかのインターフェースを利用するよう予め構成される。入力にＸＰＳ　ｄｏｃｕｍｅｎｔ形式のインターフェースを利用するフィルターは、ＸＰＳドキュメントを解析してＸＰＳパートを抽出する必要が無い。そのため、ＸＰＳ　ｓｔｒｅａｍ形式のフィルターに比べ、ＦＰの変更が必要なレイアウト処理等の処理を容易に実施可能である。

【００１７】

図４は、実施形態１のフィルター構成例とそのときの動作を示す概念図である。チケットフィルター３０７は、入出力インターフェースとしてＸＰＳ　ｄｏｃｕｍｅｎｔ形式を利用する。データフィルター３０８は、入出力インターフェースとしてＸＰＳ　ｓｔｒｅａｍ形式を利用する。

チケットフィルター３０７は、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第一のフィルターの一例である。データフィルター３０８は、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第二のフィルターの一例である。

スプールデータとして格納されているＸＰＳドキュメント４０１は、チケットフィルター３０７に送られる。ここで、チケットフィルター３０７の入力インターフェースは、ＸＰＳ　ｄｏｃｕｍｅｎｔ形式なので、ＸＰＳドキュメントはＯＳのフィルター管理機能４

０２により複数のＸＰＳパート４０３に分解されチケットフィルター３０７に送られる。チケットフィルター３０７は、分解されたＸＰＳパートを、順次、受信することができる。例えば、スプールデータとして、ＸＰＳドキュメントの一部しか格納されていない状態であっても、チケットフィルター３０７は、格納が完了しているＸＰＳパートから順次、受信することができる。

チケットフィルター３０７は、ＸＰＳパートのうち、ＰＴを参照し、アプリケーションによって指定された印刷データの保存場所情報を取得する。チケットフィルター３０７は、印刷データの保存場所情報をデータフィルター３０８に通知する。チケットフィルター３０７の出力インターフェースもＸＰＳ　ｄｏｃｕｍｅｎｔ形式なので、チケットフィルター３０７は、受信したＸＰＳパートをＸＰＳパート単位で出力する。

【００１８】

チケットフィルター３０７が出力した複数のＸＰＳパート４０４は、データフィルター３０８に送られる。ここで、データフィルター３０８の入力インターフェースは、ＸＰＳ　ｓｔｒｅａｍ形式なので、ＯＳのフィルター管理機能４０２は、複数のＸＰＳパート４０４をＸＰＳドキュメント４０５に変換する。データフィルター３０８は、変換されたＸＰＳドキュメント４０５をバイトストリームとして受け取る。データフィルター３０８は、入力したＸＰＳドキュメント４０５を、チケットフィルター３０７により通知された印刷データの保存場所に保存する。データフィルター３０８の出力インターフェースもＸＰＳ　ｓｔｒｅａｍ形式なので、データを出力する場合は、バイトストリームとして出力する。但し本実施形態におけるデータフィルター３０８は、前述の通りあらゆるデータを出力しないように構成する。

この構成により、ＰＴに指定された印刷データの保存場所にＸＰＳドキュメントを保存するという目的を容易に実現することができる。何らかの機能を実現するために、それぞれのフィルター前後に別のフィルターが存在してもよい。仮に入出力インターフェースとしてＸＰＳ　ｓｔｒｅａｍを利用するフィルターが全ての処理を行う場合、ＸＰＳドキュメントの構造に従い独自でＸＰＳパートに展開する処理が必要であるが、これは技術的に容易では無い。また、バージョンアップ等に伴いＸＰＳドキュメントの構造が変化した場合等に、独自の展開処理では正しく展開できない可能性がある。

【００１９】

図５は、チケットフィルター３０７の印刷データの保存場所情報の取得処理の一例を示すフローチャートである。

前述の通り、チケットフィルター３０７は、入出力インターフェースとして、ＸＰＳ　ｄｏｃｕｍｅｎｔ形式を利用する。チケットフィルター３０７は、ＸＰＳパートを、ＯＳのフィルター管理機能４０２に対して要求する（Ｓ５０１）。ＯＳのフィルター管理機能４０２は、要求に応じて、ＸＰＳパートを準備ができ次第、１つずつチケットフィルター３０７に渡す。

チケットフィルター３０７は、Ｓ５０１でＸＰＳパートを取得できたかどうかを判定する（Ｓ５０２）。

ＸＰＳパートを取得できた場合、チケットフィルター３０７は、Ｓ５０１で入力したＸＰＳパートがＦＤＳであるかどうか判定する（Ｓ５０３）。

Ｓ５０３で入力したＸＰＳパートがＦＤＳであると判定した場合、チケットフィルター３０７は、ＦＤＳに関連付けられたＰＴを更に取得する（Ｓ５０４）。

【００２０】

チケットフィルター３０７は、取得したＰＴに印刷設定の一つとして記載されている印刷データの保存場所情報を読み出し、ＰｒｏｐｅｒｔｙＢａｇに格納する（Ｓ５０５）。ＰｒｏｐｅｒｔｙＢａｇは、ＯＳのフィルター管理機能４０２により提供されるデータ管理機能である。全てのフィルターは、ＰｒｏｐｅｒｔｙＢａｇに、任意のデータを格納したり、取得したりできる。

