

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4785580号
(P4785580)

(45) 発行日 平成23年10月5日(2011.10.5)

(24) 登録日 平成23年7月22日(2011.7.22)

(51) Int.Cl.

G 06 F 3/12 (2006.01)
B 41 J 29/38 (2006.01)

F 1

G 06 F 3/12
B 41 J 29/38K
Z

請求項の数 8 (全 28 頁)

(21) 出願番号 特願2006-75550 (P2006-75550)
 (22) 出願日 平成18年3月17日 (2006.3.17)
 (65) 公開番号 特開2007-249859 (P2007-249859A)
 (43) 公開日 平成19年9月27日 (2007.9.27)
 審査請求日 平成21年3月16日 (2009.3.16)

(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 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 中村 忠弘
 東京都大田区下丸子3丁目30番2号 キ
 ャノン株式会社内

最終頁に続く

(54) 【発明の名称】 画像形成装置及びその制御方法、プログラム

(57) 【特許請求の範囲】

【請求項 1】

第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置であって、前記情報処理装置より印刷データを受信する受信手段と、

前記受信手段で受信した印刷データが、前記第1アプリケーションと第1プリンタドライバの組み合わせによる処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第2プリンタドライバの組み合わせによる処理経路を介して生成された印刷データでないと解析された場合、前記受信手段によって受信された印刷データの印刷を禁止するように制御する制御手段と

を備えることを特徴とする画像形成装置。

【請求項 2】

第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置であって、前記情報処理装置より印刷データを受信する受信手段と、

前記受信手段で受信した印刷データが、前記第1アプリケーションと第2プリンタド

10

20

イバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第1プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データであると解析された場合、前記描画変換を伴わない種類のプリンタドライバを特定する特定手段と、

前記特定手段によって特定された種類のプリンタドライバを含むデバイス情報を前記情報処理装置に対して通知する通知手段

を備えることを特徴とする画像形成装置。

【請求項3】

前記通知手段は、デバイス探索メッセージに含まれるメッセージ送信元情報に対して前記デバイス情報を通知する

10

ことを特徴とする請求項2に記載の画像形成装置。

【請求項4】

第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置の制御方法であって、

前記情報処理装置より印刷データを受信する受信工程と、

前記受信工程で受信した印刷データが、前記第1アプリケーションと第1プリンタドライバの組み合わせによる処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第2プリンタドライバの組み合わせによる処理経路を介して生成された印刷データでないと解析された場合、前記受信工程によって受信された印刷データの印刷を禁止するように制御する制御工程と

20

を備えることを特徴とする画像形成装置の制御方法。

【請求項5】

第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置の制御方法であって、

30

前記情報処理装置より印刷データを受信する受信工程と、

前記受信工程で受信した印刷データが、前記第1アプリケーションと第2プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第1プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データであると解析された場合、前記描画変換を伴わない種類のプリンタドライバを特定する特定工程と、

前記特定工程によって特定された種類のプリンタドライバを含むデバイス情報を前記情報処理装置に対して通知する通知工程

を備えることを特徴とする画像形成装置の制御方法。

【請求項6】

第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置の制御をコンピュータに実行させるためのプログラムであって、

40

前記情報処理装置より印刷データを受信する受信工程と、

前記受信工程で受信した印刷データが、前記第1アプリケーションと第1プリンタドライバの組み合わせによる処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第2プリンタドライバの組み合わせによる処理経路を介して生成された印刷データでないと解析された場合、前記受信工程によって受信された印刷データの印刷を

50

禁止するように制御する制御工程と
をコンピュータに実行させることを特徴とするプログラム。

【請求項 7】

第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置の制御をコンピュータに実行させるためのプログラムであって、

前記情報処理装置より印刷データを受信する受信工程と、

前記受信工程で受信した印刷データが、前記第1アプリケーションと第2プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第1プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データであると解析された場合、前記描画変換を伴わない種類のプリンタドライバを特定する特定工程と、

前記特定工程によって特定された種類のプリンタドライバを含むデバイス情報を前記情報処理装置に対して通知する通知工程

をコンピュータに実行させることを特徴とするプログラム。

【請求項 8】

前記通知工程は、デバイス探索メッセージに含まれるメッセージ送信元情報を対して前記デバイス情報を通知する

ことを特徴とする請求項7に記載のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、印刷データを受信して、印刷を行う画像形成装置及びその制御方法、プログラムに関するものである。

【背景技術】

【0002】

ホストコンピュータ上のアプリケーションから画像やテキストを印刷装置にて印刷する際には、図1に示すような構成が一般的である。

【0003】

即ち、ホストコンピュータ3000上のアプリケーション101は、グラフィクスエンジン103に対して描画データを渡し、その描画データはグラフィクスエンジン103にて加工されてからプリンタドライバ104に渡される。次に、プリンタドライバ104にて、プリンタ1500に対する印刷データ(例えば、PDL: Page Description Language)を生成し、その印刷データはスプーラ105に格納された後、プリンタ1500に送られるという構成である。

【0004】

特に、グラフィクスエンジン103は、アプリケーション101で作成された描画データの解像度を変換したり、プリンタドライバ104の処理能力に合わせてシミュレート処理等の描画データの加工を行う。これによって、アプリケーション101とプリンタドライバ104が互いに非依存な関係で動作することができる。このグラフィクスエンジン103は、通常、OS(Operating System)102の一部として提供される。

【0005】

一方、このグラフィクスエンジンは必ずしも一つであるとは限らず、二つ以上のグラフィクスエンジンが共存する構成が存在する。

【0006】

例えば、2005年米国シアトルにてWinHEC 2005というハードウェアエンジニアリングカンファレンスがMicrosoft社によって開催された。この場にて、

10

20

30

40

50

Microsoft社の新しいOS (Windows (登録商標) Vista) が図2に示すような2つのグラフィクスエンジン (GDI 202、WPF 212) が共存する構成になることが紹介されている (非特許文献1)。尚、WPFとは、Windows (登録商標) Presentation Foundationの略称である。

【0007】

従来のMicrosoft社の印刷処理システムは、Win32 APIというAPIを使用するアプリケーションからGDIと呼ばれるグラフィクスエンジンを活用して、GDIから呼び出されるプリンタドライバにて描画データから印刷データを作成する。この印刷処理フローを、GDIプリントパスと称している。

【0008】

ここで、APIは、Application Programming Interfaceの略称である。また、GDIは、Graphic Device Interfaceの略称である。また、以下では、Win32 APIを使用するアプリケーションを、Win32アプリケーション201と略称する。また、GDIから呼び出されるプリンタドライバを、GDIプリンタドライバ203と略称する。

【0009】

一方、Windows (登録商標) Vistaでは、従来のGDIプリントパスに加え、XPSプリントパスという新しい印刷処理フローが追加される。XPSプリントパスとは、WinFX APIを使用するアプリケーションからWPFと呼ばれるグラフィクスエンジンを活用して、XPS形式の描画データをプリンタドライバにて印刷データに変換する印刷処理フローである。

【0010】

ここで、XPSは、XML Paper Specificationの略称である。また、以下では、WinFX APIを使用するアプリケーションを、WinFXアプリケーション211と略称する。また、XPS形式の描画データをプリンタドライバを、XPSプリンタドライバ213と略称する。

【0011】

このように、Windows (登録商標) Vistaにおいて、アプリケーションは、Win32アプリケーション201とWinFXアプリケーション211の2種類に区別される。また、グラフィクスエンジンはGDI 202とWPF 212の2つがOS 10 2上に存在する。

【0012】

ここで、Win32アプリケーション201はGDI 202に対して描画データを渡し、WinFXアプリケーション211はWPF 212に対して描画データを渡す。また、プリンタドライバは、GDIプリンタドライバ203とXPSプリンタドライバ213の2種類に区別される。GDIプリンタドライバ203は、GDI 202から描画データを受け取りプリンタ1500に対する印刷データに変更する。また、XPSプリンタドライバ213は、WPF 212から描画データを受け取りプリンタ1500に対する印刷データに変更する。そして、プリンタドライバで作成された印刷データはスプーラ105を経てプリンタ1500に送信される。

【0013】

更に、このGDI 202とWPF 212といったグラフィクスエンジンは、互いに連携することができる。そして、この連携により、Win32アプリケーション201からXPSプリンタドライバ213に描画データを渡すことができる。一方で、WinFXアプリケーション211からGDIプリンタドライバ203に描画データを渡すことができる。

【0014】

図3にて、Windows (登録商標) Vistaにおける2つのグラフィクスエンジンの連携の仕組みを説明する。

【0015】

10

20

30

40

50

印刷処理フロー(1)は、従来のGDIプリントパスを示している。Win32アプリケーション201から渡される描画データは、GDI202にてEMF(Enhanced Metafile)形式でEMFスプールファイル301として格納される。その後、GDIプリンタドライバ203にて印刷データへの変換が行われる。

【0016】

印刷処理フロー(4)は、Windows(登録商標)Vistaで追加されるXPSプリントパスを示している。WinFxアプリケーション211から渡される描画データは、WPF212にてXPSスプールファイル311として格納される。その後、XPSプリンタドライバ213にて印刷データへの変換が行われる。

【0017】

印刷処理フロー(3)は、WinFxアプリケーション211からGDIプリンタドライバ203で印刷する場合の印刷処理フローを示している。WinFxアプリケーション211から渡される描画データは、WPF212を経由してXPS GDI変換モジュール312にて、EMF形式の描画データに変換され、EMFスプールファイル301として格納される。その後、GDIプリンタドライバ203にて印刷データへの変換が行われる。

