

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2011-34592

(P2011-34592A)

(43) 公開日 平成23年2月17日 (2011.2.17)

(51) Int.Cl.	F I	テーマコード (参考)
G 0 6 F 3/12 (2006.01)	G O 6 F 3/12 P	2 C 1 8 7
B 4 1 J 21/00 (2006.01)	B 4 1 J 21/00 Z	
	G O 6 F 3/12 C	

審査請求 有 請求項の数 12 O L (全 55 頁)

(21) 出願番号	特願2010-246751 (P2010-246751)	(71) 出願人	000001007
(22) 出願日	平成22年11月2日 (2010.11.2)		キヤノン株式会社
(62) 分割の表示	特願2001-280042 (P2001-280042)		東京都大田区下丸子3丁目30番2号
	の分割	(74) 代理人	100076428
原出願日	平成13年9月14日 (2001.9.14)		弁理士 大塚 康徳
(31) 優先権主張番号	特願2000-333337 (P2000-333337)	(74) 代理人	100112508
(32) 優先日	平成12年10月31日 (2000.10.31)		弁理士 高柳 司郎
(33) 優先権主張国	日本国 (JP)	(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

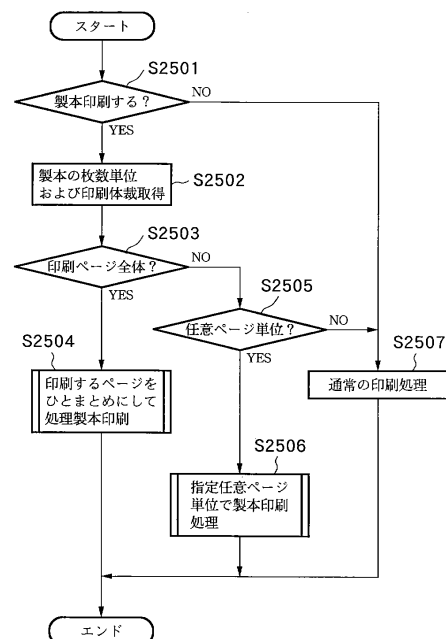
(54) 【発明の名称】 印刷制御装置、印刷制御方法およびプログラム

(57) 【要約】

【課題】 用紙を折って製本の体裁をなす印刷物を作成する場合に、各印刷領域に対して、複数の印刷ページデータを配置した印刷物を作成することが可能な印刷制御装置を提供する。

【解決手段】 1枚の用紙に印刷データの複数ページを組み合わせる印刷する際の各ページの配置を決める配置データを入力し、入力された前記配置データに従って印刷データから単一のページとして印刷すべきページデータ部分を抽出して保持し、保持されるページデータから生成される印刷データを出力する印刷制御装置は、その配置データに対して、用紙に印刷した結果を製本した場合の1ページ上にさらに複数の印刷データページを組み合わせ配置することを指定する指定ユニットと、配置データに従い、製本した場合の1ページ上に複数のページデータを配置した印刷データを生成する生成ユニットと、を備える。

【選択図】 図 2 5



【特許請求の範囲】**【請求項 1】**

第 1 のサイズのページデータと、前記第 1 のサイズの 2 倍である第 2 のサイズのページデータとが混在した文書データを入力する入力手段と、

第 2 のサイズのページデータを 2 つの前記第 1 のサイズのページデータに分割することにより前記第 1 のサイズの第 1 ページと第 2 ページとを生成する分割手段と、

用紙を 2 つに折ることで本を生成する製本印刷が指定され、かつ、前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記第 1 ページと第 2 ページとの間に隙間が生じないような配置処理を実行し、前記製本印刷が指定され、かつ、前記見開き領域に前記第 1 ページと第 2 ページとが配置されない場合、前記配置処理とは異なる配置処理を実行する配置制御手段と、

を備えることを特徴とする印刷制御装置。

【請求項 2】

前記分割手段により分割されることにより生成されたページが前記見開き領域に配置されない場合、前記配置制御手段は、レイアウト領域のサイズを縮小するための縮小処理およびレイアウト位置を変更するための移動処理のうち少なくとも 1 つが適用されたページを各領域に配置することを特徴とする請求項 1 に記載の印刷制御装置。

【請求項 3】

前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記配置制御手段は、前記第 1 ページと第 2 ページとに分割される前の前記第 2 のサイズのページを前記見開きに領域に配置することを特徴とする請求項 1 または 2 に記載の印刷制御装置。

【請求項 4】

前記見開き領域は、異なる印刷用紙にまたがる領域であることを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の印刷制御装置。

【請求項 5】

印刷制御装置の入力手段が、第 1 のサイズのページデータと、前記第 1 のサイズの 2 倍である第 2 のサイズのページデータとが混在した文書データを入力する入力工程と、

前記印刷制御装置の分割手段が、第 2 のサイズのページデータを 2 つの前記第 1 のサイズのページデータに分割することにより前記第 1 のサイズの第 1 ページと第 2 ページとを生成する分割工程と、

前記印刷制御装置の配置制御手段が、用紙を 2 つに折ることで本を生成する製本印刷が指定され、かつ、前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記第 1 ページと第 2 ページとの間に隙間が生じないような配置処理を実行し、前記製本印刷が指定され、かつ、前記見開き領域に前記第 1 ページと第 2 ページとが配置されない場合、前記配置処理とは異なる配置処理を実行する配置制御工程と、

を有することを特徴とする印刷制御方法。

【請求項 6】

前記分割工程により分割されることにより生成されたページが前記見開き領域に配置されない場合、前記配置制御工程は、レイアウト領域のサイズを縮小するための縮小処理およびレイアウト位置を変更するための移動処理のうち少なくとも 1 つが適用されたページを各領域に配置することを特徴とする請求項 5 に記載の印刷制御方法。

【請求項 7】

前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記配置制御工程は、前記第 1 ページと第 2 ページに分割される前の前記第 2 のサイズのページを前記見開きに領域に配置することを特徴とする請求項 5 または 6 に記載の印刷制御方法。

【請求項 8】

前記見開き領域は、異なる印刷用紙にまたがる領域であることを特徴とする請求項 5 乃

10

20

30

40

50

至 7 のいずれか 1 項に記載の印刷制御方法。

【請求項 9】

印刷制御装置のコンピュータを、

第 1 のサイズのページデータと、前記第 1 のサイズの 2 倍である第 2 のサイズのページデータとが混在した文書データを入力する入力手段、

第 2 のサイズのページデータを 2 つの前記第 1 のサイズのページデータに分割することにより前記第 1 のサイズの第 1 ページと第 2 ページとを生成する分割手段、

用紙を 2 つに折ることで本を生成する製本印刷が指定され、かつ、前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記第 1 ページと第 2 ページとの間に隙間が生じないように配置処理を実行し、前記製本印刷が指定され、かつ、前記見開き領域に前記第 1 ページと第 2 ページとが配置されない場合、前記配置処理とは異なる配置処理を実行する配置制御手段、

として機能させるためのプログラム。

【請求項 10】

前記分割手段により分割されることにより生成されたページが前記見開き領域に配置されない場合、前記配置制御手段は、レイアウト領域のサイズを縮小するための縮小処理およびレイアウト位置を変更するための移動処理のうち少なくとも 1 つが適用されたページを各領域に配置する請求項 9 に記載のプログラム。

【請求項 11】

前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記配置制御手段は、前記第 1 ページと第 2 ページとに分割される前の前記第 2 のサイズのページを前記見開きに領域に配置する請求項 9 または 10 に記載のプログラム。

【請求項 12】

前記見開き領域は、異なる印刷用紙にまたがる領域である請求項 9 乃至 11 のいずれか 1 項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、印刷制御装置、印刷制御方法およびプログラムに関するものである。

【背景技術】

【0002】

文字や表、画像など、データの種類が異なると、データを定義する構造やそれらデータに対する編集操作が異なるために、データの種類に応じた様々なアプリケーションプログラムが提供されている。利用者は、文字を編集するためには文字処理プログラム、表を編集するためには表計算プログラム、画像を編集するためには画像編集プログラムといった具合に、データの種別別にアプリケーションを使い分けている。

【0003】

このように、利用者はデータの種類ごとにアプリケーションプログラムを使い分けるのが一般的である。しかし、利用者が作成しようとする文書としては、例えば文字のみや、表のみ、画像のみといった 1 種類のデータのみから構成される文書よりも、文字と表、文字と画像など、複数種類のデータから構成される文書の方が一般的である。そこで複数種類のデータを含む目的文書を作成するために、利用者は各種アプリケーションが備えている印刷機能を利用してアプリケーションごとにデータを印刷させ、印刷物を所望の順序に組み合わせる必要があった。

【0004】

あるいは、いわゆるオフィススイートと呼ばれる、各種アプリケーションによってひとつの統合アプリケーションを形成するプログラムには、各アプリケーションで生成されたデータを組み合わせ、ひとつの文書を構成する機能を提供するものもある。この統合アプリケーションを使用する場合には、利用者は、各アプリケーションで作成したデータを、

10

20

30

40

50

統合アプリケーションに含まれる特定のアプリケーションによってひとつの目的文書にまとめることができる。

【0005】

ページ単位のデータをレイアウトする処理はプリンタドライバなどで行われている。例えば、1ページに対して、一つ of 原稿のページデータを出力するだけでなく、複数の原稿ページデータを印刷するNページ印刷と呼ばれる機能がある。このNページ印刷は、複数のページを用紙の1面に縮小配置するもので、両面印刷とともに用紙の節約等の目的で利用されている。

【0006】

また、両面印刷機能を有する出力装置において、出力された用紙を2つに折るとページ順序が合う様にページ順序を組かえて印刷する製本印刷と呼ばれる機能がある。1枚の用紙の場合、2つ折りによって表裏合わせて4つの印刷領域ができるので、4ページを配置して印刷する体裁となる。

【0007】

さらに複雑に、用紙を4つ折りもしくはそれ以上の折りをして、3辺を裁断することで製本の体裁をなすように現行のページを配置し印刷する方法もある。

【0008】

また、Nページレイアウトとは逆に、1ページのデータを複数の用紙に拡大連続印刷するレイアウト方法がある。これはポスター印刷や拡大連写などと呼ばれ、印刷された用紙を貼り合わせることで元の1ページを拡大した印刷結果を得ることができる。

【0009】

また、印刷情報をページ単位にレイアウトするような処理はプリンタドライバなどで行われている。代表的なものとして、「Nページ印刷」と呼ばれるレイアウト方法がある。これは、複数のページを用紙の1面に縮小配置するもので、両面印刷とともに用紙の節約等の目的で利用されている。

【0010】

このNページ印刷とは逆に、1ページ分のデータを複数の用紙に拡大連続印刷するレイアウト方法がある。これは「ポスター印刷」や「拡大連写」などと呼ばれ、印刷された用紙を貼り合わせることで元の1ページを拡大した印刷結果を得ることができる。

【先行技術文献】

【特許文献】

【0011】

【特許文献1】特開平10-093809号公報

【特許文献2】特開平09-026865号公報

【特許文献3】特開平09-261385号公報

【発明の概要】

【発明が解決しようとする課題】

【0012】

しかしながら、利用者が各種アプリケーションによる印刷物を組み合わせてひとつの目的文書を作成する場合、例えばページ番号を各ページに振るためには、いったん必要なデータをすべて印刷出力し、文書としてまとめてから付すべきページ番号を決定する必要がある。そして、各アプリケーションによって、そのアプリケーションで作成された原稿の各ページ（これを論理ページあるいは原稿ページと呼ぶ）ごとに、決定されたページ番号を書き込んでいく。アプリケーションプログラムがページ番号を振る機能を有している場合であっても、不連続部分があればその部分についてはやはり利用者がページ番号を指定しなければならない。また、目的文書のページを再配置した場合には、ページ番号もそれに合わせて振り直さなければならない。あるいは、複数の原稿ページを、印刷物としての1ページ（これを物理ページあるいは印刷ページと呼ぶ）にまとめたり、片面印刷から両面印刷に変更するなど、データの内容にかかる変更ではなく、単に書式を変更する場合であっても、アプリケーションによって編集および印刷のし直しが必要となる。

【0013】

このように、データの種類ごとにそれを管理できるアプリケーションが異なることから、アプリケーション間のインターフェースを利用者自身が人手によって提供しなければならない。このことは、利用者に多大な労力が要求されるということであり、生産性の低下をもたらす。さらに、多くの人手を介することからエラーの発生も生じやすかった。

【0014】

一方、統合アプリケーションを利用して目的文書を作成する場合、印刷出力せずに、データの状態で各種データを配置することができる。このため、印刷物を組み合わせて目的文書を作成する場合ほどには労力は必要とされることはない。しかしながら、各種データを編集・作成するためのアプリケーションは、統合アプリケーションに含まれるものに限られてしまい、利用者が望むアプリケーションが使用できるとは限らない。また、統合アプリケーションによって作成された目的文書はひとつの文書ファイルであり、ファイル単位で編集・出力等の管理が行われる。そのために、文書ファイルの一部について書式を設定しようとしてもアプリケーションの機能による制約が多く、例えば書式が変わる部分ごとに書式設定を変更しては印刷し直すといった作業が必要とされる。このため、多くの労力が必要とされて生産性が低い点においては、前述した方法と変わりはなかった。

【0015】

従来の技術では、製本の体裁にした場合の一つの印刷領域に原稿データの1ページを配置することはできるが、複数のページを印刷することはできない。用紙を折って製本する場合には、物理的な折り処理の都合上用紙サイズがある程度制限を受けるため、1つの印刷領域の大きさもそれに応じて制限を受けることになる。この限られた印刷領域を有効に利用して、情報量を多く出力したくても、1つの印刷領域に1ページしか出力できないという欠点がある。

【0016】

また、カタログなどでは、製本結果の見開きで大きな1ページを表現することがある。しかし、このようなレイアウトをするには、印刷処理の前に、見開きのそれぞれのページに配置される小さなページの原稿、いわゆる泣き別れの原稿、を用意し、それぞれを見開き位置に合うように調整する等の特別な処理が必要となるため、従来のレイアウト方式では対応できない。このような複雑で特殊なレイアウトを行うには、専門のオペレータが介在するようなシステムを用いて作成する必要があった。

【課題を解決するための手段】

【0017】

本発明は、上述の課題に鑑みてなされたものであり、その目的とするところは、用紙を折って製本の体裁をなす印刷物を作成する場合において、各印刷領域に対して、複数の印刷ページデータを配置した印刷物を作成できる印刷制御装置および方法を提供することにある。

【0018】

本発明の一つの側面にかかる印刷制御装置は、第1のサイズのページデータと、前記第1のサイズの2倍である第2のサイズのページデータとが混在した文書データを入力する入力手段と、

第2のサイズのページデータを2つの前記第1のサイズのページデータに分割することにより前記第1のサイズの第1ページと第2ページとを生成する分割手段と、

用紙を2つに折ることで本を生成する製本印刷が指定され、かつ、前記用紙を2つに折ることにより得られる見開き領域に前記第1ページと第2ページとが配置される場合、前記第1ページと第2ページとの間に隙間が生じないような配置処理を実行し、前記製本印刷が指定され、かつ、前記見開き領域に前記第1ページと第2ページとが配置されない場合、前記配置処理とは異なる配置処理を実行する配置制御手段と、

を備えることを特徴とする。

【0019】

本発明の他の側面にかかる印刷制御方法は、印刷制御装置の入力手段が、第1のサイズ

10

20

30

40

50

のページデータと、前記第 1 のサイズの 2 倍である第 2 のサイズのページデータとが混在した文書データを入力する入力工程と、

前記印刷制御装置の分割手段が、第 2 のサイズのページデータを 2 つの前記第 1 のサイズのページデータに分割することにより前記第 1 のサイズの第 1 ページと第 2 ページとを生成する分割工程と、

前記印刷制御装置の配置制御手段が、用紙を 2 つに折ることで本を生成する製本印刷が指定され、かつ、前記用紙を 2 つに折ることにより得られる見開き領域に前記第 1 ページと第 2 ページとが配置される場合、前記第 1 ページと第 2 ページとの間に隙間が生じないような配置処理を実行し、前記製本印刷が指定され、かつ、前記見開き領域に前記第 1 ページと第 2 ページとが配置されない場合、前記配置処理とは異なる配置処理を実行する配置制御工程と、を有することを特徴とする。

10

【発明の効果】

【0020】

本発明にかかる、印刷制御装置および方法によれば、用紙を折って製本の体裁をなす印刷物を作成する場合に、各印刷領域に対して、複数の印刷ページデータを配置した印刷物を作成することが可能になる。

【図面の簡単な説明】

【0021】

【図 1】本発明の一実施形態を示す印刷制御装置の構成を説明するブロック図である。

【図 2】プリンタが接続されたホストコンピュータの典型的なプリントシステムの構成を示すブロック図である。

20

【図 3】アプリケーションからの印刷命令をプリンタ制御コマンドに変換する前に、一旦中間コードスプールするプリントシステムの構成を示すブロック図である。

【図 4】本発明におけるプリンタについて説明した図である。

【図 5】スプーラ 302 における処理を示したフローチャートである。

【図 6】スプールファイルマネージャ 304 における印刷制御について示したフローチャートである。

【図 7】デスプーラ 305 における処理を示したフローチャートである。

【図 8】印刷設定画面の一例である。

【図 9】印刷スプール設定画面の一例である。

30

【図 10】ステップ S608 において、スプールファイルマネージャ 304 が生成する印刷可能となった物理ページを構成する情報を保存しているジョブ出力用設定ファイルの例を示す図である。

【図 11】図 10 のフィールド 1002 に図示されたジョブ設定情報の一例を示す図である。

【図 12】図 10 のフィールド 1004 に図示された物理ページ情報の一例を示す図である。

【図 13】図 12 の 1202 の物理ページ設定情報の例を示す図である。

【図 14】スプールファイルマネージャ 304 からデスプーラ 305 に対して物理ページの印刷要求を行う際に渡すデータ形式の一例を示した図である。

40

【図 15】設定変更エディタ 307 における設定変更処理について示したフローチャートである。

【図 16】スプールファイルマネージャ 304 でスプールされている印刷ジョブ一覧を表示する画面の一例である。

【図 17】プレビューア 306 の画面の一例である。

【図 18】設定変更エディタ 307 の画面の一例である。

【図 19】製本印刷の製本印刷処理の概要を示す図である。

【図 20】製本印刷の製本印刷処理の概要を示す図である。

【図 21】第 1 実施形態で説明するページ配置を求める方法を示す図である。

【図 22】第 2 実施形態で説明するページ配置を求める方法を示す図である。

50

【図 2 3】全ページを一まとめにして製本印刷を行う方法を説明するフローチャートである。

【図 2 4】ページの配置を定める方法を説明するフローチャートである。

【図 2 5】複数の製本印刷の選択について示した図である。

【図 2 6】用紙節約の方法について説明したフローチャートである。

【図 2 7】任意の用紙枚数を 1 ブロックとして製本印刷を行う方法を説明したフローチャートである。

【図 2 8】第 1 実施形態の配置方式を設定するための G U I の例である。

【図 2 9】第 3 実施形態の配置方式を設定するための G U I の例である。

【図 3 0】第 3 実施形態で説明するページ配置を求める方式の図である。

10

【図 3 1】アプリケーションからの印刷命令をプリンタ制御コマンドに変換する前に、一旦中間コードスプールするプリントシステムの構成を示すブロック図である。

【図 3 2】中間コードデータファイルのデータフォーマットの概略を説明する図である。

【図 3 3】ページ分割の概念を示す図で、(a) は原稿ページを示し、(b) は、原稿ページを見開きに配置するために、原稿ページを左右に 2 等分した状態を示す図である。

【図 3 4】分割データを表現するページ情報の例を示す図である。

【図 3 5】ページを分割する属性がつけられたページデータのレイアウトの概念を示す図である。

【図 3 6】ページ分割後のデータをレイアウトする際に生じるデータの欠けを説明する図であり、(a) は A3、Landscape の原稿ページを示し、(b) は、その原稿ページを 2 分割したデータを、A4、Portrait 用紙のプリンタの描画領域を考慮に入れてレイアウトする場合の例を示す図である。

20

【図 3 7】本発明のデータの欠けを防止するためのレイアウトの一例を説明する図である。

【図 3 8】ページ分割したデータを、従来のレイアウト方法で配置した例と、本発明のレイアウトを適用した一例を示す図である。

【図 3 9】用紙を折って本を作成する製本印刷のレイアウトをする図であり、(a) は製本した背表紙側から見た状態を示す図であり、(b) は製本した内側から見た状態を示す図である。

【図 4 0】ページ分割後のデータ構造の一例を示す図である。

30

【図 4 1】ページ分割を行う場合の、処理手順の説明するフローチャートである。

【図 4 2】ページレイアウトを適用する場合の、処理手順を説明するフローチャートである。

【図 4 3】分割ページのレイアウト制御を説明する図である。

【図 4 4】スタンドアロン型の文書処理システムのブロック図である。

【図 4 5】文書処理システムを実現するコンピュータのブロック図である。

【図 4 6】ブックファイルの構造の一例を示す図である。

【図 4 7】ブック属性の一覧を示す図である。

【図 4 8】章属性の一覧を示す図である。

【図 4 9】ページ属性の一覧を示す図である。

40

【図 5 0】ブックファイルを開く手順のフローチャートである。

【図 5 1】電子原稿ファイルをブックファイルにインポートする手順のフローチャートである。

【図 5 2】アプリケーションデータを電子原稿ファイルに変換する手順のフローチャートである。

【図 5 3】既存のブックファイルを開いた際のユーザインターフェース画面の一例を示す図である。

【図 5 4】新規のブックファイルを開いた際のユーザインターフェース画面の一例を示す図である。

【図 5 5】クライアントサーバ型の文書処理システムのブロック図である。

50

【発明を実施するための形態】

【0022】

＜システム概要＞

本発明の実施形態である文書処理システムの概要を、図44～図55を参照して説明する。この文書処理システムでは、一般アプリケーションにより作成されたデータファイルが、電子原稿ライターによって電子原稿ファイルに変換される。製本アプリケーションはその電子原稿ファイルを編集する機能を提供している。以下、その詳細は説明する。