チケットフィルター３０７は、Ｓ５０１で入力したＸＰＳパートを次のフィルターに渡すよう、ＯＳのフィルター管理機能４０２に対して指示する（Ｓ５０６）。チケットフィ

ルター３０７が出力した印刷データは、ＯＳのフィルター管理機能４０２によって、次のフィルターの入力インターフェースに応じた形式に変換された後、次のフィルターに渡される。

Ｓ５０３で入力したＸＰＳパートがＦＤＳではないと判定した場合、チケットフィルター３０７は、Ｓ５０４とＳ５０５との処理をスキップし、Ｓ５０６の処理を実行する。その後、チケットフィルター３０７は、Ｓ５０１の処理に戻り、取得したＸＰＳに対してＳ５０１からＳ５０６までの処理を繰り返す。Ｓ５０２で、ＸＰＳパートを取得できなかったと判定した場合、チケットフィルター３０７は処理を終了する。

【００２１】

図６は、データフィルター３０８の印刷データ保存処理の一例を示すフローチャートである。

前述の通り、データフィルター３０８は、入出力インターフェースとして、ＸＰＳ ｓｔｒｅａｍ形式を利用する。データフィルター３０８は、ＰｒｏｐｅｒｔｙＢａｇに、印刷データの保存場所情報がないか問い合わせる（Ｓ６０１）。印刷データの保存場所情報は、チケットフィルター３０７のＳ５０５の処理によってＰｒｏｐｅｒｔｙＢａｇに格納される。

データフィルター３０８は、Ｓ６０１の処理が成功したか否か判定する（Ｓ６０２）。

Ｓ６０２で、印刷データの保存場所情報の取得に失敗したと判定した場合、データフィルター３０８は、一定間隔、例えば１００ｍｓｅｃの間スリープした後、再度ＰｒｏｐｅｒｔｙＢａｇから印刷データの保存場所情報の取得を試みる（Ｓ６０１）。Ｓ６０２で、印刷データの保存場所情報の取得に成功したと判定した場合、データフィルター３０８は、一定サイズ、例えば２ＭＢｙｔｅ分の印刷データを、ＯＳのフィルター管理機能４０２に対して要求する（Ｓ６０３）。ＯＳのフィルター管理機能４０２は、印刷データの先頭から順に要求されたサイズの印刷データが準備でき次第、データフィルター３０８に渡す。

【００２２】

次に、データフィルター３０８は、取得した一定サイズの印刷データを、外部記憶装置１５のＳ６０１で取得した印刷データの保存場所情報に従った場所に、ファイルとして書き出す（Ｓ６０４）。外部記憶装置１５は、記憶領域の一例である。

次に、データフィルター３０８は、全ての印刷データを入力したかどうか判定する（Ｓ６０５）。ＯＳのフィルター管理機能４０２は、データフィルター３０８による印刷データの要求（Ｓ６０３）がデータの末尾まで到達した場合、そのことを示す情報を返却する。データフィルター３０８は、この情報を参照し、全ての印刷データを入力したかどうかを判定する。Ｓ６０５で、全ての印刷データを入力していないと判定した場合、データフィルター３０８は、入力済みの印刷データの続きの印刷データの入力とファイルへの書き出し処理を行う（Ｓ６０３－Ｓ６０４）。Ｓ６０５で、全ての印刷データを入力したと判定した場合、データフィルター３０８は、処理を終了する。

【００２３】

図６のフローチャートでは、データフィルター３０８は、印刷データ全体を保存することを目的としている。もし印刷データの一部のみを保存することが目的である場合、データフィルター３０８は、Ｓ６０５の判定前に、印刷データのうち所望する部分を既に入力しているかどうかを判定し、入力していた場合は処理を終了する。仮想ドライバ２０３が印刷データの一部のみを保存することを目的とする場合、このように構成することで、仮想ドライバ２０３の処理を更に高速化できる。

本実施形態の仮想ドライバ２０３は、ＯＳの機能を利用した簡易、かつ、堅牢性の高い構成で、印刷データを、印刷データ中の印刷設定情報に記載の印刷データ保存場所に保存することができる。

