

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3962559号

(P3962559)

(45) 発行日 平成19年8月22日(2007.8.22)

(24) 登録日 平成19年5月25日(2007.5.25)

(51) Int. Cl.

F I

G 0 6 F 3 / 1 2 (2 0 0 6 . 0 1)
B 4 1 J 5 / 4 4 (2 0 0 6 . 0 1)

G O 6 F 3 / 1 2 G
 B 4 1 J 5 / 4 4

請求項の数 18 (全 24 頁)

(21) 出願番号 特願2001-197286 (P2001-197286)
 (22) 出願日 平成13年6月28日(2001.6.28)
 (65) 公開番号 特開2003-15849 (P2003-15849A)
 (43) 公開日 平成15年1月17日(2003.1.17)
 審査請求日 平成16年6月11日(2004.6.11)

(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 大村 宏
 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

最終頁に続く

(54) 【発明の名称】 印刷制御装置及び印刷制御方法

(57) 【特許請求の範囲】

【請求項1】

アプリケーションからの印刷命令を所定の印刷装置に適した制御コマンドに変換して出力する印刷制御装置であって、

前記印刷命令を前記制御コマンドとは異なる形式を有する中間データに変換し、記憶する中間データ生成手段と、

前記中間データを前記制御コマンドに変換して出力する制御コマンド生成手段とを有し

、
 前記中間データ生成手段が、

前記印刷命令に埋め込みフォントが含まれているか否かを判断するフォント判定手段 10
 と、

前記埋め込みフォントが含まれていると判定された場合、前記埋め込みフォントを取得し、前記中間データとともに記憶するフォント記憶手段と、

前記中間データに記憶された埋め込みフォントをグラフィックエンジンに登録する登録手段とを有し、

前記制御コマンド生成手段が登録手段によりグラフィックエンジンに登録された埋め込みフォントを用いて前記中間データを前記制御コマンドに変換することを特徴とする印刷制御装置。

【請求項2】

前記制御コマンド生成手段が、前記記憶された埋め込みフォントを用いる際、前記フォ 20

ント記憶手段が記憶した際のフォント名と異なるフォント名を有するフォントとして用いることを特徴とする請求項 1 記載の印刷制御装置。

【請求項 3】

前記中間データ生成手段が、前記フォント記憶手段が前記埋め込みフォントの取得に失敗した場合、前記埋め込みフォントをビットマップ置き換えして前記中間データを生成することを特徴とする請求項 1 又は請求項 2 記載の印刷制御装置。

【請求項 4】

前記フォント記憶手段が前記埋め込みフォントの取得に失敗した場合、前記制御コマンド生成手段が、前記埋め込みフォントを別のフォントに置き換えて前記制御コマンドを生成することを特徴とする請求項 1 又は請求項 2 記載の印刷制御装置。

10

【請求項 5】

アプリケーションからの印刷命令を所定の印刷装置に適した制御コマンドに変換して出力する印刷制御方法であって、

前記印刷命令を前記制御コマンドとは異なる形式を有する中間データに変換し、記憶する中間データ生成ステップと、

前記中間データを前記制御コマンドに変換して出力する制御コマンド生成ステップとを有し、

前記中間データ生成ステップが、

前記印刷命令に埋め込みフォントが含まれているか否かを判断するフォント判定ステップと、

20

前記埋め込みフォントが含まれていると判定された場合、前記埋め込みフォントを取得し、前記中間データとともに記憶するフォント記憶ステップと、
前記中間データに記憶された埋め込みフォントをグラフィックエンジンに登録する登録ステップとを有し、

前記制御コマンド生成ステップが前記登録ステップによりグラフィックエンジンに登録された埋め込みフォントを用いて前記中間データを前記制御コマンドに変換することを特徴とする印刷制御方法。

【請求項 6】

前記制御コマンド生成ステップが、前記記憶された埋め込みフォントを用いる際、前記フォント記憶ステップが記憶した際のフォント名と異なるフォント名を有するフォントとして用いることを特徴とする請求項 5 記載の印刷制御方法。

30

【請求項 7】

前記中間データ生成ステップが、前記フォント記憶ステップが前記埋め込みフォントの取得に失敗した場合、前記埋め込みフォントをビットマップ置き換えして前記中間データを生成することを特徴とする請求項 5 又は請求項 6 記載の印刷制御方法。

【請求項 8】

前記フォント記憶ステップが前記埋め込みフォントの取得に失敗した場合、前記制御コマンド生成ステップが、前記埋め込みフォントを別のフォントに置き換えて前記制御コマンドを生成することを特徴とする請求項 5 又は請求項 6 記載の印刷制御方法。

【請求項 9】

コンピュータを請求項 1 乃至請求項 4 のいずれか 1 項に記載の印刷制御装置として機能させることを特徴とするコンピュータプログラム。

40

【請求項 10】

請求項 9 記載のコンピュータプログラムを格納したコンピュータ読みとり可能な記憶媒体。

【請求項 11】

文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納するスプーラと、

前記スプールファイルに埋め込まれたグラフィックエンジンにフォントに登録するデス

50

プーラと、

前記デスクプーラにより登録されたフォントに基づき文字グリフの取得を行い、前記文字描画を表す情報をPDLに変換するドライバとを有することを特徴とする印刷制御装置。

【請求項12】

前記デスクプーラは、フォントを登録する際にフェイス名を変更することを特徴とする請求項11記載の印刷制御装置。

【請求項13】

文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納する格納ステップと、

前記スプールファイルに埋め込まれたフォントをグラフィックエンジンに登録する登録ステップと、

登録されたフォントに基づき文字グリフの取得を行い、前記文字描画を表す情報をPDLに変換する変換ステップとを有することを特徴とする印刷制御方法。

【請求項14】

前記登録ステップは、フォントを登録する際にフェイス名を変更することを特徴とする請求項13記載の印刷制御方法。

【請求項15】

文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納する格納工程のプログラムと、

前記スプールファイルに埋め込まれたフォントをグラフィックエンジンに登録する登録工程のプログラムと、

登録されたフォントに基づき文字グリフの取得を行い、前記文字描画を表す情報をPDLに変換する変換工程のプログラムとを有することを特徴とするコンピュータプログラム。

【請求項16】

前記登録工程のプログラムは、フォントを登録する際にフェイス名を変更することを特徴とする請求項15記載のプログラム。

【請求項17】

文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納する格納工程のプログラムと、

前記スプールファイルに埋め込まれたフォントをグラフィックエンジンに登録する登録工程のプログラムと、

登録されたフォントに基づき文字グリフの取得を行い、前記文字描画を表す情報をPDLに変換する変換工程のプログラムとを記憶するコンピュータ読みとり可能な記憶媒体。

【請求項18】

前記登録工程のプログラムは、フォントを登録する際にフェイス名を変更することを特徴とする請求項17記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は、プリンタ等の印刷装置に印刷用の制御コマンドを供給する印刷制御装置及び印刷制御方法に関し、特に、アプリケーションプログラムからの印刷命令を一旦、最終的に前記プリンタに送付する印刷データとは異なる形式のデータ形式（いわゆる中間コード）に変換した後、再度制御コマンドに変換する形式の印刷制御装置及び印刷制御方法に関する。

【0002】

【従来の技術】

10

20

30

40

50

図3は、従来、直接、あるいはネットワーク経由で接続されているプリンタ1500に対して印刷指示を行う印刷制御装置としてのホストコンピュータにおいて、印刷処理に関わる機能構成を大まかに示した図である。印刷処理は、ホストコンピュータのユーザが、ワードプロセッサプログラムや表計算プログラム等のアプリケーション201から、印刷指示を行うことによって開始する。

【0003】

アプリケーション201は、文字描画データ等を含むアプリケーションデータを解析して、例えばOS(基本ソフト)が提供するグラフィックエンジン202をコールする。グラフィックエンジン202は、プリンタ1500の種類に合わせて用意されたプリンタドライバ203をロードし、アプリケーション201の出力をプリンタドライバ203に受け渡す。そして、アプリケーション201から受け取るGDI(Graphic Device Interface)関数からDDI(Device Driver Interface)関数に変換して、プリンタドライバ203へDDI関数を出力する。

10

【0004】

プリンタドライバ203は、グラフィックエンジン202から受け取ったDDI関数に基づいて、プリンタが認識可能な制御コマンド、例えばPDL(Page Description Language)に変換する。変換されたプリンタ制御コマンドは、システムスプーラ204を経てインターフェース21経由でプリンタ1500へ印刷データとして出力される。プリンタ1500ではこの制御コマンドを解釈して、ビットマップデータへ展開し、最終的に紙などの記録媒体に印刷結果が出力される。

