

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4963663号

(P4963663)

(45) 発行日 平成24年6月27日(2012.6.27)

(24) 登録日 平成24年4月6日(2012.4.6)

(51) Int.Cl.

G06F 3/12 (2006.01)

F I

G06F 3/12

C

請求項の数 14 (全 37 頁)

(21) 出願番号 特願2007-277906 (P2007-277906)
(22) 出願日 平成19年10月25日(2007.10.25)
(65) 公開番号 特開2009-104541 (P2009-104541A)
(43) 公開日 平成21年5月14日(2009.5.14)
審査請求日 平成22年10月18日(2010.10.18)

(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 100145827
弁理士 水垣 親房
(72) 発明者 加藤 央
東京都大田区下丸子3丁目30番2号 キ
ヤノン株式会社内

審査官 衣川 裕史

最終頁に続く

(54) 【発明の名称】 情報処理装置、情報処理方法、記憶媒体、プログラム

(57) 【特許請求の範囲】

【請求項1】

プリンタドライバよりプリンタの能力情報を取得する第1の取得手段と、
前記第1の取得手段より取得したプリンタの能力情報に基づき機能に対応した文字列情
報を取得する第2の取得手段と、

プリンタドライバの画面に表示される機能の文字を前記第2の取得手段により取得され
た文字列情報の文字で表示させるための指示を行う指示手段と、
を有することを特徴とする情報処理装置。

【請求項2】

プリンタの能力情報をアプリケーションに出力する出力手段と、
前記出力手段によりアプリケーションに出力されたプリンタの能力情報に基づき取得さ
れた機能に対応した文字列情報でプリンタドライバの画面に表示される機能の文字を表示
させるためのアプリケーションからの指示を受信する受信手段と、

前記受信手段により受信された指示に基づき画面に表示される機能の文字を表示させる
表示手段と、
を有することを特徴とする情報処理装置。

【請求項3】

前記表示手段は、あらかじめ設定される機能を表示する文字列の最大表示文字数を超え
る文字列を表示しないことを特徴とする請求項2記載の情報処理装置。

【請求項4】

10

20

前記表示手段で表示される機能を示す文字列がユーザの操作で指示されることに応じて、指示されている文字列の文字数が最大表示文字数を超えているかどうかを判断する判断手段を備え、

前記表示手段は、前記判断手段が最大表示文字数を超えていると判断した場合、表示されていない文字列を含む全ての文字列を表示することを特徴とする請求項 3 記載の情報処理装置。

【請求項 5】

前記表示手段は、前記最大表示文字数を超えて表示されない文字列と、前記全ての文字列との双方を表示することを特徴とする請求項 4 記載の情報処理装置。

【請求項 6】

前記表示手段で表示される機能を示す文字列がユーザの操作で指示されることに応じて、プリンタドライバの画面に表示される機能の文字が置き換えられているかどうかを判別する判別手段と、

前記判別手段がプリンタドライバの画面に表示される機能の文字が置き換えられていると判断した場合、

前記表示手段は、前記受信手段により受信された指示に基づき画面に表示される機能の文字と、プリンタドライバの画面に表示される機能の文字との双方を表示することを特徴とする請求項 2 記載の情報処理装置。

【請求項 7】

プリンタドライバよりプリンタの能力情報を取得する第 1 の取得工程と、

前記第 1 の取得工程より取得したプリンタの能力情報に基づき機能に対応した文字列情報を取得する第 2 の取得工程と、

プリンタドライバの画面に表示される機能の文字を前記第 2 の取得工程により取得された文字列情報の文字で表示させるための指示を行う指示工程と、
を有することを特徴とする情報処理方法。

【請求項 8】

プリンタの能力情報をアプリケーションに出力する出力工程と、

前記出力工程によりアプリケーションに出力されたプリンタの能力情報に基づき取得された機能に対応した文字列情報でプリンタドライバの画面に表示される機能の文字を表示させるためのアプリケーションからの指示を受信する受信工程と、

前記受信工程により受信された指示に基づき画面に表示される機能の文字を表示させる表示工程と、
を有することを特徴とする情報処理方法。

【請求項 9】

前記表示工程は、あらかじめ設定される機能を表示する文字列の最大表示文字数を超える文字列を表示しないことを特徴とする請求項 8 記載の情報処理方法。

【請求項 10】

前記表示工程で表示される機能を示す文字列がユーザの操作で指示されることに応じて、指示されている文字列の文字数が最大表示文字数を超えているかどうかを判断する判断工程を備え、

前記表示工程は、前記判断工程が最大表示文字数を超えていると判断した場合、表示されていない文字列を含む全ての文字列を表示することを特徴とする請求項 9 記載の情報処理方法。

【請求項 11】

前記表示工程は、前記最大表示文字数を超えて表示されない文字列と、前記全ての文字列との双方を表示することを特徴とする請求項 10 記載の情報処理方法。

【請求項 12】

前記表示工程で表示される機能を示す文字列がユーザの操作で指示されることに応じて、プリンタドライバの画面に表示される機能の文字が置き換えられているかどうかを判別する判別工程と、

前記判別工程がプリンタドライバの画面に表示される機能の文字が置き換えられていると判断した場合、

前記表示工程は、前記受信工程により受信された指示に基づき画面に表示される機能の文字と、プリンタドライバの画面に表示される機能の文字との双方を表示することを特徴とする請求項 8 記載の情報処理方法。

【請求項 13】

請求項 7 乃至 12 のいずれか 1 項に記載の情報処理方法をコンピュータに実行させるためのプログラムを格納したことを特徴とするコンピュータが読み取り可能な記憶媒体。

【請求項 14】

請求項 7 乃至 12 のいずれか 1 項に記載の情報処理方法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。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、印刷設定を行うためのユーザインタフェースを制御する情報処理装置の情報処理に関するものである。

【背景技術】

【0002】

Microsoft 社のオペレーティングシステム (OS) である Microsoft Windows (登録商標) Vista において印刷システムが大きく変更された。

【0003】

従来の GDI (Graphic Device Interface) ベースの印刷システムと置換可能な、XPS (XML Paper Specification) ベースの新しい印刷システムが導入された (下記非特許文献 1)。

【0004】

XPS とは、電子文書フォーマットの一つであり、その中身は、文書構造、描画情報、印刷設定などの情報を保持する XML とビットマップやフォントなどのバイナリリソースといった様々なパーツから構成される。

【0005】

XPS を構成する特に重要なパーツを図 28 に示す。

【0006】

図 28 は、情報処理装置で処理可能な電子文書フォーマットの構造を説明する図である。

【0007】

図 28 において、1 つの電子文書は、ジョブに対応すべく FixedDocumentSequence パーツを必ず存在し、ドキュメントに対応すべく FixedDocument 1 ~ DocumentN パーツを含み、さらに、各ドキュメントに対応すべく FixedDocument 1 ~ DocumentN パーツは、FixedPage 1 ~ FixedPageN パーツを含む構成となっている例である。

【0008】

さらに、これら任意の FixedDocumentSequence パーツ、FixedDocument パーツ、FixedPage パーツには、図 29 に示すように、各々印刷設定を関連付けることができる。

【0009】

図 29 は、図 28 に示す電子文書の各パーツに関連づけ可能な印刷設定の一例を示す図である。

【0010】

図 29 に示すように、電子文書の各パーツに関連づけ可能な印刷設定をプリントチケットと呼ぶ。

【0011】

ここで、FixedDocumentSequence パーツに関連付けた印刷設定をジョブレベルのプリントチケットと呼ぶ。また、FixedDocument パーツに関連付けた印刷設定をドキュメントレ

10

20

30

40

50

ベルのプリントチケットと呼ぶ。さらに、FixedPageパーツに関連付けた印刷設定をページレベルのプリントチケットと呼ぶ。

【0012】

なお、ジョブレベルのプリントチケットはジョブの印刷設定を行う。ドキュメントレベルのプリントチケットはドキュメントの印刷設定を行う。ページレベルのプリントチケットはページの印刷設定を行う。

【0013】

次に、従来の印刷設定形式とプリンタドライバのインタフェースについて説明する。

【0014】

従来のMicrosoft Windows（登録商標）の印刷システムにおいて、Win32アプリケーションが使用するDEVMODEと呼ばれるメモリブロックが印刷設定データとして重要な役割を担ってきた。

【0015】

図30は、従来の印刷システムにおけるメモリブロックを説明する図である。本例は、Win32アプリケーションが使用するDEVMODE全体像を簡略化して示したものである。

【0016】

DEVMODE構造体の特徴は、仕様が公開されているパブリック部301と仕様が公開されないプリンタドライバのみアクセス可能なプライベート部302の2つの領域から構成されることにある。

【0017】

一般に、DEVMODE構造体のパブリック部301は、用紙、用紙の向き、解像度などのごく基本の部分を記述するのみで、大部分を占める残りの全てをプライベート部302に記述する。

【0018】

図31は、図30に示したDEVMODEパブリック部301の一例を示す図である。

【0019】

図31に示すように、DEVMODE構造体はMicrosoft社によってC言語のヘッダファイルとして厳密に規定されている。

【0020】

DEVMODEパブリック部301には、スコープの概念が無いことわかる。一方、プリンタドライバを開発する各社が独自に定義するプライベート部302は、この構造体の直後に続き、その拡張サイズはdmDriverExtraで指定される。

【0021】

Microsoft Windows（登録商標）Vista から導入されたプリントチケットは、このようなレガシーな印刷設定フォーマットに替わる新しいフォーマットである。

【0022】

以上をまとめると、従来形式であるDEVMODEは、（1）～（8）に示す特徴を備える。

【0023】

（1）構造体で記述される。大部分を占めるプライベート部の仕様は非オープン。

【0024】

（2）バイナリなので可読性なし。

【0025】

（3）プリンタドライバのみアクセス可能。

【0026】

（4）各印刷設定にスコープの概念が無い。

【0027】

という次第であるのに対し、新形式であるプリントチケットは、XMLで記述される。

【0028】

（5）仕様はオープン。

【0029】

10

20

30

40

50

(6) テキストなので可読性あり。

【0030】

(7) プリンタドライバ以外にアプリケーションなども自由にアクセス可能。

【0031】

(8) 各印刷設定にスコープを指定する。

【0032】

このようにプリントチケットはDEVMODEに比べ大きく機能拡張されている。

【0033】

従来印刷システムでは、アプリケーションがデバイスの能力情報を取得するために、DeviceCapabilities APIが用意されていた。

10

【0034】

アプリケーションは、DeviceCapabilities APIを用いて、デバイスのサポート用紙サイズ情報や両面印刷の能力情報などを取得し、アプリケーションの印刷画面を構築していた。

【0035】

DeviceCapabilities API では、Windows（登録商標）システムで定義された特定の機能の情報は取得できるが、それ以外の能力情報が取得できないという課題があった。

【0036】

これに対して、Windows（登録商標）Vistaでは、DeviceCapabilitiesに代わる仕組みとして、PrintCapabilitiesと呼ばれるXML形式のデバイス能力情報を導入している。

20

【0037】

図32は、情報処理装置にインストールされるプリンタドライバがアプリケーションに返すPrintCapabilitiesの一例を示す図である。

【0038】

図32において、1つのPrintCapabilitiesタグ1301内に、機能を示すFeatureタグ1302が複数存在する。さらにFeatureタグ1302内の複数のOptionタグ1303が機能の設定項目候補となる。

【0039】

PrintCapabilitiesはデバイスでサポートしている機能とそれらの設定項目のすべてを列挙する。アプリケーションは、このPrintCapabilitiesを元に各機能の設定を選択し、印刷設定としてプリントチケットを作成する。

30

【0040】

また、PrintCapabilitiesでは、機能の一覧とその選択肢に加え、DisplayNameプロパティを記述する仕様がある。DisplayNameプロパティには、それぞれの機能と設定項目に対する表示用の文字列を格納する。アプリケーションでは、プリンタドライバから取得したPrintCapabilities内のDisplayNameプロパティを参照し、印刷設定画面のUI文字列として表示することができる。

【0041】

DeviceCapabilities APIに比べ、PrintCapabilitiesでは、システムの規定の印刷機能以外の能力およびUI表示用の文字列が取得できるので、より高度な印刷設定画面を構成することができる。

40

【0042】

図33は、情報処理装置における印刷制御モジュールの構造を説明するブロック図である。本例は、印刷設定の処理において、アプリケーション層からOS層、そしてドライバ層に至る各階層でDEVMODEやプリントチケットがどのようにやり取りされるかを示した図である。

【0043】

まず、左上にあるWin32アプリケーション504がWin32API503を介して、プリンタドライバ501とDEVMODEをやり取りする。また、デバイスの能力情報をWin32API503 APIを通じてDeviceCapabilitiesを取得している。

50

【0044】

一方、右上にあるフローは、プリントチケット対応アプリケーション506がプリンタドライバ501とプリントチケット、PrintCapabilitiesをやり取りする処理例である。

【0045】

プリントチケット対応アプリケーション506は、プリントチケットAPI505を通じて能力情報をPrintCapabilitiesで取得し、印刷設定をプリントチケットにより指定する。

【0046】

実際にプリントチケット対応アプリケーション506からプリンタデバイスに印刷する場合は、プリントチケット対応アプリケーション506とプリンタドライバのそれぞれの印刷設定画面を操作して、ユーザは最終的な印刷設定を確定させる。