なお、本実施形態の構成は、印刷データを印刷データ中の印刷設定情報に従って処理する他の目的にも適用可能である。

【００２４】

＜実施形態２＞
図７は、実施形態２の印刷プログラム中の仮想ドライバ２０３の全体構成を概念的に表した一例を示す図である。図７に示す通り、本実施形態における仮想ドライバ２０３では、データフィルター３０８とチケットフィルター３０７との処理順が、実施形態１における仮想ドライバ２０３とは逆である。それらのフィルターの処理の詳細は後述する。フィルターの処理順と処理内容を除き、図７で示した仮想ドライバ２０３の構成は、実施形態１の図３で示した構成と同等である。なお、本実施形態における印刷システムの全体構成は実施形態１の図１と同等である。以後、特に言及が無い点は、実施形態１と同等の構成である。
図７の例では、チケットフィルター３０７は、データフィルター３０８の後に位置している。
実施形態１の図４で示したとおり、連続するフィルター間で、前のフィルターの出力インターフェースと、後のフィルターの入力インターフェースと、が異なる場合、ＯＳのフィルター管理機能４０２による変換処理が動作する。ＸＰＳドキュメントの複雑さに依存するが、この変換処理は一定の処理時間を要する。したがって、ＸＰＳドライバで高速な処理を期待する場合、この変換処理ができるだけ発生しないようにフィルターを構成することが望ましい。
【００２５】
図８は、実施形態２のフィルター構成例とそのときの動作を示す概念図である。前述の通り、本実施形態における仮想ドライバ２０３は、データフィルター３０８とチケットフィルター３０７という２つのフィルターを持つ。データフィルター３０８は、入出力インターフェースとしてＸＰＳ　ｓｔｒｅａｍ形式を利用する。したがって、スプールされたＸＰＳドキュメント８０１は、ＯＳのフィルター管理機能４０２により変換されることなく、バイトストリームとしてデータフィルター３０８に渡される。データフィルター３０８は、入力したＸＰＳドキュメント８０２を外部記憶装置１５やＲＡＭ１４に保存すると共に、バイトストリームとして出力する。
チケットフィルター３０７は、入出力インターフェースとしてＸＰＳ　ｄｏｃｕｍｅｎｔ形式を利用する。したがって、データフィルター３０８が出力したＸＰＳドキュメント８０３は、ＯＳのフィルター管理機能４０２により複数のＸＰＳパート８０４に分解され、チケットフィルター３０７に渡される。チケットフィルター３０７は、ＸＰＳパートのうち、ＰＴを参照し、アプリケーションによって指定された印刷データの保存場所情報を取得する。チケットフィルター３０７は、印刷データの保存場所情報をデータフィルター３０８に通知した後、それ以降のＸＰＳパートを受信せず終了する。データフィルター３０８は、保存したＸＰＳドキュメントを、チケットフィルター３０７により通知された印刷データの保存場所に、移動、又は複製する。
データフィルター３０８は、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第一のフィルターの一例である。チケットフィルター３０７は、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第二のフィルターの一例である。
【００２６】
この構成により、ＰＴに指定された印刷データの保存場所にＸＰＳドキュメントを保存するという目的を容易に実現することができる。更に、ＯＳのフィルター管理機能４０２によるＸＰＳドキュメントの変換回数を一回に抑えることにより、処理は高速である。処理の高速化において重要なのはデータフィルター３０８とチケットフィルター３０７との処理順であり、何らかの機能を実現するために、それぞれのフィルター前後に別のフィルターが存在してもよい。
【００２７】
図９は、データフィルター３０８の印刷データ保存処理の一例を示すフローチャートである。
まず、データフィルター３０８は、後述する、格納した印刷データが残っていた場合、

それらを削除する（Ｓ９０１）。この処理は、データフィルター３０８の処理がエラー終了する等して残った、以後の処理では不必要なファイルを削除することを目的とする。
前述の通り、データフィルター３０８は、入出力インターフェースとして、ＸＰＳ ｓｔｒｅａｍ形式を利用する。データフィルター３０８は、一定サイズ、例えば２ＭＢｙｔｅ分の印刷データを、ＯＳのフィルター管理機能４０２に対して要求する（Ｓ９０２）。
次に、データフィルター３０８は、入力した一定サイズの印刷データを次のフィルターに渡すよう、ＯＳのフィルター管理機能４０２に対して指示する（Ｓ９０３）。
次に、データフィルター３０８は、取得した一定サイズの印刷データを、外部記憶装置１５にファイルとして書き出す（Ｓ９０４）。一例として、データフィルター３０８は、仮想ドライバ２０３が管理する規定のフォルダの下にランダムな名称でファイルを作成する。なお、印刷データは、外部記憶装置１５にではなく、ＲＡＭ１４上に記憶されてもよい。

【００２８】
次に、データフィルター３０８は、全ての印刷データを入力したかどうか判定する（Ｓ９０５）。Ｓ９０５で、全ての印刷データを入力していないと判定した場合、データフィルター３０８は、入力済みの印刷データの続きの印刷データの入出力とファイルへの書き出し処理を行う（Ｓ９０２－Ｓ９０４）。
Ｓ９０５で、全ての印刷データを入力したと判定した場合、データフィルター３０８は、ＰｒｏｐｅｒｔｙＢａｇに、印刷データの保存場所情報がないか問い合わせる（Ｓ９０６）。印刷データの保存場所情報は、後述するチケットフィルター３０７の処理によってＰｒｏｐｅｒｔｙＢａｇに格納される。
データフィルター３０８は、Ｓ９０６の処理が成功したか否か判定する（Ｓ９０７）。
Ｓ９０７で、印刷データの保存場所情報の取得に失敗したと判定した場合、データフィルター３０８は、一定間隔、例えば１００ｍｓｅｃの間スリープした後、再度、ＰｒｏｐｅｒｔｙＢａｇから印刷データの保存場所情報の取得を試みる（Ｓ９０６）。
Ｓ９０７で、印刷データの保存場所情報の取得に成功したと判定した場合、データフィルター３０８は、Ｓ９０４で格納した印刷データを、取得した印刷データの保存場所に移動する（Ｓ９０８）。なお、この移動処理は、チケットフィルター３０７が実施してもよい。その場合、データフィルター３０８は、Ｓ９０４で格納した印刷データの場所を示す情報を、ＰｒｏｐｅｒｔｙＢａｇに格納する。チケットフィルター３０７は、ＰｒｏｐｅｒｔｙＢａｇから取得した場所に格納されている印刷データを、ＰＴ内の印刷データの保存場所に移動する。また、この移動処理は、移動処理専用のフィルターが実施してもよい。

