

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4799253号  
(P4799253)

(45) 発行日 平成23年10月26日 (2011.10.26)

(24) 登録日 平成23年8月12日 (2011.8.12)

(51) Int. Cl.

F I

G 0 6 F 3/12 (2006.01)  
B 4 1 J 29/40 (2006.01)G 0 6 F 3/12 F  
B 4 1 J 29/40 Z

請求項の数 7 (全 31 頁)

(21) 出願番号	特願2006-106624 (P2006-106624)	(73) 特許権者	000001007
(22) 出願日	平成18年4月7日 (2006.4.7)		キヤノン株式会社
(65) 公開番号	特開2007-280124 (P2007-280124A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成19年10月25日 (2007.10.25)	(74) 代理人	100076428
審査請求日	平成21年4月6日 (2009.4.6)		弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(72) 発明者	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

最終頁に続く

(54) 【発明の名称】 情報処理装置及びその処理方法

(57) 【特許請求の範囲】

【請求項 1】

ページに対して可変付加情報を付加する情報処理装置の処理方法であって、  
前記情報処理装置の第1選択手段が、1レコードあたり複数の項目で構成された複数レコードのデータベースファイルを選択する第1選択工程と、  
前記情報処理装置の第1表示制御手段が、前記第1選択工程において選択されたデータベースファイルの項目名を表示する第1表示制御工程と、  
前記情報処理装置の第2選択手段が、前記第1表示制御工程において表示された項目名から前記可変付加情報として使用される項目を選択する第2選択工程と、  
前記情報処理装置の第2表示制御手段が、印刷されるページ上に前記第2選択工程において選択された前記データベースファイルの項目の第1レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示する第2表示制御工程とを有し、  
前記第1レコードが第2レコードに変更された場合に、前記第2表示制御工程において前記印刷されるページと同一ページ上に前記第2選択工程において選択された前記データベースファイルの項目の前記第2レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示することを特徴とする情報処理装置の処理方法。

【請求項 2】

前記情報処理装置の登録手段が、前記レコードの変更により表示される内容が切り替わる可変付加情報と、前記可変付加情報とは異なる固定付加情報との両方を登録する登録工程を更に有することを特徴とする請求項1記載の情報処理装置の処理方法。

10

20

## 【請求項 3】

前記情報処理装置の第 3 表示制御手段が、前記固定付加情報と前記可変付加情報とを識別可能に表示する第 3 表示制御工程を更に有することを特徴とする請求項 2 記載の情報処理装置の処理方法。

## 【請求項 4】

前記情報処理装置の位置変更手段が、前記可変付加情報と前記ページの文字列とが重なるように、前記可変付加情報の位置を変更する位置変更工程を更に有することを特徴とする請求項 1 乃至 3 の何れか一項記載の情報処理装置の処理方法。

## 【請求項 5】

前記位置変更工程は、前記可変付加情報と前記ページの文字列とが重なるように移動する方向として、最も移動量の少ない方向に前記可変付加情報の位置を変更することを特徴とする請求項 4 に記載の情報処理装置の処理方法。

10

## 【請求項 6】

ページに対して可変付加情報を付加する情報処理装置であって、

1 レコードあたり複数の項目で構成された複数レコードのデータベースファイルを選択する第 1 選択手段と、

前記第 1 選択手段によって選択されたデータベースファイルの項目名を表示する第 1 表示制御手段と、

前記第 1 表示制御手段によって表示された項目名から前記可変付加情報として使用される項目を選択する第 2 選択手段と、

20

印刷されるページ上に前記第 2 選択手段によって選択された前記データベースファイルの項目の第 1 レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示する第 2 表示制御手段とを有し、

前記第 1 レコードが第 2 レコードに変更された場合に、前記第 2 表示制御手段によって前記印刷されるページと同一ページ上に前記第 2 選択手段によって選択された前記データベースファイルの項目の前記第 2 レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示することを特徴とする情報処理装置。

## 【請求項 7】

請求項 1 乃至 5 の何れか 1 項に記載の情報処理装置の処理方法の各工程をコンピュータに実行させるためのプログラム。

30

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、スタンプ機能によりスタンプを登録及び編集する情報処理装置及びその処理方法に関するものである。

## 【背景技術】

## 【0002】

文書データに対して、スタンプ等の付加情報を付加して印刷させる仕組みとして、3 つの手法が考案されている。

## 【0003】

40

第 1 の手法は、プリンタドライバにおける一連の印刷データ生成処理内で、文書データにスタンプを付加する手法である。ま情報処理装置上でプリンタに送信する印刷データを生成する前に、最終的にプリンタに送付する印刷データとは異なる形式のデータ形式（いわゆる中間コード）で一時保存しておく。そして、中間コード形式で一時保存したデータから改めて最終的にプリンタに送付する印刷データを生成する。

## 【0004】

このとき、上述の印刷システムにおいて、中間コード形式のデータに、原稿ページ又は印刷ページに「Confidential」や「コピー禁止」などの文字を自由なサイズ、フォント、角度で文書全体や一部のページに追加する機能が存在する。以下、この機能を「スタンプ機能」と称す。例えば、特許文献 1 参照。

50

## 【 0 0 0 5 】

また、第2の手法は、任意のアプリケーションで生成された文書をフォーマット変換して、新たな文書として取り込み、取り込んだ文書をプレビュー画面を介して表示させる統合アプリケーションによりスタンプを付加する手法である。この統合アプリケーションでは、文書を、ブック、章、ページと階層構造で管理するために、各階層において、スタンプを付加することができ、また、文書のプレビューを確認しながら操作できるため、操作性がよいという効果がある。例えば、特許文献2参照。

## 【 0 0 0 6 】

更に、第3の手法は、アプリケーション印刷システムにおいて、可変データ（バリエブル）の差替印刷、いわゆるバリエブル印刷を利用することで、文書データにバリエブル情報

10

【特許文献1】特開2000-025276号公報

【特許文献2】特開2003-091526号公報

【特許文献3】特開2003-296070号公報

【発明の開示】

【発明が解決しようとする課題】

## 【 0 0 0 7 】

しかしながら、上述の第1の手法では、プリンタドライバのユーザインタフェースを介して予め用意した一部の文字列や、事前に登録した文字列しか使えなかった。そのため、人の名前等、自由な文字列を複数使用したい場合は、その都度、事前に文字列を登録する

20

必要があり、手間がかかるという欠点があった。

## 【 0 0 0 8 】

また、登録されたスタンプを付加して印刷を行う場合、スタンプを選択して印刷という操作となり、複数のスタンプ印刷を行うには、この操作を複数回行う必要があり、非常に

手間と時間がかかるという問題があった。

## 【 0 0 0 9 】

また、上述の第2の手法でも、プレビューを見ながらスタンプの付加状況を確認できるものの、統合アプリケーションのユーザインタフェースを介して予め用意した一部の文字列や、事前に登録した文字列しか使えなかった。そのため、人の名前等、自由な文字列を

30

複数使用したい場合は、その都度、事前に文字列を登録する必要があり、手間がかかるという欠点があった。

## 【 0 0 1 0 】

また、登録されたスタンプを付加して印刷を行う場合、スタンプを選択して印刷という操作となり、複数のスタンプ印刷を行うには、この操作を複数回行う必要があり、非常に

手間と時間がかかるという問題があった。

## 【 0 0 1 1 】

また、上述の第3の手法のように、バリエブル機能で同様のことを行う場合、テキストフィールドの設定として、文字列を置きたい場所、角度、大きさで全ページ定義する必要

40

があった。また、修正も全ページに対して行う必要があり、操作性において、問題があった。

## 【 0 0 1 2 】

本発明は、レコードの変更指示に従って付加情報の内容を切り替えることを目的とする。

【課題を解決するための手段】

## 【 0 0 1 3 】

本発明は、ページに対して可変付加情報を付加する情報処理装置の処理方法であって、前記情報処理装置の第1選択手段が、1レコードあたり複数の項目で構成された複数レコードのデータベースファイルを選択する第1選択工程と、前記情報処理装置の第1表示制御手段が、前記第1選択工程において選択されたデータベースファイルの項目名を表示する第1表示制御工程と、前記情報処理装置の第2選択手段が、前記第1表示制御工程にお

50

いて表示された項目名から前記可変付加情報として使用される項目を選択する第2選択工程と、前記情報処理装置の第2表示制御手段が、印刷されるページ上に前記第2選択工程において選択された前記データベースファイルの項目の第1レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示する第2表示制御工程とを有し、前記第1レコードが第2レコードに変更された場合に、前記第2表示制御工程において前記印刷されるページと同一ページ上に前記第2選択工程において選択された前記データベースファイルの項目の前記第2レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示することを特徴とする。

【0014】

また、本発明は、ページに対して可変付加情報を付加する情報処理装置であって、1レコードあたり複数の項目で構成された複数レコードのデータベースファイルを選択する第1選択手段と、前記第1選択手段によって選択されたデータベースファイルの項目名を表示する第1表示制御手段と、前記第1表示制御手段によって表示された項目名から前記可変付加情報として使用される項目を選択する第2選択手段と、印刷されるページ上に前記第2選択手段によって選択された前記データベースファイルの項目の第1レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示する第2表示制御手段とを有し、前記第1レコードが第2レコードに変更された場合に、前記第2表示制御手段によって前記印刷されるページと同一ページ上に前記第2選択手段によって選択された前記データベースファイルの項目の前記第2レコードのデータが前記可変付加情報として配置されたプレビュー画像を表示することを特徴とする。

【発明の効果】

【0015】

本発明によれば、レコードの変更指示に従って付加情報の内容を切り替えることにより、可変情報を文書に付加した印刷物を容易に得ることができる。

【発明を実施するための最良の形態】

【0016】

以下、図面を参照しながら発明を実施するための最良の形態について詳細に説明する。

【0017】

<システム概要>

まず、本発明の情報処理装置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む。そして、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。

【0018】

<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（ホストコンピュータとも呼ばれる）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、フォトタッチ、ドロー、或いはペイント、プレゼンテーション、テキスト編集等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等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う。そして、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ィ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。

#### 【0019】

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、例えば米国アドビシステムズ社によるPDF形式やW3Cで公開されているSVG形式などが電子原稿形式として採用できる。ここで、PDFはPortable Document Formatの略であり、SVGはScalable Vector Graphicsの略である。

