

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5489619号
(P5489619)

(45) 発行日 平成26年5月14日 (2014. 5. 14)

(24) 登録日 平成26年3月7日 (2014. 3. 7)

(51) Int. Cl.

F I

G 0 6 F 3/12 (2006.01)

G 0 6 F 3/12

C

G 0 6 F 3/12

D

G 0 6 F 3/12

A

請求項の数 19 (全 25 頁)

(21) 出願番号 特願2009-224832 (P2009-224832)
 (22) 出願日 平成21年9月29日 (2009. 9. 29)
 (65) 公開番号 特開2011-76226 (P2011-76226A)
 (43) 公開日 平成23年4月14日 (2011. 4. 14)
 審査請求日 平成24年9月27日 (2012. 9. 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 100126240
 弁理士 阿部 琢磨
 (74) 代理人 100124442
 弁理士 黒岩 創吾
 (72) 発明者 中田 浩暁
 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 田中 友章

最終頁に続く

(54) 【発明の名称】 情報処理装置及びその方法

(57) 【特許請求の範囲】

【請求項 1】

ユーザからの印刷指示を受け付け、印刷ジョブを接続されたプリンタに送信する情報処理装置であって、

ユーザからの印刷指示に応じてアプリケーションから受け取ったデータを中間データとしてスプールするスプール手段と、

前記ユーザからの印刷指示に応じた前記スプール手段によるスプール処理の後に、自装置に接続されているプリンタから取得した識別情報を用いて、当該プリンタに対応するプリンタドライバのインストール処理を制御するインストール制御手段と、

前記インストール制御手段によりインストールされたプリンタドライバに対して、前記スプール手段によりスプールした中間データを渡すデスプール手段と、

前記デスプール手段により渡された中間データから、前記インストール制御手段によりインストールされたプリンタドライバを用いて生成された印刷ジョブを、当該プリンタドライバに対応する前記プリンタに対して送信する送信手段と、を備えることを特徴とする情報処理装置。

【請求項 2】

検索により見つかった自装置に接続されているプリンタの一覧を表示するための表示制御手段を更に有し、

前記インストール制御手段は、前記表示制御手段により表示されたプリンタの一覧から選択されたプリンタから取得した識別情報を用いて、当該プリンタに対応するプリンタド

10

20

ライバのインストール処理を制御することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

前記インストール制御手段は、前記プリンタに対応するプリンタドライバを検索して、見つかったプリンタドライバのインストール処理を制御し、

検索する対象として、情報処理装置の備える記憶領域、情報処理装置と接続された外部装置の記憶領域、情報処理装置と接続されたプリンタの記憶領域の少なくとも何れかを含むことを特徴とする請求項 1 または 2 に記載の情報処理装置。

【請求項 4】

前記インストール制御手段は、前記識別情報として前記プリンタから取得したハードウェア情報に基づき、当該プリンタに対応するプリンタドライバを検索して、見つかったプリンタドライバのインストール処理を制御することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の情報処理装置。

10

【請求項 5】

前記インストール制御手段は、前記プリンタに対応するプリンタドライバが複数ある場合、ユーザに対して何れか 1 つを選択させ、当該選択されたプリンタドライバのインストール処理の制御を行うことを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の情報処理装置。

【請求項 6】

前記インストール制御手段は、前記プリンタに対応するプリンタドライバが複数ある場合、プリンタドライバの属性に基づいてプリンタドライバを選択し、当該選択されたプリンタドライバのインストール処理の制御を行うことを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の情報処理装置。

20

【請求項 7】

前記プリンタドライバの属性は、バージョン及びコマンドタイプ情報の少なくとも何れかを含むことを特徴とする請求項 6 に記載の情報処理装置。

【請求項 8】

前記プリンタドライバには、複数のモデルのプリンタが共通にもつ機能を利用した印刷を可能とする複数のモデルのプリンタに対応できる汎用プリンタドライバ、及び単一のモデルのプリンタがもつ機能を全て利用できる当該単一のモデルのプリンタに対応する専用プリンタドライバが含まれることを特徴とする請求項 1 乃至 7 の何れか 1 項に記載の情報処理装置。

30

【請求項 9】

汎用プリンタドライバを優先するモードか否かを判断する判断手段を備え、

前記判断手段により汎用プリンタドライバを優先するモードであると判断された場合、前記インストール制御手段の制御に従う前記プリンタに対応する専用プリンタドライバをインストール処理が行われることなく、前記送信手段が、前記デスプूल手段により渡された中間データから前記汎用プリンタドライバを用いて生成された印刷ジョブを、前記プリンタに対して送信することを特徴とする請求項 8 に記載の情報処理装置。

【請求項 10】

ユーザからの印刷指示を受け付け、印刷ジョブを接続されたプリンタに送信する情報処理装置における印刷制御方法であって、

40

ユーザからの印刷指示に応じてアプリケーションから受け取ったデータを中間データとしてスプूलするスプूल工程と、

前記ユーザからの印刷指示に応じたスプूल処理の後に、自装置に接続されているプリンタから取得した識別情報を用いて、当該プリンタに対応するプリンタドライバのインストール処理を制御するインストール制御工程と、

前記インストール制御工程でインストールされたプリンタドライバに対して、前記スプूल工程でスプूलした中間データを渡すデスプूल工程と、を有し、

前記デスプूल工程で渡された中間データから、前記インストール制御工程でインストールされたプリンタドライバを用いて生成された印刷ジョブが、当該プリンタドライバに

50

対応する前記プリンタに対して送信されることを特徴とする印刷制御方法。

【請求項 1 1】

検索により見つかった自装置に接続されているプリンタの一覧を表示するための表示制御工程を更に有し、

前記インストール制御工程では、前記表示制御工程で表示されたプリンタの一覧から選択されたプリンタから取得した識別情報を用いて、当該プリンタに対応するプリンタドライバのインストール処理が制御されることを特徴とする請求項 1 0 に記載の印刷制御方法。

【請求項 1 2】

前記インストール制御工程では、前記プリンタに対応するプリンタドライバを検索して、見つかったプリンタドライバのインストール処理が制御され、

検索する対象として、情報処理装置の備える記憶領域、情報処理装置と接続された外部装置の記憶領域、情報処理装置と接続されたプリンタの記憶領域の少なくとも何れかを含むことを特徴とする請求項 1 0 または 1 1 に記載の印刷制御方法。

【請求項 1 3】

前記インストール制御工程では、前記識別情報として前記プリンタから取得したハードウェア情報に基づき、当該プリンタに対応するプリンタドライバを検索して、見つかったプリンタドライバのインストール処理が制御されることを特徴とする請求項 1 0 乃至 1 2 のいずれか 1 項に記載の印刷制御方法。

【請求項 1 4】

前記インストール制御工程では、前記プリンタに対応するプリンタドライバが複数ある場合、ユーザに対して何れか 1 つを選択させ、当該選択されたプリンタドライバのインストール処理が制御されることを特徴とする請求項 1 0 乃至 1 2 の何れか 1 項に記載の印刷制御方法。

【請求項 1 5】

前記インストール制御工程では、前記プリンタに対応するプリンタドライバが複数ある場合、プリンタドライバの属性に基づいてプリンタドライバを選択し、当該選択されたプリンタドライバのインストール処理が制御されることを特徴とする請求項 1 0 乃至 1 2 の何れか 1 項に記載の印刷制御方法。

【請求項 1 6】

前記プリンタドライバの属性は、バージョン及びコマンドタイプ情報の少なくとも何れかを含むことを特徴とする請求項 1 5 に記載の印刷制御方法。

【請求項 1 7】

前記プリンタドライバには、複数のモデルのプリンタが共通にもつ機能を利用した印刷を可能とする複数のモデルのプリンタに対応できる汎用プリンタドライバ、及び単一のモデルのプリンタがもつ機能を全て利用できる当該単一のモデルのプリンタに対応する専用プリンタドライバが含まれることを特徴とする請求項 1 0 乃至 1 6 の何れか 1 項に記載の印刷制御方法。

【請求項 1 8】

汎用プリンタドライバを優先するモードか否かを判断する判断工程をさらに有し、前記判断工程で汎用プリンタドライバを優先するモードであると判断された場合、前記インストール制御工程での制御に従う前記プリンタに対応する専用プリンタドライバをインストール処理が行われることなく、前記デスプूल手段により渡された中間データから前記汎用プリンタドライバを用いて生成された印刷ジョブが、前記プリンタに対して送信されることを特徴とする請求項 1 7 に記載の印刷制御方法。

【請求項 1 9】

請求項 1 0 乃至 1 8 の何れか 1 項に記載の工程をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【 0 0 0 1 】

本発明は情報処理装置に関し、詳しくはユーザが印刷時に設置されたプリンタの中から利用するものを特定するだけで、効率よく印刷できるシステムに関するものである。

【 背景技術 】

【 0 0 0 2 】

パーソナルコンピュータなどの情報処理装置がプリンタなどの周辺装置を利用するには、一般的にその周辺装置に対応した特定のデバイスドライバを組み込む必要がある。ここでは、周辺装置の例としては、プリンタやスキャナなどといったものが挙げられる。したがって、利用する周辺装置が複数台ある場合はその台数分だけデバイスドライバを組み込まなければならず、ユーザはその度に周辺装置のIPアドレスを調べたり、対応するデバイスドライバを探して来たり、といった面倒なインストール作業を強いられてきた。

10

【 0 0 0 3 】

そこで、近年、周辺装置の種類によらず利用可能な共通デバイスドライバである「仮想ドライバ」が研究・開発されている。例えば、非特許文献1には、アプリケーションから仮想プリンタドライバに対して印刷指示を出した後に、出力可能なネットワークプリンタを選んで印刷する方法が開示されている。この仮想プリンタドライバのみを情報処理装置に組み込んでおくだけで、その仮想プリンタドライバが対応する複数のプリンタを利用することが可能である。

【 先行技術文献 】

【 非特許文献 】

20

【 0 0 0 4 】

【 非特許文献 1 】 HP Universal Print Driver Series for Windows - overview and features (<http://h20338.www2.hp.com/hpsub/cache/342988-0-0-225-121.html>)

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 5 】

しかしながら、非特許文献1に示された仮想プリンタドライバは、対応する複数のプリンタが共通で処理可能な形式のコマンドを生成し送信している。そのため、各プリンタの機能に特化した専用のプリンタドライバを用いて印刷する場合に比べて、ユーザが選択できる機能が制限されてしまう。また、仮想プリンタドライバが生成・送信するコマンドを解釈、処理できないプリンタに関しては、結局、印刷等の処理ができない。

30

【 0 0 0 6 】

そこで、本発明は、従来の仮想デバイスドライバが持つ課題を克服し、ユーザが印刷時に設置されたプリンタの中から利用するものを特定するだけで、とくに機能制限なく任意のプリンタの機能を十分に利用することができる手法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 0 7 】

