

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4298642号  
(P4298642)

(45) 発行日 平成21年7月22日 (2009. 7. 22)

(24) 登録日 平成21年4月24日 (2009. 4. 24)

(51) Int. Cl.

F I

G O 6 F 17/21 (2006. 01)

G O 6 F 17/21 5 3 6

G O 6 T 11/60 (2006. 01)

G O 6 F 17/21 5 4 0

G O 6 T 11/60 1 0 0 A

請求項の数 15 (全 47 頁)

(21) 出願番号 特願2004-361482 (P2004-361482)  
 (22) 出願日 平成16年12月14日 (2004. 12. 14)  
 (65) 公開番号 特開2006-171979 (P2006-171979A)  
 (43) 公開日 平成18年6月29日 (2006. 6. 29)  
 審査請求日 平成18年12月11日 (2006. 12. 11)

前置審査

(73) 特許権者 000001007  
 キヤノン株式会社  
 東京都大田区下丸子3丁目30番2号  
 (74) 代理人 100090538  
 弁理士 西山 恵三  
 (74) 代理人 100096965  
 弁理士 内尾 裕一  
 (72) 発明者 細坪 利彦  
 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 金子 幸一

最終頁に続く

(54) 【発明の名称】 レイアウト処理方法およびレイアウト処理装置およびレイアウト処理プログラム

(57) 【特許請求の範囲】

【請求項 1】

調整手段と、第1判定手段と、第2判定手段と、決定手段とを備え、複数のデータ領域を有するテンプレートのレイアウトを、各データ領域に流し込まれるコンテンツデータに従って決定するレイアウト処理装置におけるレイアウト処理方法であって、

前記調整手段が、複数のデータ領域に対して流し込まれる前記コンテンツデータに応じて前記複数のデータ領域のレイアウトを調整する調整工程と、

前記第1判定手段が、前記調整工程によって調整された前記レイアウトに予め設定された第1ページ領域をはみ出すデータ領域がある場合、当該データ領域のはみ出し量に基づいて当該データ領域を当該第1ページ領域に配置するのか、当該データ領域を当該第1ページ領域に続く第2ページ領域に配置するのかを判定する第1判定工程と、

前記第1判定工程が、前記はみ出すと判定されたデータ領域を前記第1ページ領域に配置すると判定した場合、前記第2判定手段が、当該第1ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第1ページ領域に配置できるか否かを判定する第2判定工程と、

前記第2判定工程が当該第1ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第1ページ領域に配置できると判定した場合、前記決定手段が、前記はみ出すと判定されたデータ領域を前記第1ページ領域に配置すると決定し、前記第2判定工程が当該第1ページ領域に既に配置されている他のデータ領域を縮小しても当該はみ出すと判定されたデータ領域を当該第1ページ領域に配

10

20

置できないと判定した場合、前記はみ出すと判定されたデータ領域を前記第 2 ページ領域に配置すると決定する決定工程を有することを特徴とするレイアウト処理方法。

【請求項 2】

前記レイアウト処理装置は、更に表示制御手段を備え、

前記第 1 判定工程が、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定できない場合、前記表示制御手段が、当該はみ出すと判定されたデータ領域が配置されるページ領域を指定するための指定画面を表示する表示制御工程を更に備えることを特徴とする請求項 1 に記載のレイアウト処理方法。

【請求項 3】

前記第 1 判定工程が、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定できない場合とは、当該データ領域のはみ出し量が、当該データ領域のサイズの半分の量であることを特徴とする請求項 2 に記載のレイアウト処理方法。

【請求項 4】

前記決定工程によって当該はみ出すと判定されたデータ領域が配置されるページ領域が決定された後、前記調整工程は、各ページ領域内で再度レイアウト調整を実行することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載のレイアウト処理方法。

【請求項 5】

前記レイアウト処理装置は、更に関連付け手段を備え、

前記関連付け手段が、前記複数のデータ領域の中の第 1 データ領域と第 2 データ領域とを関連付ける関連付け工程を更に有し、

前記調整工程は、前記第 1 データ領域に入力されるコンテンツデータのサイズと当該第 1 データ領域のレイアウト後のサイズの差、および、前記第 2 データ領域に入力されるコンテンツデータのサイズと当該第 2 データ領域のレイアウト後のサイズの差が等しくなるように、前記第 1 および第 2 データ領域のサイズを決定することを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載のレイアウト処理方法。

【請求項 6】

複数のデータ領域を有するテンプレートのレイアウトを、各データ領域に流し込まれるコンテンツデータに従って決定するレイアウト処理装置であって、

複数のデータ領域に対して流し込まれる前記コンテンツデータに応じて前記複数のデータ領域のレイアウトを調整する調整手段と、

前記調整手段によって調整された前記レイアウトに予め設定された第 1 ページ領域をはみ出すデータ領域がある場合、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定する第 1 判定手段と、

前記はみ出すと判定されたデータ領域を前記第 1 ページ領域に配置すると判定した場合、当該第 1 ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第 1 ページ領域に配置できるか否かを判定する第 2 判定手段と、

前記第 2 判定手段が当該第 1 ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第 1 ページ領域に配置できると判定した場合、前記はみ出すと判定されたデータ領域を前記第 1 ページ領域に配置すると決定し、前記第 2 判定手段が当該第 1 ページ領域に既に配置されている他のデータ領域を縮小しても当該はみ出すと判定されたデータ領域を当該第 1 ページ領域に配置できないと判定した場合、前記はみ出すと判定されたデータ領域を前記第 2 ページ領域に配置すると決定する決定手段を有することを特徴とするレイアウト処理装置。

【請求項 7】

当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判

10

20

30

40

50

定できない場合、当該はみ出すと判定されたデータ領域が配置されるページ領域を指定するための指定画面を表示する表示制御手段を更に備えることを特徴とする請求項 6 に記載のレイアウト処理装置。

【請求項 8】

前記第 1 判定手段が、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定できない場合とは、当該データ領域のはみ出し量が、当該データ領域のサイズの半分の量であることを特徴とする請求項 7 に記載のレイアウト処理装置。

【請求項 9】

前記決定手段によって当該はみ出すと判定されたデータ領域が配置されるページ領域が決定された後、前記調整手段は、各ページ領域内で再度レイアウト調整を実行することを特徴とする請求項 6 乃至 8 のいずれか 1 項に記載のレイアウト処理装置。

【請求項 10】

前記複数のデータ領域の中の第 1 データ領域と第 2 データ領域とを関連付ける関連付け手段を更に有し、

前記調整手段は、前記第 1 データ領域に入力されるコンテンツデータのサイズと当該第 1 データ領域のレイアウト後のサイズの差、および、前記第 2 データ領域に入力されるコンテンツデータのサイズと当該第 2 データ領域のレイアウト後のサイズの差が等しくなるように、前記第 1 および第 2 データ領域のサイズを決定することを特徴とする請求項 6 乃至 9 のいずれか 1 項に記載のレイアウト処理装置。

【請求項 11】

調整手段と、第 1 判定手段と、第 2 判定手段と、決定手段とを備えたコンピュータであり、複数のデータ領域を有するテンプレートのレイアウトを、各データ領域に流し込まれるコンテンツデータに従って決定するレイアウト処理を前記コンピュータに実行させるためのレイアウト処理プログラムであって、

前記調整手段が、複数のデータ領域に対して流し込まれる前記コンテンツデータに応じて前記複数のデータ領域のレイアウトを調整する調整工程と、

前記第 1 判定手段が、前記調整工程によって調整された前記レイアウトに予め設定された第 1 ページ領域をはみ出すデータ領域がある場合、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定する第 1 判定工程と、

前記第 1 判定工程が、前記はみ出すと判定されたデータ領域を前記第 1 ページ領域に配置すると判定した場合、前記第 2 判定手段が、当該第 1 ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第 1 ページ領域に配置できるか否かを判定する第 2 判定工程と、

前記第 2 判定工程が当該第 1 ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第 1 ページ領域に配置できると判定した場合、前記決定手段が、前記はみ出すと判定されたデータ領域を前記第 1 ページ領域に配置すると決定し、前記第 2 判定工程が当該第 1 ページ領域に既に配置されている他のデータ領域を縮小しても当該はみ出すと判定されたデータ領域を当該第 1 ページ領域に配置できないと判定した場合、前記はみ出すと判定されたデータ領域を前記第 2 ページ領域に配置すると決定する決定工程を前記コンピュータに実行させるための当該コンピュータが読み取り可能なレイアウト処理プログラム。

【請求項 12】

前記コンピュータは、更に表示制御手段を備え、

前記第 1 判定工程が、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定できない場合、前記表示制御手段が、当該はみ出すと判定されたデータ領域が配置されるページ領域を指定するための指定画面を表示する表示制御工程を更に備えることを特徴とする請求項 11 に記載のレイアウト処理プログラム。

**【請求項 1 3】**

前記第 1 判定工程が、当該データ領域のはみ出し量に基づいて当該データ領域を当該第 1 ページ領域に配置するのか、当該データ領域を当該第 1 ページ領域に続く第 2 ページ領域に配置するのかを判定できない場合とは、当該データ領域のはみ出し量が、当該データ領域のサイズの半分の量であることを特徴とする請求項 1 2 に記載のレイアウト処理プログラム。

**【請求項 1 4】**

前記決定工程によって当該はみ出すと判定されたデータ領域が配置されるページ領域が決定された後、前記調整工程は、各ページ領域内で再度レイアウト調整を実行することを特徴とする請求項 1 1 乃至 1 3 のいずれか 1 項に記載のレイアウト処理プログラム。

10

**【請求項 1 5】**

前記コンピュータは、更に関連付け手段を備え、

前記関連付け手段が、前記複数のデータ領域の中の第 1 データ領域と第 2 データ領域とを関連付ける関連付け工程を更に有し、

前記調整工程は、前記第 1 データ領域に入力されるコンテンツデータのサイズと当該第 1 データ領域のレイアウト後のサイズの差、および、前記第 2 データ領域に入力されるコンテンツデータのサイズと当該第 2 データ領域のレイアウト後のサイズの差が等しくなるように、前記第 1 および第 2 データ領域のサイズを決定することを特徴とする請求項 1 1 乃至 1 4 のいずれか 1 項に記載のレイアウト処理プログラム。

**【発明の詳細な説明】**

20

**【技術分野】****【0001】**

本発明は、自動レイアウトシステムにおける装置 / 媒体、およびレイアウト手法に関するものである。

**【背景技術】****【0002】**

近年、商品の多品種化で商品寿命が短くなっていること、あるいはインターネット利用の普及により消費者のカスタマイズサービス指向が高まってきていること、などに起因して 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。これらの手法は、顧客満足度を高め、顧客の開拓や囲い込みを目指すという目的に対して非常に効果的なものである。

30

**【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、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、その内容を分析したうえで、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。特に最近では 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とに伴って、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、顧客毎に異なる量のコンテンツを最適にレイアウトすることが求められるようになってきている。

40

**【0004】**

一般に、バリアブルプリントシステムではそのようなカスタマイズ文書を作成するにあたり、ドキュメント上にコンテナをレイアウトする。コンテナとはオブジェクトの 1 つでありデータベース化されたコンテンツ ( 描画内容 ) を描画するための矩形の部分領域をいう ( フィールド領域と呼ばれることもある )。かかるバリアブルプリントシステムによれば、ドキュメント上にこのようなコンテナをレイアウトした後、データベースとコンテナとを関連付ける ( データベースの各コンテンツと各コンテナとを関連付ける ) ことにより、カスタマイズ文書 ( ドキュメント ) を作成することができる。

**【0005】**

しかし、バリアブルプリントシステムの場合、コンテナのサイズが固定であるためデー

50

データベース内の各コンテンツを構成するデータがコンテナに流し込まれた際、コンテンツデータ量がコンテナサイズより大きいとコンテンツデータの一部が欠落するオーバーフロー、またコンテンツデータ量がコンテナサイズより小さいとコンテナに隙間が空いてしまうという問題があった。

【 0 0 0 6 】

かかる問題を解決するために、コンテンツによってコンテナのレイアウトを動的に変更するシステムが考えられ、流し込まれるテキストデータおよびイメージデータのコンテンツデータ量に応じてコンテナサイズを決定することで上述したオーバーフローや隙間が生じるといった問題点を解決することができる。

【 0 0 0 7 】

具体的には、コンテナのサイズを可変とし、流し込まれるコンテンツデータのデータ量に応じてコンテナのサイズを大きくしてレイアウトしたり、またテキストデータの場合にあっては、コンテナ内のフォントのサイズを可変とし、流し込まれるデータのデータ量に応じてフォントのサイズを縮小してレイアウトすることにより、コンテナ内に入りきらないデータ量のテキストデータが流し込まれた場合であっても、コンテナ内に全てのテキストデータを表示させることが可能である。

【 0 0 0 8 】

上述した技術の一部を実現しているレイアウトシステムとして下記特許文献 1 が挙げられる。

【 0 0 0 9 】

下記特許文献 1 においてはコンテナ内に文字を入力していき、あらかじめ設定したサイズのコンテナ内に文字が収まりきれない量の文字が入力された場合は、文字の入力に応じてコンテナサイズを拡大していき、その拡大に応じて隣接するコンテナサイズが縮小されるといった技術が開示されている。

【特許文献 1】特開平 7 - 1 2 9 6 5 8 号

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 0 】

しかしながら、上記レイアウトシステムの場合、コンテナ内に入力されるテキストデータのデータ量の増加に応じて該コンテナは拡大され、その拡大に応じて隣接するコンテナのサイズは縮小されていくが、コンテナは入力が続く限り拡大処理が続けられるため、結果として隣接するコンテナは縮小を続けなければならない、バランスの取れたレイアウトを作成することはできない。さらに、上記レイアウトシステムではバリアブルプリントについて想定しておらず、各コンテナにコンテンツを流し込んでいくということも考慮されていない。

【 0 0 1 1 】

また、バリアブルデータプリントを考慮したレイアウト方法として、固定のコンテナを作成し、そこにコンテンツデータを流し込んでいくものがある。しかし、固定サイズのコンテナを用いる場合、コンテナサイズより大きいサイズのコンテンツデータが流し込まれると、オーバーフローが生じてしまったり、フォントサイズを縮小して無理に流し込もうとするとフォントサイズが極端に小さくなってしまふといった問題が生じてしまう。

【 0 0 1 2 】

そこで、上記問題を解決するためのレイアウト方法として、関連付けられた複数の可変コンテナ間のサイズバランスを考慮したレイアウト方法が考えられる。これにより、各コンテナに流し込まれるコンテンツデータのサイズに応じて各コンテナのサイズが変更され、さらに関連付けられているコンテナ間で各コンテナサイズの変化量を調整したレイアウト処理を行うことでコンテナのサイズバランスが考慮されたレイアウト結果を得ることができる。

【 0 0 1 3 】

さらに、上記レイアウト方法を複数ページについて適応することも考えられる。つまり

10

20

30

40

50

、コンテンツデータが流し込まれるコンテナを予め複数ページに設定することにより、コンテンツデータは複数ページに設定された各コンテナに流し込まれるため、複数ページにおけるページ毎に上記レイアウト方法を行うことが可能となる。

