

(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2006-202025
(P2006-202025A)

(43) 公開日 平成18年8月3日(2006. 8. 3)

(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	C	2 C 1 8 7		
B 4 1 J 5/30 (2006.01)	B 4 1 J	5/30	Z	5 B O 2 1		

審査請求 未請求 請求項の数 3 O L (全 21 頁)

(21) 出願番号	特願2005-12788 (P2005-12788)	(71) 出願人	000001007
(22) 出願日	平成17年1月20日 (2005. 1. 20)		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	100090538
			弁理士 西山 恵三
		(74) 代理人	100096965
			弁理士 内尾 裕一
		(72) 発明者	金谷 航
			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
		最終頁に続く	

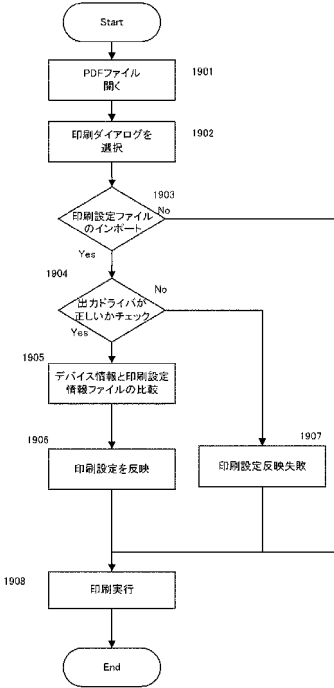
(54) 【発明の名称】 文書処理方法及び印刷制御方法及びそのシステム

(57) 【要約】

【課題】 印刷体裁が設定された文書ファイルからPDF等の共通フォーマットでファイル変換しエクスポート出力する場合において、エクスポートした文書ファイルを任意にプリンタで印刷する場合も印刷体裁がまもられる仕組みを提供する。

【解決手段】 文書データを印刷か別フォーマットのファイルのエクスポート出力かの処理の違いを判定し、ファイルのエクスポート出力時にはプリンタの能力を実際のエクスポートドライバ能力よりも高度な状態に変換することで、印刷体裁を電子データファイルに反映して出力させる。

【選択図】 図19



【特許請求の範囲】

【請求項 1】

文書を編集操作に応じて編集可能で且つ印刷体裁を設定する文書処理方法であって、文書データを印刷か別フォーマットのファイルのエクスポート出力かの処理の違いを判定する手段と、ファイルのエクスポート出力時にはプリンタの能力を実際のエクスポートドライバ能力よりも高度な状態に変換するドライバ能力変換手段と印刷設定による表示結果を電子データファイルに反映して出力させる変換手段とを備えることを特徴とする文書処理装置及びその制御方法。

【請求項 2】

請求項 1 の制御方法は操作によって原稿データでのファイル出力、及び面つけ制御方法を選択することができることを特徴とする。 10

【請求項 3】

出力する電子データファイルだけでなく、文書処理装置で設定を行った印刷設定情報（出力ドライバ情報、レイアウト設定、両面／製本設定等）を付加またはファイルとして出力させる手段と出力した電子データファイルと印刷設定情報を再度インポートした場合に、ファイル出力する前を再現できる機能を有することを特徴とする請求項 1 の文書処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、電子ファイルデータのエクスポート出力に関し、特にパーソナルコンピュータ等の情報処理とプリンタとを有するシステムにおける文書処理方法及び文書処理装置に関するものである。 20

【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、データの種類の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。 30

【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、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。 40

【0005】

また、データをページ単位で保持し、各種設定値にしたがって、用紙上にレイアウトして印刷するような文書処理システムがある。このようなシステムでは、データを階層構造で保持していても、設定値は文書全体でしか保持しておらず、部分的な設定をすることはできなかった。例えば、アプリケーションプログラム等で作成された文書について、1 枚 50

のシートの片面上にNページ(Nとしては例えば2, 4, 9, 16ページがある。)を配置するレイアウト設定(Nページ印刷あるいはNイン1印刷と呼ぶ。)が可能な印刷システムがあった。Nページ印刷を例に説明すると、文書全体について2ページ印刷の設定はできるが、2ページ印刷が指定された文書の一部分を4ページ印刷に変更することはできなかった。すなわち、ひとつの設定項目について、文書の一部分について文書全体と異なる設定値を設定することができなかった。また、文書の部分毎にレイアウト等の設定ができるようなシステムでは、各部分だけで属性(レイアウト設定等の設定値)を保持しているために、全体を統一して一斉に変更することができなかった。

【0006】

そこで、文書データを、たとえばページ、ページのまとまり、文書全体といった単位で階層的に保持するとともに、各種設定項目の値も文書データの各階層ごとに保持することで、文書全体に統一的な設定をすることも、その一部について部分的な設定をすることも可能であるような文書処理システムが登場した(特許文献1等参照)。

【0007】

このようなシステムでは、編集操作として、各設定項目の設定値の変更だけでなく、ページ単位やページをまとめた単位など任意の単位を文書内で、あるいは文書外に移動したりコピーペーストしたりする事が可能である。

【0008】

また、プリンタに印刷を行う際には、アプリケーションやプリンタドライバの機能を用いて、様々な印刷体裁設定を行うことが可能である。印刷体裁設定は、プリンタの機能によって、設定可能な設定項目と設定不可能な設定項目がある。従って、すべてのプリンタで同じ印刷体裁設定画面を用いて、ユーザに設定値を入力させると、設定不可能な設定を行われてしまい、設定値にコンフリクトが生じることがある。また、ある2つの設定項目に何かしらの関係が存在し、1つの設定項目がある値の時には、もう1つの設定項目の設定値はある値を設定することができないなどの関係がある場合がある。従って、この場合にもすべての設定値を表示する印刷体裁設定画面を用いると、コンフリクトが生じることがある。

【0009】

そこで、コンフリクトが生じる設定を行わせないように制御するために、コンフリクトが生じる設定値を表示しない手法が用いられている。これは、コンフリクトがプリンタの機能によって生じる場合には、プリンタの機能情報を予め取得し、機能に基づいた印刷体裁設定画面を表示することで行う。また、コンフリクトが設定値によって生じる場合には、設定値が設定された時に、設定された設定値と他の設定を用いてコンフリクトが生じない印刷体裁画面を表示する手法が用いられている。