【0023】

＜システム構成及び動作＞

図44は、本実施形態の文書処理システムのソフトウェア構成を示す図である。文書処理システムはデジタルコンピュータ1010によって実現されている。一般アプリケーション1010は、ワードプロセッシングやスプレッドシート、フォトレタッチ、ドローあるいはペイント、プレゼンテーション、テキスト編集などの機能を提供するアプリケーションプログラムであり、印刷機能を有している。これらアプリケーションは、作成された文書データや画像データなどのアプリケーションデータを印刷するにあたって、オペレーティングシステム（OS）により提供される所定のインタフェースを利用する。すなわち、アプリケーション1010は、作成したデータを印刷するために、前記インタフェースを提供するOSの出力モジュールに対して、あらかじめ定められる、OSに依存する形式の出力コマンドを送信する。出力コマンドを受けた出力モジュールは、プリンタ等の出力デバイスが処理可能な形式にそのコマンドを変換し、それを出力する。出力デバイスが処理可能な形式はデバイスの種類やメーカ、機種などによって異なるために、デバイスごとにデバイスドライバが提供されており、OSではそのデバイスドライバを利用してコマンドの変換を行う。OSとしてマイクロソフト社のウインドウズを利用する場合には、前述した出力モジュールとしてはGDIと呼ばれるモジュールが相当する。

【0024】

電子原稿ライター1020は、前述のデバイスドライバのひとつとして、本文書処理システム実現のために提供されるソフトウェアモジュールである。ただし、電子原稿ライター1020は特定の出力デバイスを目的としておらず、後述の製本アプリケーション1040やプリンタドライバ1060により処理可能な形式に出力コマンドを変換する。この電子原稿ライター1020による変換後の形式（以後電子原稿形式と呼ぶ）は、ページ単位の原稿を詳細な書式をもって表現可能であれば特に問わない。実質的な標準形式のうちでは、例えばアドビシステムズによるPDF形式や、SGML形式などが電子原稿形式として採用できる。アプリケーション1010により電子原稿ライター1020を利用させる場合には、出力に使用するデバイスドライバとして電子原稿ライター1020を指定してから印刷を実行させる。ただし、電子原稿ライター1020によって作成されたままの電子原稿ファイルは、電子原稿ファイルとして完全な形式を備えていない。そのため、デバイスドライバとして電子原稿ライター1020を指定するのは製本アプリケーション1040であり、その管理下でアプリケーションデータの電子原稿ファイルへの変換が実行される。製本アプリケーション1040は、電子原稿ライター1-2が生成した新規の不完全な電子原稿ファイルを後述する形式を備えた電子原稿ファイルとして完成させる。以下では、この点を明瞭に識別する必要がある際には、電子原稿ライター1020によって作成されたファイルを電子原稿ファイルと呼び、製本アプリケーションによって構造を与えられた電子原稿ファイルをブックファイルと呼ぶ。

【0025】

このようにデバイスドライバとして電子原稿ライター1020を指定し、一般アプリケーション1010によりそのデータを印刷させることで、アプリケーションデータはアプリケーション1010によって定義されたページ（以後論理ページあるいは原稿ページと呼ぶ）を単位とする電子原稿形式に変換され、電子原稿ファイル1030としてハードディスクなどの記憶媒体に格納される。なお、ハードディスクは、本実施形態の文書処理システムを実現するコンピュータが備えているローカルドライブであってもよいし、ネットワ

ークに接続されている場合にはネットワーク上に提供されるドライブであっても良い。

【0026】

製本アプリケーション1040は電子原稿ファイルあるいはブックファイル1030を読み込み、それを編集するための機能を利用者に提供する。ただし製本アプリケーション1040は、各ページの内容を編集する機能は提供しておらず、ページを最小単位として構成される、後述する章やブックの構造を編集するための機能を提供している。

【0027】

製本アプリケーション1040によって編集されたブックファイル1030を印刷する際には、製本アプリケーション1040によって電子原稿デスプーラ1050が起動される。電子原稿デスプーラ1050は、指定されたブックファイルをハードディスクから読み出し、ブックファイルに記述された形式で各ページを印刷するために、前述したOSの出力モジュールに適合する出力コマンドを生成し、不図示の出力モジュールに出力する。その際に、出力デバイスとして使用されるプリンタ1070のドライバ1060がデバイスドライバとして指定される。出力モジュールは、指定されたプリンタ1070のデバイスドライバ1060を用いて受信した出力コマンドを、プリンタ1070で解釈実行可能なデバイスコマンドに変換する。そしてデバイスコマンドはプリンタ1070に送信され、プリンタ1070によってコマンドに応じた画像が印刷される。

【0028】

図45は、コンピュータ1000のハードウェアブロック図である。図45において、CPU2010は、ROM2030のプログラム用ROMに記憶された、あるいはハードディスク2110からRAM2020にロードされたOSや一般アプリケーション、製本アプリケーションなどのプログラムを実行し、図44のソフトウェア構成や、後述するフローチャートの手順を実現する。RAM2020は、CPU2010の主メモリ、ワークエリア等として機能する。キーボードコントローラ(KBC)2050は、キーボード2090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。CRTコントローラ(CRTC)2060は、CRTディスプレイ2100の表示を制御する。ディスクコントローラ(DKC)2070は、ブートプログラム、種々のアプリケーション、フォントデータ、ユーザファイル、後述する編集ファイル等を記憶するハードディスク(HD)2110やフロッピー(登録商標)ディスク(FD)等とのアクセスを制御する。PRTC2080は、接続されたプリンタ1070との間の信号の交換を制御する。NC2120はネットワークに接続されて、ネットワークに接続された他の機器との通信制御処理を実行する。

【0029】

<電子原稿データの形式>

編集アプリケーション1040の詳細に言及する前に、ブックファイルのデータ形式を説明する。ブックファイルは紙媒体の書物を模した3層の層構造を有する。上位層は「ブック」と呼ばれ、1冊の本を模しており、その本全般に係る属性が定義されている。その下の中間層は、本でいう章に相当し、やはり「章」と呼ばれる。各章についても、章ごとの属性が定義できる。下位層は「ページ」であり、アプリケーションプログラムで定義された各ページに相当する。各ページについてもページごとの属性が定義できる。ひとつのブックは複数の章を含んでいてよく、また、ひとつの章は複数のページを含むことができる。

【0030】

図46(A)は、ブックファイルの形式の一例を模式的に示す図である。この例のブックファイルは、ブック、章、ページは、それぞれに相当するノードにより示されている。ひとつのブックファイルはひとつのブックを含む。ブック、章は、ブックとしての構造を定義するための概念であるから、定義された属性値と下位層へのリンクとをその実体として含む。ページは、アプリケーションプログラムによって出力されたページごとのデータを実体として有する。そのため、ページは、その属性値のほか、原稿ページの実体(原稿ページデータ)と各原稿ページデータへのリンクを含む。なお、紙媒体等に出力する際の

印刷ページは複数の原稿ページを含む場合がある。この構造に関してはリンクによって表示されず、ブック、章、ページ各階層における属性として表示される。

【0031】

図46において、ブック3010には、ブック属性が定義されているとともに、2つの章3020A、3020Bがリンクされている。このリンクにより、章3020A、3020Bがブック3010に包含されていることが表示される。章3020Aには、ページ3030A、3030Bがリンクされ、これらページが含まれることが示されている。各ページ3030A、3030Bにはそれぞれ属性値が定義され、その実体である原稿ページデータ(1)、(2)へのリンクが含まれる。これらリンクは、図46(B)に示す原稿ページデータ3040のデータ(1)、(2)を指し示し、ページ3030A、3030Bの実体が、原稿ページデータ(1)、(2)であることを表示する。

10

【0032】

図47は、ブック属性のリストである。下位層と重複して定義可能な項目に関しては、下位層の属性値が優先採用される。そのため、ブック属性にのみ含まれる項目に関しては、ブック属性に定義された値はブック全体を通して有効な値となる。しかし、下位層と重複する項目については、下位層において定義されていない場合における既定値としての意味を有する。なお、図示された各項目は具体的に1項目に対応するのではなく、関連する複数の項目を含むものもある。

【0033】

図48は章属性の、図49はページ属性のリストである。章属性とページ属性との関係もブック属性と下位層の属性との関係と同様である。

20

【0034】

ブック属性に固有の項目は、印刷方法、製本詳細、表紙／裏表紙、インデックス紙、合紙、章区切りの6項目である。これらは、ブックを通して定義される項目である。印刷方法属性としては、片面印刷、両面印刷、製本印刷の3つの値を指定できる。製本印刷とは、別途指定する枚数の用紙を束にして2つ折りにし、その束をつづり合わせることで製本が可能となる形式で印刷する方法である。製本詳細属性としては、製本印刷が指定されている場合に、見開き方向や、束になる枚数等が指定できる。

【0035】

表紙／裏表紙属性は、ブックとしてまとめられる電子原稿ファイルを印刷する際に、表紙および裏表紙となる用紙を付加することの指定、及び付加した用紙への印刷内容の指定を含む。インデックス紙属性は、章の区切りとして、印刷装置に別途用意される耳付きのインデックス紙の挿入の指定およびインデックス(耳)部分への印刷内容の指定を含む。この属性は、印刷用紙とは別に用意された用紙を所望の位置に挿入するインサート機能を持ったインサータが使用する印刷装置に備えられている場合か、あるいは、複数の給紙カセットを使用可能である場合に有効となる。これは合紙属性についても同様である。

30

【0036】

合紙属性は、章の区切りとして、インサータからあるいは給紙カセットから供給される用紙の挿入の指定、および、合紙を挿入する場合には、給紙元の指定などを含む。

【0037】

章区切り属性は、章の区切り目において、新たな用紙を使用するか、新たな印刷ページを使用するか、特に何もしないか等の指定を含む。片面印刷時には新たな用紙の使用と新たな印刷ページの使用とは同じ意味を持つ。両面印刷時には、「新たな用紙の使用」を指定すれば連続する章が1枚の用紙に印刷されることは無いが、「新たな印刷ページの使用」を指定すれば、連続する章が1枚の用紙の表裏に印刷されることがあり得る。

40

【0038】

章属性に関しては、章に固有の項目はなく、すべてブック属性と重複する。したがって、章属性における定義とブック属性における定義とが異なれば、章属性で定義された値が優先する。ブック属性と章属性とにのみ共通する項目は、用紙サイズ、用紙方向、N-up印刷指定、拡大縮小、排紙方法の5項目である。このうち、N-up印刷指定属性は、

50

1 印刷ページに含まれる原稿ページ数を指定するための項目である。指定可能な配置としては、 1×1 や 1×2 、 2×2 、 3×3 、 4×4 などがある。排紙方法属性は、排出した用紙にステイプル処理を施すか否かを指定するための項目であり、この属性の有効性は使用する印刷装置がステイプル機能を有するか否かに依存する。

【0039】

ページ属性に固有の項目には、ページ回転属性、ズーム、配置指定、アノテーション、ページ分割などがある。ページ回転属性は、原稿ページを印刷ページに配置する際の回転角度を指定するための項目である。ズーム属性は、原稿ページの変倍率を指定するための項目である。変倍率は、仮想論理ページ領域のサイズを100%として指定される。仮想論理ページ領域とは、原稿ページを、Nup等の指定に応じて配置した場合に、1原稿ページが占める領域である。例えば 1×1 であれば、仮想論理ページ領域は1印刷ページに相当する領域となり、 1×2 であれば、1印刷ページの各辺を約70パーセントに縮小した領域となる。

【0040】

ブック、章、ページについて共通な属性として、ウオーターマーク属性およびヘッダ・フッタ属性がある。ウオーターマークとは、アプリケーションで作成されたデータに重ねて印刷される、別途指定される画像や文字列などである。ヘッダ・フッタは、それぞれ各ページの上余白および下余白に印刷されるウオーターマークである。ただし、ヘッダ・フッタには、ページ番号や日時など、変数により指定可能な項目が用意されている。なお、ウオーターマーク属性およびヘッダ・フッタ属性において指定可能な内容は、章とページとは共通であるが、ブックはそれらと異なっている。ブックにおいてはウオーターマークやヘッダ・フッタの内容を設定できるし、また、ブック全体を通してどのようにウオーターマークやヘッダ・フッタを印刷するかを指定することができる。一方、章やページでは、その章やページにおいて、ブックで設定されたウオーターマークやヘッダ・フッタを印刷するか否かを指定できる。

【0041】

＜ブックファイルの生成手順＞

ブックファイルは上述したような構造および内容を有している。次に、製本アプリケーション1040および電子原稿ライター1020によってブックファイルを作成する手順を説明する。ブックファイルの作成は、製本アプリケーション1040によるブックファイルの編集操作の一環として実現される。図50は、製本アプリケーション1040によりブックファイルを開く際の手順である。

【0042】

まず、開こうとするブックファイルが、新規作成すべきものであるか、それとも既存のものであるか判定する（ステップS7010）。新規作成の場合には、章を含まないブックファイルを新規に作成する（ステップS7020）。新規に作成されるブックファイルは、図46の例で示せば、ブックノード3010のみ有し、章のノードに対するリンクが存在しないブックのノードとなる。ブック属性は、新規作成用としてあらかじめ用意された属性のセットが適用される。そして、新規ブックファイルを編集するためのユーザーインターフェース（UI）画面を表示する（ステップS7040）。図54は、新規にブックファイルが作成された際のUI画面の一例である。この場合には、ブックファイルは実質的な内容を持たないために、UI画面11000には何も表示されない。

【0043】

一方、既存のブックファイルがあれば、指定されたブックファイルを開き（ステップS7030）、そのブックファイルの構造、属性、内容に従ってユーザーインターフェース（UI）画面を表示する。図53は、このUI画面の一例である。UI画面11000は、ブックの構造を示すツリー部11010と、印刷された状態を表示するプレビュー部11020とを含む。ツリー部11010には、ブックに含まれる章、各章に含まれるページが、図46（A）のような木構造で表示される。ツリー部11010に表示されるページは原稿ページである。プレビュー部11020には、印刷ページの内容が縮小されて表示

される。その表示順序は、ブックの構造を反映したものとなっている。

【0044】

さて、開かれたブックファイルには、電子原稿ライターによって電子原稿ファイルに変換されたアプリケーションデータを、新たな章として追加することができる。この機能を電子原稿インポート機能と呼ぶ。図50の手順によって新規に作成されたブックファイルに電子原稿インポートすることで、そのブックファイルには実体を与えられる。この機能は、図53の画面にアプリケーションデータをドラッグアンドドロップ操作することで起動される。図51に電子原稿インポートの手順を示す。

【0045】

まず、指定されたアプリケーションデータを生成したアプリケーションプログラムを起動し、デバイスドライバとして電子原稿ライター1020を指定してアプリケーションデータを印刷出力させることで、電子原稿データに変換する（ステップS8010）。変換を終えたなら、変換されたデータが画像データであるか否かを判定する（ステップS8020）。この判定は、ウィンドウズOSの下であれば、アプリケーションデータのファイル拡張子に基づいて行える。例えば、拡張子が「bmp」であればウィンドウズビットマップデータであり、「jpg」であればjpeg圧縮された画像データ、「tiff」であればtiff形式の画像データであると判定できる。

【0046】

画像データでなかった場合には、ステップS8010で生成された電子原稿ファイルを、現在開かれているブックファイルのブックに、新たな章として追加する（ステップS8030）。章属性としては、ブック属性と共通するものについてはブック属性の値がコピーされ、そうでないものについては、あらかじめ用意された規定値に設定される。

【0047】

画像データである場合には、原則として新たな章は追加されず、指定されている章に、ステップS8010で生成された電子原稿ファイルに含まれる各原稿ページが追加される（ステップS8040）。ただし、ブックファイルが新規作成されたファイルであれば、新たな章が作成されて、その章に属するページとして電子原稿ファイルの各ページが追加される。ページ属性は、上位層の属性と共通のものについてはその属性値を与えられ、アプリケーションデータにおいて定義された属性を電子原稿ファイルに引き継いでいるものについてはその値を与えられる。例えば、Nup指定などがアプリケーションデータにおいてされていた場合には、その属性値が引き継がれる。このようにして、新規なブックファイルが作成され、あるいは、新規な章が追加される。

【0048】

図52は、図51のステップS8010において、電子原稿ライター1020により電子原稿ファイルを生成させる手順のフローチャートである。まず、新たな電子原稿ファイルを作成してそれを開く（ステップS9010）。指定したアプリケーションデータに対応するアプリケーションを起動し、電子原稿ライターをデバイスドライバとして、OSの出力モジュールに対して出力コマンドを送信させる。出力モジュールは、受信した出力コマンドを電子原稿ライターによって電子原稿形式のデータに変換し、出力する（ステップS9020）。出力先はステップS9010で開いた電子原稿ファイルである。指定されたデータすべてについて変換が終了したか判定し（ステップS9030）、終了していれば電子原稿ファイルを閉じる（ステップS9040）。電子原稿ライター1020によって生成される電子原稿ファイルは、図46（B）に示した、原稿ページデータの実体を含むファイルである。

【0049】

<ブックファイルの編集>

以上のようにして、アプリケーションデータからブックファイルを作成することができる。生成されたブックファイルについては、章及びページに対して次のような編集操作が可能である。

（1）新規追加

10

20

30

40

50

- (2) 削除
- (3) コピー
- (4) 切り取り
- (5) 貼り付け
- (6) 移動
- (7) 章名称変更
- (8) ページ番号名称振り直し
- (9) 表紙挿入
- (10) 合紙挿入
- (11) インデックス紙挿入
- (12) 各原稿ページに対するページレイアウト。

10

【0050】

このほか、いったん行った編集操作を取り消す操作や、さらに取り消した操作をやり直す操作が可能である。これら編集機能により、例えば複数のブックファイルの統合、ブックファイル内で章やページの再配置、ブックファイル内で章やページの削除、原稿ページのレイアウト変更、合紙やインデックス紙の挿入などといった編集操作が可能となる。これらの操作を行うと、図47乃至5に示す属性に捜査結果が反映されたり、あるいはブックファイルの構造に反映される。たとえば、ブランクページの新規追加操作を行えば、指定された箇所にブランクページが挿入される。このブランクページは原稿ページとして扱われる。また、原稿ページに対するレイアウトを変更すれば、その変更内容は、印刷方法やNup印刷、表紙／裏表紙、インデックス紙、合紙、章区切りといった属性に反映される。

20

【0051】**<ブックファイルの出力>**

以上のように作成・編集されるブックファイルは印刷出力を最終目的としている。利用者が図53に示す製本アプリケーションのUI画面11000からファイルメニューを選択し、そこから印刷を選択すると、指定した出力デバイスにより印刷出力される。この際、まず製本アプリケーション1040は、現在開かれているブックファイルからジョブチケットを作成して電子原稿デスプーラ1050に渡す。電子原稿デスプーラ1050は、ジョブチケットをOSの出力コマンド、例えばウィンドウズのGDIコマンドに変換し、それを出力モジュール、例えばGDIに送信する。出力モジュールは、指定されたプリンタドライバ1060によってデバイスに適したコマンドを生成し、そのデバイスに送信する。

30

【0052】

ジョブチケットは原稿ページを最小単位とする構造を有するデータである。ジョブチケットにおける構造は、用紙上における原稿ページのレイアウトを定義している。ジョブチケットは1ジョブにつき1つ発行される。そのため、まず最上位にドキュメントというノードがあり、文書全体の属性、例えば両面印刷／片面印刷などが定義されている。その下には、用紙ノードが属し、用いるべき用紙の識別子や、プリンタにおける給紙口の指定などの属性が含まれる。各用紙ノードには、その用紙で印刷されるシートのノードが属する。1シートは1枚の用紙に相当する。各シートには、印刷ページ（物理ページ）が属する。片面印刷ならば1シートには1物理ページが属し、両面印刷ならば1シートに2物理ページが属する。各物理ページには、その上に配置される原稿ページが属する。また物理ページの属性として、原稿ページのレイアウトが含まれる。

40

【0053】

電子原稿デスプーラ1050は、上述のジョブチケットを、出力モジュールへの出力コマンドに変換する。

【0054】**<そのほかのシステム構成>**

本実施形態の文書処理システムの概要は以上のようなものである。これはスタンドアロ

50

ン型のシステムであるが、これを拡張したサーバクライアントシステムでもほぼ同様の構成・手順でブックファイルが作成・編集される。ただし、ブックファイルや印刷処理はサーバによって管理される。

【0055】

図55はサーバクライアント型文書処理システムの構成を示すブロック図である。クライアント文書処理システムは、スタンドアロン型システムに、クライアントモジュールであるDOMS（文書出力管理サービス）ドライバ1090およびDOMSプリントサービスモジュール1100、DS（文書サービス）クライアントモジュール1080を加えた構成を有する。このクライアント文書処理システム12000に、文書管理サーバ12010および印刷集中管理サーバ12020およびプリントサーバ12030が接続されている。これらサーバは、通常ネットワークによってクライアント文書処理システムと接続されるが、サーバが同時にクライアントとしても機能する場合には、ネットワーク間の通信をシミュレートするプロセス間通信によって接続される。なお図55では文書管理サーバ12010と印刷集中管理サーバ12020の両サーバがクライアントに接続されているが、いずれか一方のみがネットワーク上に存在する場合もあり得る。接続されているサーバが文書管理サーバであれば、そのクライアントモジュールを含む文書管理サーバクライアントシステム12010SCが、印刷集中管理サーバ12020であれば、そのクライアントモジュールを含む印刷管理サーバクライアントシステム12020SCが、スタンドアロン型文書管理システムに追加される。

【0056】

文書管理サーバ12010は、製本アプリケーション1040により作成・編集されたブックファイルを格納するサーバである。文書管理サーバ12010によってブックファイルを管理する場合、ブックファイルは、クライアントPCのローカルHDに代わって、あるいはそれに加えて、文書管理サーバ12010のデータベース12110に保存される。製本アプリケーション1040と文書管理サーバ12010との間のブックファイルの保存および読み出しは、DSクライアント1080及びDSコア12120を介して行われる。

【0057】

印刷集中管理サーバ12020は、クライアント文書管理システム12000に格納された、あるいは文書管理サーバ12010に格納されたブックファイルの印刷を管理するサーバである。クライアントにおける印刷要求は、DOMSドライバ1090およびDOMSプリントサービスモジュール1100を介して印刷集中管理サーバ12020のDOMSWGサーバモジュール12210に送信される。集中印刷管理サーバ12020は、クライアントのプリンタで印刷する場合にはクライアントのDOMSプリントサービスモジュール1100を介して電子原稿デスプーラ1050に電子原稿データを渡し、プリントサーバ12030により印刷する場合には、プリントサーバ12030のDOMSプリントサービスモジュール12030に送信する。集中印刷管理サーバは、例えば保存されているブックファイルに対して印刷要求を発行した利用者の資格などについてセキュリティチェックを行ったり、印刷処理のログを保存したりする。このように、文書処理システムは、スタンドアロンとしても、クライアントサーバシステムとしても実現できる。

【0058】

<プレビュー表示の内容>

すでに説明したとおり、ブックファイルが製本アプリケーションによって開かれると、図53に示すユーザインターフェース画面11000が表示される。ツリー部11010には、開いているブック（以下、注目ブックと呼ぶ）の構造を示すツリーが表示される。プレビュー部には、利用者の指定に応じて、3通りの表示方法が用意されている。第1は原稿ページをそのまま表示する原稿ビューと呼ばれるモードである。原稿ビューモードでは、注目ブックに属する原稿ページの内容が縮小されて表示される。プレビュー部の表示にレイアウトは反映されない。第2は印刷ビューモードである。印刷ビューモードでは、プレビュー部11020には、原稿ページのレイアウトが反映された形で原稿ページが表