20

【0005】

図5は、印刷処理における文字描画に必要な情報を表す模式図である。図5に示すように、一般的に文字描画では以下の情報が必要である。

・文字コード

文字描画で伝えたいところの文字情報。ASCII、ShiftJIS、Unicode等、予め定められたコード体系に従ったコードが用いられる。

・フォント属性

文字描画に適應するフェイス名、グリフへの修飾を表すイタリック/ボールド指定有無、文字の大きさを表すポイント数等の情報。

・フォント

ゴシックや明朝等のフェイス名で指定される文字描画形状を表す情報。内部に文字毎の描画形状情報(グリフと以下呼ぶ)を有する。フォントの情報形式としては、大きくビットマップフォントとアウトラインフォントに分かれ、TrueType、OpenType等の各種フォーマットが定義されている。

30

・描画属性

文字の描画位置や、色、クリップの状態等を表す情報。

【0006】

そして、これらの情報全てが揃わなければ、ユーザが意図した通りの文字描画は成立しない。しかし、文字描画形状を表す情報であるフォントは、例えば著作権の関係等によって、すべての環境に必要とするフォントが存在する保証はない。言い換えるとユーザが文字描画データを作成した環境に存在したフォントが、文字描画が行われる環境で存在しない可能性がある。描画が行われる環境にユーザが意図したフォントが存在しない場合にも、ユーザが意図したフォントもしくはそれに近い形状のフォントで文字描画を行うために、従来、1)フォント置き換え、2)ビットマップ置き換え及び、3)フォント埋め込みという仕組みが知られている。以下、それぞれの仕組みを用いた場合の描画結果を示す図6を参照して、それぞれの仕組みについて説明する。

40

【0007】

1. フォント置き換え

文字描画実行環境において、文字描画データ中のフォント属性に近い別のフォントを用いて描画を行う方法。別のフォントを用いるため、図6右最上段に示すように、ユーザが意

50

図していたものとは全く違った形状の文字描画が行われる可能性がある。最悪の場合、意図した文字コードに対応したフォントが文字描画実行環境において存在せず、文字として識別不可能な描画（印刷）が行われることもある。

【 0 0 0 8 】

2 . ビットマップ置き換え

文字描画データを作成する時点で文字描画をビットマップに変換して、文字描画をビットマップ描画に変換する方法。通常の見た目はユーザが意図した文字描画と同じになる。しかしビットマップであるために図 6 右中段に示すように拡大 / 縮小時にビットマップの解像度の影響で文字品位が落ちる場合がある。

【 0 0 0 9 】

3 . フォント埋め込み

文字描画データに対してフォントを埋め込み、文字描画が実際行われる環境までフォントを持っていく方法。埋め込まれたフォントは、文字描画前に文字描画実行環境へ登録されるため、すべての情報が揃いユーザが意図した形式で文字描画される。注意として登録の際には著作権を侵害しないように配慮する必要がある。例えば埋め込みフォントが描画する環境に元々存在しない場合には、印刷終了後に登録したフォントを必ず消去する必要がある。フォント埋め込みは指定されたフォントが文字描画領域になくてもユーザが意図した出力を得られるという利点があるが、フォント情報を文字描画データに埋め込むため、文字描画データが大きくなる傾向にある。特に日本語フォントのような文字種の多いフォントを埋め込む場合にこの問題が顕著である。

【 0 0 1 0 】

図 7 に示すシーケンスチャートを用いて、フォントが埋め込みされた文字描画データを印刷する際の従来のアプリケーション 2 0 1、グラフィックエンジン 2 0 2、及びプリンタドライバ 2 0 3 の動作について説明する。

【 0 0 1 1 】

ステップ A 1 . アプリケーション 2 0 1 は印刷ジョブが開始することを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する（グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してプリンタドライバ 2 0 3 へ引き渡す）。

ステップ D R V 1 . プリンタドライバ 2 0 3 は、ジョブ開始制御の P D L コマンドを生成する。

【 0 0 1 2 】

ステップ A 2 . アプリケーション 2 0 1 はページ印刷が開始することを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する（グラフィックエンジン 2 0 2 は A 1 . と同様にプリンタドライバ 2 0 3 へ引き渡す）。

ステップ D R V 2 . プリンタドライバ 2 0 3 は、ページ開始制御の P D L コマンドを生成する。

【 0 0 1 3 】

ステップ A 3 . アプリケーション 2 0 1 は、アプリケーションデータ内の埋め込みフォントをグラフィックエンジン 2 0 2 に登録する。この登録により文字描画に不可欠なフォントの情報をこの環境下でも使用可能にすることができる。ステップ G 1 . グラフィックエンジン 2 0 2 は、フォント情報を保持する。

ステップ A 4 . アプリケーション 2 0 1 がステップ A 3 . で登録したフォントを用いた G D I 関数によって文字描画を行う（グラフィックエンジン 2 0 2 はステップ A 1 . と同様にプリンタドライバ 2 0 3 へ引き渡す）。

【 0 0 1 4 】

ステップ D R V 3 . プリンタドライバ 2 0 3 は、文字描画コマンドを生成するために、以下の処理を行い文字のグリフを取得する。

ステップ D R V 3 . 1 . プリンタドライバ 2 0 3 は、グラフィックエンジン 2 0 2 に対して文字グリフの取得要求を行う。

【 0 0 1 5 】

10

20

30

40

50

ステップG2. グラフィックエンジン202はステップG1.で登録されたフォントより文字グリフを生成しプリンタドライバ203へ受け渡す。

ステップA5. アプリケーション201はステップA3で登録したフォントをグラフィックエンジン202より削除する。

ステップG3. グラフィックエンジン202で保持していた埋め込みフォントを破棄する。

【0016】

ステップA6. アプリケーション201はページ印刷が終了したことを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202はステップA1.と同様にプリンタドライバ203へ引き渡す)。

10

ステップDRV4. プリンタドライバ203は、ページ終了制御のPDLコマンドを生成する。

【0017】

ステップA7. アプリケーション201は印刷ジョブが終了することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202はステップA1.と同様にプリンタドライバ203へ引き渡す)。

ステップDRV5. プリンタドライバ203は、ジョブ終了制御のPDLコマンドを生成する。

【0018】

上記の処理は1つのプロセスで行われており、アプリケーション201がGDI関数をコールすることにより、コールされたGDI関数の機能を実現するために必要なグラフィックエンジン202及びプリンタドライバ203の処理ルーチンが動作し、コール内容に対応するPDLコマンドが生成される。

20

【0019】

アプリケーション201のGDI関数コールとグラフィックエンジン202-プリンタドライバ203の処理が完全にリンクして動作しているので、アプリケーション201-グラフィックスエンジン202-プリンタドライバ203間での動作手順は保証される。そのためプリンタドライバ203での処理ステップであるステップDrv3.1.文字グリフ取得の際に、グラフィックエンジン202でフォントを保持していることが保証されている。

30

【0020】

図4は、図3に示した構成に加え、プリンタに送信される印刷データを生成する前に、一旦、最終的にプリンタに送付する印刷データとは異なる形式のデータ形式(いわゆる中間コード)での一時保存を行うスプーラを備え、この中間コード形式で一時保存されたデータから改めて最終的に前記プリンタに送付する印刷データを生成するデスクトップ及びその関連機能を備えるホストコンピュータの機能ブロック図を示す。すなわち、図4のホストコンピュータでは、グラフィックエンジン202からプリンタドライバ203へ印刷命令を送る際に、一旦、中間コードからなるスプールファイル303を生成し、その後デスクトップによってスプールファイルを印刷データに変換してプリンタに出力する。

【0021】

40

上述した図3のホストコンピュータにおいて、アプリケーション201が印刷処理から開放されるのは、プリンタドライバ203がグラフィックエンジン202からのすべての印刷命令をプリンタ1500の制御コマンドへ変換し終わった時点である。これに対して、図4のホストコンピュータでは、スプーラ302がすべての印刷命令を中間コードデータに変換し、スプールファイル303に出力した時点でアプリケーション201が印刷処理から開放される。通常、後者(図4構成)の方が短時間でアプリケーション201が印刷処理から解放される。

【0022】

また、図4で示す構成を用いた場合、スプールファイル303の内容に対して印刷前に加工することができる。これにより拡大/縮小印刷や、複数(N)ページを1ページに縮小

50

して印刷するNアップ印刷等、アプリケーションの持たない機能を実現する事ができる。尚、スプールファイル303の加工を行うためには、通常、プリンタドライバ203が提供するウィンドウから設定を行い、プリンタドライバ203がその設定内容をRAMあるいはHD等のメモリ上に保管する。

このような利点から、上記図3に示した構成から、図4の如く中間コードデータでスプールするような構成への移行が進んでいる。

【0023】

次に、図4に示す機能構成を有するホストコンピュータにおいて、フォント埋め込みデータを印刷する際の処理を図8を参照して説明する。この場合、図7で説明した図3の機能構成の際の処理とは異なり、1つのプロセスでは処理できない。そして、複数のジョブを制御するために、アプリケーションプロセスとデスプーラプロセス間での待ち行列処理を実現する必要がある。実際には上記2つのプロセス間の通信はスプールファイルマネージャ304を介して行われるが、説明を簡潔に行うために待ち行列処理を含めスプールファイルマネージャ304を一つの通信媒介だと捉え、図8及び以下の説明からはあえて省略した。

