

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4912139号
(P4912139)

(45) 発行日 平成24年4月11日(2012.4.11)

(24) 登録日 平成24年1月27日(2012.1.27)

(51) Int. Cl. F 1
G06T 11/60 (2006.01) G06T 11/60 100A
G06F 17/21 (2006.01) G06F 17/21 536

請求項の数 5 (全 49 頁)

(21) 出願番号	特願2006-346654 (P2006-346654)	(73) 特許権者	000001007
(22) 出願日	平成18年12月22日(2006.12.22)		キヤノン株式会社
(65) 公開番号	特開2008-158797 (P2008-158797A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成20年7月10日(2008.7.10)	(74) 代理人	100076428
審査請求日	平成21年12月17日(2009.12.17)		弁理士 大塚 康德
前置審査		(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
		(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】 情報処理装置

(57) 【特許請求の範囲】

【請求項1】

複数種類のデータフィールドからなるレコードから選択されたデータフィールドのデータを挿入するためのフィールド領域を配置したテンプレートを、指定されたサイズのドキュメントにレイアウトする情報処理装置であって、

前記テンプレートを、前記指定されたサイズのドキュメント上に設定されているフロー領域に挿入して、前記テンプレートのレイアウト処理を実行するレイアウト手段と、

前記フロー領域に挿入された前記テンプレート内のフィールド領域にデータを挿入した場合の該テンプレートの第1サイズを取得する取得手段と、

前記第1サイズと、前記フロー領域の第2サイズを比較する比較手段と、

前記比較手段の比較の結果、前記第1サイズが前記第2サイズより大きい場合、前記テンプレートを再定義する定義手段と、

前記テンプレートの再定義に関する設定を行うための設定手段とを備え、

前記設定手段は、前記テンプレートの再定義の可否を設定するための第1設定項目と、再定義を許可する場合に再配置するフィールド領域の外部余白を設定するための第2設定項目に関する設定を行い、

前記定義手段は、前記設定手段の設定情報に基づいて、前記テンプレートを再定義するか否かを判定する判定手段と、

前記判定手段の判定の結果、前記テンプレートを再定義する場合、前記テンプレート内に配置したフィールド領域のレイアウト構造を変更するために、その配置を再定義する再

10

20

定義対象のフィールド領域を、該フィールド領域にデータを挿入した場合のフィールド領域のサイズに基づいて判別する判別手段と、

前記判別手段で判別した前記再定義対象のフィールド領域の配置を変更することで、前記テンプレート内に配置したフィールド領域のレイアウト構造を変更する変更手段と
を備え、

前記判別手段は、再定義対象として判別した第 1 フィールド領域に第 2 フィールド領域を接続するためのリンクが設定されている場合には、前記第 2 フィールド領域も再定義対象のフィールド領域に判別する

ことを特徴とする情報処理装置。

【請求項 2】

前記定義手段は、前記第 1 サイズが前記第 2 サイズに収まるように、前記テンプレート内に配置したフィールド領域のレイアウト構造を変更する

ことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

前記定義手段によって、前記フィールド領域のレイアウト構造を変更した後の前記テンプレートの形状が前記フロー領域の形状に合うように、該テンプレートの形状を変形する変形手段を更に備える

ことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 4】

前記定義手段によるテンプレートを再定義する場合に前記リンクを解除するか否かを設定するためのリンク設定手段を更に備える

ことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 5】

前記判定手段の判定の結果、前記テンプレートを再定義しない場合、前記テンプレートを再定義しないで、前記テンプレート内に配置したフィールド領域のレイアウト構造を変更するための配置ルールを指定する指定手段を更に備える

ことを特徴とする請求項 1 に記載の情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数種類のデータフィールドからなるレコードから選択されたデータフィールドのデータを挿入するためのフィールド領域を配置したテンプレートをドキュメントにレイアウトする情報処理装置及びその制御方法、コンピュータプログラムに関する。

【背景技術】

【0002】

近年、商品の多品種化で商品ライフが短くなっていることが注目されている。また、インターネット利用の普及による消費者のカスタマイズサービス指向などの要因から CRM (Customer Relationship Management)、One-to-One マーケティングの必要性が注目されている。これらの手法により、顧客満足度を高め、顧客の開拓や囲い込みを目指すものである。

【0003】

One-to-One マーケティングは、データベース・マーケティングの一種である。これは、顧客の年齢、性別、趣味、嗜好、購買履歴等の個人属性情報をデータベース化し、その内容を分析、顧客のニーズに合った提案を行うものであり、その代表的な手法としてバリアブルプリントが挙げられる。ここ最近では、DTP (デスクトップパブリッシング) 技術の進展とデジタル印刷装置の普及に伴って、文書を顧客毎にカスタマイズして出力するバリアブルプリントシステムが開発されている。このようなバリアブルプリントシステムでは、顧客毎に異なる量のコンテンツが最適にレイアウトされたカスタマイズ文書を作成することが求められる。

【0004】

10

20

30

40

50

一般にバリアブルプリントシステムにおいて、そのようなカスタマイズ文書を作成する際には、文書上にコンテナをレイアウトする。ここで、コンテナとは、コンテンツ（描画内容（例えば、イメージ、テキスト等））を描画するための描画領域であり、フィールド領域と呼ばれることもある。

【0005】

そして、文書上にこのようなコンテナをレイアウトし、データベースとレイアウトを関連付ける（データベース内の各種コンテンツとコンテナとを関連付ける）作業を行う。これにより、所望のカスタマイズ文書（ドキュメントテンプレートと呼ばれる）を作成することができる。そして、このカスタマイズ文書上のコンテナ内のコンテンツを適宜切り換える（関連付けを変更する）ことで、その内容を可変（バリアブル）にすることができる。そのため、このような文書を、バリアブルデータ文書（ドキュメント）と呼んでおり、このバリアブルデータドキュメントを用いるプリントシステムが、バリアブルプリントシステムである。

10

【0006】

ここで、従来のバリアブルプリントシステムでは、コンテンツであるテキストやイメージに関連付けられるコンテナのサイズが固定である。そのため、データベース内のコンテンツがコンテナに挿入された（流し込まれた）ときに、そのデータ量がコンテナサイズより多いと、次のようなことが発生する。つまり、そのデータがテキストであれば、テキストのオーバーラップが発生し、また、そのデータがイメージであればイメージのクリッピングが発生する。また、データ量がコンテナサイズより小さいと、コンテナとその内部のコンテンツとの間に隙間が空いてしまう等の適切な表示がなされない可能性がある。

20

【0007】

これらの問題を解決するために、さらなるレイアウトを変更する技術として、あるコンテナのサイズが大きくなった場合、このコンテナに隣接する他のコンテナのサイズを小さくする技術が特許文献1の「レイアウトデザイン装置」に開示されている。

【0008】

また、従来のバリアブルプリントシステムにおいて、データベースの1データを1ドキュメントにレイアウトする技術だけでなく、複数のデータを1ドキュメントにレイアウトするマルチデータの技術も知られている。

【0009】

この技術により、一人の顧客に対して、レイアウトするデータの数が異なることが可能になり、さらに一人一人に対してよりカスタマイズされたドキュメントを作成することができる。

30

【0010】

そして、上記「レイアウトデザイン装置」に対して、さらにレイアウトエリアと称される機構を導入したものである。これは、基本となる形状を定めたサブテンプレートを、このレイアウトエリア内部に挿入させる（流し込ませる（フローさせる））ことにより、複数の類似項目をページ中の一定の領域内に挿入させる（流し込ませる（フローさせる））技術がある。

【0011】

例えば、特許文献2の「レイアウト調整方法及び装置並びにプログラム」では、サブテンプレート内に貼り付けられるコンテンツサイズに応じて、サブテンプレートのサイズやレイアウトを好適に設定する構成が開示されている。この構成によれば、レイアウトを行わせるユーザの意図をより反映させることができる技術が開示されている。

40

【0012】

ここで、「コンテンツ」とは、イメージデータ（画像データ）、テキストデータ等、バリアブルプリントシステムにおいて、ドキュメントを作成する際に、レイアウト上に配置、挿入される電子データの総称を意味するものである。

【0013】

また、「コンテナ」とは、バリアブルプリントシステムにおいて、ドキュメントのレイ

50

アウトを作成する際に、コンテンツの配置及び挿入対象となるレイアウト枠を意味するものである。

【0014】

また、「サブテンプレート」とは、バリアブルプリントシステムにおいて、ドキュメントのレイアウトを作成する際に、レイアウトに配置することを想定して、予め定義された小規模のテンプレート構造を意味するものである。

【特許文献1】特開平7-129658号公報（段落0049、図8）

【特許文献2】特開2006-074226号公報

【発明の開示】

【発明が解決しようとする課題】

10

【0015】

バリアブルプリントシステムでは、サイズの異なるコンテンツを適切にドキュメント上にレイアウトするために、コンテンツのサイズを挿入するコンテンツサイズに合わせて動的に変化させることを可能にするシステムが存在する。そして、商品カタログ等のドキュメントのレイアウトを作成するときに、各商品のレイアウト（商品名、商品イメージ、商品説明等）を予めサブテンプレートとして定義する機能を備えている。

【0016】

また、ドキュメント上で、サブテンプレートのフローを行うことを可能とする領域（以下、フローエリア（流し込み領域））を設定する構成を備えている。つまり、フローエリアは、ドキュメントテンプレート内に定義され、サブテンプレートがレイアウトされる領域である。そして、データベースのデータ種類に応じたサブテンプレートをフローエリアに流し込むことで、1ドキュメント上に複数のデータをそれぞれに対応したレイアウトで自動的に配置することが可能となる。

20

【0017】

但し、サブテンプレートは対応するデータ、つまり、挿入するコンテンツによってレイアウト構造を維持しつつ、サイズが変動する。

【0018】

サブテンプレートのサイズが、配置候補のフローエリアのサイズを超えた場合、フローエリア内でサブテンプレート単位でのコンテンツの回り込み（フローエリア内の空き領域への移動）が生じる。

30

【0019】

より具体的に説明すると、例えば、特許文献2では、各サブテンプレートは、フローエリアに設定された配置順に従って、フローエリアにレイアウトされる。例えば、左上から右下へZ配置（垂直方向への配置）を行うことが配置順で指定されている場合、行方向（水平方向）に順にサブテンプレートを配置し、配置するスペースがなくなると、列方向（垂直方向）に改行することになる。

【0020】

しかし、このようにサブテンプレートをフローエリアに配置すると、サブテンプレートのサイズによっては、フローエリア内に無駄なスペースができる場合があり、見た目が悪くなるケースが想定される。

40

【0021】

このように、フローエリアにそれぞれにサイズの異なるサブテンプレートが流し込まれることで、ユーザの意図の反映が少ないレイアウト結果が生じる。

【0022】

別の例としては、1ページ分のドキュメントに設定されたフローエリアに対して、ユーザが希望する数以下のサブテンプレートしか流し込めない場合は、その流し込めないサブテンプレートは、別ページのドキュメントに流し込むことになる。そのため、結果として出力するドキュメントの総ページ数が増加することが考えられる。

【0023】

また、1ページ分のドキュメントに設定されたフローエリアに対して、ユーザが希望す

50

る数以上のサブテンプレートが流し込まれる場合は、ページ内に余分な空白が生じることも考えられる。

【0024】

このように、バリアブルプリントシステムでは、サブテンプレートのサイズがコンテンツに応じてサブテンプレートのレイアウト構造を維持したまま変動することで、上記のようなユーザの意図しないレイアウト結果が生じるという課題がある。

【0025】

本発明は上記の課題を解決するためになされたものであり、ユーザの意図をより反映することができるドキュメント上のレイアウト制御を実現することができる情報処理装置及びその制御方法、コンピュータプログラムを提供することを目的とする。

【課題を解決するための手段】

【0026】

上記の目的を達成するための本発明による情報処理装置は以下の構成を備える。即ち、複数種類のデータフィールドからなるレコードから選択されたデータフィールドのデータを挿入するためのフィールド領域を配置したテンプレートを、指定されたサイズのドキュメントにレイアウトする情報処理装置であって、

前記テンプレートを、前記指定されたサイズのドキュメント上に設定されているフロー領域に挿入して、前記テンプレートのレイアウト処理を実行するレイアウト手段と、

前記フロー領域に挿入された前記テンプレート内のフィールド領域にデータを挿入した場合の該テンプレートの第1サイズを取得する取得手段と、

前記第1サイズと、前記フロー領域の第2サイズを比較する比較手段と、

前記比較手段の比較の結果、前記第1サイズが前記第2サイズより大きい場合、前記テンプレートを再定義する定義手段と、

前記テンプレートの再定義に関する設定を行うための設定手段とを備え、

前記設定手段は、前記テンプレートの再定義の可否を設定するための第1設定項目と、再定義を許可する場合に再配置するフィールド領域の外部余白を設定するための第2設定項目に関する設定を行い、

前記定義手段は、前記設定手段の設定情報に基づいて、前記テンプレートを再定義するか否かを判定する判定手段と、

前記判定手段の判定の結果、前記テンプレートを再定義する場合、前記テンプレート内に配置したフィールド領域のレイアウト構造を変更するために、その配置を再定義する再定義対象のフィールド領域を、該フィールド領域にデータを挿入した場合のフィールド領域のサイズに基づいて判別する判別手段と、

前記判別手段で判別した前記再定義対象のフィールド領域の配置を変更することで、前記テンプレート内に配置したフィールド領域のレイアウト構造を変更する変更手段と

を備え、

前記判別手段は、再定義対象として判別した第1フィールド領域に第2フィールド領域を接続するためのリンクが設定されている場合には、前記第2フィールド領域も再定義対象のフィールド領域に判別する。

【発明の効果】

【0029】

本発明によれば、ユーザの意図をより反映することができるドキュメント上のレイアウト制御を実現することができる情報処理装置及びその制御方法、コンピュータプログラムを提供できる。

【発明を実施するための最良の形態】

【0030】

以下、本発明の実施の形態について図面を用いて詳細に説明する。

<<実施形態1>>

<システム構成>

まず、実施形態1の情報処理システム及び、その構成要素であるホストコンピュータの

10

20

30

40

50

ハードウェア構成について、図 1 A 及び図 1 B を用いて説明する。

【 0 0 3 1 】

図 1 A は本発明の実施形態 1 の情報処理システムの構成例を示す図である。また、図 1 B は本発明の実施形態 1 の情報処理システムの一構成要素であるホストコンピュータのハードウェア構成を示す図である。

【 0 0 3 2 】

尚、図 1 A 中のホストコンピュータ以外の各種コンピュータ（例えば、データベースサーバ 1 1 7、ファイルサーバ 1 1 5 及びプリントサーバ 1 0 9）も、例えば、図 1 B のハードウェア構成と同等のハードウェア構成を有している。

【 0 0 3 3 】

図 1 A において、情報処理システム 1 0 0 は、ネットワーク 1 0 7 を介して、本発明の情報処理装置であるホストコンピュータ 1 0 1、データベースサーバ 1 1 7、ファイルサーバ 1 1 5 及びプリントサーバ 1 0 9 が相互に接続されている。

【 0 0 3 4 】

データベースサーバ 1 1 7 は、データベース 1 1 9 を備えている。また、プリントサーバ 1 0 9 は、プリンタ 1 1 3 を接続しており、ネットワーク 1 0 7 を介して受信する印刷データを適宜プリンタ 1 1 3 に出力して、その印刷データを印刷させることが可能である。

【 0 0 3 5 】

図 1 A の情報処理システム 1 0 0 は、特に、バリエブルデータドキュメントを印刷するバリエブルプリントシステムの構成例を示している。実施形態 1 で説明されるバリエブルプリント処理は、レイアウト編集装置として機能するホストコンピュータ 1 0 1（汎用コンピュータモジュールによって構成される）で実現される。

【 0 0 3 6 】

バリエブルプリントシステム 1 0 0 上で実施可能となるレイアウト編集アプリケーション 1 2 1 は、ホストコンピュータ 1 0 1 において、その全体あるいは一部分が実行される。特に、レイアウト編集に関する処理や、バリエブルデータドキュメントの印刷に関する処理は、ホストコンピュータ 1 0 1 によって実行されるソフトウェアにより実現される。

【 0 0 3 7 】

レイアウト編集アプリケーション 1 2 1 等のソフトウェアやコンピュータプログラムはコンピュータ可読媒体に格納され、そのコンピュータ可読媒体からホストコンピュータ 1 0 1 のメモリ 1 3 6 にロードされ、実行される。そのようなソフトウェアやコンピュータプログラムを格納したコンピュータ可読媒体は、コンピュータプログラム製品である。そして、例えば、ホストコンピュータ 1 0 1 において、そのコンピュータプログラム製品を使用することにより、バリエブルデータドキュメントのレイアウト編集やバリエブルプリントに好適な装置が提供されることになる。

【 0 0 3 8 】

図 1 B に示されるように、ホストコンピュータ 1 0 1 には、I/O（入出力）インタフェース 1 4 3 を介してキーボード 1 3 2 や、ポインティングデバイスであるマウス 1 3 3 が入力装置として接続される。また、出力装置としてのディスプレイ装置 1 4 4 がビデオインタフェース 1 3 7 を介して接続される。更に、プリンタ 1 4 5 を I/O インタフェース 1 3 8 を介して接続することも可能である。

【 0 0 3 9 】

また、I/O インタフェース 1 3 8 は、ホストコンピュータ 1 0 1 を、ネットワーク 1 0 7 へ接続する機能も有する。これにより、ネットワーク 1 0 7 を介して、バリエブルプリントシステム 1 0 0 内の他のコンピュータ装置（外部機器）にホストコンピュータ 1 0 1 を接続することができる。ネットワーク 1 0 7 の典型的な例としては、ローカルエリアネットワーク（LAN）、あるいはワイドエリアネットワーク（WAN）が挙げられる。

【 0 0 4 0 】

また、図 1 B に示すように、ホストコンピュータ 1 0 1 は、少なくとも 1 つのプロセッ

10

20

30

40

50

サ 1 3 5、例えば、半導体メモリである、ランダムアクセスメモリ (R A M) やリードオンリーメモリ (R O M) から構成されるメモリ 1 3 6 を含んでいる。記憶装置 1 3 9 は、プログラム等の各種データを格納するコンピュータ可読媒体との間でデータのやり取りが可能なハードディスクドライブ (H D D) 1 4 0 やフロッピー (登録商標) ディスクドライブ (F D D) 1 4 1 を含む。

【 0 0 4 1 】

尚、図 1 B には示されていないが、磁気テープドライブやメモリカード等の各種記憶装置も記憶装置 1 3 9 として使用することが可能である。C D - R O M ドライブ 1 4 2 は、不揮発性のデータソースとして提供される (もちろん、C D - R O M によってコンピュータプログラムを提供してもよい) 。

10

【 0 0 4 2 】

ホストコンピュータ 1 0 1 は、相互接続バス 1 3 4 を介して、ホストコンピュータ 1 0 1 の各種構成要素 1 3 5 ~ 1 3 4 と通信する。この通信は、例えば、G N U / L I N U X やマイクロソフトウィンドウズ (登録商標) のようなオペレーティングシステムや、典型的にはオペレーティングシステムに従う形で実現される。あるいは、この通信は、関連のある技術で知られているもので形成されたコンピュータシステムの常套的なオペレーションモードによる方法によって、実現される。即ち、上述の各種構成要素 1 3 5 ~ 1 4 3 は、相互接続バス 1 3 4 を介して通信可能に接続されており、ホストコンピュータ 1 0 1 にインストールされたオペレーティングシステムにより利用される。