【００２９】
図９のフローチャートでは、データフィルター３０８は、印刷データ全体を保存することを目的としている。もし印刷データの一部のみを保存することが目的である場合、データフィルター３０８は、Ｓ９０５の判定の前に、Ｓ９０６とＳ９０７との処理を行う。その後、印刷データのうち所望する部分を既に入力していれば、データフィルター３０８は、印刷データの入出力処理（Ｓ９０２－Ｓ９０４）を途中で止め、Ｓ９０８の処理を行い終了する。仮想ドライバ２０３が印刷データの一部のみを保存することが目的である場合、このように構成することで、仮想ドライバ２０３の処理を更に高速化できる。

【００３０】
図１０は、チケットフィルター３０７の印刷データの保存場所情報の取得処理の一例を示すフローチャートである。
前述の通り、チケットフィルター３０７は、入出力インターフェースとして、ＸＰＳ ｄｏｃｕｍｅｎｔ形式を利用する。チケットフィルター３０７は、ＸＰＳパートを、ＯＳのフィルター管理機能４０２に対して要求する（Ｓ１００１）。
チケットフィルター３０７は、Ｓ１００１で入力したＸＰＳパートがＦＤＳであるかどうか判定する（Ｓ１００２）。Ｓ１００２で入力したＸＰＳパートがＦＤＳでは無いと判定した場合、チケットフィルター３０７は、ＦＤＳを入力するまでＸＰＳパートの要求を

繰り返す（Ｓ１００１－Ｓ１００２）。ＦＤＳは、ＸＰＳドキュメントの論理構造においてＦＰやＦＤの上位に存在する。ＯＳのフィルター管理機能４０２は、ＸＰＳドキュメントの論理構造における上位パーツから順番に渡すので、チケットフィルター３０７は、ＸＰＳパートの入力を要求して間もなく、ＦＤＳを入力することができる。

【００３１】

Ｓ１００２で入力したＸＰＳパートがＦＤＳであると判定した場合、チケットフィルター３０７は、ＦＤＳに関連付けられたＰＴを更に取得する（Ｓ１００３）。

チケットフィルター３０７は、取得したＰＴに印刷設定の一つとして記載されている印刷データの保存場所情報を読み出し、ＰｒｏｐｅｒｔｙＢａｇに格納する（Ｓ１００４）。このように、チケットフィルター３０７は、ＦＳＤ以降のＸＰＳパートの入力を全部又は一部要求せず、処理を終了する。

ＯＳのフィルター管理機能４０２がＸＰＳドキュメントを変換処理中であっても、全てのフィルターが処理を終了した場合、印刷ジョブは終了する。チケットフィルター３０７が不必要なパーツを全部又は一部要求しないことで、データフィルター３０８による印刷データの保存処理（Ｓ９０８）が完了後、プリントキュー３０４の印刷ジョブも速やかに終了する。それにより、複数の印刷ジョブが連続して投入された場合の、仮想ドライバ２０３の生産性を向上させることができる。

【００３２】

本実施形態における印刷システムでは、ユーザーがアドオン２０２に対して要求した印刷やプレビュー等さまざまな操作によって、仮想ドライバ２０３に印刷データの保存要求が発生する。仮想ドライバ２０３の構成により、印刷データの保存は高速に実行され、印刷システム全体として、レスポンスのよい操作感をユーザーに提供することができる。

なお、本実施形態の構成を、印刷データを印刷データ中の印刷設定情報に従って処理する他の目的にも適用可能である。

【００３３】

＜実施形態３＞

図１１は、実施形態３のパソコン１上で実行される印刷プログラムにおいて、プリンタドライバ１１０７を中心に全体構成を概念的に表した一例を示す図である。なお、本実施形態における印刷システムの全体構成は実施形態１の図１と同等である。以後、特に言及が無い点は、実施形態１と同等の構成である。

プリンタドライバ１１０７は、アプリケーション１１０１からの印刷要求を受け、印刷装置２に印刷コマンドを送信する、標準的なプリンタドライバである。したがって、図２に示した実施形態１の印刷プログラムの構成に限定せず、任意のアプリケーションから印刷要求を受ける。

アプリケーション１１０１からユーザーインターフェースモジュール１１０６までは、図３の３０１から３０６までと同等である。プリンタドライバ１１０７は、プレビューフィルター１１０８とレンダーフィルター１１０９との２つのフィルターで構成される。プレビューフィルター１１０８は、入出力インターフェースとして、ＸＰＳ　ｓｔｒｅａｍ形式を利用するフィルターである。プレビューフィルター１１０８は、印刷設定情報中の指定に応じて、印刷プレビュー表示モジュール１１１０を別プロセスで起動する。印刷プレビュー表示モジュール１１１０は、プレビューフィルター１１０８から受け取った印刷データを元に、想定される印刷結果を表す印刷プレビューを表示部１７に表示する。印刷プレビュー表示モジュール１１１０、ユーザーから印刷の指示を受けると、印刷データ中の印刷設定情報の印刷プレビューの表示指示をオフに変更した上で、印刷データを再度、スプールデータ１１０５としてプリントキュー１１０４に追加する。