示される。第3は簡易印刷ビューモードである。簡易印刷ビューモードでは、各原稿ページの内容はプレビュー部の表示には反映されず、レイアウトのみが反映される。

【0059】

以下、製本アプリケーション1040に関連した実施形態を説明する。

【0060】

<第1実施形態>

以下、本発明を適用するのに好適な実施形態について説明を行う。図1は本発明の第1実施形態を示す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。なお、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。

10

【0061】

図1において、ホストコンピュータ3000は、ROM103のプログラム用ROMあるいは外部メモリ111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CPU101を備え、システムバス4に接続される各デバイスをCPU101が総括的に制御する。また、このROM103のプログラム用ROMあるいは外部メモリ111には、CPU101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）等を記憶し、ROM103のフォント用ROMあるいは外部メモリ111には上記文書処理の際に使用するフォントデータ等を記憶し、ROM103のデータ用ROMあるいは外部メモリ111には上記

20

【0062】

キーボードコントローラ（KBC）105は、キーボード109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。CRTコントローラ（CRTC）106は、CRTディスプレイ（CRT）110の表示を制御する。107はディスクコントローラ（DKC）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）等を記憶するハードディスク（HD）、フロッピーディスク（FD）等の外部メモリ111とのアクセスを制御する。プリンタコントローラ（PRTC）108は、双方向性インタフェース（インタフェース）121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。

30

【0063】

なお、CPU101は、例えば、RAM102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、CRT110上でのWYSIWYGを可能としている。また、CPU101は、CRT110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処理方法の設定を行える。

40

【0064】

プリンタ1500は、CPU112により制御される。プリンタCPU112は、ROM113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114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115に接続される印刷部（プリンタエンジン）117に出力情報としての画像信号を出力する。また、このROM113のプログラムROMには、CPU112の制御プログラム等を記憶する。ROM113のフォント用ROM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が記憶され、ROM113のデータ用ROMには、ハードディスク等の外部メモリ114がないプリンタの場合には、ホストコンピュータ上で利用される情報等が記憶されている。

【0065】

50

CPU 112は入力部118を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ3000に通知できる。RAM 119は、CPU 112の主メモリや、ワークエリア等として機能するRAMで、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。なお、RAM 119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114は、メモリコントローラ(MC)120によりアクセスを制御される。外部メモリ114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。また、入力部118には前述した操作パネル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。

10

【0066】

また、前述した外部メモリ114は1個に限らず、複数個備えられ、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、図示しないNVRAMを有し、操作パネル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。

【0067】

図2は、プリンタ等の印刷装置が直接接続されているか、あるいはネットワーク経由で接続されているホストコンピュータにおける典型的な印刷処理の構成図である。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111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 102にロードされ実行されるプログラムモジュールである。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111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ディスク11のHDに追加することが可能となっている。外部メモリ111に保存されているアプリケーション201はRAM 102にロードされて実行されるが、このアプリケーション201からプリンタ1500に対して印刷を行う際には、同様にRAM 102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。

20

【0068】

グラフィックエンジン202は、印刷装置ごとに用意されたプリンタドライバ203を同様に外部メモリ111からRAM 102にロードし、アプリケーション201の出力をプリンタドライバ203に設定する。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。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、OSによってRAM 102にロードされたシステムスプーラ204を経てインタフェース121経由でプリンタ1500へ印刷データとして出力される仕組みとなっている。

30

40

【0069】

本実施形態の印刷システムは、図2で示すプリンタとホストコンピュータからなる印刷システムに加えて、更に図3に示すように、アプリケーションからの印刷データを一旦中間コードデータでスプールする構成を有する。

【0070】

図3は、図2のシステムを拡張したもので、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を生成する構成をとる。図2のシステムでは、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の制御コマンドへ変換し終った時点である。これに対して、図3のシステ

50

ムでは、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 に出力した時点である。通常、後者の方が短時間で済む。

【0071】

また、図 3 で示すシステムにおいては、スプールファイル 303 の内容に対して加工することができる。これによりアプリケーションからの印刷データに対して、拡大縮小や、複数ページを 1 ページに縮小して印刷する等、アプリケーションの持たない機能を実現することができる。

【0072】

これらの目的のために、図 2 のシステムに対し、図 3 の様に中間コードデータでスプールする様、システムの拡張がなされてきている。なお、印刷データ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 102 上あるいは外部メモリ 111 上に保管する。

【0073】

以下、図 3 の詳細を説明する。図に示す通り、この拡張された処理方式では、グラフィックエンジン 202 からの印刷命令である DDI 関数をディスパッチャ 301 が受け取る。ディスパッチャ 301 がグラフィックエンジン 202 から受け取った印刷命令 (DDI 関数) が、アプリケーション 201 からグラフィックエンジン 202 へ発行された印刷命令 (GDI 関数) に基づくものである場合には、ディスパッチャ 301 は外部メモリ 111 に格納されているスプーラ 302 を RAM 102 にロードし、プリンタドライバ 203 ではなくスプーラ 302 へ印刷命令 (DDI 関数) を送付する。

【0074】

スプーラ 302 は受け取った印刷命令を解析し、ページ単位に中間コードに変換してスプールファイル 303 に出力する。このページ単位に格納されている中間コードのスプールファイルをページ描画ファイル (PDF: Page Description File) と呼ぶ。また、スプーラ 302 は、プリンタドライバ 203 に対して設定されている印刷データに関する加工設定 (Nup、両面、ステイブル、カラー/モノクロ指定等) をプリンタドライバ 203 から取得してジョブ単位のファイルとしてスプールファイル 303 に保存する。この時部単位に格納されている設定ファイルをジョブ設定ファイル (簡略して SDF: Spool Description File と呼ぶこともある) と呼ぶ。

【0075】

このジョブ設定ファイルについては後述する。なお、スプールファイル 303 は外部メモリ 111 上にファイルとして生成するが、RAM 102 上に生成しても構わない。更にスプーラ 302 は、外部メモリ 111 に格納されているスプールファイルマネージャ 304 を RAM 102 にロードし、スプールファイルマネージャ 304 に対してスプールファイル 303 の生成状況を通知する。その後、スプールファイルマネージャ 304 は、スプールファイル 303 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。

【0076】

スプールファイルマネージャ 304 がグラフィックエンジン 202 を利用して印刷を行えると判断した際には、外部メモリ 111 に格納されているデスプーラ 305 を RAM 102 にロードし、デスプーラ 305 に対して、スプールファイル 303 に記述された中間コードのページ描画ファイルの印刷処理を行うように指示する。デスプーラ 305 はスプールファイル 303 に含まれる中間コードのページ描画ファイルをスプールファイル 303 に含まれる加工設定情報を含むジョブ設定ファイルに従って加工し、GDI 関数を再生成し、もう一度グラフィックエンジン 202 経由で GDI 関数を出力する。

【0077】

ディスパッチャ 301 がグラフィックエンジン 202 から受け取った印刷命令 (DDI 関数) がデスプーラ 305 からグラフィックエンジン 202 へ発行された印刷命令 (GDI 関数) に基づいたものである場合には、ディスパッチャ 301 はスプーラ 302 ではなく、プリンタドライバ 203 に印刷命令を送る。プリンタドライバ 203 はグラフィック

10

20

30

40

50

エンジン202から取得したDDI関数に基づいてページ記述言語等からなるプリンタ制御コマンドを生成し、システムスプーラ204経由でプリンタ1500に出力する。

【0078】

更に、図3では、これまで説明した拡張システムに加えて、プレビューア306、設定変更エディタ307を配し、プレビュー、印刷設定変更、複数ジョブの結合を可能にした例を示している。印刷プレビュー、印刷設定変更、複数ジョブの結合を行うためには、まずユーザが図9に示すプリンタドライバのプロパティにおいて、「出力先の指定」を行う手段であるプルダウンメニューにおいて「ストア」を指定する必要がある。なお、プレビューだけをみたい場合は、出力先の指定として「プレビュー」を選択することによっても可能である。

10

【0079】

このようにプリンタドライバのプロパティで設定されている内容は設定ファイルとしてOSが提供する構造体（Windows（登録商標）OSでは、DEVMODEと呼ばれる）に格納される。その構造体には、例えば、スプールファイル303に含まれる加工設定中にスプールファイルマネージャ304にストアを行うかどうかの設定が含まれており、スプールファイルマネージャ304がプリンタドライバを介して加工設定を読み込み、ストア指定がなされていた場合、前述したようにスプールファイル303にページ描画ファイルとジョブ設定ファイルとが生成・格納され、図16のようにスプールファイルマネージャのウィンドウ画面がポップアップされ、スプールファイル303にスプールされたジョブがリスト表示される。図16には、4つのジョブがスプールされている例を示しており、メニューバーもしくは、そのすぐ下のメニューアイコンを押下することにより、ジョブの操作を行うことができる。

20

【0080】

メニューバーとメニューアイコンの操作の数は同じである。操作種類としては、ジョブを選択した状態で、「印刷」、中間コードのスプールファイルをそのまま残して印刷を行わせる「セーブして印刷」、印刷設定を考慮したジョブの出力プレビューを見るための「プレビュー」、中間コードのスプールファイルを削除する「削除」、中間コードのスプールファイルのコピーを生成する「複製」、複数の中間コードのスプールファイルのジョブを結合して1つのジョブにする「結合」、結合ジョブを元の複数のジョブに分割する「分割」、単体ジョブもしくは結合ジョブの印刷設定（レイアウト設定やフィニッシング設定等）を変更する「ジョブ編集」、あるジョブの印刷順序を最初にする「先頭に移動」、あるジョブの印刷順序を1つ早くする「1つ上に移動」、あるジョブの印刷順序を1つ遅くする「1つ下に移動」、あるジョブの印刷順序を最後にする「最後に移動」の以上11個の操作がある。スプールファイルマネージャのウィンドウ画面（図16）上で、ある単体ジョブもしくは結合ジョブのプレビュー指定がされた場合、外部メモリ111に格納されているプレビューア306をRAM102にロードし、プレビューア306に対して、スプールファイル303に記述された中間コードのジョブのプレビュー処理を行うように指示する。

30

【0081】

プレビューア306はスプールファイル303に含まれる中間コードのページ描画ファイル（PDF）を順次読み出し、スプールファイル303に格納されているジョブ設定ファイル（SDF）に含まれる加工設定情報の内容に従って加工し、グラフィックエンジン202に対してGDI関数を出力し、グラフィックエンジン202が自身のクライアント領域に描画データを出力することによって、画面上の出力が可能となる。

40

【0082】

グラフィックエンジン202は、指定された出力先に応じて適切なレンダリングを行うことが可能である。このことから、プレビューア306は、デスプーラ305同様に、スプールファイル303に含まれる中間コードをスプールファイル303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202を利用して出力する方法で実現可能となる。このようにプリンタドライバで設定されている加工設定をジョブ設定ファイル

50

としてスプールファイル 303 に格納し、このジョブ設定ファイルに基づいてページ描画ファイルのデータを加工して出力することにより、実際の描画データがどのように印刷されるか、更には、Nup（N ページの論理ページを 1 ページの物理ページに縮小配置して印刷する処理）指定されている場合、両面印刷されている場合、製本印刷指定されている場合、スタンプが指定されている場合、それぞれに応じて、プリンタで出力されるものに近い印刷プレビューをユーザに提供することができる。なお、従来の文書作成等のアプリケーションソフトウェアが有しているプレビュー機能は、あくまでそのアプリケーションにおけるページ設定に基づいて描画しているため、プリンタドライバでの印刷設定が反映されず、実際に印刷出力されるプレビューをユーザに認識させることはできなかった。

【0083】

上記のようにプレビュー処理を行うことにより、図 17 のようにスプールファイル 303 に含まれる印刷の加工設定の大プレビューがプレビューア 306 によって画面上に表示され、その後、ユーザの非表示指示によって、プレビューア 306 がクローズされ、制御がスプールファイルマネージャのウインドウ画面（図 16）に移行する。

【0084】

ユーザがプレビューア 306 によって表示された内容に従って、印刷を行う場合には、スプールファイルマネージャ 304 上で、「印刷」もしくは「セーブして印刷」を指示することにより印刷要求を発行する。印刷要求は前述したように、デスプーラ 305 によりジョブ設定ファイルに基づいてページ描画ファイルを加工して GDI 関数を生成し、グラフィックエンジン 202 に伝えられ、ディスパッチャ 301 経由で、プリンタドライバ 203 に印刷命令が送られ、印刷が実行される。

【0085】

次に、設定変更エディタ 307 を用いた設定変更について説明する。その実現方法としては、プレビュー同様、図 9 において「ストア」指定されたジョブに関して設定可能である。同様のフローによりスプールファイルマネージャ 304 がポップアップされ、スプールされたジョブがリスト表示される。スプールファイルマネージャのウインドウ画面（図 16）上で、「ジョブ編集」が指定され、設定変更指示がされた場合、外部メモリ 111 に格納されている設定変更エディタ 307 を RAM 102 にロードし、設定変更エディタ 307 に対して、現在またはデフォルトの加工設定の表示を行うように指示する。そして図 18 のようなジョブ設定画面が表示される。

【0086】

設定変更エディタ 307 は、「ジョブ編集」が指定されたジョブのジョブ設定ファイルをスプールファイル 303 から取得し、そのジョブ設定ファイルに指定されている設定項目に基づいて図 18 のジョブ設定画面のデフォルト値を変更する。図 18 に示す例では、「ジョブ編集」指定されたジョブのジョブ設定ファイルには、部数：1 部、印刷方法：片面、ステイプル：なし、レイアウト：1 ページ／枚等が指定されていることになる。

【0087】

この設定変更エディタ 307 でもスプールファイル 303 に含まれる中間コードのページ描画ファイルをスプールファイル 303 に格納されているジョブ設定ファイル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 202 を用いて自身のクライアント領域に出力することによって、図 18 に示す画面上の小プレビュー出力が可能となる。

【0088】

またここで、スプールファイル 303 に格納されているジョブ設定ファイルに含まれる加工設定の内容を変更、修正することが可能である。その際、プリンタドライバ 203 の設定可能な項目を設定変更エディタ 307 上のユーザインターフェースに持っていて、プリンタドライバ 203 自身のユーザインターフェースを呼び出しても構わない。図 18 に示すように、分数、印刷方法（片面、両目、製本印刷）、ステイプル（サドルフィニッシャー等）、ページレイアウト、配置順等の指定ができ、また「詳細設定」を押下することにより、プリンタドライバで指定できる項目の大半を設定しなおすことが可能となる。

10

20

30

40

50

ただし、解像度、グラフィックモード等の印刷品位に関する設定の変更は許可しないものとする。

【0089】

ここで変更された変更項目は設定変更エディタ307上の認証要求に従い、変更が認証され、制御がスプールファイルマネージャ304に移行する。変更が認証されたものは、印刷設定の変更を保存することになるが、オリジナルのジョブ設定ファイルには保存せずに、ジョブ編集等で用いられるジョブ出力用設定ファイルを新たに生成して保存することになる。ジョブ出力用設定ファイルについての詳細は、図10以降で後述する。

【0090】

ユーザがプレビューア306での確認同様、設定変更内容に従って、印刷を行う場合には、スプールファイルマネージャ304上で、印刷要求を発行する。印刷要求はグラフィックエンジン202に伝えられ、ディスパッチャ301経由で、プリンタドライバ203に印刷命令が送られ、印刷が実行される。

【0091】

また、スプールファイルマネージャのウインドウ画面（図16）では、複数の印刷ジョブを結合し、一つの印刷ジョブとして印刷するように指定することが可能である。これも、プレビュー、設定変更同様、図9のプリンタドライバのプロパティにおいて出力先を「ストア」指定されたジョブが前提となる。

【0092】

ユーザが印刷ジョブの結合を行う場合、まず、アプリケーション201からプリンタドライバ203を呼び出し、図9に示すようなユーザインターフェース上からストアを選択する。前記同様、この選択により、スプールファイル303にストアされ、図16のようにスプールファイルマネージャのウインドウ画面（図16）がポップアップされる。スプールされたジョブはスプールファイルマネージャのウインドウ上にリスト表示される。アプリケーション201から同様の操作をすることにより、スプールファイルマネージャ304上に複数ジョブのリスト表示がされることになる。

【0093】

ここで、複数ジョブを選択し、「結合」が指定された場合、外部メモリ111に格納されている設定変更エディタ307をRAM102にロードし、設定変更エディタ307に対して、リスト上の先頭ジョブまたはデフォルトの加工設定の表示を行うように指示する。そして図18のような結合設定画面が表示される。ここでは、設定変更エディタ307を結合設定画面として用いているが、別モジュールのものを用いても構わない。

【0094】

この設定変更エディタ307は、スプールファイル303に含まれる中間コードのページ描画ファイルをスプールファイル303に格納されているジョブ設定情報に含まれる加工設定の内容に従って加工し、結合ジョブとして指定されたすべてのジョブに対して、グラフィックエンジン202を用いて自身のクライアント領域に出力することによって、画面上の出力を行う。その際、図18に示すプレビュー領域に選択された全てのジョブの小プレビューが可能となる。また、結合ジョブを生成する際に、それぞれの単体ジョブのジョブ設定ファイルを拡張したジョブ出力用設定ファイルを生成する。このジョブ出力用設定ファイルは、ジョブ編集を行う際にも生成されるものであり、1つのジョブに対して1つできるものであり、結合ジョブの場合もまた1つ生成される。

【0095】

ここではそれぞれのジョブに対して、結合する前の加工設定で表示することも、結合ジョブとして統一の加工設定に変更、修正して表示することも可能である。その際、プリンタドライバ203の設定可能な項目を設定変更エディタ307上のユーザインターフェースに持っても、プリンタドライバ203自身のユーザインターフェースを呼び出しても構わない。

【0096】

ここで結合されたジョブ及び変更された変更項目は、前述したように、設定変更エディ

10

20

30

40

50

タ 3 0 7 上の認証要求に従い、変更が認証され、制御がスプールファイルマネージャ 3 0 4 に移行する。これらの操作により、先に選択された複数ジョブは、スプールファイルマネージャのウィンドウ上で一つの結合ジョブとして表示される。ユーザがプレビューア 3 0 6 での確認同様、設定変更内容に従って、印刷を行う場合には、スプールファイルマネージャ 3 0 4 上で、印刷要求を発行する。印刷要求はグラフィックエンジン 2 0 2 に伝えられ、ディスパッチャ 3 0 1 経由で、プリンタドライバ 2 0 3 に印刷命令が送られ、印刷が実行される。

【 0 0 9 7 】

図 4 は、プリンタ 1 5 0 0 の一例である両面印刷機能を有するカラーレーザプリンタの断面図である。このプリンタはホストコンピュータ 3 0 0 0 より入力した印刷データに基づいて得られる各色毎の画像データで変調されたレーザ光をポリゴンミラー 3 1 により感光ドラム 1 5 を走査して静電潜像を形成する。この静電潜像をトナー現像して可視画像を得、これを中間転写体 9 へ全色について多重転写してカラー可視画像を形成する。更に、このカラー可視画像を転写材 2 へ転写し、転写材 2 上にカラー可視画像を定着させる。以上の制御を行う画像形成部は、感光ドラム 1 5 を有するドラムユニット、接触帯電ローラ 1 7 を有する一次帯電部、クリーニング部、現像部、中間転写体 9、用紙カセット 1 や各種ローラ 3、4、5、7 を含む給紙部、転写ローラ 1 0 を含む転写部及び定着部 2 5 によって構成されている。

【 0 0 9 8 】

ドラムユニット 1 3 は、感光ドラム（感光体） 1 5 と感光ドラム 1 5 のホルダを兼ねたクリーニング機構を有するクリーナ容器 1 4 とを一体に構成したものである。このドラムユニット 1 3 はプリンタ本体に対して着脱自在に支持され、感光ドラム 1 5 の寿命に合わせて容易にユニット交換可能に構成されている。上記感光ドラム 1 5 はアルミシリンダの外周に有機光導電体層を塗布して構成し、クリーナ容器 1 4 に回転可能に支持されている。感光ドラム 1 5 は、図示しない駆動モータの駆動力が伝達されて回転するもので、駆動モータは感光ドラム 1 5 を画像形成動作に応じて反時計回り方向に回転させる。感光ドラム 1 5 の表面を選択的に露光させることにより静電潜像が形成されるように構成されている。スキャナ部 3 0 では、変調されたレーザ光を、モータ 3 1 a により画像信号の水平同期信号を同期して回転するポリゴンミラーにより反射し、レンズ 3 2、反射鏡 3 3 を介して感光ドラムを照射する。

【 0 0 9 9 】

現像部は、上記静電潜像を可視画像化するために、イエロー（Y）、マゼンダ（M）、シアン（C）の現像を行う 3 個のカラー現像器 2 0 Y、2 0 M、2 0 C と、ブラック（B）の現像を行う 1 個のブラック現像器 2 1 B とを備えた構成を有する。カラー現像器 2 0 Y、2 0 M、2 0 C 及びブラック現像器 2 1 B には、スリーブ 2 0 Y S、2 0 M S、2 0 C S 及び 2 1 B S と、これらスリーブ 2 0 Y S、2 0 M S、2 0 C S、2 1 B S それぞれの外周に圧接する塗布ブレード 2 0 Y B、2 0 M B、2 0 C B 及び 2 1 B B とがそれぞれ設けられる。また 3 個のカラー現像器 2 0 Y、2 0 M、2 0 C には塗布ローラ 2 0 Y R、2 0 M R、2 0 C R が設けられている。

【 0 1 0 0 】

また、ブラック現像器 2 1 B はプリンタ本体に対して着脱可能に取り付けられており、カラー現像器 2 0 Y、2 0 M、2 0 C は回転軸 2 2 を中心に回転する現像ロータリー 2 3 にそれぞれ着脱可能に取り付けられている。

【 0 1 0 1 】

ブラック現像器 2 1 B のスリーブ 2 1 B S は感光ドラム 1 5 に対して例えば、3 0 0 μ m 程度の微小間隔を持って配置されている。ブラック現像器 2 1 B は、器内に内蔵された送り込み部材によってトナーを搬送すると共に、時計回り方向に回転するスリーブ 2 1 B S の外周に塗布ブレード 2 1 B B によって塗布するように摩擦帯電によってトナーへ電荷を付与する。また、スリーブ 2 1 B S に現像バイアスを印加することにより、静電潜像に応じて感光ドラム 1 5 に対して現像を行って感光ドラム 1 5 にブラックトナーによる可視