【 0 0 4 3 】

20

尚、図 1 B に示すホストコンピュータ 1 0 1 の例としては、I B M 互換 P C (パーソナルコンピュータ) や S U N の S p a r c s t a t i o n、あるいはそれらを含んだコンピュータシステムが考えられる。

【 0 0 4 4 】

< レイアウト編集アプリケーションの概要 >

実施形態 1 では、レイアウト編集アプリケーション 1 2 1 は、ハードディスクドライブ 1 4 0 に常駐し、プロセッサ 1 3 5 による実行や読込が制御される。レイアウト編集アプリケーション 1 2 1 の媒介記憶装置とネットワーク 1 0 7 からフェッチされるデータは、ハードディスクドライブ 1 4 0 に呼応してメモリ 1 3 6 を使用する。

【 0 0 4 5 】

30

一つの例では、レイアウト編集アプリケーション 1 2 1 のエンコードされたプログラムが、C D - R O M やフロッピー (登録商標) ディスク上に格納される。そして、対応する C D - R O M ドライブ 1 4 2 やフロッピー (登録商標) ディスクドライブ 1 4 1 を通じて読み込まれ、ハードディスクドライブ 1 4 0 にインストールされる。

【 0 0 4 6 】

あるいは、別の例として、レイアウト編集アプリケーション 1 2 1 は、ネットワーク 1 0 7 からホストコンピュータ 1 0 1 内に読み込まれて、ハードディスクドライブ 1 4 0 にインストールされてもよい。

【 0 0 4 7 】

更に、レイアウト編集アプリケーション 1 2 1 を含む各種ソフトウェアは、磁気テープまたは R O M または集積回路、光磁気ディスクからホストコンピュータ 1 0 1 内にロードされてもよい。または、ホストコンピュータ 1 0 1 とその他のデバイス間における赤外線等の無線通信からホストコンピュータ 1 0 1 内にロードされてもよい。または、P C M C I A カードのようなコンピュータ可読カード、そして E メール通信や W E B サイト上の記録情報を持つインターネットやイントラネットを含む他の適当なコンピュータからホストコンピュータ 1 0 1 内にロードされてもよい。これらは、コンピュータ可読媒体の例であり、他のコンピュータ可読媒体が使用されてもよいことは明らかである。

40

【 0 0 4 8 】

図 1 A において、レイアウト編集アプリケーション 1 2 1 は、ホストコンピュータ 1 0 1 にバリアブルプリント (バリアブルデータプリント (V D P) ともいう) を実現させる

50

ものである。このレイアウト編集アプリケーション121は、2つのソフトウェアコンポーネント、即ち、レイアウトエンジン105とユーザインタフェース103を含んでいる。

【0049】

レイアウトエンジン105は、描画領域（部分領域）であるコンテナ（矩形の範囲）に与えられたサイズや位置の制限にしたがって、データベース119にレコード単位で格納されているバリエブルデータから1レコードずつ読み込む。そして、その読み込んだバリエブルデータとコンテナの制限とから、読み込んだバリエブルデータが流し込まれるコンテナのサイズや位置等のレイアウトを計算するソフトウェアコンポーネントである。

【0050】

また、レイアウトエンジン105は、更に、コンテナに割り当てられたバリエブルデータを描画し、バリエブルデータドキュメントのイメージを生成する処理も行う。但し、本発明は、これに限るものではなく、レイアウトエンジン105は、各部分領域（コンテナ）のサイズと位置を決定するアプリケーションとして動作し、プリンタドライバ（不図示）に描画情報を出力する。これにより、プリンタドライバがバリエブルデータドキュメントのイメージ描画処理を行い、印刷データを生成してもよい。

【0051】

ユーザインタフェース103は、ユーザによるコンテナのレイアウトや属性設定を可能とし、ユーザにドキュメントテンプレートを作成させる。また、ユーザインタフェース103は、ドキュメントテンプレート内の各コンテナとデータソース（データベース119上のバリエブルデータ（コンテンツ））とを関連付けるメカニズムを提供する。

【0052】

UIモデルアナライザ104は、ユーザインタフェース103が提供するコンテナや、アンカー、スライダー、リンク等のユーザインタフェース要素をレイアウトエンジン105が認識できる内部形式に変換する。即ち、UIモデルアナライザ104により、レイアウトエンジン105に変更を加えることなく、ユーザインタフェース103を他の表示/入力形式のユーザインタフェースに容易に置換可能になる。

【0053】

ユーザインタフェース103とUIモデルアナライザ104は、通信チャンネル123を介して通信する。また、UIモデルアナライザ104とレイアウトエンジン105は、通信チャンネル124を介して通信する。

【0054】

バリエブルデータドキュメント生成のためのデータソースは、一般的には、データベースアプリケーションを実行している、他のコンピュータで構成されるデータベースサーバ117上にある典型的なデータベース119が挙げられる。

【0055】

ホストコンピュータ101は、ネットワーク107を介して、データベースサーバ117と通信する。レイアウト編集アプリケーション121は、ホストコンピュータ101、あるいは、一般的には、他のコンピュータで構成されるファイルサーバ115に保存されるドキュメントテンプレートを生成する。

【0056】

また、レイアウト編集アプリケーション121は、コンテナに流し込まれるコンテンツであるバリエブルデータ（コンテンツデータ）とマージされたドキュメントテンプレートによって構成されたバリエブルデータドキュメントを生成する。これらのバリエブルデータドキュメントは、ホストコンピュータ101のローカルファイルシステム、ファイルサーバ115に保存される、あるいはプリントサーバ109を介してプリンタ113に直接印刷される。

【0057】

ここで、プリントサーバ109は、直接ネットワーク107に接続されていないプリンタ113に、ネットワーク機能を提供するコンピュータである。プリントサーバ109と

10

20

30

40

50

プリンタ 113 は、典型的な通信チャネル 111（例えば、USB、IEEE 1394、無線 LAN 等）を介して接続される。

【0058】

次に、バリアブルプリントシステム 100 の別構成例について、図 1C を用いて説明する。

【0059】

図 1C は本発明の実施形態 1 の情報処理システムの別構成を示す図である。

【0060】

図 1C は、ネットワーク 107 上にエンジンサーバ 227 を追加して、このエンジンサーバ 227 にレイアウトエンジン 225 を構成した例を示している。つまり、ホストコンピュータ 101 内のレイアウトエンジン 105 を、エンジンサーバ 227 内で実現した構成である。この構成の場合、ホストコンピュータ 101 の処理負荷を軽減することができる。

10

【0061】

尚、このエンジンサーバ 227 は、他のサーバと同様、典型的なコンピュータである。また、ファイルサーバ 115 に保存されたドキュメントテンプレートは、印刷や他の目的がある際、レイアウトエンジン 225 によってドキュメントを生成するためにデータベース 119 に保存されたデータと結合することができる。そのようなオペレーションは、ユーザインタフェース 103 を介して要求されるか、特定のレコードのみ印刷するように要求される。

20

【0062】

次に、バリアブルデータプリントの概略について、図 2 を用いて説明する。

【0063】

図 2 は本発明の実施形態 1 のバリアブルデータプリントの概略を説明するための図である。

【0064】

レイアウト編集アプリケーション 121 のユーザインタフェース 103 により、ユーザからの操作指示に従いページ上に複数のコンテナ 181 ~ 183 を配置する。そして、各コンテナに位置やサイズに関する制約条件を付与することによりドキュメントテンプレート 180 が生成される。

30

【0065】

また、ユーザインタフェース 103 は、ドキュメントテンプレート 180 とデータソース 190（例えば、データベース 119）との関連付け、更に各コンテナとデータソース 190 内の各データフィールドとの関連付けを行う。各コンテナとデータソース 190 内の各データフィールドとの関連付けを示す関連付け情報は、ドキュメントテンプレート 180 内に記述され、該ドキュメントテンプレート 180 は、HDD 140 に格納される。また、データソース 190 は、レコード単位で項目データが記載されているファイルであり、HDD 140 に格納されている。

【0066】

レイアウトエンジン 105 は、ユーザからの印刷指示もしくはプレビュー指示に応じて、ドキュメントテンプレート 180 の各コンテナ 181 ~ 182 に、関連付け情報で関連付けられたデータをデータソース 190 から読み込む。次に、その読み込んだデータをレコード単位で流し込み（例えば、データレコード 1 のデータフィールド A ~ C をコンテナ 181 ~ 183 へ流し込む）、流し込まれたデータに応じて各コンテナのサイズ等を調整（レイアウト調整）する。

40

【0067】

プレビュー指示の場合は、レイアウト調整されたドキュメントイメージを生成し、ディスプレイ装置 144 の画面上にプレビューとして表示するべく出力される。また、印刷指示の場合は、レイアウトエンジン 105 もしくはプリンタドライバを用いて生成したドキュメントイメージを印刷データとしてプリントサーバ 109 へ出力する。データレコード

50

1, 2, 3...を順次処理することにより、バリアブルデータプリントが実現されることになる。

【0068】

<レイアウト編集アプリケーションの説明>

以下、レイアウト編集アプリケーション121について説明する。

【0069】

まず、ユーザインタフェース103によって実現されるユーザインタフェースの一例について、図3を用いて説明する。

【0070】

[メインウインドウ]

図3は本発明の実施形態1のユーザインタフェースの一例を示す図である。

【0071】

図3に示すように、ユーザインタフェース103は、操作時にアプリケーションウインドウ301によって形成されたユーザインタフェースをディスプレイ装置144に表示させる。このアプリケーションウインドウ301は、メニューバー302、ツールバー303、ワークエリア306とオプションのパレット311を有する。

【0072】

メニューバー302とツールバー303は非表示にしたり、画面上の色々な場所に移動することが可能である。また、ワークエリア306は、マウス133の操作によってその場所を移動させることが可能である。また、パレット311はオプションであり、目的に応じて表示/非表示を制御することができる。更に、カーソル/ポインタ313は、マウス133の指示位置を表示する。

【0073】

メニューバー302は、周知の技術として知られているように、メニューオプションの階層の下に拡張される多くのメニューアイテム304を有する。

【0074】

ツールバー303は、アプリケーションの特別なモードによって非表示状態にする、または表示状態にすることが可能な多くのツールボタンとウィジェット(部品)305を有する。

【0075】

ルーラー308はオプションであり、ワークエリア306内のポインタ、ページ、ライン、マージンガイド、コンテナまたはオブジェクトの位置を示すために使用される。

【0076】

パレット311は、バリアブルデータライブラリのような追加機能にアクセスするために使われる。パレット311は、移動、リサイズ、クローズをするためのウインドウコントロール312を有する。また、パレット311は、ワークエリア306の前面に表示したり、あるいはオブジェクトの背面に隠すことができる。更に、パレット311は、アプリケーションウインドウ301の範囲内のみはその表示を制限したり、アプリケーションウインドウ301の外側にその一部或いは全体を表示することが可能である。

【0077】

ツールバー303には、図4に示されるような、ユーザが選択可能な複数種類の『ボタン』403~406が配置されている。

【0078】

(1) 選択ツールボタン403: コンテナの辺を選択、移動、サイズ変更、リサイズそしてロック/ロック解除のために使われる。コンテナの選択は、コンテナの周囲に選択ボックスをドラッグすることによりなされる。また、キーボード132上のCTRLキーを押しながら、複数のコンテナについて選択操作をすることによって、複数のコンテナを選択可能である。

【0079】

(2) テキストコンテナツールボタン404: スタティックあるいはバリアブルテキスト

10

20

30

40

50

トを有するコンテナを作成するために使用される。

【0080】

(3) イメージコンテナツールボタン405: スタティックあるいはバリアブルイメージを有するコンテナを作成するために使用される。

【0081】

(4) リンクツールボタン406: コンテナ間に関連付けを行うリンクを作成するために使われ、リンクの距離をコントロールするためにも使用される。

【0082】

尚、これらのボタンは、周知の技術で知られているように、操作状況に合わせて変化するアイコンのツールチップとして実装される。

【0083】

また、アプリケーションウィンドウ301は、ページ内に各コンテナやリンクをレイアウトすることで、基本レイアウトを決定することができる。基本レイアウトとは、バリアブルデータプリントで基本となるレイアウトのことである。基本レイアウト内の各コンテナが固定コンテナである場合は、すべてのレコードの印刷結果のレイアウトは同じになる。

【0084】

また、基本レイアウト内の各コンテナが可変コンテナである場合は、レコード単位に読み込まれるデータの量やサイズにより各コンテナのサイズや位置が、制約の範囲内で変動することになる。

【0085】

よって、レイアウト編集アプリケーション121で作成されるドキュメントテンプレートは、あくまで基本レイアウトを決定するものである。そのため、動的コンテナが含まれる場合は、最終的な印刷物のレイアウトは読み込まれるデータによりレイアウト調整されることになる。

【0086】

[ドキュメントテンプレート]

図3において、ワークエリア306は、ドキュメントテンプレート(180:基本レイアウト)のデザインを表示・編集するために使われる。これは、ユーザがドキュメントテンプレートをデザインする過程において、印刷されるドキュメントの概観をユーザに提示することを可能とする。これにより、ユーザは、データソース(190)とマージされたドキュメントが、バリアブルデータの量・サイズに基づいてどのように変化するかを容易に理解することができる。

【0087】

また、データソースがドキュメントテンプレートに関連付けられていた場合は、現在のドキュメントのプレビューができるように、対応するバリアブルテキストやイメージがレイアウトされた各コンテナに表示される。

【0088】

ドキュメントテンプレートにおけるドキュメント構造とコンテナを描写する視覚的な手がかり(コンテナの枠線、アンカー、スライダー、リンク等)は、ドキュメントテンプレート作成時には常に表示される。また、バリアブルデータを流し込むプレビュー時には、視覚的な手がかりは、カーソルをコンテナ上に移動させたときや、コンテナを選択したときに表示される。

【0089】

ワークエリア306は、スクロールバー307と、オプションのルーラー308と、ドキュメントテンプレート309を含む。ドキュメントテンプレート309は、ページが複数あることを示すことができる。また、ドキュメントテンプレート309は、図2のドキュメントテンプレート180に相当するものである。

【0090】

与えられたドキュメントテンプレートのページサイズは、周知の技術を用いて、ユーザ

10

20

30

40

50

によって指定される。例えば、メニューの「ファイル」から「ページ設定」を選択することでページサイズを設定するダイアログを表示し、そこでユーザが指定したページサイズが反映されることになる。

【0091】

それぞれのドキュメントでの実際のページ数は、関連付けられたデータソース内のバリエーションによって変化する可能性がある。これは、ドキュメントテンプレート内に動的表のように、バリエーションの量により大きさが変更されるフィールドが設定されている場合である。つまり、この場合には、1ページ内にバリエーションデータをフィットできないバリエーションデータが読み込まれると、追加のページが自動的に作成されるからである。

10

【0092】

それぞれのページ内に示される境界線310は、ページ上の印刷可能なオブジェクトの最大幅を示す、任意のページマージンである。

【0093】

また、図4では、1ページのドキュメントテンプレート309上に表示され得るオブジェクトの例を示している。

【0094】

このようなオブジェクトとしては、コンテナ407、408と、任意に適用するアンカーアイコン409、固定されている辺411、414、固定されていない辺410、リンク412、そしてスライダー413がある。

20

【0095】

アンカーアイコン409は、コンテナの矩形の角、辺、またはコンテナの中央に設定することが可能である。アンカーアイコン409が設定されると、設定された個所の位置が固定となる。つまり、図4の例では、アンカーアイコン409は、コンテナ407の左上の角に設定されている。そのため、コンテナ407はバリエーションデータが流し込まれ、バリエーションデータの画像サイズもしくはテキスト量が多い場合には、右方向及び下方向に拡大可能であることを示している。

【0096】

一方、アンカーアイコン409が辺に設定されている場合は、その辺が固定となり、その他の3辺の各方向に拡大可能である。また、アンカーアイコン409がコンテナの中央に設定されている場合は、コンテナの中央位置が固定となり、コンテナ矩形の中央位置が変わらないように、4方向に拡大可能である。

30

【0097】

リンク412についての詳細は後述するが、これは、コンテナ407とコンテナ408が関連付けられていることを示しており、このリンク412に設定されている長さ(範囲指定可能)を保ちつつ、コンテナ408が右方向に移動可能であることを示している。スライダー413は、設定されている辺と水平方向に移動可能であることを示している。

【0098】

[コンテナ]

次に、データベース中の各レコードに含まれる複数種類のデータフィールドのデータを差し込むためのフィールド領域であるコンテナについて説明する。

40

【0099】

コンテナは、ドキュメントテンプレート内にバリエーションデータファイルから固定あるいは可変のテキスト/イメージ(複数種類のデータフィールドのデータ)が流し込まれ、描画されるフィールド領域(これを部分領域と呼ぶ)である。そして、このコンテナは、図4に示されるように他のコンテナやオブジェクトと共にレイアウトされる。ユーザインタフェース103を介して、ユーザからの操作指示により、コンテナは、マウス133の操作により移動、サイズ調整、再作成される。

【0100】

より正確には、コンテナは、設定の集まり、視覚的表現、そしてインタラクションと編

50

集動作をもっている。以下、実施形態 1 におけるコンテナの定義を示す。

【 0 1 0 1 】

(1) コンテナは、固定あるいは可変のコンテンツを持つ。可変コンテンツ (バリアブルデータ) は、データソースから取得したデータがドキュメント毎、つまり、レコード毎に異なる可能性があるという意味でダイナミック (動的) であるということができる。但し、実施形態 1 の可変コンテンツは、アニメーション化されたもの、あるいは他の方法で時間的に変化するコンテンツは印刷には適していないため、ここでは、意図していない。

【 0 1 0 2 】

同様に、固定コンテンツは、コンテナを使って生成される全てのドキュメントで、同じように表示される。しかしながら、可変コンテンツとリンクが設定されている場合、可変コンテンツの影響を受けて、固定コンテンツはそれぞれのドキュメントで位置が異なる可能性がある。

【 0 1 0 3 】

(2) コンテナは、コンテンツに適用される背景色、ボーダー、フォント・スタイルのようなテキスト設定と同様の装飾機能を持っている。このような設定を、コンテナ属性と呼ぶ。コンテナ属性は、各コンテナごとに設定可能であるが、あるコンテナと同じコンテナ属性であるという設定を行うことも可能である。

【 0 1 0 4 】

(3) コンテナは、ドキュメントを生成する際にデータソースからのデータとマージされる。装飾機能は、どんな固定コンテンツでもそうであるように、印刷された出力物において可視である。可変コンテンツは、データソースからの特定のデータの表示を提供する。コンテナのこの表現は、例えば、印刷されるか、ディスプレイ装置 1 4 4 のスクリーン上に表示されるか、その両方が可能である。

【 0 1 0 5 】

(4) コンテナは、図 4 に示されるように視覚的な手がかりとしてのユーザインタフェースを有している。例えば、コンテナの編集そして表示設定のためのインタラクティブなグラフィカルユーザインタフェース (G U I) を有する。G U I コンポーネントは、ディスプレイ装置 1 4 4 のスクリーン上に表示されるが、ドキュメントとしては印刷されない。レイアウト編集アプリケーション 1 2 1 のユーザインタフェース 1 0 3 は、背景色やフォントのようなコンテナの装飾機能のいくつかを表示し、さらにコンテナの設定の編集や表示を可能にするための機能を有している。

【 0 1 0 6 】

ユーザインタフェースの機能の特別な目的の例としては、ボーダー、あるいは、コンテナのサイズや位置を対話的に変更、表示するための角アイコンがある。あるいは、コンテナがデータソースからデータをマージされたとき、コンテナの動作を示すための上塗りした数、線、アイコン、テキストがある。

【 0 1 0 7 】

[コンテナの制約]