【0010】

以上の手法は、通常のデスクトップアプリケーションで行われている。

【特許文献1】特開2003-162404号公報

【発明の開示】

【発明が解決しようとする課題】

【0011】

上記のような手法を使用することで、印刷体裁設定画面でのコンフリクト制御を行うことができる。実際には、印刷体裁設定画面で、プリンタの機能や他の設定値などの必要な情報を利用することで、設定可能な設定項目だけ表示する設定画面を作成し、コンフリクト制御を行う。しかしこの制御ではドライバの能力範囲での出力結果しか得られない。例えば印刷体裁設定を施したアプリケーションデータファイルを用紙への印刷出力ではなく、ドライバを利用してPDFなどの共通フォーマットのデータファイルとしてエクスポートする場合には、従来ではドライバのコンフリクト処理が動作してしまうため、製本体裁が施された状態ではなく、原稿ページがファイル変換され、出力されていた。あるいは、印刷レイアウトは反映されるがシフト幅やページ順など製本体裁が施された印刷出力物と同一の表示形態でファイル出力に反映される形態ではなかった。例えば、両面印刷時にス

10

20

30

40

50

テーブル設定を行った際にとじ代幅の反映ができなかった。

【 0 0 1 2 】

このことは、文書出力に際して生産性を低下させるとともに、文書の操作性や内容の視認性を低下させることになっていた。

【 0 0 1 3 】

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、印刷、製本設定された文書ファイルから P D F 等の共通フォーマットでファイル変換しエクスポート出力する場合において、製本設定による表示をエクスポートする電子データファイルに反映することを目的とする。ここでいう製本設定とは、アプリケーションが設定可能な全てのレイアウト機能を指すので、用紙上に複数のページを配置する N ページ設定や、印刷後に断裁を想定して用紙上に同一ページを複数配置する多連印刷設定も勿論含まれる。

10

【 0 0 1 4 】

また、ファイルへのエクスポートを行う場合に、印刷体裁を反映させず、文書の原稿ページをそのままファイルに出力させるモードと全ての印刷設定における表示をエクスポートするファイルに反映させる印刷向けモードを提供し、ユーザが選択することができる機能を提供することを目的とする。

【 0 0 1 5 】

また、本発明では、多連印刷における印刷情報の制御において、エクスポートするだけでなく文書処理装置で設定した印刷設定（ドライバの種類、両面、綴じ代、ステイプル等）をパラメータファイルとして出力する機能を有し、ドライバより出力したパラメータファイルを読み込むことによってエクスポートしたファイルを印刷出力する際に、印刷設定を反映させることを目的とする。

20

【課題を解決するための手段】

【 0 0 1 6 】

上記目的を達成するために本発明の文書処理装置は以下の構成を備える。

【 0 0 1 7 】

即ち文書を編集操作に応じて編集可能で且つ印刷体裁を設定する文書処理方法であって、文書データを印刷か別フォーマットのファイルのエクスポート出力かの処理の違いを判定する手段と、ファイルのエクスポート出力時にはプリンタの能力を実際のエクスポートドライバ能力よりも高度な状態に変換するドライバ能力変換手段と印刷設定による表示結果を電子データファイルに反映して出力させる変換手段とを備えることを特徴とする。

30

【 0 0 1 8 】

また、上記制御方法は操作によって原稿データでのファイル出力、及び面つけ制御方法を選択することができることを特徴とする。

【 0 0 1 9 】

また出力する電子データファイルだけでなく、文書処理装置で設定を行った印刷設定情報（出力ドライバ情報、レイアウト設定、両面／製本仕上げ設定等）を付加またはファイルとして出力させる手段と出力した電子データファイルと印刷設定情報を再度インポートした場合に、ファイル出力する前を再現できる機能を有することを特徴とする。

【発明の効果】

40

【 0 0 2 0 】

本発明によれば、アプリケーションで文書データを編集し、製本、多連等の印刷体裁設定を施したデータファイルを、共通フォーマットでエクスポートによるファイル交換をして、別のアプリケーションやシステムで印刷してもレイアウトや各種編集結果を反映することができるようになる。

【発明を実施するための最良の形態】

【 0 0 2 1 】

以下、図面を参照しながら本発明に係る実施の形態を詳細に説明する。

【 0 0 2 2 】

< システム概要 >

50

まず、本発明に係る文書処理システムの概要について、図１乃至図１２を参照して以下に説明する。尚、この文書処理システムは、一般のアプリケーションによって作成されたデータファイルを電子原稿ファイルに変換する電子原稿ライタと、その電子原稿ファイルを編集する機能を提供する製本アプリケーションとを含み、作成されたデータを一まとめにした文書の作成及び編集を可能とし、その操作性を向上させて文書編集を効率的に行えるものである。

【００２３】

<システム構成及び動作>

図１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、本発明の情報処理装置に好適な実施形態であるデジタルコンピュータ１０１（以下、ホストコンピュータとも呼ぶ）によって実現されている。図１に示す一般アプリケーション１０１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、ＯＳに対する印刷機能を有している。これらの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際に、オペレーティングシステム（ＯＳ）によって提供される所定のインタフェース（一般に、ＧＤＩと呼ばれる）を利用する。即ち、アプリケーション１０１は、作成したデータを印刷するために、上述の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（ＧＤＩ関数と呼ばれる）を送信する。一方、出力コマンドを受けた出力モジュールは、その出力コマンドをプリンタ等の出力デバイスが処理可能な形式に変換して、変換されたコマンド（ＤＤＩ関数と呼ばれる）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、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い、印刷データを生成し、ＪＬ（Ｊｏｂ Language）でくくることにより印刷ジョブが生成される。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（Ｇｒａｐｈｉｃ Ｄｅｖｉｃｅ Ｉｎｔｅｒｆａｃｅ）と呼ばれるモジュールが相当する。