【0014】

しかし、上記レイアウト方法では、例えばコンテナAを1ページ目にコンテナBを2ページ目に配置して、コンテンツデータを各コンテナに流し込み上記サイズ調整をした場合、コンテナAに流し込まれたコンテンツデータのサイズが小さいため、コンテンツデータが流し込まれたコンテナBを1ページ目に配置する隙間が生じる場合が考えられる。

【0015】

このような場合であっても、上記レイアウト方法では予め設定されたページ内でコンテナのサイズを調整してレイアウトすることしか考慮しておらず、隙間が生じることによって見栄えの悪いレイアウトになるだけでなく、無駄なページが発生してしまう恐れがある。

【0016】

本願発明は上記課題に鑑みてなされたものであり、複数のコンテナを関連付けるリンクを複数ページを跨いで設定できるようにすることで、流し込まれたコンテンツデータに応じて予め設定されたページにコンテナを配置するのか、それとも予め設定されたページとは異なるページにコンテナを配置するのかを判断することが可能となり、複数ページを考慮して全体的にバランスの取れた自動レイアウト処理を行うことを目的とする。

【課題を解決するための手段】

【0017】

上記目的を達成するために、本発明は次のような構成からなる。すなわち、調整手段と、第1判定手段と、第2判定手段と、決定手段とを備え、複数のデータ領域を有するテンプレートのレイアウトを、各データ領域に流し込まれるコンテンツデータに従って決定するレイアウト処理装置におけるレイアウト処理方法であって、前記調整手段が、複数のデータ領域に対して流し込まれる前記コンテンツデータに応じて前記複数のデータ領域のレイアウトを調整する調整工程と、前記第1判定手段が、前記調整工程によって調整された前記レイアウトに予め設定された第1ページ領域をはみ出すデータ領域がある場合、当該データ領域のはみ出し量に基づいて当該データ領域を当該第1ページ領域に配置するのか、当該データ領域を当該第1ページ領域に続く第2ページ領域に配置するのかを判定する第1判定工程と、前記第1判定工程が、前記はみ出すと判定されたデータ領域を前記第1ページ領域に配置すると判定した場合、前記第2判定手段が、当該第1ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第1ページ領域に配置できるか否かを判定する第2判定工程と、前記第2判定工程が当該第1ページ領域に既に配置されている他のデータ領域を縮小することで当該はみ出すと判定されたデータ領域を当該第1ページ領域に配置できると判定した場合、前記決定手段が、前記はみ出すと判定されたデータ領域を前記第1ページ領域に配置すると決定し、前記第2判定工程が当該第1ページ領域に既に配置されている他のデータ領域を縮小しても当該はみ出すと判定されたデータ領域を当該第1ページ領域に配置できないと判定した場合、前記はみ出すと判定されたデータ領域を前記第2ページ領域に配置すると決定する決定工程を有することを特徴とする。

【発明の効果】

【0018】

複数のコンテナを関連付けるリンクを複数ページを跨いで設定できるようにすることで、流し込まれたコンテンツデータに応じて予め設定されたページにコンテンツを流し込むのか、それとも予め設定されたページとは異なるページにコンテンツを流し込むのかを判断し、複数ページを考慮して全体的にバランスの取れたレイアウト処理を行うことができる。

【発明を実施するための最良の形態】

【0019】

本発明の実施例を適用するのに好適である実施例について説明を行う。

【実施例１】

【００２０】

本発明の実施例を説明する前に、本発明を適用可能なシステム構成、およびアプリケーション構成について説明する。

【００２１】

<システム構成図>

まず、図１Ａおよび図１Ｂを参照して、本発明の一実施形態にかかるホストコンピュータ（情報処理装置）を備える自動レイアウトシステムの構成を説明する。図１Ａは自動レイアウトシステム１００の構成例を示すブロック図である。また、図１Ｂは図１Ａに示されたホストコンピュータ（情報処理装置）１０１の構成を更に詳細に示すブロック図である。

10

【００２２】

本実施形態で説明されるレイアウト編集処理ならびに自動レイアウト処理は、ホストコンピュータ１０１（汎用コンピュータモジュールで構成される）によって実行される。自動レイアウトシステム１００上で実施可能となるレイアウト編集アプリケーションプログラム１２１はホストコンピュータ１０１において、そのソフトウェアの全体、あるいは一部分が実行され、これにより、上記レイアウト編集処理ならびに自動レイアウト処理が実現される。

【００２３】

20

レイアウト編集アプリケーションプログラム１２１はコンピュータの可読媒体に格納され、そのコンピュータの可読媒体からホストコンピュータ１０１のメモリ１３６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の可読媒体はコンピュータプログラム製品である。コンピュータ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、ドキュメントの作成や作成されたドキュメントの印刷に好適な装置が提供されることになる。

【００２４】

コンピュータモジュール１０１はキーボード１３２やマウス１３３のようなポインティングデバイスなどの入力装置をつなぎ、ディスプレイ装置１４４や状況に応じてはローカルプリンタ１４５を含む出力装置を連結する。入力／出力インターフェース１３８はコンピュータモジュール１０１をネットワーク接続１０７から接続してシステム１００の他のコンピュータ装置につなげることができる。そのネットワーク接続１０７の典型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）である。

30

【００２５】

コンピュータモジュール１０１は典型的に少なくとも１つのプロセッサユニット１３５、例えば半導体のランダムアクセスメモリ（ＲＡＭ）やリードオンリーメモリ（ＲＯＭ）から構成されるメモリユニット１３６、ビデオインタフェース１３７を含むＩＮＰＵＴ／ＯＵＴＰＵＴ（Ｉ／Ｏ）インターフェース、キーボード１３２やマウス１３３のためのＩ／Ｏインターフェース１４３を含んでいる。記憶装置１３９は典型的にハードディスクドライブ１４０やフロッピー（登録商標）ディスクドライブ１４１を含んでいる。図には示されていないが磁気テープドライブもまた使用される可能性がある。ＣＤ－ＲＯＭドライブ１４２は不揮発性のデータソースとして提供される。コンピュータモジュール１０１はＧＮＵ／ＬＩＮＵＸやマイクロソフトウィンドウズ（登録商標）のようなオペレーションシステムや、典型的にはオペレーションシステムに従う形で、あるい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、相互接続バス１３４を介して通信を行うコンピュータモジュール１０１のコンポーネント１３５から１４３を利用する。図に記述した配置コンピュータの例ではＩＢＭ互換ＰＣやＳＵＮ社のＳｐａｒｃ Ｓｔａｔｉｏｎ、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。

40

50

## 【 0 0 2 6 】

レイアウト編集アプリケーションプログラム 1 2 1 は典型的に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 により実行、読み込みが制御される。プログラム 1 2 1 の媒介記憶装置とネットワーク 1 0 2 0 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 に呼応して半導体メモリ 1 3 6 を使用する。いくつかのインスタンスではアプリケーションプログラム 1 2 1 が C D - R O M やフロッピー（登録商標）ディスク上でエンコードされ、対応するドライブ 1 4 2 や 1 4 1 を通じて読み込まれユーザに提供される。あるいはもう一つの方法としてレイアウト編集アプリケーション 1 2 1 はネットワーク接続 1 0 7 からユーザによって読み込まれる場合もある。さらに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、光磁気ディスクまたは無線またはコンピュータモジュール 1 0 1 とその他のデバイス間の赤外線通信、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大きさのコンピュータ可読媒体からコンピュータモジュール 1 0 1 内にロードされる可能性もある。前述は単に関連するコンピュータ可読メディアの一例であり、他のコンピュータ可読媒体も使用される可能性はある。

10

## 【 0 0 2 7 】

またレイアウト編集アプリケーション 1 2 1 はコンピュータに自動レイアウト処理ならびにレイアウト編集処理を行わせるものであり、3 つのソフトウェアコンポーネントを含んでいる。

## 【 0 0 2 8 】

20

これらのうち 1 つめのコンポーネントはレイアウトエンジン 1 0 5 であり、レイアウトエンジン 1 0 5 は、矩形の部分領域であるコンテナに与えられたサイズや位置の設定にしたがって、データベース 1 1 9 に格納されているデータからドキュメント単位でデータを読み込み、読み込んだデータとコンテナの設定とから、読み込んだデータが流し込まれるコンテナの大きさや位置等を計算する自動レイアウト処理を行うソフトウェアコンポーネントである。また、本実施形態におけるレイアウトエンジン 1 0 5 は、更に、コンテナに割り当てられたデータを描画し、ドキュメントのイメージを生成する処理も行う。ただし、本発明はこれに限るものではなく、レイアウトエンジン 1 0 5 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、図示省略したプリンタドライバに描画情報を出力することで、プリンタドライバがドキュメントのイメージ描画処理を行い、印刷データを生成するようにしてもよい。

30

## 【 0 0 2 9 】

2 つめのコンポーネントであるユーザインターフェース 1 0 3 はユーザによるレイアウト編集処理を可能とするもので、具体的には、ユーザはユーザインターフェース 1 0 3 を介してコンテナの作成・移動を行い、ドキュメントテンプレートを作成する。また、ドキュメントテンプレート内の各コンテナとデータとを関連付けるメカニズムを提供する。

## 【 0 0 3 0 】

3 つめのコンポーネントである U I モデルアナライザ 1 0 4 は、ユーザインターフェース 1 0 3 が提供するコンテナや、アンカー、スライダー、リンクなどのユーザインターフェース要素または各属性をレイアウトエンジン 1 0 5 が認識できる内部形式に変換する。すなわち、U I モデルアナライザ 1 0 4 により、レイアウトエンジン 1 0 5 に変更を加えることなく、ユーザインターフェース 1 0 3 を他の表示または入力形式のユーザインターフェースに容易に取替え可能になる。なお、リンクとはコンテナ間を関連付けるためのものである。

40

## 【 0 0 3 1 】

ユーザインターフェース 1 0 3、U I モデルアナライザ 1 0 4、レイアウトエンジン 1 0 5 はコミュニケーションチャンネル 1 2 3、1 2 4 を介して通信する。

## 【 0 0 3 2 】

ドキュメント生成のためのデータソースは一般的にデータベースアプリケーションを動かしている他のコンピュータによって構成されたデータベースサーバー 1 1 7 上にある典

50



型的なデータベース 119 である。ホストコンピュータ 101 はネットワーク接続 107 の手段によっ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 が一般的に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。またレイアウト編集アプリケーション 121 は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ドキュメントを生成する。これらの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に保存されるか、ファイルサーバ 115 に保存されるか、あるいはプリンタ 113 に直接印刷される。プリントサーバ 109 は直接ネットワークにつながっていないプリンタ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 とプリンタ 113 は典型的な通信チャンネル 111 を介して接続される。

10

#### 【0033】

図 2 はエンジンサーバ 227 の追加を除き、レイアウトエンジン 105 の分離バージョン 225 を含む図 1 A・図 1 B の類似の図である。エンジンサーバ 227 は典型的なコンピュータである。レイアウトエンジン 225 は、印刷やその他の目的におじてドキュメントを生成するために、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とデータベース 119 に保存されたデータとを結合する。そのようなオペレーションはユーザインターフェース 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。

#### 【0034】

〔アプリケーション構成図〕

20

〔メインウィンドウ〕

図 3 で参照されるように、ユーザインターフェース 103 は、操作時にビデオディスプレイ 144 に表示されるアプリケーションウィンドウ 301 によって形成されたユーザインターフェースを含んでいる。ウィンドウ 301 は、非表示にすることや、スクリーン上の色々な場所に移動することが可能なメニューバー 302 とツールバー 303、そしてマウス 133 の位置・動作によって場所を移動可能なワークエリア 306 とオプションのパレット 311 とカーソル/ポインタデバイス 313 から構成される。

#### 【0035】

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 を持つ。

30

#### 【0036】

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にする、または表示することが可能な多くのツールボタンとウィジェット 305 を持つ。

#### 【0037】

オプションのルーラー 308 はワークエリア内のポインタ、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われる。

#### 【0038】

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ィンドウコントロール 312 を持つ。パレット 311 はオプションで、ワークエリアの前面に表示される、あるいはオブジェクトの背面に隠される。パレット 311 はアプリケーションウィンドウ 301 の範囲内のみに表示されることも、あるいはアプリケーションウィンドウ 301 の外側に部分的または全体を表示することもできる。

40

#### 【0039】

ツールバーエリア 303 は図 4 に示されるように少なくとも次のユーザ選択可能な『ツールボタン』を持つ。

#### 【0040】

(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・ロック解除するために使われる。コンテナは、(複数)コンテナの周りに選択ボックスをドラッグする、あるいは複数コンテナを選択する間に C T R L キーを押して

50

づけることによって、複数選択を可能にする。

【 0 0 4 1 】

( 2 ) イメージコンテナツールボタン 4 0 5 : スタティックあるいはバリアブルイメージを持つコンテナを作成するために使われる。

【 0 0 4 2 】

( 3 ) テキストコンテナツールボタン 4 0 4 : スタティックあるいはバリアブルテキストを持つコンテナを作成するために使われる。

【 0 0 4 3 】

( 4 ) リンクツールボタン 4 0 6 : コンテナ間の距離をコントロールするリンクを作成するために使われる。

【 0 0 4 4 】

これらのボタンは、周知の技術であ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。

【 0 0 4 5 】

[ ドキュメントテンプレート ]

ワークエリア 3 0 6 はドキュメントテンプレートのデザインを表示・編集するために使われる。これによりユーザは下準備で印刷されるドキュメントの概観をデザインすること、そして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理解することが可能なる。

【 0 0 4 6 】

もし外部データソースとの関連づけがテンプレートに設定されていたなら、現在のドキュメントのプレビューができるように、バリアブルテキストとイメージがそれらのコンテナに表示される。

【 0 0 4 7 】

ドキュメントの構造とバリアブルデータコンテナの描写をする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いつも表示される。

【 0 0 4 8 】

ワークエリア 3 0 6 はスクロールバー 3 0 7 とオプションのルーラー 3 0 8 とドキュメントテンプレート 3 0 9 に特徴付けられる。ドキュメントテンプレート 3 0 9 はページが複数あることを示すことができる。

【 0 0 4 9 】

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、ユーザによって指定される。それぞれのドキュメントでの実際のページ数は、流し込まれるコンテンツデータに応じて変化することもあり、もし 1 ページ内にフィットできなかった時、追加のページは自動的に作成される。

【 0 0 5 0 】

それぞれのページ内の境界線は、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ガイド 3 1 0 である。

【 0 0 5 1 】

また、図 4 は 1 ページのドキュメントテンプレート 3 0 9 上に表示することが可能なオブジェクトの例である、それらは、複数のコンテナ 4 0 7、4 0 8 と、任意に適用するアンカーアイコン 4 0 9 と固定されていない辺 4 1 0、リンク 4 1 2 そしてスライダー 4 1 3 を持つ。

【 0 0 5 2 】

[ コンテナ ]

コンテナは、ドキュメントテンプレート内にデータベースから固定あるいは可変のコンテンツデータ(テキストまたはイメージ)が流し込まれ、描画される部分領域(データ領域)であり、図 4 に示されるように他のコンテナと共にドキュメントテンプレート 3 0 9 内に配置される。ユーザインターフェース画面を介して、ユーザからの操作指示により、コンテナはマウス 1 3 3 の操作により移動、サイズ調整、再作成される。下記は本実施形