コンテナはそれぞれのドキュメントで表示されるコンテンツをどのように結びつけるかの制御に関する制約がある。これらの制約 (固定 / 可変コンテンツをコンテナと結びつけることを含む) は、ユーザが一つのドキュメントテンプレートから多数のドキュメントの生成をコントロールする主要な方法である。

【 0 1 0 8 】

制約の一つの例には、『このコンテナのコンテンツの高さは最大 4 インチです』がある。また、別の制約の例には、『コンテナのコンテンツの左辺は、それぞれのドキュメントにおいて同じ水平位置で表示しなければならない』がある。ここに記述される内容は、G U I を使ってこのような制約を表示、編集するためのいろいろな方法である。

【 0 1 0 9 】

イメージがページ上に定義された場所をもっているように、固定コンテンツの配置を指定するコンテンツプレイスホルダーは、デジタル印刷技術でよく知られている。コンテナ

10

20

30

40

50

は、位置とサイズを持ち、それらは公知の技術で知られている手法で編集され、表示される。よって、以下の説明では、バリエブルデータプリントに特化した方法における表示・編集に焦点を合わせる。

【0110】

コンテナを用いることにより、ユーザは、ドキュメントにおけるコンテンツのサイズ（描画サイズ）や位置を指定することが可能となる。いく種類ものドキュメントが一つのドキュメントテンプレートから生成されるので、コンテナに多数の可能性と制約を設定することになるが、これらの設定（指定）や表示のために所定のユーザインタフェースが利用される。

【0111】

1つのコンテナの辺は、関連付けられたコンテンツがドキュメント内で表示される仮想の境界線を定義する。従って、コンテナの左辺を論じることは、関連付けられたコンテンツが、各ドキュメントにおいて、表示可能であるエリア内の最も左の辺を論じることと同じである。

【0112】

同様に、コンテナの高さを論じることは、生成されたドキュメントで関連付けられたコンテンツの高さの制約を論じることとして理解される。本明細書では、ユーザインタフェース103を参照して、コンテナの辺あるいは大きさを論じるところで、この区別は明らかにされるであろう。

【0113】

以下の記載において、コンテンツの表示を制限するために使われるある値を定義している用語『固定』は、全てのドキュメントで同じである。

【0114】

(1) コンテナの幅が固定である場合、関連付けられたコンテンツに割り当てられる幅は、全てのドキュメントで同じになる。

【0115】

(2) コンテナの高さが固定である場合、関連付けられたコンテンツに割り当てられる高さは、全てのドキュメントで同じになる。

【0116】

(3) 距離（リンクの長さ）が固定である場合、指定された距離は全てのドキュメントにおける制約となる。

【0117】

(4) コンテナの左右辺が固定の場合、ページに関する辺の水平位置は全てのドキュメントで同じであることを意味している。但し、コンテナの高さあるいは垂直方向の位置は、変化する可能性がある。例えば、コンテナの左辺が固定である場合、関連付けられたコンテンツの表示位置は、全てのドキュメントでその左辺の位置は同じ水平位置となる。しかしながら、あるドキュメントではページの上の方に表示され、他のドキュメントではページの下の方に表示される可能性がある。

【0118】

(5) コンテナの上下辺が固定の場合、ページにおける辺の垂直位置は全てのドキュメントで同じとなることを意味している。但し、コンテナの幅あるいは水平位置はドキュメントによって変わる可能性がある。

【0119】

(6) コンテナの垂直軸は、コンテナの右辺と左辺に平行で、それらの中間に位置される仮想の垂直線である。もしコンテナの垂直軸が固定なら、当該コンテナの左右辺の水平位置の平均（即ち、左右の中央位置）は、すべてのドキュメントで同じとなる。この制約において、コンテナの幅は変化する可能性がある。しかしながら、左右辺が垂直軸にもっとも遠いものからもっともものまで、垂直軸は全てのドキュメントで同じ水平位置となる。尚、コンテナの高さと垂直位置はこの制約によって影響されない。

【0120】

10

20

30

40

50

(7) 同様に、もし水平軸が固定なら、コンテナの上辺と下辺の平均が同一の垂直方向位置に配置される。但し、コンテナの幅と水平位置はこの制約によって影響されることはない。

【0121】

(8) 水平軸と垂直軸の両方が固定である場合、コンテナの中心位置が固定されていることを意味する。但し、コンテナの幅・高さは、この制約によって影響されない。

【0122】

(9) コンテナの角位置、コンテナの辺の中間位置、あるいはコンテナの中心位置が固定である場合、それぞれの位置はすべてのドキュメントで同じ場所となる。例えば、もしコンテナの左上角が固定なら、配置されたコンテナの左上位置が全てのドキュメントで同じになることを意味している。

10

【0123】

(10) 垂直辺あるいは垂直軸は、ページの左辺もしくは右辺、あるいは左ページマージンもしくは右ページマージン、あるいは他の水平位置に関連付けられて固定することができる。同様に、水平辺あるいは水平軸はページの上辺もしくは下辺、あるいは上下ページマージン、あるいは他の垂直位置に関連付けられて固定することができる。

【0124】

『固定』の反対は、コンテナの辺、軸、角、中間位置、あるいはドキュメント制約がドキュメント間(レコード間)で変化するかもしれないことを意味する『可変』である。例えば、ページ内では、バリエブルデータのサイズや量により、動的にレイアウトが変更されることを期待する。一方で、特定のコンテナについては、大きさや位置を固定にしたり、また、ページの角のコンテナの四隅は固定にしたいということを所望する場合がある。

20

【0125】

そのため、レイアウト編集アプリケーション121では、各コンテナ(部分領域)について、辺、軸、角、中間位置等を固定にするか、可変にするかを適宜設定できるようにしている。これにより、ユーザはドキュメントテンプレート180の基本レイアウトを決定する場合に、ユーザが所望とするように基本レイアウトを作成することができる。

【0126】

[コンテナ表示・編集]

- - 新規コンテナの作成方法 - -

コンテナは、テキストコンテナとイメージコンテナの2種類で記述される。テキストコンテナはテキストおよび埋め込みのイメージを有する。イメージコンテナは、イメージだけを有する。

30

【0127】

図4で示されるように、新規のテキストコンテナあるいはイメージコンテナは、テキストコンテナツールボタン404あるいはイメージコンテナツールボタン405をマウス133でクリックし、ドキュメントテンプレート309上に四角形をドラッグする。これによって、当該ドキュメントテンプレート309上に作成される。

【0128】

あるいは、コンテナは、所望のテキストコンテナツールボタン404あるいはイメージコンテナツールボタン405をアクティブにした後に、ドキュメントテンプレート309上で単にクリックすることによって作成されるようにしてもよい。この場合、マウス133のクリック操作に応じてデフォルトサイズのコンテナがテンプレート上に挿入されるとともに、当該新規コンテナの寸法等を設定するためのダイアログボックスあるいは他のプロンプトが提供される。

40

【0129】

尚、コンテナサイズは、自動的に前もって定義されるようにしてもよいし、あるいは、計算されたスキーマによって作成・配置される等、種々の方法が考えられる。ここで、生成されたコンテナをマウス等の入力装置により選択し、右クリックでプロパティを指示する等の操作を行うことにより、コンテナのプロパティダイアログが表示され、コンテナの

50

制約を設定することができる。

【0130】

[コンテナの表示方法]

図5(A)から(D)は、コンテナの辺に関する表示ルールを例示している。

【0131】

レイアウト編集アプリケーション121は、コンテナの辺の状態を表現するために、実線503(アイテム)あるいは点線504を用いて辺を表す。これに加えて、アンカーアイコン506、507、509(コンテナの辺の近くに描画される線、形状、アイコン)を用いる。更に、ハンドル502(移動、修正するために領域の辺上あるいは近傍に描画されるコントロール点)、スライダー413(辺の両側に描画される短い並行線、図4参照)、拡張アイコン505、色を用いる。

10

【0132】

図5(A)~(D)に示されるコンテナ表示方法のルールは、以下の通りである。

【0133】

(1)それぞれの辺を固定するために、実線で描画する。

【0134】

(2)幅が固定の場合は、左と右の辺を実線で描画する。

【0135】

(3)高さが固定の場合は、上と下の辺を実線で描画する。

【0136】

(4)軸は描画しない。

20

【0137】

(5)(1)~(3)によって描画されていないそれぞれの辺の近くには、拡張アイコンが描画され、それらの辺を点線で描画する。

【0138】

(6)垂直辺と水平辺、あるいは垂直軸と水平軸のそれぞれのペアで、もし両者が固定なら、それらの交差点にアンカーが描画される。

【0139】

(7)それぞれの固定辺で、もし辺のどこにもアンカーが描画されていなければ、辺の中央にスライダーが描画される。

30

【0140】

(8)垂直及び水平辺、あるいは垂直及び水平軸のそれぞれのペアで、アンカーやスライダーが描画されていない場合、それらの交差点にハンドルが描画される。

【0141】

ルール(1)、(2)、(3)で定義された線は、前述したように固定あるいは制限されているため実線で描画される。ルール(5)のように、可変の辺は、点線で描画される。ルール(6)、(7)、(8)で定義された固定された点は、アンカーを表示し、いくつかの固定された辺は、スライダーを表示し、他はハンドルを表示する。

【0142】

上記のルールは、ユーザにより後で設定された制約が優先される。つまり、後で別の制約が設定された場合、上記のルールが描画されるべき辺に影響すれば、実線や点線の描画内容が変更されることになる。例えば、もしコンテナがとても小さくアイコンがお互い重なるか、あるいは他の表示機能を不明瞭にするならば、アイコンは異なって、あるいは省略されて描画される可能性がある。

40

【0143】

可変の辺が描画される場所は、コンテナのコンテンツに依存する。後に説明するように、ドキュメントテンプレートにコンテンツがマージされて、ユーザインタフェースで可視になることを意味する、『動的な校正処理』が使われる。代替の実行は、すべてのドキュメントで平均化されるコンテナのコンテンツエリアで、あるいは、可変の辺がユーザインタフェースで、どこにレイアウトされるべきか決定するほかの手段で使われることがで

50

きる。

【0144】

これらのコンテンツ表現は、コンテナの各辺の状態を表示するグラフィック機能を提供する。その表現の解釈は、下記の通りである。

【0145】

(1) 図4の辺410のように、点線は、コンテナのコンテンツに依存してドキュメント内の辺の位置が変化することを意味する。

【0146】

(2) 実線の辺414は固定されている、あるいはコンテナの幅・高さが固定されている(コンテナ408では、4辺が実線であり、両方が固定されている)ために制限された辺であることを意味する。

10

【0147】

(3) アンカーは、辺および軸が交差した場所が固定されていることを意味する。それゆえ、アンカー点は、すべてのドキュメントの水平、垂直位置で現れることになる。アンカーは当然固定される。図4のアイコン409は、辺414の交差する位置が固定されていることを意味しているアンカーアイコンの例である。

【0148】

(4) スライダーは、関係付けられた辺の長さが固定されているが、並行移動する可能性があることを意味する。例えば、図4で、スライダー413は、コンテナ408のコンテンツが、ドキュメント内で特定のダイアグラムで表される位置の、左あるいは右に表示されるかもしれない。

20

【0149】

例えば、コンテナ408と関連付けられている(リンク設定されている)コンテナ407に流し込まれるデータの画像サイズもしくはテキスト量が少ない場合は、コンテナ407のサイズが小さくなる。そのため、コンテナ408は、左方向にスライド(並行移動)してレイアウトされて表示されることになる。また、コンテナ407のサイズが大きくなる場合は、逆にコンテナ408は右方向にスライドしてレイアウトされることになる。

【0150】

これらのアイコン・辺のいくつかあるいは全ては、どのツール、どのコンテナを、選択・ハイライトあるいはアクティブにするかによって、描画されたりされなかったりする。一般的に、コンテナの辺・アイコンはドキュメントテンプレートのデザインの手助けであるため、印刷物には描画されない。

30

【0151】

尚、前述したように、コンテナの幅・高さの基本値・最小値・最大値の基本パターンの設定は、副次的なダイアログウインドウに表示される。

【0152】

図5(A)で、コンテナ501は、幅・高さの両方が固定されていない(可変である)。固定された辺503は実線で表現され、可変の辺504は点線で表現されている。拡縮アイコン505は、隣接する辺504が可変であることを示す。他の形態のインジケータを代わりにあるいは追加的に用いてもよい。

40

【0153】

図5(B)において、コンテナ501は、幅・高さ両方が可変である。アンカーアイコン506が、交差している両方の辺503の角の位置が固定されていることを明示的に表すべく追加されている。

【0154】

図5(C)において、コンテナ501は、コンテナの幅及び高さの両方が可変であり、任意のアンカーアイコン507で示されるような中心点の周りを平等に広がるという状態を示している。即ち、コンテナ501は、アンカーアイコン507を中心に拡大あるいは縮小が可能である。ここでの拡大/縮小は、アンカーアイコン507の位置が常にコンテナ501の中心点となるようにレイアウト調整される。

50

【 0 1 5 5 】

図5 (D) において、コンテナ5 0 1は、上辺5 0 8が固定されているが、幅・高さの両方が可変である。上辺5 0 8の中心に位置付けられて示されるアンカーアイコン5 0 9は、固定されている。そしてコンテナ5 0 1の左辺・右辺 (5 0 2) は、アンカーアイコン5 0 9を通して垂直な中心軸 (垂直軸) の周りを、拡大・縮小する。

【 0 1 5 6 】

[リンク]

リンクは、コンテナとコンテナの関連を示している。関連とは、コンテナ間の距離を示しており、リンクによって関連付けられたコンテナ同士は、互いのレイアウト変更の影響を受けて動的レイアウト計算を実行する。例えば、上述のように、図4のリンク4 1 2は、コンテナ4 0 7とコンテナ4 0 8とを関連づけている。リンクの設定方法、及びリンクで関連付けられたコンテナの動的レイアウト計算方法については、後述する。

10

【 0 1 5 7 】

[リンクの設定方法]

次に、コンテナ同士を関連付けるためのリンクの設定について説明する。

【 0 1 5 8 】

図6は本発明の実施形態1のリンクの設定処理を示すフローチャートである。また、図7は本発明の実施形態1のリンク設定時のユーザインタフェースの遷移例を示す図である。以下、図6及び図7を用いてコンテナにリンクを設定する方法について説明する。

【 0 1 5 9 】

まず、ステップS 6 0 1において、レイアウト編集アプリケーション1 2 1は、ユーザインタフェースのワークエリア3 0 6上に編集対象の選択されたドキュメントテンプレートを表示する。リンクを設定するためには、リンクを設定するためのコンテナ (最低2つ) がドキュメントテンプレート上に作成されている必要がある。図7 (A) ~ (C) では、ステップS 6 0 1で、2つのコンテナを作成してリンクを設定する場合のユーザインタフェースの遷移例を示している。

20

【 0 1 6 0 】

次に、ステップS 6 0 2において、レイアウト編集アプリケーション1 2 1は、リンクツールを選択状態にする (図4のボタン4 0 6をクリックすることにより選択状態となる) 。

30

【 0 1 6 1 】

図7 (A) において、コンテナ7 0 1と7 0 2はすべて固定されている辺で構成されているものとする。また、7 0 3と7 0 4は、図4の4 0 9と同じであり、アンカーを意味する。7 0 5は、マウスポインタを意味している。

【 0 1 6 2 】

さて、リンクツールが選択状態となっている間に、ユーザはリンクを設定する2つのコンテナの内的一方 (コンテナ7 0 1とする) をクリックして選択する。この操作に応じて、レイアウト編集アプリケーション1 2 1のユーザインタフェース1 0 3は、第1のコンテナが選択されたことを認識し (ステップS 6 0 3) 、選択されたコンテナを特定する情報を保持する。

40

【 0 1 6 3 】

また、以降のマウスカソールの移動に応じた軌跡を画面に表示するようにする。例えば、図7 (B) における線分7 0 6は、 (A) の状態におけるクリック位置と現在のマウスポインタ7 0 5の位置とを結んだ線を示しており、このUIによりどの位置にリンクが設定されるのかをユーザに明示することができる。

【 0 1 6 4 】

次に、ユーザは、図7 (B) で示されるように、もう一方のコンテナ (コンテナ7 0 2) までマウスポインタ7 0 5を移動してクリックする。この操作に応じて、ユーザインタフェース1 0 3は第2のコンテナが選択されたことを認識し (ステップS 6 0 4) 、選択されたコンテナを特定する情報を保持する。

50

【0165】

レイアウト編集アプリケーション121は、ステップS603で選択された第1のコンテナと、ステップS604で選択された第2のコンテナとの間にリンクを設定する。

【0166】

こうして、ユーザにより選択された2つのコンテナ701、702の間にリンクが設定されると、リンク707を表示する(ステップS605)。更に、このリンク設定を受けて、コンテナの表示状態は、図7(C)の状態になる(ステップS606)。

【0167】

即ち、リンクが設定されたことにより、コンテナのUIが自動的に変更される。ここでは、リンクによって関連付けられた辺が可変となり、点線で示されている状態を示している。つまり、図7(C)において、708は点線で示されている辺であり、前述した通り可変の辺を示すものである。

10

【0168】

尚、図7(C)のようなコンテナの辺の状態の変化は、リンクを設定したことによりコンテナの辺を可変にする必要が生じたことにより自動的に実行されたものである。これは、リンクを設定したにもかかわらず全ての辺が固定であるという矛盾の発生を防ぐことを目的としている。また、709は図5の505と同じで、リンクを設定したことにより、コンテナが変化できる方向をユーザに視覚的に示したマークである。また、図7(C)の例では、左のコンテナの右辺と右のコンテナの左辺が可変な状態へ変化したが、これは一例であり、例えば、右コンテナが図4のスライダー413を持つ設定に変化しても構わない。

20

【0169】

<レイアウトエンジンによるレイアウト計算処理>

[レイアウト計算方法(全体フロー)]

実施形態1のレイアウト編集アプリケーション121は、少なくとも2つのモードを備えている。1つのモードは、ユーザインタフェース103を用いてコンテナを作成し、そのコンテナ間に関連付け(リンク設定)を行ってレイアウトを作成するレイアウトモードである。もう1つのモードは、レイアウトエンジン105により、作成したレイアウトにデータソースの各レコードを挿入して、実際にレコードが挿入された後のレイアウト結果をプレビューするプレビューモードである。

30

【0170】

このプレビューモードにおいて、実際のレコードが挿入され、レイアウトを計算する。但し、プレビューモードは、表示上でのレイアウト計算である。実際に印刷する場合においても、レイアウトエンジン105が各コンテナにデータを挿入してレイアウトを計算するが、その際の計算方法はプレビューモードと同じである。

【0171】

図8は本発明の実施形態1のレイアウト計算処理を示すフローチャートである。

【0172】

まず、プレビューモードが選択される(ステップS801)。プレビューモードになったら、レイアウト編集アプリケーション121は、ユーザにプレビューするレコードをデータソースより選択させ、選択されたレコードの各フィールドデータを各コンテナに挿入する(ステップS802)。

40

【0173】

各コンテナへフィールドデータを挿入すると、レイアウト編集アプリケーション121は、そのレコードをレイアウトするためのレイアウト計算を行い、必要に応じてレイアウト調整を行う(ステップS803)。尚、ステップS803におけるレイアウト計算の詳細については後述する。

【0174】

そして、レイアウト編集アプリケーション121は、ステップS803で計算されたレイアウトを表示(プレビュー)する(ステップS804)。レイアウト編集アプリケーシ

50

ョン121は、他のレコードについてもプレビューを行うか否かをユーザの指示により判定する(ステップS805)。ステップS805で、他のレコードについてプレビューを行う必要がない場合(ステップS805でNO)、プレビューモードを終了する(ステップS807)。