#### 【0020】

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(又は文書データ)と呼ぶ。

#### 【0021】

このようにデバイスドライバとして電子原稿ライタ102を指定し、一般アプリケーション101によりデータを印刷させる。これにより、アプリケーションデータはアプリケーション101によって定義されたページ(以後論理ページ或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103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。

#### 【0022】

製本アプリケーション104は、電子原稿ファイル(或いはブックファイル)103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104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。

#### 【0023】

製本アプリケーション104によって編集されたブックファイル103を印刷する際には、製本アプリケーション104によって電子原稿デスプーラ105が起動される。電子原稿デスプーラ105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る。そして、製本アプリケーションで利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105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107用のプリンタドライバ106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107用のプリンタドライバ106に出力し、そのプリンタドライバ106がプリンタ107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106から不図示のシステムスプーラを介してプリンタ107に送信され、プリンタ107によってコマンドに応じた画像が印刷される。

#### 【0024】

図2は、コンピュータ100のハードウェアブロック図である。図2において、CPU201は、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。ここで、プログラムは、ROM203のプログラム用ROMに記憶された、或いはハードディスク211からRAM202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である。RAM202は、CPU201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)205は、キーボード209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)206は、CRTディスプレイ210の表示を制御する。ディスクコントローラ(DKC)207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)211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。PRTC208は、接続されたプリンタ107との間の信号の交換を制御する。NC212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。

#### 【0025】

##### <電子原稿データの形式>

編集アプリケーション104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。

#### 【0026】

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。

#### 【0027】

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。

#### 【0028】

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、303

B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。

#### 【0029】

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。

10

#### 【0030】

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。

#### 【0031】

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、とじ代/とじ方向、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。

20

#### 【0032】

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の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、或いは、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

#### 【0033】

合紙属性は、章の区切りとして、インサータから、或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。

#### 【0034】

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。

#### 【0035】

40

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-up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-up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。

#### 【0036】

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、

50

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。

#### 【0037】

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。また、ここでは、このウォーターマーク属性情報は、固定スタンプ情報だけでなく、後述する可変スタンプ情報を保持することができる。

10

#### 【0038】

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。

20

#### 【0039】

<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。

#### 【0040】

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、製本アプリケーション104は、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。

30

#### 【0041】

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。

40

#### 【0042】

50



さて、開かれたブックファイルには、電子原稿ライタ 1 0 2 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が与えられる。この機能は、図 1 0 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。

#### 【 0 0 4 3 】

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 1 0 2 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 S 8 0 1 ）。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 S 8 0 2 ）。この判定は、ウインドウズ O S 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「 b m p 」であればウインドウズビットマップデータであり、「 j p g 」であれば j p e g 圧縮された画像データ、「 t i f f 」であれば t i f f 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 S 8 0 1 0 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、 S 8 0 1 0 の処理を省略することも可能である。

#### 【 0 0 4 4 】

ここで、画像データでなかった場合には、製本アプリケーション 1 0 4 は、ステップ S 8 0 1 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 S 8 0 3 ）。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。

#### 【 0 0 4 5 】

また、ステップ S 8 0 2 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 S 8 0 1 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 S 8 0 4 ）。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、 N u p 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、或いは、新規な章が追加される。

#### 【 0 0 4 6 】

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。

#### 【 0 0 4 7 】

##### < 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。

10

20

30

40

50

- ( 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
- ( 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

**【 0 0 4 8 】**

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、或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。

20

**【 0 0 4 9 】****< 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、例えばウインドウズの 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。

30

**【 0 0 5 0 】**

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。

40

**【 0 0 5 1 】**

電子原稿デスプーラ 1 0 5 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 2 】****< 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。

#### 【 0 0 5 3 】

図 1 2 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 D O M S ドライバ 1 0 9 及び D O M S プリントサービスモジュール 1 1 0、D S (文書サービス) クライアントモジュール 1 0 8 を加えた構成を有する。上記の D O M S は、Document Output Management Service の略である。このクライアント文書処理システム 1 2 0 0 に文書管理サーバ 1 2 0 1、印刷集中管理サーバ 1 2 0 2 及びプリントサーバ 1 2 0 3 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。

10

#### 【 0 0 5 4 】

尚、図 1 2 に示す例では、文書管理サーバ 1 2 0 1 と印刷集中管理サーバ 1 2 0 2 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 1 0 8 を含む文書管理サーバクライアントシステム 1 2 0 1 S C が、スタンドアロン型文書管理システムに追加される。また、印刷集中管理サーバ 1 2 0 2 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 1 2 0 2 S C が、スタンドアロン型文書管理システムに追加される。

20

#### 【 0 0 5 5 】

文書管理サーバ 1 2 0 1 は、製本アプリケーション 1 0 4 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 1 2 0 1 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 P C のローカル H D に代わって、或いはそれに加えて、文書管理サーバ 1 2 0 1 のデータベース 1 2 1 1 に保存される。製本アプリケーション 1 0 4 と文書管理サーバ 1 2 0 1 との間のブックファイルの保存及び読み出しは、D S クライアント 1 0 8 及び D S コア 1 2 1 2 を介して行われる。

#### 【 0 0 5 6 】

印刷集中管理サーバ 1 2 0 2 は、クライアント文書管理システム 1 2 0 0 に格納された、或いは文書管理サーバ 1 2 0 1 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、D O M S ドライバ 1 0 9 および D O M S プリントサービスモジュール 1 1 0 を介して印刷集中管理サーバ 1 2 0 2 の D O M S W G サーバモジュール 1 2 2 1 に送信される。集中印刷管理サーバ 1 2 0 2 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 D O M S プリントサービスモジュール 1 1 0 を介して電子原稿デスプーラ 1 0 5 に電子原稿データを渡す。プリントサーバ 1 2 0 3 により印刷する場合には、プリントサーバ 1 2 0 3 の D O M S プリントサービスモジュール 1 2 0 3 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。

30

40

#### 【 0 0 5 7 】

##### < 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 1 0 に示したユーザインタフェース画面 1 1 0 0 が表示される。ツリー部 1 1 0 1 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 1 1 0 2 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

50

。この印刷ビューモードでは、プレビュー部 1 1 0 2 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。

【 0 0 5 8 】

< 印刷システム >

次に、両面印刷機能を有するプリンタと接続するホストコンピュータ 1 0 0 の製本アプリケーション 1 0 4 によって作成・編集されたブックファイル 1 0 3 をプリンタ 1 0 7 で印刷する際の制御について説明する。具体的には、製本印刷・分冊等の機能が指定された印刷設定を読み取り、プリンタ 1 0 7 の能力に応じて印刷設定を複数のジョブで実現するか 1 つのジョブで実現するかを制御する方法について説明する。

10

【 0 0 5 9 】

図 1 3 は、本実施形態における印刷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、印刷システムは、図 2 に示したホストコンピュータ 1 0 0 と、両面印刷機能を有するプリンタ 1 0 7 とで構成される。

【 0 0 6 0 】

以下、プリンタ 1 0 7 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。

20

【 0 0 6 1 】

図示するように、プリンタ 1 0 7 において、1 3 0 1 はプリンタ CPU であり、制御プログラム等に基づいてシステムバス 1 3 0 4 に接続される印刷部 I / F 1 3 0 5 を介して、印刷部（プリンタエンジン）1 3 0 6 に出力情報としての画像信号を出力する。この制御プログラムは、ROM 1 3 0 2 内のプログラム用 ROM や外部メモリ 1 3 0 3 に記憶されている。また、この ROM 1 3 0 2 内のプログラム ROM には、CPU 1 3 0 1 の制御プログラム等が記憶され、ROM 1 3 0 2 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ている。また、ROM 1 3 0 2 内のデータ用 ROM には、ハードディスク等の外部メモリ 1 3 0 3 がないプリンタの場合、ホストコンピュータ 1 0 0 上で利用される情報等が記憶されている。

30

【 0 0 6 2 】

また、CPU 1 3 0 1 は、入力部 1 3 0 7 を介してホストコンピュータ 1 0 0 との通信処理が可能となっており、プリンタ 1 0 7 内の情報等をホストコンピュータ 1 0 0 に通知できる。RAM 1 3 0 8 は、CPU 1 3 0 1 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 によりメモリ容量を拡張することができるように構成されている。

【 0 0 6 3 】

尚、RAM 1 3 0 8 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 1 3 0 3 は、メモリコントローラ（MC）1 3 0 9 によりアクセスを制御される。外部メモリ 1 3 0 3 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 1 3 1 1 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。

40

【 0 0 6 4 】

また、上述の外部メモリ 1 3 0 3 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 1 3 1 1 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。

【 0 0 6 5 】

50

次に、図 1 4 ~ 図 3 3 を用いて、本実施形態におけるスタンプ及びバリアブルスタンプ機能について説明する。

#### 【 0 0 6 6 】

まず、図 1 4 及び図 1 5 を用いて、一般的なスタンプ機能について説明する。尚、この図 1 4 及び図 1 5 で説明するスタンプ機能は、固定文字列や固定画像の設定及び新規追加であり、このようなスタンプ情報を、可変スタンプ情報と区別して固定スタンプ情報と呼ぶこともある。図 1 0 に示す製本アプリケーションの UI 画面からスタンプ機能を使用したい場合、メニューの印刷形式を選択することにより、文書の印刷設定を行うための図 1 4 に示すスタンプ付加用の UI 画面を開き、付加したいスタンプを選択する。そして、原稿ページにスタンプをつけるか、印刷ページにスタンプを付けるかを選択する。この選択は図 1 4 に示す 1 4 0 1、1 4 0 2 にチェックを入れることで行う。ここで使用できるスタンプは、初めから用意してある文字列か、もしくは事前に登録したものである。

10

#### 【 0 0 6 7 】

尚、ユーザが所望の文字列をスタンプとして登録するには、図 1 4 に示すスタンプ編集 1 4 0 3 を選択し、図 1 5 に例示するスタンプ編集 UI 画面を表示させる。この UI 画面より既存のスタンプのフォント種やサイズ、色等を変更することが可能である。そして、1 5 0 1 に示すように、スタンプの配置を設定すると、原稿にかかわらず同じ位置に配置される。

#### 【 0 0 6 8 】

また、「新規追加」を選択し、新規追加 UI 1 5 1 0 を表示させ、ここから新規追加を行うことが可能である。ここで追加したスタンプは以降、図 1 4 に示すスタンプ付加 UI 画面にリストアップされ、使用可能となる。