【００２４】

電子原稿ライタ１０２は、上述のデバイスドライバを改良したものであり、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。但し、電子原稿ライタ１０２は特定の出力デバイスを目的としておらず、詳細は後述する製本アプリケーション１０４やプリンタドライバ１０６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１０２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 of 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、例えばアドビシステムズによるＰＤＦ形式やＳＶＧ形式などが電子原稿形式として採用できる。

【００２５】

一般アプリケーション１０１から電子原稿ライタ１０２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１０２を指定してから印刷を実行させる。但し、電子原稿ライタ１０２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１０２を指定するのは製本アプリケーション１０４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。そして、製本アプリケーション１０４は電子原稿ライタ１０２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、この点を明瞭に識別する必要がある場合、電子原稿ライタ１０２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１０４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。また、特に区別する必要がない場合は、アプリケーションにより生成されるドキュメントファイル、電子原稿ファイル、及びブックファイルをいずれも文書ファイル（または文書データ）と呼ぶ。

10

20

30

40

50

【 0 0 2 6 】

このようにデバイスドライバとして電子原稿ライター 1 0 2 を指定し、一般アプリケーション 1 0 1 により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 1 0 1 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 1 0 3 としてハードディスクなどの記憶媒体に格納される。尚、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良く、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。

【 0 0 2 7 】

製本アプリケーション 1 0 4 は電子原稿ファイル（あるいはブックファイル） 1 0 3 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。但し、製本アプリケーション 1 0 4 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。

【 0 0 2 8 】

製本アプリケーション 1 0 4 によって編集されたブックファイル 1 0 3 を印刷する際には、製本アプリケーション 1 0 4 によって電子原稿デスプーラ 1 0 5 が起動される。電子原稿デスプーラ 1 0 5 は、製本アプリケーションと共にコンピュータ内にインストールされるプログラムモジュールであり、製本アプリケーションで使用するドキュメント（ブックファイル）を印刷する際に、プリンタドライバへ描画データを出力するために使用されるモジュールである。電子原稿デスプーラ 1 0 5 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 1 0 7 用のプリンタドライバ 1 0 6 がデバイスドライバとして指定される。上述の出力モジュールは、受信した出力コマンドをデバイスコマンドに変換して指定されたプリンタ 1 0 7 用のプリンタドライバ 1 0 6 に出力し、そのプリンタドライバ 1 0 6 がプリンタ 1 0 7 で解釈実行可能なページ記述言語等のコマンドに変換する。そして、変換されたコマンドはプリンタドライバ 1 0 6 から不図示のシステムスプーラを介してプリンタ 1 0 7 に送信され、プリンタ 1 0 7 によってコマンドに応じた画像が印刷される。

【 0 0 2 9 】

図 2 は、コンピュータ 1 0 0 のハードウェアブロック図である。図 2 において、CPU 2 0 1 は、ROM 2 0 3 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 2 1 1 から RAM 2 0 2 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 1 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。RAM 2 0 2 は、CPU 2 0 1 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） 2 0 5 は、キーボード 2 0 9 や不図示のポインティングデバイスからのキー入力を制御する。CRT コントローラ（CRTC） 2 0 6 は、CRT ディスプレイ 2 1 0 の表示を制御する。ディスクコントローラ（DKC） 2 0 7 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） 2 1 1 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。PRTC 2 0 8 は、接続されたプリンタ 1 0 7 との間の信号の交換を制御する。NC 2 1 2 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。

【 0 0 3 0 】

< 電子原稿データの形式 >

編集アプリケーション 1 0 4 の詳細を言及する前に、ブックファイルのデータ形式について説明する。ブックファイルは紙媒体の書物を模倣した 3 層の層構造を有する。まず、上位層は「ブック」と呼ばれ、1 冊の本を模倣しており、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、章毎の属性が定義できる。そして、下位層は「ページ」であり、アプリケーション

ンプログラムで定義された各ページに相当する。尚、各ページについてもページ毎の属性が定義できる。更に、一つのブックは複数の章を含んでいてよく、また一つの章は複数のページを含むこともできる。

【0031】

図3に示す(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。図示するように、この例では、ブックファイルにおけるブック、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。一つのブックファイルは一つのブックを含む。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毎のデータを実体として有する。そのため、ページは、その属性値のほかに、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。

10

【0032】

尚、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。

【0033】

図3において、ブック301には、ブック属性が定義されていると共に、2つの章302A、302Bがリンクされている。このリンクにより、章302A、302Bがブック301に包含されていることが表示される。章302Aには、ページ303A、303B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303A、303B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3に示す(B)のように原稿ページデータ304のデータ(1)、(2)を指し示し、ページ303A、303Bの実体が原稿ページデータ(1)、(2)であることを表示している。

20

【0034】

図4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。尚、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。

30

【0035】

図5は章属性の、図6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。

【0036】

図4～図6から明らかなように、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。印刷方法の属性としては、片面印刷、両面印刷、製本印刷の3つの値を指定できる。ここで、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。

40

【0037】

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、表紙及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指定及びインデックス(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。

50

【0038】

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、及び合紙を挿入する場合には、給紙元の指定などを含む。

【0039】

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。

【0040】

章属性に関しては、章に固有の項目はなく、全てブック属性と重複する。従って、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 - u p印刷指定、拡大縮小、排紙方法の5項目である。これらのうち、N - u p印刷指定属性は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 、 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ブ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ブル機能を有するか否かに依存する。

【0041】

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、N u p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。

【0042】

ブック、章、ページについて共通な属性としては、ウォーターマーク属性及びヘッダ・フッタ属性がある。ここで、ウォーターマークとは、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及び下余白に印刷されるウォーターマークである。但し、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。尚、ウォ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ォ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ォ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ォ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。

【0043】

<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及び内容を有している。次に、製本アプリケーション104及び電子原稿ライタ102によるブックファイルの作成手順について説明する。ブックファイルの作成手順は、製本アプリケーション104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。

【0044】

図7は、製本アプリケーション104によりブックファイルを開く際の手順である。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S701)。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S702)。新規に作成されるブックファイルは、図3

10

20

30

40

50

の例で示せば、ブックノード301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インターフェース（UI）画面を表示する（ステップS703）。図11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。この場合は、ブックファイルは実質的な内容を持たないため、UI画面1100には何も表示されない。