上記課題を解決するために、本発明の情報処理装置は、ユーザからの印刷指示を受け付け、印刷ジョブを接続されたプリンタに送信する情報処理装置であって、ユーザからの印刷指示に応じてアプリケーションから受け取ったデータを中間データとしてスプールするスプール手段と、前記ユーザからの印刷指示に応じた前記スプール手段によるスプール処理の後に、自装置に接続されているプリンタから取得した識別情報を用いて、当該プリンタに対応するプリンタドライバのインストール処理を制御するインストール制御手段と、前記インストール制御手段によりインストールされたプリンタドライバに対して、前記スプール手段によりスプールした中間データを渡すデスプール手段と、前記デスプール手段により渡された中間データから、前記インストール制御手段によりインストールされたプリンタドライバを用いて生成された印刷ジョブを、当該プリンタドライバに対応する前記プリンタに対して送信する送信手段と、を備えることを特徴とする。

40

50

【発明の効果】

【0008】

本発明により、従来の仮想デバイスドライバが持つ課題を克服し、ユーザが印刷時に設置されたプリンタの中から利用するものを特定するだけで、とくに機能制限なく任意のプリンタの機能を十分に利用することができる。

【図面の簡単な説明】

【0009】

【図1】システムの構成図およびホストコンピュータならびに出力デバイス内部の装置構成を示すブロック図

【図2】プリンタドライバインストーラのユーザインターフェース例を表す図

【図3】プリンタドライバ、プリントキュー、ポートの各情報とそれらの関連を表す図

【図4】一般的なプリンタドライバのインストール処理を示すフローチャート

【図5】本実施例における仮想・汎用・専用プリンタドライバのインストール処理を示すフローチャート

【図6】印刷処理のデータフロー図

【図7】印刷処理のシーケンス図

【図8】アプリケーションおよびプリントジョブマネージャのユーザインターフェース例を表す図

【図9】ユーザインターフェースの遷移図

【図10】出力準備処理のフローチャート

【図11】汎用・専用プリンタドライバインストール処理のフローチャート

【図12】プリントキュー整理処理のフローチャート

【図13】プリントキュー削除処理のフローチャート

【図14】出力デバイス特定ダイアログへ戻る場合のシーケンス図

【発明を実施するための形態】

【0010】

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。

【0011】

図1(a)は、本発明の実施形態に係るネットワークシステムの概略構成を示すブロック図である。このネットワークシステムは、ホストコンピュータ(101)と、このホストコンピュータ(101)よりジョブを受信して処理する周辺装置、及び周辺装置に対応するデバイスドライバの配信などを行う配信サーバ(103)を備える。これらはネットワーク(104)を介して接続されている。なお、本実施形態では、周辺装置として印刷データを受信して印刷を行う出力デバイス(プリンタ)を挙げて説明する。

【0012】

図1(b)は、図1(a)のホストコンピュータ(101)と出力デバイス(102)の装置構成を示すブロック図である。

ホストコンピュータ(101)において、CPU(111)は、RAM(112)に記憶されているプログラムに従って、システムバス(114)に接続される各デバイスを総括的に制御している。このRAM(112)は、CPU(111)の主メモリ、ワークエリア等としても機能している。ROM(113)は、各種プログラム及びデータを格納しており、各種フォントを記憶するフォントROM(113a)、ブートプログラムやBIOS等を記憶するプログラムROM(113b)、及び各種データを記憶するデータROM(113c)に区分けして構成されている。キーボードコントローラI/F(115)は、キーボード(119)や不図示のポインティングデバイス(マウス)からのキー入力を制御する。ディスプレイI/F(116)は、ディスプレイ(120)への表示を制御している。外部メモリI/F(117)は、例えばハードディスク(HD)、Solid State Disk(SSD)等の外部メモリ(121)とのアクセスを制御する。外部メモリ(121)は、オペレーティングシステムプログラム(以下、OS)(122)をはじめ各種アプリケーション(123)、印刷処理関連プログラム(124)、さらに

10

20

30

40

50

、不図示のユーザファイル、編集ファイル等を記憶するコンピュータが読み取り可能な記憶媒体として機能する。なお、ここではOS(122)としてMicrosoft Windows(登録商標)を使用するものとする。ネットワークI/F(118)は、ネットワーク(104)を介して出力デバイス(102)に接続され、出力デバイス(102)との間で通信制御処理を行う。なお、配信サーバ(103)もホストコンピュータ(101)と同様の装置構成を備える。

【0013】

ここで、印刷処理関連プログラム(124)は、本実施形態に係るプリントジョブマネージャ(125)、仮想プリンタドライバ(126)及びドライバインストーラ(129)などを含むプログラム領域となる。これらプログラムを任意の手段でダウンロードし、後述するようなインストール処理を行うことで、このプログラムに基づく印刷処理が可能となる。また、印刷処理関連プログラム(124)においては、情報処理装置にインストールされる汎用プリンタドライバ(127)や専用プリンタドライバ(128)も含まれることになる。仮想プリンタドライバとは、本実施形態特有のドキュメントを物理的な出力デバイスに対して直接ではなく、間接的に印刷するために利用されるプログラムである。間接的なプログラムであるため「仮想」と呼称しているが、仮想プリンタドライバ(126)はOS(122)上では一般的なプリンタドライバとして登録されるため、ユーザは通常の印刷作業手順に近い感覚で作業を行うことができる。一方、汎用プリンタドライバと専用プリンタドライバは、一般的なプリンタドライバとして分類される。具体的には、汎用プリンタドライバとは、複数のモデルに印刷可能なプリンタドライバである。汎用プリンタドライバは、対応するモデルが共通で持つ機能を利用することができる。一方、専用プリンタドライバとは、単一のモデルにのみ対応したプリンタドライバである。専用プリンタドライバは、対応するモデルの機能を最大限に引き出すことが可能である。

【0014】

次に出力デバイス(102)の構成について説明する。CPU(132)は、出力デバイス(102)の全体の動作を制御している。RAM(139)は、CPU(132)の主メモリ、ワークエリア等として機能するとともに、出力情報展開領域、環境データ格納領域としても用いられる。またこのRAM(139)は、NVRAM(Non-volatile RAM:不揮発性RAM)領域も備えており、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。ROM(133)は、各種フォントを記憶するフォントROM(133a)、CPU(132)により実行される制御プログラム等を記憶するプログラムROM(133b)、及び各種データを記憶するデータROM(133c)を備えている。ネットワークI/F(138)は、ホストコンピュータ(101)との間で印刷ジョブやステータスなどのデータの送受信を行う。印刷部I/F(136)は、プリンタエンジンである印刷部(137)とのインターフェースを制御している。外部メモリ(134)は、外部メモリI/F(140)によりアクセスが制御されており、オプションとして接続されるハードディスク(HD)、Solid State Disk(SSD)等を含んでおり、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。さらに、本実施形態に係る専用プリンタドライバ(128)を記憶することも可能である。なお、ハードディスク等の外部メモリ(134)が接続されていない場合には、ROM(133)のデータROM(133c)に、ホストコンピュータ(101)で利用される情報等を記憶することになる。なお、この外部メモリ(134)は1個に限らず複数備えるものであってもよく、例えば、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等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。操作部(142)には、ユーザによる操作を受け付ける操作パネルが設けられ、その操作パネルには操作のためのスイッチ及びLED表示器等が配されている(不図示)。また、不図示のNVRAMを有し、操作パネル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。CPU(132)は、ROM(133)のプログラムROM(133b)に記憶された制御プログラム等に基づき、印刷部I/F(136)を介して印

刷部(137)に出力情報としての画像信号を出力する。また、CPU(132)はネットワークI/F(138)を介してホストコンピュータ(101)との通信処理が可能となっており、ホストコンピュータ(101)から送信される印刷データを受信すると共に、出力デバイス(102)内の情報等をホストコンピュータ(101)に通知可能に構成されている。

【0015】

図2は、図1(b)のドライバインストーラ(129)が備えるユーザインターフェース(200)である。ドライバインストーラ(129)を実行すると、ディスプレイ(120)上に表示される。プリンター一覧(201)にインストール可能なプリンタドライバの一覧が表示される。

10

Device Model Aは、モデルAの出力デバイスに対応した専用プリンタドライバである。

Device Model Bは、モデルBの出力デバイスに対応した専用プリンタドライバである。

Device Model Cは、モデルCの出力デバイスに対応した専用プリンタドライバである。

Device Model Dは、モデルDの出力デバイスに対応した専用プリンタドライバである。

Generic Device Model 1は、モデルA、Bに対応した汎用プリンタドライバである。

20

Generic Device Model 2は、モデルC、Dに対応した汎用プリンタドライバである。

Virtual Device Modelは、モデルA、B、C、Dだけではなく、あらゆる出力デバイスに対応可能な仮想プリンタドライバである。

【0016】

ユーザが任意のモデルを選択し、追加ボタン(202)を押下し、インストールするプリンター一覧(203)に選択したモデルを加える。その後、ユーザがインストールボタン(204)を押下することで、ドライバインストーラ(129)は指定されたモデルのプリンタドライバをインストールする。

【0017】

30

図3は、プリンタドライバとプリントキューとポートの各情報とそれらの関係を説明した図である。UML(Unified Modeling Language)のオブジェクト図を使って表現している。

プリントキュー(302)は、アプリケーションから印刷する際の印刷対象オブジェクトである。同一モデルのプリンタドライバ(301)から複数作成することができる。例えば、オフィスにDevice Model Aの出力デバイスが3台導入された場合、Device Model Aに対応したプリンタドライバを1つインストールして、そのプリンタドライバを使ったプリントキューを3つ作成することになる。ポート(303)は、ネットワークの出力先を特定するオブジェクトである。プリントキューとポートの関係は通常は一對一であるが、図のように同じポートを複数のプリントキューが使用してもかまわない。

40

【0018】

プリンタドライバ(301)はプリンタドライバ構成情報(304)を保持している。プリンタドライバ構成情報(304)は、ドライババージョン、モデル名、ハードウェア情報、ドライバモジュール名、等から構成される。このプリンタドライバ構成情報(304)は、OS(122)によって管理される。

【0019】

プリントキュー(302)は、プリントキュー構成情報(305)を保持している。プリントキュー構成情報(305)は、プリントキュー名、ポート名、印刷設定情報、ジョブ投入時刻、ジョブステータス、等から構成される。このプリントキュー構成情報(30

50

5)は、OS(122)によって管理される。さらに、プリントキュー(302)は、本実施形態特有の自動生成識別子(306)と更新日時(307)をも保持する。これらの情報は、プリントジョブマネージャー(125)によって管理される。詳細は後述する。

【0020】

ポート(303)は、ポート構成情報(308)を保持している。ポート構成情報(308)は、ポート名、モジュール名、IPアドレス、等から構成される。このポート構成情報(308)は、OS(122)によって管理される。

【0021】

図4は、図1(b)のドライバインストーラ(129)による一般的なプリンタドライバのインストール処理を示すフローチャートである。一般的なプリンタドライバとは、専用プリンタドライバおよび汎用プリンタドライバを指す。

10

【0022】

プリンタドライバのインストールを開始(S400)すると、まずユーザによるモデルの選択を受け付ける(S401)。ここでは、図2のプリンター一覧(201)において、Virtual Device Model以外のモデルを選択された場合を想定している。本実施形態に特有のVirtual Device Modelの選択も含んだインストール処理については図5にて後述する。専用プリンタドライバもしくは汎用プリンタドライバの選択を受け付けると、選択されたモデルのプリンタドライバをインストールする(S402)。