10

20

30

40

50

画像を形成する。

【0102】

3個のカラー現像器20Y、20M、20Cは、画像形成に際して現像ロータリー23の回転に伴って回転し、所定のスリーブ20YS、20MS、20CSが感光ドラム15に対して300 μ m程度の微小間隔を持って対向することになる。これにより所定のカラー現像器20Y、20M、20Cが感光ドラム15に対向する現像位置に停止し、感光ドラム15に可視画像が作成される。

【0103】

カラー画像形成時には、中間転写体9の1回転毎に現像ロータリー23が回転し、イエロー現像器20Y、マゼンダ現像器20M、シアン現像器20C、次いでブラック現像器21Bの順で現像処理がなされ、中間転写体9が4回転してイエロー、マゼンダ、シアン、ブラックのそれぞれのトナーによる可視画像を順次形成し、その結果フルカラー可視画像を中間転写体9上に形成する。

【0104】

中間転写体9は、感光ドラム15に接触して感光ドラム15の回転に伴って回転するように構成されたもので、カラー画像形成時に時計回り方向に回転し、感光ドラム15から4回の可視画像の多重転写を受ける。また、中間転写体9は画像形成時に後述する転写ローラ10が接触して転写材2を挟持搬送することにより転写材2に中間転写体9上のカラー可視画像を同時に多重転写する。中間転写体の外周部には、中間転写体9の回転方向に関する位置を検知するためのTOPセンサ9a及びRSセンサ9bと、中間転写体に転写されたトナー像の濃度を検知するための濃度センサ9cが配置されている。

【0105】

転写ローラ10は、感光ドラム15に対して接離可能に支承された転写帯電器を備えたもので、金属軸を中抵抗発泡弾性体により巻回することによって構成されている。転写ローラ10は、図4に実線で示すように中間転写体9上にカラー可視画像を多重転写している間は、カラー可視画像を乱さぬように下方に離開している。そして、上記中間転写体9上に4色のカラー可視画像が形成された後は、このカラー可視画像を転写材2に転写するタイミングに合わせてカム部材（不図示）により転写ローラ10を図示点線で示す上方に位置させる。これにより転写ローラ10は転写材2を介して中間転写体9に所定の押圧力で圧接すると共に、バイアス電圧が印加され、中間転写体9上のカラー可視画像が転写材2に転写される。

【0106】

定着部25は、転写2を搬送させながら、転写されたカラー可視画像を定着させるものであり、転写材2を加熱する定着ローラ26と転写材2を定着ローラ26に圧接させるための加圧ローラ27とを備えている。定着ローラ26と加圧ローラ27とは中空状に形成され、内部にそれぞれヒータ28、29が内蔵されている。即ち、カラー可視画像を保持した転写材2は定着ローラ26と加圧ローラ27とにより搬送されると共に、熱及び圧力を加えることによりトナーが表面に定着される。可視画像定着後の転写材2は、その後排紙ローラ34、35、36によって排紙部37へ排出して画像形成動作を終了する。

【0107】

クリーニング手段は、感光ドラム15上及び中間転写体9上に残ったトナーをクリーニングするものであり、感光ドラム15上に形成されたトナーによる可視画像を中間転写体9に転写した後の廃トナーあるいは、中間転写体9上に作成された4色のカラー可視画像を転写材2に転写した後の廃トナーは、クリーナ容器14に蓄えられる。

【0108】

印刷される転写材（記録用紙）2は、給紙トレイ1から給紙ローラ3により取り出されて中間転写体9と転写ローラ10との間に挟まれるようにして搬送されてカラートナー画像が記録され、定着部25を通過してトナー像が定着される。片面印刷の場合には、案内38が上方の排紙部に記録用紙を導くように搬送経路を形成するが、両面印刷に対しては、下方の両面ユニットに導くように経路を形成する。

【0109】

両面ユニットに導かれた記録用紙は、搬送ローラ40によりトレイ1の下部（二点鎖線で示す搬送経路）に一旦送り込まれた後に逆方向に搬送され、両面トレイ39に送られる。両面トレイ39上では、用紙は給紙トレイ1に載置された状態とは表裏が逆になり、また搬送方向について前後が逆になっている。この状態で再びトナー像の転写、定着を再度行うことで、両面印刷ができる。

【0110】

図5は、スプーラ302における、スプールファイル303の生成におけるページ単位保存ステップの処理をフローチャートで示したものである。まずステップS501では、スプーラ302は、アプリケーションからグラフィックエンジン202を介して印刷要求を受け付ける。アプリケーションにおいては、図8に示すような印刷設定を入力するダイアログが表示され、このダイアログから入力された印刷設定がプリンタドライバよりスプーラ302に渡される。図8に示す設定入力ダイアログにおいては、801のような1物理ページにレイアウトする論理ページの数を決めるような設定項目等を含んでいる。

【0111】

ステップS502では、スプーラ302は、受け付けた印刷要求がジョブ開始要求か判定し、もしステップS502でジョブ開始要求であると判断した場合には、ステップS503に進み、スプーラ302は、中間データを一時的に保存するためのスプールファイル303を作成する。続いて、ステップS504では、スプーラ302は、スプールファイルマネージャ304へ印刷処理の進捗を通知し、続くステップS505でスプーラ302のページ数カウンタを1に初期化する。ここで、スプールファイルマネージャ304においては、印刷が開始されたジョブに対するジョブの情報や加工設定などをスプールファイル303より読み込み、記憶する。

【0112】

一方、ステップS502において、ジョブ開始要求ではなかったと判断した場合には、ステップS506に進む。ステップS506では、スプーラ302は、受け付けた要求がジョブ終了要求かどうかの判別を行う。ジョブ終了要求でないと判断した場合には、ステップS507に進み、改ページかどうかの判別を行う。もしもステップS507で改ページであると判断した場合には、ステップS508に進み、スプールファイルマネージャ304へ印刷処理の進捗を通知する。そしてページ数カウンタをインクリメントして、中間コードを格納しているページ描画ファイルを閉じ、次のページ描画ファイルを生成する。

【0113】

ステップS507において、受け付けた印刷要求が改ページではないと判断した場合には、ステップS509に進み、スプーラ302は、ページ描画ファイルへの中間コードの書き出しの準備を行う。次に、ステップS510では、印字要求をスプールファイル303へ格納するため、スプーラ302は、印字要求のDDI関数の中間コードへの変換処理を行う。ステップS511では、スプーラ302は、ステップS510において格納可能な形に変換された印刷要求（中間コード）をスプールファイル303のページ描画ファイルへ書き込む。その後、ステップS501に戻り、再びアプリケーションからの印刷要求を受け付ける。この一連のステップS501からステップS511までの処理を、アプリケーションよりジョブ終了要求（End Doc）を受け取るまで続ける。また、スプーラ302は、同時にプリンタドライバ203からDEVMODE構造体に格納されている加工設定等の情報を取得し、ジョブ設定ファイルとしてスプールファイル303に格納する。

【0114】

一方、ステップS506にて、アプリケーションからの印刷要求がジョブ終了であると判断した場合には、アプリケーションからの印刷要求は全て終了であるので、ステップS512に進み、スプールファイルマネージャ304へ印刷処理の進捗を通知し、処理を終える。

【0115】

図6は、スプールファイルマネージャ304における、スプールファイル303生成

10

20

30

40

50

ロセスと以降説明する印刷データ生成プロセスの間での制御の詳細をフローチャートで示したものである。図6において、ステップS601では、スプールファイルマネージャ304は、スプーラ302あるいはデスプーラ305からの印刷処理の進捗通知を受け付ける。ステップS602では、スプールファイルマネージャ304は、もし進捗通知が前述のステップS504において通知されるスプーラ302からの印刷開始通知であるかどうか判定し、もしそうであればステップS603へ進み、印刷の加工設定をスプールファイル303から読み込み、ジョブの管理を開始する。

【0116】

一方、ステップS602において、スプーラ302からの印刷開始通知でなければステップS604へ進み、スプールファイルマネージャ304は、進捗通知が前述のステップS508において通知されるスプーラ302からの1論理ページの印刷終了通知であるかどうか判定する。ここで1論理ページの印刷終了通知であればステップS605へ進み、この論理ページに対する論理ページ情報を格納する。そして、続くステップS606では、この時点でスプールが終了したn論理ページに対して、1物理ページの印刷を開始できるかを判定する。ここで、印刷可能である場合はステップS607へ進み、印刷する1物理ページに対して割り付けられる論理数から物理ページ番号を決定する。

【0117】

物理ページの計算については、例えば、加工設定が1物理ページに4論理ページを配置するような設定の場合、第1物理ページは第4論理ページがスプールされた時点で印刷可能となり、第1物理ページとなる。続いて、第2物理ページは第8論理ページがスプールされた時点で印刷可能となる。また、論理ページ数の総数が1物理ページに配置する論理ページ数の倍数でなくても、ステップS512におけるスプール終了通知によって1物理ページに配置する論理ページが決定可能である。

【0118】

ステップS608では、図10に示すような形式で、印刷可能となった物理ページを構成する論理ページ番号と、その物理ページ番号などの情報がジョブ出力用設定ファイル（物理ページ情報を含むファイル）に保存され、物理ページ情報が1物理ページ分追加されたことがデスプーラ305に通知される。その後ステップS601に戻り、次の通知を待つ。本実施形態においては、印刷データ1ページ、即ち1物理ページを構成する論理ページがスプールされた時点で印刷ジョブのスプールが全て終了していなくても印刷処理が可能である。

【0119】

一方、ステップS604において、進捗通知がスプーラ302からの1論理ページの印刷終了通知でなかった場合ステップS609へ進み、スプールファイルマネージャ304は、前述のステップS512において通知されるスプーラ302からのジョブ終了通知であるかどうかを判定する。ここで、ジョブ終了通知である場合、前述のステップS606へ進む。一方、ジョブ終了通知でない場合、ステップS610へ進み、スプールファイルマネージャ304は、受け付けた通知がデスプーラ305からの1物理ページの印刷終了通知であるかどうか判定する。ここで、1物理ページの印刷終了通知である場合はステップS612へ進み、加工設定の印刷が全て終了したかを判定する。印刷終了した場合、ステップS612へ進み、デスプーラ305に印刷終了の通知を行う。

【0120】

一方、加工設定に対する印刷がまだ終了していないと判断した場合、前述の606へ進む。本実施形態におけるデスプーラ305は印刷処理を行う単位として1物理ページ数を想定している。また、ステップS608では、1物理ページの印刷処理を行うのに必要な情報をファイルに逐次保存し、再利用可能な形式にしているが、再利用不要な場合には、共有メモリ等高速な媒体を使用し、1物理ページ単位で次々と上書きする実装にして、速度とリソースを節約するような実装形式であってもよい。また、デスプールの進捗よりもスプールの進捗の方が早い場合や全ページのスプール終了後からデスプールが開始されるような場合には、ステップS608で1物理ページ毎にページ印刷可能を通知せずに、デ

10

20

30

40

50

スプール側の進捗に応じて、複数物理ページもしくは全物理ページが印刷可能になったという通知内容にして、通知回数を節約することが可能である。

【0121】

ステップS610において、通知がデスプーラ305からの1物理ページの印刷終了通知でないと判断された場合、ステップS613へ進み、スプールファイルマネージャ304は、デスプーラ305からの印刷終了通知かどうかを判定する。通知がデスプーラ305からの印刷終了通知と判定された場合、ステップS614へ進み、スプールファイルマネージャ304は、スプールファイル303の該当するページ描画ファイルの削除を行い処理を終える。ただし、一方、デスプーラ305からの印刷終了通知でなかった場合はステップS615へ進み、その他通常処理を行い、次の通知を待つ。

10

【0122】

図7は、デスプーラ305における、印刷データの生成プロセスの詳細をフローチャートで示したものである。図7において、デスプーラ305は、スプールファイルマネージャ304からの印刷要求に応じて、スプールファイル303から必要な情報（ページ描画ファイルおよびジョブ設定ファイル）を読み出して印刷データを生成する。生成された印刷データにおけるプリンタへの転送方法については図3で説明した通りである。

【0123】

印刷データの生成では、まず、ステップS701において、前述のスプールファイルマネージャ304からの通知を入力する。続くステップS702では、デスプーラ305は、入力された通知がジョブの終了通知かどうか判定し、ジョブ終了通知であるならばステップS703へ進み、終了フラグを立て、ステップS705へ進む。一方、ステップS702においてジョブ終了通知でない場合は、ステップS704に進み、前述のステップS608における1物理ページの印刷開始要求が通知されたかどうか判定する。ステップS704において開始要求と判定されなかった場合は、ステップS710へ進み、その他エラー処理を行い、ステップS701へ戻り次の通知を待つ。

20

【0124】

一方、ステップS704において1物理ページの印刷開始要求と判定された場合は、ステップS705へ進み、デスプーラ305は、ステップS704で通知を受けた印刷処理可能な物理ページのIDを保存する。続くステップS706では、デスプーラ305は、ステップS705で保存した物理ページIDのすべてのページに関して印刷処理が済んでいるかどうか判定する。ここで全物理ページの処理が済んでいる場合は、ステップS707へ進み、前述のステップS703で終了フラグが立てられているのか判定する。終了フラグがたっている場合は、ジョブの印刷が終了したとみなし、デスプーラ305の処理終了の通知をスプールファイルマネージャ304に通知し、処理を終える。ステップS707で、終了フラグが立っていないと判定された場合は、ステップS701へ戻り次の通知を待つ。

30

【0125】

一方、ステップS706で、印刷可能な物理ページが残っていると判定された場合には、ステップS708へ進み、デスプーラ305は、保存された物理ページIDから未処理の物理ページIDを順に読み出し、読み出した物理ページIDに対応する物理ページの印刷データ生成に必要な情報を読み込み、印刷処理を行う。印刷処理はスプールファイル303に格納された印刷要求命令をデスプーラ305においてグラフィックエンジン202が認識可能な形式（GDI関数）に変換し、転送する。

40

【0126】

本実施形態のような、複数論理ページを1物理ページにレイアウトするような加工設定（以下Nページ印刷）については、このステップSで縮小配置を考慮にいれながら変換する。必要な印刷処理が終えたならば、続くステップS709において1物理ページの印刷データ生成終了の通知をスプールファイルマネージャ304に対して行う。そして再びステップS706へ戻り、ステップS705で保存しておいた印刷可能な物理ページIDすべてについて印刷処理を行うまで繰り返す。

50

【0127】

以上が、ディスパッチャ301、スプーラ302、スプールファイルマネージャ304、デスプーラ305を用いた印刷処理の流れである。上記のように処理することにより、スプーラ302が中間コードを生成してスプールファイル303に格納するタイミングでアプリケーション201が印刷処理から開放されるので、プリンタドライバ203に直接出力するよりも短時間で済む。また、スプールファイル303にプリンタドライバの印刷設定を踏まえた中間ファイル（ページ描画ファイル、ジョブ設定ファイル）として一時保存しているので、実際に印刷されるべき印刷プレビューをユーザに認識させることや、複数のアプリケーションにより生成した印刷ジョブの結合や並び替えが可能となり、印刷設定の変更を行う場合にも、再度アプリケーションを立ち上げて印刷をすることなしにユーザに行わせることを可能とする。

10

【0128】

ここで、スプーラ302を用いた印刷処理において、デスプーラ305によりグラフィックエンジン202への印刷要求時にジョブ出力用設定ファイルが生成されるが、プレビューやジョブ結合等を行う場合もジョブ出力用設定ファイルが生成される。ジョブ出力用設定ファイルは、単体ジョブの場合はジョブ設定ファイルと同等のものであり、結合ジョブの場合は複数のジョブ設定情報に基づいて生成されるものである。ここでジョブ出力用設定ファイルについて説明する。

【0129】

図10は、ステップS608において、スプールファイルマネージャ304が生成する印刷可能となった物理ページを構成する情報を保存しているジョブ出力用設定ファイルの例を示す。フィールド1001は、ジョブを識別するためのIDで、本情報を保存しているファイル名や共有メモリの名称という形で保持することも可能である。フィールド1002はジョブ設定情報である。ジョブ設定情報には、グラフィックエンジン202に対してジョブの印刷を開始するために必要な構造体、Nページ印刷の指定、ページ枠などの追加描画の指定、部数、ステイプルなどのフィニッシング指定など、1つのジョブに対して1つしか設定できない情報が含まれている。ジョブ設定情報1002には、ジョブに対する機能に応じて必要なだけ情報が保存される。フィールド1003はジョブの物理ページ数で、本フィールド以降、この数の分だけ物理ページ情報が保存されていることを示す。本実施形態では、印刷可能な物理ページ数を通知する方式であるので、このフィールドは無くても動作可能である。これ以降、フィールド1004から最後までフィールド1003の数だけ物理ページ情報が格納される。物理ページ情報については図12で説明する。

20

30

【0130】

図11は、図10のフィールド1002に図示されたジョブ設定情報の一例である。フィールド1101は全物理ページ数である。フィールド1102は、全論理ページ数である。フィールド1101および1102は、印刷データに追加して、ページ数などを付加情報として印刷する場合などに利用する。印刷が続いている際には、両フィールドは暫定的な値、もしくは、印刷が終了するまでスプールファイルマネージャ304は印刷可能な物理ページの情報の作成を延期する。フィールド1103は本印刷ジョブを何部印刷するかを指定する部数情報である。フィールド1104は、フィールド1103で複数部印刷する設定の場合、部単位で印刷するかどうかの指定である。フィールド1104はステイプル、パンチ、乙折などのフィニッシング情報で、プリンタ本体もしくは外部にフィニッシャーがある場合に指定される。フィールド1106は付加印刷情報で、ページ枠などの飾り、日付などの付加情報、ユーザ名、ページ数、ウォーターマーク印刷等、ジョブに対して付加する情報が保存される。機能が増えるに従って本ジョブ設定情報に含まれるフィールドの数も増加し、例えば、両面印刷が可能な場合は、両面印刷の指定を保存するフィールドが追加される。

40

【0131】

図12は、図10のフィールド1004に図示された物理ページ情報の一例を示す。最初のフィールド1201は物理ページ番号で、印刷順序の管理や、物理ページ番号を追加

50

印刷する際に使用される値である。フィールド 1 2 0 2 は物理ページ設定情報で、物理ページ毎にレイアウトやカラー・モノクロの指定が可能である場合、レイアウトやカラー・モノクロの設定が保存される。フィールド 1 2 0 3 は本物理ページに割り付けられる論理ページ数で、1 物理ページに 4 ページを割り付ける場合には 4 もしくは 4 ページ印刷を示す ID が保存される。フィールド 1 2 0 4 以降はフィールド 1 2 0 3 で指定された数だけ論理ページの情報が保存される。アプリケーション 2 0 1 から印刷されたページ数によっては、1 2 0 3 で指定されるページ数よりも実際のページデータ数が少なくなる場合がある。その場合には、論理ページ情報に空ページを示す特別なデータを保存して対応する。

【0 1 3 2】

図 1 3 は、1 2 0 2 の物理ページ設定情報の例である。フィールド 1 3 0 1 は物理ページ上への論理ページの配置順で、N ページ印刷で、物理ページ上に論理ページを配置する順番（左上から横へ、左上から下へ等）の指定が保存されている。システムによっては、配置順ではなく、フィールド 1 2 0 4 以降の論理ページ情報の順番をページ番号順ではなく、配置順に応じた順序で配することで 1 3 0 1 の設定を代用する場合もある。フィールド 1 3 0 2 は両面印刷の表・裏の情報で、例えば、綴じ代を表裏でそろえる際に使用される。

【0 1 3 3】

フィールド 1 3 0 3 はカラーページかモノクロページかの指定で、プリンタがモノクロモードとカラーモードを持つ場合、カラーページとモノクロページが混在する文書で、カラーページをカラーモードで、モノクロページをモノクロモードで印刷したい場合などに使用される値である。この情報を持つことにより、オートカラーモードとして、ページ単位にカラープリンタで処理を変更することが可能となる。つまり、カラーページは、中間転写体（中間転写ドラム、中間転写ベルト）もしくは転写体（転写ドラム、転写ベルト）がデバイスカラーの数分、Y M C K なら 4 回転し、モノクロページは、ブラックだけ 1 回転することにより転写制御することを可能とする。フィールド 1 3 0 4 は付加印刷情報で、物理ページに対して、ページ数や、日付などの付加情報を印刷する場合に使用される。物理ページ設定情報も、システムの機能に応じてフィールドが追加される。

【0 1 3 4】

図 1 4 は、1 2 0 4 で示された論理ページ情報の一例を示す。フィールド 1 4 0 1 は論理ページの ID で、この ID を利用して、スプールファイル 3 0 3 から論理ページに対応するページ描画ファイルの中間コードを参照する。この ID を利用して論理ページの中間コードへアクセス可能であれば良く、ファイルやメモリポインタであっても、論理ページを構成する中間コード自身が入っていてもよい。フィールド 1 4 0 2 は論理ページ番号で論理ページ番号を付加情報として印刷する場合や、論理ページ ID の補助情報に使用される。フィールド 1 4 0 3 のフォーマット情報には、論理ページ単位で指定可能である各種設定項目が保存される。例えば、ページ枠などの付加印刷情報、拡大率などの論理ページ単位に指定される各種設定の情報が保存される。また、必要であれば、論理ページ単位のカラー・モノクロ情報などの論理ページに対する属性情報を保存することも可能である。逆に、論理ページ単位で設定を切りかえることや論理ページ単位での属性情報が不要であるようなシステムでは、フィールド 1 4 0 3 は不要である。

【0 1 3 5】

ジョブ出力用設定ファイルは、上記のように構成されている。なお、ジョブ設定ファイルもほぼ同様であり、印刷体裁（片面、両面、製本印刷）、印刷レイアウト（N up、ポスター印刷）、付加情報（ウォーターマーク、日付、ユーザ名の付加）、部数、用紙サイズ情報がジョブとして有しており、物理ページ毎に、論理ページの配置順、両面印刷の表面か、裏面か、カラーモード等から構成されている。

【0 1 3 6】

更に、図 3 では、これまで説明した拡張システムに加えて、ジョブの設定変更機能を持つ設定変更エディタ 3 0 7 を配した例を示している。本実施形態ではジョブの設定内容は、単体ジョブは、ジョブ設定ファイルに、また結合ジョブは、図 1 0 に示したジョブ出力

10

20

30

40

50

用設定ファイル中に含まれており、中間コードを保存しているページ描画ファイル303とは独立しているため、ジョブ出力用設定ファイルを作り変えることでジョブの設定変更が可能である。設定変更エディタ307は単独で、あるいはスプールファイルマネージャ304と連携して、ジョブ出力用設定ファイルを作り変え、あるいは、一部を書き換えることでジョブの設定変更機能を実現している。