【0045】

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S703）、そのブックファイルの構造、属性、内容に従ってユーザインターフェース（UI）画面を表示する。図10は、既存のブックファイルから指定されたブックファイルを表示するUI画面の一例である。UI画面1100は、ブックの構造を示すツリー部1101と、印刷された状態を表示するプレビュー部1102とを含む。ツリー部1101には、ブックに含まれる章、各章に含まれるページが図3に示す（A）のような木構造で表示される。ツリー部1101に表示されるページは原稿ページである。またプレビュー部1102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。

【0046】

さて、開かれたブックファイルには、電子原稿ライタ102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7に示す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10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。

【0047】

図8に電子原稿インポートの手順を示す。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102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S801）。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S802）。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われる。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。また、このような画像データの場合はS8010のようにアプリケーションを起動せずに、画像データから直接電子原稿ファイルを生成することが可能であるため、S8010の処理を省略することも可能である。

【0048】

ここで、画像データでなかった場合には、ステップS801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S803）。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、予め用意された規定値に設定される。

【0049】

また、ステップS802で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S801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S804）。但し、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が与えられ、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が与えられる。例えば、Nup指定などがアプリケーションデータにおいて指定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。

【 0 0 5 0 】

図 9 は、図 8 に示すステップ S 8 0 1 において、電子原稿ライタ 1 0 2 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 S 9 0 1）。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、電子原稿ライタをデバイスドライバとして O S 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 1 0 2 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 S 9 0 2）。その出力先はステップ S 9 0 1 で開いた電子原稿ファイルである。指定されたデータ全てについて変換が終了したか判定し（ステップ S 9 0 3）、終了していれば電子原稿ファイルを閉じる（ステップ S 9 0 4）。電子原稿ライタ 1 0 2 によって生成される電子原稿ファイルは、図 3 の（ B ）に示される原稿ページデータの実体を含むファイルである。

10

【 0 0 5 1 】

< 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。

【 0 0 5 2 】

- （ 1 ）新規追加
- （ 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
- （ 1 0 ）合紙挿入
- （ 1 1 ）インデックス紙挿入
- （ 1 2 ）各原稿ページに対するページレイアウト。

20

【 0 0 5 3 】

このほか、一旦行った編集操作を取り消す操作や、更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の編集操作が可能となる。これらの操作を行うと、図 4、図 5 に示す属性に操作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。例えば、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 N - u p 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。

30

【 0 0 5 4 】

< 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するものである。ここで、利用者が図 1 0 に示した製本アプリケーションの U I 画面 1 1 0 0 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 1 0 4 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、そのジョブチケットを電子原稿デスプーラ 1 0 5 に渡す。一方、電子原稿デスプーラ 1 0 5 は、ジョブチケットを O S の出力コマンド、例えばウィンドウズの G D I 関数に変換し、それを出力モジュール、例えば G D I 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 1 0 6 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。

40

50

【 0 0 5 5 】

ここで、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。

10

【 0 0 5 6 】

電子原稿デスプーラ105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。

【 0 0 5 7 】

<その他のシステム構成>

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を作成・編集できる。但し、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。

20

【 0 0 5 8 】

図12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(Document Output Management Service:文書出力管理サービス)ドライバ109及びDOMSプリントサービスモジュール110、DS(文書サービス)クライアントモジュール108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1200に文書管理サーバ1201、印刷集中管理サーバ1202及びプリントサーバ1203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。

30

【 0 0 5 9 】

尚、図12に示す例では、文書管理サーバ1201と印刷集中管理サーバ1202の両方のサーバがクライアントに接続されているが、何れか一方のみネットワーク上に存在する場合もあり得る。例えば、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108を含む文書管理サーバクライアントシステム1201SCが、また印刷集中管理サーバ1202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1202SCが、スタンドアロン型文書管理システムに追加される。

【 0 0 6 0 】

文書管理サーバ1201は、製本アプリケーション104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1201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1201のデータベース1211に保存される。製本アプリケーション104と文書管理サーバ1201との間のブックファイルの保存及び読み出しは、DSクライアント108及びDSコア1212を介して行われる。

40

【 0 0 6 1 】

印刷集中管理サーバ1202は、クライアント文書管理システム1200に格納された、あるいは文書管理サーバ1201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109およびDOMSプリ

50

ントサービスモジュール１１０を介して印刷集中管理サーバ１２０２のＤＯＭＳＷＧサーバモジュール１２２１に送信される。集中印刷管理サーバ１２０２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１１０を介して電子原稿デスプーラ１０５に電子原稿データを渡し、プリントサーバ１２０３により印刷する場合には、プリントサーバ１２０３のＤＯＭＳプリントサービスモジュール１２０３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。

【００６２】

10

<プレビュー表示の内容>

既に説明した通り、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１０に示したユーザインターフェース画面１１００が表示される。ツリー部１１０１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。まず、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。この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。尚、プレビュー部１１０２の表示にレイアウトは反映されない。次に、第２は印刷ビューモードである。この印刷ビューモードでは、プレビュー部１１０２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。そして、第３は簡易印刷ビューモードである。この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。

20

【００６３】

<ステイブル制御>

次に、ステイブル機能を有するプリンタと接続するコンピュータ１００の製本アプリケーション１０４によって行われるステイブル制御について説明する。

【００６４】

図１３は、ステイブル制御システムの構成を示すブロック図である。図示するように、ステイブル制御システムは、図２に示したホストコンピュータ１００とステイブル機能を有するプリンタ１０７とで構成される。

30

【００６５】

以下、プリンタ１０７の構成について説明する。尚、本発明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，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。

【００６６】

図示するように、プリンタ１０７において、１３０１はプリンタＣＰＵであり、ＲＯＭ１３０２内のプログラム用ＲＯＭに記憶された制御プログラム等や外部メモリ１３０３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１３０４に接続される印刷部Ｉ／Ｆ１３０５を介して、印刷部（プリンタエンジン）１３０６に出力情報としての画像信号を出力する。また、このＲＯＭ１３０２内のプログラムＲＯＭには、ＣＰＵ１３０１の制御プログラム等が記憶され、ＲＯＭ１３０２内のフォント用ＲＯＭには、出力情報を生成する際に使用するフォントデータ等が記憶され、ＲＯＭ１３０２内のデータ用ＲＯＭには、ハードディスク等の外部メモリ１３０３がないプリンタの場合、ホストコンピュータ１００上で利用される情報等が記憶されている。