10

【0047】

このとき、プリントチケット対応アプリケーション506の印刷設定UIでは、プリンタドライバのUIの内容と同じ機能を選択させる場合がある。

【0048】

このようなアプリケーションとプリンタドライバの重複する機能をユーザに通知する技術として、下記特許文献1に記載のような技術がある。

【0049】

特許文献1では、支援ソフトを使ってプリンタドライバとアプリケーションUIを同時に表示させることでユーザに重複した機能を通知している。

20

【特許文献1】特開平11-143664号公報

【非特許文献1】<http://www.microsoft.com/japan/whdc/device/print/default.mspx>

【発明の開示】

【発明が解決しようとする課題】

【0050】

情報処理装置にインストールされているアプリケーションからプリンタドライバのUIを情報処理装置に接続される表示装置に表示すると、たとえ同じ機能を示すものであっても異なる文字列で機能名が表示されることがある。このため、ユーザにとって操作ミスを誘発する要因となっている。

【0051】

30

例えば、複数部数印刷する場合に、1部毎に出力する機能を示す用語が、「部単位印刷」と「ソート」となっている場合である。これは一例であって、その他にも1枚の物理用紙に複数の論理ページを割り付けて印刷する機能を示す用語が、「割り付け印刷」と「1枚あたりのページ数」となっている場合などがある。

【0052】

その他の問題としては、アプリケーションの表示言語と、プリンタドライバの表示言語が異なる場合にも、対応付けがわかりづらいという課題がある。

【0053】

上記特許文献1の技術では、同時に表示させて重複した機能がわかるようになってはいるが、特別な支援ソフトを必要があり、ユーザにとって煩わしい作業が発生していた。

40

【0054】

先に述べたとおり、オペレーティングシステム、例えばWindows（登録商標）Vistaでは、PrintCapabilitiesを使ってプリンタの能力情報をアプリケーションに通知する際に、各機能のUI表示文字列を提供する仕組みがある。

【0055】

本機能を使用すると、プリンタドライバの用語でアプリケーションの用語は統一される。

【0056】

しかしながら、この仕組みをアプリケーションのUIで使用すると、印刷対象となるプリンタのドライバ毎に表示するUI文字列が異なることが考えられる。特に、出力デバイスが

50

複数のプリンタベンダーのものである場合などである。

【0057】

このようにアプリケーションで出力デバイスを切り替えると、同じ機能でも異なる文言で表示され、一つのアプリケーション内で一貫性がなくなるため、ユーザにとってわかりづらいという課題があった。このように、アプリケーションとプリンタドライバとの関係において、ユーザインタフェースを介してアプリケーションから印刷機能を設定する場合と、プリンタドライバから印刷機能を設定する場合とで、UI上に表示される文字列が異なる場合が生じる。このため、ユーザが印刷設定項目を明確に認識できないという課題があった。

【0058】

本発明は、上記の課題を解決するためになされたもので、本発明の目的は、プリンタドライバの機能で表示される文字をアプリケーションの機能で表示する文字に合わせて表示できる仕組みを提供することである。

【課題を解決するための手段】

【0059】

上記目的を達成する本発明の情報処理装置は以下に示す構成を備える。

【0060】

プリンタドライバよりプリンタの能力情報を取得する第1の取得手段と、前記第1の取得手段より取得したプリンタの能力情報に基づき機能に対応した文字列情報を取得する第2の取得手段と、プリンタドライバの画面に表示される機能の文字を前記第2の取得手段により取得された文字列情報の文字で表示させるための指示を行う指示手段と、を有することを特徴とする。

【発明の効果】

【0061】

本発明によれば、プリンタドライバの機能で表示される文字をアプリケーションの機能で表示する文字に合わせて表示できる。

【発明を実施するための最良の形態】

【0062】

次に本発明を実施するための最良の形態について図面を参照して説明する。

【0063】

<システム構成の説明>

〔第1実施形態〕

図1は、本発明の第1実施形態を示す情報処理装置、印刷装置を含むプリンタ制御システムの一例を示すブロック図である。

【0064】

なお、特に断らない限り、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN、WAN等のネットワークを介して接続が為され処理が行われるシステムであっても本発明を適用できる。

【0065】

図1において、300はホストコンピュータで、双方向インタフェース21Aを介してプリンタ150と通信可能に接続されている。

【0066】

ホストコンピュータ300において、1はCPUで、ROM3のプログラム用ROMあるいは外部メモリ11に記憶された文書処理プログラム等に基づいて図形、イメージ、文字、表（表計算等を含む）等が混在した文書処理を実行する。なお、CPU1は、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。また、外部メモリ11は、フレキシブルディスクやハードディスク等が含まれる。

【0067】

また、ROM3のプログラム用ROMあるいは外部メモリ11には、CPU1の制御プログラムであるオペレーティングシステムプログラム（以下OS）等が記憶されている。

さらに、ROM 3 のフォント用 ROM あるいは外部メモリ 11 には上記文書処理の際に使用するフォントデータ等が記憶されている。

【0068】

また、ROM 3 のデータ用 ROM あるいは外部メモリ 11 には上記文書処理等を行う際に使用する各種データが記憶されている。

【0069】

2 は RAM で、CPU 1 の主メモリ、ワークエリア等として機能する。また、RAM 2 は、オプション RAM を装着することで、メモリ容量を増量することができるように構成されている。

【0070】

5 はキーボード I / F で、キーボード 9 や不図示のポインティングデバイスからのキー入力を制御する。6 はディスプレイ I / F で、ディスプレイ 10 の表示を制御する。

【0071】

7 は外部メモリ I / F で、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、プリンタドライバ等を記憶するハードディスク (HD)、フレキシブルディスク (FD) 等の外部メモリ 11 とのアクセスを制御する。

【0072】

8 はプリンタ I / F で、所定の双方向インタフェース 21 A を介してプリンタ 150 に接続されて、プリンタ 150 との通信制御処理を実行する。

【0073】

なお、CPU 1 は、例えば RAM 2 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 (ラスターライズ) 処理を実行し、ディスプレイ 10 上での WYSIWYG を可能としている。

【0074】

また、CPU 1 は、CRT 10 上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウィンドウを開き、種々のデータ処理を実行する。

【0075】

ユーザは、印刷を実行する際、印刷の設定に関するウィンドウを開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバに対する印刷処理方法の設定を行う。

【0076】

プリンタ 150 において、12 はプリンタ CPU (CPU) で、システムバス 15 に接続されるデバイスを総括的に制御する。

【0077】

また、CPU 12 は、ROM 14 のプログラム用 ROM に記憶された制御プログラム等あるいは外部メモリ 21 に記憶された制御プログラム等に基づいてシステムバス 15 に接続される印刷部 17 に出力情報としての画像信号を出力する。

【0078】

また、この ROM 14 のプログラム ROM には、CPU 12 の制御プログラム等を記憶する。ROM 14 のフォント用 ROM には出力情報を生成する際に使用するフォントデータ等を記憶し、ROM 14 のデータ用 ROM にはハードディスク等の外部メモリ 21 がないプリンタの場合には、ホストコンピュータ 300 上で利用される情報等を記憶している。

【0079】

CPU 12 は、入力部 16 を介してホストコンピュータ 300 との通信処理が可能となっており、プリンタ 150 内の情報等をホストコンピュータ 300 に通知可能に構成されている。

【0080】

13 は RAM で、CPU 12 の主メモリ、ワークエリア等として機能し、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。

10

20

30

40

50

【0081】

なお、RAM 13は、出力情報展開領域、環境データ格納領域、NVRAM等に用いられる。前述したハードディスク（HD）、ICカード等の外部メモリ21は、外部メモリI/F18によりアクセスが制御される。

【0082】

外部メモリ21は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。17は印刷部インタフェースで、印刷部19へ画像データを出力するとともに、印刷部19の各種センサから検出される情報をCPU12に通知する。

【0083】

また、20は操作部で、操作のためのスイッチおよびLED表示器等が配されている。また、前述した外部メモリ21は1個に限らず、少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていてもよい。

【0084】

さらに、図示しないNVRAMを有し、操作パネル20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。なお、操作パネル20は、タッチパネル型の液晶ディスプレイを備え、ユーザに機能設定画面を表示可能に構成されている。

【0085】

図2は、図1に示したホストコンピュータ300のモジュール構成を説明するブロック図である。本例は、ホストコンピュータ300における典型的な印刷データ生成処理を行うモジュールに対応する。

【0086】

図2において、アプリケーション201、グラフィックエンジン202、プリンタドライバ203、およびシステムスプーラ204は、外部メモリ11に保存されたファイルとして存在する。これらのファイルは、OSやそのモジュールを利用するモジュールによってRAM2にロードされCPU1により実行されるプログラムモジュールである。

【0087】

また、アプリケーション201およびプリンタドライバ203は、外部メモリ11のFDや不図示のCD-ROM、あるいは不図示のネットワークを経由して外部メモリ11に追加することが可能となっている。

【0088】

なお、外部メモリ11は、例えばハードディスク（HD）等で構成される。

【0089】

外部メモリ11に保存されているアプリケーション201は、RAM2にロードされてCPU1により実行される。

【0090】

一方、アプリケーション201からプリンタ150に対して印刷を行う際には、同様にRAM2にロードされ実行可能となっているグラフィックエンジン202を利用して出力（描画）を行う。

【0091】

グラフィックエンジン202は、プリンタ毎に用意されたプリンタドライバ203を、外部メモリ11からRAM2にロードし、プリンタドライバ203は、アプリケーション210の出力をプリンタの制御コマンドに変換する。

【0092】

このようにして、プリンタドライバ203により変換されたプリンタ制御コマンドは、OSによってRAM2にロードされたシステムスプーラ204を経て双方向インタフェース21A経由でプリンタ150へ出力される仕組みとなっている。

【0093】

図3は、図1に示したホストコンピュータ300にインストールされるアプリケーショ

10

20

30

40

50

ンと、各プリンタ用のプリンタドライバとの関係を説明する模式図である。

【0094】

図3において、アプリケーション311は、オペレーティングシステム(OS)312が提供する各種関数(API)をコールして処理を行う。OS312は印刷に関わる処理を選択されているプリンタのプリンタドライバ313、314が提供する関数(DDI)をコールして処理を行い、アプリケーション311から選択されたプリンタへの印刷が可能となっている。なお、図中では、プリンタドライバ313とプリンタドライバ314の違いを、プリンタドライバA、プリンタドライバBと表記する。

【0095】

このときプリンタドライバ313、314とOS312、及び、アプリケーション間で、図4に示すようにプリンタの印刷設定データがやり取りされる。

10

【0096】

図4は、図1に示したホストコンピュータ300にインストールされるアプリケーションと、各プリンタ用のプリンタドライバとの関係を詳細に説明する模式図である。

【0097】

図4において、Win32アプリケーション504やプリントチケット対応アプリケーション506から印刷設定する場合については図33にて説明済みなので、ここでは説明を省略する。

【0098】

プリンタドライバ501にプリントチケットの入出力を直接管理するプリントチケット入出力部901を設ける。そして、図5に示すようなOSの関与しない独自のインタフェースを設ける。

20

【0099】

この新インタフェース対応アプリケーション902（以下、アプリケーション902と呼ぶ）は、プリントチケットAPI505を呼び出すことなく、図5に示すプリンタドライバが公開する独自のインタフェースを呼び出す。これにより、アプリケーション902から直接プリントチケットをプリントチケット入出力部901へ渡すことができる。

【0100】

図5、図6は、本実施形態を示す情報処理装置にインストールされるプリンタドライバが公開するインタフェース関数の一例を示す図である。また、本例は、プリンタドライバが公開する独自のインタフェースの一部である。

30

【0101】

図5は、プリンタドライバが公開する独自のインタフェース関数をC言語の関数として定義している例である。OSが定義するプリンタドライバインタフェース関数DrvDocumentPropertySheetsとは名前が異なるだけで関数仕様はほぼ同等なので、ここでは説明を省略する。

【0102】

なお、DrvDocumentPropertySheets関数の詳細については、例えば<http://msdn2.microsoft.com/en-us/library/ms801131.aspx>にて確認することができる。

【0103】

40

図6は、図5に示した関数で使用される構造体の一部を説明する図である。

【0104】

本例は、OSが定義するDOCUMENTPROPERTYHEADER構造体を模した仕様になっている例である。なお、DOCUMENTPROPERTYHEADER構造体の詳細については、例えば<http://msdn2.microsoft.com/en-us/library/aa506306.aspx>にて確認することができる。

【0105】

以下、DOCUMENTPROPERTYHEADER構造体とCN_DOCUMENTPROPERTYHEADER構造体の違いのみを説明する。

【0106】

DOCUMENTPROPERTYHEADER構造体には、入出力するDEVMODE構造体をメンバとして保持し

50

、さらに出力用DEVMODEのサイズもメンバとして保持しているが、これらはCN_DOCUMENTPROPERTYHEADER構造体には一切存在しない。

【0107】

その代わりに、入出力するプリントチケットをメンバ (IStream *pPrintTicket (引数)) として保持する。

【0108】

以下では、アプリケーションの印刷設定画面の表示フローと、図5、図6に示すインタフェース関数を用いて、プリンタドライバの印刷設定画面の表示フローについて説明する。

【0109】

図7Aは、本実施形態を示す情報処理装置におけるユーザインタフェースの表示制御を説明する図である。本例は、アプリケーションのユーザインタフェースで表示される印刷設定機能を示す文字列に合わせて、プリンタドライバが提供するユーザインタフェースで表示される印刷設定機能文字列を置き換える表示制御に対応する。なお、図4と同一のものには同一の符号を付してある。