【0137】

図15は、設定変更エディタ307におけるジョブ設定変更処理プロセスの詳細をフローチャートで示したものである。まずステップS1501では、設定変更エディタは、ジョブ設定ファイルもしくはジョブ出力用設定ファイルを読み込む。ジョブ出力用設定ファイルはプレビューア305、デスプーラ303が読み込むものと同じファイルである。次に、ステップS1502へ進み、読み込んだ結果を、ユーザに表示する。ステップS1503で、図18に示したようなユーザインターフェース上で、ユーザとの対話を行い、前述したメニューの指定等により設定内容を変更する。このステップは、対話形式でなく、ファイルなどに書きこまれた設定変更の内容に応じて変更するバッチ形式でもよい。

【0138】

次にステップS1504へ進み、ステップS1501で設定変更エディタは、最初を読み込んだ内容と、現在指定されている設定内容に変更があったかどうかの判定を行う。設定内容に変更があった場合は、ステップS1505へ進み、新規のジョブ出力用設定ファイルを生成し、変更があったことをスプールファイルマネージャに通知して終了する。ステップS1504で、変更がないと判定された場合は、変更がなかったことをスプールファイルマネージャに通知して終了する。このように新規のジョブ出力用設定ファイルを生成するが、図18のユーザインターフェース画面において、「OK」ボタンが選択されることにより、新規のジョブ出力用設定ファイルが有効となり、古いジョブ出力用設定ファイルは削除される。また、ジョブ出力用設定ファイルからの変更ではなく、単体ジョブのジョブ設定ファイルの場合は削除せずに保存しておく。

【0139】

また、図18の画面で「初期状態に戻す」ボタンが選択された場合は、新規のジョブ出力用設定ファイルを削除し、古いジョブ出力用設定ファイルが有効となり、表示に反映させる。本実施形態では、設定変更エディタ307を別モジュールとして説明しているが、単にスプールファイルマネージャ304のユーザインターフェースの一部であってもよい。設定変更エディタ307で実際に変更内容をジョブ出力用設定ファイルに書きこまずに、設定変更の内容のみをスプールファイルマネージャ304へと通知するだけで、実際のジョブ出力用設定ファイルの変更はスプールファイルマネージャ304側で行う実装形式でもよい。

【0140】

図3では、更に、複数印刷ジョブを結合し、一つの印刷ジョブとして印刷する拡張システムが図示されているが、結合ジョブをデスプール・プレビューするための拡張について説明する。

【0141】

通常、中間形式のスプールファイル303はジョブ単位で作成される。単独ジョブの場合は、処理対象ジョブファイル中の各論理ページの間コードを順に読み出して処理を行うので、フィールド1401の論理ページIDは、各論理ページがファイルのどこに位置しているのかを示す相対あるいは絶対オフセットで実現可能である。結合ジョブの場合はフィールド1401のジョブIDから、スプールファイルと、そのジョブに属するページ情報を特定する必要がある。本実施形態では、スプールファイルを識別するIDを論理ページIDに付加することで、スプールファイルを特定する方式とする。この場合、主な変更点はフィールド1401のみで済む。スプールファイルが識別できれば、ページ部分の読み込みは単独ジョブの処理と同じロジックで処理することが可能であるからである。また、スプールファイルが各論理ページ毎に別ファイルの形で保存されている場合は、論理ページのファイル名をそのままフィールド1401の論理ページIDとする実装形もある

。

【0142】

図19は、本発明で述べる製本印刷とその実現結果について説明したものである。製本印刷とは、紙の両面に出力された印刷結果を2つ折りにした結果、ページ順序があうように印刷するものである。これには、以下のような場合がある。

【0143】

1. 印刷される全ページを製本の枚数単位とする場合
2. 任意のページを製本の枚数単位とする場合

ここでいう製本の枚数単位とは、出力された用紙を何枚単位でまとめて2つ折りにして製本印刷をすればよいかをあらわす。また、図20に示すように、製本印刷によって出力される製本の体裁には以下の場合がある。

10

【0144】

1. 左から昇順にページ配置（左開き）
2. 右から昇順にページ配置（右開き）

また、印刷されるページが横向きの場合も、ページの回転を行うだけで、ページの順序は変わらない。

【0145】

以下、製本印刷の処理について詳しく述べる。まず、図25のフローチャートについて説明する。図25において、ステップS2501において、ユーザが製本印刷の設定をしていればステップS2502へ進み、していなければ通常の印刷を行うとしてステップS2507へ進む。ステップS2502において、製本印刷の枚数単位に関する設定と印刷体裁に関する設定を取得する。印刷枚数単位の設定には以下の選択肢がある。

20

【0146】

1. 印刷するページすべてをひとまとめにして製本印刷
 2. 指定された任意ページ単位をひとまとめにして製本印刷
- また、印刷体裁の設定には以下の選択肢がある。

【0147】

1. 左開き
2. 右開き

ステップS2503において、製本印刷の枚数単位に全ページが選択されていればステップS2504へ進み、選択されていなければステップS2505へ進む。ステップS2504において、製本の枚数単位を印刷ページ全体に設定し、製本印刷を行う。この処理の詳細については後述する。

30

【0148】

ステップS2505において、製本印刷の枚数単位に特定の枚数が選択されていればステップS2506へ進み、選択されていなければステップS2507へ進む。ステップS2506において、指定された枚数単位で製本印刷を行う。この処理の詳細については後述する。ステップS2507において、通常の印刷処理を行う。

【0149】

以下に図25におけるステップS2504に関する処理を、図23を用いて詳しく説明する。その前に、処理の概要を、図21を用いて説明する。

40

【0150】

一番上には、元となる印刷データが並んでいる。これを図19に示すように二つ折りで製本する形式に配置する。折ることで、表裏合わせて4つの印刷可能な面ができるので、各面に2ページのデータを配置して印刷する場合の例を説明しているのが図21である。まず、配置用の2次元配列（ $2 \times N$ ）を用意する。 $2 \times N$ となっているのは、一つの面に2つのページデータを配置するからで、一つの面に4ページのデータを配置するならば、 $4 \times N$ の配列を用意する。配列の大きさNは、印刷データのページ数に応じて変わるので、十分大きなサイズの配列を確保するか、適宜配列のリサイズを行いながら処理を進める必要がある。

50

【0151】

印刷すべき順序に並んでいる元データを、配列用データに並べて行く。並べる様子は、一番上の元データと配列用データの間に線を引いて示している。次に、実際に印刷する際に、この配置用データから2ページのデータを取り出して、用紙上に配置、印刷する。印刷結果と配置データの関係は真中の配置データと一番下に示す印刷結果を見開いた模式図の間に矢印を引いて示している。このようにもとの印刷データと実際に用紙上に出力される位置の関係が定められる。

【0152】

この設定をするためのGUI例を図28に示す。製本印刷の詳細を設定するダイアログに、2ページ印刷などのレイアウトを設定するためのコントロールおよびそのページ構成順序を設定するためのコントロールが付加されている。

10

【0153】

次に、図23の説明をする。図23において、ステップS2301において、印刷ページ数を取得する。この印刷ページはアプリケーションからの印刷データがすべてスプール終了した時点で取得可能である。ステップS2302において、印刷体裁を取得する。ステップS2303において、製本印刷の進捗カウンタを0にセットする。このカウンタが偶数のときに印刷されるページは折り込まれたときに内側になるように印刷され、奇数のときに印刷されるページは折り込まれたときに外側になるように印刷される。

【0154】

ステップS2304において、ページ配置用のデータを作成して使用する紙の枚数を求める。この処理の詳細については後述する。ステップS2305において2つ折りにしたとき一番内側になる用紙の内側に印刷されるデータ2面分を以下の式を用いて計算する。この2ページを元に以降印刷されるページを算出する。

20

【0155】

基本配置データ1 = 用紙枚数 × 2

基本配置データ2 = 用紙枚数 × 2 + 1

ここの配置データは、図21の中央に示すページ配置を決める際に使用したデータ配列である。図21の例で言えば、基本配置データが3と求めた場合、配置用データ配列の3番目、つまり、元データの5ページと6ページが配置されるページである。

【0156】

例えば、アプリケーションから50ページの印刷データが送られてきた場合は、図21に示すように、一つの面上に2ページを印刷する指定の製本印刷の場合、製本印刷の結果使用する枚数は7枚となり、2つ折りしたときに折り込まれる用紙に印刷される基本配置データは、 $7 \times 2 = 14$ 番目と $7 \times 2 + 1 = 15$ 番目となる。

30

【0157】

ステップS2306において、製本印刷処理が終了した場合には処理を終え、終わっていない場合にはステップS2307へ進む。ステップS2307において、印刷要求を行う2ページを以下の式で計算する。

【0158】

印刷配置データ1 = 基本配置データ1 - 進捗カウンタ

印刷配置データ2 = 基本配置データ2 + 進捗カウンタ

例えば、上の50ページ製本印刷の場合は、次の出力は配置用配列の $14 - 1 = 13$ 番目と $15 + 1 = 16$ 番目となる。以降進捗カウンタが増加するたびに12番目と17番目、11番目と18番目・・・となっていく。また、始めは進捗カウンタが0なので基本配置データが最初の印刷ページとなる。

40

【0159】

ステップS2306において、計算された印刷配置データ2がステップS2304で求めた配置用データの数より大きいかどうか判定し、大きくなった場合は余白扱いとなり、ステップS2309へ進む。大きくなければステップS2310へスキップする。ステップS2309において、印刷配置データ2は余白が配置されるので、例えば、余白をあら

50

わすページ番号 0 を代入する。

【0160】

ステップ S 2 3 1 0、ステップ S 2 3 1 1 およびステップ S 2 3 1 3 において、印刷体裁（左開きか右開きか）および進捗カウンタから（折り込んだ際に内側にくるか外側にくるか）から計算された 2 ページの配置を決定する。具体的には、内側外側の判定は印刷要求回数が偶数の場合内側、奇数の場合外側となる。

【0161】

左開きで折り込み内側になるページの場合ステップ S 2 3 1 2 へ進む

左開きで折り込み外側になるページの場合ステップ S 2 3 1 4 へ進む

右開きで折り込み外側になるページの場合ステップ S 2 3 1 2 へ進む

右開きで折り込み内側になるページの場合ステップ S 2 3 1 4 へ進む

ステップ S 2 3 1 2 において、印刷配置データ 1 を左側に、印刷配置データ 2 を右側に配置する。ステップ S 2 3 1 4 において、印刷配置データ 1 を右側に、印刷配置データ 2 を左側に配置する。ステップ S 2 3 1 5 において、印刷配置データの示すもとデータのページを印刷領域上に配置する。ステップ S 2 3 1 6 において、印刷進捗カウンタを 1 増加させる。ステップ S 2 3 1 7 において、次の印刷要求が可能かどうかを判定し、可能ならばステップ S 2 3 1 8 へ進み、不可能ならば次の印刷要求が可能になるまで待つ。ステップ S 2 3 1 8 において、配置完了した印刷ページに対して紙 1 面分の印刷要求を行う。ステップ S 2 3 1 9 において、次のページの出力要求が可能かどうか判定し、可能ならばステップ S 2 3 0 6 へ進み、可能でないならば可能になるまで待つ。

【0162】

以下に図 2 3 におけるステップ S 2 3 0 4 に関する処理を、図 2 4 を用いて詳しく説明する。図 2 3 において、ステップ S 2 4 0 1 において、元となるデータを出力順に並べる。通常、アプリケーションから印刷された順に印刷するが、ページ順序を入れ替えられるようなシステムでは、この入れ替えた順序で並べる必要がある。ステップ S 2 4 0 2 において、配置データを取得する。図 2 1 の場合、1 つの配置可能領域に対して 2 ページ分のデータを配置するという指定がされているデータが取得される。

【0163】

ステップ S 2 4 0 3 において、ステップ S 2 4 0 1 で並べた元データを指し示すカウンタと、配置用データを差し示すカウンタとを先頭を指し示すように初期化する。ステップ S 2 4 0 4 において、元データを指し示すカウンタが元データ数を超えているかを判定する。元データ数を超えていれば全てのデータが配置用データに格納されたので処理を終えて戻る。そうでなければ、ステップ S 2 4 0 5 へと進む。ステップ S 2 4 0 5 において、元データ用のカウンタが指し示す先のデータを取り出す。一番最初は先頭を示しているので先頭ページが取り出される。ステップ S 2 4 0 6 において、取り出したデータを格納する先の配置用データがいっぱいかどうかを判定する。既に確保してある配置用データの配列がいっぱいに埋まっている場合は、ステップ S 2 4 0 7 へすすむ。まだ配置用データに余裕がある場合は、ステップ S 2 4 0 8 へ進む。

【0164】

ステップ S 2 4 0 7 において、配置用データを適宜サイズを増やす。ステップ S 2 4 0 6 ~ 2 4 0 7 は最初に十分大きなサイズの配置用データを確保している場合や、元データサイズが小さいことが分かっている場合には無くてもかまわない。ステップ S 2 4 0 8 において、配置用データのカウンタが指し示している領域にステップ S 2 4 0 5 で取り出した元データを格納する。ステップ S 2 4 0 9 において、元データのカウンタ、配置用データのカウンタをそれぞれインクリメントする。

【0165】

以下に図 2 5 におけるステップ S 2 5 0 6 に関する処理を図 2 7 を用いて詳しく説明する。ここでいう枚数単位とは何枚毎に折り込んで製本印刷を行うかを表している。例えば、アプリケーションからの 40 ページの印刷データを一つの印刷領域に 1 ページずつ、2 枚単位で製本印刷する場合、2 つ折りした 1 つのブロックには 8 ページ分のデータが配置

されることになる。よって、印刷データ全体を5ブロックに分けて製本印刷することになる。

【0166】

ステップS2701において、印刷体裁および枚数単位を取得する。ステップS2702において、印刷終了ページを0にセットする。ステップS2703において、製本印刷処理が終わりかどうか判定し、終了ならば処理を終え、終了でなければステップS2704へ進む。ステップS2704において、製本印刷処理可能になる印刷配置データ（製本印刷可能判定配置データ）を取得する。ステップS2704で求められる製本印刷可能判定配置データ数は、ブロック内の最大配置データ数となる。始めのブロックにおける製本印刷可能判定配置データ数は以下によって求められる。

10

【0167】

製本印刷可能判定配置データ＝枚数単位×4

第2ブロック以降については、製本印刷可能判定配置データ数は以下によって求められる。

【0168】

製本印刷可能判定ページ＝前のブロックの製本印刷可能判定ページ＋枚数単位×4

例えば、上記40ページ2枚単位の製本印刷の場合、最初のブロックにおける製本印刷可能判定ページは $2 \times 4 = 8$ ページとなり、次のブロックは $8 + 2 \times 4 = 16$ ページとなる。以降24ページ、32ページ、40ページとなっていく。

【0169】

20

ステップS2705において、スプールが終了しているページ数を取得する。ステップS2706において、製本印刷処理可能になる印刷配置データ（製本印刷可能判定配置データ）までスプール終了しているかあるいは全ページのスプールが終了しているかどうかを判定し、処理可能ならばステップS2707へ進み、可能でなければステップS2705へ戻る。

【0170】

ステップS2707において、進捗カウンタを0にセットする。ステップS2708において、2つ折りにしたとき一番内側になる用紙の内側に印刷される配置データ2つ分を以下の式を用いて計算する。この2配置データを元に以降印刷される配置データを算出する。

30

【0171】

基本配置データ1＝製本印刷可能判定配置データ－枚数単位×2＋1

基本配置データ2＝製本印刷可能判定配置データ－枚数単位×2

ステップS2709において、現在のブロックの印刷要求が終了したか判定し、終了ならばステップS2703へすすみ、終了していなければステップS2710へ進む。具体的には、進捗カウンタが枚数単位に等しくなった場合に印刷要求が終了したと判定する。ステップS2710において、印刷要求を行う2配置データを以下の式で計算する。

【0172】

印刷配置データ1＝基本配置データ1＋進捗カウンタ

印刷配置データ2＝基本配置データ2－進捗カウンタ

40

始めは進捗カウンタが0なので基本配置データが最初の印刷ページとなる。ステップS2711において、計算された印刷配置データが配置データのサイズより大きいかどうかを判定し、大きくなった場合は余白扱いとなり、ステップS2712へ進む。大きくなければステップS2713へスキップする。ステップS2712において、印刷ページは余白が配置されるので、例えば、余白をあらわすページ番号0を代入する。ステップS2713、ステップS2714およびステップS2716において、印刷体裁（左開きか右開きか）および進捗カウンタから（折り込んだ際に内側にくるか外側にくるか）から計算された2配置データの配置を決定する。具体的には、内側外側の判定は印刷要求回数が偶数の場合内側、奇数の場合外側となる。

【0173】

50

左開きで折り込み内側になるページの場合ステップ S 2 7 1 5 へ進む

左開きで折り込み外側になるページの場合ステップ S 2 7 1 7 へ進む

右開きで折り込み外側になるページの場合ステップ S 2 7 1 5 へ進む

右開きで折り込み内側になるページの場合ステップ S 2 7 1 7 へ進む

ステップ S 2 7 1 5 において、印刷配置データ 1 を左側に、印刷配置データ 2 を右側に配置する。ステップ S 2 7 1 7 において、印刷配置データ 1 を右側に、印刷配置データ 2 を左側に配置する。ステップ S 2 7 1 8 において、二つ折りのどちら側か決まった方の印刷可能領域に対して、印刷配置データ中の元データを配置する。ステップ S 2 7 1 9 において、印刷進捗カウンタを 1 増加させる。ステップ S 2 7 2 0 において、次の印刷要求が可能かどうかを判定し、可能ならばステップ S 2 7 2 1 へ進む、不可能なら次の印刷要求が可能になるまで待つ。ステップ S 2 7 2 1 において、配置完了した印刷ページに対して紙 1 面分の印刷要求を行う。この印刷要求の処理は、具体的には、デスプーラ 3 0 5 が、配置完了したページデータをスプールファイル 3 0 3 から読み出し、G D I 関数である描画データを再生成してグラフィックエンジン 2 0 2 に対して出力する処理に相当する。

10

【0174】

このように、製本の体裁時の 1 ページに相当する一つの印刷領域に対して、指定される任意の論理ページを配置した印刷データを生成することが可能となり、結果として、印刷装置で製本印刷される際に 1 ページに複数のページがレイアウトされる印刷物を得ることが可能となる。

20

【0175】

<第2実施形態>

全ページがスプール終了してから任意枚数単位の製本印刷処理を行う場合において、その最終ブロックに印刷されるページ数が少ない場合、最終ブロックに限り指定枚数より少ない枚数単位で製本印刷を行うことによって、使用する用紙を少なくすることが可能である。例えば、残り配置データ数が 3 個に対して枚数単位が 2 の場合、配置可能領域 5 個分が余白になってしまうが、これを 1 枚単位に変更した場合、余白は 1 配置領域分ですみ、用紙の枚数が 1 枚少なくてすむ。

【0176】

以降図 2 6 によって最終ブロックの枚数調整処理について詳しく説明する。これは、印刷ページに対して必要以上に大きい枚数単位が指定された場合にも適用される。例えば、アプリケーションからの 6 0 ページの印刷データを一つの配置可能領域（印刷可能領域）に 2 ページ配置し、用紙 1 0 枚単位での分冊製本印刷が指定された場合、2 ページ配置するため配置データは 3 0 個となり用紙枚に対して 4 つの配置データが配置されるため、3 0 を 4 つに分けることにより、実際に必要な用紙枚数は 8 枚であることがわかる。よって本第 2 実施形態では、枚数単位は 1 0 枚から 8 枚へ調整されることにより用紙枚数の節約を行う。なお、この処理は分冊製本印刷や複数ページ配置に限らず、まとめ製本印刷や論理ページを印刷可能領域全体に製本印刷する場合も同じ出力結果になる。

30

【0177】

以下の図 2 6 の処理は、第 2 7 図におけるステップ S 2 7 0 6 とステップ S 2 7 0 7 の間において処理することによって目的が達成される。図 2 6 において、ステップ S 2 6 0 1 において、全ページのスプールが終了したかどうか判定し、終了していればステップ S 2 6 0 2 に進み、進んでいなければ処理を抜ける。ステップ S 2 6 0 2 において、残り配置データ数を以下の式で計算する。

40

【0178】

残り配置データ数 = 配置データ数 - 現在の製本印刷可能判定配置データ数

ステップ S 2 6 0 3 において、現在設定されている枚数単位におけるブロックに収まる配置可能データ数と、残り配置データ数を比較して、残り配置データ数が少なければステップ S 2 6 0 4 へ進む、そうでなければ調整処理は必要ないとして処理を抜ける。ステップ S 2 6 0 4 において、残りページがおさまるもっとも小さい製本の枚数単位を以下の式によって求める。

50

【0179】

新しい製本枚数単位＝残り配置データ数÷4（小数点以下切り上げ）

よって、スプールファイルマネージャ304は、適する枚数を製本印刷の枚数として設定することになり、ユーザがプリンタドライバのUIを用いて最初に指定した枚数より少ない用紙枚数に配置された印刷データが生成されることになる。

【0180】

よって、本第2実施形態では、使用する用紙を少なくすることが可能になる。

【0181】

<第3実施形態>

配置方法を指定する際に、全てを統一した形式ではなく、配置可能な領域（1枚の用紙を二つ折りすると4つの領域となる）毎に異なる形式として指定できるようにしてもよい。この場合の配置決定は、図23のステップS2304の処理を変更することで実現可能である。例えば、先頭ページを表紙として別に扱い、1領域に1ページで出力し、その他のページは各2ページずつを一つの配置可能な領域に配置するようなことも可能である。先頭ページを別扱いして、常に1つの領域に1ページを配置するような設定をすることが可能であるようなGUIの例を図29に示す。

【0182】

図29のGUI例は、図28のGUIにさらに、カバーページを別扱いするかどうかを選択かのようなチェックボックスコントロールが追加されている。このチェックボックスにチェックがなされている場合には、先頭ページを別として扱い、通常の1領域に1ページの印刷を行い、その他の部分に関しては、一つの領域に複数のページを配置するという設定をすることが可能である。図22はこの場合の、配置データ作成の様子を示した図である。まず、元データを出力順に並べる。配置用データは、基本的に各2ページであるので2×Nの配列として確保する。そして、配置用データの先頭の第1領域に対して、元データの先頭ページの1ページを配置する。次の部分は飛ばして、配置用データの2つめに移り、配置用データの第1領域に元データの2ページ目、配置用データ2番目の第2領域に元データの3ページ目、配置用データの次に移り3番目の第1領域に元データの4ページ目という用に処理を続ける。このようにして作成した配置用データと、実際の出力の関係は、図22の中央の配置用データと一番下の出力模式図の間に矢印で示している。