10

20

30

40

50

態におけるコンテナの定義である。

【 0 0 5 3 】

( 1 ) コンテナ ( データ領域とも呼ばれる ) は固定あるいは可変のコンテンツと関連付けられる。可変コンテンツは、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、レコードごとに異なる可能性があるという意味でダイナミック ( 動的 ) であるといえることができる。ただし、本実施形態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適していない。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。

10

【 0 0 5 4 】

( 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のような装飾機能を持っている。このような設定をコンテナ情報と呼ぶ。コンテナ情報は、各コンテナに設定可能であるが、あるコンテナと同じコンテナ情報であるという設定を行うことも可能である。

【 0 0 5 5 】

( 3 ) コンテナはドキュメントを生成したとき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特定のデータの表示をもたらす。コンテナのこの表現は例えば印刷されるか、スクリーン 1 4 4 上で表示されるか、その両方が可能である。

20

【 0 0 5 6 】

( 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ー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ーフェース ( G U I ) を持つ。G U I の各要素はビデオディスプレイ 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 のユーザインターフェース 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 0 5 7 】

30

[ コンテナ制約 ]

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制御する制約がある。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と共に ) 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制御する主要な方法である。制約の例は『このコンテナのコンテンツの高さは、最大値 4 インチです。』である。もうひとつの制約は『コンテナのコンテンツの左端は、それぞれのドキュメントで同じ水平位置で表示しなければならない。』である。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、編集するための方法である。

【 0 0 5 8 】

イメージがページ上に定義された場所を持っているように、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。なお、以下の説明ではバリエブルデータ印刷に特化した方法での表示・編集に焦点を合わせる。

40

【 0 0 5 9 】

コンテナは、ユーザにドキュメントのコンテンツのサイズ・位置を指定することを可能にする。いくつかのドキュメントは一つのドキュメントテンプレートから生成されるので、コンテナは多数の可能性と制約を指定・表示するためにユーザインターフェースが利用される。

【 0 0 6 0 】

50

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。このように、本明細書でコンテナの左辺を論じることが、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。同様に、コンテナの高さを論じることが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、ユーザインターフェース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と考えられる。

#### 【0061】

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いくつかの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。

#### 【0062】

(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。

(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。

(3) 距離(リンクの長さ)の制約が固定である場合、指定された距離は全てのドキュメントのための制約である。

(4) コンテナの左右辺が固定である場合、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。しかし、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同じ水平位置となるが、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。

(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。ただし、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。

(6) コンテナの垂直軸はコンテナの右辺と左辺の平行で、そして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ば、当該コンテナの左右辺の水平位置の平均(すなわち左右の中央位置)は、すべてのドキュメントで同じである。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近いものまで、垂直軸は全てのドキュメントで同じ水平位置となる。なお、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。

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

(8) 水平軸および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。しかし、コンテナの幅・高さはこの制約によって影響されない。

(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ば、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) 垂直辺あるいは垂直軸はページの左辺、あるいは右辺、あるいは左ページマージン、あるい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あるいは下辺、あるいは上下マージン、あるいは他の垂直位置に関連付けられて固定になる。用語『固定』の改良は、もしページサイズが全てのドキュメントで同じならこれらの可能性はドキュメント生成で相違はないため、ドキュメント間でページサイズが変わるときにのみ重要となる。

#### 【0063】

『固定』の反対は、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している『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、動的にレイアウト変更されることを

10

20

30

40

50

期待するが、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、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とする場合がある。そのため、本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、中央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た。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。

#### 【0064】

〔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、そして埋め込みのイメージを持つ。イメージコンテナは、イメージだけを持つ。

10

#### 【0065】

図4で参照されるように、新規テキストコンテナとイメージコンテナは、テキストコンテナツール404、あるいはイメージコンテナツール405をマウス133でクリックし、テンプレート309に四角形をドラッグすることによって、ドキュメントテンプレート309上に作成される。あるいは、コンテナは適切なツール404、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、デフォルトサイズのコンテナが挿入されるか、新規コンテナの寸法を入れるために、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。いくつかのコンテナは自動的に前もって定義されたか、計算されたスキーマによって作成・配置されるなどいろいろな方法が考えられる。

20

#### 【0066】

ここで生成されたコンテナをマウス等の入力手段により選択し、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、コンテナの制約を設定することができる。コンテナのプロパティダイアログUI（部分領域設定手段に相当する）では、上述した各種の制約を設定することができる。また、コンテナのプロパティダイアログでは、コンテナのサイズ（幅、高さ）や位置を決定することができ、可変サイズにする場合は、コンテナの基本パターン（基本サイズと基準位置）を設定し、更に、最大コンテナサイズ（幅、高さ）と最小コンテナサイズ（幅、高さ）を設定することが可能となっている。

30

#### 【0067】

〔コンテナの表示方法〕

図5の（A）から（D）はコンテナの表示に関するルールを例示している。

#### 【0068】

レイアウト編集アプリケーション 121 は、辺の状態を表現するために実線（アイテム 503）あるいは点線（504）で辺を描き、アンカー（辺の近くに描画された506、507、509によって示されるような線、形状、アイコン）、ハンドル（移動、修正するために辺、形の近くに描画されたコントロール点、502）、スライダー（辺の両サイドに描画された短い並行線、図4の413）、拡張アイコン（505）、そして色を特徴として持っている。

40

#### 【0069】

図5の（A）から（D）のコンテナ表示方法のルールは、つぎの通りである。

#### 【0070】

- （1）それぞれの辺を固定するために、実線で描画する。
- （2）幅が固定である場合、左と右の辺を実線で描画する。
- （3）高さが固定である場合、上と下の辺を実線で描画する。
- （4）軸は描画しない。
- （5）（1）から（3）によって描画されていない辺は、それぞれの辺の近くに拡張アイコンが描画され、点線で描画される。
- （6）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、エッ

50

ジの中央にスライダーが描画される。

( 7 ) 垂直辺および水平辺、あるいは垂直および水平軸のそれぞれのペアで、もしアンカーやスライダーが描画されていなければ、それらの交差点にハンドルが描画される。

【 0 0 7 1 】

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ならため実線で描画される。ルール ( 5 ) で定義された可変の辺は、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。

【 0 0 7 2 】

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。

【 0 0 7 3 】

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。後で記述されるように、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、ユーザインター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。代替の実行手段としては、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、あるいは可変の辺がユーザインターフェースで、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きる。

【 0 0 7 4 】

これらのコンテンツ表現は、コンテナの辺の状態を表示するグラフィカルな手段を提供する。その表現の解釈は下記のとおりである。

【 0 0 7 5 】

( 1 ) 図 4 の 4 1 0 の辺のように、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。

( 2 ) 実線は固定された辺を意味する ( 辺 4 1 4 )、あるいはコンテナの幅・高さが固定されている ( コンテナ 4 0 8 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。

( 3 ) アンカーは辺そして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。当然アンカーは固定される。図 4 のアイコン 4 0 9 は、交差する辺 4 1 4 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。

( 4 ) スライダーは関係付けられた辺が固定されていることおよび平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 4 1 3 はコンテナ 4 0 8 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可能性がある。

【 0 0 7 6 】

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。

【 0 0 7 7 】

なお、上述したようにコンテナの幅・高さの基本値・最小値・最大値の設定は、ダイアログウィンドウに表示される。

【 0 0 7 8 】

図 5 の ( A ) で、コンテナ 5 0 1 は幅・高さ両方が固定されていない。固定された辺 5 0 3 は実線で表現される。可変の辺 5 0 4 は点線で表現される。拡張アイコン 5 0 5 は、隣接する辺 5 0 4 が可変であることを示している。

10

20

30

40

50

## 【 0 0 7 9 】

図 5 の ( B ) で、コンテナ 5 0 1 は幅・高さ両方が可変である。アンカーアイコン 5 0 6 は、交差している両方の辺 5 0 3 が固定されていることを意味する。

## 【 0 0 8 0 】

図 5 の ( C ) で、コンテナ 5 0 1 はコンテナの拡大あるいは縮小が、アンカーアイコン 5 0 7 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であり、幅・高さ両方が可変である。

## 【 0 0 8 1 】

図 5 の ( D ) で、コンテナ 5 0 1 は上辺 5 0 8 が固定されていることを除けば、幅・高さ両方が可変である。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。そしてコンテナの左・右辺が、アンカーアイコンを通る中心軸 ( 垂直軸 ) の周りを、拡大・縮小することを示す。

## 【 0 0 8 2 】

## [ リンク ]

リンクは、コンテナ同士を関連付けるためのものである。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レイアウトを計算する。図 4 の 4 1 2 で示されているものがリンクであり、この図ではコンテナ 4 0 7 と 4 0 8 とを関連づけている。リンクの設定方法および、リンクで関連付けられたコンテナのレイアウト計算方法については、後述する。

## 【 0 0 8 3 】

## [ 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 0 8 4 】

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ー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 0 8 5 】

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、ユーザインター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。また、図 7 の ( A ) ~ ( C ) は、リンクを設定する際のユーザインターフェースの遷移例を示している。

## 【 0 0 8 6 】

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 ) 。図 7 の ( 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 0 8 7 】

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 つのコンテナのうち的一方 ( 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 のユーザインターフェース 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。

## 【 0 0 8 8 】

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 の ( 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 0 8 9 】

次に、ユーザは、図 7 の ( 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、ユーザインターフェース 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。

【 0 0 9 0 】

レイアウト編集アプリケーション 1 2 1 は、ステップ S 6 0 3 で選択された第 1 のコンテナと、ステップ S 6 0 4 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。

【 0 0 9 1 】

こうして、ユーザにより選択された 2 つのコンテナ 7 0 1 、 7 0 2 の間にリンクが設定されると、リンク 7 0 7 を表示する ( ステップ S 6 0 5 ) 。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 の ( C ) の状態になる ( ステップ S 6 0 6 ) 。

10

【 0 0 9 2 】

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 の ( C ) において、7 0 8 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。

【 0 0 9 3 】

[ レイアウト計算方法 ( 全体フロー ) ]

図 8 はレイアウト計算のフローを示している。なお、以下のフローチャートにおける処理は、実際にはホストコンピュータ内のプロセッサ 1 3 5 の制御によって、各種演算や情報

20

【 0 0 9 4 】

実施形態 1 のレイアウト編集アプリケーション 1 2 1 は、ユーザインターフェース 1 0 3 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と、レイアウトエンジン 1 0 5 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に分けられる。

【 0 0 9 5 】

このプレビューモードでは実際のレコードが流し込ま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 1 0 5 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。

30

【 0 0 9 6 】

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。

【 0 0 9 7 】

まず、レイアウト編集アプリケーション 1 2 1 はプレビューモードが選択されていることを認識する ( ステップ S 8 0 1 ) 。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 1 2 1 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 S 8 0 2 )

40

【 0 0 9 8 】

各コンテナへコンテンツデータを流し込むと、レイアウト編集アプリケーション 1 2 1 は、そのコンテンツデータ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 S 8 0 3 ) 。尚、ステップ S 8 0 3 におけるレイアウト計算の詳細については後述する。なお、レイアウト調整はレイアウト計算の結果を受けて実行されるものであり、レイアウト計算と一体として行われる処理である。

【 0 0 9 9 】

そして、レイアウト編集アプリケーション 1 2 1 は、ステップ S 8 0 3 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 S 8 0 4 ) 。レイアウト編集アプリケーション 1 2 1 は、他のレコードについてもプレビューを行うかどうかをユーザの指示により

50



判定する（ステップ S 8 0 5）。ステップ S 8 0 5 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 S 8 0 5 で N O）、プレビューモードを終了する（ステップ S 8 0 7）。

【 0 1 0 0 】

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 S 8 0 5 で Y E S）、レイアウト編集アプリケーション 1 2 1 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 S 8 0 6）。

【 0 1 0 1 】

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 S 8 0 4 は存在せず、ステップ S 8 0 5 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 S 8 0 3 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。

【 0 1 0 2 】

[ レイアウト計算方法（詳細フロー） ]

図 9 はレイアウト計算の詳細を示したフロー図である。

【 0 1 0 3 】

まず、レイアウト編集アプリケーション 1 2 1 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 S 9 0 1）。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。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ため、例えば図 1 1 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、各コンテナにリンクが設定されている。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。したがって、コンテナ A および B が集合 1、コンテナ C および D が集合 2 となる。前述した通りであるが、1 1 0 1 はアンカー、1 1 0 2 は固定された辺、1 1 0 3 はコントローラー、1 1 0 4 は可変の辺の変化方向を示している矢印、1 1 0 5 は可変の辺、1 1 0 6 はリンク、そして1 1 0 7 はスライダーを示している。

【 0 1 0 4 】

次に、レイアウト編集アプリケーション 1 2 1 は、ステップ S 9 0 1 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の集合を一つ選択する（ステップ S 9 0 2）。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。

【 0 1 0 5 】

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。

【 0 1 0 6 】

具体的には、レイアウト編集アプリケーション 1 2 1 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情報により判定でき、コンテナ情報は Memory 1 3 6 内に図 2 5 のような形式で保存されている。

【 0 1 0 7 】

次にレイアウト編集アプリケーション 1 2 1 は、コンテナ A に流し込まれるデータを読み込み、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、および解像度）がコンテナ A の制約を受けない場合の大きさになる。

【 0 1 0 8 】

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 のコンテナ情報で指定されているフォントタイプ、フォントサイズ、文字ピッチ、行ピッチなどの文字情報に基づいて、コンテナ A に流し込まれるべきデータ量が計

10

20

30

40

50

算できる。

【0109】

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11の例では、コンテナAは、左上およ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）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。

【0110】

すべての文字を流し込めると判定された場合、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、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の幅がどれだけ横方向に拡大されると、計算したデータ量のテキストコンテンツ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。

10

【0111】

次に、レイアウト編集アプリケーション121は、コンテンツデータの流し込みによって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）。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。

20

【0112】

レイアウト編集アプリケーション121は、ステップS903で算出した最適化におけるコンテナの集合のサイズ、つまり、コンテンツの流し込みによって変化したコンテナ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し、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11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。

30

【0113】

続いて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がレイアウト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。

【0114】

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。なお、可変リンクについては後述する。

40

【0115】

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。

【0116】

また、図10はそのときのUI表示例を示した図である。図10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。

50

## 【 0 1 1 7 】

図 1 0 の ( 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。1 0 0 1 と 1 0 0 2 はアンカー、1 0 0 3 と 1 0 0 4 は固定された辺、1 0 0 5 は可変の辺、1 0 0 6 は可変の辺の変化方向を示している矢印、1 0 0 8 はリンクを示している。この状態において次々にレコードに含まれるコンテンツデータを流し込んでいく。

## 【 0 1 1 8 】

図 1 0 の ( B ) は、次に流し込むべきレコードに含まれるコンテンツデータのサイズを重ねて示している。1 0 0 9 はそれぞれのコンテナに挿入されるコンテンツデータのサイズを表している。図 1 0 の ( B ) では流し込まれたコンテンツデータのサイズが収まりきらない様子を示しており、これをコンテナ間で調整するためにレイアウト計算が実行される。