20

#### 【 0 0 6 9 】

このようにスタンプ登録を行い、それを指定し、印刷した結果を例示した図が図 1 6 であり、スタンプとして「PM 花子」を選択して印刷した例である。また、同様の作業を繰り返し行うことで、図 1 7 に示すように複数の任意の文字列によるスタンプ付加機能を使用した印刷結果を得ることができる。尚、異なる印刷結果を得るためには、事前に登録したものを選択し直し、複数回印刷を実行する必要がある。

#### 【 0 0 7 0 】

次に、図 1 8 を用いて、バリアブルスタンプ（可変スタンプ情報とも呼ぶ）を登録又は削除する処理について説明する。図 1 8 は、本実施形態におけるバリアブルスタンプ登録削除処理を示すフローチャートである。ユーザがバリアブルスタンプの登録を行う場合、図 1 0 に示す製本アプリケーションの UI 画面からメニューの「ツール」を開き「バリアブルエディタ」の項目を選択する。そして、製本アプリケーションは、この選択指示を受け付けると、ステップ S 1 8 0 1 において、製本アプリケーション 1 0 4 は、内部モジュールであるバリアブルエディタの起動処理を行う。このバリアブルエディタとは、バリアブル印刷を所望する時に、バリアブルフィールドの定義を行うためのエディタである。この処理によって起動されたバリアブルエディタを例示した図が図 1 9 である。

30

#### 【 0 0 7 1 】

次に、ステップ S 1 8 0 2 において、バリアブルエディタは、ユーザ操作により指定されたデータベースファイルを選択し、このデータベースファイルに対して接続処理を行う。バリアブル印刷を行う場合、通常データベースに接続を行い、データベースからデータを取得し、元データに付加して印刷を行う。この場合も、データベースへの接続が必要であり、図 1 9 に示すメニューの「データベース」コントロールから、データベースとなるファイルが選択される。ここでは、接続先データベースの形態、種類は問わないが、例えば、米国マクロソフト社の Excel（エクセル）ファイルがデータベースファイルとして利用可能である。つまり、本実施例でいう可変スタンプ情報のためのデータベースファイルとは、複数レコード分の可変データを定義している可変情報ファイルを表す。

40

#### 【 0 0 7 2 】

次に、ステップ S 1 8 0 3 において、バリアブルエディタは、レコード設定処理を行う

50

。これは接続したデータベースファイル中のレコードについて、レコードの並べ替えを行う等の詳細設定を行う処理である。この処理は、ユーザの選択によって行われる。具体的には、ステップS 1 8 0 2で選択されたデータベースファイルのデータが解析され、エクセルファイルの場合は、1行目の各列が代表項目となり、2行目から1レコードとして扱われる。そして、列が項目の値として表示され、この関連付けでよいかをユーザに確認させることができる。

【0073】

上述した一連の処理により、バリアブルスタンプの登録のための事前準備が行われたことになる。

【0074】

次に、ステップS 1 8 0 4において、バリアブルエディタは、スタンプ登録が選択されたか否かの判定を行う。これはユーザがスタンプの登録をすべく、図19に示すように「ツール」-「バリアブルスタンプの登録」が選択されたか否かを判定する処理である。ここでは、ユーザは先に接続を行ったデータベースを元にバリアブルスタンプの登録を所望し、その選択がされたとする。また、選択がされなければ、ステップS 1 8 1 4において、その他の処理実行し、この処理を終了する。

【0075】

ここで登録が選択されると、ステップS 1 8 0 5において、バリアブルエディタは、既に登録されたバリアブルスタンプがあるか否かの判定を行う。これは、以降説明するシーケンスにより、既に登録されたバリアブルスタンプが存在するか否かを判定する処理である。ここでは、既に登録されたバリアブルスタンプがあるとする。また、登録場所として、ホスト（パソコン等）に固有のものとしてホスト上に登録と、データに固有のものとして電子原稿ファイル中に登録の2種類が考えられる。尚、登録においては、その形態は問わないものとする。ホスト上に登録する場合は、電子原稿ファイルは、ホスト上のデータベースファイルが格納されている位置（パス）を情報として保持し、データベースのレコード設定が保持される。また、電子原稿ファイル中に登録する場合は、接続されたデータベースファイルを電子原稿ファイル中に取り込み、データベースのレコード設定も保持されることになる。電子原稿ファイル中に取り込むことで、電子原稿ファイル自体を他のPCに移動させた場合にも、バリアブルスタンプ情報を利用することが可能となる。また、ホスト上に登録する場合には、そのホストPCで利用することが可能であり、かつ、電子原稿ファイル自体のデータサイズを少なくすることができる。但し、ホスト上に登録する場合には、他のPCに電子原稿ファイルを移動させた場合や、接続先のデータベースファイルが削除された場合に、バリアブルスタンプ情報を利用することができなくなる。

【0076】

上述の判定結果に従い、ステップS 1 8 0 6において、バリアブルエディタは、既存のバリアブルスタンプ取得処理を行う。これは既に登録してあるバリアブルスタンプを取得する処理である。ここでは、3つのバリアブルスタンプが既に登録されているものとする。そして、「バリアブルスタンプ\_Text」、「バリアブルスタンプ\_Image」、「バリアブルスタンプ」が取得されたとする。

【0077】

次に、ステップS 1 8 0 7において、バリアブルエディタは、バリアブルスタンプ一覧表示処理を行う。これは先に取得した既存のバリアブルスタンプを一覧表示する処理である。上述のように、この例では3つのバリアブルスタンプが取得され、これらが一覧表示される。これを例示した図が図20である。

【0078】

次に、ステップS 1 8 0 8において、図20に示す一覧表示画面から、ユーザが新しいバリアブルスタンプを所望し、「新規追加」2001が選択されたか否かの判定を行う。ここでは、新しいバリアブルスタンプを追加するものとし、「新規追加」2001が選択されたものとする。尚、バリアブルスタンプは複数個、登録可能であり、登録されると、一覧表示に追加表示される。

10

20

30

40

50

## 【 0 0 7 9 】

ここで登録する場合は、ステップ S 1 8 0 9 において、バリアブルエディタは、バリアブルスタンプの追加設定処理を行う。これはバリアブルスタンプを追加する場合に、ユーザに詳細設定を促すための設定ダイアログを表示する処理である。この UI 例は図 2 1 に示す。それぞれのコントロールの詳細は図 2 2 に示す。ここでは、それぞれのコントロールで、図 2 1 で示されるような選択、設定がされているものとする。尚、図 2 1 の設定ダイアログでは、データベースファイル中のどのデータベースフィールド、つまり項目をバリアブルスタンプ情報として登録するかを ( 3 ) のプルダウンメニューを用いてユーザは選択することができる。( 3 ) のプルダウンメニューには、バリアブルエディタによりデータベースファイルから読み込まれた、1 レコード目の各項目、もしくは代表項目が表示されている。これにより、1 つのレコードに複数の項目を有するデータベースファイルであっても、その中からバリアブルスタンプ情報として登録する 1 つの可変情報を選択して設定することが可能となっている。

10

## 【 0 0 8 0 】

その後、ステップ S 1 8 1 3 において、バリアブルエディタは、追加終了か否かの判定を行う。ここで「OK」2 1 1 0 のコントロールが選択されたとすると、ステップ S 1 8 1 5 において、バリアブルエディタは、登録完了表示処理を行う。そして、バリアブルエディタは、登録内容を製本アプリケーション 1 0 4 に通知し、図 2 1 で設定されたバリアブルスタンプの登録情報を電子原稿ファイル内に格納する。これは、ステップ S 1 8 0 9 で設定された内容で登録が完了したことをユーザに知らせる処理である。これを例示した図が図 2 3 である。ここでは、スタンプの登録のみで、実際に描画を行う設定はクライアントで行うことが明示されている。そして「OK」2 3 0 1 のコントロールが押下されると、図 2 0 に示すバリアブルスタンプ一覧表示にリストアップされ、登録が確認できる。

20

## 【 0 0 8 1 】

また、ステップ S 1 8 0 8 において、新規追加が選択されなかった場合は、ステップ S 1 8 1 0 において、バリアブルエディタは、削除が選択されたか否かの判定を行う。これは図 2 0 に示す一覧表示に、既に登録されているバリアブルスタンプから削除を行うコントロールである「削除」2 0 0 2 が選択されたか否かを判定する処理である。ここで、「削除」2 0 0 2 のコントロールが選択されると、ステップ S 1 8 1 1 へ進み、一覧表示にリストアップされているバリアブルスタンプを削除する。

30

## 【 0 0 8 2 】

また、ステップ S 1 8 1 0 において、「削除」2 0 0 2 が選択されていなければ、ステップ S 1 8 1 2 へ進み、バリアブルエディタは、「閉じる」2 0 0 3 が押下されたか否かの判定を行う。ここで、「閉じる」2 0 0 3 が押下されなければ、ステップ S 1 8 0 5 に戻り、上述の処理を繰り返すが、押下されると、この処理を終了する。

## 【 0 0 8 3 】

上述した一連の処理により、バリアブルスタンプの登録又は削除が行われる。

## 【 0 0 8 4 】

次に、図 2 4 を用いて、バリアブルスタンプを文書 ( 電子原稿ファイルのページデータ ) に付加又は編集する処理について説明する。図 2 4 は、本実施形態におけるバリアブルスタンプ付加編集処理を示すフローチャートである。まず、ステップ S 2 4 0 1 において、製本アプリケーション 1 0 4 は、スタンプの表示が選択されたか否かの判定を行う。これは図 1 0 に示す製本アプリケーションの UI 画面よりスタンプ付加やスタンプ編集のため、図 1 4 で例示されるスタンプ付加 UI の表示が選択されたか否かを判定する処理である。ここでは、選択がされたとし、スタンプのリストアップに先立ち、ステップ S 2 4 0 2 において、製本アプリケーション 1 0 4 は、バリアブルスタンプの取得処理を行う。これは既存の通常スタンプに加え、登録済のバリアブルスタンプを取得する処理である。ここでは、先の一連の処理により、バリアブルスタンプの登録が行われているものとし、取得されるものは「バリアブルスタンプ\_Text」、「バリアブルスタンプ\_Image」、「バリアブルスタンプ」であるとする。尚、図 2 1 で設定されたバリアブルスタンプの登録情報は