10

【0024】

ステップA1. アプリケーション201は印刷ジョブが開始することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

20

ステップS1. スプーラ302は用紙サイズ等のジョブに関する情報を保持するジョブファイルのスプールファイル303に生成し、別プロセスであるデスプーラ305を起動する。

【0025】

ステップD1. デスプーラ305はジョブファイルを読み印刷ジョブ開始のためのGDI関数をコールする(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

ステップDRV1. プリンタドライバ203は、ジョブ開始制御のPDLコマンドを生成する。

30

【0026】

ステップA2. アプリケーション201はページ印刷が開始することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

ステップS2. スプーラ302はページに関する情報を保持するページファイルのスプールファイル303に生成する。

【0027】

ステップA3. アプリケーション201は、アプリケーションデータ内の埋め込みフォントをグラフィックエンジン202に登録する。この登録により文字描画に不可欠なフォントの情報をこの環境下でも使用可能にすることができる。

40

ステップG1. グラフィックエンジン202は、フォント情報を保持する。

【0028】

ステップA4. アプリケーション201がステップA3で登録したフォントを含むフォントを用いて文字描画を行う(グラフィックエンジン202は描画時にコールされるGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

ステップS3. スプーラ302は、文字描画を表す情報を中間コードとしてスプールファイルのページファイルに書き込む。この中間コードにはDDI関数より取得可能な“文字コード、フォント属性、描画属性”のみ含まれており、フォントは含まれていない。

50

【 0 0 2 9 】

ステップ A 5 . アプリケーション 2 0 1 はステップ A 3 で登録したフォントをグラフィックエンジン 2 0 2 より削除する。

ステップ G 3 . グラフィックエンジン 2 0 2 で保持していた埋め込みフォントを破棄する。

【 0 0 3 0 】

ステップ A 6 . アプリケーション 2 0 1 はページ印刷が終了したことを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をスプーラ 3 0 2 へ引き渡す)。

10

ステップ S 4 . スプーラ 3 0 2 は、ページファイルをクローズし、ページ再生をデスプーラ 3 0 5 に依頼する。

【 0 0 3 1 】

ステップ A 7 . アプリケーション 2 0 1 は印刷ジョブが終了することを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をスプーラ 3 0 2 へ引き渡す)。

ステップ S 4 . スプーラ 3 0 2 は、ジョブファイルをクローズし、これ以上ページが増えないことをデスプーラ 3 0 5 に通知する。

【 0 0 3 2 】

20

ステップ D 2 . デスプーラ 3 0 5 が、ステップ S 2 で生成されたページファイルに記述された中間コードを再生するために、ページ開始のための G D I 関数をコールする (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をプリンタドライバ 2 0 3 へ引き渡す)。

【 0 0 3 3 】

ステップ D R V 2 . プリンタドライバ 2 0 3 は、ページ開始制御の P D L コマンドを生成する。

ステップ D 3 . デスプーラ 3 0 5 が、中間コードに記述された文字描画を再生するために、G D I 関数をコールする (グラフィックエンジン 2 0 2 は描画時にコールされる G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をプリンタドライバ 2 0 3 へ引き渡す)。

30

【 0 0 3 4 】

ステップ D R V 3 . プリンタドライバ 2 0 3 は、文字描画コマンドを生成するために、以下の処理を行い文字のグリフを取得する。

ステップ D R V 3 . 1 . プリンタドライバ 2 0 3 は、グラフィックエンジンに対して文字グリフの取得を行う。

【 0 0 3 5 】

ステップ G 4 . グラフィックエンジン 2 0 2 は、フォント属性に指定されているフェース名のフォントを探す。しかし (埋め込みされ、かつ当該環境に元々は存在しない) フォントは既にステップ G 3 . で廃棄されており、該当フォントのグリフを要求されても受け渡すことはできない。よって予め定められた関係に基づきグリフを要求されたフォントに近い種類の代替えフォントを用いてグリフを作成し、その結果をプリンタドライバ 2 0 3 へ受け渡す。

40

【 0 0 3 6 】

ステップ D 6 . デスプーラ 3 0 5 はページ印刷が終了したことを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をプリンタドライバ 2 0 3 へ引き渡す)。

ステップ D R V 4 . プリンタドライバ 2 0 3 は、ページ終了制御の P D L コマンドを生

50

成する。

【 0 0 3 7 】

ステップ D 7 . デスプーラ 3 0 5 は印刷ジョブが終了することを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をプリンタドライバ 2 0 3 へ引き渡す)。

ステップ D R V 5 . プリンタドライバ 2 0 3 は、ジョブ終了制御の P D L コマンドを生成する。

【 0 0 3 8 】

上記の処理は図 8 に示すようにアプリケーションプロセスとデスプーラプロセスの 2 つのプロセスで行われており、アプリケーション 2 0 1 がグラフィックエンジン 2 0 2 に登録した埋め込みフォントが、デスプーラによるスプールファイルの再生時にもグラフィックエンジン 2 0 2 に存在することは保証されていない。

10

【 0 0 3 9 】

グラフィックエンジン 2 0 2 に埋め込みフォントが存在しない場合は、ステップ G 4 . で説明したようにフォント置き換えが発生し、中間データで指定されていたフォントとは異なるフォントに置き換えられてしまうため、図 6 右最上段に示したように、ユーザが意図した形状とは異なる形状で出力されてしまう可能性がある。

この問題を回避するため、ステップ G 4 . でのフォント置き換えを防止するため、スプール時点でビットマップ置き換えを行う方法が考えられる。

20

【 0 0 4 0 】

このようなスプール時にビットマップ置き換えを行う印刷処理として考えられる動作を、図 9 を参照して説明する。

ステップ A 1 . アプリケーション 2 0 1 は印刷ジョブを開始することを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をスプーラ 3 0 2 へ引き渡す)。

【 0 0 4 1 】

ステップ S 1 . スプーラ 3 0 2 は用紙サイズ等のジョブに関する情報を保持するジョブファイルのスプールファイル 3 0 3 に生成し、別プロセスであるデスプーラ 3 0 5 を起動する。

30

ステップ D 1 . デスプーラ 3 0 5 はジョブファイルを読み印刷ジョブ開始のための G D I 関数をコールする (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をプリンタドライバ 2 0 3 へ引き渡す)。

【 0 0 4 2 】

ステップ D R V 1 . プリンタドライバ 2 0 3 は、ジョブ開始制御の P D L コマンドを生成する。

ステップ A 2 . アプリケーション 2 0 1 はページ印刷を開始することを、G D I 関数を介してグラフィックエンジン 2 0 2 に通知する (グラフィックエンジン 2 0 2 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 3 0 1 へ出力し、ディスパッチャ 3 0 1 はその内容をスプーラ 3 0 2 へ引き渡す)。

40

【 0 0 4 3 】

ステップ S 2 . スプーラ 3 0 2 はページに関する情報を保持するページファイルのスプールファイル 3 0 3 に生成する。

ステップ A 3 . アプリケーション 2 0 1 は、アプリケーションデータ内の埋め込みフォントをグラフィックエンジン 2 0 2 に登録する。この登録により文字描画に不可欠なフォントの情報をこの環境下でも使用可能にすることができる。

【 0 0 4 4 】

ステップ G 1 . グラフィックエンジン 2 0 2 は、フォント情報を保持する。

50

ステップA4. アプリケーション201がステップA3で登録したフォントを含むフォントを用いて文字描画を行う(グラフィックエンジン202は描画時にコールされるGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

【0045】

ステップS3. スプーラ302は、文字描画を表す情報を中間コードとしてスプールファイルのページファイルに書き込む。この中間コードに含まれるフォント属性が埋め込みフォントであった場合は、フォント置き換えによる文字描画の差異を無くすために、以下の処理を行う。

ステップS3.1. スプーラ302は、グラフィックエンジン202に対して文字グリフの取得を行う。

10

【0046】

ステップG2. グラフィックエンジン202はフォントより文字グリフを生成しスプーラ302へ受け渡す。

ステップS3.2. スプーラ302は受け取った文字グリフを用いて文字描画をビットマップ描画に変換し、中間コードとしてスプールする。

【0047】

ステップA5. アプリケーション201はステップA3で登録したフォントをグラフィックエンジン202より削除する。

ステップG3. グラフィックエンジン202で保持していた埋め込みフォントを破棄する。

20

【0048】

ステップA6. アプリケーション201はページ印刷が終了したことを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

ステップS4. スプーラ302は、ページファイルをクローズし、ページ再生をデスプーラ305に依頼する。

【0049】

ステップA7. アプリケーション201は印刷ジョブが終了することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