10

## 【 0 1 1 9 】

図 1 0 の ( 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 1 0 の ( C ) で示されるように、図 1 0 の ( B ) で示した挿入されるコンテンツサイズ 1 0 0 9 と計算後のコンテンツサイズ 1 0 1 0 は、双方において同等な差異がある。

## 【 0 1 2 0 】

## [ 可変リンクの設定 ]

図 1 2 では一般的な可変リンクが設定された状態を表している。図 4 と同様にアプリケーションウィンドウ 3 0 1 とツールバー 3 0 3 があり、ドキュメントテンプレート 3 0 9 上にコンテナ 1 2 0 3 とコンテナ 1 2 0 4 が存在する。それぞれのコンテナは典型的にアンカーアイコン 1 2 0 1、アンカーアイコン 1 2 0 2 と固定された辺 1 2 0 5、辺 1 2 0 6 から成り立つ。コンテナ 1 2 0 3 と 1 2 0 4 の間には可変リンク 1 2 0 9 があり、それぞれのコンテナを結んでいる。コンテナ 1 2 0 3 とコンテナ 1 2 0 4 の間にはリンクが設定されているのでそれぞれの右辺 1 2 0 7 と左辺 1 2 0 8 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 1 2 1 0、インジケータ 1 2 1 1 が表示され、これは辺 1 2 0 7 と辺 1 2 0 8 が可変であることを示している。

20

## 【 0 1 2 1 】

また図 1 4 はリンク 1 2 0 9 の情報をセットするためのダイアログウィンドウ 1 4 0 1 の例である。このダイアログは典型的にタイトルバー 1 4 0 2、ツールボタン 1 4 0 3、ダイアログウィンドウの開閉を行うボタン 1 4 0 4、各種の情報をセットするエリア 1 4 0 9 で構成されている。このダイアログウィンドウではリンクタイプが可変 ( 1 4 0 7 ) あるいは固定 ( 1 4 0 6 ) の選択や、可変の場合、リンクの長さの最大値 ( 1 4 1 0 ) そして最小値 ( 1 4 1 2 )、また基準となる距離 ( 1 4 1 1 ) を設定することができる。

30

## 【 0 1 2 2 】

図 1 3 は可変リンクの設定方法についてのフローチャートである。なお、これらの処理はホストコンピュータ内のプロセッサ 1 3 5 の処理によって実行される。

## 【 0 1 2 3 】

例えば、図 1 1 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 1 3 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 1 1 0 6 ( 図 1 1 ) の状態から可変サイズのリンク 1 2 0 9 ( 図 1 2 ) へと変更することができる。

40

## 【 0 1 2 4 】

まず、レイアウト編集アプリケーション 1 2 1 はマウスにより選択されたリンク ( 例えば、図 1 1 のリンク 1 1 0 6 ) を選択状態とする ( ステップ S 1 3 0 2 )。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と、レイアウト編集アプリケーション 1 2 1 のユーザインターフェース 1 0 3 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ィンドウ 1 4 0 1 ( 図 1 4 ) を表示する ( ステップ S 1 3 0 3

50

）。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 1 3 3 の右クリックあるいはキーボード 1 3 2 の特定のキー操作等、いかなるものであってもよい。表示されるダイアログウィンドウ 1 4 0 1 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 1 1 0 6 が選択されたので、この段階ではリンクサイズは固定であり、リンクタイプフィールド 1 4 0 5 においては固定長を示す距離 1 4 0 6 が選択されていることとなる（なお、図 1 4 では可変リンク設定後の状態を表しているため 1 4 0 7 にチェックが入っているが、リンク 1 1 0 6 を選択した時点では固定長にチェックが入っている。）

#### 【 0 1 2 5 】

レイアウト編集アプリケーションはマウス等によってダイアログウィンドウ 1 4 0 1 にある固定リンクまたは可変リンクの選択状態を判定し、可変リンクを表す 1 4 0 7 が選択されていることを認識する（ステップ S 1 3 0 4 ）。これにより、リンク距離フィールド 1 4 0 8 内に配置されている最大距離フィールド 1 4 1 2、最小距離フィールド 1 4 1 0、基準値フィールド 1 4 1 1 が有効になり、数値の設定が可能となる。ユーザにより可変リンクの長さの最大値を最大距離フィールド 1 4 1 2 に、最小値を最小距離フィールド 1 4 1 0 に、現在の値を基準値フィールド 1 4 1 1 に設定が行われ、レイアウト編集アプリケーションが認識する（ステップ S 1 3 0 5 ）。

#### 【 0 1 2 6 】

設定を終えると、ユーザは一般的なダイアログウィンドウ開閉ボタン 1 4 0 4 によって当該設定の適用を指示する。ユーザインターフェース 1 0 3 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 1 2 のリンク 1 2 0 9 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 S 1 3 0 6 ）。

#### 【 0 1 2 7 】

以上のダイアログウィンドウ 1 4 0 1 の設定情報は、例えば、Memory 1 3 6 に格納される。

#### 【 0 1 2 8 】

次に、レイアウト結果の一例について、図 1 5 及び図 1 6 を用いて説明する。

#### 【 0 1 2 9 】

図 1 5 は本実施形態における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。

#### 【 0 1 3 0 】

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 1 5 において、コンテナ 1 5 0 1 とコンテナ 1 5 0 2 に予め設定されたコンテナサイズとは異なる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コンテンツデータの大きさを最適とみなし、コンテナ 1 5 0 1 は挿入されたイメージサイズになる枠 1 5 0 4（最適コンテナサイズ）に近づこうと右方向へ、同様にコンテナ 1 5 0 2 は挿入されたイメージサイズになる枠 1 5 0 5（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。

#### 【 0 1 3 1 】

しかし、コンテナ 1 5 0 1 では左辺に対してアンカーが、コンテナ 1 5 0 2 では右辺に対してアンカーがそれぞれ設定されているため移動することができない。そのため、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 1 5 0 1 とコンテナ 1 5 0 2 の間には固定サイズのリンク 1 5 0 3 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 1 5 0 1 とコンテナ 1 5 0 2 のサイズが変更されることになる。

#### 【 0 1 3 2 】

その結果、コンテナ 1 5 0 1 とコンテナ 1 5 0 2 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 1 5 に示すように、最適なサイズ（枠 1 5 0 4、枠 1 5 0 5）よりも小さくなってしまふ。即ち、リンク 1 5 0 3 のサイズが固定であるため、コンテナ 1 5 0 1 とコンテナ 1 5 0 2 は最適サイズを達成できない。

10

20

30

40

50

## 【 0 1 3 3 】

一方、図 1 6 は図 1 5 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。

## 【 0 1 3 4 】

この場合、上記の例で、コンテナ 1 5 0 1 とコンテナ 1 5 0 2 の間には図示の通り可変サイズのリンク 1 6 0 3 が設定されている。従って、コンテナ 1 5 0 1 とコンテナ 1 5 0 2 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 1 5 0 1 とコンテナ 1 5 0 2 のサイズを、図 1 5 の例より大きくすることができる。

## 【 0 1 3 5 】

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 1 6 はこの結果を示しており、図 1 2 の可変リンク 1 2 0 9 はレイアウト計算の結果、可変リンク 1 6 0 3 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 1 5 0 1 とコンテナ 1 5 0 2 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。

以上が、本発明の前提となる基本構成である。

## 【 0 1 3 6 】

## [ ページ間リンクの設定 ]

図 1 7 は複数ページのドキュメントテンプレートを持つレイアウトデザインを表している。

## 【 0 1 3 7 】

図 3 と同様にアプリケーションウィンドウ 3 0 1、ツールバー 3 0 3、メニューアイテム 3 0 4、ワークエリア 3 0 6 がある。ドキュメントテンプレート 3 0 9 には二つのページ 1 7 0 1、1 7 0 2 がある。ページ 1 7 0 1 には、図 3 の 3 1 0 に相当するページマージンガイド 1 7 0 3、ページ 1 7 0 2 には、ページマージンガイド 1 7 0 4 がある。

## 【 0 1 3 8 】

ドキュメントテンプレート 3 0 9 のページ 1 7 0 1 上には、コンテナ 1 7 0 5、コンテナ 1 7 0 6、コンテナ 1 7 0 7 が、ドキュメントテンプレート 3 0 9 のページ 1 7 0 2 上には、コンテナ 1 7 0 8、コンテナ 1 7 0 9 が存在する。コンテナ 1 7 0 5 とコンテナ 1 7 0 6 の間にはリンク 1 7 1 0 が、コンテナ 1 7 0 6 とコンテナ 1 7 0 7 の間にはリンク 1 7 1 1 が、コンテナ 1 7 0 8 とコンテナ 1 7 0 9 の間にはリンク 1 7 1 2 がそれぞれ設定されており、コンテナ 1 7 0 5、1 7 0 8 にはアンカー 1 7 1 0 が設定されている。なお、ユーザによってドキュメントテンプレート 3 0 9 上に設定された情報はプロセッサ 1 3 5 が認識して、Memory 1 3 6 に保存しており、各情報は例えば図 2 5 および図 3 1 のような形式で保存されている。

## 【 0 1 3 9 】

図 1 8 において、ページを跨ったリンクが設定された場合のレイアウトデザインが表示部に表示されている。図 1 7 と異なる点のみを記載すると、コンテナ 1 7 0 7 とコンテナ 1 7 0 8 の間にはページを跨るリンク 1 8 0 1 が設定され、それに基づきコンテナ 1 7 0 8 に設定されていたアンカー 1 7 1 0 が解除されている。その他の設定は、図 1 7 の設定と同様である。なお、ページ跨ぎリンク 1 8 0 1 は通常のコンテナ間を関連付けるリンクとは属性が異なるため、通常のリンクとは識別可能となるような形態で表示される。

## 【 0 1 4 0 】

ページ跨ぎリンクについて説明をする。コンテナ間を関連付ける通常のリンクは図 9 におけるレイアウト調整時に影響を与えることがあるが、ページ跨ぎリンクは現状ではレイアウト調整に影響を与えるものではない。しかし、移動処理または後述するページ跨ぎ処理により、ページ跨ぎリンクで関連付けられたコンテナが同一ページ領域に配置された場合に、通常のレイアウト調整に影響を与えるリンクへと属性を変更する。そのため、ページ跨ぎリンクには、レイアウト調整に影響を与えるリンク属性に変更された場合、どのようなリンク属性となるのか予め設定しておく必要がある。そこで、ページ跨ぎリンクに対して所定の操作が行われると、レイアウト編集アプリケーション 1 2 1 のユーザインターフェース 1 0 3 は、図 3 8 のようなページ跨ぎリンクの属性を設定するためのダイアログ

ウィンドウを表示する。

【 0 1 4 1 】

図 3 8 では、ページ跨ぎリンクから通常のレイアウト調整に影響を与えるリンクに属性が変更された場合、固定リンクとするか可変リンクとするかを選択する。この選択処理により固定リンクが選択された場合には、固定サイズを設定し、可変リンクが選択された場合には、上述した図 1 4 と同様に最大サイズ、最小サイズ、基準値サイズを設定する。後述するページ移動処理またはページ跨ぎ処理が行われ、ページ跨ぎリンクにより関連付けられているコンテナが同一ページ領域に配置された場合、図 3 8 における設定に基づくレイアウト調整に影響を与えるリンクへと変更される。なお、図 3 8 においては固定リンクが選択されたため、可変リンクの最大サイズ、最小サイズ、基準値サイズを設定できないように表示変更されている。また、本実施例では所定の処理により図 3 8 のようなダイアログウィンドウが表示されると記載したが、ページ跨ぎリンクが設定されたことに応じて自動的に表示されるようにしても構わない。

10

【 0 1 4 2 】

図 1 9 はページを跨ってリンクを設定する際のフローチャートを示している。なお、以下のフローチャートにおける処理は、ホストコンピュータ内のプロセッサ 1 3 5 の制御によって、各種演算や情報の入出力等を実行するものである。

【 0 1 4 3 】

レイアウト編集アプリケーション 1 2 1 は、マウス等によりリンクツールボタン 4 0 6 が選択されたことを認識すると、リンクツール 4 0 6 を選択状態にする ( S 1 9 0 1 ) 。

20

【 0 1 4 4 】

リンクツールが選択されている状態から、リンクを設定するべく所望のコンテナが選択されると、レイアウト編集アプリケーション 1 2 1 が選択されたコンテナを認識する ( S 1 9 0 2 ) 。次に S 1 9 0 2 で選択されたコンテナと関連づけるべきコンテナが選択されたことをレイアウト編集アプリケーション 1 2 1 が認識する ( S 1 9 0 3 ) 。詳細には、S 1 9 0 2 において最初に選択されたコンテナの情報を一時的に Memory 1 3 6 に保存しておき、続いて他方のコンテナが選択されたことが認識されると、先程保存しておいたコンテナとの関連付けを行い、ドキュメントテンプレート 3 0 9 上にリンクが表示されることとなる。なお、図 1 8 において、ページ 1 7 0 1 のコンテナ 1 7 0 7 およびページ 1 7 0 2 のコンテナ 1 7 0 8 が選択されたものとする。

30

【 0 1 4 5 】

S 1 9 0 2 および S 1 9 0 3 において、リンクによって関連付ける 2 つのコンテナが選択 ( Memory 上に保存 ) された場合、レイアウト編集アプリケーション 1 2 1 は、S 1 9 0 2 で選択されたコンテナ ( 本実施例ではコンテナ 1 7 0 7 ) と S 1 9 0 3 で選択されたコンテナ ( 本実施例ではコンテナ 1 7 0 8 ) が同じページ内にあるか否かを判定する ( S 1 9 0 4 ) 。詳細には各コンテナが選択された時点でレイアウト編集アプリケーションを構成するコンポーネントであるレイアウトエンジン 1 0 5 が、それぞれのコンテナの位置を計算して図 2 5 のような形式で Memory 1 3 6 に保存することができる。レイアウト編集アプリケーション 1 2 1 は、Memory 1 3 6 の「ページ初期配置ページ」を参照することによって、同じページ内にあるか否かの判定が可能となる。ここでレイアウト編集アプリケーションが Memory 1 3 6 を参照した結果、2 つのコンテナ ( 本実施例ではコンテナ 1 7 0 7 および 1 7 0 8 ) が同じページ内にあると判定した場合は、図 7 の ( C ) に示したような通常のリンク設定が行われディスプレイ上に表示される ( S 1 9 0 5 ) 。

40

【 0 1 4 6 】

一方、S 1 9 0 4 において関連付けられたコンテナが同一ページ内ではないとレイアウト編集アプリケーション 1 2 1 が判定した場合、図 1 8 に示すリンク 1 8 0 1 のようにページを跨ってリンクが設定され、表示部に表示される ( S 1 9 0 6 ) 。

【 0 1 4 7 】

なお、S 1 9 0 5 および S 1 9 0 6 のようにリンクが設定された時点で各リンク情報が

50

図 3 1 のような形式で Memory 1 3 6 に保存されることとなる。

【 0 1 4 8 】

本実施例では図 1 8 で示したページ跨りリンク 1 8 0 1 が設定された場合、図 3 2 のようなプロパティが表示され、ここで「コンテナ属性自動設定」を選択することにより図 2 0 の処理に入ることとなる。なお、以下の処理はホストコンピュータに含まれるプロセッサ 1 3 5 によって実行される。