40

50

、電子原稿ファイル内に格納されている。

【 0 0 8 5 】

バリアブルスタンプを取得すると、ステップ S 2 4 0 3 において、製本アプリケーション 1 0 4 は、登録済スタンプ一覧表示処理を行う。これは先に取得したバリアブルスタンプを含む登録済スタンプを一覧表示する処理である。上述したように、ここでは既に 3 つのバリアブルスタンプが登録済として取得されているので、これら通常のスタンプと共に、登録したバリアブルスタンプが列挙される。これを例示した図が図 2 5 である。尚、本実施例では、バリアブルスタンプである可変スタンプ情報と、固定スタンプ情報とをユーザが識別可能なように、固定スタンプ情報の名称の近傍に、識別アイコンを表示している。もちろん、可変スタンプ情報の名称の近傍側に識別アイコンを表示させても良い。

10

【 0 0 8 6 】

次に、ステップ S 2 4 0 4 において、スタンプ編集が選択されたか否かの判定を行う。図 2 5 に示す UI より「スタンプ編集」が選択されたか否かを判定する処理である。ここでは、ユーザがバリアブルスタンプの編集をするものとし、「スタンプ編集」が選択されたものとする。

【 0 0 8 7 】

ここで選択されると、ステップ S 2 4 0 7 において、スタンプ編集ダイアログ表示処理によりスタンプ編集のためのダイアログを表示する。これを例示した図が図 2 6 である。図 2 6 に示す UI 上のスタンプ一覧 2 6 0 1 から編集したいバリアブルスタンプを選択し、属性、配置、スタンプの印刷方法のタブ 2 6 0 2 により詳細な設定を行うことが可能である。尚、前述した図 2 0 及び図 2 1 では、バリアブルスタンプのデータ内容についての登録であるため、この登録処理だけでは、図 2 6 及び図 2 7 で設定される、大きさ、位置、印刷の種類について、デフォルトの値しか備えていない。

20

【 0 0 8 8 】

図 2 7 は、スタンプ編集ダイアログ内で、スタンプの属性、配置、印刷方法を設定する例を示す図である。図 2 6 及び図 2 7 に示す ( A ) のように、配置及び印刷方法については、バリアブルスタンプについても通常のスタンプと同様な設定が可能である。また、選択されているスタンプの種類がテキスト ( 図 2 7 に示す ( B ) 参照 ) かイメージ ( 同 ( C ) 参照 ) かによって属性タブの詳細設定が異なる。

【 0 0 8 9 】

可変スタンプ情報及び固定スタンプ情報がテキストである場合、図 2 7 に示す ( B ) の属性タブでは、文字列の内容、フォント種類、スタイル、サイズ、色の設定ができる。また、付加する固定スタンプ情報がイメージの場合は、図 2 7 に示す ( C ) の属性タブでは、イメージファイルのパスの指定、拡大 / 縮小の設定ができる。

30

【 0 0 9 0 】

また、図 2 6 に示すプレビュー画面 2 6 0 3 は、バリアブルスタンプが選択されている場合、1 レコード目の取得を行い、その表示を行う。

【 0 0 9 1 】

その後、ステップ S 2 4 1 0 において、製本アプリケーション 1 0 4 は、編集終了か否かの判定を行う。ここで「OK」2 6 0 4 のコントロールが選択されたとすると、編集が反映され、図 2 5 に示すスタンプ付加ダイアログに戻る。

40

【 0 0 9 2 】

また、ステップ S 2 4 0 4 において、スタンプ編集でなければ、ステップ S 2 4 0 5 へ進み、付加選択がされたか否かの判定を行う。これは、図 2 5 に示す UI で「原稿ページにスタンプを付ける」又は「印刷ページにスタンプを付ける」が選択されているか否かを判定する処理である。「印刷ページにスタンプを付ける」とは印刷用紙に対してスタンプを付ける処理で、前者の原稿ページと重複指定可能であるとする。また、ここでは「原稿ページにスタンプを付ける」が選択されたとする。

【 0 0 9 3 】

この選択により、ステップ S 2 4 0 6 において、製本アプリケーション 1 0 4 は、バリ

50



アブルスタンプが選択されているか否かの判定を行う。これは図 2 5 上でリストアップされているスタンプの内、選択されていたのがバリアブルスタンプであるか否かを判定する処理である。ここでは、バリアブルスタンプが選択されているものとし、ステップ S 2 4 0 8 へ進み、バリアブルスタンプ付加処理を行う。これは選択された原稿ページもしくは印刷ページにバリアブルスタンプを付加するために、その情報をフラグとして保持する処理である。ここで、選択されていたのがバリアブルスタンプでない場合は、ステップ S 2 4 0 9 へ進み、通常スタンプ付加処理を行う。

【 0 0 9 4 】

その後、図 2 5 に示す UI 上の「 O K 」コントロールによって実際にアプリケーション本体に情報が渡される。

10

【 0 0 9 5 】

次に、図 2 8 を用いて、バリアブルスタンプを含む製本アプリケーションの表示処理について説明する。図 2 8 は、本実施形態における製本アプリケーションの表示処理を示すフローチャートである。ここでは、上述したバリアブルスタンプ付加 UI よりバリアブルスタンプの選択、「原稿ページにスタンプを付ける」が設定されているものとする。

【 0 0 9 6 】

まず、ステップ S 2 8 0 1 において、製本アプリケーション 1 0 4 は、バリアブルスタンプ設定があるか否かの判定を行う。ここでは、設定がされているものと仮定しているので、バリアブルスタンプの設定があると判定する。そして、ステップ S 2 8 0 2 において、1 レコード目の取得処理を行う。これは既に選択したバリアブルスタンプの 1 レコード目を取得する処理である。また、レコード自体は先の登録において、データベースに接続した際に、電子原稿ファイル内部のメモリテーブル上に取得してあるとして、その中から 1 レコード目を取得する処理になる。ここではメモリテーブルとしたが、外部ファイルに保存されていても良く、その取得先は問わない。

20

【 0 0 9 7 】

次に、ステップ S 2 8 0 3 において、製本アプリケーション 1 0 4 は、レコードプレビュー処理を行う。これは先に取得した 1 レコード目と原稿を合わせて製本アプリケーション上で表示する処理である。この例では、ステップ S 2 8 0 2 で、「 P M 太郎」が 1 レコード目として取得されているとする。このときの表示を例示した図が図 2 9 である。図 2 9 に示すように、原稿データにプラスして「 P M 太郎」がスタンプとして付加され、表示される。

30

【 0 0 9 8 】

次に、ステップ S 2 8 0 4 において、製本アプリケーション 1 0 4 は、レコード移動選択がされたか否かの判定を行う。これはユーザが次のレコードのプレビューを所望し、図 2 9 に示すレコードの移動のコントロールが選択されたか否かを判定する処理である。ここでは、そのレコードのインクリメント処理のためのコントロールが選択されたとする。

【 0 0 9 9 】

そして、ステップ S 2 8 0 5 において、製本アプリケーション 1 0 4 は、次レコード取得処理を行う。これは 1 レコード目の取得処理 ( S 2 8 0 2 ) と同様の処理により、次レコードが取得される処理である。ここでは、このバリアブルスタンプの次レコードとして、「 i w 花子」が取得されたとする。

40

【 0 1 0 0 】

その後、レコードプレビュー処理 ( S 2 8 0 3 ) に戻り、次のレコードである「 i w 花子」を含むプレビューが行われる。このときの表示を例示した図が図 3 0 ある。図 3 0 に示すように、原稿データにプラスして「 i w 花子」がスタンプとして付加され、表示される。

【 0 1 0 1 】

このようにして、製本アプリケーション本体上で、バリアブルスタンプを付加した原稿のプレビュー及びレコードの移動が可能になる。

【 0 1 0 2 】

50

尚、図 28 に示すフローチャートでは、バリアブルスタンプを含む製本アプリケーションの表示処理について説明したが、印刷処理については、以下のように改良することで実現可能である。

【0103】

まず、ステップ S2803 の「レコードプレビュー処理」を、「レコード印刷処理」と置き換えて実現する。レコード印刷処理では、製本アプリケーション 104 は、電子原稿ファイルの各ページデータを取得し、電子原稿デスクトップ 105 を用いて、図示省略した OS のグラフィックエンジンを介して、プリンタドライバ 106 へ描画データの出力を行う。このとき、製本アプリケーション 104 は、ステップ S2802 で取得したバリアブルスタンプ情報を、図 26 及び図 27 で設定されている大きさ、配置、印刷方法の属性に従ってデータベースファイル中の対象レコードのデータを描画データとして出力する。

10

【0104】

そして、ステップ S2804 の「レコード移動が選択されたか？」を「次のレコードを印刷処理するか？」と置き換えて実現する。即ち、製本アプリケーション 104 は、ステップ S2803 で印刷処理したレコードの次のレコードを印刷処理するかを判断する。バリアブルスタンプ情報を用いた印刷を行う場合には、ユーザは印刷設定画面を介して印刷するレコード数を指定することが可能となる。ここで、1 レコード目から 3 レコード目というような設定が可能である。製本アプリケーション 104 が、次のレコードの印刷処理を行うと判断すると、ステップ S2805 へ処理を進める。他のステップは、図 28 に示す処理と同様である。

20

【0105】

尚、描画データを受け取ったプリンタドライバ 106 は、プリンタが解釈可能な印刷データに変換して、印刷ジョブを生成し、プリンタへ送信する。

【0106】

このようにして、製本アプリケーション 104 は、バリアブルスタンプとして可変スタンプ情報を用いた印刷データ生成処理を実現することが可能となる。

【0107】

本実施形態によれば、予め複数の文字列登録してあるデータベースに接続し、そのフィールド定義をスタンプに登録することが可能となり、その可変データ（バリアブル）でのスタンプ機能を同システム上から使用することが可能となる。従って、複数の使用したい文字列をその都度文字列毎に登録することなく付加することができる。

30

【0108】

また、印刷時においても、印刷したいレコード範囲を指定することにより、所望する複数のスタンプ結果を得ることができる。

【0109】

更に、これらは同システムの従来のバリアブル機能のテキストフィールド定義で代替しようとした場合と比較し、大幅な操作性向上となる。

【0110】

〔他の実施形態〕

次に、図 31 ~ 図 34 を用いて、本発明に係る他の実施形態について詳細に説明する。他の実施形態は、複製の防止という観点から原稿データにできる限りバリアブルスタンプを重ねるように自動的にスタンプ位置を調整するものである。