【0018】

印刷処理フロー(2)は、Win32アプリケーション201からXPSプリンタドライバ213で印刷する場合の処理フローを示している。Win32アプリケーション201から渡される描画データは、GDI202を経由してGDXPS変換モジュール302にてXPS形式の描画データに変換され、XPSスプールファイル311として格納される。その後、XPSプリンタドライバ213にて印刷データへの変換が行われる。

【0019】

このように、Windows(登録商標)Vistaにおいては、4つの印刷処理フローが用意されている。また、プリンタに対するプリンタドライバは、GDIプリンタドライバ203あるいはXPSプリンタドライバ213の一方を用意することで、Win32アプリケーション201及びWinFxアプリケーション211の両方からの印刷処理に対応できる。

【0020】

一方、XPS形式そのものに注目すると、その特徴の一つとして、セキュリティ機能を持たせられる点があげられる。一例として、XPS形式の文書データに電子署名を付加し、文書データの改ざんを検知することが可能である。この電子署名付加機能を利用すると、プリンタと連携したセキュリティソリューションが考えられる。

【0021】

例えば、プリンタに電子署名検証機能を持たせ、プリンタが受信したXPS文書に付加された電子署名を検証し、文書が改ざんされていないと判定された場合のみ印刷を実行するような印刷システムを構築することが可能となる。

【0022】

尚、プリンタが実際に印刷を行うかどうかを受信した印刷データを解析して判定する技術は、例えば、特許文献1で提案されている。この特許文献1においては、その判定方法として、印刷データにより指定されたジョブ属性が処理可能かどうかという基準を採用している。

【非特許文献1】<http://www.microsoft.com/whdc/device/print/default.mspx> (Advances in Windows(登録商標)Printing : TWPR05001_WinHEC05.ppt)

【特許文献1】特開2003-226061号公報

【発明の開示】

【発明が解決しようとする課題】

【0023】

しかしながら、XPS形式とEMF形式は異なるため、図3の印刷処理フロー(2)及び印刷処理フロー(3)のように、描画データの変換が発生してしまう恐れのある印刷処

10

20

30

40

50

理フローに対して印刷品質や機能面、印刷速度への課題が存在する。

【0024】

印刷品質面では、例えば、印刷処理フロー(2)において、EMF形式でサポートされているROP演算(ラスター操作演算)のような論理演算を含む描画データは、XPS形式ではサポートされていない。そのため、米国マイクロソフト社が決定する仕様によっては、そのような論理演算を含む描画データに対しては、GDI XPS変換モジュール302で変換する際に論理演算情報が欠落してしまう恐れがある。XPSプリンタドライバ213には論理演算情報が削除された描画データが渡される場合、Win32アプリケーション201において意図された出力結果をXPSプリンタドライバ213で作成することができなくなる可能性がある。10

【0025】

また、逆に、印刷処理フロー(3)において、XPS形式においてサポートされている高度なグラフィクス描画データは、EMF形式においてサポートされていない。そのため、そのような描画データに対してはXPS GDI変換モジュール312で変換する際に、Flatteningと呼ばれる局所的なBitmap化が行われる。その場合、グラフィクスデータがBitmapデータに変換されてしまうことにより、GDIプリンタドライバ203では元のオブジェクト属性が判断できなくなってしまい、最適な出力結果を作成することはできなくなる恐れがある。

【0026】

そのため、印刷処理フロー(2)や(3)が使用された場合、プリンタから出力された印刷結果はユーザの意図したものから品質的に劣ってしまう恐れがある。20

【0027】

セキュリティ面においては、例えば、印刷処理フロー(3)において、XPS形式においてサポートされる電子署名情報は、EMF形式においてはサポートされていない。そのため、XPS形式の文書を印刷処理フロー(3)を経由することにより、文書に付加されていた電子署名情報は失われ、プリンタは従来の系で印刷せざるを得ない。この場合、悪意あるユーザが意図的にセキュリティ制限を回避して印刷を行うことを阻止できなくなる恐れがある。

【0028】

本発明は上記の課題を解決するためになされたものであり、不正な印刷または印刷品質の低下を防止することができる画像形成装置及びその制御方法、プログラムを提供することを第1の目的とする。30

【0029】

また、本発明は、劣化した印刷品質での出力を極力減らすことができる画像形成装置及びその制御方法、プログラムを提供することを第2の目的とする。

【課題を解決するための手段】

【0031】

上記の目的を達成するための本発明による画像形成装置は以下の構成を備える。即ち、第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置であって、40

前記情報処理装置より印刷データを受信する受信手段と、

前記受信手段で受信した印刷データが、前記第1アプリケーションと第1プリンタドライバの組み合わせによる処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第2プリンタドライバの組み合わせによる処理経路を介して生成された印刷データでないと解析された場合、前記受信手段によって受信された印刷データの印刷を禁止するように制御する制御手段と

を備える。

【0032】

10

20

30

40

50

上記の目的を達成するための本発明による画像形成装置は以下の構成を備える。即ち、第1アプリケーションからの印刷指示に応じて、第1プリンタドライバが処理可能な第1描画データを生成する第1描画部と、第2アプリケーションからの印刷指示に応じて、第2プリンタドライバが処理可能な第2描画データを生成する第2描画部とが動作する情報処理装置で生成された印刷データを受信して、印刷を行う画像形成装置であって、

前記情報処理装置より印刷データを受信する受信手段と、

前記受信手段で受信した印刷データが、前記第1アプリケーションと第2プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データ、または、前記第2アプリケーションと第1プリンタドライバの組み合わせによる描画変換を伴う処理経路を介して生成された印刷データであると解析された場合、前記描画変換を伴わない種類のプリンタドライバを特定する特定手段と、

前記特定手段によって特定された種類のプリンタドライバを含むデバイス情報を前記情報処理装置に対して通知する通知手段

を備える。

【発明の効果】

【0033】

本発明によれば、不正な印刷または印刷品質の低下を防止することができる画像形成装置及びその制御方法、プログラムを提供できる。

【0034】

また、本発明によれば、劣化した印刷品質での出力を極力減らすことができる画像形成装置及びその制御方法、プログラムを提供できる。

【発明を実施するための最良の形態】

【0035】

以下、本発明の実施の形態について図面を用いて詳細に説明する。

図3は、Windows(登録商標) Vistaにおける2つのグラフィクスエンジンの連携の仕組みを説明する。

【0036】

印刷処理フロー(1)は、従来のGDIプリントパスを示している。Win32アプリケーション201から渡される描画データは、GDI202にてEMF形式でEMFスプールファイル301として格納される。その後、GDIプリンタドライバ203にて印刷データへの変換が行われる。

【0037】

印刷処理フロー(4)は、Windows(登録商標)Vistaで追加されるXPSプリントパスを示している。WinFxアプリケーション211から渡される描画データは、WPF212にてXPSスプールファイル311として格納される。その後、XPSプリンタドライバ213にて印刷データへの変換が行われる。

【0038】

印刷処理フロー(3)は、WinFxアプリケーション211からGDIプリンタドライバ203で印刷する場合の印刷処理フローを示している。WinFxアプリケーション211から渡される描画データは、WPF212を経由してXPS GDI変換モジュール312にて、EMF形式の描画データに変換され、EMFスプールファイル301として格納される。その後、GDIプリンタドライバ203にて印刷データへの変換が行われる。

【0039】

印刷処理フロー(2)は、Win32アプリケーション201からXPSプリンタドライバ213で印刷する場合の処理フローを示している。Win32アプリケーション201から渡される描画データは、GDI202を経由してGDXPS変換モジュール302にてXPS形式の描画データに変換され、XPSスプールファイル311として格納される。その後、XPSプリンタドライバ213にて印刷データへの変換が行われる。

【0040】

10

20

30

40

50

このように、Windows（登録商標） Vistaにおいては、4つの印刷処理フローが用意されている。また、プリンタに対するプリンタドライバは、GDIプリンタドライバ203あるいはXPSプリンタドライバ213の一方を用意することで、Win32アプリケーション201及びWinFxアプリケーション211の両方からの印刷処理に対応できる。

【0041】

図2、図3では、情報処理装置は、第1アプリケーションであるWin32アプリケーション201と第1プリンタドライバであるGDIプリンタドライバ203を備える。また、情報処理装置は、第2アプリケーションであるWinFXアプリケーション211と第2プリンタドライバであるXPSプリンタドライバ213を備える。

10

【0042】

また、グラフィクスエンジンは、第1描画変換部であるGDI202と第2描画変換部であるWPF212の2つがOS102上に存在する。Win32アプリケーションは、GDI202に対して第1描画データであるGDI関数を渡し、WinFxアプリケーション211はWPF212に対して第2描画データであるWPF API形式データを渡す。

【0043】

GDIプリンタドライバ203はGDI202から描画データを受け取り印刷装置1500に対する印刷データに変換する。XPSプリンタドライバ213はWPF212から描画データを受け取り印刷装置1500に対する印刷データに変換する。プリンタドライバで作成された印刷データは、スプーラ105を経て印刷装置1500に送信される。

20

【0044】

尚、本実施形態では、変換モジュールを用いた描画変換を伴う印刷処理経路(2)及び(3)をクロスパスと総称する。一方、変換モジュールを用いた描画変換を伴わない印刷処理経路(1)及び(4)をストレートパスと総称する。

【0045】

図4は本発明の実施形態のコンピュータを含むプリンタ制御システムの構成を説明するブロック図である。

【0046】

尚、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、単体の機能であっても、複数の機器からな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。更には、LAN、WAN等のネットワークを介して接続がな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。