40

【００６７】

また、ＣＰＵ１３０１は、入力部１３０７を介してホストコンピュータ１００との通信処理が可能となっており、プリンタ１０７内の情報等をホストコンピュータ１００に通知できる。ＲＡＭ１３０８は、ＣＰＵ１３０１の主メモリや、ワークエリア等として機能するＲＡＭであり、図示しない増設ポートに接続されるオプションＲＡＭによりメモリ容量

50

を拡張することができるように構成されている。

【0068】

尚、RAM 1308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。上述したハードディスク(HD)、ICカード等の外部メモリ1303は、メモリコントローラ(MC)1309によりアクセスを制御される。外部メモリ1303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、操作パネル1311は操作のためのスイッチ及びLED表示器等で構成されている。

【0069】

また、上述の外部メモリ1303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良い。更に、図示しないNVRAMを有し、操作パネル1311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良い。

10

【0070】

<本実施形態の例>

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。

【0071】

本実施の形態の係わる製本アプリケーション104では、上述までのように、ブックファイル内のブック、章、ページに対してそれぞれの属性を設定することができる。設定された属性を基に印刷ドライバへのスプールを行うのであるが、本発明は電子データファイルへのエクスポート出力において文書ファイルに設定された印刷設定属性を反映することができる。尚、エクスポートする電子データのフォーマットは別システムで利用できる共通フォーマットである。ここでは、PDFフォーマットでのエクスポートを例として説明する。

20

【0072】

エクスポート機能はアプリケーションファイルから原稿データのPDFデータまたは指定された属性で印刷された状態と同じ面付け状態のPDFデータを出力する機能である。データはユーザが指定したフォルダに出力される。但し、この機能はアプリケーションでアプリケーションファイルを開いている場合にのみ行える機能である。また、本機能でPDFファイルをエクスポートするにはファイル出力用のドライバがインストールされている。編集直後は保存が必要で、エクスポート前に実行される保存は展開ファイルへの保存とする。

30

【0073】

原稿データでのPDFファイルへのエクスポート機能は、アプリケーションで付加したアノテーションデータは対象外とし、アプリケーションファイルで保持している原稿データをそのままPDFファイルとして出力する。このため、ページ分割したページは元ページサイズで出力される。

【0074】

面付けを反映させたPDFファイルへのエクスポート機能は、印刷結果と同じ面付け形式のPDFファイルを出力する。印刷プレビューで表示されている用紙上に複数ページレイアウトするN-UPやアノテーション、スタンプ、ヘッダ、フッタの状態が再現される形で出力される。

40

【0075】

しかし面付けPDFのエクスポートの内部処理ではエクスポート用のドライバへ印刷を行うため、ドライバの能力に応じて設定の丸めが発生する。このため、以下の設定はクリアされた状態となる。

【0076】

- 1.パンチ穴
- 2.Z折り

50

- 3．ステイブル（含、章単位ステイブル）
- 4．サドルステッチ
- 5．トリマー
- 6．両面／製本印刷

よって、上記コンフリクト処理に伴い、PDFのエクスポートファイルでは両面時やステイブル設定によるシフト幅などのレイアウトが反映されない。しかし、本発明の実施形態では上記したドライバの能力によって丸められてしまった両面や上記製本設定による表示をPDFファイルに反映させる機能として、印刷モードエクスポート機能も有する。

【0077】

図14及び図15はアプリケーションの表示画面の例である。図14では、原稿ページのプレビューモードで印刷設定が施されていない状態が表示されている。図15では、アプリケーションでの印刷設定が行われた場合、印刷プレビューモードで印刷出力イメージが表示される一例である。印刷モードでのエクスポート出力は印刷プレビューモードでの表示イメージにて確認できる。

【0078】

図16は印刷モードでのエクスポート出力処理を行うフロー図である。1601のステップではアプリケーションで印刷出力の印刷製本設定を施す。続いてステップ1602でエクスポート出力を選択する。ここで特徴とするのは、複数のエクスポート方法の選択が行えることである。図17はエクスポート出力時のダイアログ例を示す。保存場所を指定しエクスポート方法を選択し、保存ボタンを押す。ここではエクスポート方法の選択はラジオボタン形式となっているが、プルダウンメニュー等、他形態でも構わない。続いて図16のステップ1603では保存形態が製本体裁を反映して出力かどうかのチェックを行う。エクスポート形態が原稿をそのまま保存の場合は、アプリケーションで設定した印刷設定を解除しステップ1605に進んでファイル変換処理を行う。製本体裁を反映して出力する場合（ここでは、図17のレイアウト体裁設定を反映して保存、及び印刷出力体裁を反映して保存）は、1604のステップでエクスポート用ドライバへのスプールを行う。1604での処理については図18に詳細フローを示す。エクスポートドライバにスプールされたファイルはまず、1701でレイアウト体裁設定を反映して保存、または印刷出力体裁を反映して保存のチェックを行う。レイアウト体裁設定を反映して保存とは、先に述べた面付けPDFでのエクスポート処理を指し、レイアウト指定やシフト指定の情報は反映されるが、両面／製本やステイブルにおけるシフトなどはコンフリクトされて反映されない。印刷出力体裁を反映して保存を選択した場合においては、1703でコンフリクトの解除を行う。続いて図16のステップ1605に進み、PDFファイルへの変換を行う。続いて、ステップ1606では、アプリケーションで設定した印刷設定情報ファイルの出力の選択を行う。ダイアログが表示され、出力選択を行うことで、印刷設定情報（パラメータ）ファイルを出力することができる。このファイルを出力することにより、エクスポートしたPDFファイルを別システムで印刷する場合に、印刷時にファイルをドライバに読み込ませることにより、両面／製本設定及び、ステイブル、パンチ等の印刷設定をドライバに再設定させることが出来る。これにより、印刷設定情報ファイルをドライバに取り込むことで、アプリケーションから印刷を行った場合の最終成果物と同じものが、印刷出力可能となる。続いてステップ1608に進み、PDFファイルの出力、保存が行われる。