S402でのプリンタドライバのインストールは、プリンタドライバインストーラが、OSのインストールAPIであるスプーラAPIに対して設定ファイル(INFファイル)のパスを渡すことにより行われる。OSの一部であるスプーラAPIは、設定ファイルに従って、プリンタドライバを構成するDLL(Dynamic Link Library)をCD-ROMやインストールセットの中から探索する。そして、スプーラAPIは、所定のOSのシステムディレクトリに対して、探索されたDLLをコピーする。スプーラAPIは、設定ファイルに従って、レジストリ内にドライバ名などを記録する。なお、スプーラAPIは、プリンタフォルダの表示や、アプリケーションからのプリンタ選択UI(例えば、図8(a)等)の表示指示があった際、OSのレジストリに登録されるプリントキューのデータ構造を走査し、プリンタドライバに対応するプリントキューが設定されていれば、それをディスプレイ表示する。なお、ここではプリントキューがまだ作成及び設定されていないため、OSのユーザインターフェース(例えばプリンタ選択UI)には、現在インストールしようとするプリンタドライバのプリンタアイコンはまだ表示されない。

20

30

【0023】

次に、ユーザによるIPアドレス等のポート情報の入力を受け付ける(S403)。そして、ポートを作成する(S404)。次に、ユーザによるプリントキュー名の入力を受け付ける(S405)。そして、S402でインストールしたプリンタドライバとS404で作成したポートを関連付けて、プリントキューを作成する(S406)。以上で、専用プリンタドライバ及び汎用プリンタドライバのインストール処理が終了する(S407)。

40

【0024】

図5は、図1(b)のドライバインストーラ(129)による本実施形態特有のプリンタドライバのインストール処理を示すフローチャートである。

【0025】

プリンタドライバのインストールを開始(S500)すると、まずユーザによるモデルの選択を受け付ける(S501)。そして、選択されたモデルが仮想プリンタドライバかどうか判断する(S502)。選択されたモデルが仮想プリンタドライバではない場合、専用・汎用プリンタドライバのインストール処理を行う(S503)。S503では、図4のS402以降の処理と同様の処理を行うことになる。そして、当インストール処理を終了する(S516)。

50

【 0 0 2 6 】

S 5 0 2 にて選択されたモデルが仮想プリンタドライバの場合、仮想プリンタドライバをインストールする (S 5 0 4)。仮想プリンタドライバ自体は、ポートに対する印刷データの出力を行わないので、ユーザによる入力を伴わずにダミーのポートを作成する (S 5 0 5)。ここで、ダミーのポートの作成とは、例えば、次のような処理になる。存在しないファイル名など適当な名前を指定してポートを O S に対して作成指示する。そうすると、当該作成指示は、エラーとして O S に認識され、作成指示されたポート名を捨てる。よって、実際にはポートは作成されない。仮想プリンタドライバに対し複数のプリントキューを作る必要性が無いので、ユーザによる入力を伴わずにモデル名と同じ名前でプリントキューを作成する (S 5 0 6)。図 4 で一般的なプリンタドライバのインストール処理について説明した際、ユーザによる入力は、S 4 0 1、S 4 0 3、及び S 4 0 5 の 3 回あった。一方、本実施形態特有のプリンタドライバのインストール処理においては、ユーザによる入力は S 5 0 1 だけである。もし、ドライバインストーラ (1 2 9) が仮想プリンタドライバだけを対象としたプログラムであれば、S 5 0 1 も不要となる。つまり、仮想プリンタドライバのインストール処理方法は、ユーザの負担と運用コストを低減させる効果がある。

10

【 0 0 2 7 】

仮想プリンタドライバは、ポートに対する印刷データの出力を行わない。そこで、印刷を最低限の保証するために汎用プリンタドライバを設置する。引き続き、汎用プリンタドライバのインストール処理について説明する。

20

【 0 0 2 8 】

インストールセットに含まれる全ての汎用プリンタドライバを列挙する (S 5 0 7 ~ S 5 1 0)。図 2 のプリンター一覧 (2 0 1) において、汎用プリンタは、Generic Device Model 1 と Generic Device Model 2 が存在する。よって、1 回目のループ処理では、Generic Device Model 1 を取り出し、2 回目のループ処理では、Generic Device Model 2 を取り出す。取り出した汎用プリンタドライバがホストコンピュータ (1 0 1) にインストール済みかどうか判断する (S 5 0 8)。まだ、インストールしていない場合は、取り出した汎用プリンタドライバをインストールする (S 5 0 9)。このとき、汎用プリンタドライバを使ったプリントキューを作成しないので、ユーザが汎用プリンタドライバのインストールを意識することはない。そして、S 5 1 0 へ進む。ループ処理が終了したら、当インストール処理を終了する (S 5 1 6)。

30

【 0 0 2 9 】

S 5 0 8 にて、取り出した汎用プリンタドライバがホストコンピュータ (1 0 1) にインストール済みの場合は、インストール済みの汎用プリンタドライバのバージョンを取得する (S 5 1 1)。バージョン数から継続使用可能かどうか判断する (S 5 1 2)。継続使用が可能と判断した場合、S 5 1 0 へ進む。ループ処理が終了したら、当インストール処理を終了する (S 5 1 6)。

【 0 0 3 0 】

S 5 1 2 にて、インストール済みの汎用プリンタドライバの継続使用が不可能と判断した場合、インストール済みの汎用プリンタドライバをアンインストールする (S 5 1 3)。アンインストールとは、O S が管理する所定のシステムディレクトリからの汎用プリンタドライバの DLL の削除、及び、汎用プリンタドライバに対応する O S のレジストリのクリアなどの処理が挙げられる。この際、プリントキューやポートの情報は事前にバックアップしておく。そして、取り出した汎用プリンタドライバをインストールする (S 5 1 4)。さらに、S 5 1 3 の処理を行う前にインストール済み汎用プリンタドライバを使って構成されていたプリントキューを、S 5 1 4 でインストールした汎用プリンタドライバを使って復元する (S 5 1 5)。ここでの復元処理とは、新しくインストールする汎用プリンタドライバの設定として、それまでのプリントキューの名称やポートの指定などを反映させる処理をさす。復元すべき情報を表示し、ユーザにドライバインストーラや O S な

40

50

どのユーザインターフェースにおいて選ばせることも可能である。その後、S 5 1 0へ進む。ループ処理が終了したら、当インストール処理を終了する(S 5 1 6)。

【0031】

以上、ドライバインストーラ(129)が、仮想プリンタドライバをインストールしてそのプリントキューを作成すると、自動的に汎用プリンタドライバをインストールすることを説明した。なお、S 5 0 7以降の汎用プリンタドライバのインストール処理については、図1(b)のドライバインストーラ(129)が実施することで説明をしたが、仮想プリンタドライバ(126)の初期化処理において実施してもかまわない。また、一般にVendor Setupと呼ばれるOS(122)から呼び出される不図示のプリンタドライバのセットアップモジュールで実施してもかまわない。

10

【0032】

図6は本発明における印刷制御の処理に関するデータフロー図である。本図を用いて、印刷処理関連プログラムの各コンポーネント間でどのような入出力が行われているか明らかにする。

【0033】

アプリケーション(123)は仮想プリンタドライバ(126)に対して、OS(122)の提供する描画インターフェースであるGDI(Graphic Device Interface)(600)を通じて印刷指示命令を出力する。次に仮想プリンタドライバ(126)は、入力された印刷指示命令をEMF(Enhanced Meta File)(601)の形式に変換してプリントジョブマネージャー(125)に出力する。ここでEMF(601)とは、GDI(600)の印刷指示命令と互換性のある中間データ形式であり、アプリケーション(123)から出力された一連の印刷指示命令を再現可能なデータとして記録したものである。本実施例では中間データ形式としてEMF(601)を用いて説明しているが、印刷指示命令を再現可能なデータであれば、例えばXPS(XML Paper Specification)、PDF(Portable Document Format)、あるいは独自に定義した任意の形式であっても良い。

20

【0034】

次にプリントジョブマネージャー(125)は、後述する出力デバイスの特定、汎用・専用プリンタドライバの管理など固有の処理を行い、GDI(600)を通じて印刷指示命令を出力する。出力された印刷指示命令は、汎用プリンタドライバ(127)あるいは専用プリンタドライバ(128)に対して入力される。次に、汎用プリンタドライバ(127)あるいは専用プリンタドライバ(128)は、出力デバイス(102)が処理可能なPDL(Printer Description Language)(602)の形式に変換し、出力デバイス(102)にジョブとして出力する。最後に出力デバイス(102)は、入力されたPDL(602)を処理して印刷出力を行う。

30

【0035】

図7は印刷処理のシーケンス図である。本図を用いて、一連の印刷処理中に関連プログラムの各コンポーネントの内部、およびそれらの間でどのような処理が行われているか明らかにする。

40

【0036】

まず、ユーザはアプリケーション(123)に対して印刷したいドキュメントの印刷指示を図8(a)で示すユーザインターフェース(8000)を通じて入力する(S700)。8001~8003はプリントキューを表すアイコンである。このうち、8001は仮想プリンタドライバのプリントキューである。8002、及び8003は専用プリンタドライバのプリントキューである。ここでは仮想プリンタドライバを通じて出力を行うため、ユーザは8001のアイコンを選択し、印刷ボタン(8004)を押して、印刷を指示する(S700)。

【0037】

次に、アプリケーション(123)は、仮想プリンタドライバ(126)に対して、印

50

刷初期化処理を行う(S 7 0 1)。仮想プリンタドライバ(1 2 6)は印刷初期化が行われるとプリントジョブマネージャー(1 2 5)の起動処理を行う(S 7 0 2)。プリントジョブマネージャー(1 2 5)は、このように仮想プリンタドライバ(1 2 6)に対するドキュメントの印刷開始後に起動されるプログラムである。後述する出力デバイスの特定の表示制御、汎用・専用プリンタドライバのインストール制御、または管理などといった固有の処理を担う。また、プリントジョブマネージャー(1 2 5)は内部的なコンポーネントとして、出力デバイス特定部(1 2 5 1)、ドライバ管理部(1 2 5 2)、及びデスプूल処理部(1 2 5 3)を含む。

【 0 0 3 8 】

次に、アプリケーション(1 2 3)はプリントジョブマネージャー(1 2 5)が起動されると、仮想プリンタドライバ(1 2 6)に対して印刷を開始し、一連の印刷描画命令を G D I (2 1 9) を通じて出力する(S 7 0 3)。仮想プリンタドライバ(1 2 6)は、入力された一連の印刷描画命令を E M F (6 0 1) 形式に変換し、あとで印刷描画命令を再現できるよう外部メモリ(1 2 1)に保存する。これをスプूल処理と呼ぶ(S 7 0 4)。アプリケーション(1 2 3)は一連の印刷描画命令の最後に印刷終了の処理を行い(S 7 0 5)、仮想プリンタドライバ(1 2 6)はそれを受けて保存された E M F (6 0 1) の外部メモリ(1 2 1)における保存位置情報をプリントジョブマネージャー(1 2 5)に通知する(S 7 0 6)。なお、S 7 0 4 は、外部メモリ(1 2 1)に対してスプूल処理を実施するが、外部メモリ(1 2 1)を介さずプリントジョブマネージャー(1 2 5)へ直接スプूल処理してもかまわない。