40

【0111】

他の実施形態における印刷システムの構成は、前述した実施形態で説明した構成と同様であり、その説明は省略する。

【0112】

図 31 は、他の実施形態におけるスタンプ位置の自動調整処理を示すフローチャートである。まず、ステップ S3101 において、スタンプ位置自動調整を行うか否かの判定を行う。これはバリアブルスタンプの自動位置調整を行うか否かの判定処理である。図 32 で例示する UI のように、スタンプの付加 UI で「自動位置調整」3201 コントロール

50

がチェックされているか否かを判定する処理である。ここでは、ユーザが自動位置調整を所望し、このコントロールがチェックされているものとする。

【0113】

ここでは、ステップS3102へ進み、レコード位置判定処理を行う。これは図26で示すスタンプ編集UIで設定されるレコードを置く位置と、先の1レコード目の取得処理(S2802)で取得されているレコードからその文字の置く位置を算出する処理である。ここでは、図33に示すように、バリアブルスタンプの1レコード目として「PM 花子」が取得され、その位置は図中の位置とする。

【0114】

次に、ステップS3103において、原稿位置判定処理を行う。これはスタンプを付加する対象となる、原稿における文字列等の位置を算出する処理である。これは後の処理により、全てのページにおいて行われるが、ここでは図33の1ページ目で示す原稿の文字列の位置が列挙される。

10

【0115】

そして、ステップS3104において、レコードと原稿の重なりがあるか否かの判定を行う。これは先に取得したそのレコードのスタンプ文字列の位置と、原稿にある文字列等の位置が重なっているかを判定する処理である。ここでは、図33に示すレコード、原稿を想定しているので、重なりがあると判定される。

【0116】

その後、ステップS3106において、最後のページか否かの判定により、原稿が最後か否かを判定する。この場合、図33に示す原稿を想定しているので、最後のページではないと判定し、ステップS3108において、対象ページの移動処理を行う。これは対象ページを次のページに移す処理で、ここでは2ページ目に対象を移動する。

20

【0117】

そして、ステップS3103に戻り、原稿位置判定処理を行い、2ページ目の原稿における文字列等の位置を算出する。ここでは、図33に示す2ページ目の原稿位置を算出する。レコード位置は同様であるため、ステップS3104において、再度レコードと原稿の重なりがあるか否かの判定処理を行い、重なりを検知する。ここで、図32で例示するUIのように、スタンプの付加UIで「自動位置調整」のコントロールがチェックされていなければ、通常のスタンププレビューとなり、図33の2ページ目のようなプレビューになる。しかし、ここでは、ステップS3105へ進み、レコード位置調整処理を行う。これはバリアブルスタンプと原稿に重なりがない場合、バリアブルスタンプが原稿に重なるように、その位置の平行移動を行う処理である。その平行移動の方向は、最も移動量の少ない方向となる。

30

【0118】

その後、全ての原稿ページに対して上述の処理が行われ、更に全てのレコードに対して処理が行われる。これにより、図34で例示するように、常に原稿に重なるようにバリアブルスタンプを全てのレコードに付加することが可能となる。

【0119】

また、前述した実施形態では、登録時に内部のメモリテーブルもしくは外部ファイルにレコードを既に取得していたが、登録時にはレコードを取得せず、必要な時に取得しても良い。例えば、1レコード目の取得処理(S2802)でデータベースから1レコード目を取得し、次レコード取得処理(S2805)で次のレコードを取得しても良い。

40

【0120】

他の実施形態によれば、原稿との重なりを持たせるために、可変データ(バリアブル)でのスタンプ位置をレコード、ページ毎に移動させることで、複製防止効果を向上させることができる。

【0121】

尚、本発明は複数の機器(例えば、ホストコンピュータ、インターフェース機器、リーダー、プリンタなど)から構成されるシステムに適用しても、1つの機器からなる装置(例

50

えば、複写機，ファクシミリ装置など）に適用しても良い。

【 0 1 2 2 】

また、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（ＣＰＵ若しくはＭＰＵ）が記録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。これによっても、本発明の目的が達成されることは言うまでもない。

【 0 1 2 3 】

この場合、記録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録媒体は本発明を構成することになる。

10

【 0 1 2 4 】

このプログラムコードを供給するための記録媒体として、例えばフレキシブルディスク，ハードディスク，光ディスク，光磁気ディスク，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。

【 0 1 2 5 】

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、次の場合も含まれることは言うまでもない。即ち、プログラムコードの指示に基づき、コンピュータ上で稼働しているＯＳ（オペレーティングシステム）などが実際の処理の一部又は全部を行い、その処理により前述した実施形態の機能が実現される場合である。

20

【 0 1 2 6 】

更に、記録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込む。その後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部又は全部を行い、その処理により前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【図面の簡単な説明】

【 0 1 2 7 】

【図 1】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 2】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。

30

【図 3】ブックファイルの構造の一例を示す図である。

【図 4】ブック属性の一覧を示す図である。

【図 5】章属性の一覧を示す図である。

【図 6】ページ属性の一覧を示す図である。

【図 7】ブックファイルを開く手順のフローチャートである。

【図 8】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 9】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 10】既存のブックファイルを開いた際のユーザインタフェース画面の一例を示す図である。

40

【図 11】新規のブックファイルを開いた際のユーザインタフェース画面の一例を示す図である。

【図 12】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】本実施形態における印刷システムの構成を示すブロック図である。

【図 14】一般的なスタンプ機能のスタンプ付加用ＵＩ画面の例を示す図である。

【図 15】スタンプ編集ＵＩ画面の例を示す図である。

【図 16】一般的なスタンプ付加機能を使用した印刷結果を示す図である。

【図 17】複数の任意の文字列によるスタンプ付加機能を使用した印刷結果を示す図である。

50

【図 18】本実施形態におけるバリアブルスタンプ登録削除処理を示すフローチャートである。

【図 19】バリアブルフィールドの定義を行うためのバリアブルエディタの一例を示す図である。

【図 20】バリアブルスタンプ一覧表示の一例を示す図である。

【図 21】バリアブルスタンプの追加設定においてユーザに詳細設定を促すための設定ダイアログの一例を示す図である。

【図 22】図 21 に示すコントロールの詳細を示す図である。

【図 23】バリアブルスタンプの登録完了を知らせるためのメッセージの一例を示す図である。

10

【図 24】本実施形態におけるバリアブルスタンプ付加編集処理を示すフローチャートである。

【図 25】登録済スタンプと登録したバリアブルスタンプの一覧表示の一例を示す図である。

【図 26】スタンプ編集のUI画面の一例を示す図である

【図 27】スタンプ編集ダイアログ内で、スタンプの属性、配置、印刷方法を設定する例を示す図である。

【図 28】本実施形態における製本アプリケーションの表示処理を示すフローチャートである。

【図 29】レコードプレビュー処理により 1 レコード目を表示した状態を示す図である。

20

【図 30】レコード移動により次のレコードを表示した状態を示す図である。

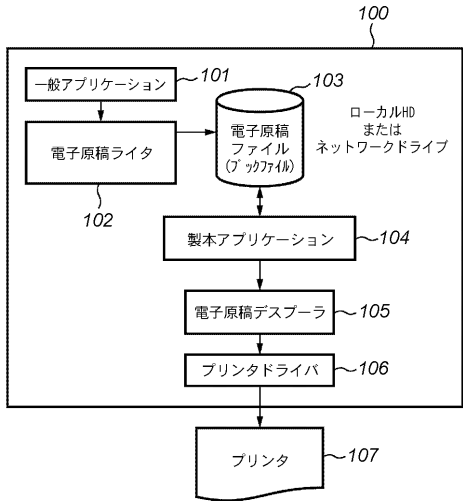
【図 31】他の実施形態におけるスタンプ位置の自動調整処理を示すフローチャートである。

【図 32】バリアブルスタンプの自動位置調整を指示するUI画面を示す図である。

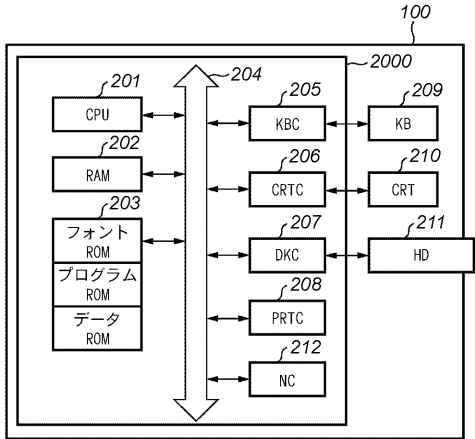
【図 33】バリアブルスタンプが原稿に載らないページの一例を示す図である。

【図 34】バリアブルスタンプをページ毎に自動的に移動させる処理の結果を示す図である。

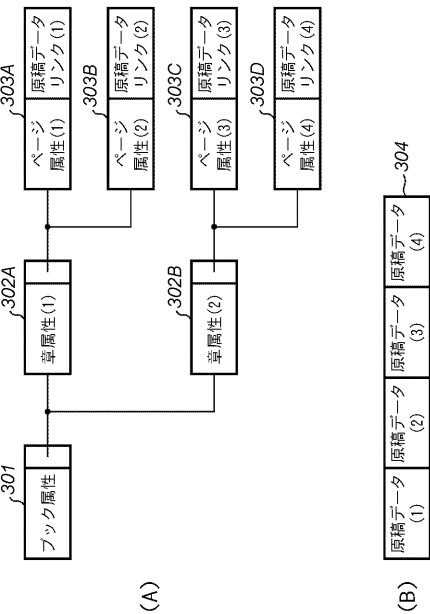
【図 1】



【図 2】



【図 3】



【図 4】

NO	属性情報	備考
1	印刷方法	
2	用紙サイズ	・ [M4493]、「レタ-サイズ」(11×17)指定の場合は7折指定 ・ 製本印刷またはup印刷が指定された場合は1巻/ページ目の原稿サイズが自動で選択される
3	用紙方向	・ 固定サイズの場合は選択可能
4	どじ代/じ方向	・ シフト/振動指定可能
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39パターン ・ 等倍印刷指定可能
6	拡大縮小	用紙サイズに固定サイズまたは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	・ 論理ページ単位、物理ページ単位に個別の指定が可能 ・ 全巻/全ページが対象 ・ 論理ページ単位、物理ページ単位に個別の指定が可能
8	ヘッダ・フッタ	・ 全巻/全ページが対象 ・ スタイブル/パンチ穴は片面/両面印刷のみ ・ スタイブルは箇所/2箇所
9	排紙方法	・ スタイブル/パンチ穴は片面/両面印刷のみ ・ スタイブルは箇所/2箇所
10	製本詳細	・ 表紙/裏表紙 ・ 表紙1/2、裏表紙1/2に対する印刷指定 ・ 表紙口(インサートを含む)指定 ・ インデックス部分への文字列印字、インデックス紙へのアノテーションを設定可能 ・ 製本印刷は指定不可
11	表紙/裏表紙	・ 表紙1/2、裏表紙1/2に対する印刷指定 ・ 表紙口(インサートを含む)指定 ・ インデックス部分への文字列印字、インデックス紙へのアノテーションを設定可能 ・ 製本印刷は指定不可
12	インデックス紙	・ 表紙1/2、裏表紙1/2に対する印刷指定 ・ 表紙口(インサートを含む)指定 ・ インデックス部分への文字列印字、インデックス紙へのアノテーションを設定可能 ・ 製本印刷は指定不可
13	合紙	・ 表紙1/2、裏表紙1/2に対する印刷指定 ・ 表紙口(インサートを含む)指定 ・ インデックス部分への文字列印字、インデックス紙へのアノテーションを設定可能 ・ 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」 ・ 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」