【0079】

以上の手順に従えば、アプリケーションファイルから指定された属性で印刷された状態と同じ面付け状態のPDFデータの出力を行うことが可能となった。

【0080】

次にエクスポートファイルと共に出力できる印刷設定情報ファイルを利用した印刷処理について説明する。図19はエクスポートしたPDFファイルを紙印刷する場合に、ドライバに印刷設定情報ファイルをインポートして印刷設定を反映させて印刷処理をするフローである。ステップ1901ではエクスポートしたPDFファイルを別システムにて開

10

20

30

40

50

く。続いて1902では、プリンタの選択をする。選択したドライバにはエクスポート時に出力した印刷設定情報ファイルをインポートでき、ドライバより該当するファイルを取り込む。設定情報をインポートしない場合は通常どおりドライバに手動で設定した印刷設定により出力される。続いてステップ1904ではドライバにインポートされた印刷設定情報ファイルのドライバ情報が正しいかをチェックする。出力するドライバとファイル内のドライバ情報が異なる場合は印刷設定情報が反映されない。もしくは可能な機能のみにコンフリクトされて丸められることになる。続いてステップ1905では出力先のデバイス情報を取得する。続いてステップ1906ではドライバの印刷設定が反映される。この場合、ステップ1905でチェックしたデバイス情報能力が印刷設定情報ファイルの設定内容より劣る場合は、ドライバへの印刷設定反映でコンフリクトされる。反映された印刷設定で印刷実行することで、PDFにエクスポートしたファイルを印刷した成果物と、本発明の実施形態におけるアプリケーションから印刷を行った場合の最終成果物と同じものが、印刷出力可能となる。

10

【0081】

また、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。

【0082】

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。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。

20

【0083】

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（登録商標）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。

【0084】

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。

30

【0085】

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。

【図面の簡単な説明】

40

【0086】

【図1】本発明の実施の形態に係るスタンドアロン型の文書処理システムの構成を説明するためのブロック図である。

【図2】本実施の形態に係る文書処理システムを実現するホストコンピュータの構成を示すブロック図である。

【図3】本実施の形態に係るブックファイルの構造の一例を示す図で、図3（A）はブックファイルの形式の一例を模式的に示す図、図3（B）は原稿データの構成を示す図である。

【図4】本実施の形態に係るブック属性の一例を示す図である。

【図5】本実施の形態に係る章属性の一例を示す図である。

50

【図 6】本実施の形態に係るページ属性の一例を示す図である。

【図 7】本実施の形態に係る文書処理システムにおけるブックファイルを開く手順を説明するフローチャートである。

【図 8】本実施の形態に係る文書処理システムにおける、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を説明するフローチャートである。

【図 9】本実施の形態に係る文書処理システムによる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を示すフローチャートである。

【図 10】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 11】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。 10

【図 12】本実施の形態に係る文書処理システムに係る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。

【図 13】ステイブル制御システムの構成を示すブロック図である。

【図 14】原稿ページのプレビューモードで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 15】印刷プレビューモードで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。

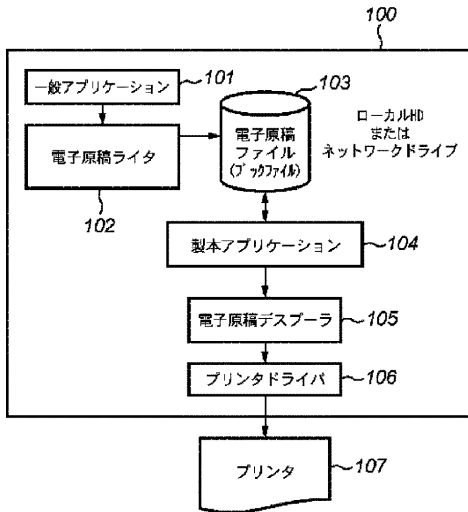
【図 16】本発明の実施の形態に係る文書処理システムにおける、エクスポート出力処理を行うフロー図である。 20

【図 17】本発明の実施の形態に係る文書処理システムにおける、エクスポート出力時のダイアログ例を示す図である。

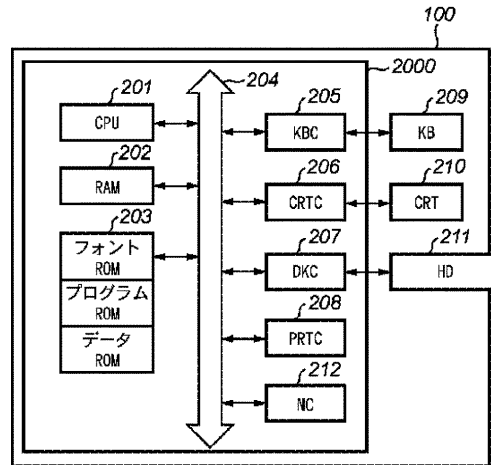
【図 18】本発明の実施の形態における、エクスポートドライバの処理を示すフロー図である。

【図 19】本発明の実施の形態における、エクスポートファイルを別システムで印刷出力する場合にアプリケーションで設定した印刷設定情報のパラメータファイルをインポートして印刷する処理を示すフロー図である。