【 0 0 3 9 】

次に、出力デバイス特定部(1 2 5 1)は、出力デバイスの検索を行う(S 7 0 7)。ここで検索とは、ネットワーク(1 0 4)に接続された出力デバイス 1 0 2 a、1 0 2 b と通信し、それらの名称、設置場所、I P アドレス、ハードウェア I D といった情報を取得することである。通信手段としては公知の技術である S N M P (S i m p l e N e t w o r k M a n a g e m e n t P r o t o c o l) などのプロトコルが使用される。通常、出力デバイス 1 0 2 a、1 0 2 b の状態、すなわち電源が入っているかや、ネットワーク(1 0 4)に接続されているかを事前に把握することは不可能であるため、出力デバイス特定部(1 2 5 1)は印刷指示の都度、すべての出力デバイスとの通信を試みる。ネットワーク内のすべての装置との通信には、これも公知の技術であるブロードキャストという手段を用いる。出力デバイス特定部(1 2 5 1)がこのブロードキャストの方法で、ネットワーク I / F (1 1 8) を通じて S N M P の情報取得用データを送信すると、ネットワーク(1 0 4)で到達可能なすべての出力デバイス(1 0 2)が情報取得用データを受信する。そして、各出力デバイス(1 0 2)が、取得対象として指定された出力デバイスの名称や設置場所などの情報を応答のデータとしてネットワーク I / F (1 3 8) を通じて送信すると、出力デバイス特定部(1 2 5 1)は順次それらの応答を受信する。そして、受信した情報を R A M (1 1 2) に保存する。出力デバイス特定部(1 2 5 1)は、順次受信した応答からデバイス名や設置場所などの情報を取り出し、リストとして出力デバイスを特定する図 8 (b) のユーザインターフェース(8 1 0 0)に表示する(S 7 0 7)。8 1 0 0 には情報を表示するためのリストコントロール(8 1 0 1)が配置され、その中からひとつをカーソル(8 1 0 2)によって選択することができる。ボタン(8 1 0 3)を押すと、ネットワーク内で再検索が行われ、利用可能な出力デバイスのリストを最新の状態に更新できる。ボタン(8 1 0 4)を押すと、カーソル(8 1 0 2)で現在選択されているデバイスが出力先として決定される。出力デバイス特定部(1 2 5 1)は、その決定を受け(S 7 0 9)、次の処理に移る。チェックボックス(8 1 0 5)とボタン(8 1 0 6)はプリントキューを管理する際にユーザが操作する。これらの操作方法については後述する。

【 0 0 4 0 】

S 7 0 9 で出力先のデバイスが決定されると、出力デバイス特定部(1 2 5 1)はドライバ管理部(1 2 5 2)に対して出力準備処理を実行するよう依頼する(S 7 1 0)。S

10

20

30

40

50

7 1 1における出力準備処理の結果、OS (1 2 2)には選択された出力デバイス (1 0 2)に印刷可能な汎用プリンタドライバ (1 2 7)または不図示の専用プリンタドライバ (1 2 8)のプリントキューが登録される。出力準備処理 (S 7 1 1)については後で図 1 0、図 1 1を用いて詳述する。

【 0 0 4 1 】

次に出力デバイス特定部 (1 2 5 1)は、図 8 (b)の出力デバイス特定用の画面から、デスプール処理部 (1 2 5 3)が有する図 8 (c)の画面 (8 2 0 0)に遷移する (S 7 1 2)。デスプール処理部 (1 2 5 3)は、プレビューエリア (8 2 0 1)に印刷ドキュメントを実際に印刷したときのイメージを表示する (S 7 1 3)。このとき表示されるイメージは、S 7 0 4においてスプールされたEMF (6 0 1)をデスプール処理部 (1 2 5 3)が読み込み、解析して、プレビューエリア (8 2 0 1)の大きさに合わせて描画したものである。スピンコントロール (8 2 0 2)やチェックボックス (8 2 0 3)はそれぞれ印刷部数、白黒印刷といった一般的な印刷設定を変更するための入力手段である。さらに特殊な印刷設定を変更するためには、ボタン (8 2 0 4)を押下することにより、不図示のプリンタドライバ固有のユーザインターフェースを表示させ、印刷設定の変更指示を入力することができる。このとき、出力先に対応するプリントキューが専用プリンタドライバ (1 2 8)のものである場合、汎用プリンタドライバ (1 2 7)のものに比べて、より高度な印刷設定が可能である。ボタン (8 2 0 6)は、現在の画面 (8 2 0 0)から、ひとつ前の画面である出力デバイス特定画面 (8 1 0 0)に遷移して戻するための入力手段である。ボタン (8 2 0 5)をユーザが押下すると (S 7 1 4)、デスプール処理部 (1 2 5 3)は汎用プリンタドライバ (1 2 7)に対してデスプール処理を行い、一連の印刷描画命令を発行する (S 7 1 5)。ここで、デスプール処理というのはスプール処理の逆で、S 7 0 6によって通知された保存位置情報からEMF (6 0 1)データを読み出し、解析して、印刷描画命令を再現することである。これにより、プリントジョブマネージャー (1 2 5)固有の処理のために中断していた印刷処理が再開され、印刷ドキュメントの出力が行われる。汎用プリンタドライバ (1 2 7)は、入力された印刷描画命令をPDL (6 0 2)に変換し、ネットワークI/F (1 3 8)を通じて出力デバイス (1 0 2)に送信する (S 7 1 6)。印刷処理が終わると出力デバイス特定部 (1 2 5 1)は、ドライバ管理部 (1 2 5 2)へプリントキュー整理処理 (S 7 1 8)を実行するよう依頼する (S 7 1 7)。プリントキュー整理処理 (S 7 1 8)については後で図 1 2 ~ 図 1 4を用いて詳述する。最後に、プリントジョブマネージャー (1 2 5)はプログラムを終了する。

【 0 0 4 2 】

図 9 にプリントジョブマネージャー (1 2 5)のユーザインターフェースの遷移の概要を示す。

【 0 0 4 3 】

まず、プリントジョブマネージャー (1 2 5)は、「出力デバイス特定」ダイアログ (8 1 0 0)を表示する。出力デバイスの特定と「次へ」のボタン (8 1 0 4)の押下をうけて、プリントジョブマネージャー (1 2 5)は、「編集プレビュー」ダイアログ (8 2 0 0)を表示する。「編集プレビュー」ダイアログ (8 2 0 0)では、プリントジョブのプレビュー確認と基本的な印刷設定が可能である。「印刷」ボタン (8 2 0 5)の押下をうけてプリントジョブマネージャー (1 2 5)は出力デバイスへ印刷を実行し同ダイアログを閉じる。加えて、「編集プレビュー」ダイアログ (8 2 0 0)には、プリンタドライバが持っている特有の機能を設定するために「詳細設定」ボタン (8 2 0 4)が存在する。「詳細設定」ボタン (8 2 0 4)の押下をうけて、プリントジョブマネージャー (1 2 5)は、プリンタドライバの「印刷設定」ダイアログ (9 0 0 0)を表示する。更に、「編集プレビュー」ダイアログ (8 2 0 0)には「戻る」ボタン (8 2 0 6)が存在する。この「戻る」ボタン (8 2 0 6)は現在の出力先デバイスでは所望の機能が実現できないことが分かった際の出力デバイスの再設定のために存在する。「戻る」ボタン (8 2 0 6)の押下をうけて、プリントジョブマネージャー (1 2 5)は、「出力デバイス特定」ダ

イアログ (8 1 0 0) を表示し、再度出力デバイスの指定が可能になる。

【 0 0 4 4 】

図 1 0 は、ドライバ管理部 (1 2 5 2) における出力準備処理 (S 7 1 1) の詳細な流れを表すフローチャートである。本処理によって、出力デバイス (1 0 2) へ印刷するためのプリントキューがあればそれを選定し、なければ新しいプリントキューを作成、あるいは最適なプリンタドライバをインストールする。

【 0 0 4 5 】

まず、ドライバ管理部 (1 2 5 2) は、出力準備処理を開始すると (S 1 0 0 0)、出力デバイス (1 0 2) の I P アドレス情報を取得する (S 1 0 0 1)。I P アドレス情報は出力デバイスの検索処理 (S 7 0 7) において取得され R A M (1 1 2) に記憶されているものを取り出す。次に O S (1 2 2) に登録されているすべてのプリントキューからプリントキューに関連付けられているポートの I P アドレスを取得する (S 1 0 0 2 ~ S 1 0 0 4)。

【 0 0 4 6 】

次に、出力デバイス (1 0 2) の I P アドレスと、ポートの I P アドレスが合致するプリントキューが存在するかどうか判断する (S 1 0 0 5)。後述するが、本システムにおいて、プリントキューは出力デバイスごとに作成される。しかし、同じ出力デバイスに印刷を行うたびに別のプリントキューを毎回作成すると大量のプリントキューが重複して登録されてしまう。このような事態を避けるため、出力デバイスとすべてのプリントキューの I P アドレス同士を比較することによって、既存のプリントキューの中に出力可能なものがないかを判断する。もし出力可能なプリントキューが存在すれば S 1 0 0 6 に、存在しなければ S 1 0 1 1 に進む。

【 0 0 4 7 】

次に S 1 0 0 6 ではさらに汎用プリンタドライバを優先させるモードであるかどうか判断する。所望の出力デバイス (1 0 2) に出力可能なプリントキューの中には専用プリンタドライバ (1 2 8) のものが含まれている可能性もある。しかしながら、高度な機能は必要とせず、シンプルで安定した汎用プリンタドライバ (1 2 7) を使用したいユーザもいるため、例えば不図示のユーザインターフェースによって汎用プリンタドライバを優先させるかどうか切り替え可能なモードを設けることが考えられる。このモードが真すなわち汎用プリンタドライバを優先させる場合は、S 1 0 0 7 に進み、そうでなければ S 1 0 0 9 に進む。

【 0 0 4 8 】

次に S 1 0 0 7 ではプリントキューのモデル名を取得する。そして、デバイスもモデル名が汎用プリンタドライバのものであるかどうか判断する (S 1 0 0 8)。プリントキューの名前がユーザによって変更可能であるのに対し、汎用プリンタドライバのモデル名は「Generic Device Model 1」のように固有の名称なので、プリントキューのモデル名を知ればどの種類のドライバなのか判断できる。汎用プリンタドライバである場合は S 1 0 0 9 へ進み、そうでなければ S 1 0 1 0 へ進む。