レンダーフィルター１１０９は、入出力インターフェースとして、ＸＰＳ　ｄｏｃｕｍｅｎｔ形式を利用するフィルターである。レンダーフィルター１１０９は、印刷データを元に印刷装置２が解釈可能な印刷コマンドを作成し、印刷装置２に送信する。印刷装置２は、受け取った印刷コマンドを元に印刷処理を実行する。

なお、プリンタドライバ１１０７は、更に印刷データのレイアウト処理を行うフィルタ

ー等を有する構成としてもよい。
図１１のレンダーフィルター１１０９は、プレビューフィルター１１０８の後に位置している。
プレビューフィルター１１０８は、入出力インターフェースとしてｓｔｒｅａｍ形式のインターフェースを持つ第一のフィルターの一例である。レンダーフィルター１１０９は、入出力インターフェースとしてｄｏｃｕｍｅｎｔ形式のインターフェースを持つ第二のフィルターの一例である。
【００３４】
図１２は、プレビューフィルター１１０８の印刷プレビュー表示モジュール起動処理の一例を示すフローチャートである。
Ｓ１２０１からＳ１２０７までの処理は、実施形態１におけるデータフィルター３０８のＳ９０１からＳ９０７までの処理と同等である。
但し、プレビューフィルターは、ＰｒｏｐｅｒｔｙＢａｇから印刷データの保存場所情報ではなく、印刷プレビューの表示情報（印刷プレビュー表示情報）を取得する（Ｓ１２０６）。
プレビューフィルター１１０８は、印刷プレビューの表示情報を参照し、印刷プレビューの表示が指示されているか否かを判定する（Ｓ１２０８）。
Ｓ１２０８で印刷プレビューの表示が指示されていると判定した場合、プレビューフィルター１１０８は、印刷プレビュー表示モジュール１１１０を起動する（Ｓ１２０９）。この際、プレビューフィルター１１０８は、Ｓ１２０４で保存した印刷データを印刷プレビュー表示モジュール１１１０に渡す。
Ｓ１２０８で印刷プレビューの表示が指示されていないと判定した場合、プレビューフィルター１１０８は、Ｓ１２０４で作成したファイルを削除する（Ｓ１２１０）。
なお、これらの処理は、レンダーフィルター１１０９が実施してもよいし、専用のフィルターが実施してもよい。
【００３５】
図１３は、レンダーフィルター１１０９の処理の一例を示すフローチャートである。
Ｓ１３０１からＳ１３０４までの処理は、実施形態１におけるチケットフィルター３０７のＳ１００１からＳ１００４までの処理と同等である。
但し、レンダーフィルター１１０９は、ＰｒｏｐｅｒｔｙＢａｇに、印刷データ保存場所情報ではなく、印刷プレビューの表示情報を格納する（Ｓ１３０４）。
レンダーフィルター１１０９は、印刷プレビューの表示情報を参照し、印刷プレビューの表示が指示されているか否かを判定する（Ｓ１３０５）。
Ｓ１３０５で印刷プレビューの表示が指示されていると判定した場合、レンダーフィルター１１０９は、処理を終了する。
Ｓ１３０５で印刷プレビューの表示が指示されていないと判定した場合、レンダーフィルター１１０９は、印刷データを元に印刷コマンドを作成し、印刷装置２に送信する（Ｓ１３０６）。
【００３６】
プリンタドライバ１１０７は、実施形態２における仮想ドライバ２０３と同等のフィルター構成によって、印刷プレビューの表示を高速に行うことができる。更に、印刷プレビューの表示が指示されていない場合にも、ＯＳのフィルター管理機能４０２により、印刷処理に不必要なＸＰＳドキュメントの変換処理が動作することは無い。それにより、プリンタドライバ１１０７は、印刷処理に必要最小限な構成とした場合と同等の速度で印刷処理を実行することができる。
なお、本実施形態の構成を、印刷データを印刷データ中の印刷設定情報に従って処理する他の目的にも適用可能である。
【００３７】
＜実施形態４＞
実施形態３におけるプリンタドライバ１１０７は、更に実施形態２における仮想ドライ

バ２０３の機能を併せ持つよう構成してもよい。この場合、プリンタドライバ１１０７は、印刷設定情報に印刷データ保存場所情報が指定されているか否かも判定し、指定されていた場合は仮想ドライバとして動作し、印刷処理を行わずに終了する。これにより、１つのプリンタドライバと追加のアプリケーションで、図２の印刷プログラムを構成することができる。この構成により、ユーザーに対し、仮想ドライバは存在しないよう隠ぺいすることができる。

【００３８】

＜その他の実施形態＞

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ（又はＣＰＵやＭＰＵ等）がプログラムを読み出して実行する処理である。