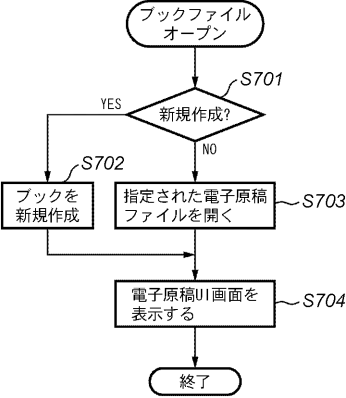
【図 5】

No	属性情報	備考	
		原稿サイズ	固定サイズ
1	用紙方向		
2	用紙縮小		
3	用紙縮小		
4	用紙縮小		
5	用紙縮小		
6	用紙縮小		
7	用紙縮小		

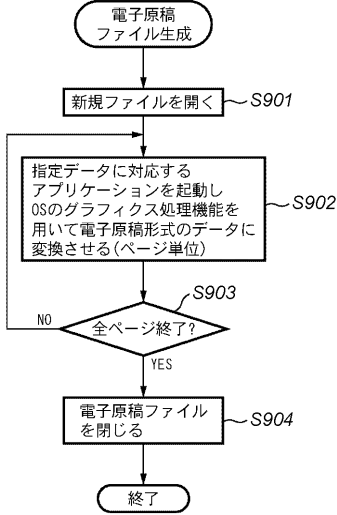
【図 6】

No	属性情報	備考	
		原稿サイズ	固定サイズ
1	用紙方向		
2	用紙縮小		
3	用紙縮小		
4	用紙縮小		
5	用紙縮小		
6	用紙縮小		
7	用紙縮小		
8	用紙縮小		

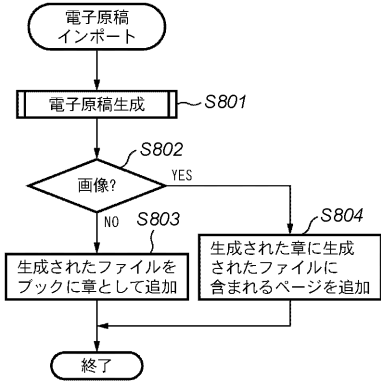
【図 7】



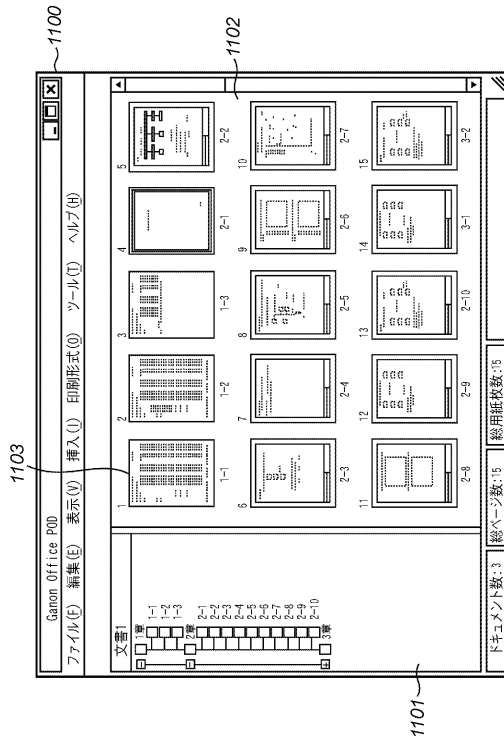
【図 9】



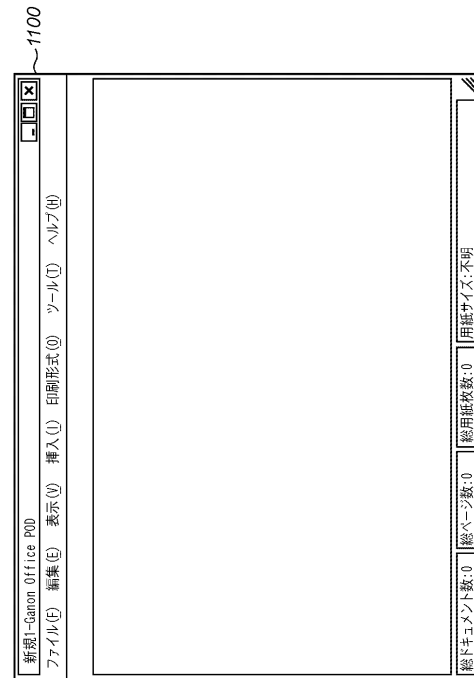
【図 8】



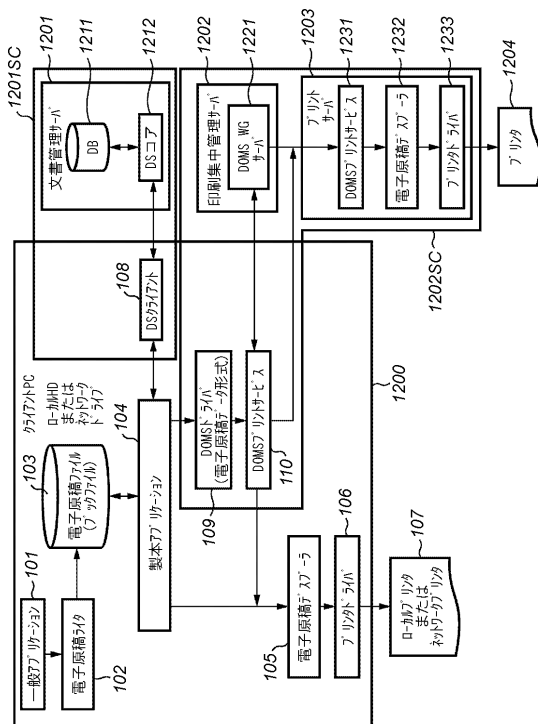
【図 10】



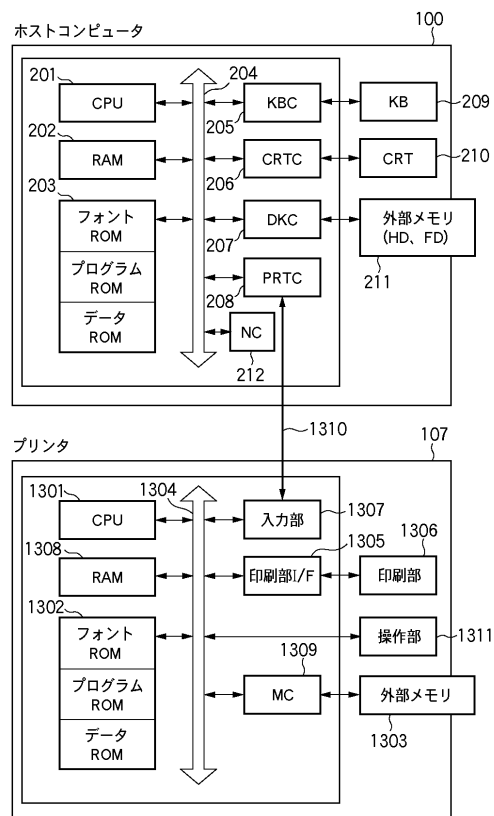
【図 11】



【図 12】

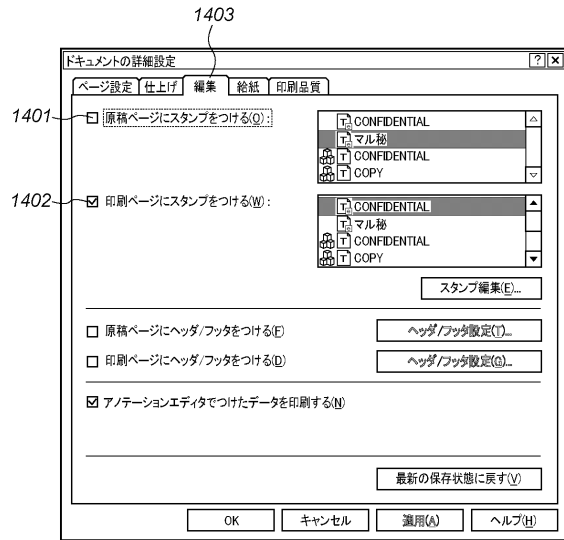


【図 13】

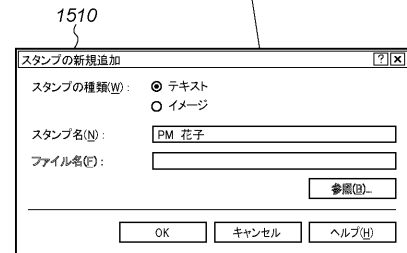
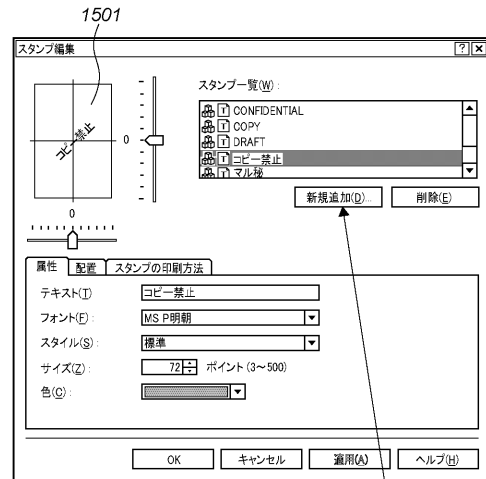