S 1 0 0 9 へ進んだ場合、S 1 0 0 5 で見つかった既存のプリントキューを出力先として決定し、出力準備処理を終了する (S 1 0 1 4)。

S 1 0 1 0 へ進んだ場合、すなわち既存のプリントキューが専用プリンタドライバのものであると判断された場合は、図 1 1 で後述する処理において、汎用プリンタドライバを優先インストールすることに決定し、S 1 0 1 3 に進む。

S 1 0 1 1 では S 1 0 0 6 と同様に汎用プリンタドライバを優先させるモードかどうかを判断し、真なら S 1 0 1 0 に、偽なら S 1 0 1 2 へ進む。S 1 0 1 2 では、図 1 1 で後述する処理において、適合する最適なドライバを検索してインストールすることに決定し、S 1 0 1 3 に進む。ここで S 1 0 1 2 では、既存のプリントキューに出力可能なものがないとせず、汎用プリンタドライバを優先させるモードでもないのに、適合する最適なドライバ (専用プリンタドライバ) を優先インストールすることが決定される。

【 0 0 4 9 】

汎用・専用プリンタドライバインストール処理（S1013）については図11を用いて詳述する。

【0050】

図11は、ドライバ管理部（1252）における汎用・専用プリンタドライバインストール処理（S1013）の詳細な流れを表すフローチャートである。

【0051】

まず、ドライバ管理部（1252）は処理を開始すると（S1100）、出力用のポートを作成する（S1101）。次に、図10の専用プリンタドライバを優先インストールすることに決定されているか否かで判断し（S1102）、真であればS1103に、偽であればS1108に進む。ここで真となり、S1103に進む場合とは、図10でS1012のステップから遷移してS1013の処理に入った場合である。一方、偽となり、S1108に進む場合とは、図10でS1010のステップから遷移してS1013の処理に入った場合である。

S1103～S1107はホストコンピュータ（101）や配信サーバ（103）、あるいは出力デバイス102a、102bの備える記憶領域の中から出力に最適な専用プリンタドライバ（128）を検索するステップである。まず、S1103では出力デバイスのハードウェア情報を取得する。本実施例ではハードウェア情報として、プラグアンドプレイによるインストールに用いられるハードウェアIDを取得する。ハードウェアIDは出力デバイスの検索処理（S707）において取得されRAM（112）に記憶されているものを取り出す。

【0052】

次に、S1104ではインストール済みプリンタドライバ内に適合するプリンタドライバがあるかどうか判断する。具体的には、ドライバ管理部（1252）は、OS（122）に登録されている各プリンタドライバのハードウェア情報と、S1103で取得した出力デバイス（102）のハードウェア情報とを比較し、一致するものがあればすべてピックアップする。1つ以上適合するものがあればS1116へ、なければS1105へ進む。

【0053】

次に、S1105ではドライバストア内に適合するプリンタドライバがあるかどうか判断する。ドライバストアとはWindows Vista（登録商標）で導入されたデバイスドライバのデータベースシステムであり、インストール前のデバイスドライバのパッケージをOS（122）に格納する仕組みである。ドライバ管理部（1252）は、このドライバストアに格納されているプリンタドライバの中に、S1103で取得した出力デバイス（102）のハードウェア情報に一致するものがないかOS（122）に問い合わせることができる。問い合わせの結果、1つ以上適合するものがあればS1112へ、なければS1106へ進む。

【0054】

次に、S1106では配信サーバ（103）内に適合するプリンタドライバがあるかどうか判断する。ドライバ管理部（1252）は、ドライバ配信サーバ（103）にRPC（Remote Procedure Call）やHTTP（Hyper Text Transfer Protocol）などの手段で、ネットワーク（104）を経由して、あるハードウェア情報に適合するプリンタドライバがあるか問い合わせることができる。問い合わせの結果、1つ以上適合するものがあれば配信サーバ（103）からプリンタドライバのパッケージをホストコンピュータ（101）に転送して、外部メモリ（121）に記憶し、S1112へ進む。そうでなければS1107へ進む。

【0055】

次に、S1107では出力デバイス（102）内を検索し、適合するドライバがあるかどうか判断する。前述したとおり、出力デバイス（102）はその外部メモリ（134）に、出力デバイス自身を利用するために必要な専用プリンタドライバ（128）を記憶することができる。ドライバ管理部（1252）は、この出力デバイス（102）に記憶さ

れたプリンタドライバも、S 1 1 0 6と同様の手段でネットワーク(104)を経由して、問い合わせることが可能である。問い合わせの結果、1つ以上適合するものがあれば出力デバイス(102)から専用プリンタドライバ(128)のデータをホストコンピュータ(101)に転送して、外部メモリ(121)に記憶し、S 1 1 1 2へ進む。外部メモリ(134)の設定変更等により専用プリンタドライバ(128)が消去されているため、問い合わせの結果が1つも適合するものがないと判断された場合はS 1 1 0 8へ進む。

【0056】

S 1 1 0 8 ~ S 1 1 1 1では、前記S 1 1 0 3 ~ S 1 1 0 6の一連の処理の結果、適合するプリンタドライバがどこにも見つからない場合の代替手段として、あらかじめS 5 0 0 ~ S 5 1 6でインストールされた汎用プリンタドライバのプリントキューを作成する。まず、ドライバ管理部(1252)はネットワーク(104)を経由して、出力デバイス(102)のコマンドタイプ情報を取得し(S 1 1 0 8)、その判断を行う(S 1 1 0 9)。ここで、コマンドタイプ情報とは、出力デバイス(102)が処理可能なPDL(602)の種類を表す情報である。汎用プリンタドライバ(127)は複数のモデルに出力可能なPDLを発行するドライバであるが、あくまで1種類のPDLしか発行できない。したがって、異なるPDLをサポートするモデルに対しては出力できないため、コマンドタイプ情報を取得して、互換性の確認を行う。本実施例においては2種類のPDL(「1」または「2」)のどちらかに対応した出力デバイスを利用することを前提としている。しかしながら、対応できないコマンドタイプ情報も存在する。S 1 1 0 9では、コマンドタイプ情報「1」、「2」、それ以外であるかどうか判断している。「1」の場合はS 1 1 1 0へ、「2」の場合はS 1 1 1 1に進む。S 1 1 1 0ではコマンドタイプ「1」に対応した「Generic Device Model 1」のプリントキューを作成し、処理を終了する(S 1 1 1 8)。S 1 1 1 1ではコマンドタイプ「2」に対応した「Generic Device Model 2」のプリントキューを作成し、処理を終了する(S 1 1 1 8)。なお、プリントキューを作成する際はS 1 1 0 1で作成したポートを関連付ける。一方、S 1 1 0 9で、「1」、「2」以外のコマンドタイプ情報が取得されたと判断された場合は、画面を介して、ユーザに「他の出力デバイスを選択してください」などの警告を通知するとともに、本処理を終了(S 1 1 1 8)する。

【0057】

S 1 1 1 2 ~ S 1 1 1 4では検索の結果として適合した専用プリンタドライバ(128)をインストールする。まず、S 1 1 1 2では適合するプリンタドライバが2つ以上見つかったかどうか判断し、真であればS 1 1 1 3、偽であればS 1 1 1 4へ進む。S 1 1 1 3ではその中から最もプリンタドライバのバージョンの新しいものをインストールすべき対象として選択する。各プリンタドライバにはバージョンが付与されており、たとえ同じモデルのプリンタドライバであっても作成された時期が違えば、通常は異なるバージョンを有する。バージョンの表現方法としては例えば「10/10/2008.2.5.0.0」のように日付と数値の組み合わせで表現し、どちらの方が新しく作成されたものが容易に判断することができる。しかしながら、まれに古いバージョンのドライバをあえて使用する必然性が生じる場合がある。そのような場合は図8(d)の8300に示すようなユーザインターフェースを表示して、ユーザに選択させることが好ましい。8301には適合すると判断された複数のプリンタドライバの情報(名称、バージョン)がリストで表示される。ユーザは8301のリストの中から所望のプリンタドライバを選び、ボタン(8302)を押すことで決定できる。次にS 1 1 1 4では適合したプリンタドライバ、もしくは複数あった場合はS 1 1 1 3で選択されたプリンタドライバをインストールする。

【0058】

ここで、S 1 1 1 6 ~ S 1 1 1 7はS 1 1 1 2 ~ S 1 1 1 4の特殊なケースに関する処理である。具体的には、プリントキューが未作成のインストール済みプリンタドライバの中に、適合するプリンタドライバがあった場合の処理であり、S 1 1 1 6はS 1 1 1 2と、またS 1 1 1 7はS 1 1 1 7と同様の処理である。S 1 1 1 4が不要なのは、もはやプリンタドライバをインストールする必要がないからである。最後にS 1 1 1 5では、S

1 1 0 1において作成したポートに関連付けてプリントキューを作成し、処理を終了する (S 1 1 1 8)。

【 0 0 5 9 】

ここまで述べてきたように、プリントジョブマネージャー (1 2 5) は印刷時にプリントキューを O S (1 2 2) に登録する。次回以降の印刷で出力デバイスがわかっている場合は、仮想プリンタドライバ (1 2 6) を介さずに直接プリントキューを指定し、出力先デバイスの指定ステップがない通常手順で印刷できる。しかしながら、オフィス内に多数の出力デバイスが設置されている場合や、複数のオフィスを移動して印刷を行う場合、その都度プリントキューは作成されるので、プリントキューは増加傾向にある。

【 0 0 6 0 】

また、図 9 のユーザインターフェースの遷移で示したように、プリントジョブマネージャー (1 2 5) は、「出力デバイス特定」ダイアログ (8 1 0 0) と「編集プレビュー」ダイアログ (8 2 0 0) の遷移を何度でも繰り返すことが可能である。「出力デバイス」のリストボックス (8 1 0 1) で選択された出力デバイス用のプリントキューがホストコンピュータ (1 0 1) に存在しない場合プリントジョブマネージャー (1 2 5) は推移のたびに新たにプリントキューを作成する。そのため、「戻る」ボタン (8 2 0 6) の押下を繰り返すと、ホストコンピュータ (1 0 1) 上のプリントキューはさらに増加傾向になる。

【 0 0 6 1 】

ホストコンピュータ (1 0 1) 上のプリントキューの総数が増えると、印刷時に最初に表示される「印刷」ダイアログ (8 0 0 0) で表示されるプリントキューが多くなり、ユーザによる適切なプリンタの選択が難しくなる。そのため自動生成されたプリントキューの削除などを行える管理手段があれば、ユーザビリティの向上に繋がる。

【 0 0 6 2 】

図 8 (e) に「削除方法指定」のダイアログ (8 4 0 0) を示す。

【 0 0 6 3 】

「出力デバイス特定」のダイアログ (8 1 0 0) 上の「詳細設定」ボタン (8 1 0 6) の押下をうけて、プリントジョブマネージャー (1 2 5) はディスプレイ (1 2 0) へ表示する。同ダイアログ上の「OK」ボタン (8 4 0 7) の押下をうけて、プリントジョブマネージャー (1 2 5) は「削除方法指定」のダイアログ (8 4 0 0) を閉じる。その後、図 8 (b) に示す「出力デバイス特定」のダイアログ (8 1 0 0) が表示され、プリントジョブマネージャー (1 2 5) はそのダイアログ (8 1 0 0) へ処理を移行する。