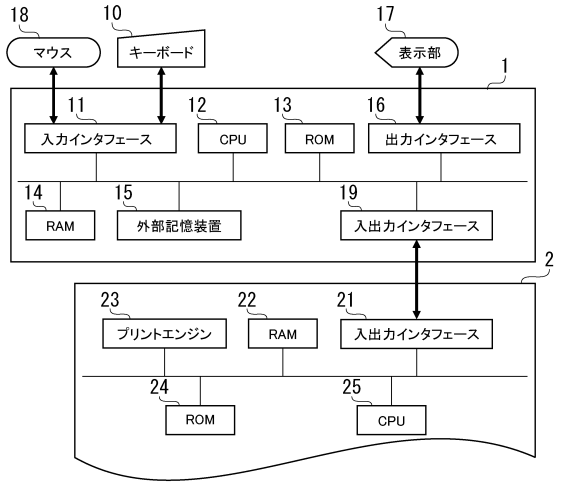
【００３９】

以上、上述した各実施形態によれば、簡易、かつ、堅牢性の高い構成で、印刷データを、印刷データ中の印刷設定情報に従って処理することができる。

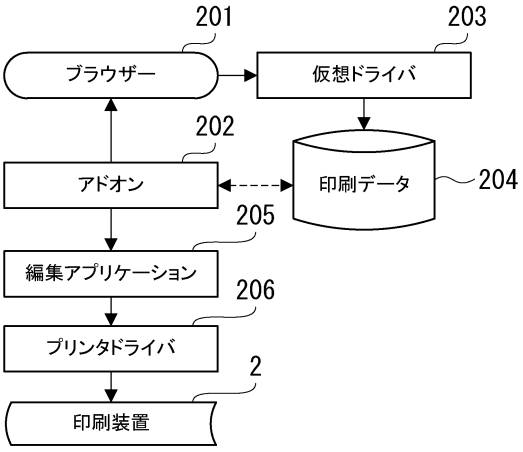
【００４０】

以上、本発明の好ましい実施形態について詳述したが、本発明は係る特定の実施形態に限定されるものではなく、特許請求の範囲に記載された本発明の要旨の範囲内において、種々の変形・変更が可能である。

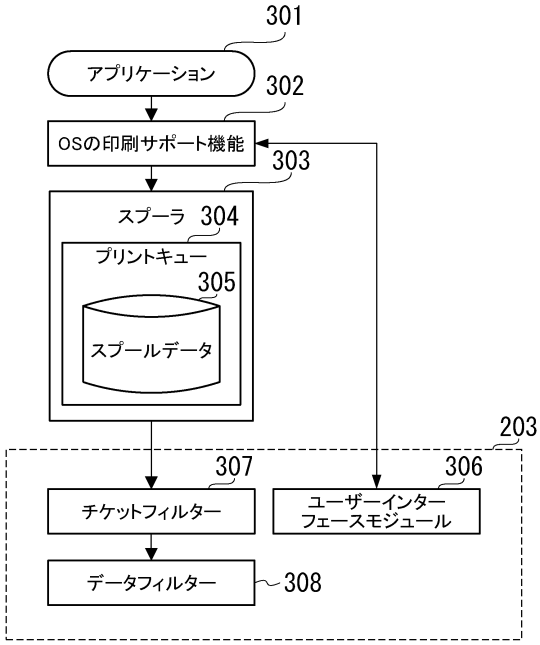
【図１】



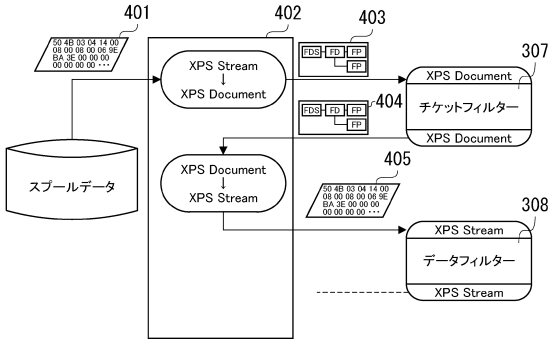
【図２】



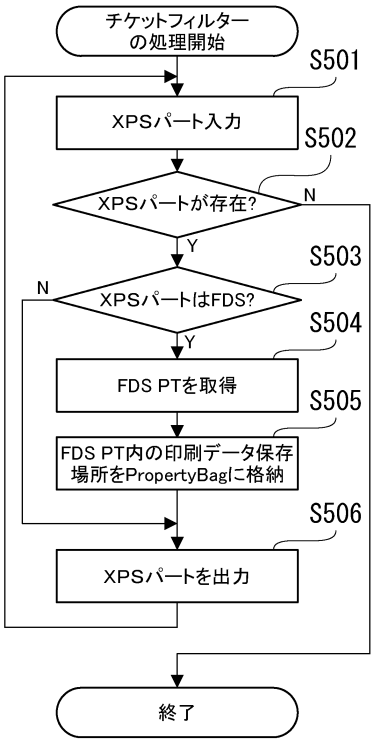
【図３】


301
アプリケーション
302
OSの印刷サポート機能
303
スプーラ
304
プリントキュー
305
スプールデータ
203
307
チケットフィルター
306
ユーザーインターフェースモジュール
データフィルター
308