30

ステップS4. スプーラ302は、ジョブファイルをクローズし、これ以上ページが増えないことをデスプーラ305に通知する。

【0050】

ステップD2. デスプーラ305が、ページファイルに記述された中間コードを再生するために、ページ開始のためのGDI関数をコールする(グラフィックエンジン202はコールされたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

40

【0051】

ステップDRV2. プリンタドライバ203は、ページ開始制御のPDLコマンドを生成する。

ステップD8. デスプーラ305は、ページファイル上の文字描画から変換したビットマップ描画の中間コードをGDI関数コールに変換する(グラフィックエンジン202はコールされたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

【0052】

ステップDRV6. プリンタドライバ203は、ビットマップ描画をPDLコマンドに変換する。

50

ステップD6. デスプーラ305はページ印刷が終了したことを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

【0053】

ステップDRV4. プリンタドライバ203は、ページ終了制御のPDLコマンドを生成する。

ステップD7. デスプーラ305は印刷ジョブが終了することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

10

ステップDRV5. プリンタドライバ203は、ジョブ終了制御のPDLコマンドを生成する。

【0054】

図9を用いて説明したような処理により、プリンタドライバ203から(元々は存在しない)埋め込みフォントのグリフ要求があったとしても、グラフィックエンジンによるフォント置き換えが発生しないので、基本的にはユーザが意図した通りの形状を有する文字出力が得られる。しかし、文字描画をビットマップ描画に変換することにより、以下のような問題点を有する。

【0055】

20

1. 拡大時の品質低下

図6右中段に示したように、スプール後に拡大処理を行った場合、文字の品位が低下する。

2. 色処理の問題

ビットマップに変換された時点で文字描画に関する情報がなくなるため、カラー印刷時に行われる色変換処理の際に、文字描画としての色処理が適応できない。

3. 圧縮の問題

文字グリフのビットマップに対して効果がある圧縮処理が適応できない。

【0056】

【発明が解決しようとする課題】

30

上述したように、プリンタに送信される印刷データを生成する前に、最終的にプリンタに送付する印刷データとは異なる形式のデータ形式(いわゆる中間コード)での一時保存を行うスプーラを備え、この中間コード形式で一時保存されたデータから改めて最終的にプリンタに送付する印刷データを生成するデスプーラとプリンタ制御コマンドを生成するプリンタドライバを備える従来の印刷制御装置においては、埋め込みフォントを用いた文字描画の際に、フォント置き換え又はビットマップ置き換えが発生し、フォント置き換えでは意図した出力結果が得られない、ビットマップ置き換えでは、印字品質の低下、色処理、圧縮処理での問題が発生するという問題があった。

【0057】

本発明はこのような従来技術の問題点を解決し、印刷命令を一旦中間コード形式に変換した後プリンタ制御コマンドに変換する形式の印刷制御装置及び印刷制御方法において、埋め込みフォントを正しく印刷可能とする印刷制御装置及び印刷制御方法を提供することを目的とする。

40

【0058】

【課題を解決するための手段】

すなわち、本発明の要旨は、アプリケーションからの印刷命令を所定の印刷装置に適した制御コマンドに変換して出力する印刷制御装置であって、印刷命令を制御コマンドとは異なる形式を有する中間データに変換し、記憶する中間データ生成手段と、中間データを制御コマンドに変換して出力する制御コマンド生成手段とを有し、中間データ生成手段が、印刷命令に埋め込みフォントが含まれているか否かを判断するフォント判定手段と、埋

50

め込みフォントが含まれていると判定された場合、埋め込みフォントを取得し、中間データとともに記憶するフォント記憶手段と、中間データに記憶された埋め込みフォントをグラフィックエンジンに登録する登録手段とを有し、制御コマンド生成手段が登録手段によりグラフィックエンジンに登録された埋め込みフォントを用いて中間データを制御コマンドに変換することを特徴とする印刷制御装置に存する。

【 0 0 5 9 】

また、本発明の別の要旨は、アプリケーションからの印刷命令を所定の印刷装置に適した制御コマンドに変換して出力する印刷制御方法であって、印刷命令を制御コマンドとは異なる形式を有する中間データに変換し、記憶する中間データ生成ステップと、中間データを制御コマンドに変換して出力する制御コマンド生成ステップとを有し、中間データ生成ステップが、印刷命令に埋め込みフォントが含まれているか否かを判断するフォント判定ステップと、埋め込みフォントが含まれていると判定された場合、埋め込みフォントを取得し、中間データとともに記憶するフォント記憶ステップと、中間データに記憶された埋め込みフォントをグラフィックエンジンに登録する登録ステップとを有し、制御コマンド生成ステップが登録ステップによりグラフィックエンジンに登録された埋め込みフォントを用いて中間データを制御コマンドに変換することを特徴とする印刷制御方法に存する。

10

【 0 0 6 0 】

また、本発明の別の要旨は、コンピュータを本発明の印刷制御装置として機能させることを特徴とするコンピュータプログラムに存する。

20

【 0 0 6 1 】

また、本発明の別の要旨は、コンピュータを本発明の印刷制御装置として機能させるコンピュータプログラムを格納したことを特徴とするコンピュータ読みとり可能な記憶媒体に存する。

【 0 0 6 2 】

また、本発明の別の要旨は、文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納するスプーラと、スプールファイルに埋め込まれたフォントを登録するデスプーラと、デスプーラにより登録されたフォントに基づき文字グリフの取得を行い、文字描画を表す情報を P D L に変換するドライバとを有することを特徴とする印刷制御装置に存する。

30

【 0 0 6 3 】

また、本発明の別の要旨は、文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納する格納ステップと、スプールファイルに埋め込まれたフォントを登録する登録ステップと、登録されたフォントに基づき文字グリフの取得を行い、文字描画を表す情報を P D L に変換する変換ステップとを有することを特徴とする印刷制御方法に存する。

【 0 0 6 4 】

また、本発明の別の要旨は、文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納する格納工程のプログラムと、スプールファイルに埋め込まれたフォントを登録する登録工程のプログラムと、登録されたフォントに基づき文字グリフの取得を行い、文字描画を表す情報を P D L に変換する変換工程のプログラムとを有することを特徴とするコンピュータプログラムに存する。

40

【 0 0 6 5 】

また、本発明の別の要旨は、文字描画を表す情報を中間コードとして書き込む際、フォントが埋め込みフォントの場合、フォントの取得を要求し、取得したフォントをスプールファイルに中間データとして格納する格納工程のプログラムと、スプールファイルに埋め込まれたフォントを登録する登録工程のプログラムと、登録されたフォントに基づき文字グ

50

リフの取得を行い、文字描画を表す情報をPDLに変換する変換工程のプログラムとを記憶するコンピュータ読みとり可能な記憶媒体に存する。

【0066】

【発明の実施の形態】

以下、図面を参照して本発明をその好ましい実施形態に基づいて説明する。

まず、本発明の実施形態に係る印刷制御装置について説明する前に、本発明の実施形態に係る印刷制御装置が制御可能な印刷装置の構成について説明する。

【0067】

(印刷装置の構成)

図1は、本発明の実施形態に係る印刷制御装置が制御可能な印刷装置の構成例を説明する概略断面図である。本実施形態において、印刷装置1500は例えばレーザービームプリンタであるが、もちろんインクジェット方式を始め他の印刷方式の印刷装置であってもよい。

10

【0068】

印刷装置(以下、プリンタという)1500は、直接又はネットワーク等を介して印刷制御装置から供給されるプリンタ制御コマンド等を基に、記録媒体の一例としての記録用紙上に像を形成する。プリンタ本体740は、プリンタ制御ユニット1000、操作部1501、レーザードライバ702、半導体レーザー703、回転多面鏡705、静電ドラム706、現像ユニット707、用紙カセット708、搬送ローラ710、外部メモリ711、フェイスダウン排出部715及び排紙トレイ716とを備えている。

20

【0069】

上記各部の構成を動作とともに詳述すると、プリンタ制御ユニット1000は、プリンタ本体740全体の制御を行い、主にプリンタ制御コマンドをビデオ信号に変換してレーザードライバ702に出力する。プリンタ制御ユニット1000には、フォントデータやページ記述言語(PDL)のエミュレーションプログラム等を供給する外部メモリ711を接続することもできる。操作部1501には、プリンタ1500の操作、設定等を行うためのキー及び表示手段(例えばLED表示器)等が設けられている。

【0070】

レーザードライバ702は半導体レーザー703を駆動するための回路であり、入力されたビデオ信号に応じて半導体レーザー703を駆動し、半導体レーザー703から発射されるレーザー光704をオン・オフする。半導体レーザー703は、回転多面鏡705に向けてレーザー光704を左右方向に反射し、それによってレーザー光704が静電ドラム706上を走査する。予め所定電位に帯電された静電ドラム706上をオン、オフするレーザー光704で走査することにより、プリンタ制御ユニット1000が出力するビデオ信号に対応した静電潜像がドラム表面に形成される。現像ユニット707は、静電ドラム706近傍に設けられ、現像材(トナー)を用いて静電潜像を現像(可視化)する。