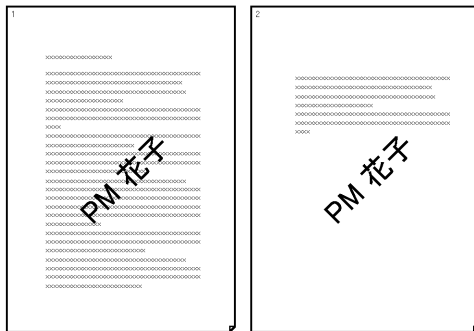
【図 14】



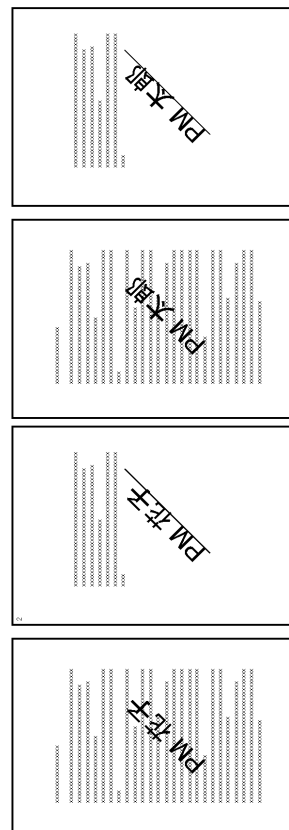
【図 15】



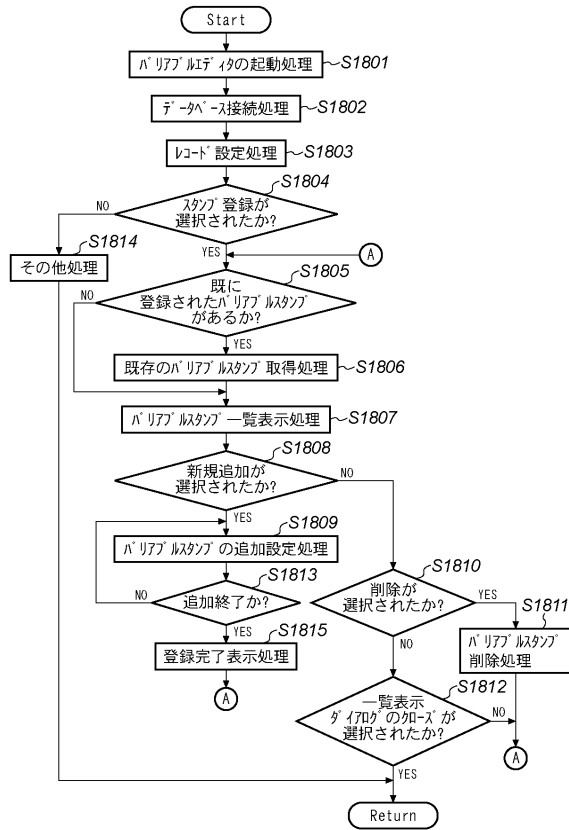
【図 16】



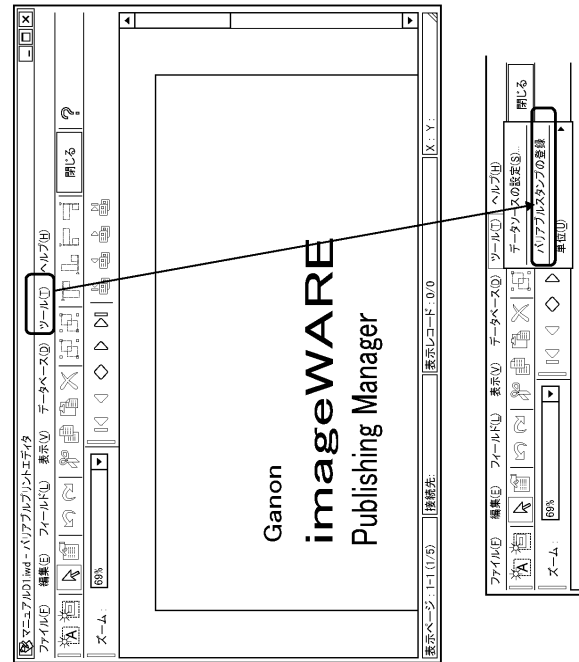
【図 17】



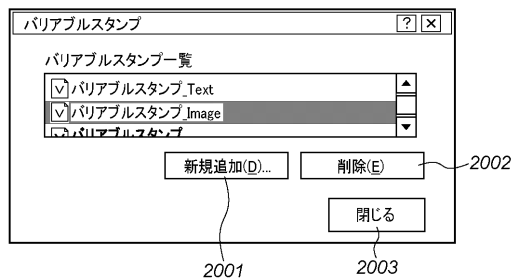
【図 18】



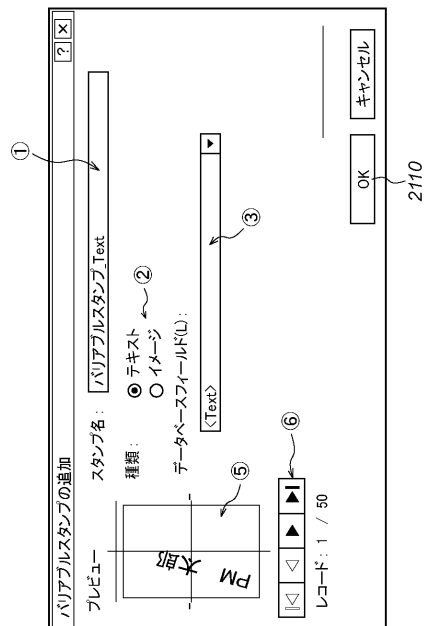
【図 19】



【図 20】



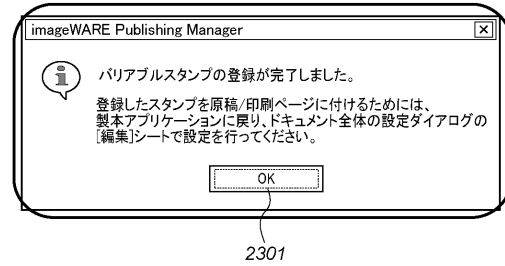
【図 21】



【図 22】

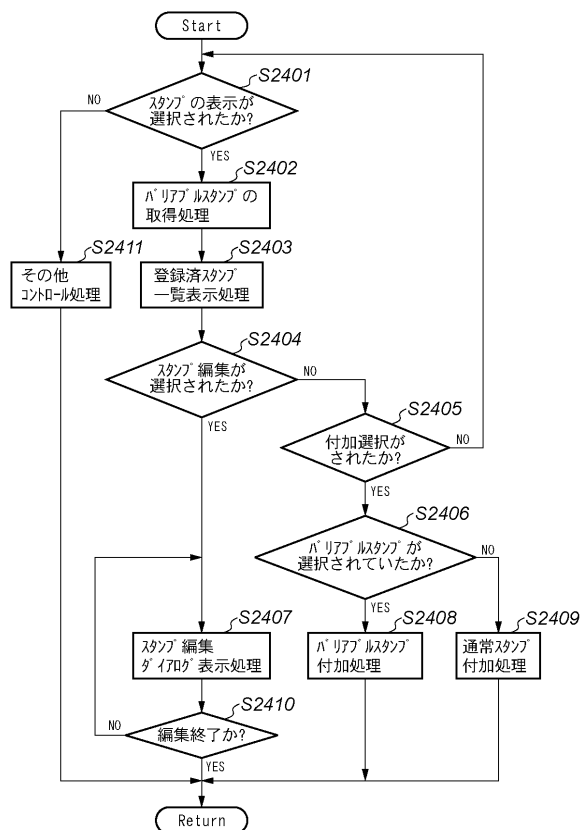
①スタンプ名指定ボックス
バリアブルスタンプの名前を設定する。 ユニークなものにする。
②バリアブルスタンプ種類選択チェックボックス
バリアブルスタンプの種類を選択する。 テキスト：テキストスタンプ イメージ：イメージスタンプ
③データベースフィールド選択リストボックス
スタンプに使う、バリアブルフィールドを選択する。 リストされるフィールド名は現在接続しているデータベースの カラム名 ※選択できるフィールド名は1つ。
⑤プレビュー
スタンプとして設定したデータベースデータをプレビューする。
⑥レコード移動ボタン
レコード移動を行うボタン。 レコードの番号によってプレビューの内容が変化する。

【図 23】

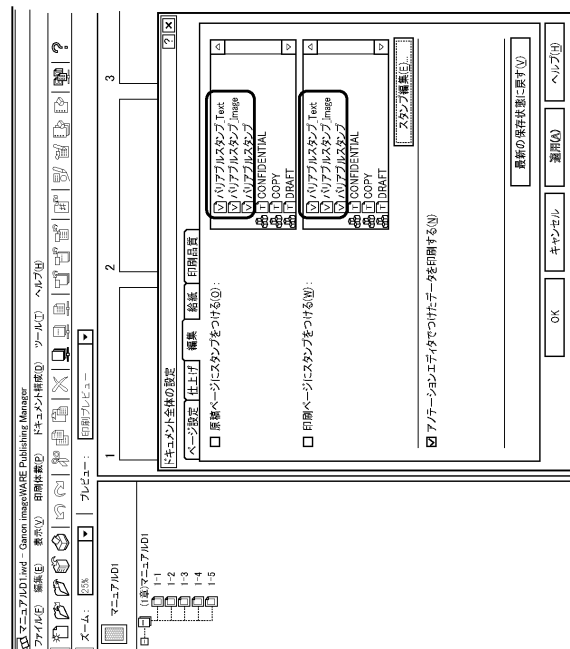


2301

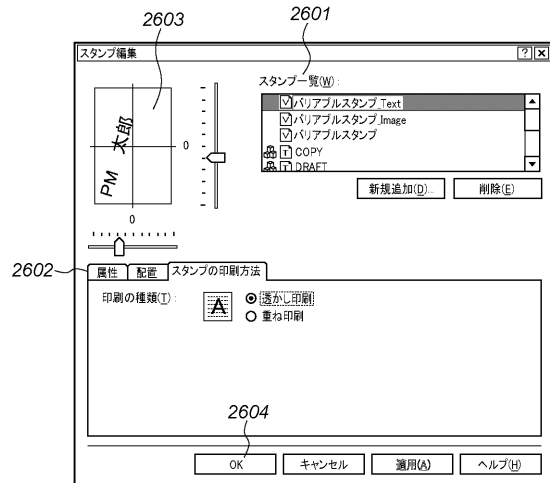
【図 24】



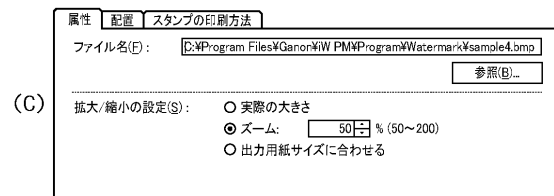
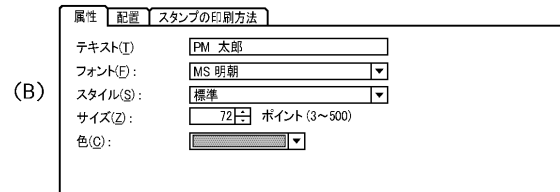
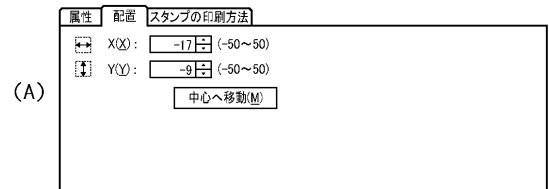
【図 25】