【 0 1 4 9 】

レイアウト編集アプリケーション 1 2 1 は、ユーザによるリンク設定操作により設定されたリンクがページ跨りリンクであるか否かの判定をする ( S 2 0 0 1 )。詳細には、リンクが設定された時点で各リンク情報が図 3 1 のように Memory 1 3 6 に保存されており、その情報を参照することによってリンクが同一ページ内に配置されレイアウト調整に影響を与えるリンク ( ページ跨ぎ O F F ) であるか、複数ページに跨ったページ跨りリンク ( ページ跨ぎ O N ) であるかを認識することが可能となる。 S 2 0 0 1 において、ページ跨りリンク ( ページ跨ぎ O N ) が設定されたと判定された場合、一方のリンク端として選択されたコンテナに対して「ページを移動して自動レイアウト可能」という属性を設定する ( S 2 0 0 2 )。同様に他方のリンク端として選択されたコンテナの属性に「ページを移動して自動レイアウト可能」という属性を設定する ( S 2 0 0 3 )。なお、図 1 8 の例では、コンテナ 1 7 0 7 およびコンテナ 1 7 0 8 の属性に「ページを移動して自動レイアウト可能」が設定されたこととなる。

【 0 1 5 0 】

通常、コンテナが初期作成された時点では該コンテナに対して「現在配置されているページ内でレイアウトする」という属性が Memory 1 3 6 に記憶されるが、 S 2 0 0 1 でページ跨りリンクが設定された場合、リンク端として選択されたコンテナの属性が「ページを移動してレイアウト可能である」という属性に自動的に変更され、保存される。なおコンテナ属性は、図 2 5 に示すコンテナ情報の 1 つとして保存されている。図 2 5 にある「レイアウト方法」は各コンテナ ( データ領域 ) に流し込まれたコンテンツデータに応じて、予め設定したページ内でレイアウトされるか ( 移動不可 )、または予め設定したページからの移動を許すか ( 移動可能 ) 等を指定するための情報であり、本願のコンテナ属性を指す。「コンテナ種」はテキストデータまたはイメージデータのどちらのデータが流し込まれるかを示すものであり、「可変 / 固定」は流し込まれるコンテンツデータに応じて、コンテナサイズが可変であるか固定であるかを表すものである。なお、可変の場合は、コンテナサイズの可変範囲についても記憶されている。「コンテナサイズ」は、現状のコンテナサイズを示すものである。また、「初期配置ページ」とはコンテナが予め設定されたページ領域を示すものであり、「コンテナ集合」とはリンクによって関連付けられている集合体を示すものである。「レイアウト方法」として移動可能または移動不可が保存されているが、後述するように「ページ跨ぎ」というレイアウト方法が選択された場合は、「ページ跨ぎ」がレイアウト方法として保存される。

【 0 1 5 1 】

なお本実施例では、ページ跨りリンクが設定されることに応じて図 3 2 のプロパティが表示され、「コンテナ属性自動設定」が選択された場合、自動で「ページを移動して自動レイアウト可能」という属性が設定されると記載したが、ページ跨りリンクが設定されたのに応じて、別のレイアウト方法 ( コンテナ属性 ) が自動設定されても構わない。また、自動設定の場合は、どのコンテナ属性を設定するかをユーザが予め設定するようにしても良い。

【 0 1 5 2 】

[ ページ跨りリンク設定時の自動レイアウト方法 ]

図 2 1 はページを跨ってリンクが設定されたドキュメントテンプレートにコンテンツデータが流し込まれた際の動作フローである。なお、各ステップの処理はホストコンピュータ内のプロセッサ 1 3 5 によって処理が実行される。

【 0 1 5 3 】

図 18 に示したドキュメントテンプレート 309 は、外部データソースであるデータベース 119 との連係が設定されており、コンテナ 1705 ~ コンテナ 1709 は、それぞれデータベース 119 にある各レコードが所持するコンテンツデータとの連係が設定されている。

#### 【0154】

まず、レイアウト編集アプリケーション 121 はドキュメントテンプレート上に配置されている全てのリンクの属性をチェックする (S2101)。各リンクの属性はホストコンピュータ内の Memory 136 内に図 31 のような形式で保存されているので、Memory 136 を参照することによって属性を判定することが可能となる。なお、図 31 は図 18 にある各リンクの属性の保存状態を示したものである。図 31 にある「関連コンテナ」はリンクが関連付けているコンテナを表している。また、「ページ跨ぎ ON/OFF」はリンクがページを跨いで設定されているか否かを表しており、「可変/固定」はリンクが固定リンクであるか、可変リンクであるかを表すものである。

#### 【0155】

S2101 のリンク属性のチェックにより、レイアウト編集アプリケーション 121 がページ跨ぎリンクであることを示す「ページ跨り ON」という属性のリンクがあるか否かの判定を行う (S2102)。S2102 で、「ページ跨り ON」というリンクがないと判定された場合は、通常どおりのレイアウト調整を行うので図 9 のフローチャートへと進む。また、S2102 で「ページ跨ぎ ON」というリンクがあると判定された場合は、図 30 に示すページ跨ぎ用編集ページ 3000 が表示される (S2103)。

#### 【0156】

図 30 に示すページ跨ぎ用編集ページ 3000 について説明する。なお、図 30 は図 18 のプレビューモードから遷移した図である。ページ跨ぎ用編集ページへ遷移前の表示画面では各ページ間には空白があり、異なるページであることが表現されている。例えば図 18 ではページ 1701 およびページ 1702 の間に空白があり、ドキュメントテンプレート 309 上に 2 つのページが存在することがユーザにとって認識できるような構成になっている。これに対してページ跨ぎ用編集ページ 3000 では、図 30 のように複数ページを 1 つの編集領域として扱えるような形式に表示変更されており、さらに各ページの位置を明確にするため境界線 3001 が表示される。具体的には図 18 のように 2 ページから構成されているドキュメントテンプレートをページ跨ぎ用編集ページ 3000 に遷移した結果、図 30 のようになり、真中には 2 つのページで構成されていることを示すために境界線 3001 が表示される。

#### 【0157】

図 21 のフローチャートに戻る。現在、処理しているドキュメントテンプレート 309 と結びつけられているデータベース 119 から 1 つ目のレコードを流し込むため、レイアウト編集アプリケーション 121 を構成するコンポーネントの 1 つであるユーザインターフェース 103 が、レコード数の初期化 ( $n = 1$ ) を行い (S2104)、データベース 119 から該当するレコードに含まれるコンテンツデータを、対応付けられている各コンテナに流し込む (S2105)。

#### 【0158】

S2105 でコンテンツデータが流し込まれた場合、レイアウト編集アプリケーションは各コンテナに設定されている制限条件またはコンテンツデータのサイズに基づいて図 9 に示したレイアウト計算さらに必要に応じてレイアウト調整を行う。なお、各コンテナに設定されている制限条件についてはレイアウト編集アプリケーションを構成するコンポーネントの 1 つである UI モデルアナライザ 104 が、ユーザによって設定された各コンテナの制限条件をレイアウトエンジン 105 が認識できる内部形式に変換しているため、レイアウト編集アプリケーション 121 は各コンテナに設定された制限情報を認識することが可能となる。

#### 【0159】

上記レイアウト計算およびレイアウト調整はリンクによって関連付けられているコンテ

10

20

30

40

50



ナの集合ごとに行うため、レイアウト編集アプリケーションは集合の初期化 (  $m = 1$  ) を行い ( S 2 1 0 6 )、1 番目の集合からレイアウト計算処理を実行する ( S 2 1 0 7 )。詳細には、図 9 に示したものと同様であるので省略する。なお、本願に記載のレイアウト計算によって各コンテナ ( データ領域 ) に流し込まれるコンテンツデータのレイアウト調整することが可能となる。

【 0 1 6 0 】

S 2 1 0 7 のレイアウト計算処理およびレイアウト調整はコンテナの集合ごとに行われるのでコンテナの全集合の計算が終了したか否か、すなわち現在処理をしているドキュメントテンプレートに配置された全てのコンテナに対してレイアウト最適化の計算が行われたか否かを判定する ( S 2 1 0 8 )。

10

【 0 1 6 1 】

S 2 1 0 8 によって全ての集合に対してレイアウト計算が終了したと判定された場合、予め設定されたページ領域をはみ出すコンテナがあるか否かの判定を行う。 ( S 2 1 0 9 )。詳細には、レイアウト編集アプリケーション 1 2 1 を構成するコンポーネントの 1 つであるレイアウトエンジン 1 0 5 によって境界線の位置および各コンテナの位置が計算され、認識される。この処理によって境界線およびドキュメントテンプレートに配置されている各コンテナの位置を把握することが可能となり、Memory 1 3 6 に保存されている各コンテナ情報と比較することによって、予め設定されたページをはみ出すコンテナがあるか否かの判定をすることが可能となる。

【 0 1 6 2 】

20

S 2 1 0 9 で予め設定されたページ領域をはみ出すコンテナがあると判定された場合、レイアウト編集アプリケーションが、該コンテナの属性を調べる ( S 2 1 1 0 )。詳細には Memory 1 2 6 に記憶されている図 2 5 のようなコンテナ属性を参照することによりコンテナ属性を調べることを可能となる。なお、S 2 1 0 9 において、予め設定されたページ領域からはみ出すコンテナがないと判定された場合、これ以降の処理は必要ないため、S 2 1 1 8 へと進む。

【 0 1 6 3 】

S 2 1 1 0 で判定されたコンテナの属性が「ページを移動してレイアウト可能」コンテナであると判定された場合、レイアウト編集アプリケーション 1 2 1 が該コンテナの境界線から次ページへのはみ出し量を計算する ( S 2 1 1 1 )。以下に 1 つの計算処理例を示す。計算処理としては座標値を用いること等が考えられる。図 3 0 においてレイアウト調整を行った結果、コンテナ 1 7 0 7 が境界線 3 0 0 1 上に配置されたとする。ここで、コンテナ 1 7 0 7 および境界線 3 0 0 1 の座標値をレイアウトエンジンが計算する。例えば、コンテナ 1 7 0 7 の左下の座標が (  $x = 20$ 、 $y = 30$  ) であり、右下の座標が (  $x = 70$ 、 $y = 30$  )、左上 (  $x = 20$ 、 $y = 80$  )、右上 (  $x = 70$ 、 $y = 80$  ) とし、さらに境界線 3 0 0 1 の左端が (  $x = 0$ 、 $y = 50$  )、右端が (  $x = 100$ 、 $y = 50$  ) であるとする。また、Memory 1 3 6 に保存されている図 2 5 からコンテナ 1 7 0 7 の初期配置ページが 1 7 0 1 であることが判断でき、コンテナ 1 7 0 7 は予め設定されたページ領域をはみ出していることが認識される。なお座標値 (  $x$ 、 $y$  ) は、 $x$  が水平方向、 $y$  が垂直方向の位置を示すものとする。さらに、この座標値からレイアウト調整後のコンテナの面積は 2 5 0 0 と計算できる。続いて、境界線からはみ出し量、つまり次ページ ( 本実施例ではページ領域 1 7 0 2 である ) へのはみ出し量を計算する。図 3 0 では垂直方向にはみ出しているため境界線 3 0 0 1 より下にある面積を求めればよい。上記の座標値から計算すると、はみ出し量は 1 5 0 0 となる。つまりコンテナ 1 7 0 1 の面積における半分以上が、次ページ ( 図 1 8 では 1 7 0 2 ) へ、はみ出していると判断される。

30

40

【 0 1 6 4 】

S 2 1 1 1 における計算の結果、レイアウト編集アプリケーション 1 2 1 が該コンテナの次ページへのはみ出し量が半分以上か否かを判定する ( S 2 1 1 2 )。S 2 1 1 2 において、次ページへのはみ出し量が半分以上と判定された場合、該コンテナは次ページへと移動処理が行われる ( S 2 1 1 3 )。詳細には移動すべきコンテナは他のコンテナとの位

50

置関係を保ちつつ、次ページに収まるような位置をレイアウトエンジンが計算し、さらに図38のページ跨ぎリンク設定により設定された属性に基づいて次ページ内に配置されることとなる。なお、境界線がコンテナをちょうど2等分にする位置にあると判定された場合は、ユーザに現ページに残すか、それとも次ページへ送りだすかを判定させるようにしても構わない。

#### 【0165】

また、S2112において、はみ出し量が半分未満であった場合、レイアウト編集アプリケーション121は、該コンテナを初期設定時に配置されたページ内に戻すことができるか否かの判定を行う(S2114)。具体的には、これから戻す対象となるページに配置されているコンテナ情報を調べる。各コンテナはテキストコンテナかイメージコンテナかの情報を持っており、テキストコンテナの場合は図25のようにコンテナ毎に現状のフォントサイズとフォントサイズの変更範囲を持っている。ここで現状フォントサイズと変更範囲を比較して、現状フォントサイズをさらに縮小することが可能であればフォントサイズの縮小を行い、その結果、はみ出したコンテナが収まるだけのスペースが移動対象ページに生じるか否かの判定をレイアウトエンジンが計算することにより実現される。また可変コンテナの場合、図25のようにコンテナサイズの制約(可変範囲)を持っているので、その制約内で拡張することにより対応することも可能である。さらに、フォントサイズの変更およびコンテナサイズの変更を両方行うことも可能である。この際、フォントサイズのみを変更するのか、コンテナサイズのみを変更するのか、両者を変更するのかはユーザによって指定可能である。

#### 【0166】

S2114によって、レイアウト編集アプリケーション121が、初期設定ページに戻すことが可能と判断すれば、境界線をはみ出したコンテナは初期設定されたページ内に残ることとなり、次ページへの移動処理は行わない(S2115)。しかし、S2114の処理で戻すことが不可能と判定された場合、S2113と同様に次ページへの移動処理が行われる。以上の処理によって全てのコンテナが配置されるべきページ領域が決定されるので、レイアウト編集アプリケーションはそれぞれのページ領域において再度レイアウトの最適化を図るためレイアウト計算およびレイアウト調整を実行する(S2116)。S2116における処理も図9のフローチャートに沿って行われるものである。この際、前述した移動処理によりページ跨ぎリンクによって関連付けられているコンテナが同一ページ領域に配置された場合、図38において設定したページ跨ぎリンク属性に変更され、変更後のリンク属性を考慮したレイアウト調整が行われる。なお属性がページ跨ぎリンクから通常のレイアウト調整に影響を与えるリンクへと変更された場合、表示形態も変更される。具体例については後述する。

#### 【0167】

そして、レイアウト編集アプリケーション121は、S2116の再レイアウト調整によって決定された各コンテナのレイアウトされるべき位置をホストコンピュータ内のMemory136に保存する(S2117)。

#### 【0168】

レイアウト編集アプリケーション121は、全レコードの流し込みが終了したか否かを判定し(S2118)、終わっていないと判定された場合は現レコード数にインクリメント処理を施し(S2119)、S2105の処理から再開する。全レコードの流し込みが終了したと判定された場合、図21における処理は終了する。