30

【0047】

図4において、3000はホストコンピュータであり、CPU1を備える。CPU1は、ROM3のプログラム用ROM3b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（アプリケーションプログラム）等のプログラムに基づいて、図形、イメージ、文字、表（表計算等を含む）等の構成要素が混在した文書処理を実行する。更に、CPU1は、ホストコンピュータ本体2000内のシステムバス4に接続される各デバイスを総括的に制御する。

40

【0048】

また、このROM3のプログラム用ROM3b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）等のプログラムも記憶している。また、ROM3のフォント用ROM3a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のデータを記憶している。また、ROM3のデータ用ROM3cあるいは外部メモリ11には、上記文書処理等の各種処理を行う際に使用する各種データを記憶する。

【0049】

2はRAMであり、CPU1の主メモリ、ワークエリア等として機能する。5はキーボードコントローラ（KBC）であり、キーボード（KB）9やポインティングデバイス（

50

不図示)からのキー入力を制御する。6はCRTコントローラ(CRTC)であり、CRTディスプレイ(CRT)10の表示を制御する。尚、CRTコントローラ6及びCRTディスプレイ10の代りに、LCDコントローラ及びLCDを用いても良い。

【0050】

7はディスクコントローラ(DKC)であり、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、編集ファイル等の外部メモリ11とのアクセスを制御する。さらに、DKC7は、印刷制御コマンド生成プログラム(以下、プリンタドライバ)等の各種データを記憶する、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とのアクセスも制御する。

【0051】

8はプリンタコントローラ(PRTC)であり、所定の双方向性インタフェース21を介してプリンタ1500に接続されて、プリンタ1500との通信制御処理を実行する。双方向性インタフェースの例には、USBインターフェース、IEEE1394インターフェース、無線LANインターフェース等がある。

【0052】

尚、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)処理を実行してGUIを提供することで、CRT10上のWYSIWYGを可能としている。また、CPU1は、CRT10上のマウスカーソル(不図示)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方法の設定を行える。

【0053】

プリンタ1500において、12はプリンタCPUであり、プリンタCPU12は、プリンタ制御部1000内のシステムバス15に接続される各デバイスを総括的に制御する。また、プリンタCPU12は、ROM13のプログラム用ROM13bあるいは外部メモリ14に記憶された制御プログラム等のプログラムに基づいて、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。

【0054】

また、このROM13のプログラム用ROM13bには、プリンタCPU12の制御プログラム等のプログラムを記憶している。また、ROM13のフォント用ROM13a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のデータを記憶している。また、ハードディスク(HD)、ICカード等の外部メモリ14がないプリンタの場合には、ROM13のデータ用ROM13cには、ホストコンピュータ3000上で利用される情報等を記憶している。

【0055】

プリンタCPU12は、入力部18を介してホストコンピュータ3000との通信処理が可能となっており、プリンタ1500内のステータス情報等の各種情報をホストコンピュータ3000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である。このRAM19は、増設ポート(不図示)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構成されている。

【0056】

尚、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。外部メモリ14は、メモリコントローラ(MC)20によりアクセスが制御される。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、フォームデータ等のデータを記憶する。また、1501は操作部であり、プリンタ1500に対する各種操作を実現するための操作パネルやスイッチ、LED表示器等が構成されている。

【0057】

また、外部メモリ14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

10

20

30

40

50

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良い。さらに、N V R A M (不図示)を有し、操作部 1501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。

【0058】

また、プリンタ 1500には、1個以上のP D L装置(P D Lボード)を接続し、C P U 12からP D L装置を制御可能とする。P D L装置は、ホストコンピュータ3000から受信した印刷データ(P D L)を解釈し、印刷部17が印刷可能なビットマップデータに変換する機能を有する画像形成部である。このP D L装置は、標準として1個以上接続されているが、後から追加することも可能である。また、プラグアンドプレイで使用されるI E E E 1284規定のデバイス情報は、このP D L装置が保持している。10

【0059】

尚、本実施形態の情報処理装置は、プリンタドライバを介して、プリンタが解釈可能な印刷データ(印刷ジョブ)を生成できる。

【0060】

また、通常、プリンタ 1500は、標準P D L装置としてP D L 1を備えており、オプションP D L装置としてP D L 2等の他の種類のP D Lを追加拡張可能であるものとする。

【0061】

この拡張は、専用のP D L装置を搭載する拡張ボードを物理的にプリンタ 1500に追加する構成であっても良い。あるいは、P D L装置がプログラムによって実現される場合には、プログラム用R O M 13bに対応するプログラムを追加する構成であっても良い。更には、予め複数種類のP D L装置それぞれに対応するプログラムをプログラム用R O M 13bに記憶しておき、それらから必要なP D L装置を、ユーザ操作によって入力されたライセンスキーによって効果にすることで追加する構成であっても良い。20

【0062】

また、P D Lの一例としては、L I P S、P o s t S c r i p t、P C L等がある。

【0063】

また、本実施形態に適用するプリンタ 1500には、レーザビームプリンタ及びインクジェットプリンタ等の他の各種印刷方式の印刷装置(画像形成装置)を使用することが可能である。また、複数種類の機能を有するM F P (Multi Function Peripheral:マルチファンクション周辺機器)であっても良い。30

【0064】

ここで、M F Pは、自装置内部に複数のジョブを記憶可能なハードディスク等の記憶装置を具備している。また、例えば、スキャナ部から出力されたジョブに対し、その記憶装置を介してプリンタ部でプリント可能にするコピー機能を具備している。また、P C(パソコン用コンピュータ)等の外部装置から出力されたジョブに対し記憶装置を介してプリンタ部でプリント可能にするプリント機能を具備している。このように、M F Pは、これらの機能を含む複数の機能を具備している。

【0065】

M F Pには、フルカラー機器とモノクロ機器があり、色処理機能や内部データ等を除いて、基本的な部分においては、フルカラー機器がモノクロ機器の構成を包含することが多い。そのため、ここでは、フルカラー機器に絞って説明し、必要に応じて随时モノクロ機器の説明を加えることとする。

【0066】

また、本実施形態で説明するシステムは、複数の機能を具備した複合機能型の画像形成装置(印刷装置)でも、プリント機能のみを具備した単一機能型の画像形成装置(印刷装置)であるS F Pを具備する構成でも良い。いずれにしても、本実施形態の制御が実現可能な構成であればよい。

【0067】

10

20

30

40

50

尚、SFPとは、Single Function Peripheral：单一機能周辺機器の略称である。

【0068】

次に、プリンタ1500のCPU12によって実行される機能構成について、図5を用いて説明する。

【0069】

図5は本発明の実施形態のプリンタの機能構成を示すブロック図である。

【0070】

ジョブ受信部501は、ホストコンピュータ3000から印刷ジョブを受信し、受信した印刷ジョブをジョブ解析部502に送信する。ジョブ解析部502は、印刷ジョブを解析し、後述の識別方法により印刷ジョブが生成された印刷処理フロー（印刷ジョブ生成経路）を識別する。尚、印刷ジョブと印刷データは同義である。10

【0071】

ジョブ解析部502の処理結果、及びモード設定部503により設定されたモードに従い印刷ジョブの処理方法が決定され、印刷ジョブを印刷する場合は、印刷ジョブは印刷部504に送信され、印刷が行われる。ジョブ削除部505は、ジョブ解析部502の処理結果に従い、印刷ジョブをキャンセルする処理を行う。通知部506は、ジョブ解析部502の処理結果に従い、ホストコンピュータ3000に通知を行う。

【0072】

次に、本実施形態のホストコンピュータ3000上とプリンタ間のデータ処理例について、図6を用いて説明する。20

【0073】

図6は本発明の実施形態のホストコンピュータ上とプリンタ間のデータ処理例を示す図である。

【0074】

WinFXアプリケーション211からプリンタ1500へデータが渡る際にGDIプリンタドライバ203またはXPSプリンタドライバ213を経由する2種類のプリントパスが存在することは前述のとおりである。

【0075】

ここで、WinFXアプリケーション211において、電子署名情報が付加された印刷ジョブを発行する状況を想定する。30

【0076】

XPSプリンタドライバ213を経由するプリントパスでは電子署名情報付きの描画データ323は、XPSプリンタドライバ213によって処理される。そして、XPSプリンタドライバ213により変換されたPDLデータと共に電子署名情報325がプリンタ1500へ送信される。ここでフォーマットについては後述する。

【0077】

一方、GDIプリンタドライバ203経由のプリントパスでは、XPS GDI変換モジュール312によってGDIに変換される際に電子署名情報を失い、電子署名情報なしの描画データ322が生成される。そして、この電子署名情報なしの描画データ322は、GDIプリンタドライバ203によって処理される。その結果、電子署名情報のないPDLデータ324となってプリンタ1500へ送信される。40

【0078】

このように、本実施形態では、この電子署名情報が失われる可能性がある印刷環境において、その印刷処理の禁止／実行を適応的に制御するためのセキュリティ機能を実現する。

【0079】

このセキュリティ機能を実現するにあたり、本実施形態では、ホストコンピュータ上で生成された印刷ジョブが生成された経路に基づいて、印刷処理の禁止／実行を設定するためのモード設定部503を構成している。そこで、このモード設定部503が生成するモ50

ード設定画面について、図7を用いて説明する。

【0080】

図7は本発明の実施形態における画像形成装置のモード設定画面の一例を示す図である。

【0081】

このモード設定画面700は、操作部1501上に表示され、プリンタの管理者が設定を行う。

【0082】

各設定項目には、それぞれON/OFFを切り替えるボタンが存在し、どちらかのボタンを選択すると、ボタンは選択状態を示す黒色表示となる。

10

【0083】

701は、セキュリティ設定の有効/無効を切り替えるためのボタンである。ボタン701がOFFの場合、ボタン702~706は、グレーアウトされ各設定項目への設定は禁止となり、設定はすべて無効となる。