【 0 0 6 4 】

「削除方法指定」のダイアログ (8 4 0 0) には、「削除方法を指定する」チェックボックス (8 4 0 1) が存在する。このチェックボックス (8 4 0 1) の初期状態はチェックなしである。このチェックボックス (8 4 0 1) がチェックされると 8 4 0 2 ~ 8 4 0 5 のアイテムが有効となり、プリントキューの削除方法の詳細指定が可能となる。

【 0 0 6 5 】

プリントキューの削除方法の詳細指定のために、「削除方法指定」のダイアログ (8 4 0 0) には、「選択したプリントキューを残す」ラジオボタン (8 4 0 2) と、「最新のプリントキューを残す」ラジオボタン (8 4 0 4) が存在する。この 2 つのラジオボタンへの指定は排他となっており、「選択したプリントキューを残す」ラジオボタン (8 4 0 2) が選択された場合は、「最新のプリントキューを残す」ラジオボタン (8 4 0 4) は非選択状態となる。またその逆に「最新のプリントキューを残す」ラジオボタン (8 4 0 4) が選択された場合は、「選択したプリントキューを残す」ラジオボタン (8 4 0 2) が非選択状態になる。デフォルト状態では「選択したプリントキューを残す」ラジオボタン (8 4 0 2) が選択状態となる。

【 0 0 6 6 】

「選択したプリントキューを残す」ラジオボタン (8 4 0 2) が選択状態の場合、リストボックス (8 4 0 3) の選択が有効になる。プリントジョブマネージャー (1 2 5) は

10

20

30

40

50

、ホストコンピュータ（１０１）上に存在するプリントキューの内、ドライバ管理部（１２５２）が作成したものをリストアップし、リストボックス（８４０３）へ表示する。但しドライバ管理部（１２５２）が作成したものではなく特殊なものであるため、このリストには仮想プリンタドライバ（１２６）のプリントキューであるVirtual Device Model（８００１）は含まれない。「選択したプリントキューを残す」ラジオボタン（８４０２）が選択状態で「ＯＫ」ボタン（８４０６）が押下された場合、プリントジョブマネージャー（１２５）は、リストボックス（８４０３）で選択されていないものを削除対象と認識する。

【００６７】

「最新のプリントキューを残す」ラジオボタン（８４０４）が選択状態の場合、個数のスピンドボックス（８４０５）での指定が有効になる。「最新のプリントキューを残す」ラジオボタン（８４０４）が選択状態で「ＯＫ」ボタン（８４０６）が押下された場合、プリントジョブマネージャー（１２５）は、個数のスピンドボックス（８４０５）での設定が有効であることを認識する。この設定で削除対象とするプリントキューは、「選択したプリントキューを残す」ラジオボタン（８４０２）で選択と同様に、ホストコンピュータ（１０１）上にドライバ管理部（１２５２）が作成したものに限定される。この設定では、指定された個数まで、利用された日時が新しいプリントキューを残し、それ以外のプリントキューを削除することになる。また仮想プリンタドライバ（１２６）のプリントキューであるVirtual Device Model（８００１）も同様に含まれない。

【００６８】

加えて、「削除方法指定」のダイアログ（８４００）には、「不要となったプリンタドライバを削除する」チェックボックス（８４０６）が存在する。デフォルト状態では、このチェックボックス（８４０６）は、チェックなしである。このチェックボックス（８４０１）がチェックされた状態で、「ＯＫ」ボタン（８４０６）が押下された場合、プリントジョブマネージャー（１２５）は、プリントキューと関連付けがなくなったプリンタドライバ自体の削除要求を受けたと判断する。「削除方法指定」のダイアログ（８４００）で設定された情報を元にプリントジョブマネージャー（１２５）は、ドライバ管理部（１２５２）へプリンタドライバ整理処理を依頼する。プリンタドライバ整理処理の詳細は図１２のフローチャートを用いて後述する。

【００６９】

図１２にドライバ管理部（１２５２）によるプリントキューの整理処理のフローチャートを示す。本実施例では、この処理は図７で示した印刷処理のシーケンスにおいて、印刷完了後のプリントキュー整理処理（Ｓ７１８）のタイミングでドライバ管理部（１２５２）によって実行される。

【００７０】

プリントキューの整理処理はＳ１２００より開始され、ドライバ管理部（１２５２）は、プリントキューのデータ構造（３０２）に対して、ドライバ管理部（１２５２）が自動生成したプリントキューであることを示す自動生成識別子（３０６）を追加する。更に加えてドライバ管理部（１２５２）は、プリントキューのデータ構造（３０２）の更新日時（３０７）に現在の日時を記す（Ｓ１２０１）。

【００７１】

次に、図７のデスプール（Ｓ７１５）のタイミングで、実際の印刷描画に用いたプリントキューが自動生成されたものであれば、プリントキューのデータ構造（３０２）の更新日時（３０７）を現在の日時に更新する（Ｓ１２０２）。ドライバ管理部（１２５２）は、一旦、図７の出力準備処理（Ｓ７１１）のタイミングで新規作成されたプリントキューのすべてのリストを取得する（Ｓ１２０３）。このリストをここでは削除候補プリントキューリストと呼称する。

【００７２】

ドライバ管理部（１２５２）は、「出力デバイス特定」ダイアログ（８１００）の「印刷したプリントキューを残す」のチェックボックス（８１０５）が選択状態かを確認する

10

20

30

40

50

(S 1 2 0 4)。もし非選択状態だった場合、S 1 2 0 6へ進む。もし選択状態だった場合、ドライバ管理部(1 2 5 2)は、実際の印刷描画に用いたプリントキューが削除候補プリントキューリストにあれば除外(S 1 2 0 5)し、その後、S 1 2 0 6へ進む。この処理により、図7のデスプール(S 7 1 5)のタイミングで実際の印刷描画に用いたプリントキューを、削除候補より外すことができる。

【0 0 7 3】

ドライバ管理部(1 2 5 2)は、「削除方法を指定する」チェックボックス(8 4 0 1)が選択状態であるか確認する。もし非選択状態だった場合、図8(e)の「削除方法指定」のダイアログ(8 4 0 0)で追加の詳細な削除方法が指定されていないので、S 1 2 1 3へ進む。その後、ドライバ管理部(1 2 5 2)は、削除候補プリントキューリスト中のプリントキューを削除し(S 1 2 1 3)、プリントキューの整理処理を終了する(S 1 2 1 4)。S 1 2 1 3の削除候補プリントキューリスト中のプリントキューの削除処理は、図13を用いて後で説明する。一方、「削除方法を指定する」チェックボックス(8 4 0 1)が選択状態の場合、「選択したプリントキューを残す」ラジオボタン(8 4 0 2)もしくは「最新のプリントキューを残す」ラジオボタン(8 4 0 4)の選択状態で処理フローを切り替える(S 1 2 0 7)。

10

【0 0 7 4】

「選択したプリントキューを残す」ラジオボタン(8 4 0 2)が選択状態である場合、S 1 2 0 8へ進む。S 1 2 0 8においてドライバ管理部(1 2 5 2)はリストボックス(8 4 0 3)上で選択されたプリントキューを削除候補プリントキューリストより除外する。その後、ドライバ管理部(1 2 5 2)は削除候補プリントキューリスト中のプリントキューを削除し(S 1 2 1 3)、プリントキューの整理処理を終了する(S 1 2 1 4)。

20

【0 0 7 5】

「最新のプリントキューを残す」ラジオボタン(8 4 0 4)が選択状態である場合、S 1 2 0 9へ進む。S 1 2 0 9においてドライバ管理部(1 2 5 2)は削除候補プリントキューリストを一旦クリアする。続けて、ドライバ管理部(1 2 5 2)はホストコンピュータ(1 0 1)上に存在するプリントキューのうち自動生成識別子(3 0 6)を有するプリントキューのリストを取得する(S 1 2 1 0)。ドライバ管理部(1 2 5 2)は取得したプリントキューのリスト中のプリントキューの個数を、「個数」スピンドボックス(8 4 0 5)で指定された数値と比較する(S 1 2 1 1)。もしプリントキューの個数が指定数を超えていない場合、S 1 2 1 4へ進み、プリントキューの整理処理を終了する。もしプリントキューの個数が指定数を超えている場合、取得したプリントキューのリストを更新日時(3 0 7)でソートし、古い日時のものから超過数分のプリントキューを削除候補プリントキューリストとする(S 1 2 1 2)。その後、ドライバ管理部(1 2 5 2)は削除候補プリントキューリスト中のプリントキューを削除し(S 1 2 1 3)、プリントキューの整理処理を終了する(S 1 2 1 4)。

30

【0 0 7 6】

図13にドライバ管理部(1 2 5 2)によるプリントキューの削除処理のフローチャートを示す。このフローチャートは図12のプリントキューの整理処理のフローチャートのS 1 2 1 3の詳細処理を示すものである。

40

【0 0 7 7】

本処理はS 1 3 0 0より開始され、ドライバ管理部(1 2 5 2)は、OS(1 2 2)提供の関数をコールすることにより、図3に示したプリンタドライバ、プリントキュー、ポートの各種情報を取得する。それらの情報を用いて、図12のプリントキューの整理処理のフローチャートで生成した削除候補プリントキューリストに含まれるすべてのプリントキュー(3 0 2)に関連付けられているプリンタドライバ(3 0 1)のリストを作成し保存する(S 1 3 0 1)。このリストは、削除候補プリンタドライバリストであり、S 1 3 0 9以降で、もし不要となったプリンタドライバがあれば削除されることになる。

【0 0 7 8】

次にドライバ管理部(1 2 5 2)は、削除候補プリントキューリストのすべてのプリン

50

トキューを処理したか判断する (S 1 3 0 2)。ドライバ管理部 (1 2 5 2) は、削除候補プリントキューリストに未処理のプリントキューがある場合は S 1 3 0 3 へ進み、未処理のプリントキューの何れかを取得し、現在の削除候補プリントキューとする。ドライバ管理部 (1 2 5 2) は、現在の削除候補プリントキューに印刷中もしくは待機中のジョブがあり、プリントキューを使用するか判断する (S 1 3 0 4)。もしプリントキューが使用中の場合は、ドライバ管理部 (1 2 5 2) は、ユーザへ削除してよいかの確認メッセージ (不図示) を表示する (S 1 3 0 5)。ここで、ドライバ管理部 (1 2 5 2) は、ユーザからの削除要求を受けた否かを判断する (S 1 3 0 6)。ユーザからの削除要求が「はい」の場合、現在の削除候補プリントキューの削除処理を行うために S 1 3 0 7 へ進む。ユーザからの削除要求が「いいえ」の場合は S 1 3 0 8 へ進み、現在の削除候補プリントキューの削除処理を行わず、S 1 3 0 2 に戻る。S 1 3 0 4 で、現在の削除候補プリントキューが使用中でないと判断された場合は、現在の削除候補プリントキューの削除処理を行うために、S 1 3 0 7 へ進む。S 1 3 0 7 において、ドライバ管理部 (1 2 5 2) は、OS (1 2 2) 提供の関数をコールすることにより、現在の削除候補プリントキューの削除処理を実施し、その後、S 1 3 0 2 へ戻る。ここで、もし何らかの理由で OS (1 2 2) が現在の削除候補プリントキューを削除できない場合は、削除できなかった旨のエラーメッセージ (不図示) を表示してから S 1 3 0 2 へ戻る。S 1 3 0 2 で、ドライバ管理部 (1 2 5 2) が削除候補プリントキューリストのすべてのプリントキューを処理したと判断した場合、S 1 3 0 9 へ進む。