#### 【0169】

図21におけるフローチャートの処理をより分かりやすく説明するため図18、図30を用いて具体的に説明する。なお、ページ跨ぎリンクに対しては図38の設定ダイアログにより、固定リンクが選択され、固定リンクサイズ「3」が設定されたものとする。

#### 【0170】

図18においてコンテナ1707およびコンテナ1708は、ページを跨ぐリンク1801が設定され、図32のプロパティにおいて「コンテナ属性自動設定」が選択された場

10

20

30

40

50

合、「ページを移動してレイアウト可能」の属性がコンテナ 1707 および 1708 に対して自動的に設定される。図 18 のドキュメントテンプレート 309 上にはリンク属性が「ページ跨ぎ ON」のリンク 1801 が存在するので、図 30 におけるページ跨ぎ用編集ページ 3000 が表示される。ここまでの処理が図 21 における S2101 ~ S2103 の処理に該当する。

#### 【0171】

続いて、データベースから現在対象となっているレコードに含まれるコンテンツデータを対応付けられた各コンテナに流し込み、図 9 のレイアウト計算に基づくレイアウトの調整を行い、その結果コンテナ 1707 が境界線 3001 上に配置されたとする。この場合、コンテナ 1707 と境界線の位置関係およびコンテナ情報（図 25）からコンテナ 1707 は予め設定されたページ 1701 をはみ出すことが認識されるので、コンテナ 1707 のコンテナ属性（次ページ領域である 1702 への移動可能か否か）がチェックされ、次ページ 1702 へのはみ出し量が計算される。

#### 【0172】

コンテナ 1707 の半分以上が次ページへはみ出している場合、コンテナ 1707 は次ページへと移動処理が行われる。またコンテナ 1707 の半分以上が初期設定されたページ 1701 に残った場合、現状のページ 1701 に配置できるか否かの判定を行う。ページ 1701 に配置されているコンテナ 1705 および 1706 についての現状の情報は図 25 のように Memory 136 に保持されている。コンテナ 1705 に関してはこれ以上フォントサイズを縮小することができないが、コンテナ 1706 に関しては現状フォントサイズが「10」であり、フォントサイズを「8」まで縮小することが可能であるので、段階的に縮小していく。フォントサイズを縮小した結果、コンテナ 1707 の全体をページ 1701 に配置することが可能と判断されればページ 1702 への移動処理はなくなる。しかし、縮小してもコンテナ 1707 を配置することが不可能であれば次ページであるページ 1702 へ移動されることとなる。また、ページ領域 1701 に配置されているコンテナが可変設定であれば、可変範囲内で縮小するか、フォントサイズおよびコンテナサイズの両者のサイズを変更することで同様の処理を行っても構わない。ここまでの処理が S2104 ~ S2115 の処理である。

#### 【0173】

以上の処理によって全てのコンテナが配置されるべきページ領域が決定されたので各ページにおいて再度図 9 に示したレイアウト計算によるレイアウトの調整が行われることとなる。この際、コンテナ 1707 がページ領域 1702 に移動した場合、ページ跨ぎリンク 1801 により関連付けられていたコンテナ 1707 および 1708 が同一ページ領域（1702）に配置されることとなるので、ページ跨ぎリンク 1801 は図 38 の設定に基づいて（固定リンク、固定リンクサイズ「3」）通常のレイアウト計算およびレイアウト調整に影響をあたえるリンクへと変更され、レイアウト計算およびレイアウト調整時に考慮されることとなる。

#### 【0174】

図 22 に、コンテナ 1707 がページ 1701 から 1702 へ移動された場合のレイアウト計算に基づくレイアウト調整後の結果を示す。コンテナ 2206 とコンテナ 2207 の間には図 18 のようなページ跨ぎリンク 1801 が設定されていたが、上述する移動処理により同一ページに配置されたため、ページ跨ぎリンク 1801 は図 38 において設定した固定リンク（リンクサイズ「3」）に変更する。その結果、コンテナ 2206 とコンテナ 2207 の間には、リンクサイズ「3」の固定リンクが設定されることとなる。この際、図 18 ではページ跨ぎリンクは、通常のレイアウト計算およびレイアウト調整に影響を与えるリンクとは異なる表示形態であったが、属性が変更されたため図 22 の 1801 A のように表示形態が変更され、レイアウト調整に影響を与えるリンクへと変更されたことが認識できるようになる。

#### 【0175】

ここまでの処理でレイアウト調整が終了したので、その結果をホストコンピュータ内の

10

20

30

40

50

Memory 136 に保存する。そして、全レコードの流し込みが終了するまで、以上の処理を繰り返すこととなる。また、レイアウト結果は図 18 のドキュメントテンプレート上に表示されることとなる。

【0176】

尚、上記の図 21 の動作フローでは省略しているが、前記 S904 で述べたように、レイアウトの最適化後ルールに違反していた場合は、再度ルールを違反しないよう再計算が行なわれる。

【0177】

以上、本実施例を用いることによって複数ページに跨いで設定されたリンクによって関連付けられたコンテナの属性は自動的に設定されるため、ページを跨ぐリンクが複数設定された場合の処理が簡略化される。さらに、該属性を有するコンテナにコンテンツを流し込んだ結果、該コンテナの半分以上が異なるページにはみ出した場合、自動的に次ページへの移動処理が行われ、該コンテナの半分未満しかはみ出さなかった場合には、初期設定されたページ内に該コンテナを配置できるか否かの判定を行い、その判定結果に基づいて次ページへ移動するか否かが決定されるため、複数ページのバランスを考慮したレイアウト処理が可能となる。なお、本実施例では、コンテナのはみ出し量が半分か否かで判断していたが、これに限ることなくユーザが設定した値（または割合など）で次ページへ移動するか否かを判定するようにしても構わない。

【実施例 2】

【0178】

実施例 1 では、図 32 のプロパティにおいて「コンテナ属性自動設定」の場合について記載したが、本実施例ではページを跨ぐリンクが設定されたコンテナの属性を手動で選択する場合について説明する。なお、本実施例では前記実施例 1 とほぼ同様の処理をするので異なる点のみを説明する。

【0179】

本実施例における、図 18 のようなページ跨ぎリンク 1801 が設定された場合、図 32 のプロパティが表示部に表示され、マウス等により「コンテナ属性手動設定」が選択された場合、図 26 のようなプロパティが表示される。これにより、ユーザはコンテナの属性を自由に設定することが可能となる。その際のフローチャートについて図 23 に示す。なお、各工程の処理はホストコンピュータ内のプロセッサ 135 によって処理される。

【0180】

レイアウト編集アプリケーション 121 は、ユーザによるリンク設定操作により設定されたリンクがページ跨りリンクであるか否かの判定をする（S2301）。S2301 において、ページ跨りリンクが設定されていると判定された場合、レイアウト編集アプリケーションは、一方のリンク端として選択されたコンテナに対して属性を設定するため図 32 のプロパティを表示し、図 32 において「コンテナ属性手動設定」が選択された場合、図 26 を表示する（S2302）。

【0181】

S2302 によって表示された図 26 のプロパティ 2601 から、ユーザによって選択されたコンテナ属性をレイアウト編集アプリケーションが認識して、選択されたコンテナ属性をホストコンピュータ内の Memory 上に保存する（S2303）。詳細にはユーザはマウス等のポインティングデバイスを用いて図 26 のプロパティからコンテナ属性の指定を行い、該指定を I/O インターフェース 143 を介してプロセッサ 135 が認識をして、該認識した情報を Memory 136 に保存している。なお、本願におけるマウス等からの選択は同様の処理によって認識される。

【0182】

また、同じく他方のリンク端として選択されたコンテナに対して属性を設定するためにレイアウト編集アプリケーションは図 26 のプロパティを表示する（S2304）。S2304 によって表示されたプロパティからユーザによって選択されたコンテナ属性を認識

して、S 2 3 0 3 と同様の処理を行う ( S 2 3 0 5 )。以上の処理によってユーザの所望とするコンテナ属性を設定することが可能となる。なお、本実施例においてもページ跨ぎリンクに対しては図 3 8 の設定ダイアログにより固定リンクが選択されているものとする。

#### 【 0 1 8 3 】

続いて、本実施例におけるレイアウトの処理フローを図 2 4 に示す。なお、各ステップの処理はホストコンピュータ内のプロセッサ 1 3 5 によって処理が実行される。また、S 2 4 0 1 ~ S 2 4 0 9 は S 2 1 0 1 ~ S 2 1 0 9 の処理と同じであるので、S 2 4 1 0 以降の処理から説明する。

#### 【 0 1 8 4 】

レイアウト編集アプリケーションはページ跨ぎ用編集ページ 3 0 0 0 において予め設定されたページ領域をはみ出したコンテナの属性を調べる ( S 2 4 1 0 )。詳細には、レイアウトエンジンがページ跨ぎ用編集ページ 3 0 0 0 にある各コンテナの位置および境界線 3 0 0 1 の位置を把握することが可能であり、例えば境界線 3 0 0 1 の座標と各コンテナの座標および Memory 1 3 6 に保存されている図 2 5 のコンテナ情報にある「初期配置ページ」を認識することにより、境界線を基準としてコンテナが初期設定されたページ領域とは異なるページ領域へはみ出しているか否かを判定することができる。ここで、はみ出すコンテナがあると判定された場合、レイアウト編集アプリケーションは Memory 1 3 6 にある該コンテナ情報のレイアウト方法 ( コンテナ属性 ) を認識することにより S 2 4 1 0 は実現される。

#### 【 0 1 8 5 】

S 2 4 1 0 において該コンテナの属性が「ページを移動して自動レイアウト可能」であれば、初期設定の段階で設定されたページ跨ぎリンクの他方に設定されたコンテナが属するページ領域への移動処理に入る。この場合の処理は図 2 1 の S 2 1 1 3 以降の処理に移ることで実現されるので、ここでは詳細を省略する。

#### 【 0 1 8 6 】

また、S 2 4 1 0 において該コンテナ属性が「ページを跨って自動レイアウト可能」であると判定された場合 ( S 2 4 1 2 ) は、図 2 8 の処理へ移る。なお、図 2 8 における処理の詳細については後述する。

#### 【 0 1 8 7 】

S 2 4 1 0 において該コンテナ属性が「ページを移動して自動レイアウト不可」と判定された場合 ( S 2 4 1 3 )、そのコンテナに関しては予め配置されたページ内に残ることとなる。しかし、該コンテナは現在、境界線上にあるか境界線を完全に越えて異なるページにはみ出しているため、初期設定されたページ内に戻ることが可能であるか否かの判定をしなければならない ( S 2 4 1 4 )。なお、S 2 4 1 4 の判定処理は図 2 1 における S 2 1 1 4 の処理と同様であるので詳細については省略する。S 2 4 1 4 の結果、初期設定されたページ内に戻ることが可能と判定された場合、該コンテナの移動は無くなり、予め配置されたページ内に残ることとなる ( S 2 4 1 5 )。S 2 4 1 5 により、全コンテナの配置されるページ領域が決定されたので、レイアウト編集アプリケーションは再度図 9 に示すレイアウトの計算処理およびレイアウト調整を実行する ( S 2 4 1 6 )。

#### 【 0 1 8 8 】

また S 2 4 1 4 の処理により、はみ出したコンテナを初期設定されたページに戻すことができないと判定された場合、レイアウト編集アプリケーションはエラー処理とするか否かの判定をユーザに選択させるための UI を表示する ( S 2 4 2 0 )。ここでレイアウト編集アプリケーションが、本レコードについてエラー処理とすることが選択されたのを認識した場合、S 2 4 1 8 の処理へと移る。また、エラー処理としないことが選択されたのを認識した場合、はみ出したコンテナの属性を変更させるべく図 2 6 のプロパティを再表示し ( S 2 4 2 1 )、選択されたコンテナ属性を認識して S 2 4 1 0 の処理へ移る。

#### 【 0 1 8 9 】

S 2 4 1 6 ~ S 2 4 1 9 の処理は図 2 1 における S 2 1 1 6 ~ S 2 1 1 9 の処理と同様

10

20

30

40

50

であるので説明は省略する。

#### 【 0 1 9 0 】

図 2 4 におけるフローチャートの処理をより分かりやすく説明するため図 1 8 および図 3 0 を用いて具体的に説明する。なお、ページ跨ぎリンクに対しては図 3 8 の設定ダイアログにより、固定リンクが選択され、固定リンクサイズ「3」が設定されたものとする。

#### 【 0 1 9 1 】

図 1 8 においてコンテナ 1 7 0 7 およびコンテナ 1 7 0 8 は、ページを跨ぐリンク 1 8 0 1 が設定されているので、コンテナ属性を設定するために図 3 2 のプロパティが表示され、「コンテナ属性手動設定」が選択された場合、図 2 6 のようなプロパティが表示される。今回の設定は、コンテナ 1 7 0 7 の属性が「ページを移動して自動レイアウト不可」であり、コンテナ 1 7 0 8 の属性が「ページを移動して自動レイアウト可能」であるとする。図 1 8 のドキュメントテンプレート上にはリンク属性が「ページ跨ぎ ON」となっているリンク 1 8 0 1 があるので、ページ跨ぎ用編集ページ 3 0 0 0 が表示される。ここまでの処理が図 2 4 における S 2 4 0 1 ~ S 2 4 0 3 の処理に該当する。

#### 【 0 1 9 2 】

続いて、データベースから現在対象となっているレコードにあるコンテンツデータを各コンテナに流し込み、レイアウト計算およびレイアウト調整を行う。その結果、図 3 0 のようにコンテナ 1 7 0 7 が境界線 3 0 0 1 上にレイアウトされているため、該コンテナは予め設定されたページをはみ出していることが判定される。ここで該コンテナ属性が「ページを移動して自動レイアウト不可」であると判定された場合、初期設定されたページに戻るることができるか否かの判定が行われる。

#### 【 0 1 9 3 】

ページ 1 7 0 1 に配置されているコンテナ 1 7 0 5 および 1 7 0 6 についての現状の情報は図 2 5 のように Memory 1 3 6 に保持されている。コンテナ 1 7 0 5 に関してはこれ以上フォントサイズを縮小することができないが、コンテナ 1 7 0 6 に関してはフォントサイズを「8」まで縮小することが可能であるので、現状フォントサイズ「10」から段階的に縮小していく。例えば、フォントサイズを「9」に縮小した結果、コンテナ 1 7 0 7 をページ 1 7 0 1 に配置することができると判定されれば、初期設定ページであるページ 1 7 0 1 に収まることが決定される。また、フォントサイズを「8」まで縮小しても、コンテナ 1 7 0 7 をページ 1 7 0 1 に配置することができないと判定された場合、ページ 1 7 0 1 から可変コンテナを検索する。図 2 5 からページ 1 7 0 1 に配置されているコンテナ 1 7 0 5 および 1 7 0 6 は可変コンテナであることが判明されるため、可変範囲においてコンテナの縮小を試みる。コンテナ 1 7 0 5 に関してはこれ以上縮小することができないが、コンテナ 1 7 0 6 の現状サイズは図 2 5 から「縦 5 横 5」と認識でき、さらにコンテナ 1 7 0 6 のコンテナサイズの可変範囲は「縦 3 ~ 8 横 5 ~ 8」であるため、コンテナ 1 7 0 6 のコンテナサイズを縮小することが可能である。よって、コンテナ 1 7 0 6 のコンテナサイズについても段階的に縮小していく。例えば、コンテナサイズを「縦 4 横 6」に縮小した結果、コンテナ 1 7 0 7 をページ 1 7 0 1 に配置することができると判定されれば、初期設定ページであるページ 1 7 0 1 に収まることが決定される。なお、本実施例ではフォントサイズを変更した後に、コンテナサイズを変更したが、これに限ることは無く、コンテナサイズから縮小しても構わない。