30

【0071】

一方、用紙カセット708は、記録紙として例えばカットシートを収納する。給紙ローラ709及び搬送ローラ710は、給紙カセット708内のカットシート記録紙をプリンタ本体740内に搬送し、静電ドラム706の対向位置に供給する。記録紙は、用紙カセット708からだけでなく、用紙カセット708の蓋部上面に設けられた手差しトレイ(図示略)から供給することもできる。そして、ドラム上で現像された潜像(トナー像)は、転写ローラ等によって搬送されてきた記録紙上に転写される。

40

【0072】

定着部712は、カットシート記録紙に転写されたトナー像を加熱、加圧してカットシート記録紙上に定着させる。画像が形成された記録紙は、切り替えくさび713を上向きにした場合にはフェイスアップ排出部714から記録面を上にした状態で排紙トレイ716に排出され、くさび713を下向きにした場合にはフェイスダウン排出部715から記録面を下にした状態で排出される。

50

【 0 0 7 3 】

(印刷制御装置の構成例)

次に、本発明の実施形態に係る印刷制御装置の一例としてのホストコンピュータの構成例を図2のブロック図を参照して説明する。尚、以下に説明する印刷制御装置の機能が実現されるのであれば、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(Local Area Network: ローカルエリアネットワーク)、WAN(Wide Area Network: 広域ネットワーク)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構わない。

【 0 0 7 4 】

図2において、本発明の実施形態に係る印刷制御装置の一例としてのホストコンピュータ3000は、上述したプリンタ1500と接続され、印刷システムを構成している。ホストコンピュータ3000は、CPU1、RAM2、ROM3キーボードコントローラ(KBC)5、CRTコントローラ(CRTC)6、ディスクコントローラ(DKC)7、プリンタコントローラ(PRTC)8、キーボード(KB)9、CRTディスプレイ(CRT)10及び外部メモリ11とを備えている。

10

【 0 0 7 5 】

先ず、ホストコンピュータ3000各部の構成を詳述すると、CPU1はシステムバスに接続された各デバイスを統括的に制御する中央処理装置であり、ROM3のプログラム用ROM3b(後述)或いは外部メモリ11に記憶された文書処理プログラム等のアプリケーション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)等が混在した文書処理を実行する。また、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(ラスターライズ)処理を実行し、CRTディスプレイ10上でのWSYIWYG(What You See Is What You Get: CRTディスプレイ画面上に見えているそのままの大きさや形で印刷できる機能)を可能としている。

20

【 0 0 7 6 】

更に、CPU1はCRTディスプレイ10上のマウスカーソル(図示略)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。ユーザはプリンタ1500を使用して印刷する際、印刷の設定に関するウィンドウを開き、プリンタ1500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ことができるようになっている。

30

【 0 0 7 7 】

RAM2はCPU1の主メモリ、ワークエリア等として機能する。ROM3はフォント用ROM3aと、プログラム用ROM3bと、データ用ROM3cとを備えている。フォント用ROM3a或いは外部メモリ11は、上記文書処理の際に使用するフォントデータなどを記憶する。プログラム用ROM3b或いは外部メモリ11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(以下、OS)等を記憶する。データ用ROM3c或いは外部メモリ11は、上記文書処理等を行う際に使用する各種データを記憶する。

【 0 0 7 8 】

キーボードコントローラ(KBC)5は、キーボード9やポインティングデバイス(図示略)からのキー入力を制御する。CRTコントローラ(CRTC)6は、CRTディスプレイ(CRT)10の表示を制御する。ディスクコントローラ(DKC)7は、外部メモリ11とのアクセスを制御する。プリンタコントローラ(PRTC)8は、双方向性インターフェース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。キーボード9は、各種キーを備えている。

40

【 0 0 7 9 】

CRTディスプレイ(CRT)10は、図形、イメージ文字、表等を表示する。外部メモリ11はハードディスク(HD)、フロッピーディスク(FD)等から構成されており、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、プリンタ制御コマンド生成プログラム(以下、プリンタドライバ)等を記憶する。

50

【0080】

上述したCPU1、RAM2、ROM3、キーボードコントローラ(KBC)5、CRTコントローラ(CRTC)6、ディスクコントローラ(DKC)7、プリンタコントローラ(PRTC)8は、コンピュータ制御ユニット2000上に配設されている。

【0081】

このようなホストコンピュータ3000は、例えばプリンタ1500と接続可能なインターフェース21を有する汎用のパーソナルコンピュータによって実現することができる。

【0082】

次に、プリンタ1500各部の構成を詳述すると、CPU12は、システムバス15に接続された各デバイスを統括的に制御する中央処理装置であり、ROM13のプログラム用ROM13b(後述)に記憶された制御プログラム等或いは外部メモリ14に記憶された制御プログラム等に基づいて印刷部(プリンタエンジン)17に出力情報としての画像信号を出力する。また、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内の情報等をホストコンピュータ3000に通知できる構成となっている。

10

【0083】

RAM19は、CPU12の主メモリや、ワークエリア等として機能し、増設ポートに接続されるオプションRAM(図示略)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。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。ROM13は、フォント用ROM13aと、プログラム用ROM13bと、データ用ROM13cとを備えている。フォント用ROM13a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などを記憶する。プログラム用ROM13bは、CPU12の制御プログラム等を記憶する。データ用ROM13cは、プリンタ1500にハードディスク等の外部メモリ711が接続されていない場合には、ホストコンピュータ3000上で利用される情報等を記憶する。

20

【0084】

入力部18は、双方向性インターフェース21を介してプリンタ1500とホストコンピュータ3000との間におけるデータの送受を行う。印刷部インターフェース(I/F)16は、CPU12と印刷部17との間におけるデータの送受を行う。メモリコントローラ(MC)20は、外部メモリ711のアクセスを制御する。印刷部17は、CPU12の制御に基づき印刷動作を行う。操作部1501は、各種操作のためのスイッチや表示手段(例えばLED表示器)等を備えている。

30

【0085】

外部メモリ711は、ハードディスク(HD)、ICカード等から構成されており、プリンタ1500にオプションとして接続される。外部メモリ14は、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ものであり、メモリコントローラ(MC)20によりアクセスを制御される。尚、外部メモリ711は、1個に限らず、複数個備えることが可能となっている。即ち、内蔵フォントに加えてオプションカード、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。更に、NVRAM(図示略)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。

40

【0086】

上述したCPU12、RAM19、ROM13、入力部18、印刷部インターフェース(I/F)16、メモリコントローラ(MC)20は、プリンタ制御ユニット1000上に配設されている。

【0087】

(印刷制御装置の機能構成)

本発明における印刷制御装置としてのホストコンピュータ3000の機能構成は、上述した図4の構成と同一でよい。重複する説明は省略する。

なお、本実施形態において、図4におけるアプリケーション201、グラフィックエンジ

50

ン 2 0 2、ディスパッチャ 3 0 1、プリンタドライバ 2 0 3、スプーラ 3 0 2、スプール
ファイルマネージャ 3 0 4、デスプーラ 3 0 5 およびシステムスプーラ 2 0 4 は、上記図
2 の外部メモリ 1 1 に保存されたファイルとして存在し、実行される場合に OS やそのモ
ジュールを利用するモジュールによって RAM 2 にロードされ実行されるプログラムモジ
ュールである。

【 0 0 8 8 】

また、アプリケーション 2 0 1 およびプリンタドライバ 2 0 3 は、外部メモリ 1 1 の F D
や C D - R O M 或いはネットワーク（以上図示略）を経由して外部メモリ 1 1 としての H
D に追加することが可能となっている。外部メモリ 1 1 に保存されているアプリケーション
2 0 1 は RAM 2 にロードされて実行されるが、該アプリケーション 2 0 1 からプリン
タ 1 5 0 0 に対して印刷を行う際には、同様に RAM 2 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 2 0 2 を利用して出力（描画）を行う。

10

【 0 0 8 9 】

グラフィックエンジン 2 0 2 は印刷装置（プリンタ）ごとに用意されたプリンタドライバ
2 0 3 を同様に外部メモリ 1 1 から RAM 2 にロードし、アプリケーション 2 0 1 の出力
（ G D I 関数）から印刷命令（ D D I 関数）を生成し、ディスパッチャ 3 0 1 へ出力する
。