10

【0084】

702は、GDI経由での印刷を禁止するか否かを設定するためのボタンである。ボタン702が、ONの場合は印刷処理フロー(1)及び(3)で処理された印刷ジョブはキャンセルされ、印刷を禁止する。但し、ボタン702がONであっても、ボタン703がONの場合は、印刷処理フロー(1)で処理された印刷ジョブは印刷する。つまり、ボタン703がONの場合は、Win32アプリケーションからの印刷ジョブについては、その印刷を許可することになる。

20

【0085】

よって、図3の印刷処理フロー(1)のプリントパスを介して生成された印刷データは印刷されるが、図3の印刷処理フロー(3)のプリントパスを介して生成された印刷データは印刷されなくなる。また、図7の設定画面にて「クロスパス経由の印刷禁止」ボタンを設けることにより、図3の印刷処理フロー(2)及び(3)のプリントパスを介して生成された印刷データの印刷を禁止することもできる。

【0086】

このように、本実施形態のプリンタは、印刷処理経路毎に印刷データの処理に係る設定を行える。例えば、図7の画面が、Win32アプリケーションとXPSドライバの組み合わせによる処理経路(2)、またはWinFXアプリケーションとGDIドライバの組み合わせによる処理経路(3)を介して生成された印刷データの印刷を禁止するか否かの設定項目を備えることができる。

30

【0087】

704は、GDI経由での印刷ジョブを受信した場合に管理者にメールで通知を行うか否かを設定するためのボタンである。尚、通知先となる管理者のメールアドレスは、管理者情報を設定する別の画面において設定済であるものとする。

【0088】

705は、GDI経由での印刷ジョブを受信した場合にユーザに通知するか否かを設定するためのボタンである。後述するように、この通知は、印刷ジョブを送信してきたホストコンピュータのIPアドレスに対してユニキャストで行われる。

40

【0089】

また、ボタン704及び705による通知の通知方法は一例であり、メッセージダイアログによって指定されたホストコンピュータに通知したり、通知内容を指定のプリンタに出力する構成としても良い。

【0090】

そして、この状態でOKボタン706を操作すると、図7中の設定状態(モード設定情報)として、例えば、RAM19上に記憶されるとともに、モード設定画面700の表示が消去される。一方、キャンセル(Cancel)ボタン707を操作すると、図7中の設定状態がキャンセルされて、このモード設定画面700の表示が消去される。

50

【0091】

尚、図7に示すモード設定画面700の設定項目は一例であり、用途や目的に応じて様々な設定項目を構成することが可能である。本実施形態の主眼とするところは、印刷データを生成するための複数種類の印刷処理フローが存在していて、その使用される印刷処理フローによっては、印刷不具合や不正な印刷が発生し得る状況を事前に回避したり、禁止することにある。そのため、この目的を達成するための設定項目であれば、様々な設定項目を構成できることは言うまでもない。

【0092】

次に、プリンタ1500が実行する処理について、図8を用いて説明する。

【0093】

図8は本発明の実施形態のプリンタが実行する処理を示すフローチャートである。尚、図8の各ステップの処理は、プリンタのCPU12によって実行される。

10

【0094】

図8は、特に、プリンタ1500がホストコンピュータ3000から印刷ジョブを受信した際の処理を示すフローチャートである。

【0095】

ステップS801で、ジョブ受信部501は、印刷ジョブを受信する。ステップS802で、ジョブ解析部502は、印刷ジョブを解析し、印刷ジョブが生成された経路（印刷処理フロー（印刷ジョブ生成経路））を識別する。尚、ジョブ生成経路の識別方法については図18にて後述する。次に、ステップS803で、モード設定部503は、RAM19上のモード設定情報を参照して、セキュリティ設定として、ボタン701の設定値がONであるか否かを判定する。

20

【0096】

セキュリティ設定がOFFである場合（ステップS803でNO）、ステップS809に進み、印刷ジョブを印刷して処理を終了する。一方、セキュリティ設定がONである場合（ステップS803でYES）、ステップS804に進み、ジョブ解析部502は、印刷ジョブ生成経路が印刷処理フロー（3）であるか否かを判定する。

【0097】

印刷処理フロー（3）である場合（ステップS804でYES）、プリンタ1500は、後述するサブルーチンSRを実行する。その後、ステップS805で、モード設定部503は、RAM19上のモード設定情報を参照して、セキュリティ設定（GDI経由の印刷禁止）として、ボタン702の設定値がONであるか否かを判定する。

30

【0098】

GDI経由の印刷禁止がONである場合（ステップS805でYES）、ステップS806に進み、印刷部504は、印刷ジョブをキャンセルした後、処理を終了する。一方、GDI経由の印刷禁止がOFFである場合（ステップS805でNO）、ステップS809に進み、印刷部504は、印刷ジョブを印刷して処理を終了する。

【0099】

一方、ステップS804において、印刷処理フロー（3）でない場合（ステップS804でNO）、ステップS807に進み、ジョブ解析部502は、印刷ジョブ生成経路が印刷処理フロー（1）であるか否かを判定する。

40

【0100】

印刷処理フロー（1）である場合（ステップS807でYES）、後述するサブルーチンSRを実行する。その後、ステップS808で、モード設定部503は、RAM19上のモード設定情報を参照して、セキュリティ設定（Win32アプリケーションからの印刷許可）として、ボタン703の設定値がONであるか否かを判定する。

【0101】

Win32アプリケーションからの印刷許可がOFFである場合（ステップS808でNO）、ステップS806に進み、プリンタ1500は、印刷ジョブをキャンセルした後、処理を終了する。一方、Win32アプリケーションからの印刷許可がONである場合

50

(ステップS808でYES)、印刷部504は、印刷ジョブを印刷して処理を終了する。このように、プリンタ1500は、ステップS802の解析結果と図7の設定内容に基づいて、受信した印刷データの処理を制御する。

【0102】

図8の処理より、プリンタ1500のCPU12は、ステップS802の解析の結果、アプリケーションとプリンタドライバの組み合わせが特定の組み合わせであり、かつ特定の組み合わせの場合に印刷を禁止することが設定される場合、印刷データの印刷を禁止するように制御する。その結果、プリンタ1500は、機密性の低下または印刷品質の劣化等を防止することが可能となる。

【0103】

次に、図8のサブルーチンSRの詳細について、図9を用いて説明する。

【0104】

図9は本発明の実施形態のサブルーチンSRの処理の詳細を示すフローチャートである。

【0105】

ステップS901で、モード設定部503は、RAM19上のモード設定情報を参照して、セキュリティ設定(管理者通知)として、ボタン704の設定値がONであるか否かを判定する。

【0106】

管理者通知がOFFである場合(ステップS901でNO)、ステップS904に進む。一方、管理者通知がONである場合(ステップS901でYES)、ステップS902に進み、通知部506は、予め設定された管理者のメールアドレスを取得する。次に、ステップS903で、通知部506は、管理者宛にメールを送信する。その後、ステップS904に進む。ここで、メールの一例を図10に示す。

【0107】

例えば、ユーザがWinFXアプリケーションにて、電子署名情報付きのデータを生成した場合を想定する。ここで、ユーザがWinFXアプリケーションを介してGDIプリンタドライバを選択した場合、WinFXアプリケーションにて生成された電子署名付きのデータの電子署名は削除されてしまい、セキュリティを保つことができなくなる恐れがある。

【0108】

プリンタは、受信した印刷データから印刷処理フロー(3)であることを解析した場合、上述したように電子署名情報が欠落した恐れがあると判定する。そして、プリンタは、適切ではないプリンタドライバを使用して印刷データが生成されたことを通知するため、図10のようなメールを生成して管理者へ送信する。

【0109】

ステップS904で、モード設定部503は、RAM19上のモード設定情報を参照して、セキュリティ設定(ユーザ通知)として、ボタン705の設定値がONであるか否かを判定する。

【0110】

ユーザ通知がOFFである場合(ステップS904でNO)、処理を終了する。一方、ユーザ通知がONである場合(ステップS904でYES)、ステップS905に進み、通知部506は、印刷ジョブを送信してきたホストコンピュータ3000にユニキャストで通知メッセージを送信し、処理を終了する。

【0111】

ここで、通知メッセージの一例を図11に示す。図11において、1101はプリンタの構成情報が変化したり、印刷ジョブのキャンセルが発生するたびに1ずつ増加していく整数である。ホストコンピュータ3000は、この値を取得することにより、プリンタ1500で上記事象が発生したことを検知することができる。

【0112】

10

20

30

40

50

このようにセキュリティ設定として、ボタン 701、702 及び 705 の設定値が ON に設定されている状態でホストコンピュータ 3000 から GDI 経由での印刷ジョブを受信すると、プリンタ 1500 は図 11 の通知メッセージを送信する。これにより、例えば、ホストコンピュータ 3000 上の GDI プリンタドライバがアンインストールされ、それ以降 GDI 経由での印刷を行わせないようにする（禁止）することが可能となる。

【0113】

この GDI プリンタドライバのアンインストールの処理について、図 12 を用いて説明する。

【0114】

図 12 は本発明の実施形態の GDI プリンタドライバのアンインストールのシーケンスを示す図である。 10

【0115】

1201 で、プリンタ 1500 はホストコンピュータ 3000 に通知メッセージを送信する。

【0116】

次に、ホストコンピュータ 3000 は、1202 で、プリンタ 1500 に対してデバイス情報要求を送信する。デバイス情報要求を受信したプリンタ 1500 は、1203 で、後述する方法でデバイス情報を生成し、生成したデバイス情報を、1204 で、ホストコンピュータ 3000 に送信する。 20