#### 【 0 1 9 4 】

以上の処理によって全コンテナの配置ページ領域が決定されたことを受け、ページ 1 7 0 1 に配置されているコンテナ 1 7 0 5 および 1 7 0 6 およびコンテナ 1 7 0 7 が最適なサイズでページ 1 7 0 1 に収まるように再度図 9 によるレイアウト計算およびレイアウト調整が行われ、同様にページ 1 7 0 2 においては、コンテナ 1 7 0 5 がページ 1 7 0 1に残ったことによりレイアウトの変更が生じるので、同様に再度レイアウト計算およびレイアウト調整が行われる。その結果を図 2 7 に示す。なお、ページ跨ぎリンク 1 8 0 1 に対して図 3 8 により設定処理を行ったが、ページ跨ぎリンク 1 8 0 1 により関連付けられているコンテナ 1 7 0 7 および 1 7 0 8 は異なるページに配置されることが決定されたので

、この場合ページ跨ぎリンク 1 8 0 1 は消去されることとなる。

【 0 1 9 5 】

また、フォントサイズおよびコンテナサイズを縮小してもコンテナ 1 7 0 7 を配置することが不可能であると判定された場合、エラー処理として次のレコードのレイアウト処理に移るか、コンテナ 1 7 0 7 の属性を変更して再度レイアウト処理を行うかが選択される。

【 0 1 9 6 】

ここまでの処理でレイアウト調整が終了したので、その結果をホストコンピュータ内の Memory 1 3 6 に保存する。そして、全レコードの流し込みが終了するまで以上の処理を繰り返すこととなる。

【 0 1 9 7 】

続いて、S 2 4 1 0 のコンテナ属性の判定において、境界線上のコンテナ属性が「ページを跨って自動レイアウト可能」と判定された場合の処理フローを図 2 8 に示す。

【 0 1 9 8 】

レイアウト編集アプリケーションは Memory 1 3 6 を参照した結果、S 2 4 1 0 において境界線 3 0 0 1 上にあるコンテナの属性が「ページを跨って自動レイアウト可能」と判定した場合、境界線 3 0 0 1 上を境にして該コンテナを分割するため、次のページへのはみ出し領域を認識する ( S 2 8 0 1 )。S 2 8 0 1 において、レイアウト編集アプリケーションは認識したはみ出し領域に基づいて分割処理を行い ( S 2 8 0 2 )、分割された各コンテナの領域が配置された各ページにおいて再度レイアウト計算およびレイアウト調整を行う ( S 2 8 0 3 )。分割されたコンテナの一部がページ跨ぎリンクにより関連付けられたコンテナと同一ページ領域に配置されるため、図 3 8 において設定したページ跨ぎリンク属性に変更され、変更後のリンク属性を考慮したレイアウト調整が行われる。

【 0 1 9 9 】

これ以降の処理である S 2 8 0 4 ~ 2 8 0 6 は図 2 1 における S 2 1 1 7 ~ S 2 1 1 9 と同様であるので説明は省略する。なお、はみ出し領域の認識 ( S 2 8 0 1 ) や分割処理 ( S 2 8 0 2 ) の具体的な処理方法は後述する。

【 0 2 0 0 】

図 2 8 におけるフローチャートの処理をより分かりやすく説明するため図 1 8 および図 2 9 および図 3 0 を用いて説明する。なお、ここで説明するのは図 2 4 のコンテナ属性の判定において「ページを跨って自動レイアウト可能」と判定された場合 ( S 2 4 1 3 ) 以降の説明であり、ページ跨ぎリンクに対しては図 3 8 の設定ダイアログにより、固定リンクが選択され、固定リンクサイズ「3」が設定されたものとする。

【 0 2 0 1 】

境界線上のコンテナ 1 7 0 7 の属性が「ページ跨って自動レイアウト可能」と判定された場合、該コンテナのはみ出し量が認識される。このはみ出し量の認識方法の一例としてはレイアウトエンジン 1 0 5 が図 3 0 にあるページ跨ぎ用編集ページの座標を認識することによって実現される。

【 0 2 0 2 】

例えば、コンテナ 1 7 0 7 の左下の座標が ( x = 2 0、y = 3 0 ) であり、右下の座標が ( x = 7 0、y = 3 0 )、左上 ( x = 2 0、y = 8 0 )、右上 ( x = 7 0、y = 8 0 ) とし、さらに境界線 3 0 0 1 の左端が ( x = 0、y = 5 0 )、右端が ( x = 1 0 0、y = 5 0 ) であったとする。なお座標値 ( x、y ) は、x が水平方向、y が垂直方向の位置を示すものとする。この座標値からコンテナ 1 7 0 7 において境界線 3 0 0 1 の左端 ( x = 0、y = 5 0 )、右端 ( x = 1 0 0、y = 5 0 ) よりも上の位置に配置される左上 ( x = 2 0、y = 8 0 )、右上 ( x = 7 0、y = 8 0 )、左下 ( x = 2 0、y = 5 0 )、右下 ( x = 7 0、y = 5 0 ) の領域がページ 1 7 0 1 に残るコンテナの領域であり、これ以外の領域である左上 ( x = 2 0、y = 5 0 )、右上 ( x = 7 0、y = 5 0 )、左下 ( x = 2 0、y = 3 0 )、右下 ( x = 7 0、y = 3 0 ) がはみ出し領域として次ページ 1 7 0 2 に移

10

20

30

40

50

動するはみ出し量として認識される。以上のように認識されたはみ出し領域をMemory 136に保存した後、図30におけるページ跨ぎ用編集ページから図17における通常ページへと表示を切り替える。

【0203】

図17において分割処理を行う。ここで以下の説明を詳細にするため、ページを跨いで自動レイアウトされた結果を図29に示す。

【0204】

まず分割対象となったコンテナ1707を複製し、複製されたコンテナデータをMemory 136に保存しておく。現状ではコンテナ1707は領域2403（図18では1703に相当）をはみ出した状態となっている。そこで、先程認識してMemory 136に保存した、はみ出し領域に基づいてクリッピング処理を行うことにより、はみ出し部分が消去された状態で表示される。つまりコンテナ1707において、はみ出すと判定された2407Bのみがクリッピング処理によって消去されたこととなる。さらに、先程複製したコンテナ1707をMemory 136から読み出して領域2404（図18では1704に相当）に貼り付ける。そして貼り付けられたコンテナ2407は領域2404から上方向にはみ出ることとなるので、Memory 136に保存したはみ出し領域に基づいてクリッピング処理することにより、はみ出し部分が消去された状態で表示されることとなる。つまり、複製されたコンテナ2407のうち2407Aの部分は領域2404から上方向（ページ2401の方向）へはみ出すこととなるので、クリッピング処理により2407Aのみが消去されることとなる。以上の処理によってコンテナ1407の分割処理が終了する。なお、複製されたデータを貼り付ける際、水平方向の位置を初期設定ページ（本実施例ではコンテナ2407A）に合わせ、さらに位置関係を初期設定と保った（コンテナ2407Bの垂直方向にコンテナ2408および2409がある状態）位置に貼り付けられることとなる。

【0205】

以上の処理によって分割されたコンテナ2407Aおよび2407Bを含むドキュメントテンプレート上のコンテナの配置されるべきページ領域が決定されたため、再度図9によるレイアウト計算およびレイアウト調整が行われる。なお、コンテナ2407Bに関してはページ跨ぎリンクにより関連付けられたコンテナ2408と同一ページ領域に配置されたため、ページ跨ぎリンク1801は図38のダイアログにより設定されたリンクサイズ「3」の固定リンクへ1801Aと変更される。この際、図18ではページ跨ぎリンクは、通常のレイアウト計算およびレイアウト調整に影響を与えるリンクとは異なる表示形態であったが、属性が変更されたため図29の1801Aのように表示形態が変更され、レイアウト調整に影響を与えるリンクへと変更されたことが認識できるようになる。

【0206】

レイアウト編集アプリケーションは変更後のリンク1801Aを考慮してレイアウト調整を行う。

【0207】

以上の結果をホストコンピュータ内のMemory 136に保存する。そして、全レコードの流し込みが終了するまで以上の処理を繰り返すこととなり、レイアウト結果が図18のドキュメントテンプレート上に表示されることとなる。また、図28の処理は境界線を境にして分割処理をするものであり、コンテンツデータを流し込んだ結果、コンテナが完全に境界線を越えて次ページの領域に移動してしまった場合、分割処理は行われず図21で示した移動処理と同等の処理になる。

【0208】

尚、上記の図28の動作フローでは省略しているが、前記S904で述べたように、レイアウトの最適化後ルールに違反していた場合は、再度ルールを違反しないよう再計算が行なわれる。

【0209】

以上、本実施例2ではコンテナの属性をユーザが設定できるようにすることで、コンテ

10

20

30

40

50



ソツの流し込みによってコンテナが予め設定されたページをはみ出した場合であっても、ユーザの設定したコンテナ属性に応じて自動レイアウト処理が実行されることとなるのでユーザの意図を反映することができ、全体的にバランスの取れたレイアウトを作成することが可能となる。

### 【実施例 3】

#### 【0210】

実施例 1 および実施例 2 ではページを跨ぐリンクが設定されたコンテナの属性を考慮したレイアウト調整を行ってきた。実施例 3 では図 33 のようにページを跨ぐリンクが設定されたコンテナとレイアウト計算およびレイアウト調整に影響を与える通常のリンクによって関連付けられているコンテナの属性を設定する方法について説明する。

10

#### 【0211】

本実施例におけるページを跨ぐリンクが設定されたコンテナ（図 33 の 1707 および 1708）と関連付けられているコンテナ（図 33 の 1706、3301、1709、3304）の属性を設定する処理を図 35 のフローチャートに示す。なお、各処理はホストコンピュータのプロセッサ 135 によって行われることとなる。

#### 【0212】

レイアウト編集アプリケーション 121 は、ユーザによるリンク設定操作により設定されたリンクがページ跨りリンクであるか否かの判定をする（S3501）。なお、S3501 の判定は実施例 1 の S2001 と同様に Memory 136 からリンクの属性（本実施例では図 34）を参照することによって判定が可能となる。S3501 において、ページ跨りリンク（図 33 の 1801 に相当）であると判定された場合、レイアウト編集アプリケーション 121 は該リンクが設定されたコンテナ（図 33 の 1707 に相当）について実施例 1 または実施例 2 の手段によってコンテナ属性を設定すると共に、該ページ跨りリンクが設定されたコンテナに対して他のリンクが設定されているか否かの判定をする（S3502）。S3502 についてもレイアウト編集アプリケーションが Memory 136 に保存されている図 34 のリンク属性を参照することにより判定可能となる。現在、図 33 においてページ跨りリンクが設定されたのはコンテナ 1707 およびコンテナ 1708 である。このコンテナ 1707 および 1708 に設定されている他のリンクを参照するとリンク 1711、1712、3303、3305 が抽出することができ、これにより他のリンクが設定されていると判定される。レイアウト編集アプリケーションは S3502 で抽出された他のリンクによって関連付けられているコンテナの属性を設定する（S3503）。なお、S3503 の詳細については後述する。レイアウト編集アプリケーションは S3503 の設定を認識して（S3504）、終了する。これ以降は、実施例 1 または実施例 2 と同様にコンテンツデータの流し込みに応じてレイアウト調整を行うこととなる。

20

30

#### 【0213】

S3503 の詳細について図 36 に示す。レイアウト編集アプリケーションは S3502 で認識した他のリンクがどういったリンクであるかを認識する（S3601）。ここで、コンテナ同士を関連付けるリンクが設定されていた場合、該リンクによって関連付けられているコンテナ（図 33 ではコンテナ 1706、1709、3304 に相当）の属性を設定させるために図 26 のようなプロパティを表示する（S3604）。また、S3601 でガイドを介したリンクが設定されていると判定された場合、ページ跨りリンクが設定されたコンテナのコンテナ属性に合わせて自動的に変更する（S3605）。図 33 でいえばページ跨りリンクが設定されたコンテナ 1707 とコンテナ 3301 はガイドを介して関連付けられているので、コンテナ 1707 の属性に合わせてコンテナ 3301 も属性が設定されることとなる。詳細には Memory 136 に保存されているコンテナ 1707 の属性を認識することで、属性を合わせることが可能となる。レイアウト編集アプリケーションは、以上の処理によって属性の設定がされたことを示す図 37 のようなプロパティ 3701 を表示する（S3606）。

40

#### 【0214】

50

ここで図 3 6 の設定処理を分かりやすくするため図 3 3 を用いて具体的に説明する。まず 1 つ目は、ページ跨ぎリンクが設定されたコンテナ 1 7 0 7 と垂直方向のリンクで直接関連付けられているコンテナ 1 7 0 6 の属性を設定する場合、コンテナ 1 7 0 6 は実施例 1 または 2 の手段によってレイアウト調整をした結果、コンテナ 1 7 0 7 が異なるページに移動することが考えられる。その際に、関連付けられているコンテナ 1 7 0 6 も連動して異なるページに移動するのか、予め設定されたページ内 ( 1 7 0 1 ) に残るのかを設定できるように図 2 6 が表示される。2 つ目はコンテナ 1 7 0 7 とガイド機能を介して関連付けられたコンテナ 3 3 0 1 のような場合を想定する。ここでガイド機能について説明する。ガイド機能とは上述したリンクやアンカー機能と同様にレイアウト調整に影響を与える機能であり、例えば図 3 3 のように水平方向に配置されたコンテナ 1 7 0 7 および 3 3 0 1 の垂直方向の位置を調整したり、または垂直方向に配置された複数のコンテナをガイドを介してリンク設定することにより、該複数のコンテナの水平方向の位置を調整することが可能となる。具体的には図 3 3 において、コンテナ 1 7 0 7 とコンテナ 3 3 0 1 は水平方向の位置関係にレイアウトされているが、レイアウト調整の結果コンテナ 1 7 0 7 が垂直方向にレイアウト位置が変更した場合、該変更に関連してコンテナ 3 3 0 1 も垂直方向にレイアウト位置の変更を行う処理をする。このようなガイド機能を介してコンテナ 1 7 0 7 と関連付けられたコンテナ 3 3 0 1 のコンテナ属性をコンテナ 1 7 0 7 の属性に合わせることで、実施例 1 または 2 によりコンテナ 1 7 0 7 が異なるページに移動処理または初期設定のページに残る処理またはコンテナ分割処理と連動して同様の処理がコンテナ 3 3 0 1 にも施されることとなる。例えば、コンテナ 1 7 0 7 がページ 1 7 0 1 からページ 1 7 0 2 に移動した場合、連動してコンテナ 3 3 0 1 もページ 1 7 0 2 へ移動されることとなる。そして、3 つ目がコンテナ 1 7 0 8 とコンテナ 3 3 0 4 のように水平方向のリンクによって直接関連付けられた場合である。このような場合は、コンテナ 3 3 0 1 に対してコンテナ属性の設定を行うために図 2 6 のようなプロパティが表示され、ユーザが選択した設定を認識して、レイアウト調整を行うこととなる。