【 0 0 9 0 】

グラフィックエンジン 2 0 2 からの印刷命令はディスパッチャ 3 0 1 が受け取る。ディス
パッチャ 3 0 1 はグラフィックエンジン 2 0 2 から受け取った印刷命令が、アプリケーシ
ョン 2 0 1 からグラフィックエンジン 2 0 2 へ発行されたものである場合には、外部メモ
リ 1 1 に格納されているスプーラ 3 0 2 を RAM 2 にロードし、プリンタドライバ 2 0 3
ではなくスプーラ 3 0 2 へ印刷命令を送付する。

20

【 0 0 9 1 】

スプーラ 3 0 2 は受け取った印刷命令を中間コードに変換してスプールファイル 3 0 3 に
出力する。また、スプーラ 3 0 2 は、プリンタドライバ 2 0 3 に対して設定されている印
刷データに関する加工設定をプリンタドライバ 2 0 3 から取得してスプールファイル 3 0
3 に保存する。尚、スプールファイル 3 0 3 は外部メモリ 1 1 上にファイルとして生成す
るが、RAM 2 上に生成されても構わない。更に、スプーラ 3 0 2 は、外部メモリ 1 1 に
格納されているスプールファイルマネージャ 3 0 4 を RAM 2 にロードし、スプールファ
イルマネージャ 3 0 4 に対してスプールファイル 3 0 3 の生成状況を通知する。

30

【 0 0 9 2 】

その後、スプールファイルマネージャ 3 0 4 は、スプールファイル 3 0 3 に保存された印
刷データに関する加工設定の内容に従って印刷を行えるか判断する。スプールファイルマ
ネージャ 3 0 4 がグラフィックエンジン 2 0 2 を利用して印刷を行えると判断した際には
、外部メモリ 1 1 に格納されているデスプーラ 3 0 5 を RAM 2 にロードし、デスプーラ
3 0 5 に対して、スプールファイル 3 0 3 に記述された中間コードの印刷処理を行うよう
に指示する。

【 0 0 9 3 】

デスプーラ 3 0 5 はスプールファイル 3 0 3 に含まれる中間コードをスプールファイル 3
0 3 に含まれる加工設定の内容に従って加工し、グラフィックエンジン 2 0 2 経由へ出力
する。グラフィックエンジン 2 0 2 はデスプーラ 3 0 5 の出力（ G D I 関数）から印刷命
令（ D D I 関数）を生成し、ディスパッチャ 3 0 1 へ出力する。ディスパッチャ 3 0 1 が
グラフィックエンジン 2 0 2 から受け取った印刷命令がデスプーラ 3 0 5 からグラフィッ
クエンジン 2 0 2 へ発行されたものである場合には、ディスパッチャ 3 0 1 はスプーラ 3
0 2 ではなく、プリンタドライバ 2 0 3 に印刷命令を送る。プリンタドライバ 2 0 3 はプ
リント制御コマンドを生成し、システムスプーラ 2 0 4 経由でプリンタ 1 5 0 0 へ出力す
る。

40

【 0 0 9 4 】

（埋め込みフォントを有する印刷データの処理）

50

次に、本実施形態のホストコンピュータ3000における、埋め込みフォントを有する印刷データの処理について、図10のシーケンスチャートを参照して説明する。なお、図8と同様、本処理も1つのプロセスでは処理できない。そして、複数のジョブを制御するために、アプリケーションプロセスとデスプーラプロセス間での待ち行列処理を実現する必要がある。実際には上記2つのプロセス間の通信はスプールファイルマネージャ304を介して行われるが、説明を簡潔に行うために待ち行列処理を含めスプールファイルマネージャ304を一つの通信媒介だと捉え、図10及び以下の説明からはあえて省略した。

【0095】

ステップA1. アプリケーション201は印刷ジョブが開始することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

10

【0096】

ステップS1. スプーラ302は用紙サイズ等のジョブに関する情報を保持するジョブファイルのスプールファイル303に生成し、別プロセスであるデスプーラ305を起動する。

【0097】

ステップD1. デスプーラ305はジョブファイルを読み印刷ジョブ開始のためのGDI関数をコールする(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタ

20

ドライバ203へ引き渡す)。
ステップDRV1. プリンタドライバ203は、ジョブ開始制御のPDLコマンドを生成する。

【0098】

ステップA2. アプリケーション201はページ印刷が開始することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

ステップS2. スプーラ302はページに関する情報を保持するページファイルのスプールファイル303に生成する。

30

【0099】

ステップA3. アプリケーション201は、アプリケーションデータ内の埋め込みフォントをグラフィックエンジン202に登録する。この登録により文字描画に不可欠なフォントの情報をこの環境下でも使用可能にすることができる。

ステップG1. グラフィックエンジン202は、フォント情報を保持する。

【0100】

ステップA4. アプリケーション201がステップA3で登録したフォントを含むフォントを用いて文字描画を行う(グラフィックエンジン202は描画時にコールされるGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をスプーラ302へ引き渡す)。

40

【0101】

ステップS3. スプーラ302は、文字描画を表す情報を中間コードとしてスプールファイルのページファイルに書き込む。この中間コードにはDDI関数より取得可能な“文字コード、フォント属性、描画属性”のみ含まれており、フォントは含まれていない。なお、フォントが埋め込みフォントであった場合は、フォント置き換えによる文字描画の差異を無くすために、以下の処理を行う。

【0102】

ステップS3.3. スプーラ302は、グラフィックエンジン202に対して文字描画で使用されるフォント属性を検査し、フォント取得の許可がある場合、グラフィックエンジン202にGDI関数をコールしてフォントの取得を要求する。

50

このように、スプーラ 302 から G D I 関数をコールすることにより、D D I 関数よりは直接入手できないフォント情報を入手し、スプールすることが可能になる。

【0103】

ステップ G 4 . グラフィックエンジン 202 はフォントをスプーラ 302 に受け渡す。
ステップ S 3 . 4 . スプーラ 302 は、グラフィックエンジン 202 から取得したフォントをスプールファイル 303 に中間コード形式でスプールする。

【0104】

ステップ A 5 . アプリケーション 201 はステップ A 3 で登録したフォントをグラフィックエンジン 202 より削除する。

ステップ G 3 . グラフィックエンジン 202 で保持していた埋め込みフォントを破棄する。 10

【0105】

ステップ A 6 . アプリケーション 201 はページ印刷が終了したことを、G D I 関数を介してグラフィックエンジン 202 に通知する (グラフィックエンジン 202 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 301 へ出力し、ディスパッチャ 301 はその内容をスプーラ 302 へ引き渡す)。

ステップ S 4 . スプーラ 302 は、ページファイルをクローズし、ページ再生をデスプーラ 305 に依頼する。

【0106】

ステップ A 7 . アプリケーション 201 は印刷ジョブが終了することを、G D I 関数を介してグラフィックエンジン 202 に通知する (グラフィックエンジン 202 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 301 へ出力し、ディスパッチャ 301 はその内容をスプーラ 302 へ引き渡す)。 20

ステップ S 4 . スプーラ 302 は、ジョブファイルをクローズし、これ以上ページが増えないことをデスプーラ 305 に通知する。

【0107】

ステップ D 2 . デスプーラ 305 が、ステップ S 2 で生成されたページファイルに記述された中間コードを再生するために、ページ開始のための G D I 関数をコールする (グラフィックエンジン 202 は通知された G D I 関数を D D I 関数に変換してディスパッチャ 301 へ出力し、ディスパッチャ 301 はその内容をプリンタドライバ 203 へ引き渡す)。 30

ステップ D R V 2 . プリンタドライバ 203 は、ページ開始制御の P D L コマンドを生成する。

【0108】

ステップ D 3 . デスプーラ 305 は、スプールファイル 303 に埋め込みフォントが含まれている場合、その埋め込みフォントの識別名 (フェース名) を変化させ、グラフィックエンジン 202 に登録する。フェース名を変更して登録するのは、ステップ A 3 のアプリケーション 201 のフォントと二重登録になる場合があるためである。この登録により文字描画に不可欠なフォントの情報をこの環境下でも使用可能にすることができる。このとき、スプールファイルに格納されている中間コードのフェース名の変更されるフォント属性のフェース名も連動して変更する。 40

また、フェース名の変更が一定の規則で行われる場合は、D R V 3 . 1 のグリフの取得を行う際にフェース名を変更することによりグリフの取得を行うことができる。

【0109】

ステップ D 3 . デスプーラ 305 は、スプールファイル 303 に埋め込みフォントが含まれている場合、その埋め込みフォントの識別名 (フェース名) を変化させ、グラフィックエンジン 202 に登録する。フェース名を変更して登録するのは、ステップ A 3 のアプリケーション 201 のフォントと二重登録になる場合があるためである。この登録により文字描画に不可欠なフォントの情報をこの環境下でも使用可能にすることができる

【0110】

ステップG1. グラフィックエンジン202はフォント情報を保持する

ステップD4. デスプーラ305は、ページファイル上の文字描画を再生するために、GDI関数をコールする(グラフィックエンジン202は描画時にコールされるGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