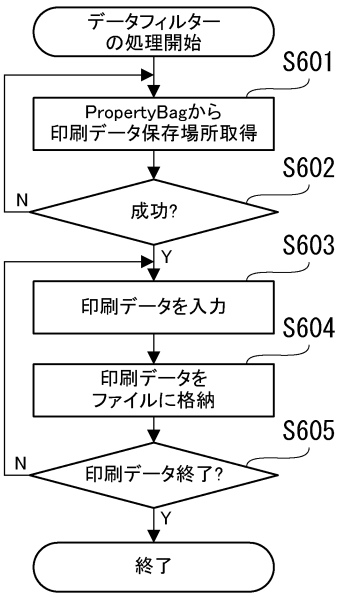
【図４】


401
402
403
XPS Stream
XPS Document
FDS
FD
FP
307
XPS Document
チケットフィルター
XPS Document
404
スプールデータ
XPS Document
XPS Stream
405
308
XPS Stream
データフィルター
XPS Stream

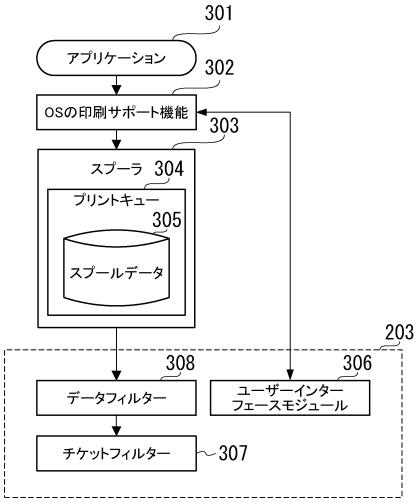
【図５】


チケットフィルターの処理開始
S501
XPSパート入力
S502
XPSパートが存在?
N
Y
S503
XPSパートはFDS?
N
Y
S504
FDS PTを取得
S505
FDS PT内の印刷データ保存場所をPropertyBagに格納
S506
XPSパートを出力
終了

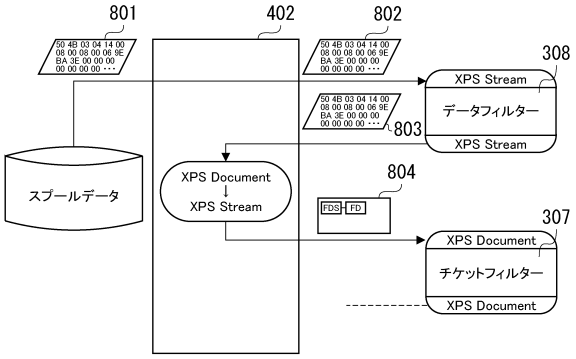
【図６】


データフィルターの処理開始
S601
PropertyBagから印刷データ保存場所取得
S602
成功?
N
Y
S603
印刷データを入力
S604
印刷データをファイルに格納
S605
印刷データ終了?
N
Y
終了

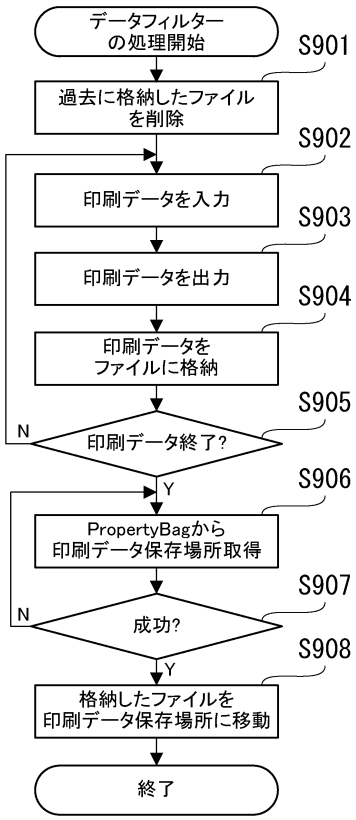
【図７】


301
アプリケーション
302
OSの印刷サポート機能
303
スプーラ 304
プリントキュー
305
スプールデータ
203
308
データフィルター
306
ユーザーインターフェースモジュール
チケットフィルター
307

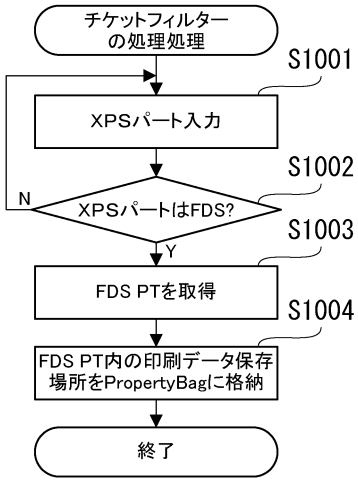
【図８】


801
402
802
308
XPS Stream
データフィルター
XPS Stream
803
スプールデータ
XPS Document
XPS Stream
804
FDS FD
307
XPS Document
チケットフィルター
XPS Document