【0110】

図7Aにおいて、7000AはUI文字テーブルで、機能名とそれに対応したUIの文字列テーブル情報1501を保持する。

【0111】

APUIはアプリケーションによる印刷設定画面で、アプリケーション902が文字列テーブル情報1501を参照してディスプレイ10に表示する。

【0112】

7000BはUI文字テーブルで、ドライバ側の機能名とそれに対応したUIの文字列テーブル情報1601を保持する。

【0113】

7000Cは文字列テーブルで、UI文字テーブル7000Bの文字列IDに対応する文字列データをID別に保持する。

【0114】

DRUIはドライバによる印刷設定画面で、通常は、印刷設定画面APUIで表示される機能の文字列と異なるが、後述する置き換え処理実行後のため、印刷設定画面DRUIと印刷設定画面APUIとが同じ表示としている状態に対応する。

【0115】

以下、アプリケーション902側の処理について説明する。なお、図7Aにおいて、(1)～(7)はそれぞれ処理を示す。

【0116】

まず、処理(1)では、プリンタドライバ501から図32に示すデータ構造を備えるプリンタの能力情報(以下、PrintCapabilitiesと呼ぶ)を取得する。

【0117】

次に、処理(2)で、アプリケーション902は、プリンタドライバ501から取得したPrintCapabilities内でアプリケーション902のUIで表示している機能を判別する。

【0118】

次に、処理(3)で、アプリケーション902は、UI文字テーブル7000A内に保持される文字列テーブル情報1501を参照して、処理(2)で判別された機能に対応する文字列情報を取得する。

【0119】

これにより、アプリケーション902は、図7Aに示すように、例えばプリンタドライバ501から取得したPrintCapabilities内の機能が「Collate」であれば、「部単位印刷」という文字列を文字列テーブル情報1501から取得する。

【0120】

次に、処理(4)で、後述する図10に示す処理に基づき、ドライバUIオープン時に指

10

20

30

40

50

定する図 1 1 に示すPrintTicket内に、文字列置き換え対象となる機能の記述部分に処理 (3) で取得した文字列をReplaceDisplayStringとして指定する。

【0 1 2 1】

以下、指定するPrintTicket の記述例を示す。

【0 1 2 2】

```
<feature name="Collate">
```

```
<property name="ReplaceDisplayString">
```

```
<value>部単位印刷</value>
```

```
</property>
```

```
<option name="ON"/>
```

```
</feature>
```

10

次に、プリンタドライバ 5 0 1 側の処理について説明する。

【0 1 2 3】

まず、処理 (5) で、プリンタドライバ 5 0 1 内の文字列テーブル 7 0 0 0 C を参照して、印刷設定画面 D R - U I の機能を表示するための文字列を取得する。この状態では、プリンタドライバ 5 0 1 で設定される機能に対する文字列が U I 画面に表示されるため、印刷設定画面 D R - U I と印刷設定画面 A P - U I とで機能に関わる文字列が異なる場合がある。

【0 1 2 4】

次に、プリンタドライバ 5 0 1 は、U I 文字テーブル 7 0 0 0 B に保持されるドライバ側の機能名とそれに対応した U I の文字列テーブル情報 1 6 0 1 を参照する。

20

【0 1 2 5】

次に、プリンタドライバ 5 0 1 は、アプリケーション 9 0 2 から生成されるPrintTicketで指示されるReplaceDisplayStringが指示された機能のUI文字列を、指定された文字列で置き換える。

【0 1 2 6】

そして、処理 (7) で、ドライバUIオープン時に、処理 (6) で置き換えた文字列を印刷設定画面 D R - U I として表示することで、印刷設定画面 D R - U I と印刷設定画面 A P - U I とが同じ表示とする。

【0 1 2 7】

30

なお、プリンタドライバ 5 0 1 から取得されるPrintCapabilitiesに含まれていない機能に対応する文字列については、後述するようにグレイ表示される。

【0 1 2 8】

このように本実施形態では、アプリケーション 9 0 2 がプリンタドライバ 5 0 1 に提供するUIを表示する際に、アプリケーション 9 0 2 は、プリンタドライバ 5 0 1 に指示するPrintTicket内で、自身のUIに表示している機能名称を指定する。

【0 1 2 9】

そして、プリンタドライバ 5 0 1 は、アプリケーション 9 0 2 から取得するPrintTicketで指定されたアプリケーションの機能名称に置き換えてドライバUIを構成する。

【0 1 3 0】

40

これにより、アプリケーションの U I 表示に対する表示言語と、プリンタドライバの U I 表示に対する表示言語が異なる場合にも、アプリケーション 9 0 2 に対応する機能文字列でドライバ側の機能文字列を統一することができる。

【0 1 3 1】

したがって、ユーザはプリンタドライバが提供する U I 画面の機能を示す文字列が、アプリケーション 9 0 2 が提供する U I 画面の機能を示す文字列とに違和感がなくなり、操作性が向上する。

【0 1 3 2】

図 7 B は、本実施形態を示す情報処理装置における第 1 のデータ処理手順の一例を示すフローチャートである。本例は、図 4 に示したアプリケーション 9 0 2 が自身で管理して

50

いる印刷設定画面を表示する処理例である。なお、S 1 9 0 1～S 1 9 0 8は各ステップを示す。また、各ステップは、C P U 1がR A M 2にアプリケーション9 0 2をロードして実行することで実現される。

【0 1 3 3】

アプリケーション9 0 2のプログラムは、ホストコンピュータ3 0 0の外部メモリ1 1に記憶されており、R A M 2に読み出されC P U 1によって実行される。

【0 1 3 4】

まず、S 1 9 0 1で、アプリケーション9 0 2は、プリンタデバイスの能力情報を取得する。プリンタデバイスのPrintCapabilitiesは、プリントチケットAPI 5 0 5を通じて取得する。ここで、図3 2に示すようなXML形式のPrintCapabilities をアプリケーション9 0 2に返す。なお、アプリケーション9 0 2がプリンタデバイス5 0 1のPrintCapabilitiesを取得する処理が第1の取得処理であり、アプリケーション9 0 2が後述するS 2 0 0 4で機能に対応した文字列を取得する処理が第2の取得処理に対応する。

10

【0 1 3 5】

図3 2の例に示すPrintCapabilitiesには、Collate、Duplex、MediaSize、OutputBinの4つのFeatureが含まれる。また、それぞれのFeatureには、DisplayNameプロパティが指定されている。

【0 1 3 6】

その後、S 1 9 0 2で、アプリケーション9 0 2は、図8に示すような、機能名とそれに対応したアプリケーション側のU Iの文字列テーブル情報1 5 0 1を取得する。

20

【0 1 3 7】

図8に示すように、文字列テーブル情報1 5 0 1には、機能名（Feature名）1 5 0 2に応じたUI文字列1 5 0 3、UIのコントロールID1 5 0 4が格納されている。

【0 1 3 8】

次に、S 1 9 0 3で、アプリケーション9 0 2は、S 1 9 0 2にて取得した文字列テーブル情報1 5 0 1を参照し、最初の機能の情報を取得する。

【0 1 3 9】

次に、S 1 9 0 4で、アプリケーション9 0 2は、S 1 9 0 3にて文字列テーブル情報1 5 0 1より取得した機能がS 1 9 0 1にてプリンタドライバ5 0 1より取得したPrintCapabilities内に含まれる機能か否かを判別する。ここで、アプリケーション9 0 2がPrintCapabilities内に存在すると判別された場合には、S 1 9 0 5へ進む。

30

【0 1 4 0】

そして、S 1 9 0 5で、アプリケーション9 0 2は、文字列テーブル情報1 5 0 1内に格納されたFeature名（機能名）1 5 0 2に応じた文字列を用いて、アプリケーションUIを、ホストコンピュータ3 0 0のディスプレイ1 0に表示する。ここで、アプリケーションUIとは、図7 Aに示した印刷設定画面A P－U Iに対応する。

【0 1 4 1】

次に、S 1 9 0 6で、アプリケーション9 0 2は、現在取得している機能情報が文字列テーブル情報1 5 0 1の最後の項目であるか否かを判別し、最後であると判断されたら、本処理を終了する。

40

【0 1 4 2】

一方、S 1 9 0 6で、アプリケーション9 0 2が最後の項目でないと判断された場合は、S 1 9 0 8に進み、文字列テーブル情報1 5 0 1内の次の項目を取得した後、S 1 9 0 4～S 1 9 0 6までの処理を繰り返す。

【0 1 4 3】

一方、S 1 9 0 4にて、アプリケーション9 0 2がプリンタドライバ5 0 1より取得したPrintCapabilities内に機能が存在しないと判別された場合には、S 1 9 0 7へ進む。そして、S 1 9 0 7で、機能に対応したUIコントロールをグレイ表示し、使用不可の機能として通知して、S 1 9 0 4へ戻る。

【0 1 4 4】

50

図9は、図1に示したディスプレイ10に表示されるアプリケーションの印刷設定画面1201の一例を示す図である。本画面は、アプリケーション902による表示画面例である。

【0145】

図9に示す画面では、用紙設定コントロール1202、両面設定コントロール1203、部単位印刷設定コントロール1205がそれぞれ文字列テーブル情報1501内のUI文字列で表示されている例である。

【0146】

また、PrintCapabilitiesに含まれていない機能に対応するステイプル設定コントロール1206がグレイ表示（図9では網掛けで示す）されている。

10

【0147】

アプリケーションの印刷設定画面1201には、さらにプリンタを選択するプリンタ選択コントロール1204、プリンタドライバ501の印刷設定画面を表示するためのプロパティボタン1207を備える。さらに、アプリケーションの印刷設定画面1201には、印刷設定画面を確定させるOKボタン1208、キャンセルボタン1209を含んで構成されている。

【0148】

次に、アプリケーションの印刷設定画面1201のプロパティボタン1207が押下され、アプリケーション902からプリンタドライバ501の印刷設定画面を表示する処理について説明する。

20

【0149】

図10は、本実施形態を示す情報処理装置における第2のデータ処理手順の一例を示すフローチャートである。本例は、アプリケーション902が表示した図9に示す画面において、ユーザの操作により、プロパティボタン1207が押下された際に実行される、アプリケーションからプリンタドライバの印刷設定画面を表示させる指示例である。

【0150】

なお、S2001～S2007は各ステップを示す。また、各ステップは、CPU1がRAM2にアプリケーション902をロードして実行することで実現される。

【0151】

S2001で、アプリケーション902は、印刷設定画面1201内の印刷設定に関するコントロールの選択状態を取得する。次に、S2002で、アプリケーション902は、S1901でプリンタドライバ501より取得したPrintCapabilities内のFeature一覧のうち、S2001で状態を取得したコントロールに対応したFeatureを参照する。

30

【0152】

次に、S2003で、アプリケーション902は、参照したFeatureのOptionの一覧の中で、選択状態の設定項目に対応した項目以外のOptionを削除する。その結果、アプリケーション902は、図11に示すように、Featureに特定の設定項目のOptionのみが記述された状態のPrintTicket1401を生成する。

【0153】

図11は、本実施形態を示す情報処理装置で生成されるPrintTicketの一例を示す図である。本例は、アプリケーション902によりFeature1402に特定の設定項目のOption1404のみが記述された状態のPrintTicketを生成した例である。

40

【0154】

図11の例では、"Collate"Featureでは、"On"Optionが、"Duplex"Featureでは、"OFF"Optionがそれぞれ選択されている状態に対応する。

【0155】

次に、S2004で、アプリケーション902は、文字列テーブル情報1501を取得する。そして、S2005で、アプリケーション902は、文字列テーブル情報1501を参照して、現在のコントロールに関連する機能の文字列情報を取得し、ReplaceDisplayStringプロパティとして図11に示すPrintTicket1401に指定する。

50

【0156】

図11の例では、"Collate"FeatureのReplaceDisplayStringプロパティ1403として、アプリケーションの文字列テーブル情報1501内のCollateに対応した文字列、"部単位印刷"が記述されている。

【0157】

次に、S2006で、アプリケーション902は、S2001で参照した印刷設定のコントロールが最後の設定項目か否かを判別する。ここで、アプリケーション902が、最後であると判定した場合、次のS2007へ進む。そして、S2007で、アプリケーション902は、プリンタドライバの印刷設定画面を表示する指示に従い、生成した図6に示すPrintTicketを指定してプリンタドライバ501の印刷設定画面表示を指示して、本

10

【0158】

一方、S2006で、最後の設定項目でないとアプリケーション902が判定した場合には、S2001に戻り、次の印刷設定のコントロールの状態を取得して、全ての印刷設定コントロールが終了するまで、処理を繰り返す。

【0159】

最終的には、図6に示すAPIの引数pPrintTicketには、アプリケーションの印刷設定画面の状態が反映される。さらに、UI文字列がReplaceDisplayStringプロパティに記述された、図11に示すようなPrintTicketが指定される。

20

【0160】

図12は、本実施形態を示す情報処理装置における第3のデータ処理手順の一例を示すフローチャートである。

【0161】

本例は、アプリケーション902が表示した図9に示す画面において、ユーザの操作により、プロパティボタン1207が押下される後のプリンタドライバ側の印刷設定画面を表示する処理例である。