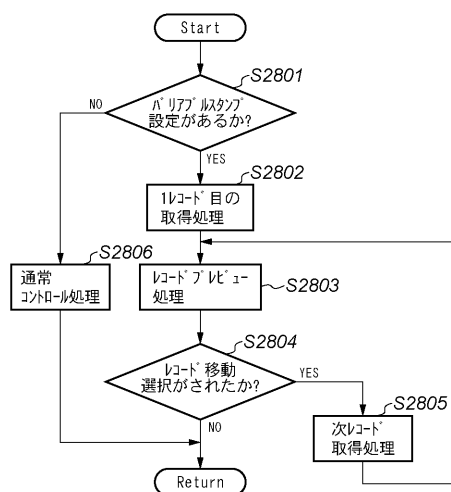
【図 26】



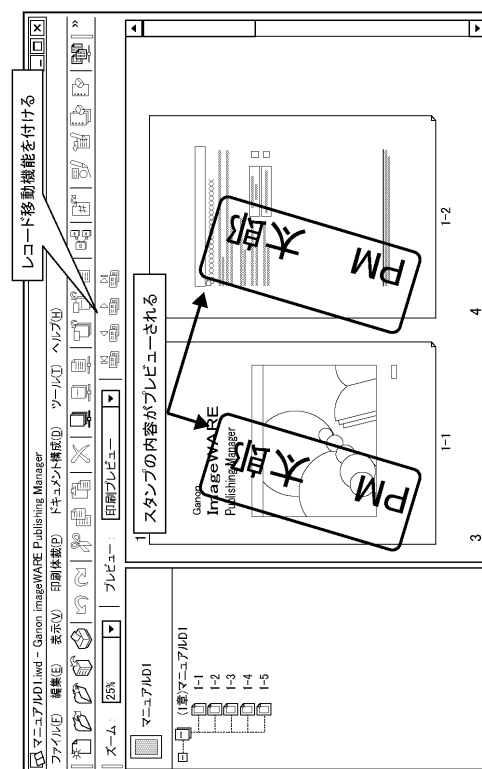
【図 27】



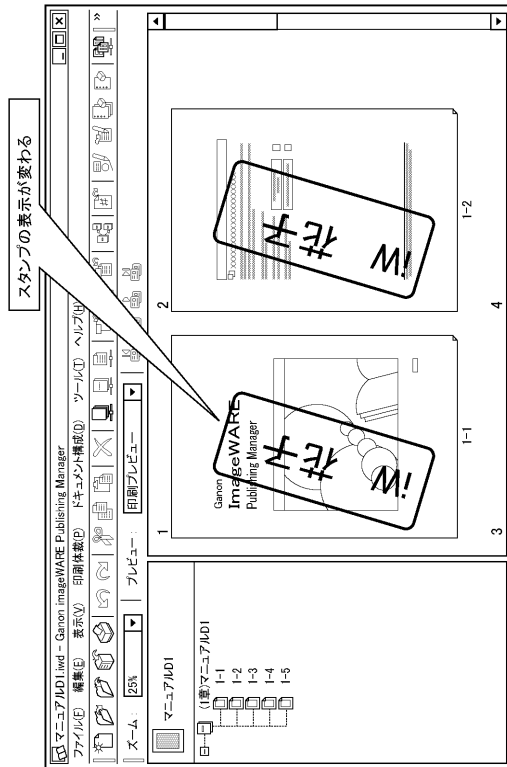
【図 28】



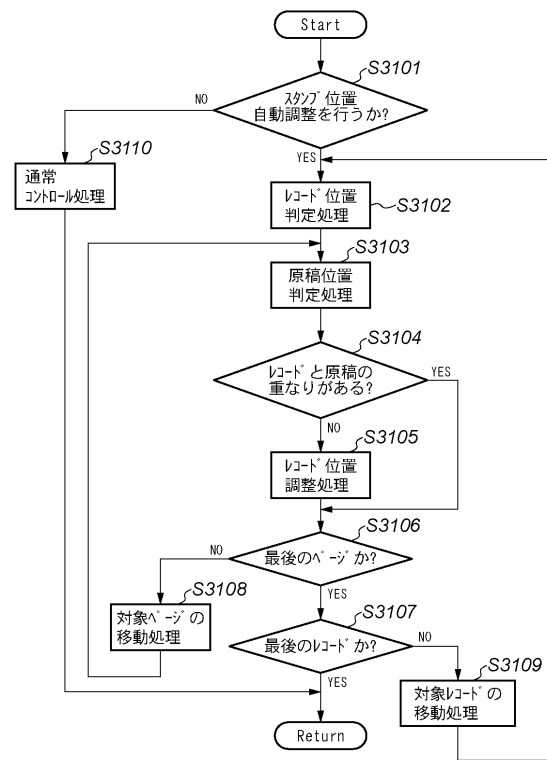
【図 29】



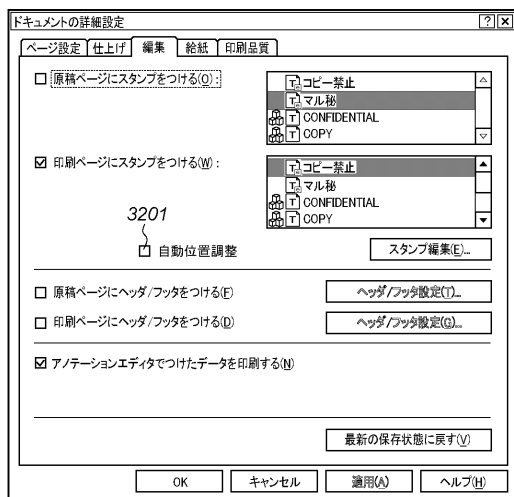
【図 30】



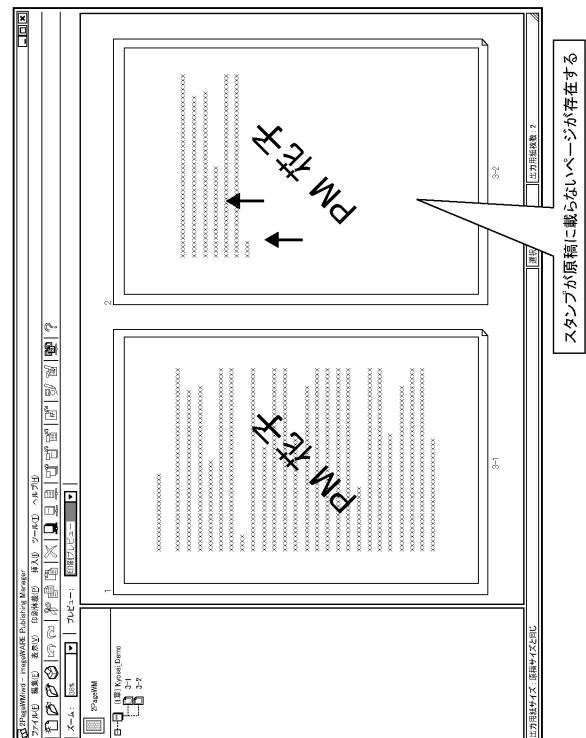
【図 31】



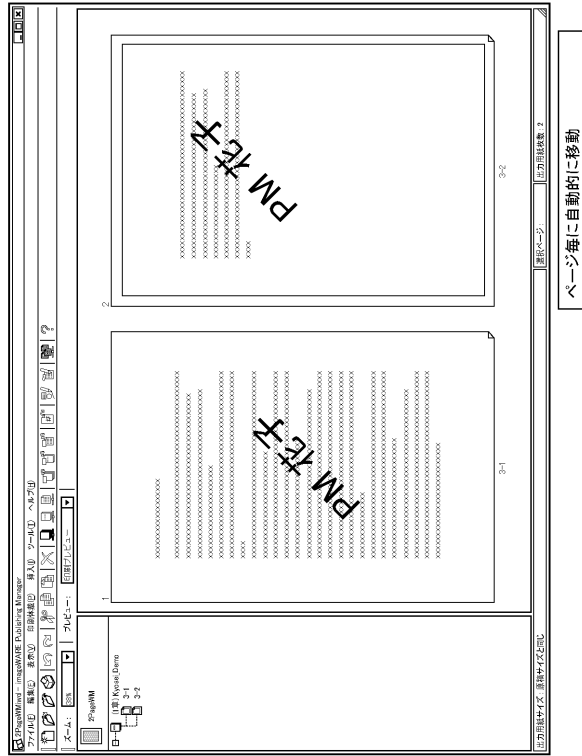
【図 32】



【図 33】



【 3 4 】



(30)

JP 4799253 B2 2011.10.26

---

フロントページの続き

審査官 山口 大志

(56)参考文献 特開2004-110354(JP,A)  
特開2001-324898(JP,A)  
特開2001-103271(JP,A)  
特開2006-011938(JP,A)  
特開平09-086015(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G06F 3/12  
B41J 29/40