【0175】

一方、他のレコードについてプレビューを行う場合(ステップS805でYES)、レイアウト編集アプリケーション121は、他のレコードを選択して再度レイアウト計算を行い、プレビューを行う(ステップS806)。

【0176】

尚、プレビューモードでなく印刷時においては、印刷するレコード全てについて順にレイアウトの計算を行う。従って、ステップS804は存在せず、ステップS805は印刷するレコードを全て処理したかの判断を行う。ステップS803で、レイアウト計算された結果を、描画出力して出力し、プリンタドライバを用いて印刷データとして生成し、プリンタに印刷データが出力される。この場合、全てのレコード(印刷対象の指定された全レコード)について印刷データの出力が終了した時点で、本処理を終了することになる。

【0177】

[レイアウト計算方法(詳細)]

次に、上記ステップS803のレイアウト計算の詳細について、図9を用いて説明する。

【0178】

図9は本発明の実施形態1のレイアウト計算処理の詳細を示すフローチャートである。

【0179】

尚、図9は、レイアウト計算処理(動的レイアウト計算処理)についてのみ説明するためのフローチャートであるため、バリアブルデータプリントの1レコードの印刷/プレビュー時のレイアウト計算処理に相当する。複数レコードの場合は、下記の処理が繰り返されることになる。

【0180】

また、この動的レイアウト計算処理は、ドキュメントテンプレートに対して、データベース119から各レコードのコンテンツを流し込み(マージ)、各コンテナ属性と、各コンテンツの量・サイズに応じて、動的にレイアウト計算を行う。更に、図9の処理は、レイアウト編集アプリケーション121のレイアウトエンジン105が、プロセッサ135及びメモリ136を用いて動作する制御となる。

【0181】

まず、レイアウト編集アプリケーション121は、レイアウトを計算するコンテナの集合を設定する(ステップS901)。レイアウト計算は、関連付けられたコンテナを一つの集合として計算を行う。

【0182】

例えば、図10を参照すると、ページ上に4つのコンテナがレイアウトされており、各コンテナに関連付けが設定されている。この場合、コンテナAとコンテナB、そしてコンテナCとコンテナDがリンクによって関連付けされている。

【0183】

従って、コンテナA、Bが集合1、コンテナC、Dが集合2となる。即ち、リンクによって接続されたコンテナ群を一つの集合として特定する。前述したように、1101はアンカー、1102は固定された辺、1103はコントローラ、1104は可変の辺の変化方向を示している矢印、1105は可変の辺、1106はリンク、そして、1107はスライダーを示している。

【0184】

次に、レイアウト編集アプリケーション121は、ステップS901で設定したコンテナの集合から、レイアウトを計算するために一つを選択する(ステップS902)。そして、選択したコンテナの集合について、レイアウトの計算を行う。

10

20

30

40

50

【0185】

まず、選択したコンテナの集合に含まれる可変要素である2つのコンテナA、Bについて、流し込まれるデータの画像サイズもしくはテキスト量から各コンテナがなにも制約を受けない場合の大きさを計算する。

【0186】

具体的には、レイアウト編集アプリケーション121は、コンテナAが画像データ用コンテナであるか、テキスト用コンテナであるかを判定する。この判定は、前述したように、コンテナに対して設定されている属性により判定できる。

【0187】

次に、レイアウト編集アプリケーション121は、コンテナAに流し込まれるデータを読み込む。そして、コンテナAが画像データ用コンテナである場合は、その画像データのサイズ（幅、高さのピクセル数、及び解像度）がコンテナAの制約を受けない場合の大きさになる。

10

【0188】

また、コンテナAがテキスト用コンテナである場合は、そのテキストデータも文字数と、コンテナAのコンテナ属性で指定されている文字属性に基づいて、コンテナAに流し込まれるべきデータ量を計算できる。この文字属性には、例えば、フォントタイプ、フォントサイズ、文字ピッチ、行ピッチ等がある。

【0189】

ここで、テキスト用コンテナの場合は、コンテナAの縦横比が制約を考えないと決定できないため、制約を当てはめる。図10の例では、コンテナAは、左上及び左下の角にアンカーが設定されているため、高さ（縦方向）が固定となる。よって、レイアウト編集アプリケーション121は、コンテナAの基本パターンとして設定されている幅（横方向）のコンテナAに、計算したデータ量（テキスト量）の文字を流し込めるか否かを判定する。

20

【0190】

すべての文字を流し込めると判定された場合は、コンテナAは、基本パターンで設定されているサイズ（幅、高さ）に変更はない。また、すべての文字を流し込めないと判定された場合は、コンテナAは、アンカー設定により高さが固定であるため、横方向に伸びることになる。ここで、レイアウト編集アプリケーション121は、コンテナAの幅がどれだけになると、計算したデータ量の文字を流し込めるかを計算し、コンテナAのサイズを算出する。

30

【0191】

次に、レイアウト編集アプリケーション121は、レイアウトされるコンテナのサイズが、実際のコンテンツのサイズとできる限り差が少なくなるように、レイアウトの最適化を行う（ステップS903）。

【0192】

レイアウトの最適化は、動的にサイズを変化することが可能なように関連付けられたコンテナにおいて、それぞれに挿入されるコンテンツのサイズとレイアウトされるサイズとの差が、できる限り同じになるように行われる。

40

【0193】

レイアウト編集アプリケーション121は、ステップS902で算出したコンテナの集合のサイズ、つまり、コンテナAとコンテナBとリンク1106（ここでは固定リンク）の合計サイズを計算する。そして、この合計サイズと、基本レイアウトにおける当該コンテナの集合のサイズ（図10の例ではコンテナAとコンテナBのそれぞれのアンカーアイコンの距離に相当する）との差を計算する。コンテナAやコンテナBの幅が大きくなると、前ステップで計算されている場合は、差分値が発生する。レイアウト編集アプリケーション121は、この差分値をコンテナの集合の各要素に均等に分配することでレイアウト調整を行う。

【0194】

50

次に、レイアウト編集アプリケーション121は、レイアウトの最適化を行い、そのレイアウトがルールに違反しているか否かを判定する(ステップS904)。ルールに違反していない場合(ステップS904でYES)、ステップS905に進む。一方、ルールに違反している場合(ステップS904でNO)、ステップS903に戻り、再度ルールを違反しないように計算をする。

【0195】

ここでルールとは、レイアウト作成時にユーザによって設定される制限であり、コンテナのサイズの可変範囲や位置の制限、可変リンクの場合はリンクの長さの変化の制限などである。ルールを違反しないようにレイアウトが計算されたら、その集合のレイアウトは完成される。

10

【0196】

そして、ステップS902~ステップS904の処理をページ上のすべての集合について施し、レイアウト編集アプリケーション121は、ページ全体のレイアウトの計算が終了したか否かを判定する(ステップS905)。計算が終了していない場合(ステップS905でNO)、ステップS902に戻る。一方、計算が終了している場合(ステップS905でYES)、処理を終了する。

【0197】

次に、上述のレイアウト計算時のUI例について、図11を用いて説明する。

【0198】

図11は本発明の実施形態1のレイアウト計算処理におけるユーザインタフェースの一例を示す図である。

20

【0199】

図11(A)は、あるレコードが挿入されレイアウトが決定されている状態を表している。1001と1002はアンカー、1003と1004は固定された辺、1005は可変の辺、1006は可変の辺の変化方向を示している矢印、1008はリンクをそれぞれ示している。この状態において、レコードを変更し、異なったサイズのコンテンツを挿入する。

【0200】

図11(B)は、図11(A)の状態に新しいコンテンツのサイズを重ねて示している。1009はそれぞれのコンテナに挿入されるコンテンツのサイズを表している。そして、レイアウト計算が行われる。

30

【0201】

図11(C)は、レイアウト計算された結果を示している。計算後の各コンテナのサイズは、実際に挿入されるコンテンツのサイズと同等に差異があるように計算され、且つ前述したルールを違反しないように計算される。図11(C)で示されるように、図11(B)で示した挿入されるコンテンツサイズ1009と計算後のコンテンツサイズ(1010は、双方において同等な差異がある。

【0202】

[可変リンクの設定]

図12は本発明の実施形態1の可変リンクの設定のためのユーザインタフェースの一例を示す図である。

40

【0203】

図12では、図4と同様に、アプリケーションウィンドウ301とツールバー303が構成されている。図12の状態では、ドキュメントテンプレート309上にコンテナ1203とコンテナ1204が存在する。それぞれのコンテナはアンカーアイコン1201、アンカーアイコン1202と固定された辺1205、辺1206を含んで構成されている。

【0204】

コンテナ1203と1204の間には、可変サイズのリンク1209があり、コンテナ1203とコンテナ1204を結んでいる。コンテナ1203とコンテナ1204の間に

50

は、リンク 1209 が設定されているのでそれぞれの右辺 1207 と左辺 1208 は点線で表現されている。このため、各コンテナにインジケータ 1210、インジケータ 1211 が表示され、それぞれ辺 1207 と辺 1208 が可変であることを示している。

【0205】

図 13 は本発明の実施形態 1 のリンク設定機能を実現するユーザインタフェースの一例を示す図である。

【0206】

ここでは、リンク 1209 の情報（リンクプロパティ）をセットするためのユーザインタフェースであるダイアログウインドウ 1401 の例を示している。このダイアログウインドウ 1401 は、タイトルバー 1402、ツールボタン 1403、ダイアログウインドウ 1401 の開閉を行うボタン 1404、各種の情報をセットするエリア 1409 で構成されている。

10

【0207】

このダイアログウインドウ 1401 では、リンクタイプが可変長（1407）のリンクであるか、あるいは固定長（1406）のリンクであるかの択一的な選択を行えるラジオボタンからなるリンクタイプフィールド 1405 が構成されている。

【0208】

リンクタイプが可変の場合には、リンクの長さの最小値フィールド 1410（最小距離）、最大値フィールド 1412（最大距離）、ならびに基準値フィールド 1411（距離）からなるリンク距離フィールド 1408 が構成されている。

20

【0209】

図 13 のダイアログウインドウ 1401 は、例えば、図 6 及び図 7 で説明したリンクの設定操作によって 2 つのコンテナ間にリンクを設定した後に、この設定されたリンクをクリック等の操作によって選択したときに表示される。あるいは、リンクを設定した直後に、当該リンクに関するダイアログウインドウ 1401 が自動的に表示されるようにしてもよい。ここで、各コンテナ間の距離の基準値は、データを流し込んだ際に各コンテナのサイズが変更されない場合に用いられるリンクの長さである。

【0210】

次に、可変リンクの設定方法について、図 14 を用いて説明する。

【0211】

図 14 は本発明の実施形態 1 の可変リンクの設定処理を示すフローチャートである。

30

【0212】

例えば、図 10 のコンテナ A とコンテナ B の間に、図 6 及び図 7 で説明したリンクの設定操作によって、2 つのコンテナ間にリンクを設定すると、まず、固定サイズのリンクが設定される。そして、このリンクを選択して、図 14 に示す処理を実行することにより、当該リンクを固定サイズのリンク 1106（図 10）の状態から可変サイズのリンク 1209（図 12）へと変更することができる。

【0213】

まず、マウス 133 により所望のリンク（例えば、図 10 のリンク 1106）を選択状態とする（ステップ S1302）。次に、リンクプロパティを表示させるための所定の操作が行われる。これにより、レイアウト編集アプリケーション 121 のユーザインタフェース 103 は、選択状態のリンク（以下、対象リンクという）に対応したダイアログウインドウ 1401（図 13）を表示する（ステップ S1303）。

40

【0214】

尚、リンクの選択操作としては、コンテナの基本パターンの設定時と同様に、マウス 133 の右クリックあるいはキーボード 132 の特定のキーの操作等、いかなるものであってもよい。

【0215】

表示されるダイアログウインドウ 1401 には、選択されたリンクの現在の状態が示される。本例では、リンク 1106 が選択されたので、この段階ではリンクサイズは固定で

50

あり、リンクタイプフィールド 1405 においては、固定長を示す距離 1406 が選択されている。

【0216】

このダイアログウインドウ 1401 において、リンクを固定サイズから可変サイズに変更するために、リンクタイプフィールド 1405 において、リンクサイズを可変に設定するために、可変長ボタン 1407 を選択する（ステップ S1304）。

【0217】

これにより、リンク距離フィールド 1408 内に配置されている最大距離フィールド 1412、最小距離フィールド 1410、基準値フィールド 1411 が有効になり、数値の設定が可能となる。ユーザは、リンクの可変サイズを設定するために、そのリンクの長さの最大値を最大距離フィールド 1412 に、最小値を最小距離フィールド 1410 に、現在の値を基準値フィールド 1411 に設定する（ステップ S1305）。

10

【0218】

設定を終えると、ユーザは一般的なダイアログウインドウ開閉ボタン 1404 によって当該設定の適用を指示する。ユーザインタフェース 103 は、この指示を検出すると、当該対象リンクに上記設定状態を反映させる。この結果、図 12 のリンク 1209 に示すような状態にリンクの UI 表示を変更する（ステップ S1306）。

【0219】

以上のダイアログウインドウ 1401 の設定情報は、例えば、メモリ 136 に格納される。

20

【0220】

次に、レイアウト結果の一例について、図 15 及び図 16 を用いて説明する。

【0221】

図 15 は本発明の実施形態 1 の固定サイズのリンクを使用した場合のレイアウト結果を示す図である。尚、コンテナの属性に基づきコンテナにコンテンツが流し込まれたレイアウト結果を表示するためのデータは、メモリまたはハードディスク等の記憶装置に記憶されている。

【0222】

レイアウト計算方法は、前述したとおりに従って行われる。例えば、図 15 において、コンテナ 1203 とコンテナ 1204 にそれぞれ違ったサイズのイメージデータが挿入された場合を考える。この場合、それぞれのコンテナはデータの大きさを最適とみなし、コンテナ 1203 は挿入されたイメージサイズになる枠 1504（最適コンテナサイズ）に近づこうと右方向へサイズを変更しようとする。同様に、コンテナ 1204 は挿入されたイメージサイズになる枠 1505（最適コンテナサイズ）に近づこうと左方向へサイズを変更しようとする。

30

【0223】

しかし、コンテナ 1203 とコンテナ 1204 はアンカーアイコン 1201 とアンカーアイコン 1202 によってそれぞれ左辺 1212 と右辺 1213 の移動ができないため、上記のようにサイズを変更しようとする両者の間隔を狭めるしかない。しかしながら、コンテナ 1203 と 1204 間には固定サイズのリンク 1503 が設定されており、レイアウト計算時にその長さが維持されるため、コンテナ 1203 とコンテナ 1204 のサイズが変更されることになる。

40

【0224】

その結果、コンテナ 1203 とコンテナ 1204 は、データの縦横比に合わせた最適なサイズを確保することができず、最終的に、図 15 に示すように、最適なサイズ（枠 1504、枠 1505）よりも小さくなってしまふ。即ち、リンク 1503 のサイズが固定であるため、コンテナ 1203 とコンテナ 1204 は最適サイズを達成できない（図 15 において、各コンテナ内の一点差線で示した範囲がデータの持つ縦横比である）。

【0225】

一方、図 16 は図 15 と同様の状態でリンクを可変サイズにした場合を示している。

50

【0226】

この場合、上記の例で、コンテナ1203とコンテナ1204の間には図示の通り可変サイズのリンク1603が設定されている。従って、コンテナ1203とコンテナ1204のサイズが変更される際には、リンクサイズが縮まることでコンテナ1203とコンテナ1204のサイズを、図15の例より大きくすることができる。

【0227】

この結果、挿入されるデータサイズに合わせた最適なサイズを達成できる、あるいはより挿入データサイズ(最適サイズ)に近いコンテナの枠を設定することができる。図16はこの結果を示しており、図12の可変リンク1209はレイアウト計算の結果、可変リンク1603に示されるようなサイズ状態となる。尚、この場合、コンテナ1203とコンテナ1204はそれぞれ最適なサイズ(データサイズに合った大きさ)になっている。

10

【0228】

次に、実施形態1の特徴的な構成について、以下、説明する。

【0229】

実施形態1では、ドキュメントテンプレート(単に、テンプレートとも呼ぶ)に定義されたコンテナ(データが流し込まれる部分領域:フィールド領域とも呼ばれる)に従って、データベースのレコード等を配置した文書を作成する。

【0230】

また、実施形態1では、特に、データをテンプレートに従って配置するのみならず、テンプレートに従ってドキュメントの一部を作成する。そして、そのドキュメントの一部を、別途テンプレートに定義された領域(フローエリアと呼ぶ)に挿入してドキュメントを完成させる機能(マルチレコード機能)を有している。挿入できるドキュメントの一部の数は、ひとつのフローエリアについて一つとは限らない。さらに、ドキュメントテンプレートに複数のフローエリアが定義されている場合、一定の条件に従って、作成済みのドキュメントの一部をフローエリア毎に振り分けることも可能である。

20

【0231】

次に、マルチレコード機能について、図17Aを用いて説明する。

【0232】

<マルチレコード(マルチデータ)機能>

図17Aは本発明の実施形態1のマルチレコード機能を説明するための図である。

30

【0233】

通常のドキュメントテンプレート(図2)では、1レコードが1ドキュメントに対応しているが、サブテンプレートに対応するマルチレコードは、基準となる項目が変化するブレイク(切れ目)が発生するまでのレコードが流し込まれるものである。そのため、1ドキュメント内に複数のレコードが入ることになる。よって、通常のドキュメントテンプレートに対応するバリエブルデータのデータベースと、サブテンプレートに対応するバリエブルデータのマルチレコードのデータベースとは異なるデータベースとすることが望ましい。

【0234】

図17Aは、マルチレコードとそのレイアウト方法の概要を示している。データベース11204は、ドキュメント11201、ページ11202、サブテンプレート11203、データベース11204を含む。図17Aでは、データベース11204には、レコード1~7が含まれている。1レコードを1ドキュメントにレイアウトする通常のバリエブルプリントでは、7つレコードがあるので、この場合、7つのドキュメントが作成される。

40

【0235】

一方、多レコード(マルチレコード)を1ドキュメントにレイアウトするマルチレコード機能をサポートするバリエブルプリントでは、次のような処理を行う。まず、マルチレコードを指定する基準の項目であるカラム(フィールド)をユーザが指定しておく。この指定されるカラムをキーカラムと呼ぶ。

50

【0236】

尚、キーカラムをマルチデータ条件と呼ぶ場合もある。この例では、フィールド名：Nameがキーカラムとして指定されたものとする。すると、その指定されたフィールドの値が同じレコード群は、1ドキュメントにレイアウトされるレコード群となる。

【0237】

この例では、レコード1～4はキーカラムであるNameフィールドの値が同一の「Tom」であるため、ひとつのドキュメントにレイアウトされる。レコード5～7は、Nameフィールドの値が同一の「Nancy」であるため、ブレイクが発生し、もうひとつのドキュメントにレイアウトされる(図17A参照)。もちろん、マルチレコードが指定されたフィールドの値が異なるレコードは、別々のドキュメントにレイアウトされる。

10

【0238】

図17Aのドキュメントのページ11202が、それぞれ異なるレイアウトとして作成されているのは、ドキュメント毎にマルチレコードの数が異なるためである。マルチレコードの数により、どのようにサブテンプレートがレイアウトされるかは、例えば、フローエリアの属性情報に基づくことになる。例えば、4つのサブテンプレートの場合は、図17Aの上図のようにレイアウトされ、3つのサブテンプレートの場合は、図17Aの下図のようにレイアウトされることになる。