【0162】

より具体的にいえば、本処理は、図5に示したCN_DrvDocumentPropertySheets関数の呼び出しにおいて、プリンタドライバ501に対しユーザインタフェースの表示が指示された場合の処理フローの一部に対応する。なお、ユーザインタフェースの表示の指示は、図6に示したCN_DOCUMENTPROPERTYHEADER構造体のfModeメンバにDM_IN_PROMPTが設定された場合に相当する。

30

【0163】

また、S2101～S2111は各ステップを示す。また、各ステップは、CPU1がRAM2にサブモジュール502をロードして実行することで実現される。なお、プリンタドライバ501のプログラムはコンピュータの外部メモリ11に記憶されており、RAM2に読み出されCPU1によって実行される。

【0164】

S2101で、サブモジュール502は、図10に示したアプリケーション902によるS2007における印刷設定画面の表示指示に応じた、プリンタドライバ側のコマンドを受信する。

40

【0165】

次に、S2102で、サブモジュール502は、図6に示したAPIのpPrintTicketに指定されているPrintTicketを取得する。

【0166】

次に、S2103で、サブモジュール502は、図13に示すようなサブモジュール502内で保持している文字列テーブル情報1601を取得し、メモリ中に展開する。

【0167】

図13は、図4に示したサブモジュール502内で保持される文字列テーブル情報1601の一例を示す図である。

50

【0168】

図13において、文字列テーブル情報1601には、機能名1602とその機能に対応したUIのコントロールID1603、UI文字列1604が関連付けられて格納されている。

【0169】

S2104では、サブモジュール502がS2102でアプリケーション902より取得したPrintTicket内の最初のFeatureを参照し、S2105で、サブモジュール502は、参照したFeature内にReplaceDisplayStringプロパティが存在するか否かを判別する。

【0170】

ここで、サブモジュール502がS2105でReplaceDisplayStringがあると判断した場合には、S2106で、サブモジュール502がS2103で取得した文字列テーブル情報1601内で現在参照しているFeatureと一致する機能名を検索する。

10

【0171】

次に、S2107では、サブモジュール502がS2106で一致する機能名が存在する否かを判断する。ここでサブモジュール502がS2106で一致する機能名が存在すると判断した場合、S2108に進む。そして、S2108で、サブモジュール502は、一致する機能名のUI文字列1603を、指定されたPrintTicket内のReplaceDisplayString文字列で置き換える。これにより、図13に示すUI文字列「ソート」は、アプリケーション902が生成したPrintTicketで指定される文字列「部単位印刷」に置き換えられる。

20

【0172】

一方、S2107で、サブモジュール502が一致した機能名が検索できないと判断した場合は、何もせずにS2109に進む。そして、S2109では、サブモジュール502が現在参照しているFeatureがPrintTicket内の最後のFeatureであるか否かを判別する。ここで、サブモジュール502が最後のFeatureであると判定された場合には、S2111で、UI文字列が置き換えられた、図14に示すような文字列テーブル情報2201のUI文字列2204に従って、プリンタドライバのUI画面をディスプレイ10上に表示する。

【0173】

図14は、図4に示したサブモジュール502が保持する文字列テーブル情報2201の一例を示す図である。

30

【0174】

この際、文字列テーブル情報2201内の各機能の機能名2202に関連付けられたコントロールID2203の文字列は、アプリケーション902から取得するPrintTicketで指定されたUI文字列2204に置き換えられて保持する。

【0175】

図14に示す文字列テーブル情報2201では、CollateのUI文字列（ソート）が、"部単位印刷"に、MediaSizeのUI文字列（用紙名称）が"用紙"に、それぞれアプリケーションが指定した文字列に置き換えられている状態を示す。

【0176】

40

この結果、図15に示すようなアプリケーションの文字列に置き換えられたプリンタドライバの印刷設定画面1800がディスプレイ10に表示される。

【0177】

図15、図16は、本実施形態を示す情報処理装置で表示可能なユーザインタフェースの一例を示す図である。なお、図15は、サブモジュール502による文字列の置き換えが実行された印刷設定画面に対応し、図16は、サブモジュール502による文字列の置き換えが実行されていない通常の印刷設定画面に対応する。

【0178】

図16では、通常のプリンタドライバの印刷設定画面を示しているが、用紙名称1701と、ソート1702のUI文字列が、図15に示す設定画面では、それぞれ"用紙"、"

50

部単位印刷"とアプリケーションの文字列に置き換えられていることを示している。

【0179】

以上の手順により、アプリケーションのUI文字列で、プリンタドライバのUI文字列の置換えが実現可能となる。

【0180】

〔第2実施形態〕

以下では、アプリケーションの印刷設定画面の表示フローと、図5、図6に示したインタフェース関数を用いて、プリンタドライバの印刷設定画面を表示する処理について図7Bに示すフローチャートを参照して説明する。

【0181】

10

本実施形態では、アプリケーションのUI文字列で、プリンタドライバのUI文字列の置換えた際に、文字列表示がUI表示可能な文字数を超えて表示すべき文字列の一部が切る状態となる場合でも、正しく置き換えた機能を示す文字列を表示可能とする例である。

【0182】

図7Bは、図4に示したアプリケーション902が自身で管理している印刷設定画面を表示する処理フローである。

【0183】

アプリケーション902のプログラムは、コンピュータの外部メモリ11に記憶されており、RAM2に読み出されCPU1によって実行される。

【0184】

20

アプリケーション902は、S1901で、プリンタデバイスの能力情報を取得するために、プリントチケットAPI505を通じて、プリンタドライバ501よりPrintCapabilitiesを取得する。

【0185】

ここで、アプリケーション902は、図13に示すようなXML形式のPrintCapabilitiesをプリンタドライバ501より取得する。図13の例に示すPrintCapabilitiesの例では、Collate、Duplex、MediaSize、OutputBinの4つのFeatureが含まれる場合である。また、それぞれのFeatureには、DisplayNameプロパティが指定されている。

【0186】

その後、S1902にて、アプリケーション902は、図8に示すような、機能名とそれに対応したUIの文字列テーブル情報1501を取得する。

30

【0187】

ここで、文字列テーブル情報1501には、機能名1502に応じたUI文字列1503、UIのコントロールID1504が格納されている。

【0188】

S1903では、アプリケーション902は、S1902にて取得した文字列テーブル情報1501を参照し、最初の機能の情報を取得する。次に、S1904で、アプリケーション902は、S1903で取得した機能がS1901でプリンタドライバ501より取得したPrintCapabilities内に含まれる機能か否かを判別する。

【0189】

40

ここで、アプリケーション902がS1901で取得した機能がPrintCapabilities内に存在すると判別した場合には、S1905へ進む。そして、S1905で、アプリケーション902は、文字列テーブル情報1501内に格納されたFeature名に応じた文字列を用いて、アプリケーションUIをディスプレイ10に表示する。

【0190】

次に、S1906で、アプリケーション902は、現在取得している機能情報が文字列テーブル情報1501の最後の項目であるか否かを判別し、最後であると判断したら、本処理を終了する。

【0191】

一方、S1905で、アプリケーション902が現在取得している機能情報が最後の項

50

目でないと判断された場合は、S 1 9 0 8に進む。そして、S 1 9 0 8で、アプリケーション9 0 2は、文字列テーブル情報1 5 0 1内の次の項目を取得した後、S 1 9 0 4～S 1 9 0 6までの処理を繰り返す。

【0 1 9 2】

一方、S 1 9 0 4にて、アプリケーション9 0 2がS 1 9 0 2で取得した機能がS 1 9 0 1でプリンタドライバ5 0 1よりPrintCapabilities内に存在しないと判別された場合には、その機能に対応したUIコントロールをグレイ表示し、使用不可の機能として通知する。

【0 1 9 3】

図9は、図7に示したのフローにより構成されたアプリケーションの印刷設定画面1 2 0 1の一例を示す。

10

【0 1 9 4】

図9において、用紙設定コントロール1 2 0 2、両面設定コントロール1 2 0 3、部単位印刷設定コントロール1 2 0 5がそれぞれ文字列テーブル情報1 5 0 1内のUI文字列で表示されている。

【0 1 9 5】

また、プリンタドライバ5 0 1より取得するPrintCapabilitiesに含まれていない機能、例えばステイブル機能に対応するステイブル設定コントロール1 2 0 6はグレイ表示されている。ここでは、ステイブル機能がグレイ表示（図中では、網掛けで示す）されている例を示す。

20

【0 1 9 6】

アプリケーションの印刷設定画面1 2 0 1には、さらにプリンタを選択するプリンタ選択コントロール1 2 0 4、プリンタドライバの印刷設定画面を表示するためのプロパティボタン1 2 0 7を備える。さらに、アプリケーションの印刷設定画面1 2 0 1には、印刷設定画面を確定させるOKボタン1 2 0 8、キャンセルボタン1 2 0 9を備える。

【0 1 9 7】

次に、アプリケーションの印刷設定画面1 2 0 1のプロパティボタン1 2 0 7が押下され、アプリケーション9 0 2からプリンタドライバ5 0 1に対して印刷設定画面を表示させる指示について説明する。

【0 1 9 8】

30

図10は、図9に示したアプリケーションのプロパティボタン1 2 0 7が押下された際に、アプリケーション9 0 2からプリンタドライバ5 0 1の印刷設定画面を表示させる指示を行う処理フローである。

【0 1 9 9】

なお、本フローに係わるアプリケーション9 0 2のプログラムはコンピュータの外部メモリ1 1に記憶されており、RAM 2に読み出されCPU 1によって実行される。

【0 2 0 0】

S 2 0 0 1で、アプリケーション9 0 2は、図9に示す印刷設定画面1 2 0 1内の印刷設定に関わるコントロールの選択状態を取得する。次に、S 2 0 0 2で、アプリケーション9 0 2は、図7に示したS 1 9 0 1でプリンタドライバ5 0 1より取得したPrintCapabilities内のFeature一覧のうち、S 2 0 0 1で印刷設定画面1 2 0 1の状態から取得したコントロールに対応したFeatureを参照する。

40

【0 2 0 1】

そして、S 2 0 0 3では、アプリケーション9 0 2は、参照したFeatureのOptionの一覧の中で、選択状態の設定項目に対応した項目以外のOptionを削除する。

【0 2 0 2】

その結果、図11に示すように、Featureに特定の設定項目のOptionのみが記述された状態のPrintTicket 1 4 0 1が生成される。図11の例では、"Collate"Featureでは、"On"Optionが、"Duplex"Featureでは、"OFF"Optionがそれぞれ選択されている状態を示す。

【0 2 0 3】

50

次に、S 2 0 0 4で、アプリケーション9 0 2は、文字列テーブル情報1 5 0 1を取得する。そして、S 2 0 0 5で、アプリケーション9 0 2は、文字列テーブル情報1 5 0 1内を参照して、現在のコントロールに関連する機能の文字列情報を取得し、ReplaceDisplayStringプロパティとしてPrintTicket 1 4 0 1に指定する。

【0 2 0 4】

図1 1に示す例では、"Collate"FeatureのReplaceDisplayStringプロパティとして、アプリケーションの文字列テーブル情報1 5 0 1内のCollateに対応した文字列、"部単位印刷"が記述されている。

【0 2 0 5】

次に、S 2 0 0 6で、アプリケーション9 0 2は、S 2 0 0 1で参照した印刷設定のコントロールが最後の設定項目か否かを判別する。S 2 0 0 6で、アプリケーション9 0 2が最後であると判定された場合、S 2 0 0 7へ進む。そして、S 2 0 0 7で、アプリケーション9 0 2は、プリンタドライバ5 0 1の印刷設定画面を表示させる指示のためのAPI 5 0 5に、作成したPrintTicketを指定してプリンタドライバ5 0 5の印刷設定画面の表示を指示する。

10

【0 2 0 6】

一方、S 2 0 0 7で、アプリケーション9 0 2が最後の設定項目でないと判定された場合には、S 2 0 0 1に戻り、次の印刷設定のコントロールの状態を取得して、全ての印刷設定コントロールが終了するまで、処理を繰り返す。

【0 2 0 7】

20

そして、最終的には図6に示すAPIの引数pPrintTicketには、アプリケーションの印刷設定画面の状態が反映され、UI文字列がReplaceDisplayStringプロパティに記述された、図1 1に示すようなPrintTicketが指定される。

【0 2 0 8】

次に、図1 2を用いて、ユーザの操作に従いプロパティボタン1 2 0 7が押下された後のプリンタドライバ5 0 1側の印刷設定画面を表示する処理について説明する。

【0 2 0 9】

本フローに係わるプリンタドライバを構成する新ドライバサブモジュール（以下、サブモジュールと呼ぶ）5 0 2のプログラムはコンピュータの外部メモリ1 1に記憶されており、RAM 2に読み出されCPU 1によって実行される。

30

【0 2 1 0】

図5に示したCN_DrvDocumentPropertySheets関数の呼び出しにおいて、プリンタドライバ5 0 1に対しユーザインタフェースの表示が指示された場合の処理フローの一部である。

【0 2 1 1】

ユーザインタフェースの表示の指示は、図6に示したCN_DOCUMENTPROPERTYHEADER構造体のfModeメンバにDM_IN_PROMPTが設定された場合に相当する。