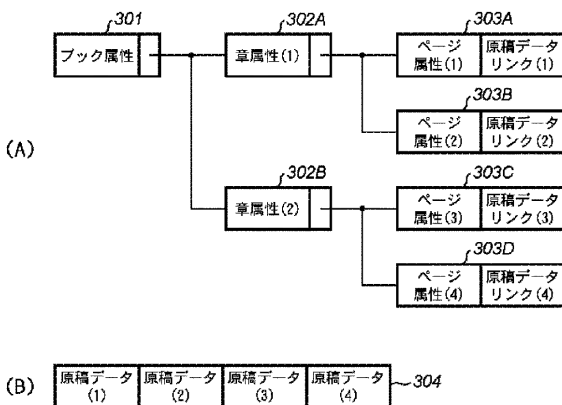
【 図 1 】



【 図 2 】



【 図 3 】



【 図 4 】

NO	属性候補	片側/両側/両面印刷	備考
1	印刷方法	片側/両側/両面印刷	
2	用紙サイズ	原稿サイズ/固定サイズ	・「A4/A3」、「B4/B3」、「レター/レジャー(11×17)」指定の場合は2通り指定 ・製本印刷またはN-up印刷が指定された場合は1面/1ページ目の原稿サイズが自動で選択される
3	用紙方向	縦/横	・固定サイズの用紙のみ選択可能
4	ふじ代/とじ方向	縦/横	・ソフト/硬紙指定が可能
5	N-up印刷	ページ数/配置数/境界線/配置位置等	・配置位置は3パターン ・等倍印刷指定可能
6	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
7	ウォーターマーク		・物理ページ単位、物理ページ単位に個別の指定が可能 ・全書/全ページが対象
8	ヘッダ・フッタ		・物理ページ単位、物理ページ単位に個別の指定が可能 ・全書/全ページが対象
9	排紙方法	ステイブル/パンチ穴	・ステイブル/パンチ穴は片面/両面印刷のみ ・ステイブルは1箇所/箇所
10	製本詳細	向き方向/中とし/紙幅指定/とじ代/分断指定等	・製本印刷時のみ
11	表紙/裏表紙		・表紙1枚、裏表紙1枚に対する印刷指定 ・総紙口(インサートを含む)指定
12	インデックス紙		・インデックス紙部分への文字列印刷、インデックス紙上へのフッター/フッタを指定可能 ・製本印刷は指定不可
13	合紙		・総紙口(インサートを含む)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、合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」

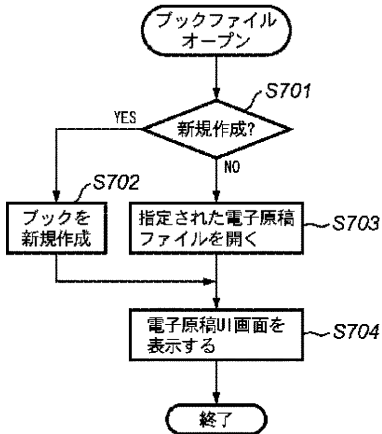
【 図 5 】

NO	属性候補	片側/両側/両面印刷	備考
1	用紙サイズ	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 ・ブロックで指定された場合は「用紙かえ」を自動指定 ・指定用紙でのみ変更可能、ブロックにあわせる指定でも用紙サイズの変更可能
2	用紙方向	縦/横	・固定サイズの用紙のみ選択可能
3	N-up印刷指定	ページ数/配置数/境界線/配置位置等	・配置位置は3パターン ・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	・ブロック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・フッタ	表示/非表示	・ブロック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	・インデックス紙、合紙が指定された場合に、OFFが可能 ・デフォルトはON

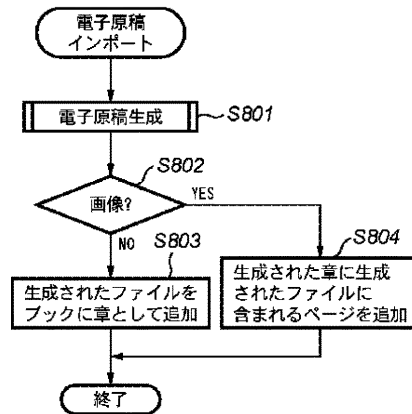
【図 6】

NO	属性情報	備考
1	ページ回転指定	・ 0/90/180/270度の指定が可能
2	ウォーターマーク	・ ブックで指定された全ウォーターマークを表示するか しないか指定
3	ヘッダー・フッター	・ ブックで指定された全ヘッダー・フッターを表示するか しないか指定
4	ズーム	・ 仮複論理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パターンおよび任意位置指定
6	アノテーション	
7	Variable項目	
8	ページ分割	