【0117】

このデバイス情報の一例を図 13 に示す。図 13 に示すように、デバイス情報には、デバイスが有するサービス（機能（画像形成部（PDL 装置）の種類等））のリストを含んでいる。また、図 13 中の <ServiceId> 要素の値は、各サービスを識別するための固有の ID、つまり、識別子であり、後でサービスを特定する際に使用される。 30

【0118】

次に、ホストコンピュータ 3000 は、1205 で、受信したデバイス情報から、必要なサービス ID を指定してサービス情報要求を送信する。サービス情報要求を受信したプリンタ 1500 は、1206 で、指定されたサービス ID に対応する IEEE1284 標準のデバイス情報の値を含むサービス情報をホストコンピュータ 3000 に送信する。このサービス情報の一例を図 14 に示す。 30

【0119】

尚、1204 で、プリンタ 1500 が複数のサービスリストを含むデバイス情報をホストコンピュータ 3000 に送信した場合、1205～1206 のシーケンスが繰り返されることになる。

【0120】

ホストコンピュータ 3000 は、1207 で、プリンタ 1500 から受信したサービス情報を参照し、プリンタが返さなかった DeviceID に対応するプリンタドライバをアンインストールする。

【0121】

次に、図 12 の 1203 において、デバイス情報要求を受信したプリンタ 1500 がデバイス情報を生成する際の処理について、図 15 を用いて説明する。 40

【0122】

図 15 は本発明の実施形態のデバイス情報要求を受信したプリンタがデバイス情報を生成する際の処理を示すフローチャートである。

【0123】

ステップ S1501 で、プリンタ 1500 は、デバイス情報要求を受信する。ステップ S1502 で、プリンタがサポートしている PDL 装置の PDL リストから、プリンタ 1500 で印刷可能な PDL を 1 個取得する。次に、ステップ S1503 で、取得した PDL 装置に対応する XPS 対応ドライバ用の DeviceID を有するサービスに対応するサービスエントリをデバイス情報に追加する。つまり、ステップ S1503 において、ブ 50

リント 1500 は、取得した PDL に対応する XPS プリンタドライバは有効なプリンタドライバであるとの情報を追加する。

【0124】

次に、ステップ S1504 で、プリンタ 1500 は、RAM19 上のモード設定情報を参照して、セキュリティ設定として、ボタン 701 及び 702 の設定値を取得し、GDI 印刷が受付可能であるか否かを判定する。受付可能でない場合（ステップ S1504 で NO）、ステップ S1506 に進む。

【0125】

一方、受付可能である場合（ステップ S1504 で YES）、ステップ S1505 に進み、取得した PDL に対応する GDI 対応ドライバ用の DeviceID を有するサービスに対応するサービスエントリをデバイス情報に追加する。つまり、ステップ S1505 において、プリンタ 1500 は、取得した PDL に対応する GDI プリンタドライバは有効であるとの情報をデバイス情報に追加する。10

【0126】

ステップ S1506 で、プリンタ 1500 は、PDL リスト中のすべての PDL 装置に対して処理が終了したか否かを判定する。終了していない場合（ステップ S1506 で NO）、ステップ S1502 に戻り、未処理の PDL 装置を取得して、上述の処理を実行する。一方、終了している場合（ステップ S1506 で YES）、ステップ S1507 に進み、生成したデバイス情報をホストコンピュータ 3000 に送信して、処理を終了する。20

【0127】

以上の処理により、プリンタ 1500 は、自ら処理可能な印刷データを生成するプリンタドライバを指定することが可能となる。例えば、ステップ S1504 において GDI 印刷が不可能と判定した場合（ステップ S1504 で NO）、GDI プリンタドライバは有効なプリンタドライバとしてデバイス情報に追加されない。

【0128】

このデバイス情報を取得したホストコンピュータは、デバイス情報に GDI プリンタドライバが含まれていないため、GDI プリンタドライバを介して印刷データを生成してはいけないと判定することができる。その結果、例えば、WinFX アプリケーションを用いて作成した電子署名付きのデータにおける、電子署名情報が削除された印刷データがプリンタに送信されるという懼れが解消できる。30

【0129】

つまり、ホストコンピュータは、デバイスから通知されるデバイス情報に従ってプリンタドライバを制御することができる。例えば、ホストコンピュータは、自ら保持しているプリンタドライバの内、通知されたデバイス情報に含まれていないプリンタドライバを特定して、そのプリンタドライバをアンインストールすることも可能である。このように、プリンタは、印刷可能な印刷データを生成できるプリンタドライバを示す識別情報を、ホストコンピュータへ通知することができる。

【0130】

また、図 12 の処理を用いることにより、クロスパスを介して生成された印刷データを受信した際に、プリンタが適切なプリンタドライバを推奨する処理について、図 19 を用いて説明する。尚、この場合、図 12 に記述されているアンインストール処理は、実行されなくて良い。

【0131】

プリンタ 1500 は、印刷データを受信したか否かを判定し（ステップ S1901）、受信した場合、印刷データがクロスパスを介して生成された印刷データか否かを判定する（ステップ S1902）。尚、クロスパスを介して生成されたか否かの解析方法は、図 18 にて後述する。

【0132】

ステップ S1902 により、ジョブ解析部 502 が、クロスパスを介して生成された印刷データであると解析した場合に、描画モジュールの変換を伴わずに印刷データを生成で40

きるプリンタドライバの種類を特定する(ステップS1903)。詳細には、ジョブ解析手段は、図18の処理を用いることにより、印刷要求元のアプリケーションの種類を判定できる。

【0133】

具体的には、要求元のアプリケーションが、WinFXアプリであるかWin32アプリであるかを判定する。その結果に基づいて、プリンタは、描画変換を伴わないプリンタドライバの種類を特定する。具体的には、印刷要求元がWinFXアプリケーションである場合、プリンタは、XPSプリンタドライバを特定する。一方、印刷要求元がWin32アプリケーションである場合、プリンタは、GDIプリンタドライバを特定する。

【0134】

ステップS1903によって、プリンタは、特定されたプリンタドライバを含んだデバイス情報を作成する(ステップS1904)。そして、そのデバイス情報をホストコンピュータへ送信する(ステップS1905)。

【0135】

ステップS1903の処理を具体例を用いて説明する。例えば、プリンタAが、GDIプリンタドライバAによって印刷データがクロスパスを介して生成されていると判定した場合、適切なプリンタドライバとしてXPSプリンタドライバAを特定する。そして、特定されたXPSプリンタドライバAを含むデバイス情報をホストコンピュータへ送信するため、ユーザはプリンタから推奨された適切なドライバに切り替えることができる。その結果、印刷品質の劣化、または、不正な印刷を極力防止することが可能となる。

【0136】

尚、プリンタは、デバイス探索メッセージ(図12の1202)に含まれるメッセージ送信元情報に対してS1904にて作成したデバイス情報を通知する。

【0137】

次に、プリンタ1500が印刷ジョブ受信時に、ジョブが生成された経路を識別する識別方法について説明する。

【0138】

この識別方法を説明するにあたり、まず、XPSプリンタドライバ213の構成について、図16Aを用いて説明する。

【0139】

図16Aは本発明の実施形態のXPSプリンタドライバの構成を示す図である。

【0140】

図2に示す環境において、GDIプリントパスではDEVMODE構造体と呼ばれるデータ構造で印刷設定データを持っている。一方、XPSプリントパスではPrintTicketと呼ばれるXML形式のデータ構造になる。ここで、GDIからXPSもしくはXPSからGDIに変換する際にDEVMODEからPrintTicketへ、PrintTicketからDEVMODEに変換する処理が必要となる。

【0141】

そして、描画データ同様に、DEVMODEで表現できる範囲とPrintTicketで表現できる範囲も異なる。そこで、この処理は、オペレーティングシステムが自動で行うのではなく、プリンタドライバ自身が拡張されたアーキテクチャを用いて行なっている。

【0142】

XPSプリンタドライバ213は、PrintTicketプロバイダ213a、UIドライバモジュール(XPS)213bを保持している。UIドライバモジュール(XPS)213bは、DEVMODEベースであり、アプリケーションから呼び出されることによって印刷設定情報を設定するためのUIを表示する。また、PrintTicketプロバイダ213aは、PrintTicket(プリントチケット)からDEVMODE、または、DEVMODEからPrintTicketへの変換処理を行う。

【0143】

10

20

30

40

50

図16Aでは、Win32アプリケーション201が、印刷設定を行うためにDEVMODEを要求する場合、XPSプリンタドライバ213は、DEVMODEベースのため変換処理の必要が生じない。よって、Win32アプリケーション201が、XPSプリンタドライバ213を用いて印刷処理を行う場合、プリントチケットプロバイダ213aは、変換処理を行う必要がない。

【0144】

一方、WinFxアプリケーション211は、印刷設定を行うためプリントチケットを要求する。XPSプリンタドライバ213は、プリントチケットを保持していないため、プリントチケットプロバイダ213aは、プリントチケットからDEVMODEへの変換処理を実行する。

10

【0145】

ここで、XPSプリンタドライバ213は、プリントチケットプロバイダ213aの変換処理を認識することができる。そのため、印刷対象のデータを受け付けた時に、プリントチケットプロバイダ213aによる、プリントチケットからDEVMODEへの変換処理が行われた場合には、印刷設定の要求元がWinFxアプリケーション211であると判定することができる。一方、印刷対象のデータを受け付けた時に、プリントチケットプロバイダ213aによる変換処理が行われていない場合には、印刷設定の要求元をWin32アプリケーション201と判定することができる。

【0146】

尚、GDIプリンタドライバ203の構造は、XPSプリンタドライバ213と同様であるが、UIドライバモジュール(XPS)213bに相当するものとして、UIドライバモジュール(GDI)が構成されることになる。