【0 2 1 2】

S 2 1 0 1で、サブモジュール5 0 2は、図1 0に示したアプリケーション9 0 2によるS 2 0 0 7の印刷設定画面の表示指示に応じたコマンドを受信する。次に、S 2 1 0 2では、サブモジュール5 0 2は、図6に示すAPIのpPrintTicketに指定されているPrintTicket取得する。

40

【0 2 1 3】

次に、S 2 1 0 3で、サブモジュール5 0 2は、図1 7に示すように、プリンタドライバ5 0 1内で保持している文字列テーブル情報2 5 0 1を取得し、メモリ中に展開する。

【0 2 1 4】

図1 7は、図5に示したプリンタドライバ5 0 1が保持する文字列テーブル情報の一例を示す図である。

【0 2 1 5】

図1 7に示すように、文字列テーブル情報には、機能名2 5 0 2とその機能に対応した

50

UIのコントロールID 2503、UI文字列 2504、最大UI表示文字数 2505が関連付けられて格納されている。

【0216】

S2104では、サブモジュール502は、S2102で取得したPrintTicket内の最初のFeatureを参照し、S2105で参照したFeature内にReplaceDisplayStringプロパティが存在するか否かを判別する。

【0217】

ここで、サブモジュール502がS2105で、ReplaceDisplayStringがあると判断された場合には、S2106へ進む。そして、S2106で、サブモジュール502は、S2103で取得した文字列テーブル情報1601内で現在参照しているFeatureと一致する機能名を検索する。

10

【0218】

次に、S2107で、サブモジュール502がS2106で一致する機能名が存在するかどうかを判断する。ここで、サブモジュール502が一致する機能名が存在すると判断した場合、S2108に進む。

【0219】

そして、S2108で、サブモジュール502が一致する機能名のUI文字列 2503を、指定されたPrintTicket内のReplaceDisplayString文字列で置き換える。

【0220】

一方、S2107で、サブモジュール502が一致した機能名が検索できないと判断した場合は、何もせずにS2109に進む。

20

【0221】

そして、S2109では、サブモジュール502は現在参照しているFeatureがPrintTicket内の最後のFeatureであるか否かを判別する。ここで、サブモジュール502が最後のFeatureであると判別した場合には、S2111へ進む。そして、S2111で、UI文字列が置き換えられた、例えば図14に示すような文字列テーブル情報2201のUI文字列2204に従って、プリンタドライバのUI画面をディスプレイ10に表示する。

【0222】

この際、サブモジュール502は、図17に示す文字列テーブル情報2501を、図18に示す文字列テーブル情報2601内の各機能に関連付けられたコントロールIDの文字列をアプリケーション902から取得したPrintTicketに従って置き換えて保持する。

30

【0223】

図18は、図5に示したプリンタドライバ501が保持する文字列テーブル情報の一例を示す図である。

【0224】

図18に示す文字列テーブル情報2601では、機能名2602がCollateのUI文字列(ソート)2604が、"部単位印刷"に、MediaSizeのUI文字列(用紙名称)が"用紙"に、それぞれアプリケーションが指定した文字列に置き換えられている。なお、2603はコントロールIDを示す。

【0225】

この結果、図18に示す文字列テーブル情報2601に従って図19に示すようなアプリケーション902の文字列に置き換えられたプリンタドライバ501の印刷設定画面がディスプレイ10に表示される。

40

【0226】

ここでは、プリンタドライバ501の印刷設定画面で設定された最大表示文字数2605を超える文字は表示されない。つまり、部単位印刷は、5文字から構成されるため、最大表示文字数「4」に従い、4文字分となる「部単位印」のみが表示された状態となる。したがって、ユーザは、一見してその項目名を正確に認識できない場合がある。

【0227】

そこで、以下、プリンタドライバ501が図20に示す表示処理を行うことで、置き換

50

えられた文字列すべてを、図 2 1 に示すように表示された項目名の近傍に表示する。これにより、置き換えられた全ての文字列と、表示されなかった文字列とを対向して表示することが可能となる。

【0228】

図 2 0 は、本実施形態を示す情報処理装置における第 4 のデータ処理手順の一例を示すフローチャートである。本例は、プリンタドライバの印刷設定画面上のコントロールにマウスオーバーした際の表示処理例である。なお、S 2 4 0 1 ~ S 2 4 0 6 は各ステップを示す。また、各ステップは、CPU 1 が RAM 2 にプリンタドライバ 5 0 1 をロードして実行することで実現される。

【0229】

10

まず、S 2 4 0 1 で、プリンタドライバ 5 0 1 がマウスオーバーイベントが発生したことを検出したら、S 2 4 0 2 で、プリンタドライバ 5 0 1 は、イベントが発生した UI のコントロール情報を取得する。

【0230】

次に、S 2 4 0 3 では、プリンタドライバ 5 0 1 は、S 2 4 0 2 で取得したコントロール情報のコントロール ID が図 1 8 に示す文字列テーブル情報 2 6 0 1 内に存在するか否かを判別する。

【0231】

ここで、プリンタドライバ 5 0 1 が文字列テーブル情報にコントロール ID が存在すると判別した場合は、プリンタドライバ 5 0 1 は、S 2 4 0 4 で文字列テーブル情報 2 6 0 1 の最大表示文字数 2 6 0 5 (内容「4」) を取得する。

20

【0232】

一方、S 2 4 0 3 で、プリンタドライバ 5 0 1 が文字列テーブル情報にコントロール ID が存在しないと判別した場合は、何もせずに処理を終了する。

【0233】

そして、S 2 4 0 5 では、プリンタドライバ 5 0 1 は、該コントロールに対応した最大表示文字数 2 6 0 5 と UI 文字列 2 6 0 4 の UI 表示文字数を比較する。

【0234】

そして、プリンタドライバ 5 0 1 は、最大表示文字数 2 6 0 5 より UI 文字列 2 6 0 4 の UI 表示文字数の方が大きいかどうかを判断する。ここで、プリンタドライバ 5 0 1 が最大表示文字数 2 6 0 5 より UI 文字列 2 6 0 4 の UI 表示文字数の方が大きいと判断された場合には、S 2 4 0 6 へ進む。

30

【0235】

具体的には、プリンタドライバ 5 0 1 は、UI 文字列である「部単位印刷」の場合、5 文字から構成されるため、最大文字数「4」を超えると判断する。

【0236】

そして、S 2 6 0 4 で、プリンタドライバ 5 0 1 は、ツールチップを用いてすべての UI 文字列 2 6 0 4 の文字列すべてを、置き換えられた文字列「部単位印」の近傍に、総文字数が 5 文字の「部単位印刷」をディスプレイ 1 0 にポップアップするように表示制御して、本処理を終了する。なお、本例は、言語が日本語の場合を示すが、仕向け地に応じて、表示する文字列の言語やその文字数や表示態様は本実施形態に限定されることはない。

40

【0237】

図 2 1 は、図 1 に示したディスプレイ 1 0 に表示されるユーザインタフェースの一例を示す図である。

【0238】

本例は、ツールチップにより、プリンタドライバ 5 0 1 で置き換えられた UI 文字列が表示された印刷設定画面例である。

【0239】

図 2 1 に示すように、本実施形態では、4 文字目で切れている「部単位印刷」の表示がツールチップで全体が表示される。

50

【0240】

一方、S2405にて、プリンタドライバ501が最大表示文字数2605よりUI文字列2604が小さいと判断した場合には、何もせずに本処理を終了する。

【0241】

以上の手順により、アプリケーションのUI文字列で、プリンタドライバのUI文字列の置換えた際に、上述した文字列表示が切れる状態となる場合においても、置き換えた文字列を全て表示することができる。これにより、ユーザは、文字列を指示する操作を行うと、その機能文字列の内容を確認できる。

【0242】

〔第3実施形態〕

以下では、アプリケーションの印刷設定画面の表示フローと、図5、図6に示したインタフェース関数を用いて、プリンタドライバの印刷設定画面の表示フローについて説明する。

【0243】

本実施形態は、アプリケーション902のUI文字列で、プリンタドライバ501のUI文字列の置換えた際に、プリンタドライバの元のUI文字列と置き換えた文字列とを同時に表示させる例である。

【0244】

図7に、アプリケーション902が自身で管理している印刷設定画面を表示する際のフローを示す。

【0245】

なお、アプリケーション902のプログラムはコンピュータの外部メモリ11に記憶されており、RAM2に読み出されCPU1によって実行される。

【0246】

アプリケーション902は、S1901で、プリンタデバイスの能力情報を取得するために、プリントチケットAPI505を通じて、PrintCapabilitiesを取得する。

【0247】

具体的には、アプリケーション902は、図32に示すようなXML形式のPrintCapabilitiesをプリンタドライバ501より取得する。図32の例に示すPrintCapabilitiesには、Collate、Duplex、MediaSize、OutputBinの4つのFeatureが含まれる。また、それぞれのFeatureには、DisplayNameプロパティが指定されている例である。

【0248】

その後、アプリケーション902は、図7に示したS1902にて、図8に示すような、機能名とそれに対応したUIの文字列テーブル情報1501を取得する。

【0249】

図8に示すように、文字列テーブル情報1501には、機能名1502に応じたUI文字列1503、UIのコントロールID1504が格納されている。

【0250】

そして、S1903で、アプリケーション902は、S1902にて取得した文字列テーブル情報1501を参照し、最初の機能の情報を取得する。

【0251】

次に、S1904で、アプリケーション902は、S1903で文字列テーブル情報1501から取得した機能がS1901でプリンタドライバ501より取得したPrintCapabilities内に含まれる機能か否かを判別する。

【0252】

ここで、アプリケーション902がS1903で取得した機能がプリンタドライバ501より取得したPrintCapabilities内に存在すると判別された場合には、S1905へ進む。そして、S1905で、アプリケーション902は、文字列テーブル情報1501内に格納されたFeature名に応じた文字列を用いて、アプリケーションUIをディスプレイ10に表示する。

10

20

30

40

50

【0253】

次に、S1906で、アプリケーション902は、現在取得している機能情報が文字列テーブル情報1501の最後の項目であるか否かを判別し、最後であると判断されたら、本処理を終了する。

【0254】

一方、S1905で、アプリケーション902が最後の項目でないと判別した場合は、S1908に進み、文字列テーブル情報1501内の次の項目を取得した後、S1904～S1906までの処理を繰り返す。

【0255】

一方、S1904にて、アプリケーション902がS1903で取得した機能がS1901でプリンタドライバ501より取得したPrintCapabilities内に存在しないと判別された場合には、S1907へ進む。そして、S1907で、アプリケーション902は、機能に対応したUIコントロールをUI画面上でグレイ表示（図中では、網掛けで示す）し、使用不可の機能として、ユーザに認知させる。

【0256】

図9は、図7に示した表示制御処理により構成されたアプリケーションの印刷設定画面1201である。

【0257】

図9において、用紙設定コントロール1202、両面設定コントロール1203、部単位印刷設定コントロール1205がそれぞれ文字列テーブル情報1501内のUI文字列で表示されている状態に対応する。また、プリンタドライバ501より取得したPrintCapabilitiesに含まれていないステイプル設定コントロール1206がグレイ表示されている状態に対応する。

【0258】

アプリケーションの印刷設定画面1201には、さらにプリンタを選択するプリンタ選択コントロール1204、プリンタドライバの印刷設定画面を表示するためのプロパティボタン1207を含んで構成されている。さらに、アプリケーションの印刷設定画面1201には、印刷設定画面を確定させるOKボタン1208、キャンセルボタン1209を含んで構成されている。

【0259】

次に、アプリケーションの印刷設定画面1201のプロパティボタン1207が押下され、アプリケーション902からプリンタドライバの印刷設定画面の表示を指示する処理について説明する。

【0260】

図10は、アプリケーション902のプロパティボタン1207が押下された際に、アプリケーション902からプリンタドライバ501の印刷設定画面を表示させる指示を行う処理フローである。

【0261】

なお、本フローに係わるアプリケーション902のプログラムは、コンピュータの外部メモリ11に記憶されており、RAM2に読み出されCPU1によって実行される。

【0262】

S2001で、アプリケーション902は、印刷設定画面1201内の印刷設定に関わるコントロールの選択状態を取得する。次に、S2002で、アプリケーション902は、図7に示したS1901において、プリンタドライバ501より取得したPrintCapabilities内のFeature一覧のうち、S2001で状態を取得したコントロールに対応したFeatureを参照する。

【0263】

次に、S2003では、アプリケーション902は、参照したFeatureのOptionの一覧の中で、選択状態の設定項目に対応した項目以外のOptionを削除する。

【0264】

その結果、アプリケーション 902 は、図 11 に示すように、Feature に特定の設定項目の Option のみが記述された状態の PrintTicket 1401 を生成する。

【0265】

図 11 に示す例では、"Collate"Feature では、"On"Option が、"Duplex"Feature では、"OFF"Option がそれぞれ選択されている例である。

【0266】

次に、S2004 で、アプリケーション 902 は、文字列テーブル情報 1501 を取得する。そして、S2005 で、アプリケーション 902 は、文字列テーブル情報 1501 内を参照して、現在のコントロールに関連する機能の文字列情報を取得し、ReplaceDisplayString プロパティとして PrintTicket 1401 に指定する。

10

【0267】

図 11 の例では、"Collate"Feature の ReplaceDisplayString プロパティとして、アプリケーションの文字列テーブル情報 1501 内の Collate に対応した文字列、"部単位印刷"が記述されている。