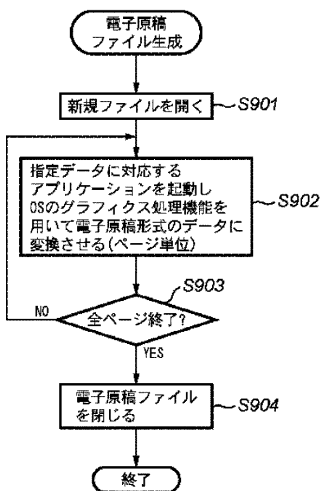
【図 7】



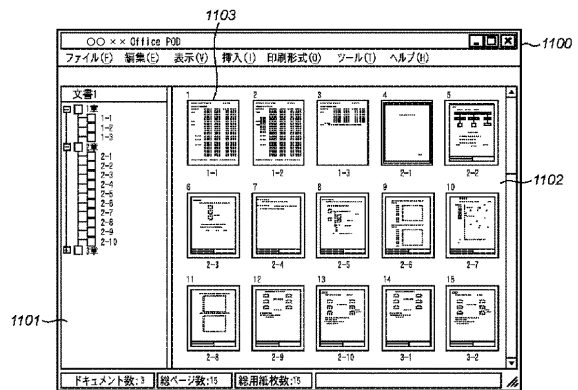
【図 8】



【図 9】



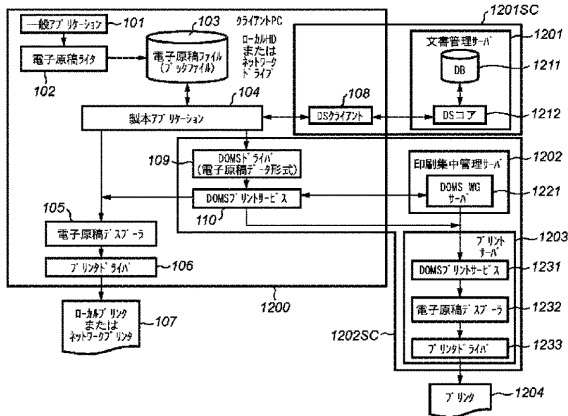
【図 10】



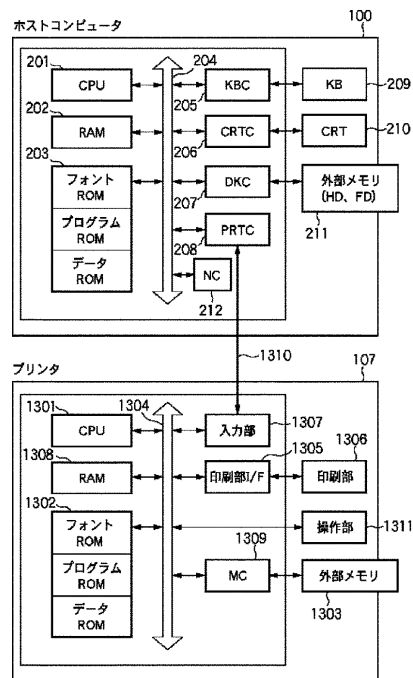
【図 11】



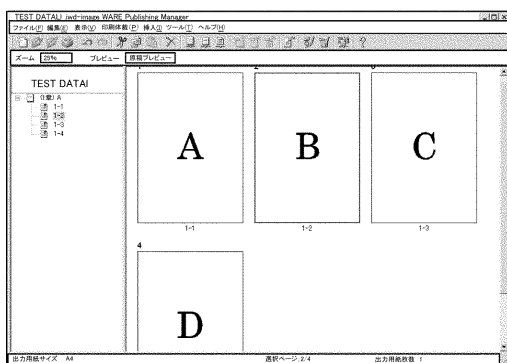
【図 12】



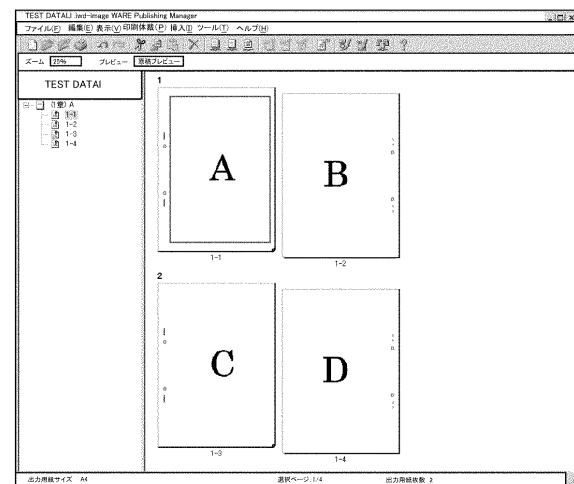
【図 13】



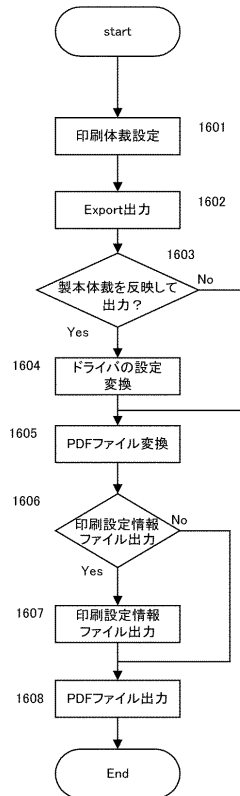
【図 14】



【図 15】



【図 16】



【図 17】

保存場所:

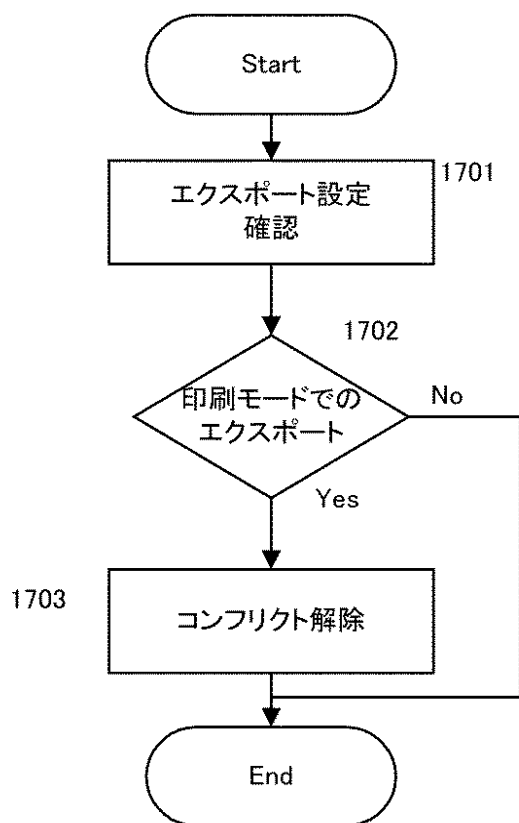
ファイル名:

ファイルの種類:

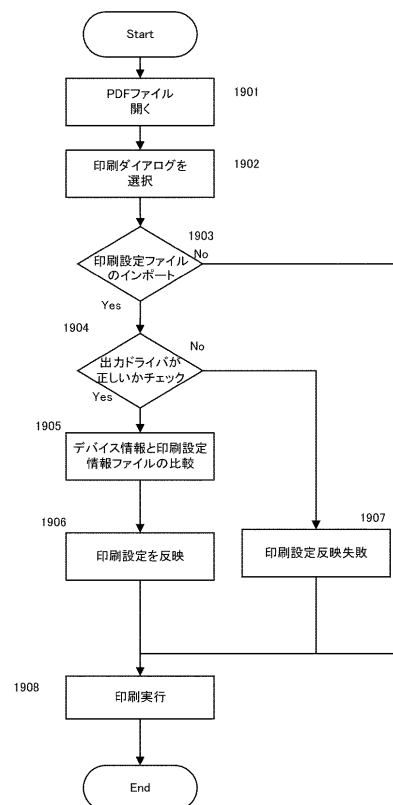
エクスポート方法:

- ☐ 原稿をそのまま保存
- ☐ レイアウト体裁の設定を反映して保存
- ☐ 印刷出力体裁を反映して保存

【図 18】



【図 19】



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 富田 信

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 佐藤 純子

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 AD03 AD04 AE07 AE13 AF01 BF09 BG03 CD17 DB04 DB11

FA01

5B021 AA01 BB01 CC05 LE00