【0239】

ここで、ドキュメントテンプレートに対して、サブテンプレートをフローエリアに配置すると、サブテンプレートのサイズによっては、フローエリア内に無駄なスペースができる場合があり、見た目が悪くなるケースが想定される。

20

【0240】

特に、従来では、サブテンプレートのサイズが配置候補のフローエリアのサイズを超えた場合、サブテンプレートの回り込み(空き領域への移動)が発生し、ユーザの意図しない出力ページ数の増加や余白の挿入が生じていた。また、意図しない出力結果を防ぐには、予めフローエリアのサイズが不足する自体を想定したサブテンプレートを準備する必要があった。

【0241】

そこで、本発明の実施形態1では、サブテンプレートのサイズに対してフローエリアのサイズが不足している場合に、フローエリアのサイズに合わせてサブテンプレートに定義されたデータのレイアウトを変更する構成を実現する。

30

【0242】

まず、実施形態1のサブテンプレートのレイアウト処理について、図17Bを用いて説明する。

【0243】

図17Bは本発明の実施形態1のサブテンプレートのレイアウト処理を示すフローチャートである。また、図18は本発明の実施形態1のドキュメントテンプレートのフローエリアに設定するサブテンプレート例を示す図である。図19は本発明の実施形態1の図18のサブテンプレートにバリアブルデータを挿入して、ドキュメントテンプレートのフローエリアにサブテンプレートを配置した配置例を示す図である。

40

【0244】

以下、図18、図19の具体例を交えて、図17Bのフローチャートの説明を行う。

【0245】

図18では、イメージコンテナ1801とテキストコンテナ1802をリンク1803で関連付けた単純なサブテンプレートを示している。サブテンプレートは、その中にイメージコンテナやテキストコンテナ等を配置して、その配置されたコンテナに対して、データベース上のどのデータ(バリアブルデータ)を割り当てるかを設定した状態で保存されるテンプレートデータである。データ自体は通常のドキュメントテンプレートと特に相違する点はない。

【0246】

50

サブテンプレートは、同じレイアウトを一つのドキュメントで数多く使う場合や、他ドキュメントで流用する場合に便利である。サブテンプレート内に配置されるイメージコンテナ、テキストコンテナ間にはリンクを設定することが可能である。従って、レイアウトするデータによって、レイアウトサイズ（位置も）を最適に変化させることが可能である。

【0247】

また、サブテンプレートは、ドキュメントテンプレート上に定義されるフローエリアにのみ連続して配置することができるテンプレートであり、同じレイアウトを一つのドキュメントテンプレート上で良く使う場合に使用される。

【0248】

サブテンプレート内に配置されるコンテナは、ドキュメントテンプレート上に直接配置されるイメージコンテナ、テキストコンテナと同様である。そして、図15、16で説明したことと同様に、レイアウトするデータによって、レイアウトサイズを動的に変化させることが可能である。

【0249】

サブテンプレートへのバリエブルデータのレイアウトフローは、図17BのステップS1701からステップS1706に示される。

【0250】

まず、ステップS1701で、レイアウト編集アプリケーション121は、レイアウトするデータを取得する。次に、ステップS1702で、レイアウト編集アプリケーション121は、サブテンプレートを構成する各コンテナにレイアウトするデータのサイズを取得する。次に、ステップS1703で、レイアウト編集アプリケーション121は、取得したデータサイズに基づいて、各コンテナにおいてレイアウト計算を行う。

【0251】

そして、ステップS1704で、レイアウト編集アプリケーション121は、サブテンプレート内のコンテナのレイアウトの最適化を行う。レイアウトの最適化の手法に関しては、上述の図8、図9と同様である。ステップS1705で、レイアウト編集アプリケーション121は、ステップS1704で最適化されたレイアウトを決定する。ステップS1706で、レイアウト編集アプリケーション121は、最適化されたレイアウトの各コンテナに対してバリエブルデータを挿入する。

【0252】

次に、ステップS1707以降のフローで、フローエリアに対してサブテンプレートを流し込む処理を実行する。

【0253】

ステップS1707で、レイアウト編集アプリケーション121は、まず、サブテンプレートの流し込み先であるフローエリアの現在残っているサイズ（配置領域サイズ）を取得する。ここで、取得するサイズ情報は、フローエリアの幅及び高さである。

【0254】

ステップS1708で、レイアウト編集アプリケーション121は、ステップS1706でバリエブルデータを各コンテナに挿入したサブテンプレート（バリエブルデータ挿入後のサブテンプレート）のサイズを取得する。ここで、取得するサイズ情報は、バリエブルデータ挿入後のサブテンプレートの幅及び高さである。

【0255】

ステップS1709で、レイアウト編集アプリケーション121は、ステップS1707で取得したフローエリアのサイズとステップS1708で取得したサブテンプレートのサイズの比較を行う。このとき、サブテンプレートのサイズ（幅もしくは高さ）がフローエリアのサイズ以下である場合（ステップS1709でYES）、レイアウト編集アプリケーション121は、フローエリアにサブテンプレートを流し込むことが可能であると判定する。そして、ステップS1710で、レイアウト編集アプリケーション121は、フローエリアに対してサブテンプレートを流し込み、レイアウト結果をドキュメントに配置

10

20

30

40

50

する。

【0256】

一方、ステップS1709で、サブテンプレートのサイズがフローエリアのサイズより大きい場合（ステップS1709でNO）、レイアウト編集アプリケーション121は、フローエリアにサブテンプレートを流し込むことが可能でないと判定する。そして、ステップS1711で、レイアウト編集アプリケーション121は、フローエリアに収まるサイズへのサブテンプレートのレイアウト構造の再定義を行うか否かを判定する。換言すれば、サブテンプレートのレイアウト構造の再定義の可否を判定する。

【0257】

特に、本発明では、サブテンプレートのレイアウト構造の再定義を行うか否かの設定を指定するためのユーザインタフェース（図28）を構成している。このユーザインタフェースの詳細に関しては後述するが、このユーザインタフェースでは、サブテンプレートのレイアウト構造の再定義の可否の設定を行うことができる。

10

【0258】

そのため、ステップS1711で、レイアウト編集アプリケーション121は、サブテンプレートのレイアウト構造の再定義を許可するか否かを判定する。この判定は、図28のユーザインタフェースで設定される設定情報に基づいて、フローエリアに合わせたサブテンプレートのレイアウト構造の再定義設定が指定されているか否かで判定する。

【0259】

ステップS1711において、サブテンプレートのレイアウト構造の再定義が許可されていない場合（ステップS1711でNO）、ステップS1714に進む。ステップS1714で、レイアウト編集アプリケーション121は、フローエリア内でサブテンプレートを配置することのできる位置までサブテンプレートを移動する回り込み処理を行う。

20

【0260】

尚、フローエリア内でのサブテンプレートの回り込み処理に関しては後述する。

【0261】

一方、サブテンプレートのレイアウト構造の再定義が許可されている場合（ステップS1711でYES）、レイアウト編集アプリケーション121は、ステップS1712で、サブテンプレートのレイアウト構造の再定義を行う。尚、サブテンプレートのレイアウト構造の再定義に関しては後述する。

30

【0262】

次に、ステップS1713で、レイアウト編集アプリケーション121は、ステップS1712におけるフローエリアのサイズに収まるサブテンプレートのレイアウト構造の再定義の結果を判定する。この判定は、レイアウト構造の再定義が成功したか否かを判定する。そして、サブテンプレートのレイアウト構造の再定義が成功した場合（ステップS1713でYES）、ステップS1710に進む。ステップS1710で、レイアウト編集アプリケーション121は、再定義されたフローエリアに対してサブテンプレートを流し込み、レイアウト結果をドキュメントに配置する。

【0263】

一方、ステップS1713において、サブテンプレートのレイアウト構造の再定義に失敗した場合（ステップS1713でNO）、ステップS1714に進む。ステップS1714で、レイアウト編集アプリケーション121は、フローエリア内でサブテンプレートを配置することのできる位置までサブテンプレートを移動する回り込み処理を行う。

40

【0264】

次に、サブテンプレートをフローエリアに流し込む処理の具体例を図19を用いて説明する。

【0265】

図19では、ドキュメント1901上のフローエリア1902内に対して2個のバリアブルデータが図18に示すレイアウト構造を持つサブテンプレート1903、1907を紹介して正常に流し込まれる例を示している。

50

【0266】

バリエブルデータは、イメージコンテナ1904とテキストコンテナ1906がリンク1905で接続されたサブテンプレート1903に流し込まれる。また、バリエブルデータは、イメージコンテナ1908とテキストコンテナ1910がリンク1909で接続されたサブテンプレート1907に流し込まれる。

【0267】

サブテンプレート1903のサイズとサブテンプレート1907のサイズとの和がフローエリア1902よりも小さいため、サブテンプレート1903と1907はドキュメント1901上に水平方向に並んで配置される。

【0268】

図20は、図19の状態のドキュメント1901上にさらにもう1個のサブテンプレートを配置する場合を示している。ドキュメント1901上では、既にフローエリア1902が定義されている。ここに、新たに、イメージコンテナ2004、テキストコンテナ2006及び両者を接続するリンク2005が定義されているサブテンプレート2003を配置する場合には、通常、フローエリア2007に流し込まれることになる。

【0269】

但し、この場合、サブテンプレート2003の高さがフローエリア2007の高さを超えているために、サブテンプレート2003はドキュメント1901上に配置することはできない。そして、従来の構成では、このような状況では、サブテンプレートの回り込み処理を行う。このサブテンプレートの回り込み処理に関しては後述する。

【0270】

これに対し、本発明では、図17BのステップS1711において、フローエリア2007に収まらないサブテンプレート2003のレイアウト構造を再定義するか否かを判定する。そして、その判定結果に基づいて、レイアウト構造を再定義する。これにより、より適切なレイアウト構造のサブテンプレート配置を行うことが可能となる。

【0271】

以下、このサブテンプレートのレイアウト構造の再定義処理(ステップS1712)について、図21を用いて説明する。

【0272】

図21は本発明の実施形態1のサブテンプレートのレイアウト構造の再定義処理を示すフローチャートである。

【0273】

まず、ステップS2101で、レイアウト編集アプリケーション121は、サブテンプレートのレイアウト構造の再定義を行うことができるか否かを判定する。サブテンプレートのレイアウト構造の再定義を行うことができる場合(ステップS2101でYES)、ステップS2102に進む。一方、サブテンプレートのレイアウト構造の再定義を行うことができない場合(ステップS2101でNO)、サブテンプレートのレイアウト構造を再定義前に戻して、処理を終了する。レイアウト構造の再定義を行うことができない場合の具体例は、図25で後述する。特に、図25に示す例は、サブテンプレートのレイアウト構造の再定義に失敗した例を示している。

【0274】

ステップS2102で、レイアウト編集アプリケーション121は、レイアウトの再定義対象のコンテナを判別する。

【0275】

図20の例では、レイアウト編集アプリケーション121は、サブテンプレート2003のサイズがフローエリア2007のサイズ(高さ)を上回る原因となるコンテナをサブテンプレート2003のレイアウト構造から判別する。この場合、サブテンプレート2003の高さがフローエリア2007を上回る原因は、テキストコンテナ2006であることが分かる。そのため、このテキストコンテナ2006が再定義対象のコンテナと判別する。

10

20

30

40

50

【0276】

そして、実施形態1では、ステップS2102で、既に定義済みのサブテンプレート2003のサイズをフローエリア2007のサイズに合うようにテキストコンテナ2006のレイアウト構造を再定義する。但し、テキストコンテナ2006には、リンク2005を介してイメージコンテナ2004が接続されている。この場合は、テキストコンテナ2006に接続されている全てのリンク、コンテナを含めてレイアウト構造の再定義対象となる。

【0277】

次に、ステップS2103で、レイアウト編集アプリケーション121は、レイアウト構造の再定義対象のコンテナからリンクを解除する。

10

【0278】

図20の例では、レイアウト構造の再定義対象となるテキストコンテナ2006からリンク2005を解除する。これによって、リンク2005が無効になるため、同様にイメージコンテナ2004からリンク2005を解除する。

【0279】

次に、ステップS2104で、レイアウト編集アプリケーション121は、リンクを解除した再定義対象のコンテナをフローエリア内で再配置する。

【0280】

図20の例では、リンク2005を解除したイメージコンテナ2004とテキストコンテナ2006をフローエリア2007内に再配置する。

20

【0281】

このステップS2104における、コンテナの再配置の過程を図22を用いて説明する。

【0282】

図22は本発明の実施形態1のコンテナの再配置の過程を説明するための図である。

【0283】

図22では、まず、フローエリア2007に対して、イメージコンテナ2004を配置する。次に、レイアウト構造の再定義対象であるテキストコンテナ006をフローエリア2007の空いている領域に配置する。そして、この配置によって、図20のコンテナの配置は、図23に示すような配置となる。つまり、図23は、図20の状態のドキュメント1901上のサブテンプレート2003に対する、ステップS2104によるレイアウト構造の再定義結果を示している。

30

【0284】

つまり、ステップS2104の処理で、ドキュメント1901上のフローエリアに、バリアブルデータが挿入されたイメージコンテナ2004、テキストコンテナ2006が本来定義されていたサブテンプレート2003と異なるレイアウト構造で配置される。

【0285】

これによって、サブテンプレート2003のサイズがフローエリア2303のサイズに合う形のサブテンプレート2304に変形され、ドキュメント1901上に3レコード分のバリアブルデータを適切に流し込むことが可能となる

40

尚、ステップS2104に対し、フローエリア内のどの位置からコンテナを配置するか、フローエリア内にコンテナをどの順番で配置するかの設定は、後述するユーザインタフェース(図26)で行う。

【0286】

図21の説明に戻る。

【0287】

ステップS2105で、レイアウト編集アプリケーション121は、ステップS2104の処理によるコンテナのレイアウト構造の再定義の結果、サブテンプレートがフローエリアに収まるか否かを判定する。フローエリアに収まる場合(ステップS2105でYES)、ステップS2106に進む。図23に示す例では、サブテンプレート2304がフ

50

ローエリア 2007 に収まる場合の例となる。

【0288】

ステップ S 2106 で、レイアウト編集アプリケーション 121 は、再定義されたレイアウトの各コンテナに対してバリアブルデータを挿入する。

【0289】

一方、フローエリアに収まらない場合（ステップ S 2105 で NO）、ステップ S 2101 に戻り、再度、レイアウト構造の再定義を行うことができるか否かを判定する。

【0290】

次に、図 17B のステップ S 1714 の回り込み処理の具体例について、図 24 を用いて説明する。

【0291】

図 24 は本発明の実施形態 1 のフローエリア内でのサブテンプレートの回り込み処理を説明するための図である。

【0292】

上述のように、フローエリアのサイズに対して、サブテンプレートのサイズが大きく、かつサブテンプレートのレイアウト構造の再定義が許可されていない場合には、フローエリア内でサブテンプレートの回り込み処理を行う。

【0293】

この例では、ドキュメント 2401 に対して、サブテンプレート 2403 はサブテンプレート 2402 の隣には配置することはできない。そこで、この場合、2 種類のサブテンプレートをドキュメント 2401 内の空き領域へ移動する回り込み処理を試行する。

【0294】

サブテンプレート 2402 を配置したドキュメント 2401 のフローエリア中にサブテンプレート 2403 のサイズよりも大きい空き領域を有する場合は、その空き領域にサブテンプレート 2403 を配置する。

【0295】

また、サブテンプレート 2402 を配置したドキュメント 2401 のフローエリア中の空き領域がサブテンプレート 2403 のサイズよりも小さい場合は、新たにドキュメント 2404 を作成する。そして、そのドキュメント 2404 上のフローエリアにサブテンプレート 2403 を配置する。

【0296】

このように、回り込み処理では、フローエリア中の空き領域に基づいて、その空き領域に配置対象のサブテンプレートを配置する、あるいは新規に作成したドキュメントテンプレート上の新規のフローエリアに配置対象のサブテンプレートを配置する。

【0297】

次に、図 21 のステップ S 2101 で、サブテンプレートのレイアウト構造の再定義を行えない（再定義の失敗）場合の具体例について、図 25 を用いて説明する。

【0298】

図 25 は本発明の実施形態 1 のサブテンプレートのレイアウト構造の再定義を行えない（再定義の失敗）場合の具体例を説明するための図である。

【0299】

図 25 に示すように図 20 の状態のサブテンプレート 2003 に対してレイアウト構造の再定義を行った場合に、サブテンプレート 2504 が得られたとする。ここで、サブテンプレート 2504 のサイズは、フローエリア 2007 のサイズより大きくなっている。そのため、サブテンプレート 2504 は、これ以上のレイアウト構造の再定義を行うことができない。

【0300】

この場合は、図 20 のサブテンプレート 2003 は、レイアウト構造の再定義を行うことができないものと判定し、図 25 のサブテンプレート 2504 のレイアウト構造を図 20 のサブテンプレート 2003 に戻す。そして、この場合には、ステップ S 1714 で、

10

20

30

40

50

フローエリア内でサブテンプレートを配置することのできる位置までサブテンプレートの回り込み処理を行う。

【0301】

次に、レイアウト構造の再定義を行う場合における、サブテンプレート内でのコンテナのフロー（流し込み）設定を行うためのユーザインタフェース例について、図26を用いて説明する。

【0302】

図26は本発明の実施形態1のコンテナのフロー設定を行うためのユーザインタフェース例を示す図である。

【0303】

このユーザインタフェース2601では、構成されている設定項目（ラジオボタン）を用いて、ドキュメント上のフローエリアに対して、設定した回り込み方向に従って、フローエリア中の空き領域に対してコンテナを配置する設定（配置ルール）が可能である。

【0304】

図26では、ラジオボタン2602から2606のいずれか1つを選択することで、サブテンプレート内のコンテナのフロー方向を設定する。

【0305】

ラジオボタン2602は、サブテンプレートの上端を起点に、フローエリアに対して上から下の縦方向にコンテナを配置するためのボタンである。

【0306】

ラジオボタン2603は、サブテンプレートの上左端を起点にサブテンプレートに対して左から右の横方向にコンテナを配置し、横方向にコンテナが配置できなくなったら、コンテナの配置を1行下げ、再度左から右の横方向に配置するためのボタンである。

【0307】

ラジオボタン2604は、サブテンプレートの上左端を起点にサブテンプレートに対して上から下の縦方向にコンテナを配置し、縦方向にコンテナが配置できなくなったら、コンテナの配置を1列右に移動し、再度上から下の横方向に配置するためのボタンである。

【0308】

ラジオボタン2605は、サブテンプレートの上右端を起点にサブテンプレートに対して右から左の横方向にコンテナを配置し、横方向にコンテナが配置できなくなったら、コンテナの配置を1行下げ、再度右から左の横方向に配置するためのボタンである。

【0309】

ラジオボタン2606は、サブテンプレートの上右端を起点にサブテンプレートに対して上から下の縦方向にコンテナを配置し、縦方向にコンテナが配置できなくなったら、コンテナの配置を1列左に移動し、再度上から下の横方向に配置するためのボタンである。

【0310】

そして、OKボタン2607を押下すると、ユーザインタフェース2601で設定したコンテナの回り込み設定を回り込み設定情報としてメモリ136に記憶する。一方、キャンセルボタン2608を押下すると、ユーザインタフェース2601で設定した回り込み設定をキャンセルし、ユーザインタフェース2601の表示を終了する。