【0111】

なお、GDI関数をコールする際、フォントの識別名であるフォント属性フェース名は、ステップD3で変更した名前を使用する。これは、例えばステップD3.で変更したフェイス名と、変更前のフェイス名とをRAM2等に一時記憶しておき、中間データに含まれるフェイス名に変更前のフェイス名が見つかった場合には変更後のフェイス名に変換することによって実現できるが、他の方法を用いてもよい。

10

【0112】

ステップDRV3. プリンタドライバ203は、スプーラファイルの文字描画を表す情報をPDLコマンドに変換する。その際に以下の処理を行う。

ステップDRV3.1. プリンタドライバ203は、グラフィックエンジン202に対して文字グリフの取得を行う。その際、ドライバ203は文字描画を表す情報に含まれるフォント属性(フェース名を含む)をグラフィックエンジン203に引き渡す。

【0113】

ステップG2. グラフィックエンジン202はフォントより文字グリフを生成し、ディスパッチャ301を介してプリンタドライバ203へ受け渡す。すなわち、グラフィックエンジン202は、文字描画を表す情報に含まれるフォント属性で指定されているフェース名のフォントを探す。ステップD3.の処理でフォントが埋め込まれているため、フォントをドライバに受け渡すことができる。プリンタドライバ203は、生成された文字グリフを取得し、取得した文字グリフ(文字のイメージデータ)を印刷するための文字描画コマンド(PDL)を作成する。

20

ステップD5. デスプーラ305はD3で登録したフォントをグラフィックエンジン202より削除する。

ステップG3. グラフィックエンジン202で保持していた埋め込みフォントを破棄する。

【0114】

30

ステップD6. デスプーラ305はページ印刷が終了したことを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

ステップDRV4. プリンタドライバ203は、ページ終了制御のPDLコマンドを生成する。

【0115】

ステップD7. デスプーラ305は印刷ジョブが終了することを、GDI関数を介してグラフィックエンジン202に通知する(グラフィックエンジン202は通知されたGDI関数をDDI関数に変換してディスパッチャ301へ出力し、ディスパッチャ301はその内容をプリンタドライバ203へ引き渡す)。

40

ステップDRV5. プリンタドライバ203は、ジョブ終了制御のPDLコマンドを生成する。

【0116】

このように、本実施形態に係る印刷制御装置においては、スプールの際にDDI関数よりは直接入手できないフォント情報を、GDI関数をコールすることによって入手しスプールし、スプールしたフォント情報はデスプール際に、フォントの識別名であるフェース名を変更して登録を行い、変更後のフェース名を用いてデスプーラからGDI関数をコールすることによって、埋め込みフォントを含む印刷データを正しく出力することができる。

【0117】

50

なお、デスクプーラが登録したフォントは著作権の問題があり使用後は必ず印刷制御装置から消去する。

【0118】

また、縦書きと横書きの情報が統合されている日本語フォントにおいて、フェイス名に縦書きを示す情報が付加されている場合がある。この場合デスクプーラが登録する際に用いる変更後のフェイス名にも、縦書きを表す情報は残しておく必要がある。変更後のフェイス名はどのように作成してもよいが、フェイス名の所定番目の文字を通常はフォント名に使用されない文字に置き換える等、簡便で、かつ変更前のフォント名もしくは印刷環境に元々存在するフォント名と重複する確率の低い方法を用いることが望ましい。

【0119】

さらに、フォントによっては、埋め込みフォントのフォント情報の取得を許可しないものも存在する。その場合は、スプール時にフォント属性を判断して、図8及び図9で説明したような、従来のフォント置き換えまたはビットマップ描画への変換を行うように構成することも可能である。

【0120】

【他の実施形態】

なお、本発明は、複数の機器(例えばホストコンピュータ、インターフェース機器、リーダー、プリンタなど)から構成されるシステムに適用しても、一つの機器からなる装置(複写機、プリンタ、ファクシミリ装置など)に適用してもよい。

【0121】

また、本発明の目的は、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コードを記憶した記憶媒体を、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、達成される。

【0122】

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。

【0123】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0124】

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なく、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動しているOS(オペレーティングシステム)など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。

【0125】

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た後、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など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。

【0126】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した(図10に示す)シーケンスチャートに対応するプログラムコードが少なくとも格納されることになる。

【0127】

【発明の効果】

以上説明したように、本発明によれば、埋め込みフォントを中間コードデータとともにスプールし、デスクプーラで登録して用いることにより、アプリケーションプログラムからの

10

20

30

40

50

印刷命令を一旦中間コードデータに変換してからプリンタへの印刷データを生成する形式の印刷制御装置において、埋め込みフォントを用いた印刷命令を正しく出力することが可能になるという効果を有する。

【図面の簡単な説明】

【図1】本発明の印刷制御装置を接続可能なプリンタの一例としてのレーザビームプリンタの構成例を示す断面図である。

【図2】本発明の一実施形態に係る印刷制御装置としてのホストコンピュータと、それに接続されたプリンタのハードウェア構成例を示すブロック図である。

【図3】従来の、アプリケーションからの印刷命令を中間コードデータへ変換せずにプリンタ制御コマンドに変換する形式の印刷制御装置の代表的な機能ブロック図である。

10

【図4】本発明の印刷制御装置に適用可能な、アプリケーションからの印刷命令を中間コードデータへ変換した後にプリンタ制御コマンドに変換する形式の印刷制御装置の代表的な機能ブロック図である。

【図5】文字描画に必要な情報と、印刷結果を説明する図である。

【図6】文字描画においてフォント置き換え処理、ビットマップ置き換えが生じた場合の印刷結果の例を示した図である。

【図7】図3に示した印刷制御装置における印刷処理のシーケンスを示す図である。

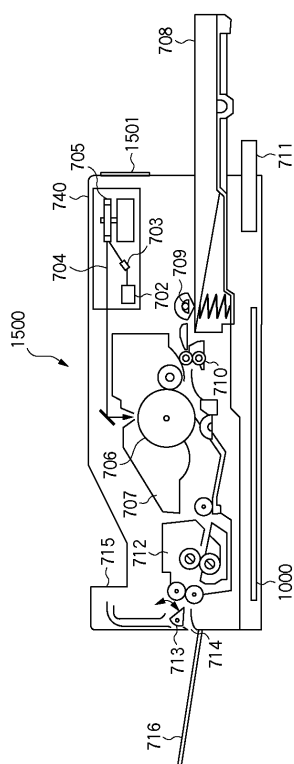
【図8】図4に示した印刷制御装置における印刷処理のうち、フォント置き換えが発生する場合のシーケンスを示す図である。