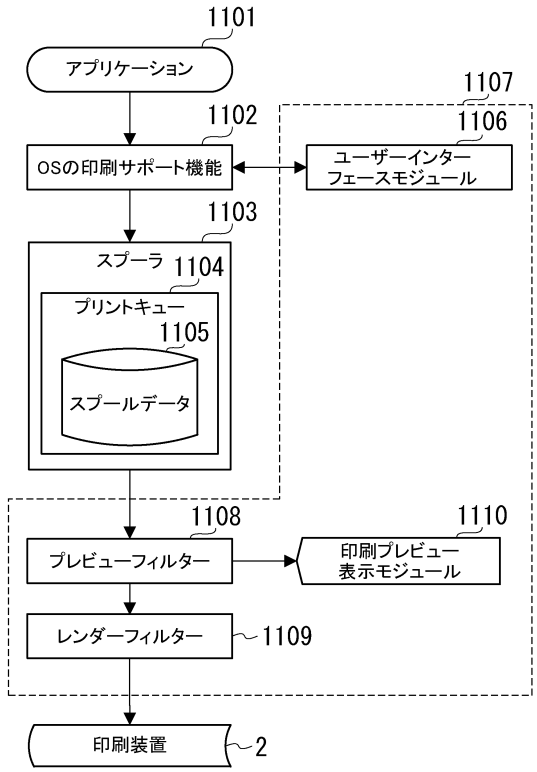
【図９】


データフィルターの処理開始
S901
過去に格納したファイルを削除
S902
印刷データを入力
S903
印刷データを出力
S904
印刷データをファイルに格納
S905
印刷データ終了?
N
Y
S906
PropertyBagから印刷データ保存場所取得
S907
成功?
N
Y
S908
格納したファイルを印刷データ保存場所に移動
終了

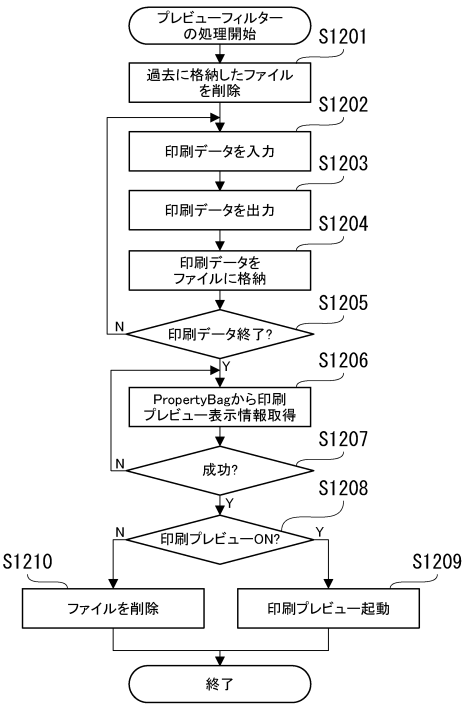
【図１０】


チケットフィルターの処理処理
S1001
XPSパート入力
S1002
XPSパートはFDS?
N
Y
S1003
FDS PTを取得
S1004
FDS PT内の印刷データ保存場所をPropertyBagに格納
終了

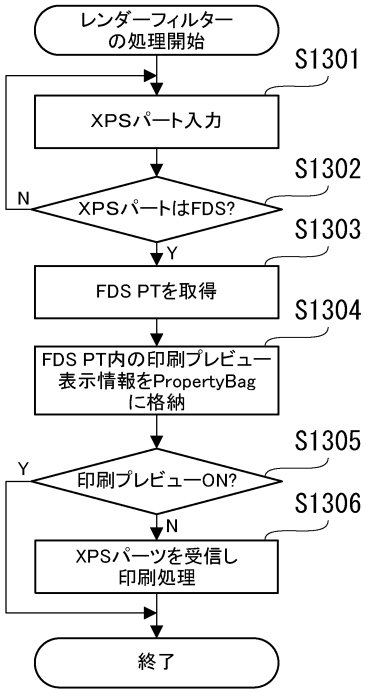
【図１１】


1101
アプリケーション
1107
1102
1106
OSの印刷サポート機能
ユーザーインター
フェースモジュール
1103
スプーラ
1104
プリントキュー
1105
スプールデータ
1108
1110
プレビューフィルター
印刷プレビュー
表示モジュール
レンダーフィルター
1109
印刷装置
2

【図１２】


プレビューフィルター
の処理開始
S1201
過去に格納したファイル
を削除
S1202
印刷データを入力
S1203
印刷データを出力
S1204
印刷データを
ファイルに格納
S1205
印刷データ終了?
N
Y
S1206
PropertyBagから印刷
プレビュー表示情報取得
S1207
成功?
N
Y
S1208
印刷プレビューON?
N
Y
S1210
S1209
ファイルを削除
印刷プレビュー起動
終了

【図１３】


レンダーフィルター
の処理開始
S1301
XPSパート入力
S1302
XPSパートはFDS?
N
Y
S1303
FDS PTを取得
S1304
FDS PT内の印刷プレビュー
表示情報をPropertyBag
に格納
S1305
印刷プレビューON?
Y
N
S1306
XPSパーツを受信し
印刷処理
終了

フロントページの続き

(56)参考文献 特開２００９－２８２８４３（ＪＰ，Ａ）
特開２００９－１３４６４５（ＪＰ，Ａ）
特開２００８－１５２７２８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　３／１２