20

【0147】

そして、GDIプリンタドライバ203と、XPSプリンタドライバ213と同様に、プリントチケットプロバイダ213aの変換処理を認識することができる。そのため、印刷対象のデータの処理経路及び、印刷設定の要求元のアプリケーションを判定することができる。

【0148】

図3において、WinFxアプリケーション211からGDIプリンタドライバ203に渡される印刷ジョブの内、描画データに関してはOS102によってXPS GDI変換モジュール312が自動で変換を行う。そのため、GDIプリンタドライバ203は変換が行われたことを検知することができない。

30

【0149】

一方、ステイブル処理等の印刷設定データに対しては、OS102が自動で変換を行うことができない。そのため、WinFxアプリケーション211から渡されるPrintTicketを、GDIプリンタドライバ203の理解できる表現であるDEVMODEに変換する作業をXPSプリンタドライバ213のUIドライバモジュール213bが行う。

【0150】

まず、OS102は描画データを自動で変換する際に、印刷設定の変換をGDIプリンタドライバ203に依頼するためにPrintTicketをGDIプリンタドライバ213が有するUIドライバモジュール213bに対して引き渡す。OS102からPrintTicketを取得したUIドライバモジュール213bは与えられたXML形式のPrintTicket情報をDEVMODEに変換する。この際に、UIドライバモジュール213bがXPS GDI変換モジュール312の変換が行われたということを示す変換フラグをDEVMODEの内部に設定する。

40

【0151】

この処理結果の具体例について、図16Bを用いて説明する。

【0152】

図16Bは本発明の実施形態のPrintTicketからDEVMODEへの変換の

50

際に変換が行われたことを示す変換フラグを立てる様子を説明するための図である。

【0153】

1601は、OSからXPSプリンタドライバ213のUIドライバモジュール213bに渡されるPrintTicketを示している。PrintTicket1601は、XML形式で表現され、用紙サイズや両面印刷の有無等の印刷設定に関わる情報を保持している。

【0154】

一方、PrintTicket1601の変換後の情報が、DEVMODE1602である。DEVMODE1602は、GDIプリンタドライバ203で使用される印刷設定であり、PrintTicket1601と同様に用紙サイズや両面印刷の有無等が示されている。
10

【0155】

XPS GDI変換モジュール312の変換フラグが、dmConvertFlag1603である。GDIプリンタドライバ変換時にDM_CNV_PTTODEVMODEを指定することで、プリントチケットプロバイダは、XPS GDI変換が行われたものであることを記録している。尚、XPS GDI変換の行われていないDEVMODEは、DM_CNV_NONEが指定されることになる。

【0156】

次に、GDIプリンタドライバが、DEVMODE及び描画データから生成する印刷データのフォーマットについて、図17を用いて説明する。
20

【0157】

図17は本発明の実施形態のGDIプリンタドライバが生成する印刷データのフォーマットを示す図である。

【0158】

印刷データは、ジョブ印刷設定部1701及び描画データ部1702から構成されている。ジョブ印刷設定部1701には、部数や両面印刷等の印刷設定に関する情報に加え、XPS GDI変換の有無（あり／なし）を示す変換フラグ、及び自身のドライバタイプ（GDIあるいはXPS）情報を含んでいる。プリンタ1500は、この印刷データ（例えば、PDLデータ）を解析して、変換フラグ及びドライバタイプ情報を抽出することにより、この印刷ジョブが生成された印刷処理フロー（印刷ジョブ生成経路）を判別することができる。
30

【0159】

GDIプリンタドライバは、プリントチケットからDevmodeへの変換が起こっていると判定した場合、印刷要求元はWinFXアプリケーションと判定できる。よって、GDIプリンタドライバは、図17に示すようにXPS to GDI変換あり、及び、自身のドライバタイプGDIと記述した印刷データを生成する。一方、プリントチケットからDevmodeへの変換がないと判定した場合、印刷要求元はWin32アプリケーションと判定できる。この場合、GDIプリンタドライバは、XPS to GDI変換なし、及び、自身のドライバタイプGDIと記述した印刷データを生成する。

【0160】

一方、XPSプリンタドライバが生成する印刷データについても説明する。
40

上述したようにXPSプリンタドライバは、WinFXアプリケーションからの印刷要求を受信した場合、印刷機能の低下や描画結果を劣化させることなく正常に印刷できる。図16Aにて上述した通り、XPSプリンタドライバは、プリントチケットからDevmodeへの変換が起こっていると判定した場合、印刷要求元はWinFXアプリケーションと判定できる。

【0161】

この場合、印刷データは、GDI XPS変換なし、及び、自身のドライバタイプXPSと記述した印刷データを生成する。一方、XPSプリンタドライバは、プリントチケットからDevmodeへの変換がないと判定した場合、印刷要求元はWin32アプリケ
50

ーションと判定できる。この場合、印刷データは、G D I X P S 変換あり、及び、自身のドライバタイプX P Sと記述した印刷データを生成する。

【 0 1 6 2 】

このように、プリンタが受信する印刷データには、描画データの変換処理の有無を示す情報と、指定されているプリンタドライバのドライバタイプが含まれている。よって、ジョブ解析部502は、受信した印刷データに基づいて印刷データの生成に使用されたアプリケーションとプリンタドライバの組み合わせを解析する。

【 0 1 6 3 】

次に、図8のステップS802の処理の詳細について、図18を用いて説明する。

【 0 1 6 4 】

図18は本発明の実施形態のステップS802の処理の詳細を示すフローチャートである。

10

【 0 1 6 5 】

ここでは、特に、プリンタ1500が、受信した印刷ジョブのジョブ生成経路（印刷処理フロー）を判別する処理について説明する。

【 0 1 6 6 】

ステップS1801で、ジョブ解析部502は、印刷データ中のジョブ印刷設定部1701に含まれるドライバタイプを識別する。図17に示すとおり、プリンタが受信する印刷データにはドライバタイプが記述されている。ドライバタイプがX P Sである場合、ステップS1802に進む。

20

【 0 1 6 7 】

ステップS1802において、ジョブ解析部503は、変換フラグが立っているか否かを判定する。ここで、変換フラグが立っている場合（ステップS1802でY E S）、つまり、「G D I X P S 変換があり」と判定される場合、ステップS1807に進み、印刷処理フロー（4）であると判定する。一方、変換フラグが立っていない場合（ステップS1802でN O）、ステップS1806に進み、印刷処理フロー（2）であると判定する。

【 0 1 6 8 】

一方、ドライバタイプがG D Iである場合、ステップS1803に進み、ジョブ解析部503は、変換フラグが立っているか否かを判定する。ここで、変換フラグが立っている場合（ステップS1803でY E S）、つまり、「X P S G D I 変換があり」と判定される場合、ステップS1805に進み、印刷処理フロー（3）であると判定する。

30

【 0 1 6 9 】

一方、変換フラグが立っていない場合（ステップS1803でN O）、ステップS1804に進み、印刷処理フロー（1）であると判定する。このように、プリンタは印刷データに含まれるドライバタイプと描画データの変換フラグに基づいて、印刷処理の要求もとのアプリケーションを特定できる。その特定の結果、プリンタが、ストレートパスを介して生成された印刷データでないと解析した場合、印刷データを禁止するように制御する。

【 0 1 7 0 】

尚、本実施形態においては、X P S G D I 変換により失われる情報として、電子署名情報を使用したセキュリティ設定を代表例としてあげて説明している。しかしながら、本発明は、セキュリティのための印刷制限だけでなく、印刷品質に対する課題に対しても、同様の構成で対応させることが可能である。

40

【 0 1 7 1 】

以上説明したように、本発明によれば、プリンタにおいて、印刷ジョブが生成された経路情報を識別して、その処理内容を適応的に変更することが可能である。そのため、例えば、図7にてG D I 経由での印刷を禁止（702がO N）であり、かつ、W i n 3 2からの印刷を許可（703がO N）の場合、印刷処理フロー（3）であると識別された時点で印刷処理がキャンセル（禁止）される。

【 0 1 7 2 】

50

また、図7においてセキュリティ設定を有効（701がON）であり、かつ、703及び704がOFFの場合、プリンタは、印刷処理フロー（2）を経由した印刷処理をキャンセルしても良い。この場合、ジョブ解析部は、受信した印刷データからドライバタイプがXPSであり、かつ、GDI to XPS変換ありと解析した場合に、印刷処理フロー（2）と判定できるので、印刷処理をキャンセルする。

【0173】

このように、プリンタは、受信した印刷データに基づいて、Win32アプリケーションとGDIプリンタドライバの組み合わせによる処理経路を介して生成された印刷データであるかを解析できる。また、プリンタは、受信した印刷データに基づいて、WinFXアプリケーションとXPSプリンタドライバの組み合わせによる処理経路を介して生成された印刷データであるかを解析できる。10

【0174】

つまり、セキュリティが確保されない恐れがある印刷ジョブや印刷品質が悪くなる可能性がある印刷ジョブについては、プリンタとして、その印刷ジョブの印刷を禁止することができる。これにより、セキュリティ制限を回避する印刷を防止したり、印刷品質が低いがためにユーザが望まない結果となってしまい用紙を無駄にしてしまうことを防止する印刷システムを構成することができる。

【0175】

以上、実施形態例を詳述したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記憶媒体等としての実施態様をとることが可能である。具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。20

【0176】

尚、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラム（実施形態では図に示すフローチャートに対応したプログラム）を、システムあるいは装置に直接あるいは遠隔から供給する。そして、そのシステムあるいは装置のコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される場合を含む。