【0183】

本第3実施形態では、配置方法を指定する際に、全てを統一した形式ではなく、配置可能な領域毎に異なる形式を指定することができる。

【0184】

<第4実施形態>

一つの印刷領域に対して複数のページを配置するのとは逆に、見開きを構成するような複数の印刷領域をあわせて大きく一つの印刷領域として使用する配置方法を取ることも可能である。この場合のページ配置を定める方法を図30に示す。図30において、元のデータの第2ページ目を配置データの2番目と3番目に入れることで、見開きを大きく同じページのデータを出力することを指定する。この配置データを使用して印刷した場合の様子が一番下の図に示してある。元データと配置データの関連は実線で、配置データと出力結果の関連の様子は矢印でそれぞれ示してある。

【0185】

第4実施形態によれば、一つの印刷領域に対して複数のページを配置するのとは逆に、見開きを構成するような複数の印刷領域をあわせて大きく一つの印刷領域として使用する配置方法を取ることも可能となる。

【0186】

<第5実施形態>

本実施形態においては、製本印刷の見開きページを想定し、その両ページで左右が別の媒体（紙）でも、一続きの印刷情報、イメージ情報の連続性を維持しつつ、該当ページに

所定の情報のレイアウト（貼り付け）することができるようにする、あるいは、表紙から裏表紙にかけてイメージが一続きになる製本を可能にする実施の形態を説明する。

【0187】

尚、説明において、上述の第1乃至第4実施形態の説明で使用した図面と共通する図面については、その図面を利用して説明するものとする。

【0188】

本実施形態の印刷システムは、図2で示すプリンタとホストコンピュータからなる印刷システムに加えて、更に、図31に示すように、アプリケーション201からの印刷データを一旦、中間コードデータでスプール構成を有する。

【0189】

図31は図2のシステムを拡張したもので、同図に示す通り、この拡張された処理方式では、グラフィックエンジン202からの印刷命令であるDDI関数をディスパッチャ301が受け取る。ディスパッチャ301がグラフィックエンジン202から受け取った印刷命令（DDI関数）が、アプリケーション201からグラフィックエンジン202へ発行された印刷命令（GDI関数）に基づくものである場合には、ディスパッチャ301は外部メモリ111に格納されているスプーラ302をRAM102にロードし、プリンタドライバ203ではなくスプーラ302へ印刷命令（DDI関数）を送付する。

【0190】

スプーラ302は受け取った印刷命令を解析し、ページ単位に中間コードに変換してスプールファイル303に出力する。

【0191】

また、スプーラ302は、プリンタドライバ203に対して設定されている印刷データに関する加工設定（見開き製本印刷等）をプリンタドライバ203から取得してジョブ単位のファイルとしてスプールファイル303に保存する。

【0192】

なお、スプールファイル303は外部メモリ111上にファイルとして生成するが、RAM102上に生成しても構わない。更にスプーラ302は、外部メモリ111に格納されているスプールファイルマネージャ304をRAM102にロードし、スプールファイルマネージャ304に対してスプールファイル303の生成状況を通知する。その後、スプールファイルマネージャ304は、スプールファイル303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。

【0193】

スプールファイルマネージャ304がグラフィックエンジン202を利用して印刷を行えると判断した際には、外部メモリ111に格納されているデスプーラ305をRAM102にロードし、デスプーラ305に対して、スプールファイル303に記述された中間コードのページ描画ファイルの印刷処理を行うように指示する。デスプーラ305はスプールファイル303に含まれる中間コードのページ描画ファイルをスプールファイル303に含まれる加工設定情報を含むジョブ設定ファイルに従って加工し、GDI関数を再生成し、もう一度グラフィックエンジン202経由でGDI関数を出力する。

【0194】

ディスパッチャ301がグラフィックエンジン202から受け取った印刷命令（DDI関数）がデスプーラ305からグラフィックエンジン202へ発行された印刷命令（GDI関数）に基づいたものである場合には、ディスパッチャ301はスプーラ302ではなく、プリンタドライバ203に印刷命令を送る。プリンタドライバ203はグラフィックエンジン202から取得したDDI関数に基づいてページ記述言語等からなるプリンタ制御コマンドを生成し、システムスプーラ204経由でプリンタ1500に出力する。

【0195】

ここで適用するシステムは、アプリケーション201で中間ファイルを読み込むことを前提としているので、ディスパッチャ301を設定せず直接スプーラ302へデータを送り、中間ファイルを作成する簡略化されたシステムでも良い。このシステムのほうが、処

10

20

30

40

50

理が少なくなるので、リソースの消費が少ないなどの利点も多い。どちらの形式を利用するかは、システムの目的に応じて構成を変更することが可能である。

【0196】

図32は、中間コードデータファイル（中間ファイル）のデータフォーマットの概略を説明する図である。レイアウト処理がしやすいように、ページごとに大きくグループ化されている。

【0197】

中間ファイルのヘッダ情報として、ページ数、ユーザ情報などの文書全体で必要となるような情報が保持される。その後にページ情報が続き、ページ情報として、ページIDと、ファイルアクセス方法やファイルフォーマットの実装の自由度を高めるために、実際のデータとの関連付けをするリンク情報を保持する。ページ情報は、この内容に限るものではなく、その他製本印刷において必要な情報をページ情報として保持することができる。

【0198】

各ページ情報が個別に格納された後に、実際のページデータ（1～N）が格納される。このページデータには、ページ毎のデータが、例えば、元の一つ一つの図形や文字列のデータ、が保持される。

【0199】

また、中間ファイルの生成は、種々のアプリケーションから生成された印刷情報に限るものではなく、スキャナなどの外部機器から取り込まれた画像ファイルを利用することも可能である。この場合、取り込まれた画像データを中間ファイル形式に合うようにヘッダ情報を付加して直接取り込むことも可能であり、また、画像データを印刷し、スプーラ302経由で中間ファイル形式に変換してから読み込むことも可能である。

【0200】

次に、印刷情報のレイアウトにおいて、見開きで1つのまとまりとなる原稿を見開きページに分割するページ分割の概念を図33を用いて説明する。ここでは、カタログのように見開きの2面を利用して1つの大きなページを表示する場合を想定している。図33（a）の「LR」と書かれたページが原稿ページである。これを見開きに配置するために、原稿ページを左右に2等分しているのが図33（b）である。例えば、原稿ページがA3、Landscapeのサイズである場合、分割後の各ページのサイズはA4、Portraitの原稿が2枚（2ページ分）となる。

【0201】

原稿ページを分割して処理するために、分割データを表現するページ情報（ページ分割の属性がつけられたデータ）の例を図34に示す。元となるページを特定するためのページIDは共通となる。ページ分割として2分割を想定しているため、分割した右側か左側かを示すデータがページ情報に追加されている。実際のページデータは共通であるので、リンク情報として同一のページデータを指し示すことでデータの重複を防ぐことが可能である。

【0202】

中間ファイルとして格納されているデータのうち、ページIDの共通性、分割情報の有無によって、そのデータが通常のデータであるか、分割されたデータであるかを識別することが可能である。

【0203】

このページ分割は、印刷形式を製本印刷として指定された場合に実行されるものであり、中間ファイルデータには分割に関する分割情報（右側のデータか左側にデータか）と、実データの指定、共通データの指示等が、分割単位のページで管理される。

【0204】

ホストコンピュータ3000は、その印刷データの処理において、データが分割データであると識別すると、実データへのリンク情報に基づき実データを特定して、所定の形態によりレイアウト処理を実行する。

【0205】

10

20

30

40

50

ページが分割されたデータを、所定の記憶媒体にレイアウト（貼り付け）するための位置座標の対応づけを図35を用いて説明する。図35は、ページを分割する属性がつけられたページデータのレイアウトの概念を示す図である。この図では、A3、Landscapeの原稿ページをA4、Portraitの用紙にレイアウトする方法を示している。原稿ページの左半分を表現する場合、原稿ページの座標と用紙座標の対応づけは以下のようになる。

【0206】

原稿ページの左上を用紙の左上に、同じく原稿ページの左下を用紙の左下に、そして、原稿ページの上辺の中央を用紙の右上に、原稿ページの下辺の中央を用紙の右下というように、座標の関係付けを行ない配置する。

【0207】

原稿ページの右半分を示す場合も同様に、原稿ページの座標と用紙座標の対応は、原稿ページの上辺中央と用紙左上、原稿ページ下辺中央と用紙左下、原稿ページの右上と用紙の右上、そして原稿ページ右下と用紙の右下となる。

【0208】

実際には、原稿の縦横比と用紙の縦横比が異なるため、微調整をする場合がある。現在流通しているプリンタのほとんどは、用紙上に描画できない領域があるが、これを考慮に入れるとさらに処理は複雑になる。

【0209】

図36はA3、Landscapeの原稿ページ（図36（a））を2分割した左側を、A4、Portrait用紙のプリンタの描画領域を考慮に入れてレイアウトする場合の例を示す図である。図36（b）で、外側実線が用紙サイズの外形を示し、内側の実線がその描画可能な領域である。原稿ページ、図36（a）の中央部に印刷されている図形である「○（丸印）」が、図36（b）の描画可能な領域の関係で、丸印の右側の欠けが多くなっていることがわかる。このように、分割の切り取り線（図36（a）中央の一点鎖線）付近に描かれている図形は描画領域の関係で、単純にページを分割して、そのデータを貼り付けただけでは、原稿データで描画されているデータを完全に分割した領域に描画することができなくなり、データの欠落が生じてしまう。本実施形態では、この欠けを防ぐために、以下の調整を行ってレイアウト制御を行なう。

【0210】

図37は、レイアウト制御のためにページを分割する際、用紙サイズと描画領域との関係で、描画することができない領域近くに配置されるデータの欠落を防止するための3種類の調整を示している。

【0211】

調整（1）は描画位置を平行移動して調整する方式である。「平行移動」とは、製本印刷において、綴じ代部分を確保するために、ページデータの配置位置を綴じ代等の印刷形式の影響を受けない位置にシフトしてデータの欠落を防止するレイアウトの態様である。移動の調整において、実線の矩形37aは用紙サイズを示し、内側の矩形37bは、その用紙の描画領域を示す。また、点線の矩形37cは調整後の原稿の描画位置を示している。用紙サイズと描画領域との関係で、描画することができない幅寸法37e（製本印刷した場合の綴じ代）に相当する幅を平行移動させて、分割前に原稿ページの描画範囲となっていた部分を確保するように調整するものである。単に描画データをずらしているの、データの表示位置は全体に左に寄って描画されることになる。

【0212】

調整（2）は、配置するデータを縮小により描画領域に収める方式で、描画領域に収まるように縮小することで、描画領域を示す部分と、調整後の原稿の描画位置とが重なったような表示となっている（37d）。この調整による場合、用紙の左右方向へのずれはないが、描画するデータの大きさ（寸法）が、縮小により変わってしまうことになる。

【0213】

調整（3）は平行移動（1）と縮小（2）とを組合わせたハイブリッド型による調整方式である。

10

20

30

40

50

【0214】

いずれの場合も完全に欠けを防ぐようにするか、多少欠けを許容して、原稿のイメージをできるだけ保存するか、また、どの方法を採用するかは、システムの用途・目的によって異なる。すべてを自動化せずに、どの方法を採用するかをユーザに選択させることも可能である。

【0215】

図38は、原稿ページ3820の分割（ページの泣き別れ）を示す図であり、分割されたページが連続した領域に配置される場合の配置例を示している。2UP印刷などと呼ばれる印刷では、ページの1面に複数の原稿ページを配置することが可能であるが、分割された原稿が連続した形で並べて配置された場合、従来技術では、レイアウト3820のように二つの分割ページデータはばらばらに無関係のデータとして配置される。この場合、左側を描画する際には、左半分のページサイズに相当する範囲でクリップ処理を行い、右側にかかる描画データが印刷されないようにする必要がある。

10

【0216】

本実施形態におけるレイアウト制御では、生成され、若しくは読み取りにより生成されたデータをページ単位に分割する場合、分割ページとして管理する分割情報を、中間データに付与し、この情報に基づきレイアウト制御においてデータの貼り付けの制御が行なわれる。1ページ内に複数ページの情報をレイアウトする場合、それぞれのデータが分割されたデータであるかどうかを、分割情報に基づき識別をする。分割されたデータが1ページ内にレイアウトされる場合は、この分割情報に基づきデータの関連付けが行なわれ、それぞれの実データのリンクからレイアウト3830のように表示させること、すなわち、原稿3820を復元することが可能となる（泣き別れのページの復元処理）。

20

【0217】

図39は、製本印刷を実行する場合に対して、本実施形態を適用した場合の例を示す図である。製本印刷とは、用紙の中央を折ることで、表裏4つの面を構成し、その4面にデータを印刷して本のように扱える出力物を作成する印刷形態である。一度に折りたたむ用紙の枚数を2枚とした場合、それぞれの分冊単位では4面x2枚で8ページ分の原稿を印刷することになる。

【0218】

図39は、2枚の用紙を中央で折り曲げて製本する場合の例を示す図で、(a)は製本した背表紙側から見た状態を示す図であり、(b)は製本した内側から見た状態を示す図である。1枚目の用紙3910の印刷面には、背表紙側に（8ページ目、1ページ目）の情報が印刷されることとなり、その裏側に（2ページ目、7ページ目）の情報が印刷されることとなる。同様に、2枚目の用紙3920の印刷面には、（6ページ目、3ページ目）の情報が印刷されることとなり、その裏側に（4ページ目、5ページ目）の情報が印刷されることとなる。

30

【0219】

ここで、原稿の2ページ目、3ページ目が分割されたページとすると、本発明を適用した場合、製本結果において見開きを構成し、すなわち連続した領域に配置されることを検出し、見開きで表示したときに、分割前の原稿イメージに近くなるように、配置することが可能となる。

40

【0220】

このように、本実施形態にかかるレイアウト方法は、分割ページが配置される用紙が、NUP印刷のように同一用紙の同一面上だけでなく、用紙の裏表や、異なる用紙上に配置される場合でも適用することが可能である。

【0221】

図41に、ページを分割する場合の処理の流れを示すフローチャートを示す。

【0222】

まず、ステップS4110で、分割する対象のページを特定して、ページ分割の指定を行う。この場合の「指定」とは、ユーザが特定のページを指定して分割する方法でも、自

50

動的にページの分割を設定することにしてもよい。自動的な分割とは、例えば、A4とA3の原稿が混在している場合に、A3の原稿をすべてA4に分割するような処理である。

【0223】

次に、ステップS4120処理を進め、ステップS4110でページ分割が指定された分割対象のページに対して、分割後のレイアウト制御のために、ページ情報を作成する。このページ情報の作成には、図34に示すような分割の属性を示すデータ構造を作成し、その分割にかかるページ情報を他のページ情報と識別するためのページIDを書き込む処理を行なう。ページIDを共通にするデータは、その元となる（原稿）データの共通性を示すものである。

【0224】

次にステップS4130に処理を進め、ステップS4120で作成したページ情報と、実ページデータとの間に関連付けを行なう（リンクを張る）。このとき、同一のページデータに対してリンクを張る方法ならばリソース使用量は削減できるが、ページ削除処理で一方のページのみが削除されたような場合に、実データが削除可能かどうかを判定する必要がある等、処理が複雑になる。実データもコピーする場合は、リソース使用量が増えるが、処理は簡単である。また、左半分にのみ出現するデータおよび分割線上に出現するデータを抜き出して左ページのデータとすることも可能である。この場合、領域判定が必要となり分割処理に時間的なコストがかかることになる。いずれの方法であってもレイアウトは適用可能である。

【0225】

次に、ステップS4140に進み、ページ情報に分割情報を設定する。図33に示す分割方法をとる場合、図34に示すように、右側半分もしくは左側半分であることを示すようなフラグをセットする。原稿がPortraitで、上下に分割する場合には、上半分／下半分を示すフラグをセットすればよい。3分割する場合は、右／中／左、もしくは上／中／下のようなフラグというように分割方法に応じたフラグをセットすればよい。この分割情報と、ステップS4120で処理したページIDとに基づき、分割の内容を特定することができる。

【0226】

次にステップS4150に処理を進め、新たに作成した分割後のページをそれぞれ単独のページとして管理できるようにする。

【0227】

そしてステップS4160に処理を進め、分割前のページを分割後のページで置き換える。すなわち、ステップS4600の処理を完了することにより、元となる原稿データは、個別のページデータ単位で管理することが可能となり、その個別の管理データに基づき、レイアウトを実行すれば、元の原稿データは復元可能となる。

【0228】

図33で示すような管理情報を保持している場合に、第2ページ目を左右に分割した場合のデータの例を図40に示す。図中にあるように、中間ファイルの構造としては、原稿ページの第2ページ目を分割する処理により、分割されたそれぞれがページ単位に管理されることとなる。すなわち、分割にかかる情報が、2ページ目と3ページ目のページ情報としてそれぞれ右側、左側を示す分割情報を保持して格納される。

【0229】

分割後のページ情報は、分割の属性を示すデータを除き、通常のページ情報と同じ形式で管理されるので、分割にかかるページであってもレイアウトの方式によっては区別なく扱うことも可能である。図40におけるページ情報1は通常のページデータを示し、ページ情報2及び3はそれぞれ分割にかかるページ情報であり、その内容として分割の属性を示すフラグ情報（例えば、原稿データの左側か、右側のデータを特定する情報）を有する。実際にレイアウトする場合には、分割ページ相互間の関連性を維持する必要がある、その関連性の有無は分割情報及びページIDにより判断される。

【0230】

10

20

30

40

50

実際に、ページ単位に管理されているデータに基づいてレイアウトをする場合の処理の流れを図42に示す。ステップS4210でページの配置順を決定する。これは、NUP印刷の場合、1枚目の用紙の表面には、右側、左側にそれぞれどのページを配置するかというレベルの配置順指定であり、細かな座標指定の準備段階である。この段階で、どのページと、どのページが隣接するかなどが確定する。

【0231】

次にステップS4220に処理を進め、ページの分割情報に基づき、隣接しているのが分割ページかどうかを判断する。単なる2UP印刷であれば隣の2ページ分のデータで判定可能であるが、4UPの場合には、上の2つ、下の2つ、右側の2つ・・・というように、4つの配置をすべて調べる必要がある。9UP16UPと増えた場合もすべてを調べる必要がある。しかし、必要なのは分割属性を持つページだけなので調べるべきパターンはある程度限定可能である。

10

【0232】

ステップS4220で分割ページ同士が並んでいないと判定された場合は(S4240-No)、特別なレイアウトをする必要がないので、ステップS4240に進み、通常のレイアウト処理を行い描画位置やサイズを確定する。

【0233】

ステップS4220の処理で分割されたページが隣接していると判断された場合には(S4220-Yes)、処理をステップS4230へ進め、その隣接ページが泣き別れのページの復元処理が必要かを判断する。

20

【0234】

本来であれば、あるページを左右に分割してできた2ページの右側が相対的に右側、左半分が相対的に左側に位置する場合には復元処理が可能である。システムによっては、同一ページを分割したことを判定に入れない場合もある。この場合には、異なるページのデータがひとつのページと判断してもよい。また、もっと簡単に、分割されたページであれば常に復元処理を行う場合もある。

【0235】

ステップS4230で泣き別れのページに該当し、復元処理が可能であると判断された場合には(S4230-Yes)、図38にあるように分割前のイメージとなるように位置や大きさをレイアウトする。ステップS4230で、泣き別れのページに該当しない、若しくは、元のデータのように復元する必要があると判断された場合には(S4230-No)、ステップS4260へ処理を進め、図37で説明したような、分割ページのレイアウト制御を行う。

30

【0236】

図38のレイアウト3830のように元ページのイメージを復元することが可能である場合には、左右それぞれでクリップをかけて印刷するのではなく、もともと1ページのデータとして描画することも可能である。このような処理をする場合には、ページデータ自体を左右で分割するようなデータ量削減処理をしていないことが前提となる。

【0237】

図43は図42のステップS4260の処理を具体的に示すフローチャートである。図37では、ページ単位のレイアウト処理として、(1)移動、(2)縮小、(3)縮小と移動の組み合わせ、の3つの態様によりレイアウト処理をすることが可能であることを説明したが、そのレイアウト処理の具体的な処理は、図43の各ステップの処理に基づくことになる。

40

【0238】

ステップS4310では、中間ファイルとしてページ単位に管理されているページデータと、レイアウトしようとするページの印刷可能領域とを比較する。その結果に基づき、ステップS4320で、ページデータが印刷可能領域にレイアウト可能か否かを判断する。レイアウト可能か否かの判断は、単純にページデータの貼り付けができるか否かに限られず、例えば、分割したページデータをレイアウトした場合に、データの欠落が生じるか

50

否かも含めて判断する。

【0239】

ステップS4310での判断に基づき、レイアウトが可能と判断された場合（S4320－Yes）、処理をステップS4330に進め、通常のページレイアウトを実行する。

【0240】

一方、ステップS4320の判断に基づき、そのページデータが印刷可能な領域に収まらなと判断された場合は、ステップS4340以降のページレイアウトを制御する処理に移行する。ステップS4340ではページレイアウトをデータの平行移動より制御することが可能か否かを判断する。「平行移動」とは、図37の「(1)移動」を説明する図で示したように、製本印刷において、綴じ代部分（例えば37eの幅）を確保するために、ページデータの配置位置を左側にシフトしてデータの欠落を防止するレイアウトの態様である。

10

【0241】

ステップS4340で、平行移動させても、印刷可能領域にページデータを収めることができないと判断された場合、例えば、図37の「(1)移動」を説明する図で、左側にページデータをシフトした場合に、ページ中央部の「○」印は印刷可能領域の収めることができるようになるが、左側にあるページデータが欠落する場合もある。このように平行移動によってもページデータを十分にレイアウトすることができない場合は（ステップS4340－No）、処理をステップS4360に進める。

【0242】

平行移動が可能な場合は（ステップS4340－Yes）、処理をステップS4350に進め、ページデータを平行移動させたレイアウト制御を実行する（S4350）。

20

【0243】

ステップS4370において、レイアウト条件（縮小の条件）が重畳して与えられているか否かを判断し、縮小のレイアウト条件が課されている場合は、処理をステップS4380に進める。重畳した条件が与えられていない場合は、そのまま処理を終了する。

【0244】

ステップS4360で縮小が可能と判断された場合は（S4360－Yes）、ページデータを縮小したレイアウトページ制御が実行される（S4380）。

【0245】

図43の各ステップの処理によらず、オペレータは、その分割ページのレイアウト方式を直接に指定することも可能である（S4385、S4390）。

30

【0246】

また、各ステップにより設定され、若しくはオペレータにより選択されたレイアウト条件、例えば、平行移動のシフト量、縮小の倍率等は、ホストコンピュータ側のメモリ（102、103）に格納され、印刷情報全体に統一して反映させることができる。