【0268】

次に、S2006 で、アプリケーション 902 は、S2001 で参照した印刷設定のコントロールが最後の設定項目か否かを判別する。ここで、アプリケーション 902 が S2006 で最後であると判定した場合、S2007 へ進む。そして、S2007 で、アプリケーション 902 は、図 6 に示すプリンタドライバの印刷設定画面を表示させる指示のための API に、生成した PrintTicket を指定してプリンタドライバ 501 に印刷設定画面を表示させる指示を行う。

20

【0269】

一方、S2007 で、アプリケーション 902 が最後の設定項目でないと判定した場合には、S2001 に戻り、次の印刷設定のコントロールの状態を取得して、全ての印刷設定コントロールが終了するまで、処理を繰り返す。

【0270】

最終的には図 6 に示す API の引数 pPrintTicket には、アプリケーションの印刷設定画面の状態が反映され、UI 文字列が ReplaceDisplayString プロパティに記述された、図 11 に示すような PrintTicket が指定される。

【0271】

30

図 22 は、本実施形態を示す情報処理装置における第 5 のデータ処理手順の一例を示すフローチャートである。本例は、図 9 に示したプロパティボタン 1207 が押下された後のプリンタドライバ 501 側の印刷設定画面を表示する処理例である。

【0272】

より具体的には、図 5 に示す CN_DrvDocumentPropertySheets 関数の呼び出しにおいて、プリンタドライバに対しユーザインタフェースの表示が指示された場合の処理フローの一部である。

【0273】

また、ユーザインタフェースの表示の指示は、図 6 に示した CN_DOCUMENTPROPERTYHEADER 構造体の fMode メンバに DM_IN_PROMPT が設定された場合に相当する。

40

【0274】

なお、S3201～S3212 は各ステップを示す。また、各ステップに係わるサブモジュール 502 のプログラムはコンピュータの外部メモリ 11 に記憶されており、RAM 2 に読み出され CPU 1 によって実行される。

【0275】

S3201 で、サブモジュール 502 は、S2007 の印刷設定画面を表示する指示に応じた、プリンタドライバ側のコマンドを受信する。S3202 では、サブモジュール 502 は、図 6 に示す API の pPrintTicket に指定されている PrintTicket をアプリケーション 902 より取得する。

【0276】

50

次に、S 3 2 0 3で、サブモジュール5 0 2は、図2 3に示すようなプリンタドライバ内で保持している文字列テーブル情報2 9 0 1を取得し、R A M 2 上中に展開する。

【0 2 7 7】

図2 3，図2 4は、本実施形態を示す情報処理装置のプリンタドライバが保持する文字列テーブル情報の一例を示す図である。

【0 2 7 8】

図2 3に示す文字列テーブル情報2 9 0 1には、機能名2 9 0 2とその機能に対応したU IのコントロールI D 2 9 0 3、U I文字列2 9 0 4、置換えフラグ2 9 0 5、オリジナル文字列2 9 0 6が関連付けられて格納されている。

【0 2 7 9】

次に、S 3 2 0 4では、サブモジュール5 0 2は、S 3 2 0 2でアプリケーション9 0 2より取得したPrintTicket内の最初のFeatureを参照する。そして、S 3 2 0 5で、サブモジュール5 0 2は、参照したFeature内にReplaceDisplayStringプロパティが存在するか否かを判別する。

【0 2 8 0】

ここで、サブモジュール5 0 2が「ReplaceDisplayStringがあると判断した場合には、S 3 2 0 6へ進む。そして、S 3 2 0 6で、サブモジュール5 0 2は、S 3 2 0 3で取得した文字列テーブル情報1 6 0 1内で現在参照しているFeatureと一致する機能名を検索する。

【0 2 8 1】

そして、S 3 2 0 7で、サブモジュール5 0 2は、S 3 2 0 6による検索で、一致する機能名が存在するかどうかを判断する。ここで、サブモジュール5 0 2が一致する機能名が存在すると判断した場合、S 3 2 0 8に進む。そして、S 3 2 0 8で、サブモジュール5 0 2は、一致する機能名のU I文字列2 9 0 3を、指定されたPrintTicket内のReplaceDisplayString文字列で置き換える。

【0 2 8 2】

一方、S 3 2 0 7で、サブモジュール5 0 2が一致した機能名が検索できないと判断した場合は、何もせずにS 3 2 1 0に進む。

【0 2 8 3】

次に、S 3 2 0 9で、サブモジュール5 0 2は、U I文字列を置き換えたことを示す図2 3に示す置換えフラグ2 9 0 5をT R U Eに設定する。次に、S 3 2 1 0では、サブモジュール5 0 2は、現在参照しているFeatureがPrintTicket内の最後のFeatureであるかを判別する。ここで、サブモジュール5 0 2が最後のFeatureであると判定された場合には、S 3 2 1 2へ進む。そして、S 3 2 1 2で、サブモジュール5 0 2は、U I文字列が置き換えられた、図2 4に示すような文字列テーブル情報3 0 0 1のU I文字列3 0 0 4に従って、U I画面をディスプレイ1 0に表示する。

【0 2 8 4】

この際、サブモジュール5 0 2は、文字列テーブル情報3 0 0 1内の各機能に関連付けられたコントロールI Dの文字列を置き換える。

【0 2 8 5】

図2 4に示す文字列テーブル情報3 0 0 1では、機能名3 0 0 2がCollateのU I文字列（ソート）3 0 0 4が、「部単位印刷」に、MediaSizeのU I文字列（用紙名称）が「用紙」に、それぞれアプリケーションが指定した文字列に置き換えられる例を示している。なお、図2 4において、3 0 0 3はコントロールI Dを示し、3 0 0 5は置換えフラグを示し、3 0 0 6はオリジナル文字列を示す。

【0 2 8 6】

この結果、サブモジュール5 0 2は、図1 9のユーザインタフェースに示すようなアプリケーションの文字列に置き換えられた印刷設定画面をディスプレイ1 0に表示する。

【0 2 8 7】

図2 5は、本実施形態を示す情報処理装置における第5のデータ処理手順の一例を示す

10

20

30

40

50

フローチャートである。本例は、サブモジュール502の印刷設定画面上のコントロールにマウスオーバーした際の処理例である。なお、S2801～S2806は各ステップに係わるサブモジュール502のプログラムはコンピュータの外部メモリ11に記憶されており、RAM2に読み出されCPU1によって実行される。

【0288】

まず、S2801で、サブモジュール502が、ユーザがポインティングデバイス进行操作することに従い、図26に示すように、マウスが移動するマウスオーバーイベントが発生したことを検出すると、S2802へ進む。

【0289】

そして、S2802で、サブモジュール502は、マウスオーバーイベントが発生したUIのコントロール情報を取得する。ここでは、図26に示すように機能を示す、例えば「用紙」の付近に移動したことを検出した場合であるので、「collate」が取得される。

【0290】

次に、S2803では、サブモジュール502は、S2802で取得したコントロール情報のコントロールIDが文字列テーブル情報3001内に存在するか否かを判別する。ここで、サブモジュール502がS2803で文字列テーブル情報にコントロールIDが存在すると判別した場合は、S2804で文字列テーブル情報3001の置換えフラグ2905を取得する。

【0291】

一方、サブモジュール502がS2803で文字列テーブル情報3001にコントロールIDが存在しないと判定した場合は、何もせずに処理を終了する。

【0292】

次に、S2805では、サブモジュール502は、置換え文字列がTRUEであるか否かを文字列テーブル情報3001の置換えフラグ3005を参照して判断する。ここで、サブモジュール502が、置換えフラグ3005がTRUEであると判断した場合には、S2806にて、該コントロールのオリジナル文字列3005をツールチップによりディスプレイ10に表示する。

【0293】

図26は、図1に示したディスプレイ10に表示されるユーザインタフェースの一例を示す図である。本例は、プリンタドライバの印刷設定画面例である。

【0294】

図26に示すように、本実施形態では、ツールチップにより、アプリケーション902で置き換えられたUI文字列（部単位印刷）と共に元々のプリンタドライバ501の文字列（ソート）が、同一のUI画面上に表示される。しかも、部単位印刷とソートとが近接した位置関係を維持した状態で表示される。これにより、簡単な操作を行うだけでユーザはドライバUI画面上で、文字列を置き換える前と置き換えた後の状態を容易に視認できる。

【0295】

一方、S2805にて、置換えフラグ3005がFALSEであるとサブモジュール502が判断した場合には、何もせずに、本処理を終了する。

【0296】

以上の手順により、アプリケーションのUI文字列で、プリンタドライバのUI文字列の置換えた際に、プリンタドライバの元のUI文字列を同時に表示させることが可能となる。

【0297】

なお、第2実施形態と第3実施形態による表示制御を有効とするモードを備えて、初期設定される当該モードに応じて、文字列の表示制御を切り換えるように構成してもよい。

【0298】

また、コマンドキー等の操作を行うことで、機能文字列の表示制御を第2実施形態に示す表示制御から第3実施形態に示す表示制御に切り換え可能に構成してもよい。この際、いずれの表示を優先させるかは、初期設定で決定される表示制御から他の表示制御に切り

10

20

30

40

50

換えるものとする。

【0299】

〔第4実施形態〕

以下、図27に示すメモリマップを参照して本発明に係る情報処理装置で読み取り可能なデータ処理プログラムの構成について説明する。

【0300】

図27は、本発明に係る情報処理装置で読み取り可能な各種データ処理プログラムを格納する記憶媒体のメモリマップを説明する図である。

【0301】

なお、特に図示しないが、記憶媒体に記憶されるプログラム群を管理する情報、例えばバージョン情報、作成者等も記憶され、かつ、プログラム読み出し側のOS等に依存する情報、例えばプログラムを識別表示するアイコン等も記憶される場合もある。

【0302】

さらに、各種プログラムに従属するデータも上記ディレクトリに管理されている。また、各種プログラムをコンピュータにインストールするためのプログラムや、インストールするプログラムが圧縮されている場合に、解凍するプログラム等も記憶される場合もある。

【0303】

本実施形態における図7B、図10、図12、図20、図22、図25に示す機能が外部からインストールされるプログラムによって、ホストコンピュータにより遂行されていてもよい。そして、その場合、CD-ROMやフラッシュメモリやFD等の記憶媒体により、あるいはネットワークを介して外部の記憶媒体から、プログラムを含む情報群を出力装置に供給される場合でも本発明は適用されるものである。

【0304】

以上のように、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コードを記録した記憶媒体を、システムあるいは装置に供給する。そして、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成されることは言うまでもない。

【0305】

この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0306】

従って、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、プログラムの形態を問わない。

【0307】

プログラムを供給するための記憶媒体としては、例えばフレキシブルディスク、ハードディスク、光ディスク、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리카ード、ROM、DVDなどを用いることができる。

【0308】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0309】

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、該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

10

20

30

40

50

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サーバやftpサーバ等も本発明の請求項に含まれるものである。

【0310】

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。

10

【0311】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけではない。例えばそのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）等が実際の処理の一部または全部を行う。そして、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0312】

さらに、記憶媒体から読み出されたプログラムコードが、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、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

20

【0313】

本発明は上記実施形態に限定されるものではなく、本発明の趣旨に基づき種々の変形（各実施形態の有機的な組合せを含む）が可能であり、それらを本発明の範囲から除外するものではない。