【0177】

従って、本発明の機能処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。30

【0178】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等の形態であっても良い。

【0179】

プログラムを供給するための記録媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスクがある。また、更に、記録媒体としては、光磁気ディスク、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモリカード、ROM、DVD（DVD-ROM、DVD-R）などがある。

【0180】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続する。そして、その接続先のホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明に含まれるものである。40

【0181】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに50

配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせる。そして、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【0182】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される。また、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現され得る。

【0183】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれる。その後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現される。

【図面の簡単な説明】

【0184】

【図1】一般的な印刷処理フロー及び印刷処理に対するコンポーネントを示す図である。

【図2】複数のグラフィクスエンジンが共存する、Windows(登録商標) Vistaにおける印刷処理フロー及び印刷処理に対するコンポーネントを示す図である。

【図3】Windows(登録商標)Vistaにおける、2つのグラフィクスエンジン共存の仕組みを示す図である。

【図4】本発明の実施形態のコンピュータを含むプリンタ制御システムの構成を説明するブロック図である。

【図5】本発明の実施形態のプリンタの機能構成を示すブロック図である。

【図6】本発明の実施形態のホストコンピュータ上とプリンタ間のデータ処理例を示す図である。

【図7】本発明の実施形態のモード設定画面の一例を示す図である。

【図8】本発明の実施形態のプリンタが実行する処理を示すフローチャートである。

【図9】本発明の実施形態のサブルーチンSRの処理の詳細を示すフローチャートである。

【図10】本発明の実施形態のプリンタが管理者に送信する電子メールの一例を示す図である。

【図11】本発明の実施形態のプリンタがホストコンピュータに送信する通知メッセージの一例を示す図である。

【図12】本発明の実施形態のGDIプリンタドライバのアンインストールのシーケンスを示す図である。

【図13】本発明の実施形態のプリンタから送信されるデバイス情報の一例を示す図である。

【図14】本発明の実施形態のプリンタから送信されるサービス情報の一例を示す図である。

【図15】本発明の実施形態のデバイス情報要求を受信したプリンタがデバイス情報を生成する際の処理を示すフローチャートである。

【図16A】本発明の実施形態1のXPSプリンタドライバの構成を示す図である。

【図16B】本発明の実施形態のPrintTicketからDEVMODEへの変換の際に変換が行われたことを示す変換フラグを立てる様子を説明するための図である。

【図17】本発明の実施形態のGDIプリンタドライバが生成する印刷データのフォーマットを示す図である。

【図18】本発明の実施形態のステップS802の処理の詳細を示すフローチャートである。

10

20

30

40

50

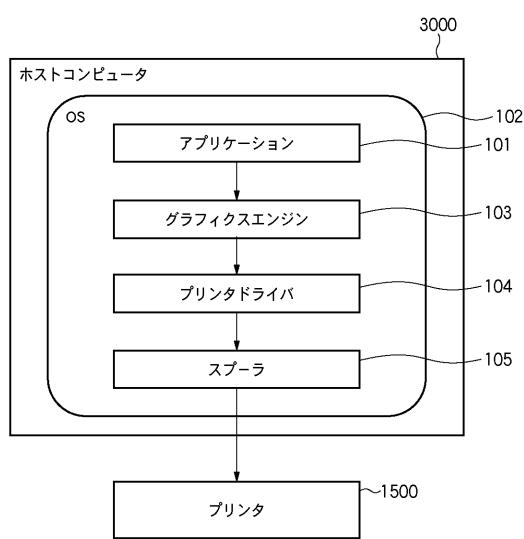
【図19】本発明の実施形態のプリンタの処理を示すフローチャートである。