【0247】

本実施形態によれば、原稿ページを分割して、分割されたページの関連性を保持しつつ複数の個別のページとして管理し、製本印刷におけるデータの復元、データの欠落を防ぐレイアウト制御が可能となる。

40

【0248】

＜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、インタフェース機器、リーダ、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。

【0249】

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モジュール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モジュールを読み出し実行することによっても、達成される。

50

【0250】

この場合、記憶媒体から読み出されたプログラムモジュール自体が前述した実施形態の機能を実現することになり、そのプログラムモジュールを記憶した記憶媒体は本発明を構成することになる。

【0251】

プログラムモジュール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。

【0252】

また、コンピュータが読み出したプログラムモジュール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、そのプログラムモジュールの指示に基づき、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。

10

【0253】

さらに、記憶媒体から読出されたプログラムモジュールが、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、そのプログラムモジュールの指示に基づき、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。

20

また、ホストコンピュータに本発明を適用することにより製本の体裁をなす印刷物における各印刷領域に対して、複数の印刷ページデータを配置した印刷物を得るための印刷データを生成することが可能になる。

【0254】

また、本発明によれば、用紙に印刷した結果を製本した場合の1ページ上にさらに複数の印刷データページを組み合わせて指定できる配置データを入力できるので、ユーザが望む配置方式を選択することが可能にする。

【0255】

また、他の発明によれば、前記入力手段は、1ページ上の複数ページデータの配置順を入力できるので、ユーザが望む配置方式を選択することが可能になる。

30

【0256】

また、本発明によれば、表紙のレイアウトと表紙以外の面のレイアウトとで異なる配置データを入力できるので、より高度にユーザが望む配置方式を選択することが可能になる。

【0257】

また、本発明によれば、見開き面を一つの印刷領域としてデータを出力させる指定ができるので、より高度にユーザが望む配置方式を選択することが可能になる。

【0258】

また、本発明にかかる、印刷制御装置及び方法によれば、原稿ページを分割して、分割されたページの関連性を保持しつつ複数の個別のページとして管理し、製本印刷におけるデータの復元、データの欠落を防ぐレイアウト制御が可能となる。

40

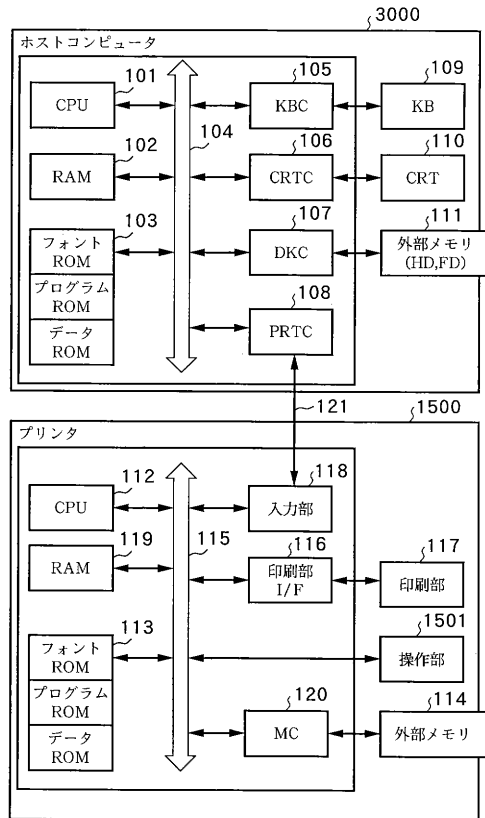
【0259】

また、本発明によれば、表紙、及び裏表紙、あるいは見開きサイズに対するレイアウトの面付け制御を容易に実行することが可能となる。

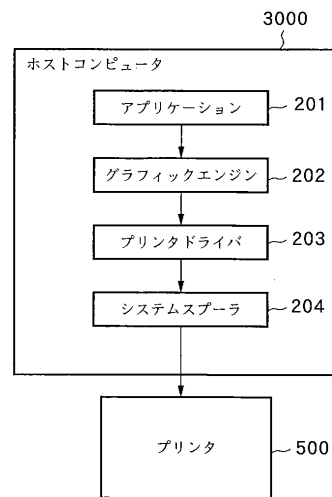
【0260】

また、本発明によれば、原稿ページが縮小レイアウトされる際の倍率値のばらつきを防止し、均一な倍率での印刷結果を得ることが可能となる。

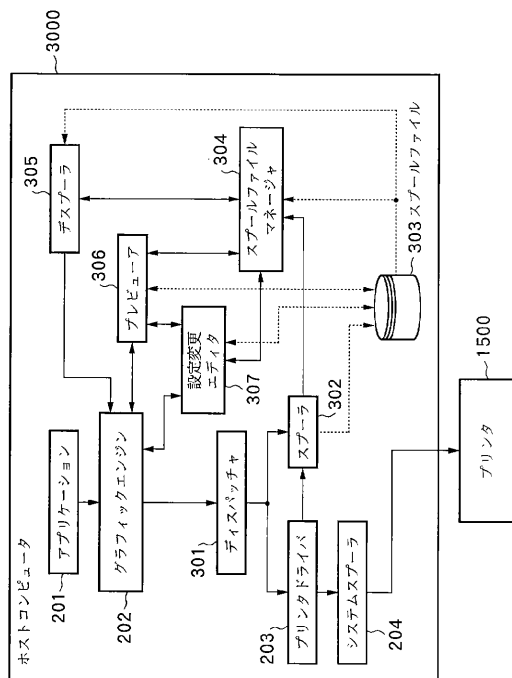
【図 1】



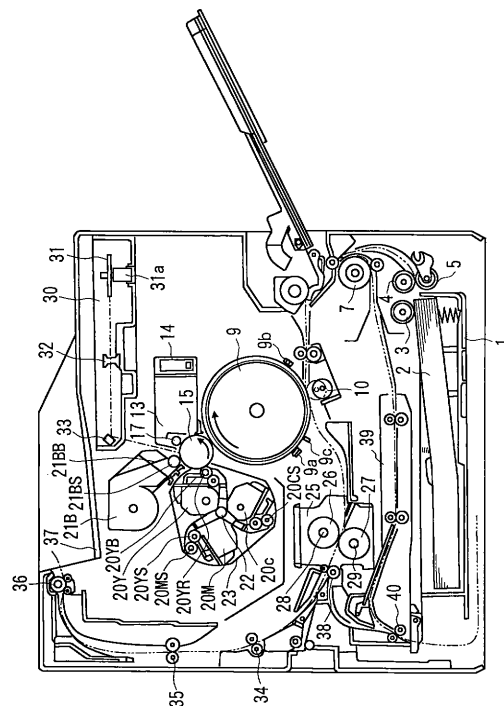
【図 2】



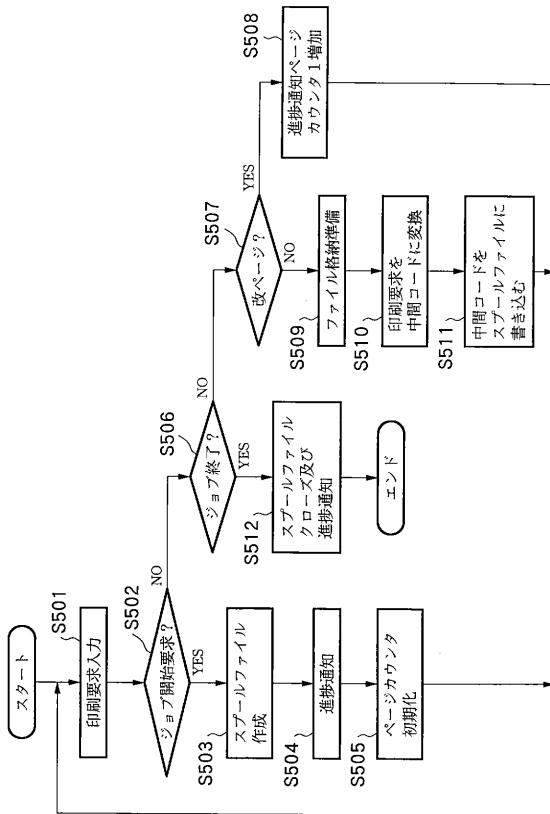
【図 3】



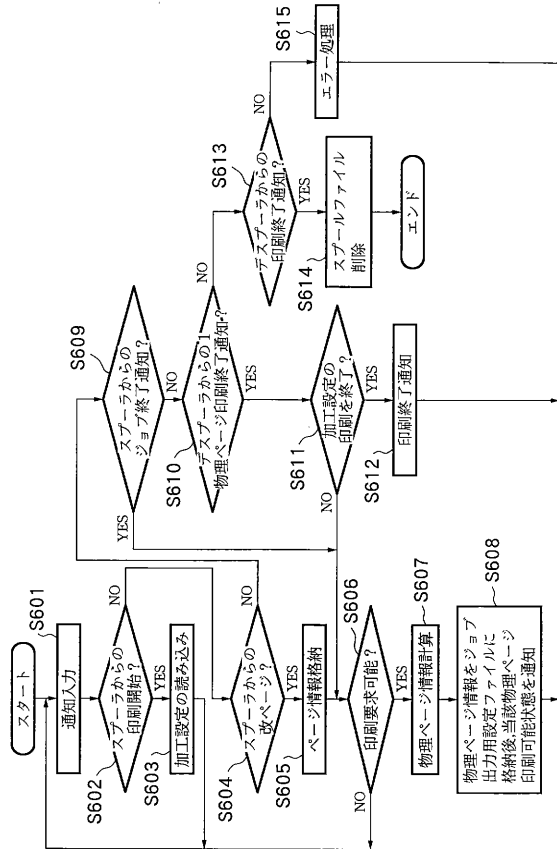
【図 4】



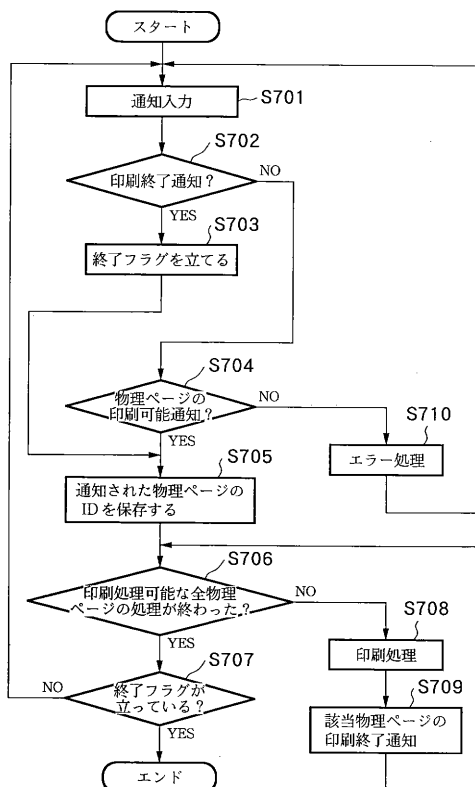
【図 5】



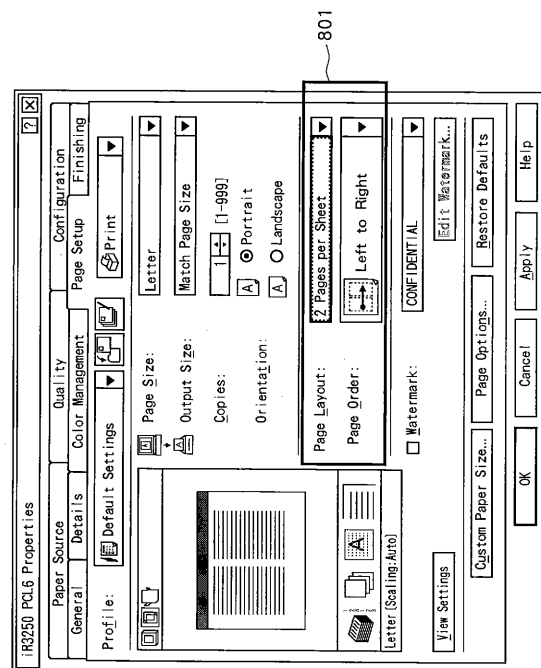
【図 6】



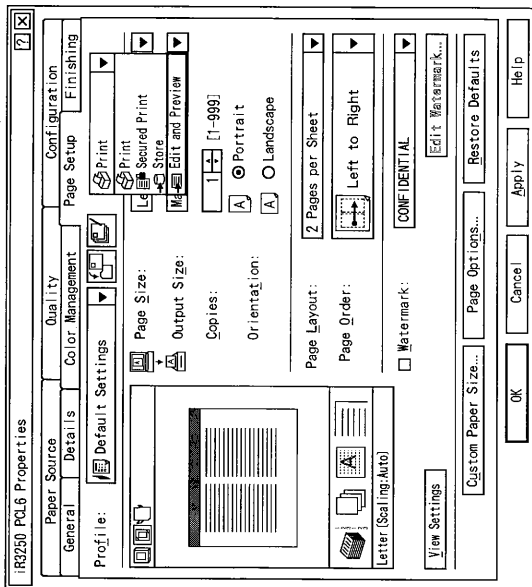
【図 7】



【図 8】



【図 9】



【図 10】

ジョブを識別可能な ID	1001
ジョブ設定情報	1002
ジョブの物理ページ数	1003
一つ目の物理ページ情報	1004
二つ目の物理ページ情報	1005
....	1006
最後の物理ページ情報	1007

【図 11】

全物理ページ数	1101
全論理ページ数	1102
部数	1103
部単位印刷	1104
フィニッシング情報	1105
付加印刷情報	1106

【図 12】

物理ページ番号	1201
物理ページ設定情報	1202
物理ページに割り付ける論理ページ数 n	1203
一つ目の論理ページの情報	1204
二つ目の論理ページの情報	1205
....	1206
n 個目の論理ページの情報	1207