【0314】

本発明の様々な例と実施形態を示して説明したが、当業者であれば、本発明の趣旨と範囲は、本明細書内の特定の説明に限定されるのではない。

【図面の簡単な説明】

30

【0315】

【図1】本発明の第1実施形態を示す情報処理装置、印刷装置を含むプリンタ制御システムの一例を示すブロック図である。

【図2】図1に示したホストコンピュータのモジュール構成を説明するブロック図である。

【図3】図1に示したホストコンピュータにインストールされるアプリケーションと、各プリンタ用のプリンタドライバとの関係を説明する模式図である。

【図4】図1に示したホストコンピュータにインストールされるアプリケーションと、各プリンタ用のプリンタドライバとの関係を詳細に説明する模式図である。

【図5】情報処理装置にインストールされるプリンタドライバが公開するインタフェース関数の一例を示す図である。

40

【図6】情報処理装置にインストールされるプリンタドライバが公開するインタフェース関数の一例を示す図である。

【図7A】本実施形態を示す情報処理装置におけるユーザインタフェースの表示制御を説明する図である。

【図7B】本実施形態を示す情報処理装置における第1のデータ処理手順の一例を示すフローチャートである。

【図8】本実施形態を示す情報処理装置で管理される文字列テーブル情報の一例を示す図である。

【図9】図1に示したディスプレイに表示されるアプリケーション印刷設定画面の一例を

50

示す図である。

【図 10】本実施形態を示す情報処理装置における第 2 のデータ処理手順の一例を示すフローチャートである。

【図 11】本実施形態を示す情報処理装置で生成される PrintTicket の一例を示す図である。

【図 12】本実施形態を示す情報処理装置における第 3 のデータ処理手順の一例を示すフローチャートである。

【図 13】図 4 に示したサブモジュール内で保持される文字列テーブル情報の一例を示す図である。

【図 14】図 4 に示したサブモジュールが保持する文字列テーブル情報の一例を示す図である。

10

【図 15】本実施形態を示す情報処理装置で表示可能なユーザインタフェースの一例を示す図である。

【図 16】本実施形態を示す情報処理装置で表示可能なユーザインタフェースの一例を示す図である。

【図 17】図 5 に示したドライバ本体が保持する文字列テーブル情報の一例を示す図である。

【図 18】図 5 に示したドライバ本体が保持する文字列テーブル情報の一例を示す図である。

【図 19】本実施形態を示す情報処理装置で表示可能なユーザインタフェースの一例を示す図である。

20

【図 20】本実施形態を示す情報処理装置における第 4 のデータ処理手順の一例を示すフローチャートである。

【図 21】図 1 に示したディスプレイ 10 に表示されるユーザインタフェースの一例を示す

【図 22】本実施形態を示す情報処理装置における第 5 のデータ処理手順の一例を示すフローチャートである。

【図 23】本実施形態を示す情報処理装置のプリンタドライバが保持する文字列テーブル情報の一例を示す図である。

【図 24】本実施形態を示す情報処理装置のプリンタドライバが保持する文字列テーブル情報の一例を示す図である。

30

【図 25】本実施形態を示す情報処理装置における第 6 のデータ処理手順の一例を示すフローチャートである。

【図 26】図 1 に示したディスプレイに表示されるユーザインタフェースの一例を示す図である。

【図 27】本発明に係る情報処理装置で読み取り可能な各種データ処理プログラムを格納する記憶媒体のメモリマップを説明する図である。

【図 28】情報処理装置で処理可能な電子文書フォーマットの構造を説明する図である。

【図 29】図 28 に示す電子文書の各パーツに関連づけ可能な印刷設定の一例を示す図である。

40

【図 30】従来の印刷システムにおけるメモリブロックを説明する図である。

【図 31】図 30 に示した DEVMODE パブリック部の一例を示す図である。

【図 32】情報処理装置にインストールされるプリンタドライバがアプリケーションに返す PrintCapabilities の一例を示す図である。

【図 33】情報処理装置における印刷制御モジュールの構造を説明するブロック図である。

【符号の説明】

【0316】

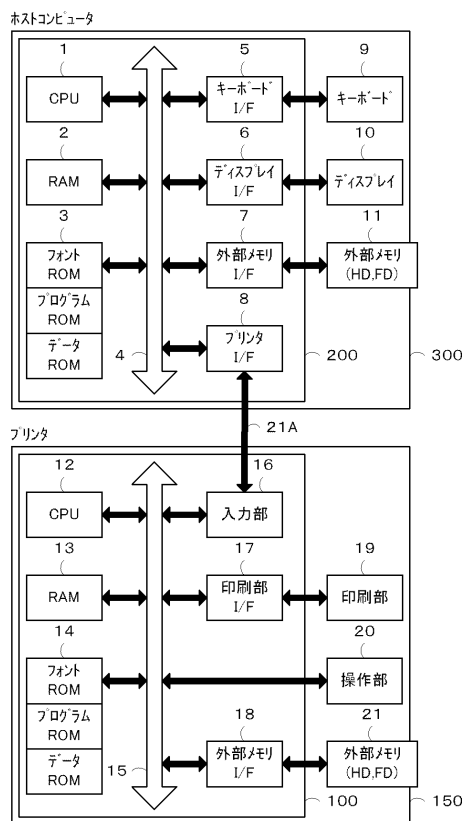
501 プリンタドライバ

502 サブモジュール

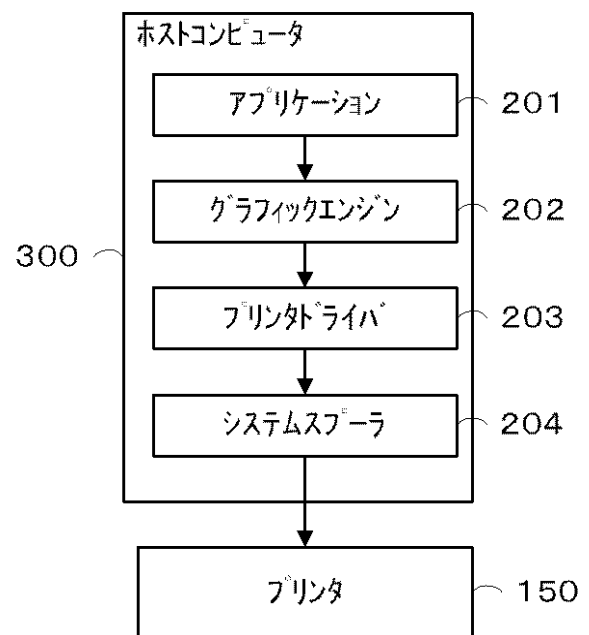
50

- 505 プリントチケットAPI
 901 プリントチケット入出力部
 902 新インタフェース対応アプリケーション

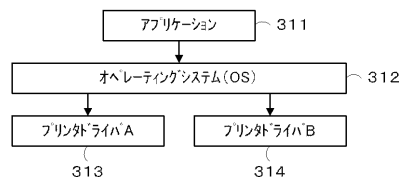
【図 1】



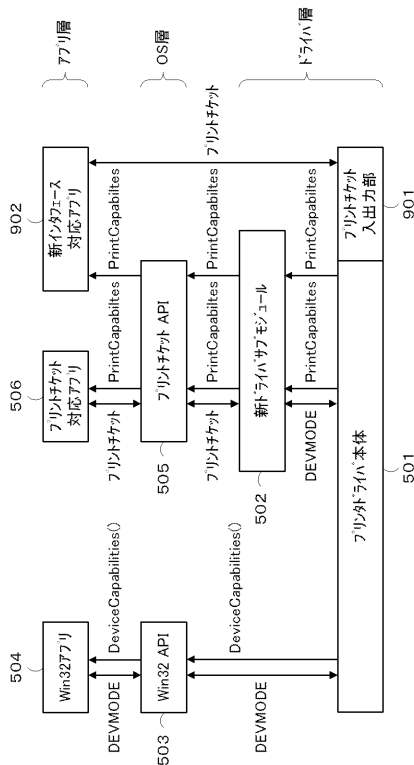
【図 2】



【図 3】



【図 4】



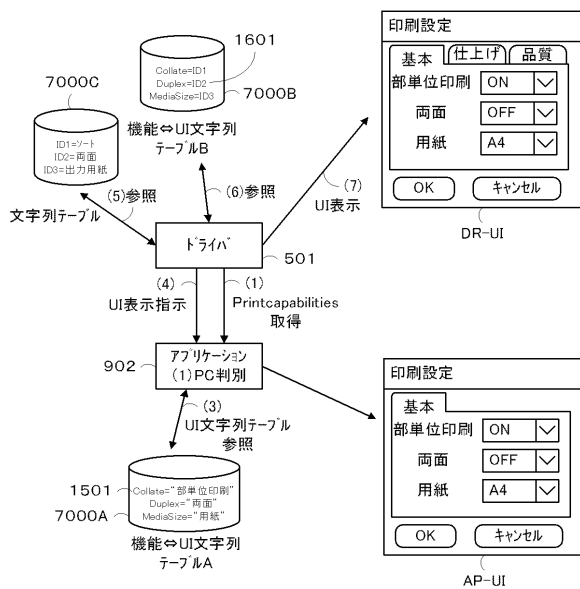
【図 5】

```
LONG CN_DrvDocumentPropertySheets (
    __in_opt PPROPSHEETUI_INFO pPSUIInfo,
    LPARAM lParam
);
```

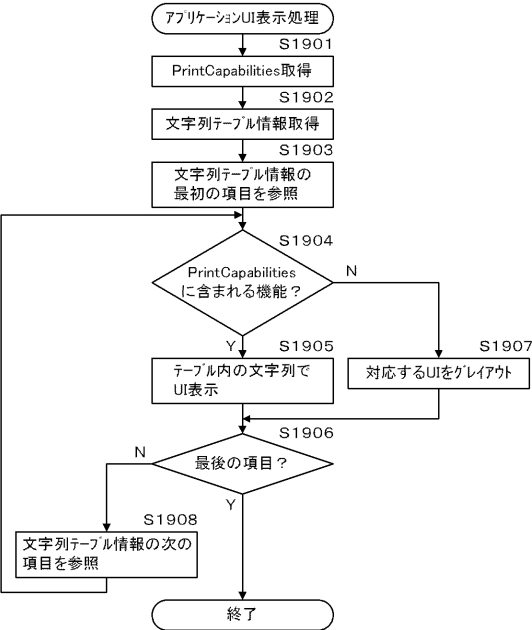
【図 6】

```
typedef struct _CN_DOCUMENTPROPERTYHEADER {
    __in WORD cbSize;
    __in HANDLE hPrinter;
    __in LPWSTR pszPrinterName;
    __in EPrintTicketScope scope;
    __inout IStream *pPrintTicket;
    __in DWORD fMode;
} CN_DOCUMENTPROPERTYHEADER;
```

【図 7 A】



【図 7 B】



【図 8】

1501		
機能名 (Feature)	UI文字列	コントロールID
Collate	部単位印刷	COLLATE.ID
Duplex	両面	DUPLEX.ID
MediaSize	用紙	MEDIA SIZE.ID
Staple	ステイプル	STAPLE.ID

【図 9】

1201 印刷設定

1202 用紙 A4 ☒ 部単位印刷

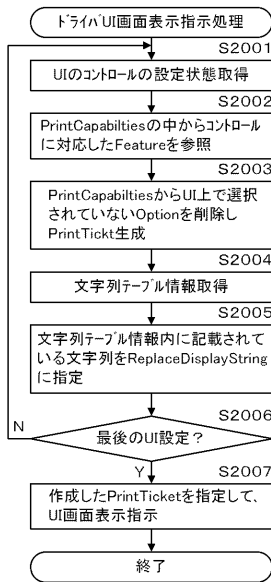
1203 両面 OFF ☒ スティブル OFF ☒

1204 プリント名 プリント1

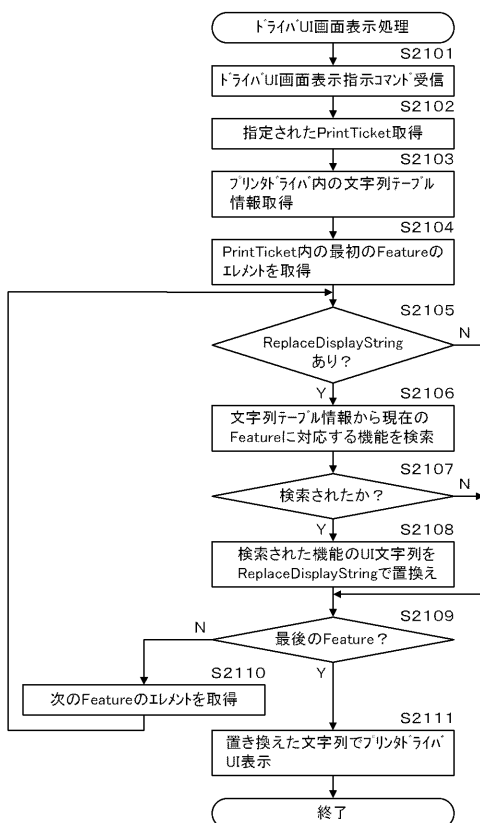
1208

1205 1206 1207 1209

【図 10】



【図 12】



【図 11】

```

1401 <PrintTicket>
1402 <Feature name="Collate">
1403 <Property name="ReplaceDisplayString">
  <psf:Value xsi:type="xs:string">部単位印刷</psf:Value>
</Property>
1404 <Option name="On"/>
</Feature>
<Feature name="Duplex">
  <Property name="ReplaceDisplayString">
    <psf:Value xsi:type="xs:string">両面</psf:Value>
  </Property>
  <Option name="OFF"/>
</Feature>
<Feature name="MediaSize">
  <Property name="ReplaceDisplayString">
    <psf:Value xsi:type="xs:string">用紙</psf:Value>
  </Property>
  <Option name="A4"/>
</Feature>
<Feature name="OutputBin">
  <Option name="Auto"/>
</Feature>
</PrintCapabilities>

```

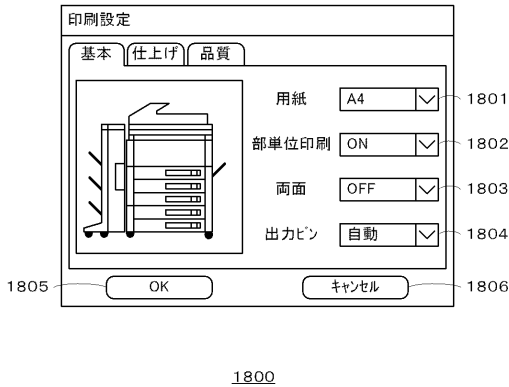
【図 13】

機能名(Feature)	コントロールID	UI文字列
Collate	ID_COLLATE	部単位印刷
Duplex	ID_Duplex	両面
MediaSize	ID_MediaSize	用紙名称
OutputBin	ID_OutputBin	排紙ビン

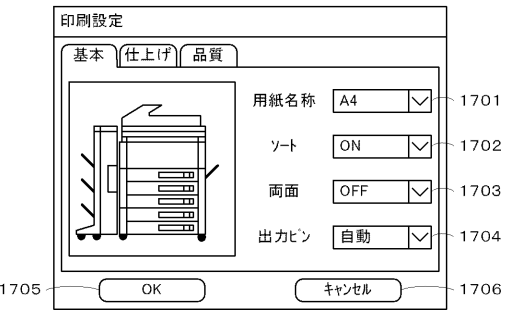
【図 14】

機能名(Feature)	コントロールID	UI文字列
Collate	ID_COLLATE	部単位印刷
Duplex	ID_Duplex	両面
MediaSize	ID_MediaSize	用紙
OutputBin	ID_OutputBin	排紙ビン