【0311】

次に、ユーザインタフェース2601で設定された回り込み設定情報に従う、サブテンプレート内でのコンテナのフロー例について、図27を用いて説明する

図27は本発明の実施形態1のサブテンプレート内でのコンテナのフロー例を示す図である。

【0312】

尚、図27の説明においては、フローエリア内でのサブテンプレートのフローも同様の設定及び挙動をするものとする。また、この設定及び挙動については、図26のユーザインタフェース2601と同様のフローエリア用のユーザインタフェースを構成することで実現できる。

10

20

30

40

50

【0313】

図27では、1からN個（Nは整数）のコンテナ2701～270n（コンテナ1～コンテナN）をサブテンプレート2706に、ユーザインタフェース2601で設定された回り込み設定情報に従ってフロー配置するものとする。

【0314】

サブテンプレート2707は、ユーザインタフェース2601で、ラジオボタン2602を選択した場合のフロー例である。この場合、コンテナ2701、2702、270nは上から順に縦に配置される。

【0315】

サブテンプレート2708は、ユーザインタフェース2601で、ラジオボタン2603を選択した場合のフロー例である。この場合、コンテナ2701、2702はサブテンプレート2708の左上を起点に、左から右の横方向に配置される。コンテナ2703を横方向に配置できなくなったら、コンテナ2703はサブテンプレート2708内を下方方向にフローし、再度左端を起点に配置される。コンテナ2704、270nも同様の手順で配置される。

10

【0316】

サブテンプレート2709は、ユーザインタフェース2601で、ラジオボタン2604を選択した場合のフロー例である。この場合、コンテナ2701、2702、2703はサブテンプレート2709の左上を起点に、上から下の縦方向に配置される。コンテナ2704を縦方向に配置できなくなったら、コンテナ2704はサブテンプレート2709内を右方向にフローし、再度左端を起点に配置される。コンテナ2705も同様の手順で配置される。

20

【0317】

サブテンプレート2710は、ユーザインタフェース2601で、ラジオボタン2605を選択した場合のフロー例である。コンテナ2701、2702はサブテンプレート2710の右上を起点に、右から左の横方向に配置される。コンテナ2703を横方向に配置できなくなったら、コンテナ2703はサブテンプレート2710内を下方方向にフローし、再度右端を起点に配置される。コンテナ2704、270nも同様の手順で配置される。

【0318】

サブテンプレート2711は、ユーザインタフェース2601で、ラジオボタン2606を選択した場合のフロー例である。コンテナ2701、2702、2703はサブテンプレート2711の右上を起点に、上から下の縦方向に配置される。コンテナ2704を縦方向に配置できなくなったら、コンテナ2704はサブテンプレート2711内を左方向にフローし、再度右端を起点に配置される。コンテナ2705も同様の手順で配置される。

30

【0319】

次に、フローエリアに対して、サブテンプレートのレイアウト構造の再定義設定及び、コンテナ配置の際の外部余白設定を行うためのユーザインタフェース例について、図28を用いて説明する。このユーザインタフェースで設定された設定情報は、図17BのステップS1711の判定に用いられる。

40

【0320】

図28は本発明の実施形態1のサブテンプレートのレイアウト構造の再定義設定及び、コンテナ配置の際の外部余白設定を行うためのユーザインタフェース例を示す図である。

【0321】

このユーザインタフェース2801では、構成されている設定項目（ラジオボタン、チェックボックス等）を用いて、サブテンプレートのレイアウト構造の再定義設定及び、コンテナ配置の際の外部余白設定を行うことができる。

【0322】

ラジオボタン2802及び2803は、フローエリアに対して、サブテンプレートのレ

50

レイアウト構造の再定義を行うか否かを設定するためのボタンである。ここでの設定情報に基づいて、ステップ S 1 7 1 1 でサブテンプレートのレイアウト構造の再定義を行うか、従来どおりのフローエリアの回り込み処理を行うかが決定される。

【 0 3 2 3 】

チェックボックス 2 8 0 4 は、ラジオボタン 2 8 0 3 を選択した場合のみ有効になるチェックボックスである。ここでは、サブテンプレート内で設定されているコンテナ間のリンクを解消し、再配置されたコンテナに対して外部余白を設定するか否かの設定を行うことができる。

【 0 3 2 4 】

特に、チェックボックス 2 8 0 4 を選択した場合は、テキストボックス 2 8 0 5 で上下左右の外部余白の大きさを決定する。ここでの上下左右とは、ドキュメントの天地左右（上下左右辺）に対応する。

【 0 3 2 5 】

そして、OK ボタン 2 8 0 6 を押下すると、ユーザインタフェース 2 8 0 1 で設定した設定（レイアウト構造の再定義設定及びコンテナ配置の際の外部余白設定）を設定情報としてメモリ 1 3 6 に記憶する。一方、キャンセルボタン 2 8 0 7 を押下すると、ユーザインタフェース 2 8 0 1 で設定した設定をキャンセルし、ユーザインタフェース 2 8 0 1 の表示を終了する。

【 0 3 2 6 】

以上説明したように、実施形態 1 によれば、フローエリアに収まるサイズにサブテンプレートのレイアウト構造を再調整することで、できる限りユーザが意図するレイアウトのドキュメントを作成することができる。また、ユーザの意図をより反映させるためにサブテンプレートを複数準備することなく、最適なレイアウトが実現することができる。

【 0 3 2 7 】

<< 実施形態 2 >>

上述の実施形態 1 の図 1 7 B 及び図 2 1 のフローチャートに示す処理では、サブテンプレートのレイアウト構造を一時的に再定義することで、配置候補のフローエリアのサイズに応じるようにサブテンプレートのサイズを調整することが可能である。

【 0 3 2 8 】

ここで、実施形態 1 では、図 2 6 及び図 2 8 に示すユーザインタフェースによって指定したフローエリアに対して、配置するサブテンプレートを構成するコンテナ、リンク全ての要素に対して、再定義の設定を行うものである。但し、サブテンプレートを構成するコンテナ、リンクによっては、再定義を行うことで、むしろユーザの意図が損なわれる場合も考えられる。

【 0 3 2 9 】

そこで、実施形態 2 では、図 2 9 に示すユーザインタフェースを用いて、サブテンプレートを構成するリンクを介して再定義時の挙動を制御する構成について説明する。つまり、実施形態 2 では、再定義時のリンク設定を行うためのユーザインタフェースを構成している。

【 0 3 3 0 】

図 2 9 は本発明の実施形態 2 のリンクに関する設定のためのユーザインタフェース例を示す図である。

【 0 3 3 1 】

図 2 9 では、サブテンプレートに定義されたレイアウト構造を構成するリンクに対して、レイアウト構造を一時的に再定義する際に、リンク解除の許可の設定を行うためのユーザインタフェース例を示している。つまり、構成されている設定項目（ラジオボタン）を用いて、ユーザインタフェース 2 9 0 1 では、サブテンプレートに定義されたレイアウト構造を構成するリンクに対して、レイアウト構造を一時的に再定義する際に、リンク解除の許可の設定を行うことができる。

【 0 3 3 2 】

10

20

30

40

50

ラジオボタン2902及び2903では、サブテンプレートで定義されたリンクに対して、再定義時に、再定義対象となるコンテナとのリンク解除の設定を行うためのボタンである。この設定は、サブテンプレート内で定義されているリンクに対して個別に行うことができる。

【0333】

ラジオボタン2902は、設定したリンクがサブテンプレートの再定義時に再定義対象となるコンテナと接続されている場合、そのリンクを解除するためのボタンである。

【0334】

ラジオボタン2903は、設定したリンクがサブテンプレートの再定義時に再定義対象となるコンテナと接続されている場合、そのリンクを解除しないためのボタンである。

10

【0335】

そして、OKボタン2904を押下すると、ユーザインタフェース2901で設定したリンク設定を設定情報としてメモリ136に記憶する。一方、キャンセルボタン2905を押下すると、ユーザインタフェース2901で設定したリンク設定をキャンセルし、ユーザインタフェース2901の表示を終了する。

【0336】

次に、図29のユーザインタフェース2901を用いて、リンクに関する設定に基づくサブテンプレートの配置例について、図30を用いて説明する。

【0337】

図30は本発明の実施形態2のサブテンプレートの配置例を示す図である。

20

【0338】

図30では、図29のユーザインタフェース2901を用いて、リンクに関する設定を行ったサブテンプレート3003、3004を、ドキュメント3001上のフローエリア3002に配置する例を示している。

【0339】

サブテンプレート3003には、イメージコンテナ3005及び3006と、テキストコンテナ3007がレイアウトされている。イメージコンテナ3005とイメージコンテナ3006間はリンク3008で接続されている。また、イメージコンテナ3006とテキストコンテナ3007間はリンク3009で接続されている。

【0340】

また、サブテンプレート3004には、イメージコンテナ3010及び3011と、テキストコンテナ3012がレイアウトされている。イメージコンテナ3010とイメージコンテナ3011間はリンク3013で接続されている。また、イメージコンテナ3011とテキストコンテナ3012間はリンク3014で接続されている。

30

【0341】

また、リンク3008と3013については、図29のユーザインタフェース2901のラジオボタン2902によって、再定義時にリンクを解除するための設定が行われているとする。更に、リンク3009と3014については、図29のユーザインタフェース2901のラジオボタン2902によって、再定義時にリンクを解除しないための設定が行われていないとする。

40

【0342】

この時の設定状態は、リンク3009については、イメージコンテナ3006の下にはテキストコンテナ3007を常に配置するというユーザの意図が反映されている。また、リンク3014については、イメージコンテナ3011の下にはテキストコンテナ3012を常に配置するというユーザの意図が反映されている。

【0343】

ここで、サブテンプレート3004は、コンテンツの挿入によって、イメージコンテナ3011がフローエリア3002の範囲を超える。このとき、図28のユーザインタフェース2801のラジオボタン2803（再定義設定）が選択されている場合は、ステップS1712で、イメージコンテナ3011を再定義対象とし、サブテンプレート3004

50

の再定義が行われることになる。

【0344】

ここで、実施形態1の構成に従って、図30のサブテンプレート3004を再定義を行う場合には、サブテンプレート3004内の再定義対象のイメージコンテナ3011を再配置することで、サブテンプレート3004の再定義を行うことになる。この場合、サブテンプレート3004内のリンク3013及び3014は解除され、図31に示すような構成となる。

【0345】

一方、実施形態2の構成では、リンク解除の設定の有無に基づいて、サブテンプレートの再定義を行うことになる。

10

【0346】

以下、実施形態2における、サブテンプレートのレイアウト構造の再定義処理について、図32を用いて説明する。

【0347】

図32は本発明の実施形態2のサブテンプレートのレイアウト構造の再定義処理を示すフローチャートである。

【0348】

尚、図32のフローチャートは、基本的には実施形態1の図21のフローチャートに準ずる。

【0349】

まず、ステップS3201で、レイアウト編集アプリケーション121は、サブテンプレートのレイアウト構造の再定義を行うことができるか否かを判定する。ここでの判定方法は、図21のステップS2101と同様である。

20

【0350】

サブテンプレートのレイアウト構造の再定義を行うことができる場合(ステップS3201でYES)、ステップS3202に進む。一方、サブテンプレートのレイアウト構造の再定義を行うことができない場合(ステップS3201でNO)、サブテンプレートのレイアウト構造を再定義前に戻して、処理を終了する。

【0351】

ステップS3202で、レイアウト編集アプリケーション121は、レイアウトの再定義対象のコンテナを判別する。

30

【0352】

図30の例では、レイアウト編集アプリケーション121は、サブテンプレート3004のサイズがフローエリア3002のサイズ(幅)を上回る原因となるコンテナをサブテンプレート3004のレイアウト構造から判別する。この場合、サブテンプレート3004の幅がフローエリア3002を上回る原因は、イメージコンテナ3011であることが分かる。そのため、このイメージコンテナ3011が再定義対象のコンテナと判別される。

【0353】

そして、実施形態2では、サブテンプレートのレイアウト構造の再定義対象となるコンテナを判別した後、新たに、ステップS3203を実行する。つまり、ステップS3203で、レイアウト編集アプリケーション121は、再定義対象のコンテナに接続されているリンクを解除できるか否かを判定する。これは、図29のユーザインタフェース2901で設定される設定情報に基づいて判定する。

40

【0354】

リンクを解除できない場合(ステップS3203でNO)、ステップS3204に進む。一方、リンクを解除できる場合(ステップS3203でYES)、ステップS3205に進む。

【0355】

特に、リンクを解除できる場合は、ステップS3205～ステップS3207の処理を

50

実行する。これらの処理は、図 21 のステップ S 2 1 0 4 ~ ステップ S 2 1 0 6 に対応するため、その詳細は省略する。

【 0 3 5 6 】

一方、リンクを解除できない場合は、ステップ S 3 2 0 4 で、レイアウト編集アプリケーション 1 2 1 は、再定義対象のコンテナの再判別を行う。

【 0 3 5 7 】

具体例を、図 30 を用いて説明する。

【 0 3 5 8 】

イメージコンテナ 3 0 1 1 は、リンク 3 0 1 3 によってイメージコンテナ 3 0 1 0 に接続されている。また、イメージコンテナ 3 0 1 1 はリンク 3 0 1 4 によってテキストコンテナ 3 0 1 2 と接続されている。

10

【 0 3 5 9 】

リンク 3 0 1 3 は、ラジオボタン 2 9 0 2 (図 2 9) によって、リンクを解除するに設定されているため、イメージコンテナ 3 0 1 1 との接続は解除される。また、リンク 3 0 1 4 は、ラジオボタン 2 9 0 3 (図 2 9) によって、リンクを解除しないに設定されているため、イメージコンテナ 3 0 1 1 との接続は解除されない。

【 0 3 6 0 】

このため、ステップ S 3 2 0 4 では、レイアウト編集アプリケーション 1 2 1 は、イメージコンテナ 3 0 1 1 とリンク 3 0 1 3 で接続されているテキストコンテナ 3 0 1 2 をレイアウトの再定義対象として拡大して、サブテンプレートの再定義処理を実行する。

20

【 0 3 6 1 】

この処理後、ステップ S 3 2 0 3 の判定を経て、ステップ S 3 2 0 5 では、レイアウト編集アプリケーション 1 2 1 は、イメージコンテナ 3 0 1 1 とリンク 3 0 1 4 で接続されているテキストコンテナ 3 0 1 2 をフローエリア 3 0 0 4 内に配置することになる。

【 0 3 6 2 】

この場合の配置例は、図 33 のようになる。つまり、図 33 に示されるように、イメージコンテナ 3 0 1 1 とテキストコンテナ 3 0 1 2 を接続するリンク 3 0 1 4 は解除せずに、サブテンプレート 2 3 0 4 内にフローした、レイアウト構造が新たに定義されることになる。

【 0 3 6 3 】

以上説明したように、実施形態 2 によれば、実施形態 1 で説明した効果に加えて、ユーザの意図に応じて、リンクの解除の可否を踏まえたサブテンプレートのレイアウト構造を再定義することができる。これにより、よりユーザが意図するレイアウトのドキュメントを作成することができる。

30

【 0 3 6 4 】

以上、実施形態例を詳述したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記憶媒体等としての実施態様をとることが可能である。具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

【 0 3 6 5 】

尚、本発明は、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 (実施形態では図に示すフローチャートに対応したプログラム) を、システムあるいは装置に直接あるいは遠隔から供給する。そして、そのシステムあるいは装置のコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される場合を含む。

40

【 0 3 6 6 】

従って、本発明の機能処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

【 0 3 6 7 】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタによ

50

り実行されるプログラム、OSに供給するスクリプトデータ等の形態であっても良い。

【0368】

プログラムを供給するための記録媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスクがある。また、更に、記録媒体としては、光磁気ディスク、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモリカード、ROM、DVD（DVD-ROM、DVD-R）などがある。

【0369】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続する。そして、その接続先のホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイル（10）をハードディスク等の記録媒体にダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明に含まれるものである。

【0370】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせる。そして、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能（20）である。

【0371】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される。また、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現され得る。

【0372】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれる。その後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU（30）などが実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現される。

【図面の簡単な説明】

【0373】

【図1A】本発明の実施形態1の情報処理システムの構成例を示す図である。

【図1B】本発明の実施形態1の情報処理システムの一構成要素であるホストコンピュータのハードウェア構成を示す図である。

【図1C】本発明の実施形態1の情報処理システムの別構成例を示す図である。

【図2】本発明の実施形態1のバリエブルデータプリントの概略を説明するための図である。（40）

【図3】本発明の実施形態1のユーザインタフェースの一例を示す図である。

【図4】本発明の実施形態1のユーザインタフェースにおけるコンテナの表示例を示す図である。

【図5】本発明の実施形態1のコンテナの表示ルールを説明するための図である。

【図6】本発明の実施形態1のリンクの設定処理を示すフローチャートである。

【図7】本発明の実施形態1のリンク設定時のユーザインタフェースの遷移例を示す図である。

【図8】本発明の実施形態1のレイアウト計算処理を示すフローチャートである。

【図9】本発明の実施形態1のレイアウト計算処理の詳細を示すフローチャートである。

【図10】本発明の実施形態1のレイアウト計算処理時におけるコンテナの集合を説明す（50）

るための図である。

【図 1 1】本発明の実施形態 1 のレイアウト計算処理におけるユーザインタフェースの一例を示す図である。

【図 1 2】本発明の実施形態 1 の可変リンクの設定のためのユーザインタフェースの一例を示す図である。

【図 1 3】本発明の実施形態 1 のリンク設定機能を実現するユーザインタフェースの一例を示す図である。

【図 1 4】本発明の実施形態 1 の可変リンクの設定処理を示すフローチャートである。

【図 1 5】本発明の実施形態 1 の固定サイズのリンクを使用した場合のレイアウト結果を示す図である。

【図 1 6】本発明の実施形態 1 の可変サイズのリンクを使用した場合のレイアウト結果を示す図である。

【図 1 7 A】本発明の実施形態 1 のマルチレコード機能を説明するための図である。

【図 1 7 B】本発明の実施形態 1 のサブテンプレートのレイアウト処理を示すフローチャートである。

【図 1 8】本発明の実施形態 1 のドキュメントテンプレートのフローエリアに設定するサブテンプレート例を示す図である。

【図 1 9】本発明の実施形態 1 の図 1 8 のサブテンプレートにバリアブルデータを挿入して、ドキュメントテンプレートのフローエリアにサブテンプレートを配置した配置例を示す図である。

【図 2 0】本発明の実施形態 1 のフローエリアへのサブテンプレートの配置が不可な場合の状態を示す図である。

【図 2 1】本発明の実施形態 1 のサブテンプレートのレイアウト構造の再定義処理を示すフローチャートである。

【図 2 2】本発明の実施形態 1 のコンテナの再配置の過程を説明するための図である。

【図 2 3】本発明の実施形態 1 の再定義されるサブテンプレートのレイアウト構造を示す図である。

【図 2 4】本発明の実施形態 1 のフローエリア内でのサブテンプレートの回り込み処理を説明するための図である。

【図 2 5】本発明の実施形態 1 のサブテンプレートのレイアウト構造の再定義を行えない（再定義の失敗）場合の具体例を説明するための図である。

【図 2 6】本発明の実施形態 1 のコンテナのフロー設定を行うためのユーザインタフェース例を示す図である。

【図 2 7】本発明の実施形態 1 のサブテンプレート内でのコンテナのフロー例を示す図である。

【図 2 8】本発明の実施形態 1 のサブテンプレートのレイアウト構造の再定義設定及び、コンテナ配置の際の外部余白設定を行うためのユーザインタフェース例を示す図である。