【図9】図4に示した印刷制御装置における印刷処理のうち、ビットマップ置き換えが発生する場合のシーケンスを示す図である。

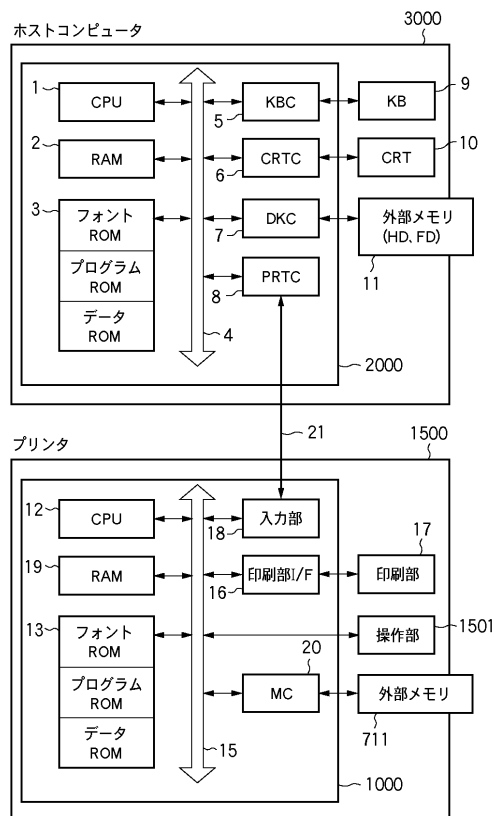
20

【図10】本発明の実施形態に係る印刷制御装置における印刷処理のうち、埋め込みフォントが含まれる印刷命令の処理のシーケンスを示す図である。

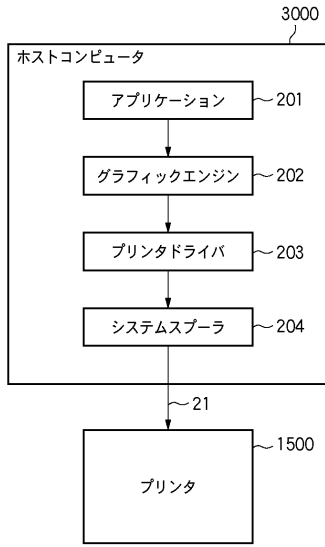
【図1】



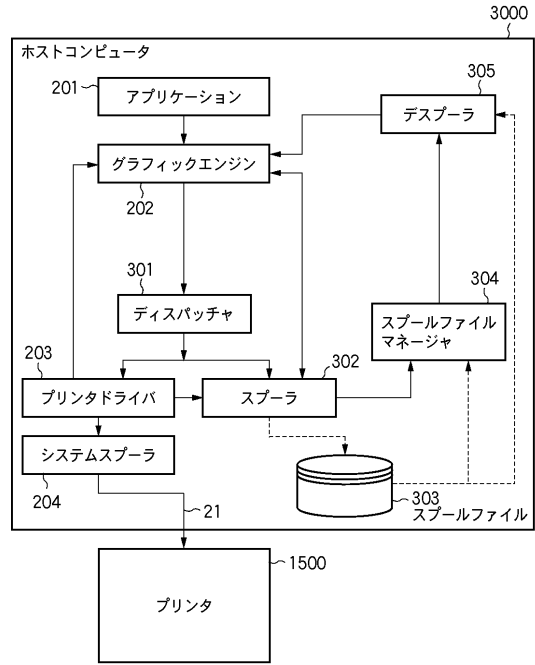
【図2】



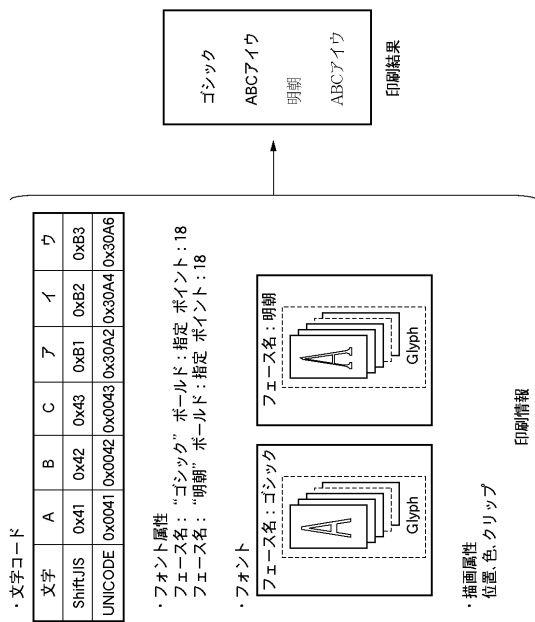
【 図 3 】



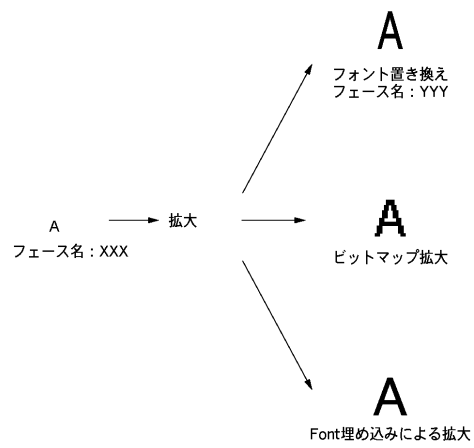
【 図 4 】



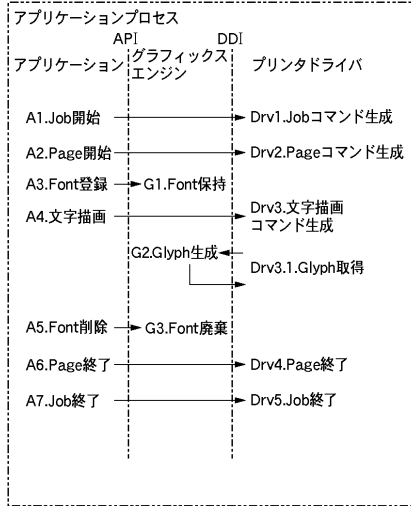
【 図 5 】



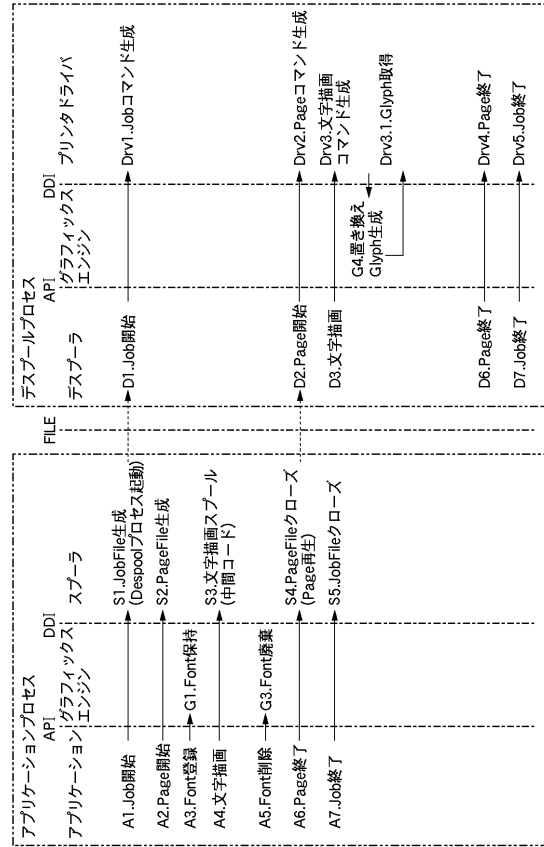
【 図 6 】



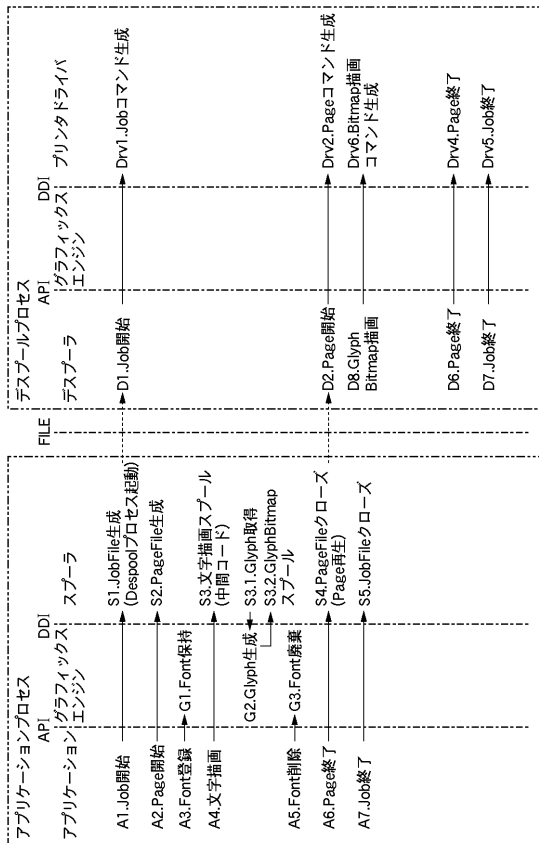
【 図 7 】



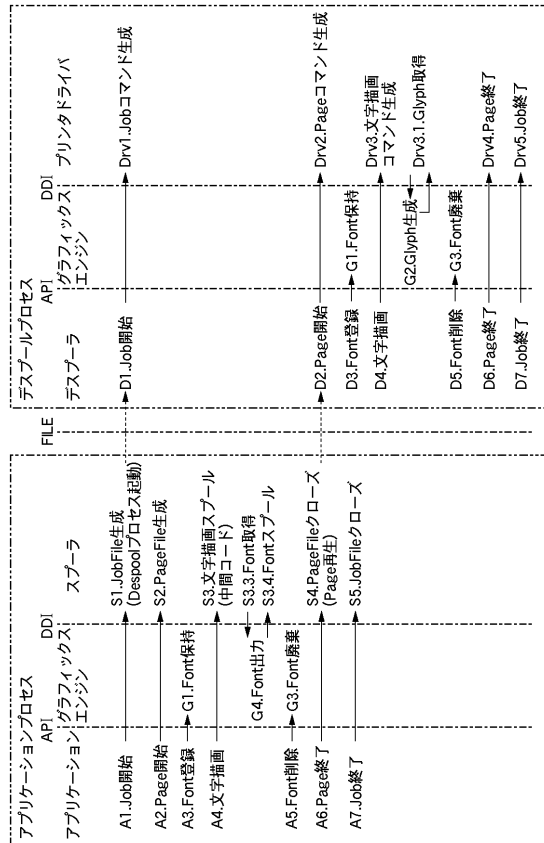
【 図 8 】



【 図 9 】



【 図 10 】



フロントページの続き

- (72)発明者 戸田 雅成
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
- (72)発明者 内田 達郎
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 三好 洋治

- (56)参考文献 特開平03-008059(JP,A)
特開2000-052610(JP,A)
特開2000-224414(JP,A)
特開2000-301803(JP,A)
特開2003-015849(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|------|------|
| G06F | 3/12 |
| B41J | 5/44 |