【符号の説明】

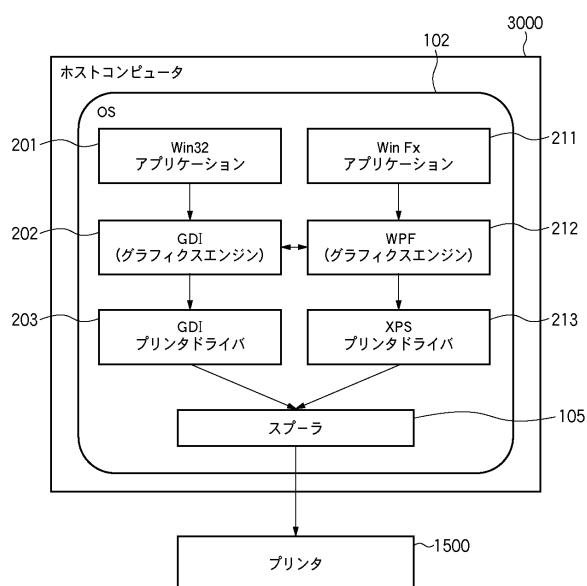
【0185】

- 501 ジョブ受信部
- 502 ジョブ解析部
- 503 モード設定部
- 504 印刷部
- 505 ジョブ削除部
- 506 通知部

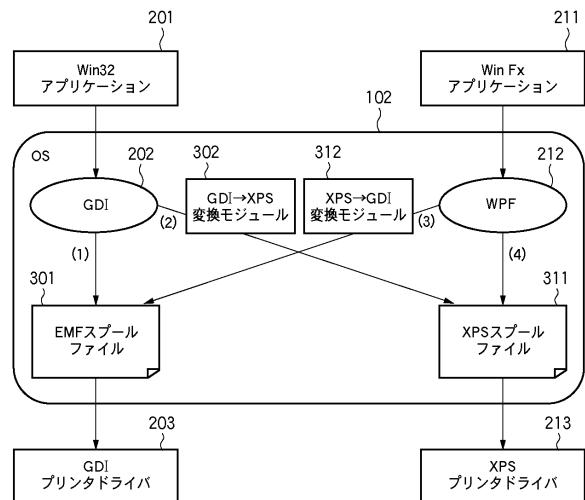
【図1】



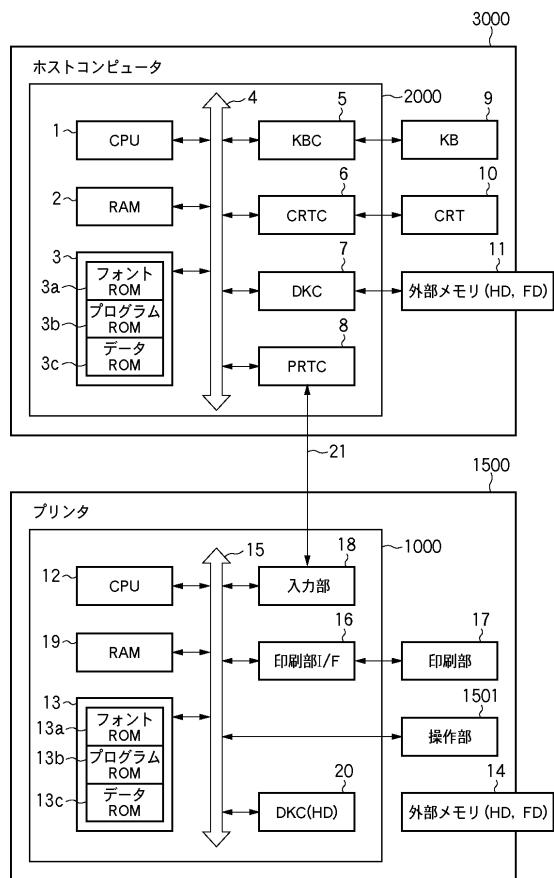
【図2】



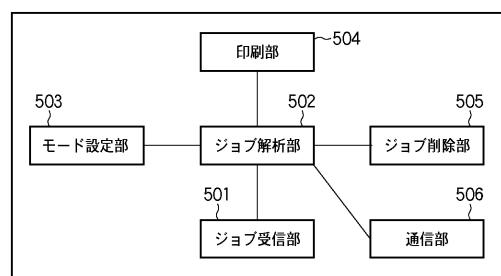
【図3】



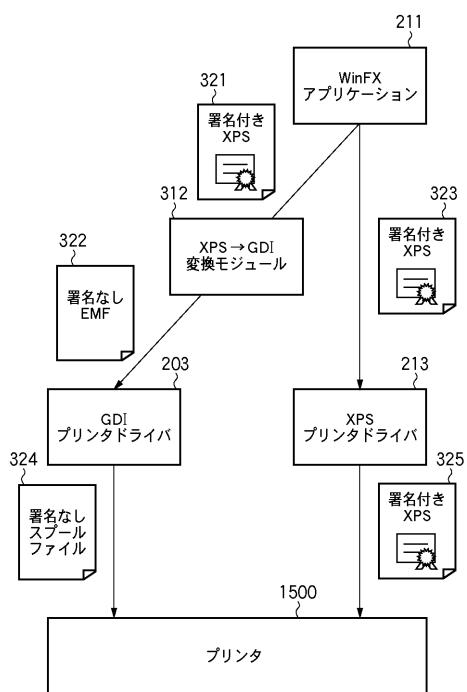
【図4】



【図5】



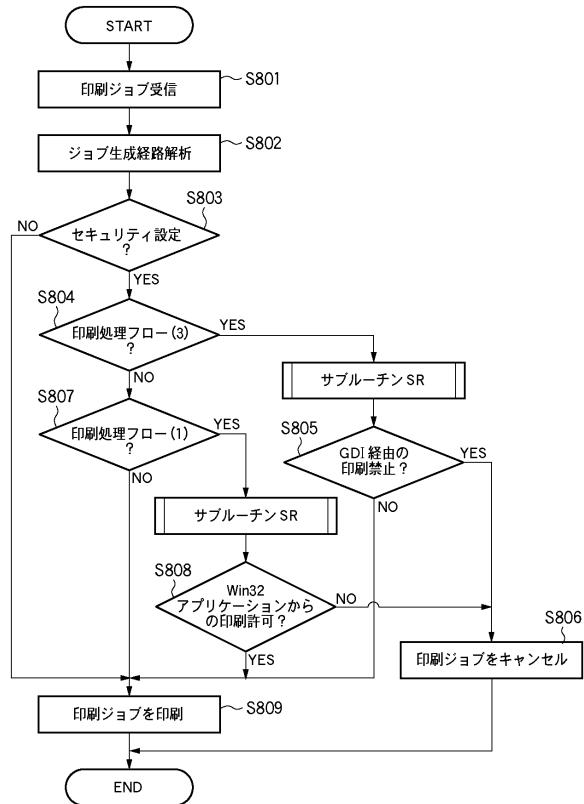
【図6】



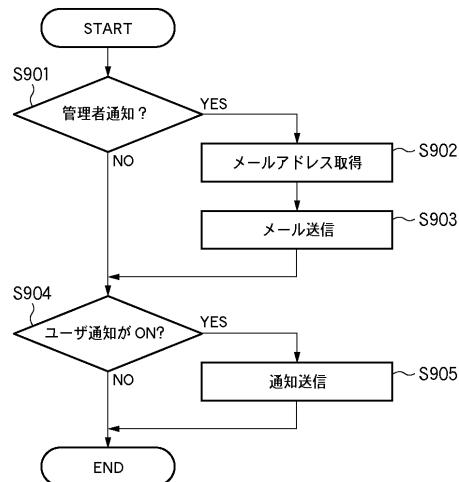
【図7】



【図8】



【図9】



【図10】

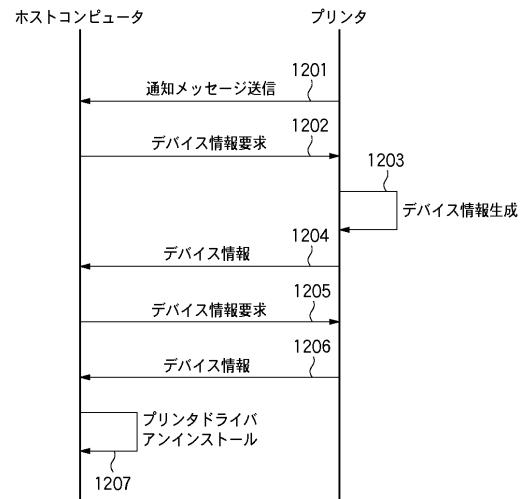
Subject : Driver Warning
 To : Administrator
 From : Printer1
 Undesirable driver is used for Printer1
 User : user1
 IP address : X.X.X.X
 Date : 2005/12/31

【図11】

```

<d>Hello>
<a:EndpointReference>
<a:Address>uuid:98190dc2-0890-4ef8-1234567890ab</a:Address>
</a:EndpointReference>
<d:MetadataVersion>75965</d:MetadataVersion>
</d>Hello>
    }
    1101
  
```

【図12】



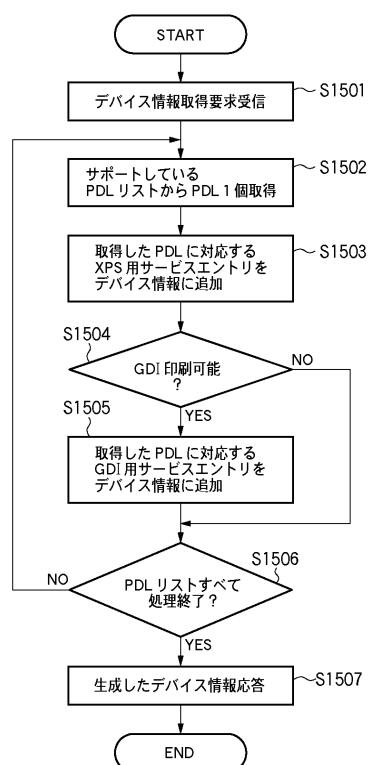
【図13】

```
<Metadata>
  <Service>
    <EndpointReference>
      <Address>http://192.168.0.1/print</Address>
    </EndpointReference>
    <Types>PrintService</Types>
    <ServiceId>1</ServiceId>
  </Service>
  <Service>
    ...
  </Service>
</Metadata>
```

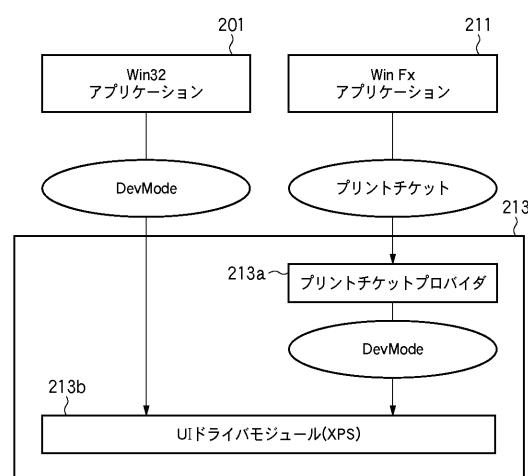
【図14】

```
<Metadata>
  <PrinterMetadata>
    <PrinterName>Printer in Copy_Room</PrinterName>
    <DeviceID>CLSPRINTER[MFG:ABC:MDL:BP-XXX PDL1(XPS):CMD:PDL1,IEEE1284]</DeviceID>
  </PrinterMetadata>
</Metadata>
```

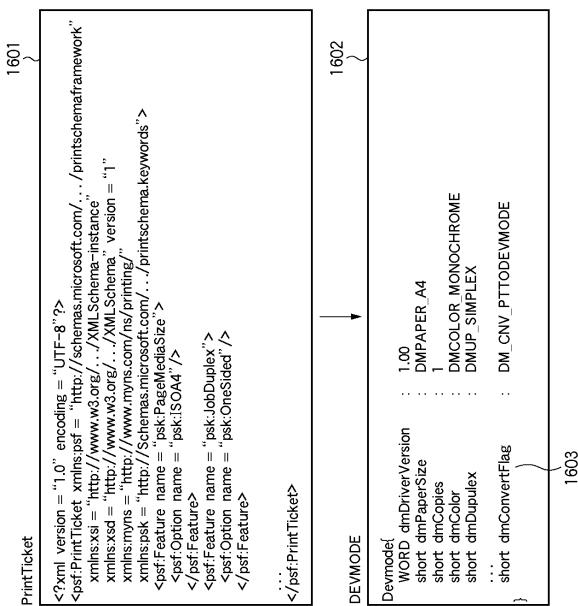
【図15】



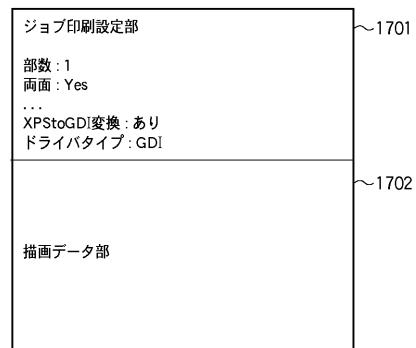
【図16A】



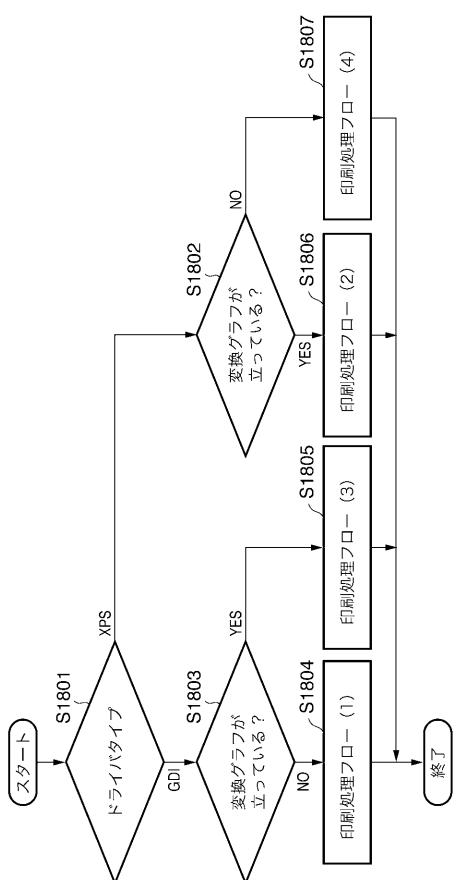
【図16B】



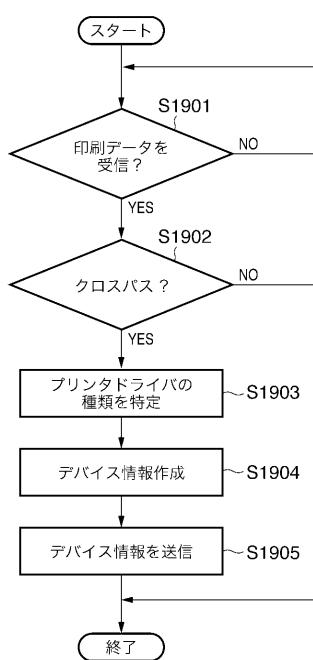
【図17】



【図18】



【図19】



フロントページの続き

審査官 緑川 隆

(56)参考文献 特開2003-263286(JP,A)

特開2003-108331(JP,A)

特開2001-184194(JP,A)

特開2006-285870(JP,A)

特開2006-239946(JP,A)

特開2007-249854(JP,A)

特開2007-249855(JP,A)

特開2007-249856(JP,A)

特開2007-249857(JP,A)

特開2007-249860(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 3 / 12

B 41 J 29 / 38