【 0 0 7 9 】

S 1 3 0 9 においてドライバ管理部 (1 2 5 2) は、「不要となったプリンタドライバを削除する」チェックボックス (8 4 0 6) が選択されているか確認する。もし選択されていない場合は、S 1 3 1 6 へ進み、プリントキューの削除処理を終了する。もし選択されている場合は、S 1 3 1 0 へ進む。

【 0 0 8 0 】

S 1 3 1 0 において、ドライバ管理部 (1 2 5 2) は、S 1 3 0 1 で生成した削除候補プリンタドライバリストのすべてのプリンタドライバについて S 1 3 1 1 以降の処理を行ったかを判断する。もし削除候補プリンタドライバリストのすべてのプリンタドライバについて処理した場合、ループ処理から抜けて S 1 3 1 6 へ進み、プリントキューの削除処理を終了する。もし削除候補プリンタドライバリストに未処理のものがある場合は S 1 3 1 1 へ進み、未処理のプリンタドライバの何れかを取得して現在の削除候補プリンタドライバとする。S 1 3 1 2 では、ドライバ管理部 (1 2 5 2) は、現在の削除候補プリンタドライバが、汎用プリンタドライバ (1 2 7) か、もしくは仮想プリンタドライバ (1 2 6) でないことを、そのモデル名を用いて判断する。もし現在の削除候補プリンタドライバが、それらのプリンタドライバである場合、次回以降の仮想プリンタドライバ (1 2 6) を用いた印刷を可能とするために、S 1 3 1 5 へ進み、ドライバ管理部 (1 2 5 2) は現在の削除候補プリンタドライバを削除せずに、S 1 3 1 0 に戻る。もし現在の削除候補プリンタドライバが、汎用もしくは、仮想プリンタドライバでない場合、S 1 3 1 3 へ進む。なお本実施例では、モデル名を用いて、汎用プリンタドライバ (1 2 7) か、もしくは仮想プリンタドライバ (1 2 6) であるかを判断したが、特に識別情報や方法を限定する必要はない。例えばプリントードライバ構成情報 (3 0 4) の一部の情報として、別に識別子を持っていたてもよい。

【 0 0 8 1 】

S 1 3 1 3 において、ドライバ管理部 (1 2 5 2) は、OS (1 2 2) 提供の関数をコールすることにより、図 3 に示したプリンタドライバ、プリントキュー、ポートの各種情報を取得する。現在の削除候補プリンタドライバが他のプリントキューに関連付けられているかを判断する。もし現在の削除候補プリンタドライバに関連付けられたプリントキューが存在しなければ、S 1 3 1 4 に進み、現在の削除候補プリンタドライバを削除する。もし現在の削除候補プリンタドライバに関連付けられたプリントキューが存在する場合、S 1 3 1 5 に進み、現在の削除候補プリンタドライバを削除せずに、S 1 3 1 0 に戻る。

S 1 3 1 4において、ドライバ管理部(1 2 5 2)は、OS(1 2 2)提供の関数をコールすることにより、現在の削除候補プリントドライバの削除処理を実施し、その後S 1 3 1 0へ戻る。もし何らかの理由でOS(1 2 2)が現在の削除候補プリントドライバを削除できない場合は、削除できなかった旨のエラーメッセージ(不図示)を表示し、S 1 3 1 0に戻る。

【 0 0 8 2 】

以上より、ホストコンピュータ上のプリントキューを適切に管理することにより、ユーザが必要とするプリントキューのみ残すことが可能となり、次回以降のプリントキューの選択が容易になる。よって、ユーザの利便性が著しく向上する。

【 0 0 8 3 】

図14のシーケンス図を用いて「編集プレビュー」ダイアログ(8 2 0 0)の「戻る」ボタン(8 2 0 6)を押下した場合のプリントキュー整理処理を示す。S 7 0 0 ~ S 7 1 3までシーケンスは図7と同じ処理である。ここでは説明を省略する。

【 0 0 8 4 】

S 1 4 0 0で「戻る」ボタン(8 2 0 6)のユーザによる押下をうけて、プリントジョブマネージャー(1 2 5)は、「編集プレビュー」ダイアログ(8 2 0 0)から「出力デバイス特定」ダイアログ(8 1 0 0)へ画面遷移を行う(S 1 4 0 1)。その画面遷移をうけて出力デバイス特定処理(1 2 5 1)は、ドライバ管理部(1 2 5 2)へプリントキュー整理依頼を出す(S 7 1 7)。そして、ドライバ管理部(1 2 5 2)は、「削除方法指定」のダイアログで指定された情報でプリントキュー整理処理を実行する(S 7 1 8)。この処理により、印刷処理の過程であっても、プリントキューの総数を抑えることができる。

【 0 0 8 5 】

なお、「編集プレビュー」ダイアログ(8 2 0 0)から「出力デバイス特定」ダイアログ(8 1 0 0)へ画面遷移を行う度に、ドライバ管理部(1 2 5 2)はプリントキュー整理処理(S 7 1 8)を実行する必要はない。例えば現在の印刷シーケンスの「戻る」ボタン(8 2 0 6)の回数をカウントし、一定回数毎にドライバ管理部(1 2 5 2)がプリントキュー整理処理を実行する方法でもよい。その後の印刷シーケンスは図7のS 7 0 9以降の手順と同様であり、ここでは説明を省略する。

【 0 0 8 6 】

以上、図7と図14のシーケンスを用いて、仮想プリンタドライバを用いた印刷処理シーケンス内でのプリントキュー整理処理の実行を説明した。

【 0 0 8 7 】

実際に印刷につかったプリントキューなどの外部メモリに保存管理することにより、プリントキュー整理処理(S 7 1 8)の実行は、上記タイミングに限定する必要はなく、他のタイミングでも実施することが可能になる。例えば、次の印刷のタイミングで、出力準備処理(S 7 1 1)を行う前にプリントキュー整理処理(S 7 1 8)を実行しても良い。また仮想プリンタドライバ(1 2 6)などのユーザインターフェースにボタンを設け、印刷とは関係のないタイミングでユーザによるその押下を受け実行してもよい。さらに、例えば印刷処理と直接関係ないタイミングであるホストコンピュータ(1 0 1)へのユーザのログオンまたはログオフ時や、定期的なタイマー間隔を設けて実行してもよい。

【 0 0 8 8 】

OS(1 2 2)が提供する機能として、設置後のプリントキュー名称を別名に変更できる機能が存在する。このプリントキュー名称の変更に追従するために、上記では、プリントキューのデータ構造(3 0 2)に、自動生成識別子(3 0 6)と更新日時(3 0 7)を保存し、プリントキュー名称変更に影響を受けない構成とした。もし特に設置後のプリントキュー名称がなされない運用を前提とすれば、それらの情報保存先をプリントキューのデータ構造(3 0 2)に限定する必要はない。例えばプリントキュー名称をキーとしたプリントキューのデータ構造(3 0 2)と同様なデータ構造を用意し、外部メモリ(1 2 1)へ自動生成識別子(3 0 6)や更新日時(3 0 7)を保存することも可能である。また

10

20

30

40

50

アクセス制限などにより、プリントキューのデータ構造(302)に書き込めない場合も同様に他のデータ構造を用意して管理することもできる。

【0089】

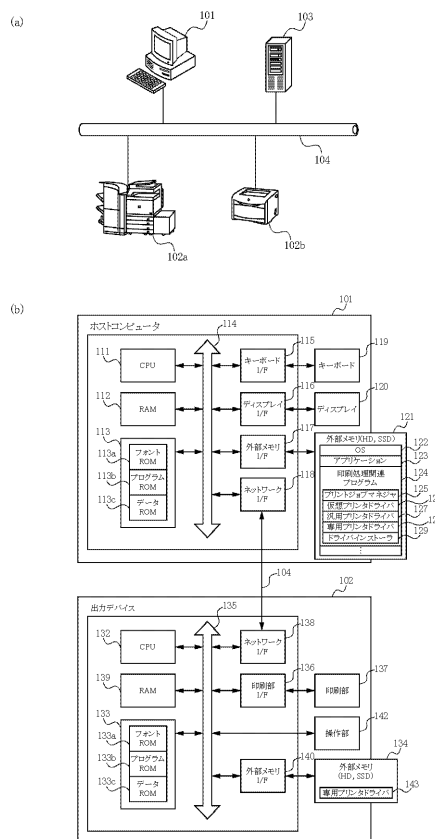
図12のS1202では、ドライバ管理部(1252)が、プリントキューのデータ構造(302)の更新日時(307)に現在の日時を記した。仮に汎用プリンタドライバ(127)や専用プリンタドライバ(128)が印刷完了時の更新日時(307)の更新をする仕組み(S1202)を持てば、仮想プリンタドライバ経由でないそれらの直接指定の印刷時も更新日時(307)の更新が可能になる。更にOS(122)が印刷の監査機能を持っている場合、その監査ログよりプリントキューの利用状況を判断することができる。その場合、更新日時(307)として独自データ構造を持たなくても実現可能である。

10

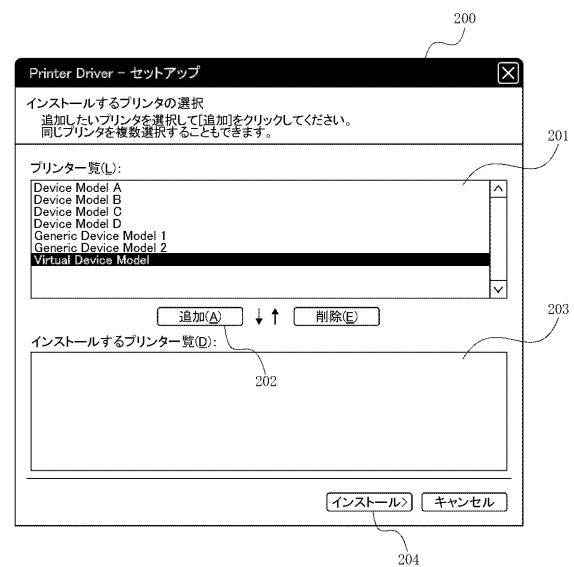
【0090】

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(またはCPU等)がプログラムを読み出して実行する処理である。