【図 2 9】本発明の実施形態 2 のリンクに関する設定のためのユーザインタフェース例を示す図である。

【図 3 0】本発明の実施形態 2 のサブテンプレートの配置例を示す図である。

【図 3 1】本発明の実施形態 2 の構成を説明するにあたり、実施形態 1 の構成を用いる場合のサブテンプレートの再定義の配置例を示す図である。

【図 3 2】本発明の実施形態 2 のサブテンプレートのレイアウト構造の再定義処理を示すフローチャートである。

【図 3 3】本発明の実施形態 2 のサブテンプレートの再定義の配置例を示す図である。

【符号の説明】

【 0 3 7 4 】

1 0 1 ホストコンピュータ

1 0 3 ユーザインタフェース

1 0 4 UIモデルアナライザ

10

20

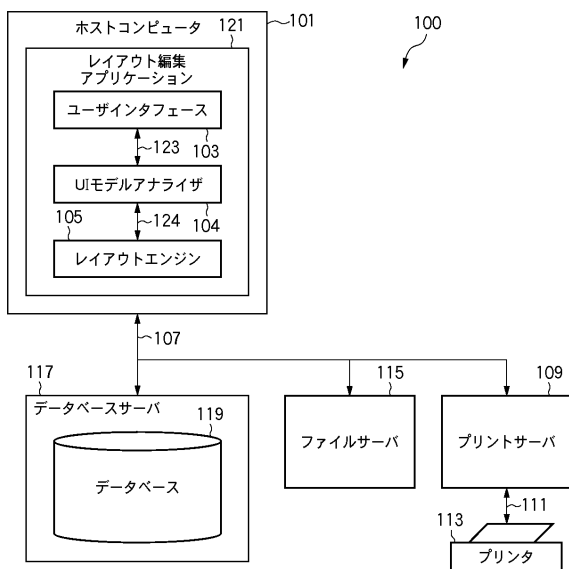
30

40

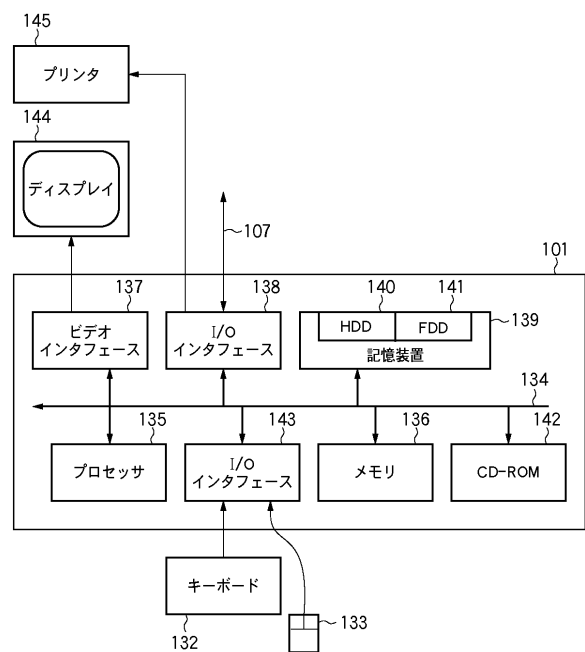
50

- 1 0 5 レイアウトエンジン
- 1 0 7 ネットワーク
- 1 0 9 プリントサーバ
- 1 1 1 通信チャンネル
- 1 1 3 プリンタ
- 1 1 5 ファイルサーバ
- 1 1 7 データベースサーバ
- 1 1 9 データベース
- 1 2 1 レイアウト編集アプリケーション
- 1 2 3、1 2 4 通信チャンネル
- 2 2 5 レイアウトエンジン