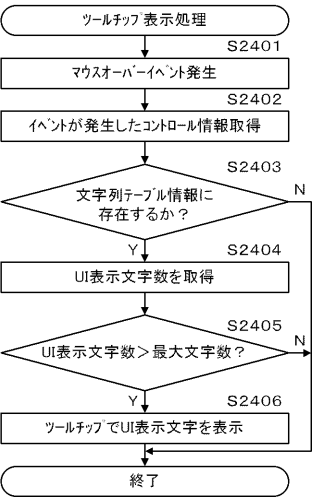
【図 1 5】



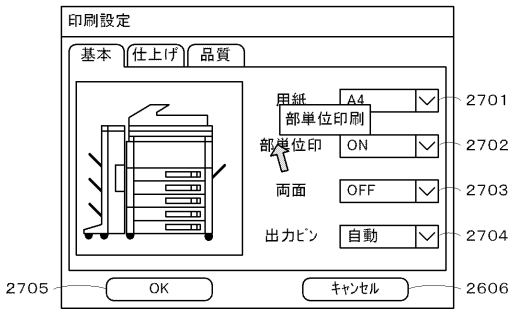
【図 1 6】



【図 2 0】



【図 2 1】



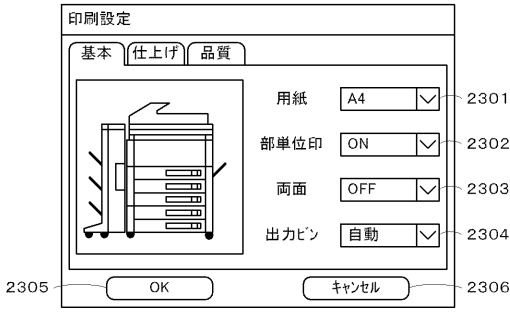
【図 1 7】

機能名(Feature)	コントロールID	UI文字列	最大表示文字数
Collate	ID_COLLATE	ソート	4
Duplex	ID_Duplex	両面	4
MediaSize	ID_MediaSize	用紙名称	4
OutputBin	ID_OutputBin	排紙ビン	4

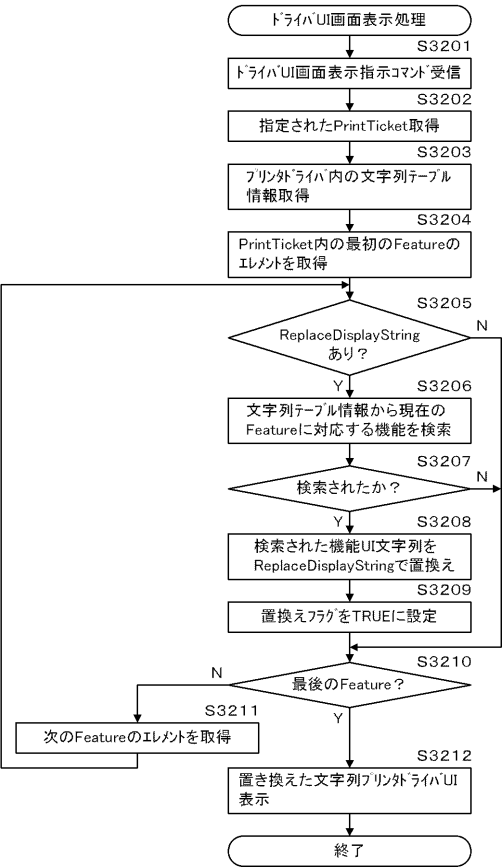
【図 1 8】

機能名(Feature)	コントロールID	UI文字列	最大表示文字数
Collate	ID_COLLATE	部単位印刷	4
Duplex	ID_Duplex	両面	4
MediaSize	ID_MediaSize	用紙	4
OutputBin	ID_OutputBin	排紙ビン	4

【図 1 9】



【図 2 2】



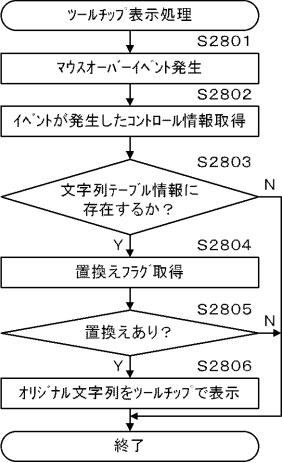
【図 2 3】

2901				
機能名 (Feature)	コントロールID	UI文字列	置換えフラグ	オリジナル文字列
Collate	ID_COLLATE	ソート	TRUE	ソート
Duplex	ID_Duplex	両面	FALSE	両面
MediaSize	ID_MediaSize	用紙名称	TRUE	用紙名称
OutputBin	ID_OutputBin	排紙ビン	FALSE	排紙ビン

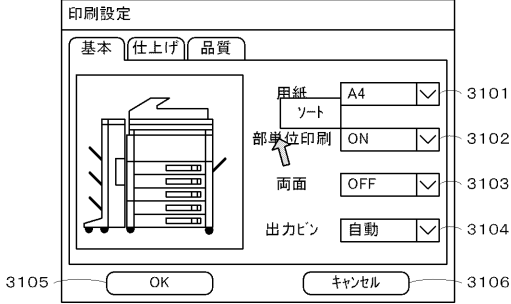
【図 2 4】

3001				
機能名 (Feature)	コントロールID	UI文字列	置換えフラグ	オリジナル文字列
Collate	ID_COLLATE	部単位印刷	TRUE	ソート
Duplex	ID_Duplex	両面	FALSE	両面
MediaSize	ID_MediaSize	用紙	TRUE	用紙名称
OutputBin	ID_OutputBin	排紙ビン	FALSE	排紙ビン

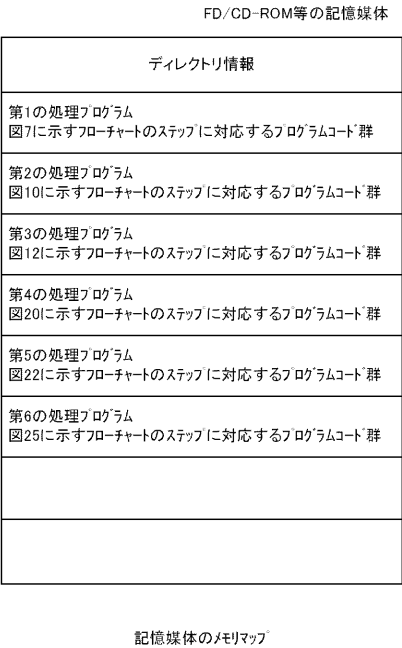
【図 2 5】



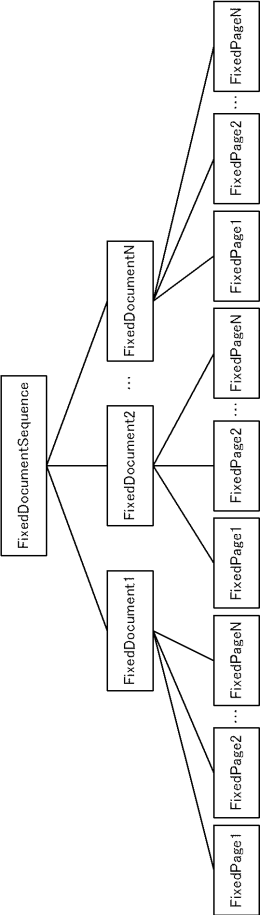
【図 2 6】



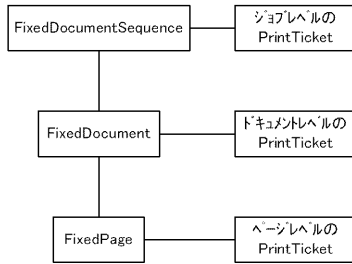
【図 2 7】



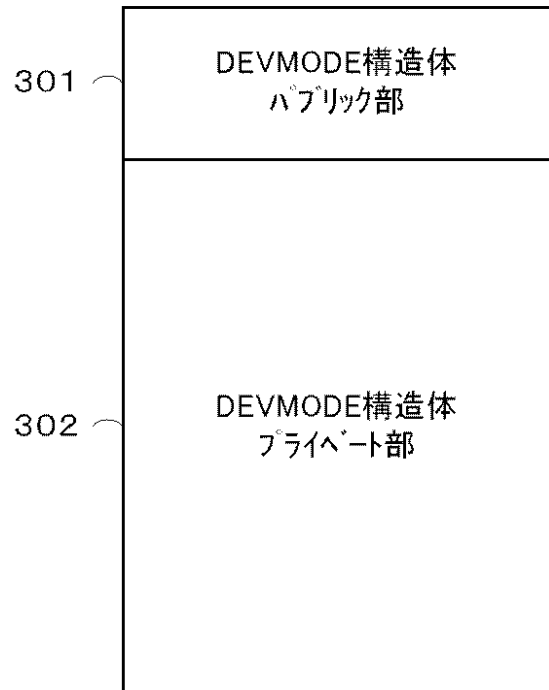
【図 2 8】



【図 29】



【図 30】



【図 31】

```

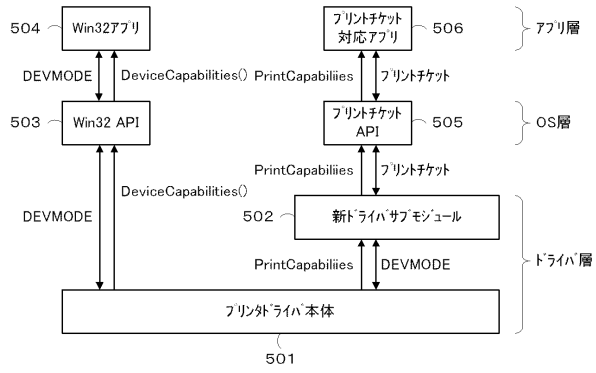
typedef struct _devicemode {
    TCHAR    dnDeviceName[32]; // ドライバがサポートするデバイス名
    WORD     dnSpecVersion;    // 構造体の基準になった初期化データ仕様のバージョン番号
    WORD     dnDriverVersion;  // フリ外ライハのバージョン番号
    WORD     dnSize;           // フライヘッドドライバデータを除いたサイズ(バイト単位)
    WORD     dnDriverExtra;    // この構造体に続くドライバデータなドライバデータのバイト長
    DWORD    dnFields;         // 以降のメンバで、初期化されているメンバを指定
    Short    dnOrientation;    // 用紙の方向
    Short    dnPaperSize;      // 用紙のサイズ
    Short    dnPaperLength;    // 用紙の長さ
    Short    dnPaperWidth;     // 用紙の幅
    Short    dnScale;          // スケーリング係数
    Short    dnCopies;         // コピー数
    Short    dnDefaultSource;   // 予約
    Short    dnPrintQuality;    // フリタの解像度
    Short    dnColor;          // カラープリンタで、カラーとモノクロを切り替え
    Short    dnDuplex;         // 画面印刷
    Short    dnResolution;     // y方向の解像度
    Short    dnTTOption;       // TrueTypeフォントの印刷方法
    Short    dnCollate;        // 複写部数を印刷するとき、ページ順にそろえるかどうかを指定
    TCHAR    dnFormName[32];   // 使用するフォーム名
    WORD     dnUnusedPadding;  // 構造体をDWORD境界に揃えるために使います
    USHORT   dnBitsPerPel;     // テクスチャレイトのカラー解像度(ピクセル当りのビット数)
    DWORD    dnPelisWidth;     // 可視のデバイス表面の幅(ピクセル単位)
    DWORD    dnPelisHeight;    // 可視のデバイス表面の高さ(ピクセル単位)
    DWORD    dnDisplayFlags;   // デバイスのディスプレイモード
    DWORD    dnDisplayFrequency; // 特定のモードでのディスプレイの周波数
} DEVMODE;
  
```

【図 32】

```

1301 <PrintCapabilities>
1302 <Feature name="Collate">
1303 <Property name="DisplayName">
    <psf:Value xsi:type="xs:string">ソート</psf:Value>
  </Property>
1304 <Option name="On"/>
    <Option name="OFF"/>
  </Feature>
  <Feature name="Duplex">
    <Property name="DisplayName">
      <psf:Value xsi:type="xs:string">両面</psf:Value>
    </Property>
    <Option name="On"/>
      <Option name="OFF"/>
    </Feature>
    <Feature name="MediaSize">
      <Property name="DisplayName">
        <psf:Value xsi:type="xs:string">出力用紙</psf:Value>
      </Property>
      <Option name="A4"/>
      <Option name="A3"/>
      <Option name="Letter"/>
      <Option name="B5"/>
    </Feature>
    <Feature name="OutputBin">
      <Property name="DisplayName">
        <psf:Value xsi:type="xs:string">排紙</psf:Value>
      </Property>
      <Option name="Auto"/>
      <Option name="Tray1"/>
      <Option name="Tray2"/>
    </Feature>
  </PrintCapabilities>
  
```

【図 33】



フロントページの続き

- (56)参考文献 特開2006-026972 (JP, A)
特開2001-001596 (JP, A)
特開2007-272857 (JP, A)
特開2007-272870 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 3/12