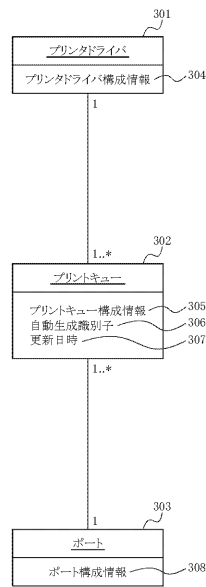
【図1】



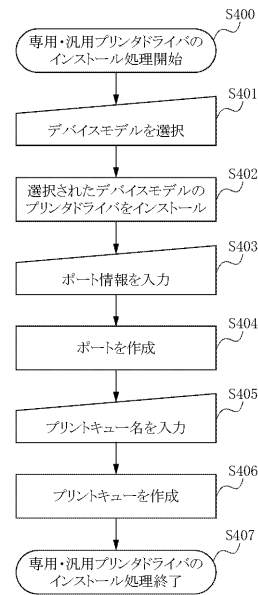
【図2】



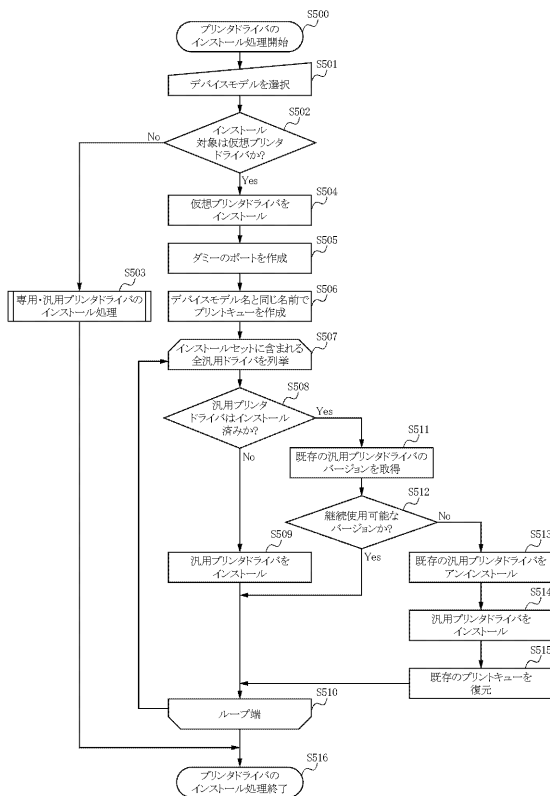
【図 3】



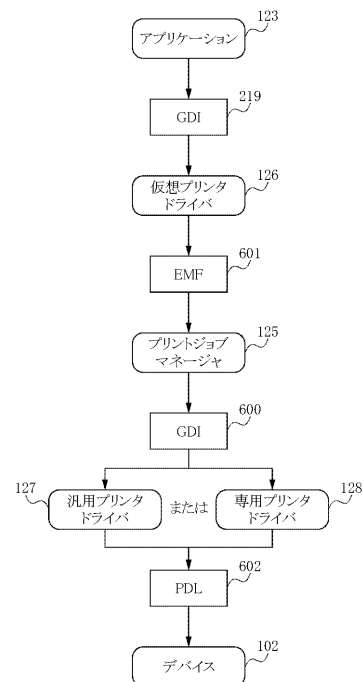
【図 4】



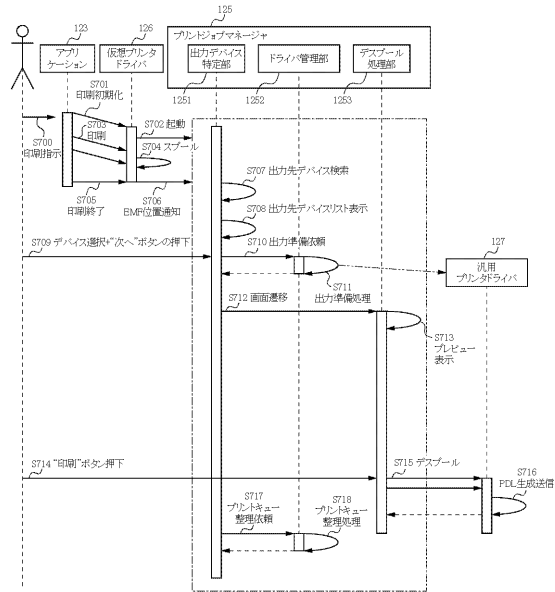
【図 5】



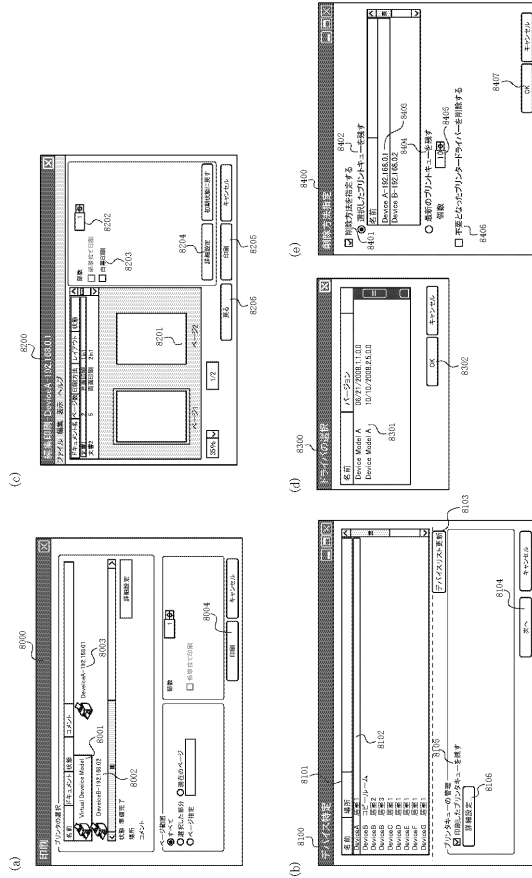
【図 6】



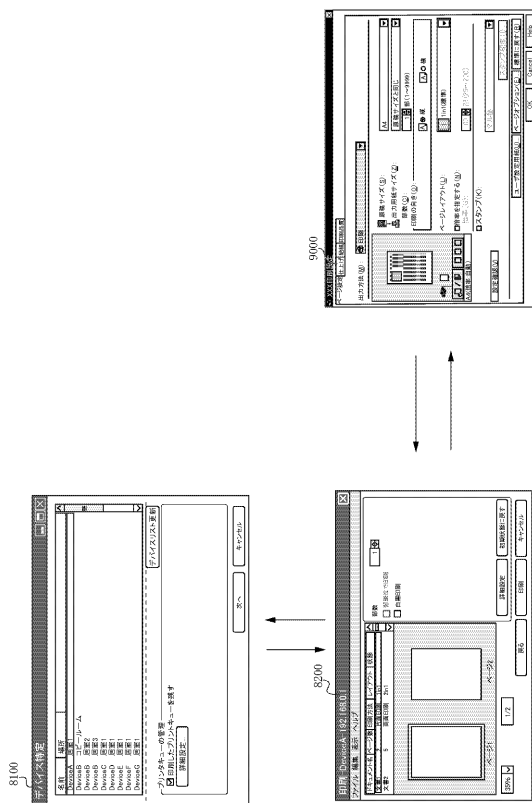
【図 7】



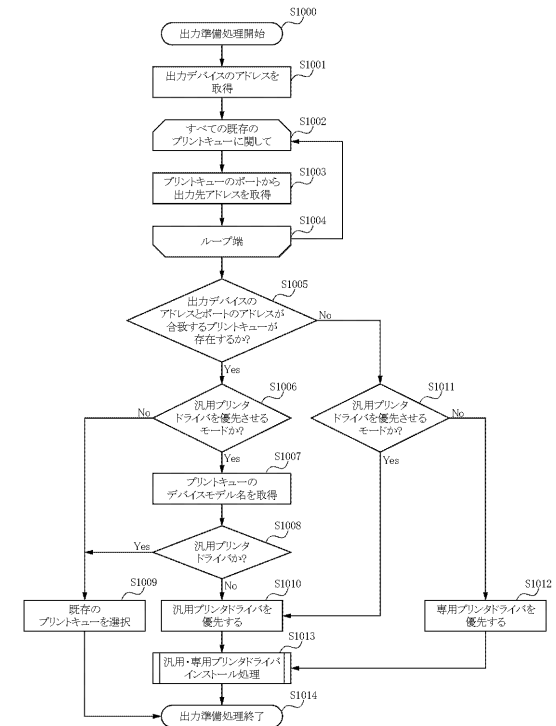
【図 8】



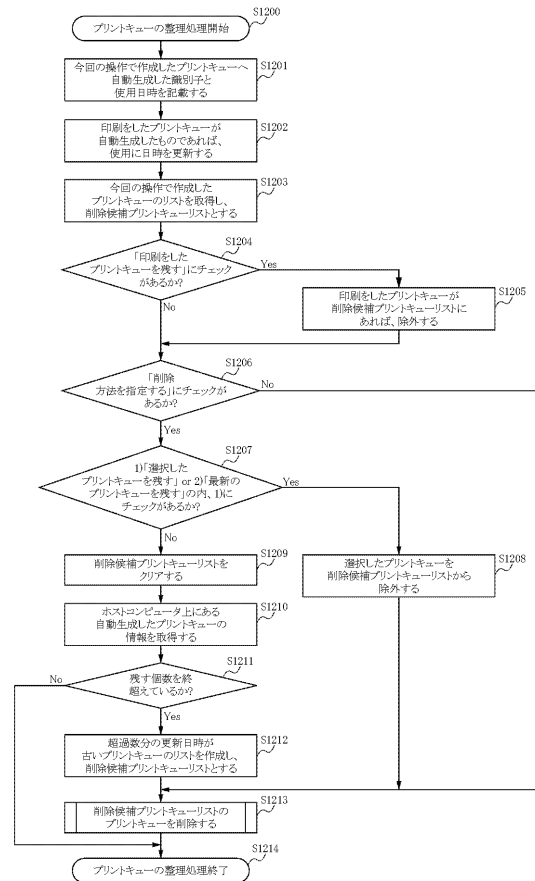
【図 9】



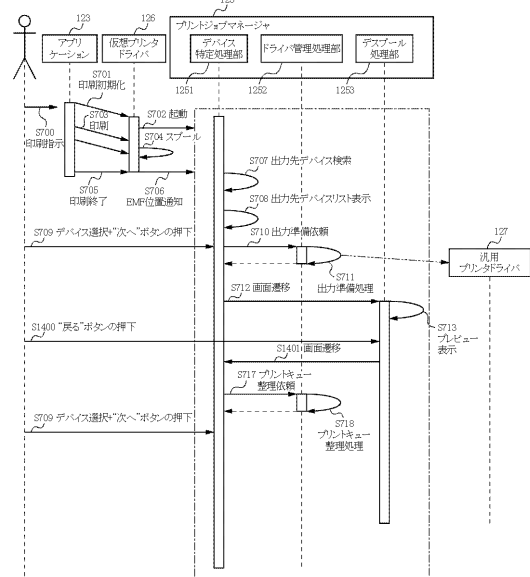
【図 10】



【 図 1 2 】



【 図 1 4 】



フロントページの続き

(56)参考文献 特開2007-065894(JP,A)
特開2009-201060(JP,A)
特開2001-243168(JP,A)
特開2002-091732(JP,A)
特開2006-155289(JP,A)
特開2008-269058(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 3/12
G06F 9/445