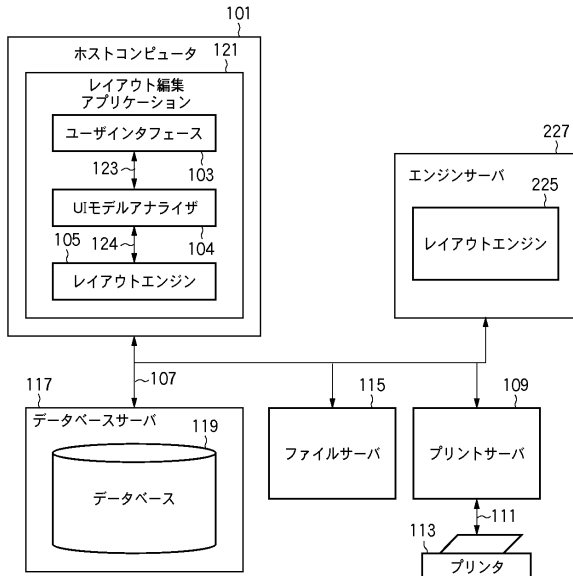
【図 1 A】



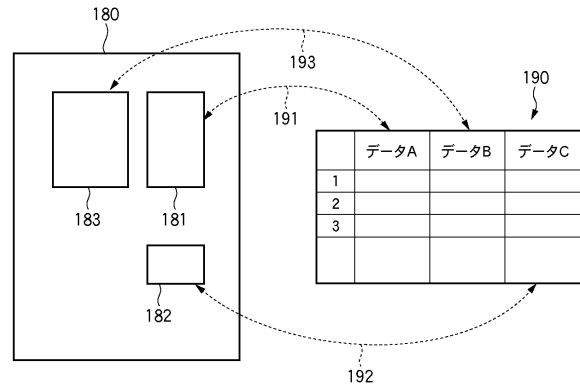
【図 1 B】



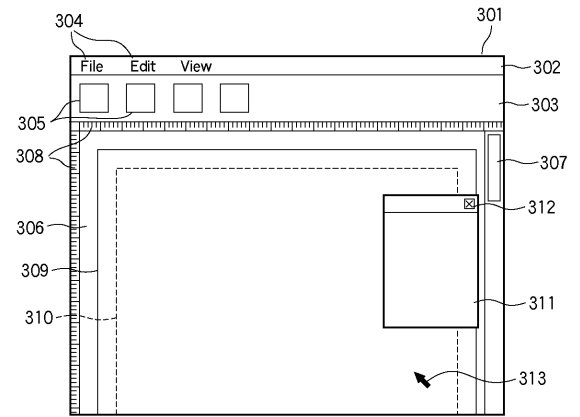
【図1C】



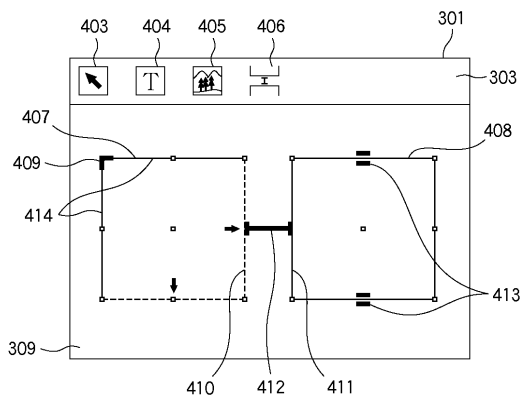
【図2】



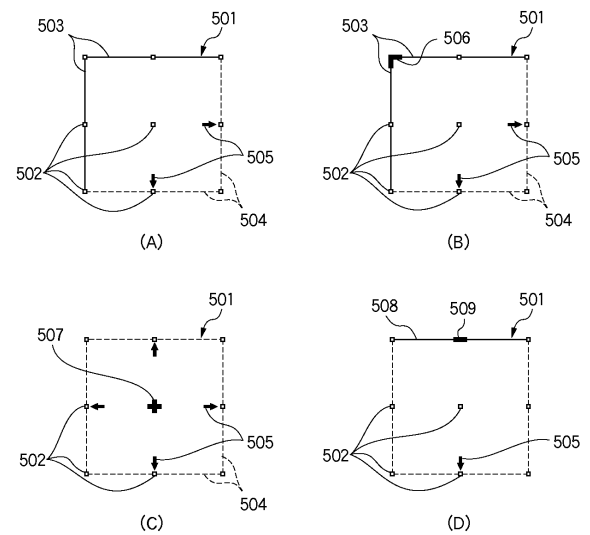
【図3】



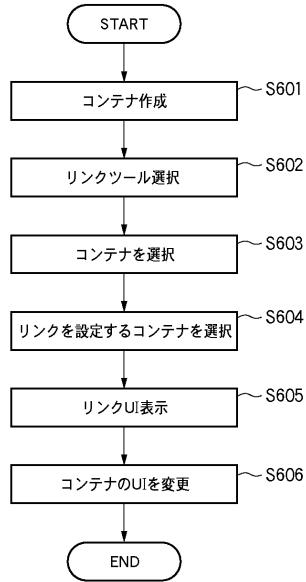
【図4】



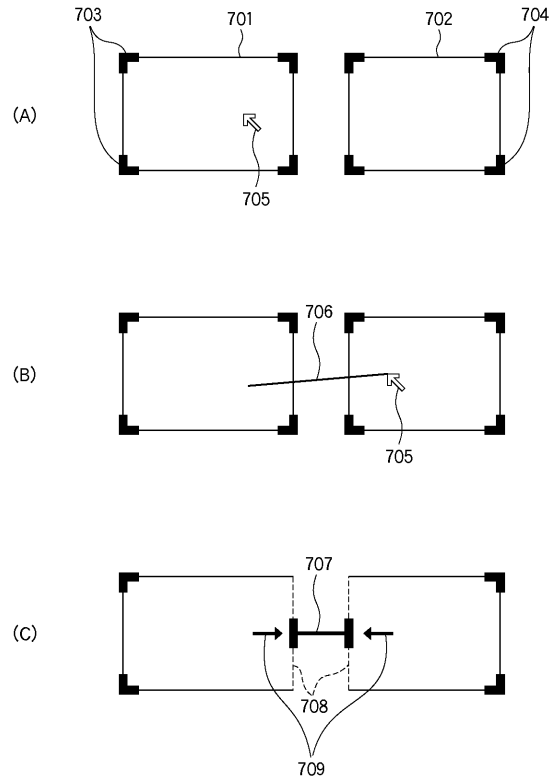
【図5】



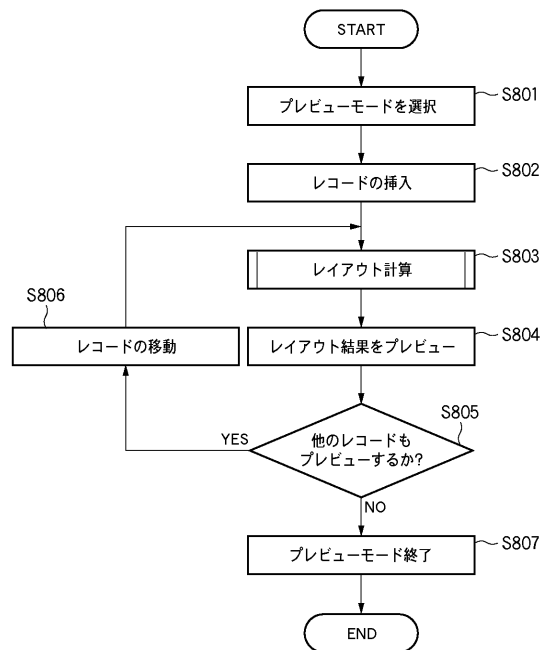
【図6】



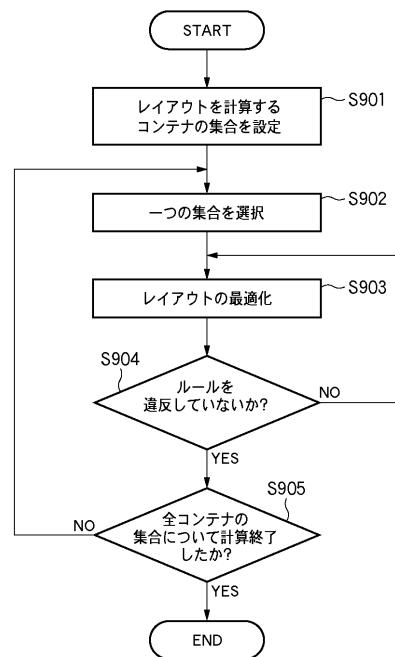
【図7】



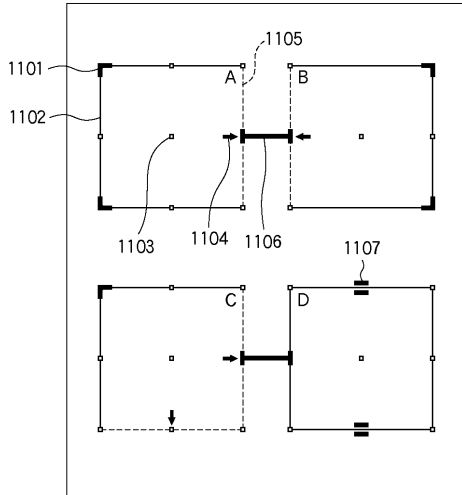
【図8】



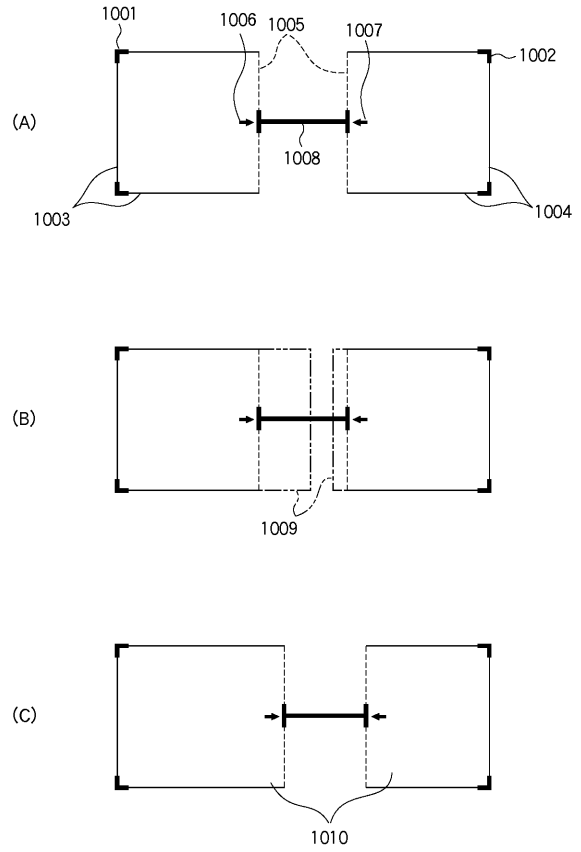
【図9】



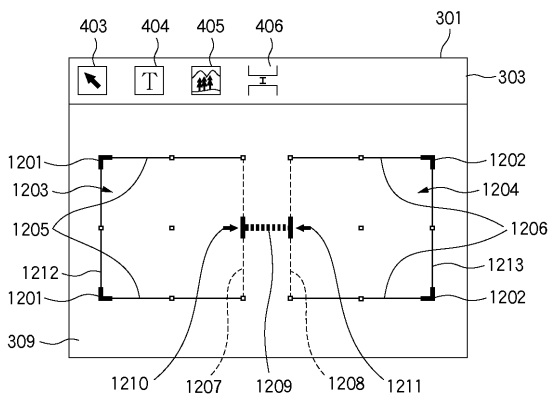
【図10】



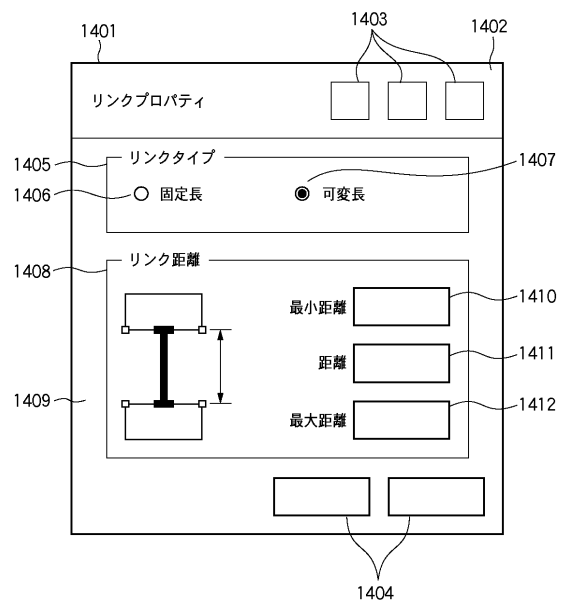
【図11】



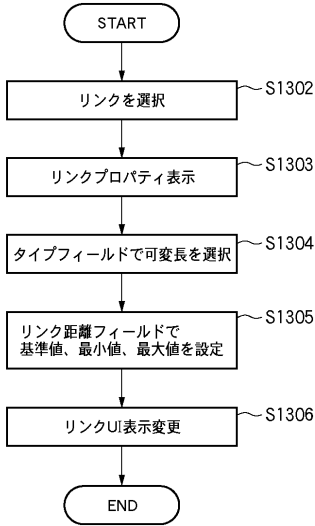
【図12】



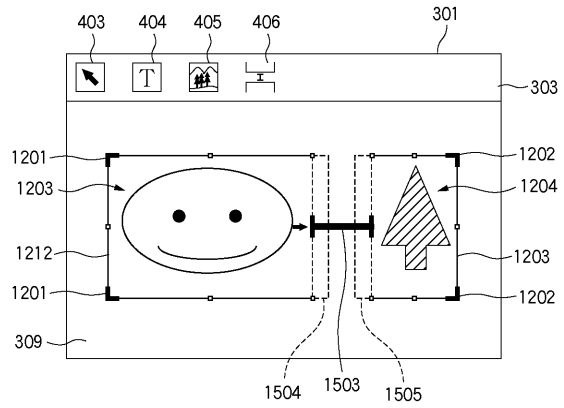
【図13】



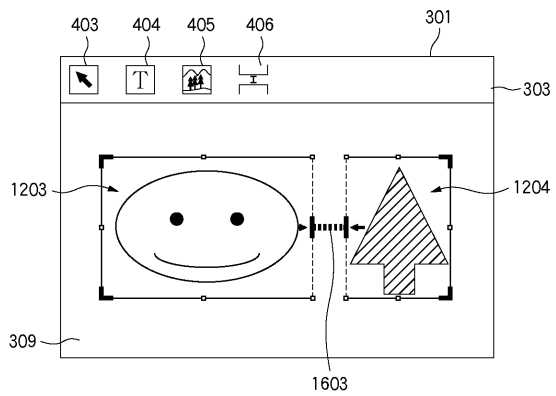
【図14】



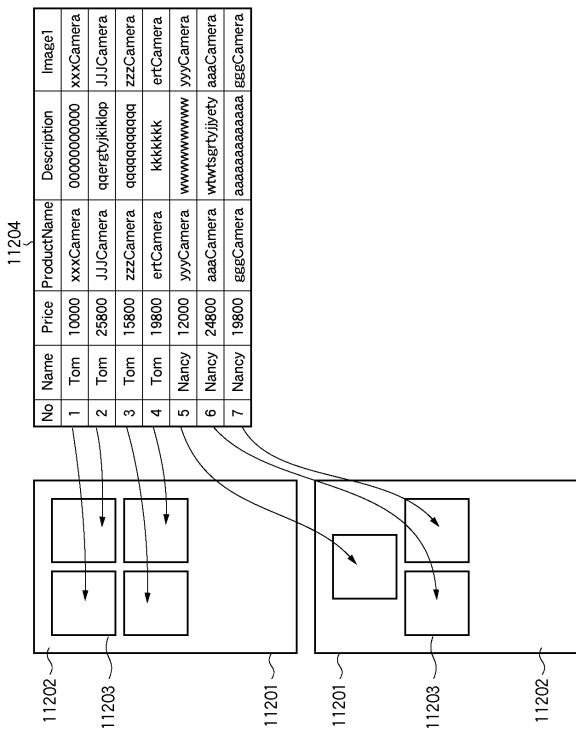
【図15】



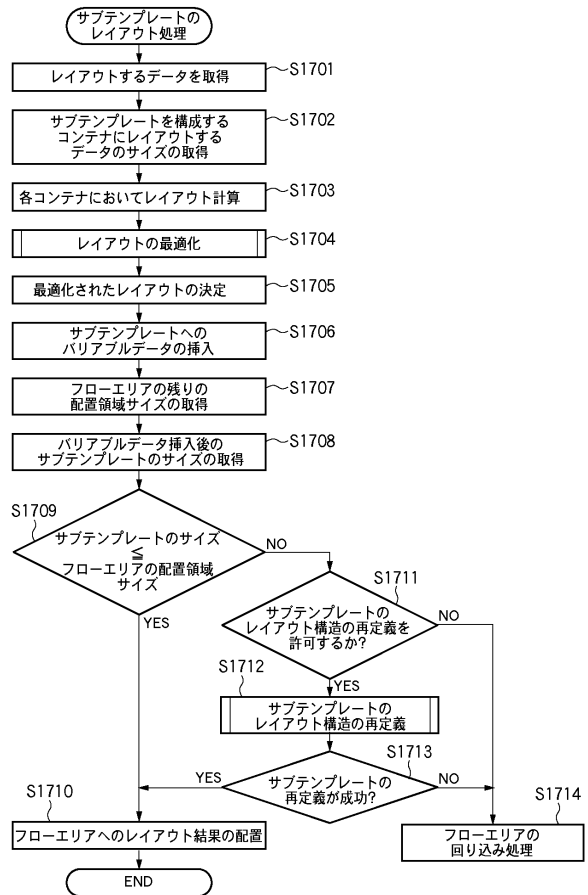
【図16】



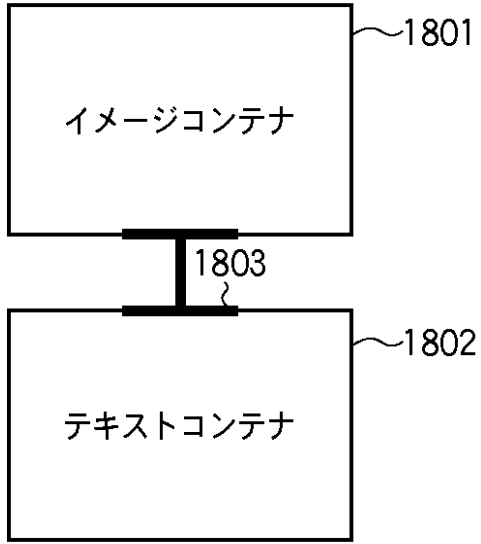
【図17A】



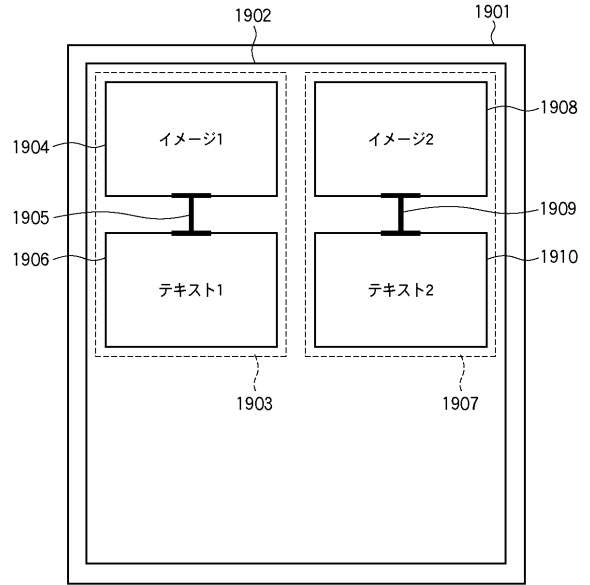
【図17B】



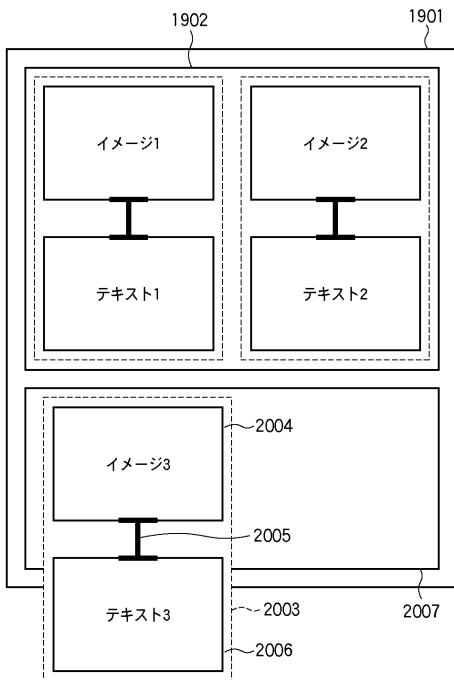
【図18】



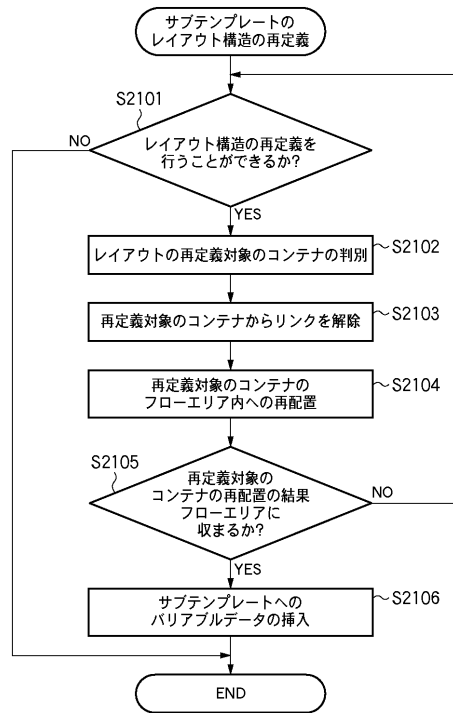
【図19】



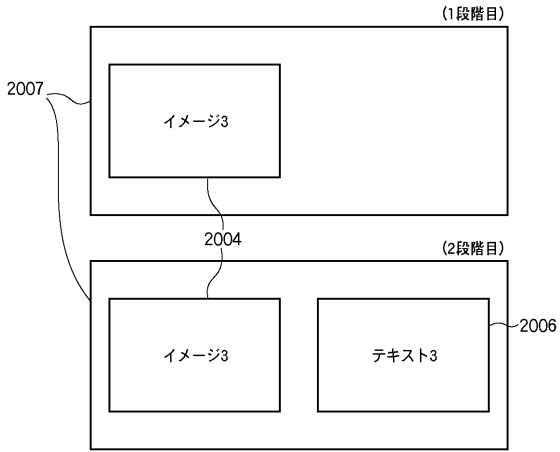
【図20】



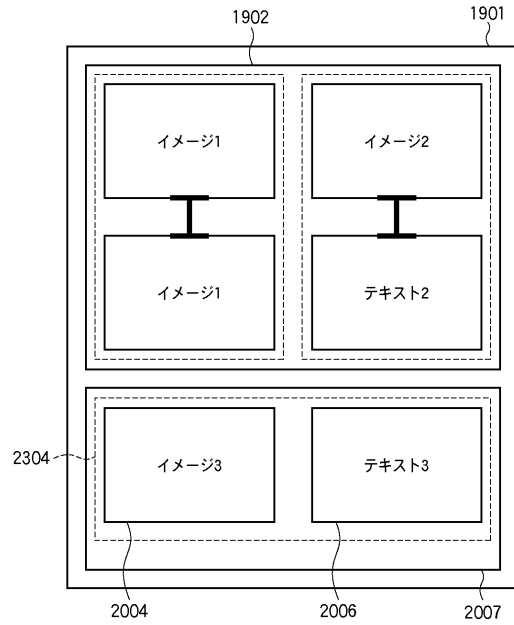
【図21】



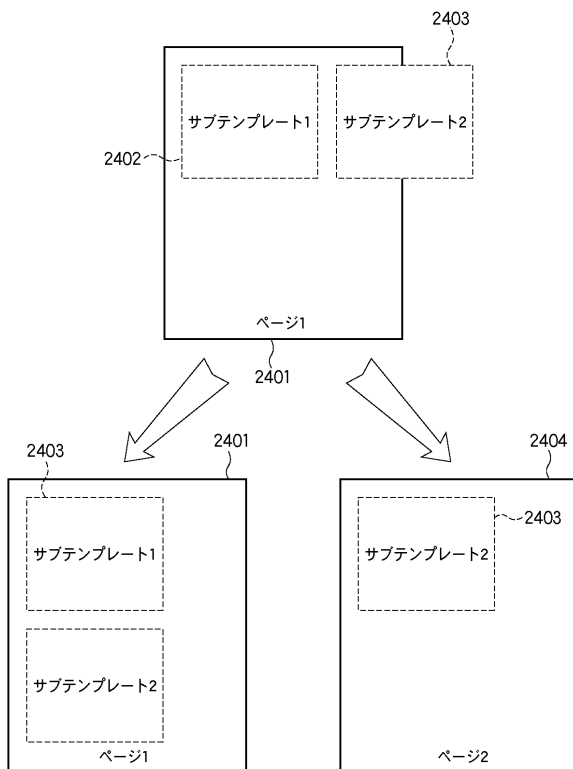
【図22】



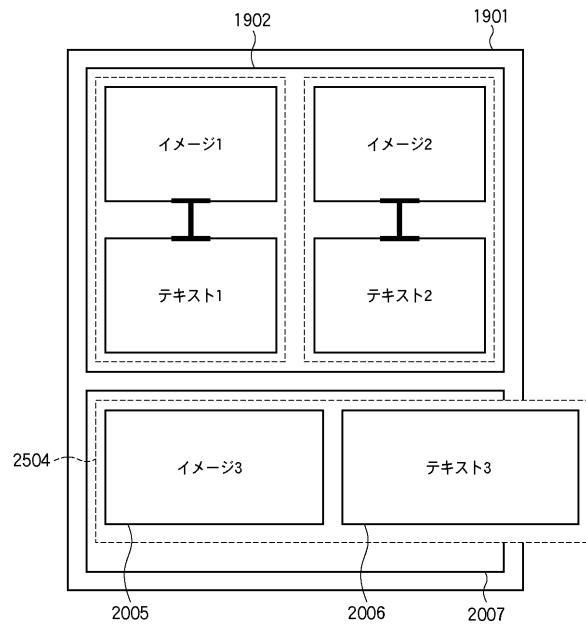
【図23】



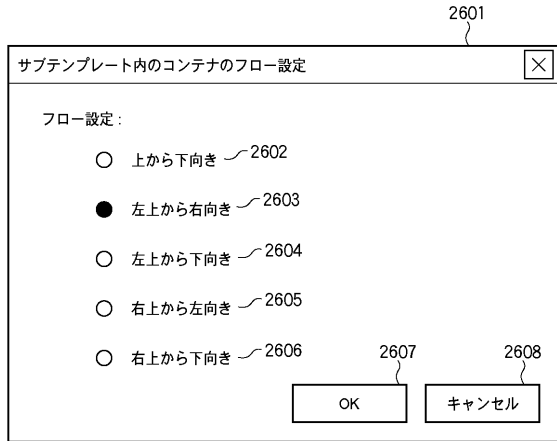
【図24】



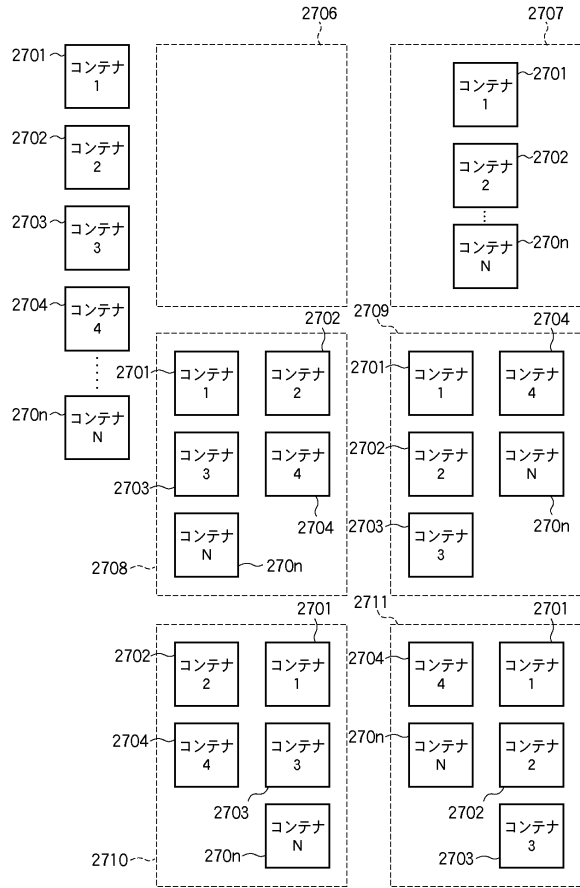
【図25】



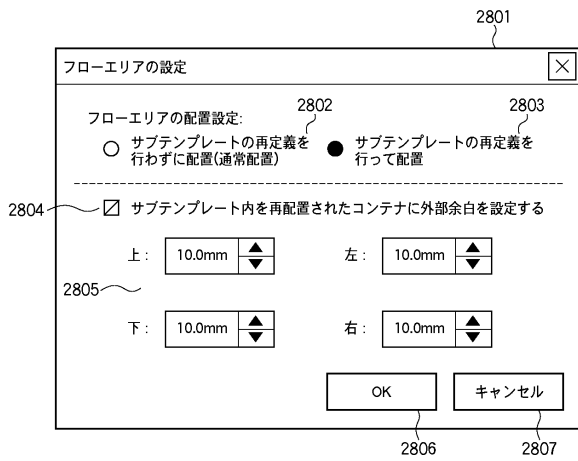
【図 26】



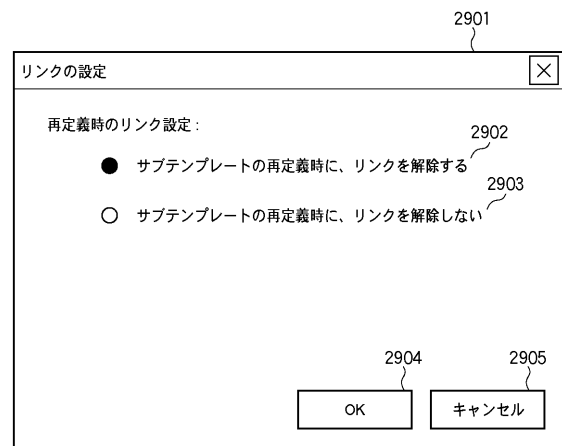
【図 27】



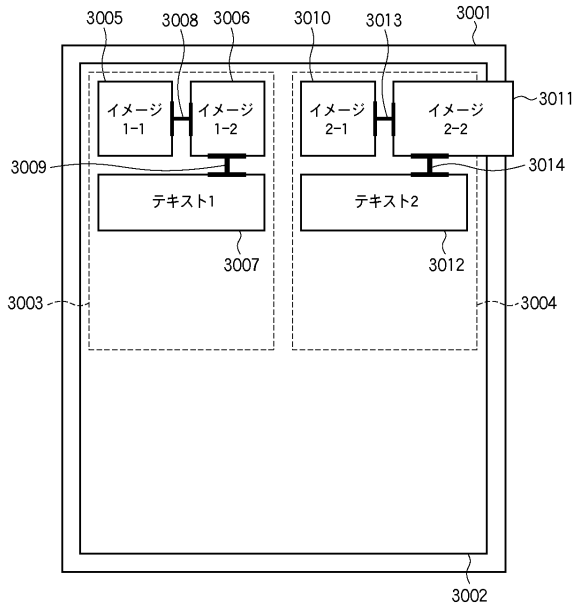
【図 28】



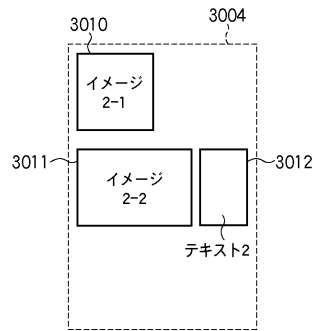
【図 29】



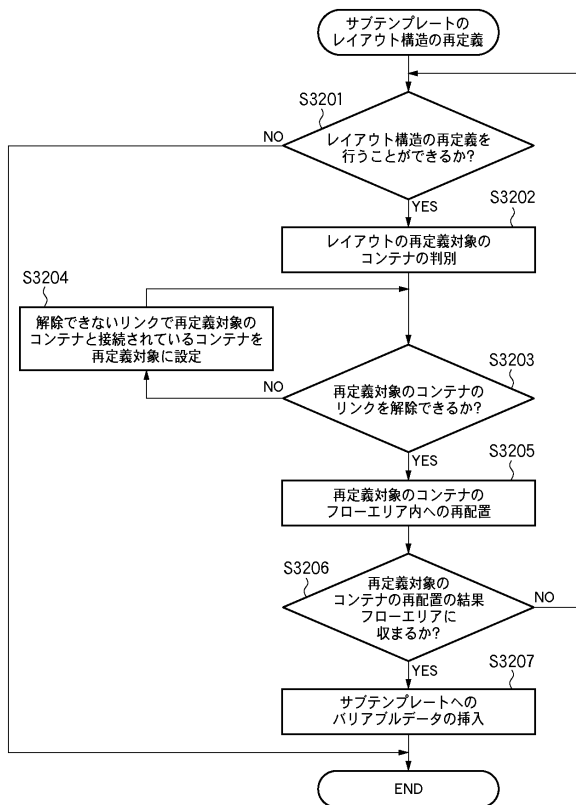
【図30】



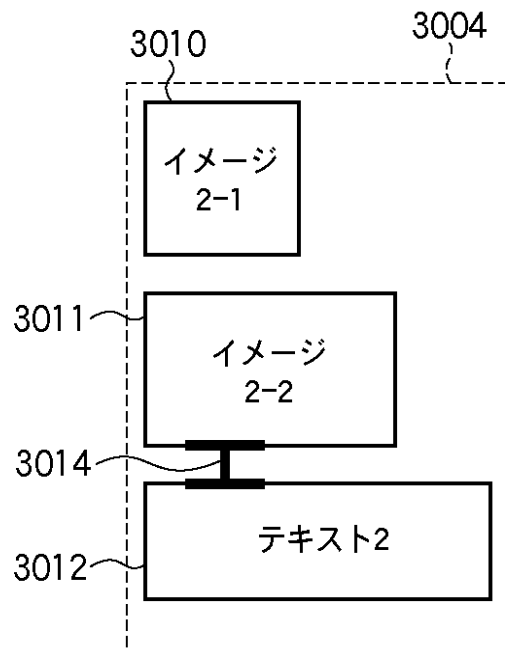
【図31】



【図32】



【図33】



フロントページの続き

(72)発明者 矢部 健太
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 西出 隆二

(56)参考文献 特開2005-216179(JP,A)
特開2000-200270(JP,A)
特開平10-340330(JP,A)
特開2006-099741(JP,A)
特開2005-071187(JP,A)
亀岡 慎平 SHINPEI KAMEOKA, SuperSQLを用いた組版におけるレイアウト自動修正
Automatic Layout Modification on Publishing System Using SuperSQL, 情報処理学会論文誌
第46巻 No. SIG13(TOD27) IPSJ, 日本, 社団法人情報処理学会 Information Processing Society of Japan, 2005年 9月15日, 第46巻 第SIG13TOD27号, 78~93
頁

(58)調査した分野(Int.Cl., DB名)
G06T 11/60
G06F 17/21