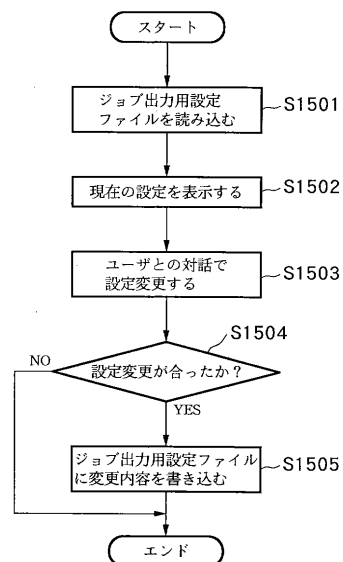
【図 13】

物理ページ上への論理ページの配置順	1301
両面印刷の表面か裏面か	1302
カラーページかモノクロページか	1303
付加印刷情報	1304

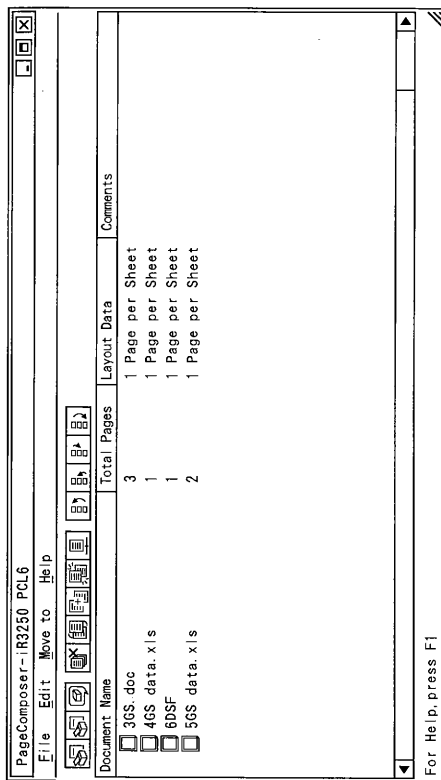
【図 14】

論理ページを識別可能な ID	1401
論理ページ番号	1402
フォーマット情報	1403

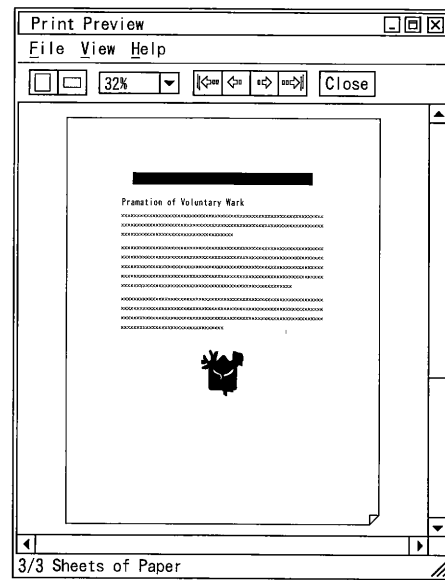
【図 15】



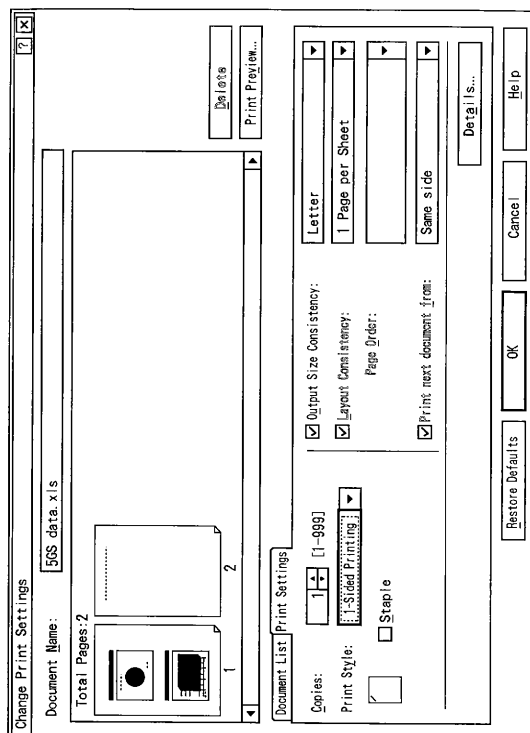
【図 16】



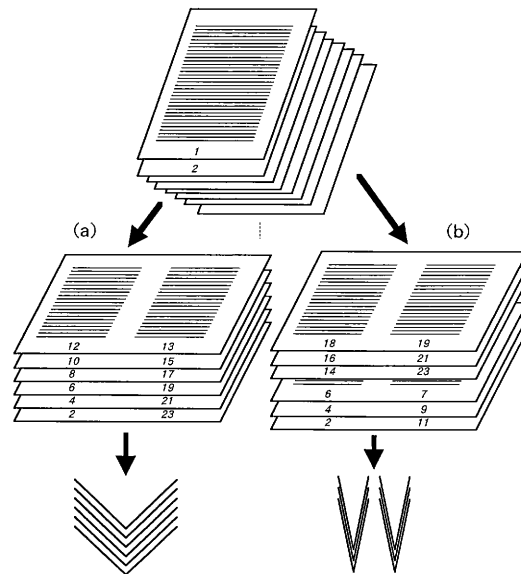
【図 17】



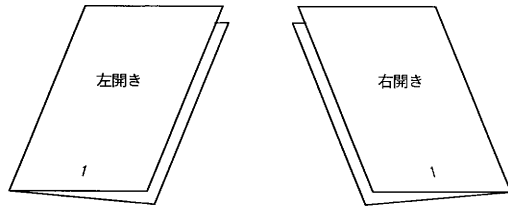
【図 18】



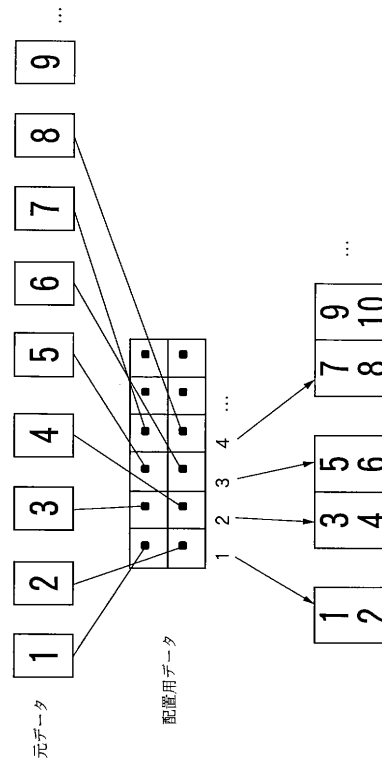
【図 19】



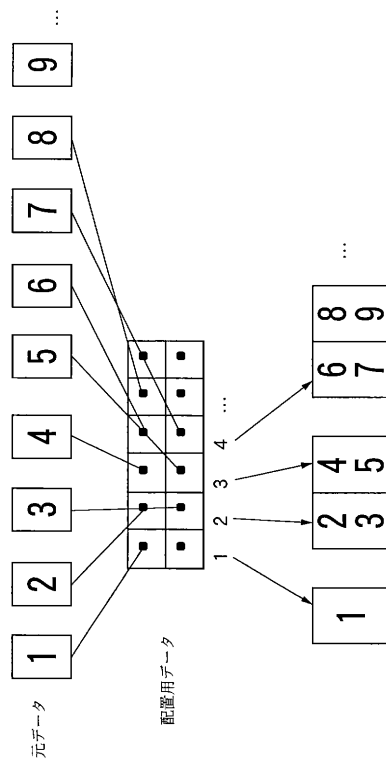
【図 20】



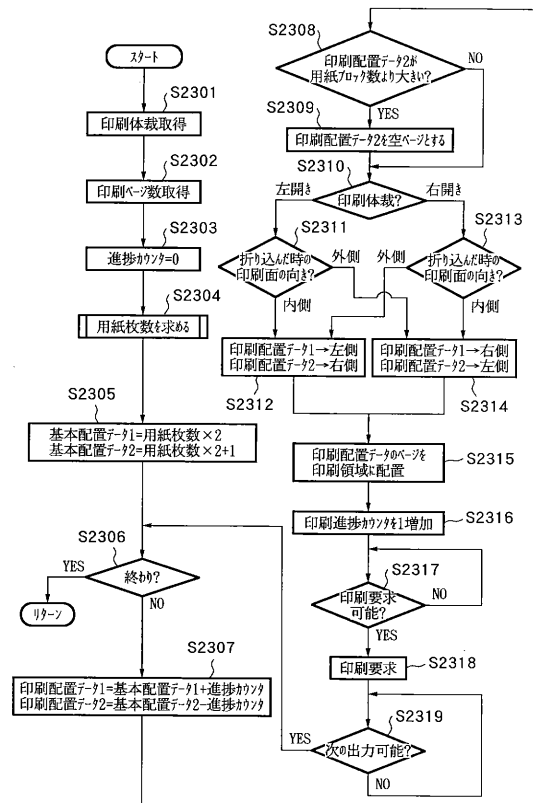
【図 21】



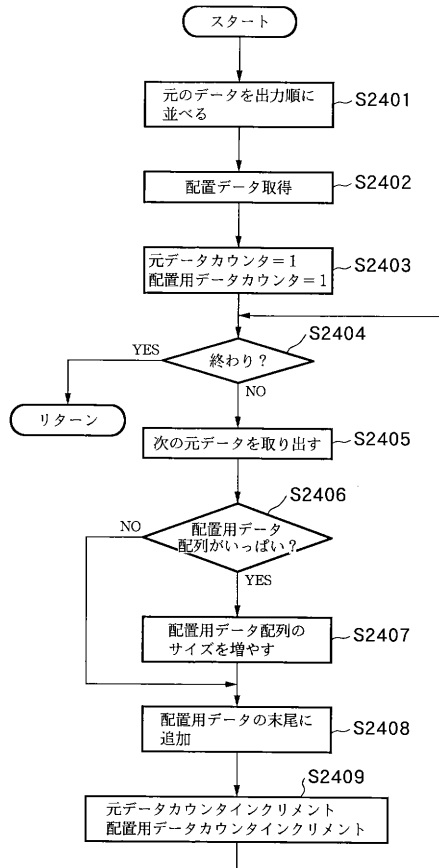
【図 22】



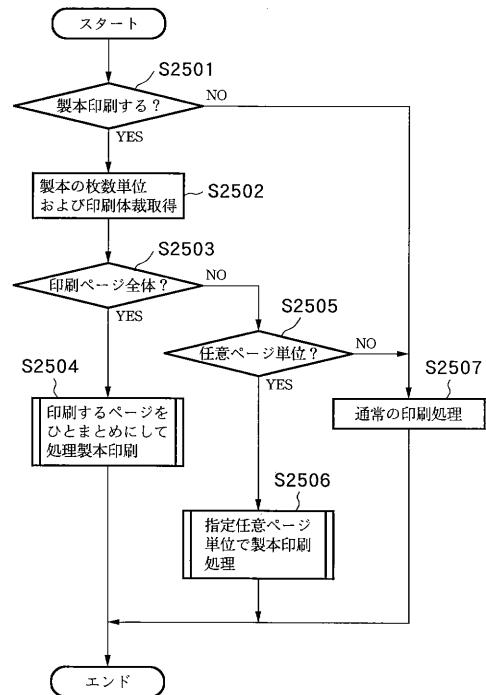
【図 23】



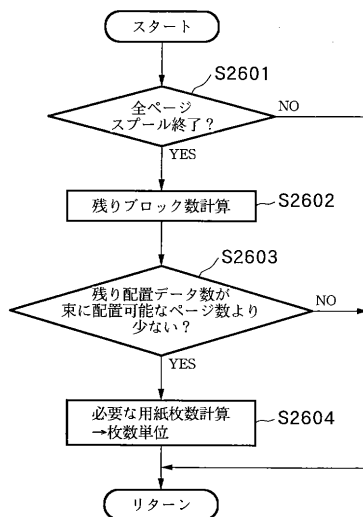
【図 24】



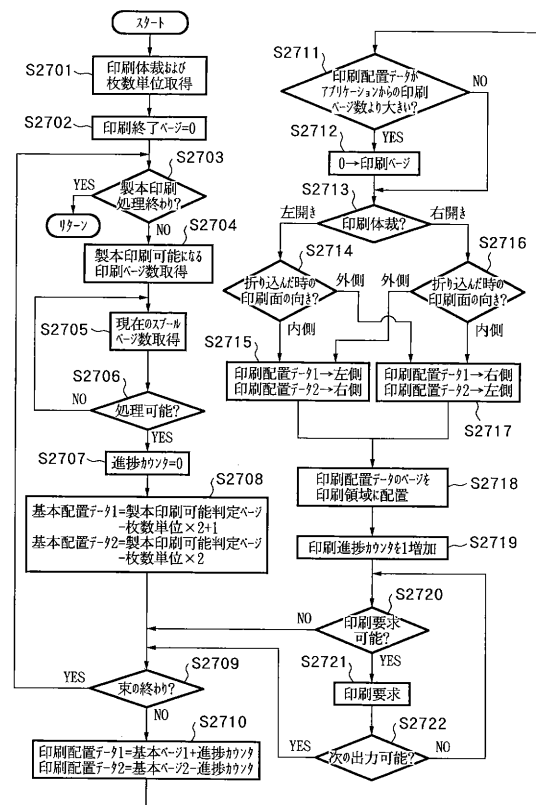
【図 25】



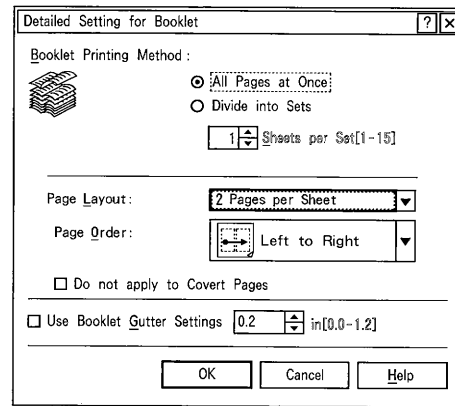
【図 26】



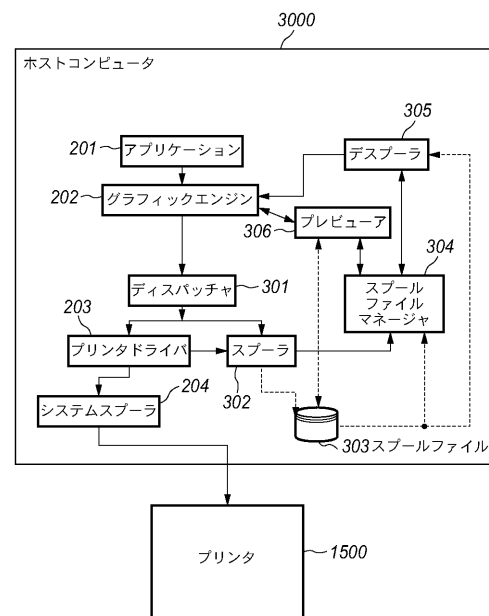
【図 27】



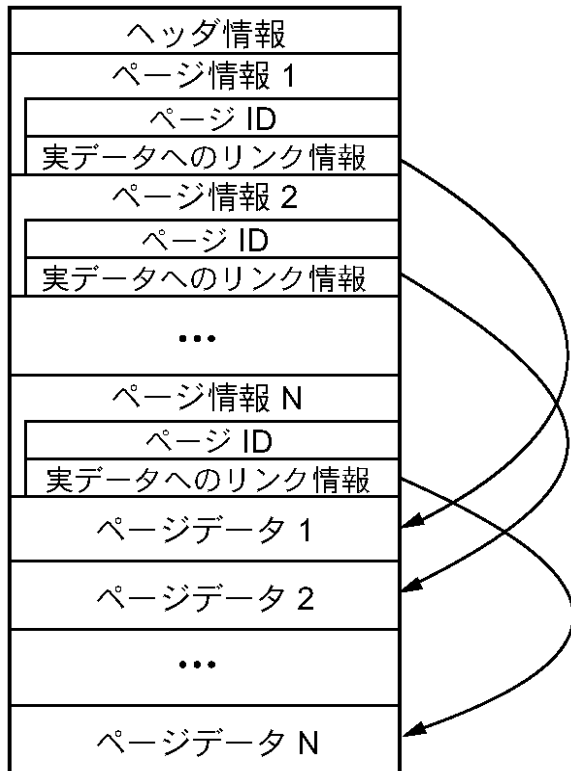
【 図 2 9 】



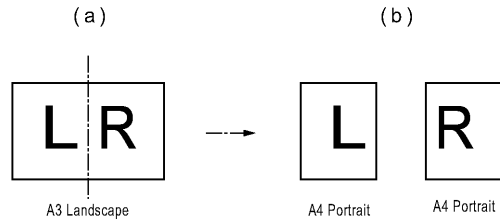
【图 3 1】



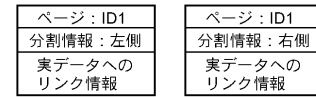
【図 3 2】



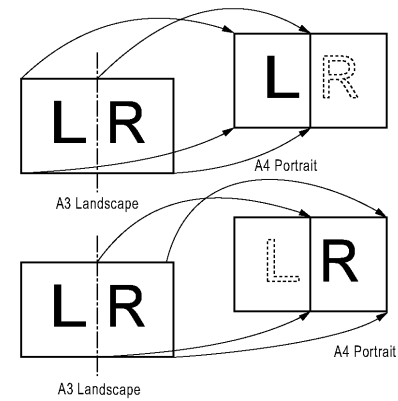
【図 3 3】



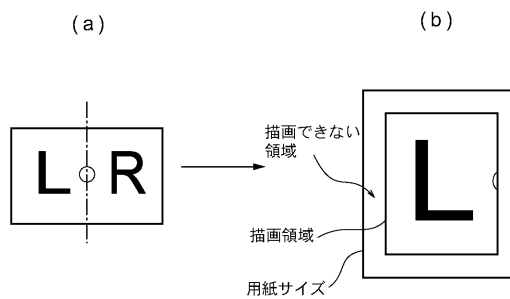
【図 3 4】



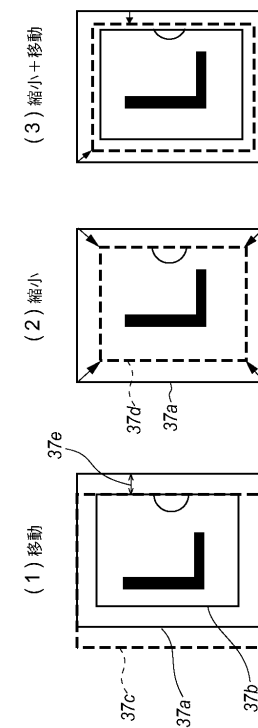
【図 3 5】



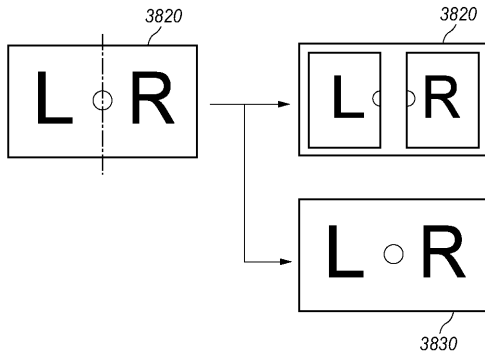
【図 3 6】



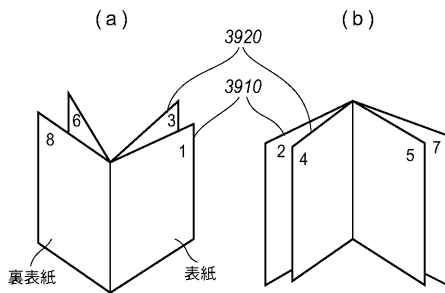
【図 3 7】



【図 38】



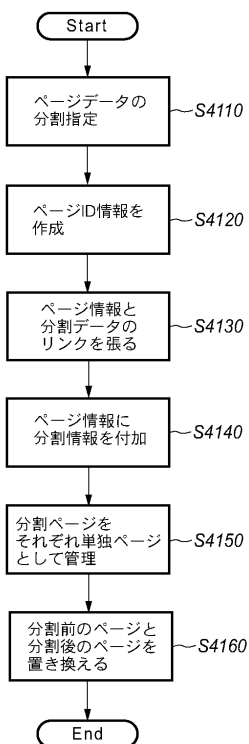
【図 39】



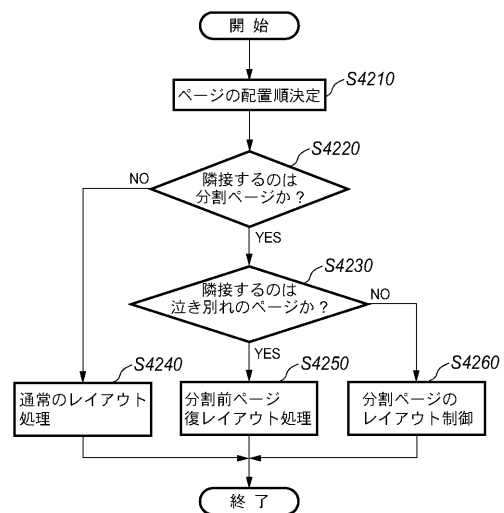
【図 40】



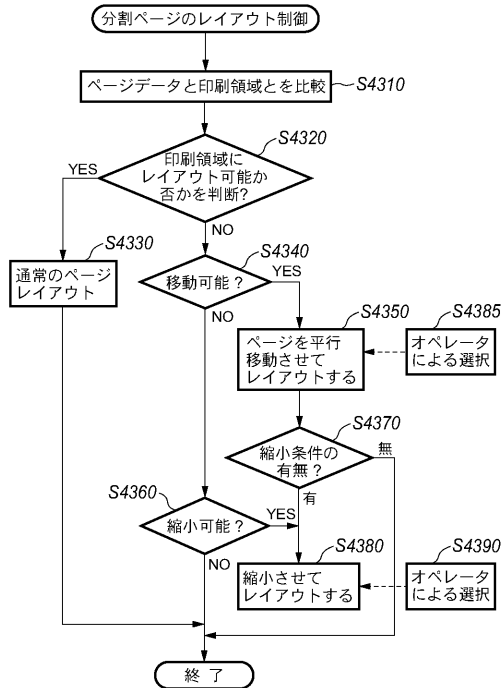
【図 41】



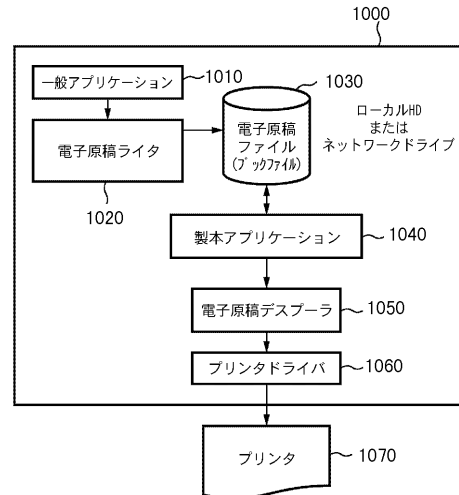
【図 42】



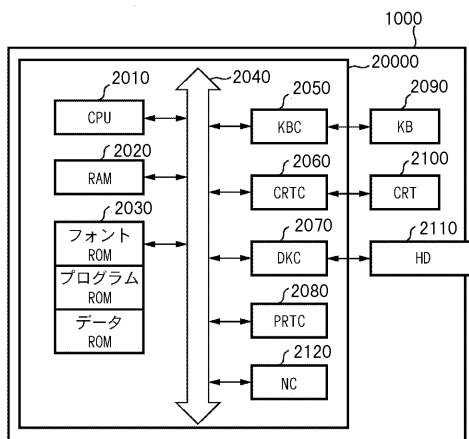
【図 4 3】



【図 4 4】



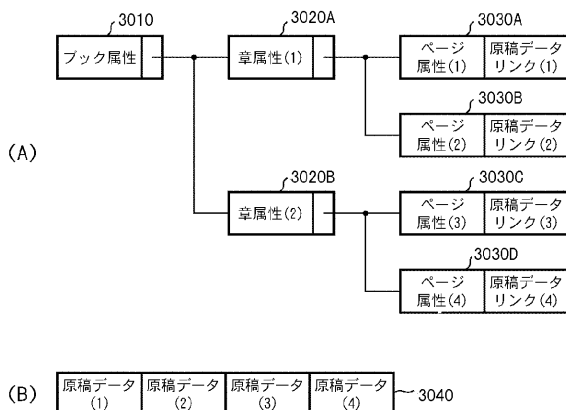
【図 4 5】



【図 47】

NO	属性情報	画面/画面/製本印刷	備考
1	印刷方法	1	
2	用紙サイズ	原稿サイズ/固定サイズ	・ I44+A3、「[94H3]」[「センター+レンジャー」(11×17)指定の場合は折り指定 ・ 製本印刷またはN-up印刷が指定された場合は1章/1ページ目の原稿サイズが自動で選択される ・ 固定サイズの指定のみ選択可能 ・ シフト/拡張指定が可能 ・ 配位位置は9/8パターン ・ 等倍印刷/縮小印刷指定可能
3	用紙方向	縦/横	
4	とじ代/とじ方向		
5	N-up印刷	ページ数/配/選/画/境界線/配位位置等	
6	拡大縮小	ON/OFF	用紙サイズに固定サイズのまはN-up印刷を選択した場合は自動でONに指定、OFFに指定可能
7	ウォーターマーク		・ 絵画ページ単位、物理ページ単位に個別の指定が可能 ・ 全章/全ページが対象
8	ヘッダ・フッタ		・ 論理ページ単位、物理ページ単位に個別の指定が可能 ・ 全章/全ページが対象
9	排紙方法	スタイルフル/パンチ穴	・ スタイルフル/パンチは片面/両面印刷のみ ・ スタイルフルは1箇所/2箇所
10	製本詳細	開き方向/中とし/拡張指定/とじ代/分冊指定等	・ 製本印刷時のみ
11	表紙・裏表紙		・ 表紙1/2、裏表紙1/2に対する印刷指定 ・ 総括口(インサータを含む)指定
12	インデックス紙		・ インデックス部分の文字列印字、インデックス紙上へのフタデ書きを指定可能 ・ 製本印刷は指定不可
13	台紙		・ 総括口(インサータを含む)指定 ・ 挿入用紙に原稿データを印刷可能 ・ 製本印刷は指定不可
14	章区切り	「なし」/「ページかえ」/「用紙かえ」	・ インデックス紙、台紙が指定された場合は「用紙かえ」固定 ・ 片面印刷は「用紙かえ」

【图 4 6】



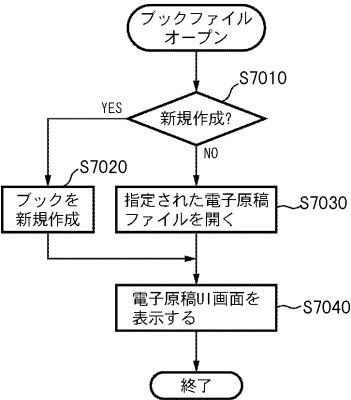
【図 4 8】

備考		
原紙情報	原稿サイズ/固定サイズ	固定サイズ選択の場合は「用紙かえ」を自動指定 ・ プラットで複数用紙が選択された場合は、 指定用紙でのみ変更可能。ブックにあわせる指定 でも用紙サイズの変更可能
1 用紙サイズ		
2 用紙方向	縦/横	固定サイズの場合はのみ選択可能
3 N-up印刷指定	ページ数/配置順/境界線/ 配置位置等	・ 配置位置は9/16パターンの ・ 等倍印刷指定可能
4 拡大縮小	ON/OFF	・ 用紙サイズに固定サイズのまたはN-up印刷を 選択した場合は自動でONに指定。OFFに指定可能
5 ウォーターマーク	表示/非表示	・ プラットで指定された全ウォーターマークを表示するか しないかの指定
6 ヘッダー・フッター	表示/非表示	・ プラットで指定された全ヘッダ・フッタを表示するか しないかの指定
7 排紙方法	ステイブル	・ プラットでステイブル指定された場合に、OFFが可能 デフォルトはON

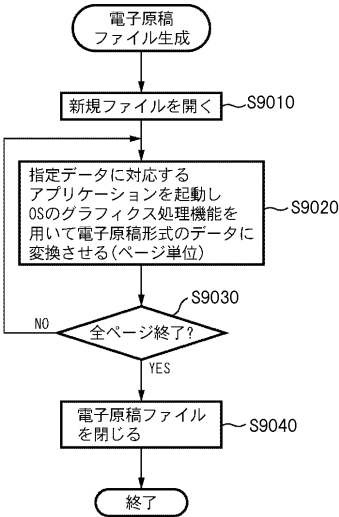
【図 4 9】

備考		
属性情報		
1 ページ回数指定		・ 0/90/180/270度の指定が可能
2 ウォーターマーク	表示/非表示	・ プラットで指定された全ウォーターマークを表示するか しないかの指定
3 ヘッダー・フッター	表示/非表示	・ プラットで指定された全ヘッダ・フッタを表示するか しないかの指定
4 ズーム	50%-200%	・ 仮想的なページ領域にフィットしたサイズを 100%とした相対倍率指定
5 配置位置		・ 固定9/16パターンおよび任意位置指定
6 アノテーション		
7 Variabel項目		
8 ページ分割		

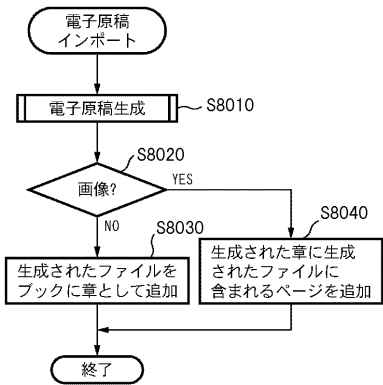
【図 5 0】



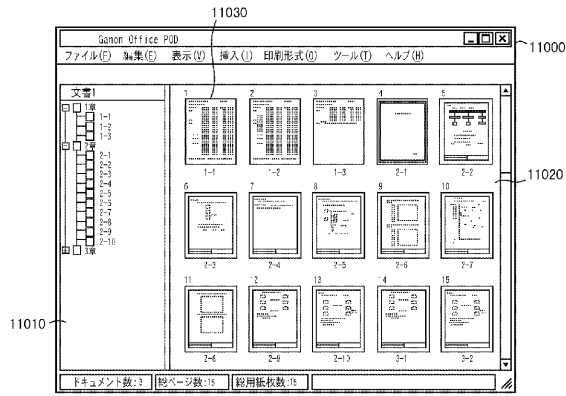
【図 5 2】



【図 5 1】



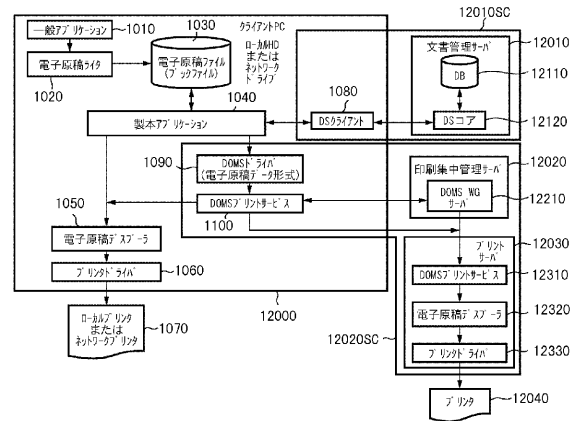
【図 5 3】



【図 5 4】



【図 5 5】



フロントページの続き

(72)発明者 森 安生

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 西川 智

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 中桐 孝治

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

Fターム(参考) 2C187 AC07 AD03 AD04 AE07 AF01 AF03 BF42 DB04 DB27