#### 【 0 2 1 5 】

実施例 3 による処理により、例えば図 3 3 においてはページ跨ぎリンクが設定されたコンテナ 1 7 0 7 がページ領域 1 7 0 1 からページ領域 1 7 0 2 に移動処理された場合、コンテナ 1 7 0 7 と関連付けられているコンテナのレイアウト処理をユーザが所望とする属性設定に応じて、連動したレイアウト処理を行うことが可能となるため、ドキュメント全体のバランスを考慮したレイアウト調整を行うことが可能となる。

#### 【 0 2 1 6 】

上記の設定によってコンテナに属性が設定されたとしても、流し込まれるコンテンツデータによっては全てその属性に基づいたレイアウト処理を行えない場合も考えられる。そのような場合は、エラー処理を行うかコンテナ属性またはテンプレート上の各種の設定を変更して再度レイアウト計算およびレイアウト調整を行うこととなる。

#### 【 0 2 1 7 】

上記実施例 1 乃至 3 においては、予めコンテナ属性を設定しておくように記載したが、各コンテナにコンテンツデータを流し込み、レイアウト調整を行った結果、ページ跨ぎリンクが設定されたコンテナが予め設定されたページからはみ出すと判定された段階で、コンテナ属性を設定するようにしても構わない。さらに、ページを跨ぐリンクが設定されたことにより属性を自動設定していたが、ページを跨ぐリンクが設定されていなくとも、予め各コンテナに属性を設定しておき、コンテンツデータの流し込みによりコンテナが異なるページにはみ出した場合、該属性に応じてはみ出し先のページへ移動または跨ってレイアウトできるようにしても構わない。

#### 【 0 2 1 8 】

なお、本発明は、上記形態の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体 ( 又は記録媒体 ) を、システム又は装置に供給し、そのシステム又は装置のコンピュータ ( 又は C P U や M P U ) が記憶媒体に格納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ことは云うまでもない。

## 【 0 2 1 9 】

この場合、記憶媒体から読み出されたプログラムコード自体が上記実施の形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。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、上記実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（ＯＳ）などが実際の処理の一部又は全部を行い、その処理によって上記実施の形態の機能が実現される場合も含まれることは云うまでもない。

## 【 0 2 2 0 】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カ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記憶媒体に書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるＣＰＵ等が実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは云うまでもない。

## 【 0 2 2 1 】

また、本発明は、上記実施の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がネットワークを介して配信されることにより、システム又は装置のハードディスクやメモリ等の記憶手段又はＣＤ－ＲＷ、ＣＤ－Ｒ等の記憶媒体に格納され、そのシステム又は装置のコンピュータ（又はＣＰＵやＭＰＵ）が当該記憶手段や当該記憶媒体に格納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、達成されることは云うまでもない。

## 【図面の簡単な説明】

## 【 0 2 2 2 】

【図 1】 A はコンピュータシステム構成図、 B はコンピュータモジュールの概略図

【図 2】 図 1 A のコンピュータシステムにエンジンサーバーを追加した構成図

【図 3】 アプリケーションのメインウインドウ

【図 4】 コンテナの条件設定されたウインドウ

【図 5】 コンテナルール

【図 6】 リンク作成フロー

【図 7】 リンク作成時 UI 図

【図 8】 レイアウト計算の全体フロー

【図 9】 レイアウト計算の詳細フロー

【図 10】 図 9 のフローに対応する UI

【図 11】 レイアウト計算時におけるコンテナの集合

【図 12】 可変リンクによるコンテナ配置

【図 13】 可変リンク設定フロー

【図 14】 リンクの設定ダイアログウインドウ

【図 15】 固定リンクによるレイアウト結果

【図 16】 可変リンクによるレイアウト結果

【図 17】 複数ページのドキュメントテンプレート

【図 18】 ページを跨ったリンクを持つレイアウトデザイン

【図 19】 リンク設定およびリンク表示フロー

【図 20】 コンテナ属性設定フロー

【図 21】 ページ間リンク時の自動レイアウトの動作フロー

【図 22】 ページ移動レイアウト結果

【図 23】 コンテナ属性設定フロー

【図 24】 第二実施例における自動レイアウトの動作フロー

【図 25】 コンテナ情報

【図 26】 コンテナ属性の設定プロパティダイアログ

【図 27】 ページ移動不可レイアウト結果

10

20

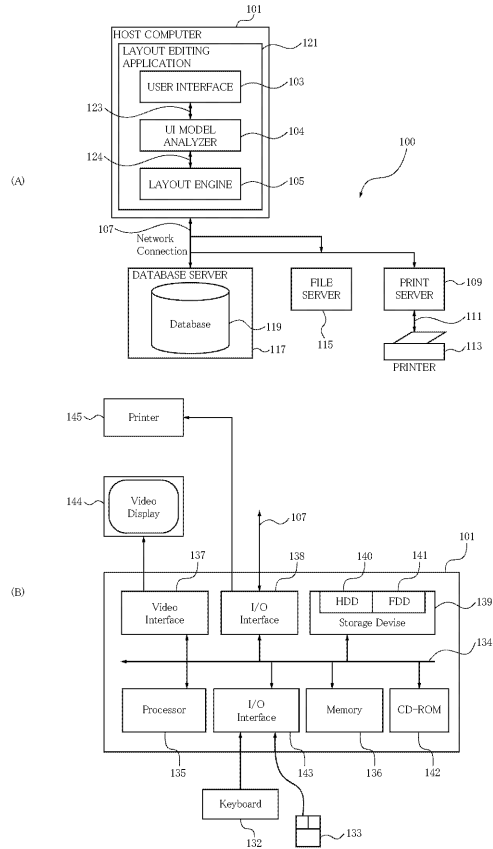
30

40

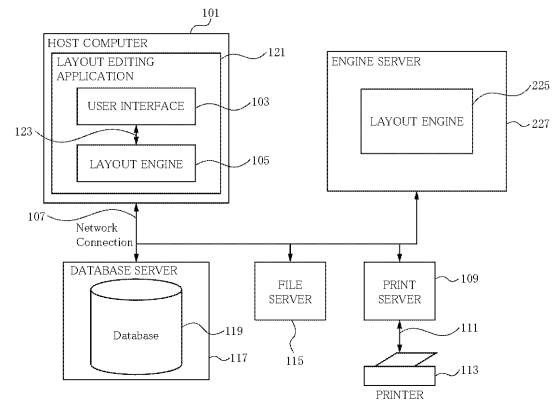
50

【図 2 8】	ページ跨りの自動レイアウトフロー	
【図 2 9】	コンテナのページ跨りレイアウト結果	
【図 3 0】	ページ跨ぎ用編集ページ	
【図 3 1】	リンク属性	
【図 3 2】	実施例 3 におけるコンテナ属性設定プロパティ	
【図 3 3】	実施例 3 におけるドキュメントテンプレート	
【図 3 4】	実施例 3 におけるリンク属性	
【図 3 5】	実施例 3 におけるコンテナ属性設定フロー	
【図 3 6】	実施例 3 におけるリンクに基づくコンテナ属性設定フロー	
【図 3 7】	コンテナ属性設定結果表示プロパティ	10
【図 3 8】	ページ跨ぎリンク設定ダイアログウィンドウ	
【符号の説明】		
【 0 2 2 3 】		
1 0 1	汎用コンピュータモジュール	
1 0 3	ユーザインターフェース	
1 0 5	レイアウトエンジン	
1 1 9	データベース	
1 2 1	レイアウトアプリケーションプログラム	
1 3 2	キーボード	
1 3 3	マウス	20
1 4 3	Ｉ／Ｏインタフェース	
1 4 4	ビデオディスプレイ	
3 0 1	アプリケーションウィンドウ	
3 0 3	ツールバー	
3 1 3	カーソル / ポインタ	
4 0 6	リンクツールボタン	
4 0 7、4 0 8	コンテナ	
4 0 9	アンカーアイコン	
4 1 0	固定されていない辺	
4 1 2	リンク	30

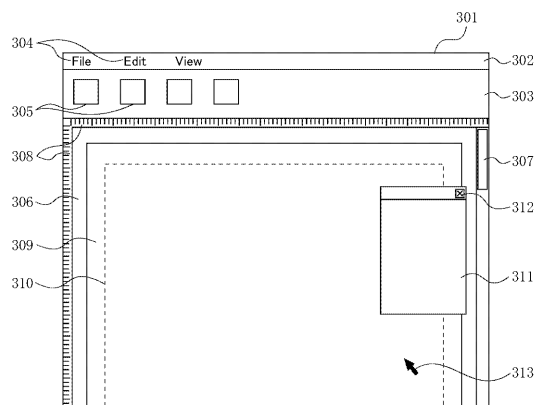
【図 1】



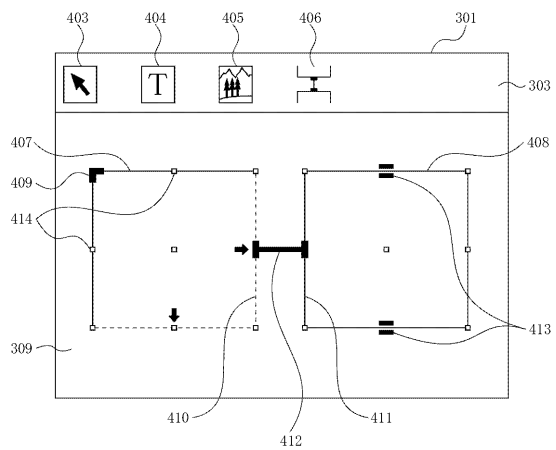
【図 2】



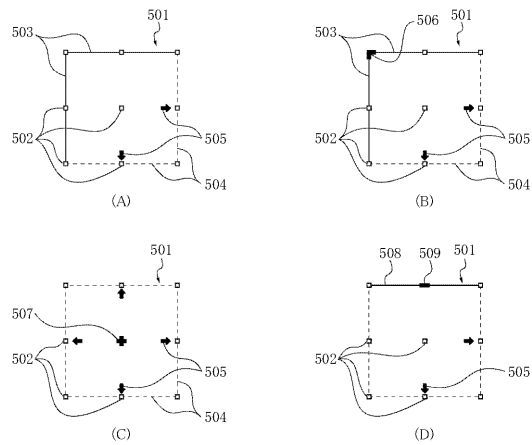
【図 3】



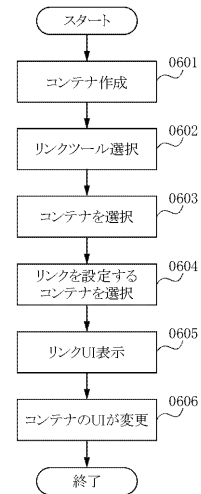
【図 4】



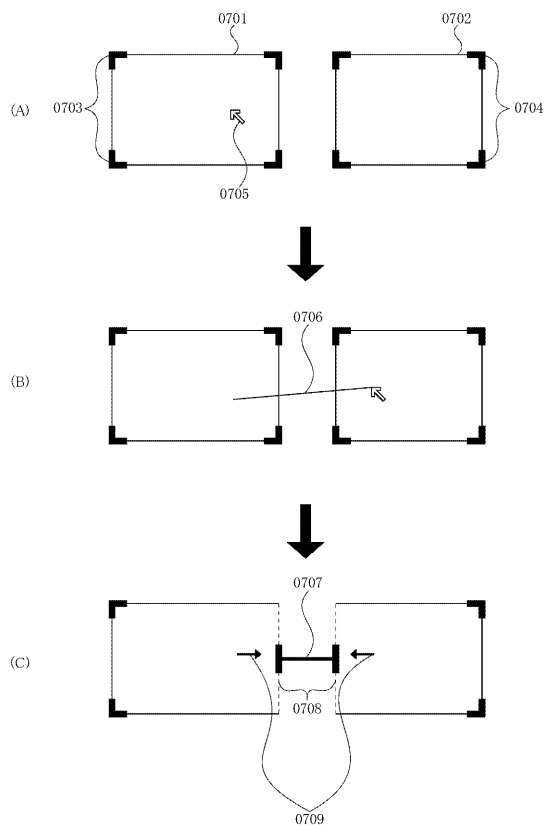
【図 5】



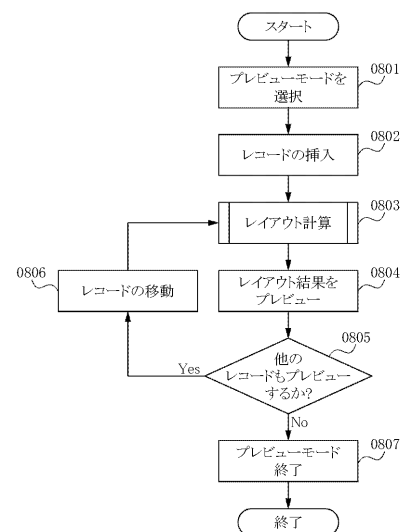
【図 6】



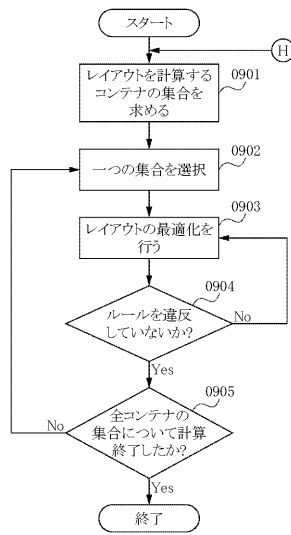
【図 7】



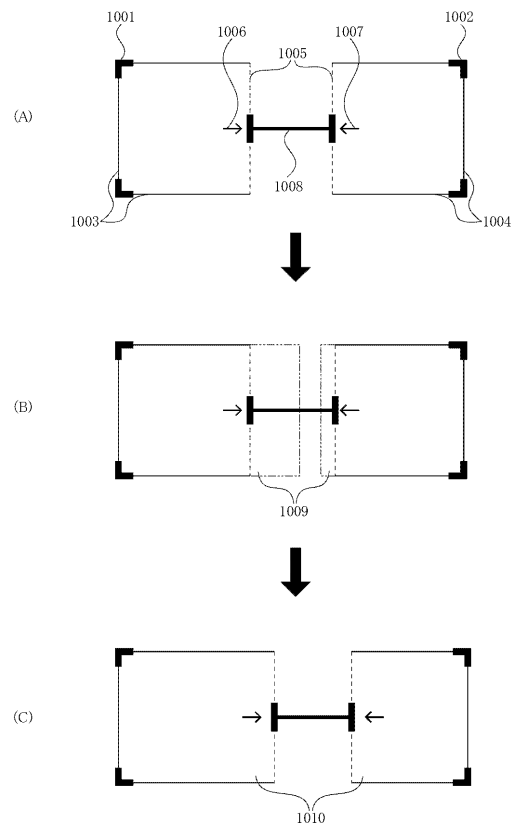
【図 8】



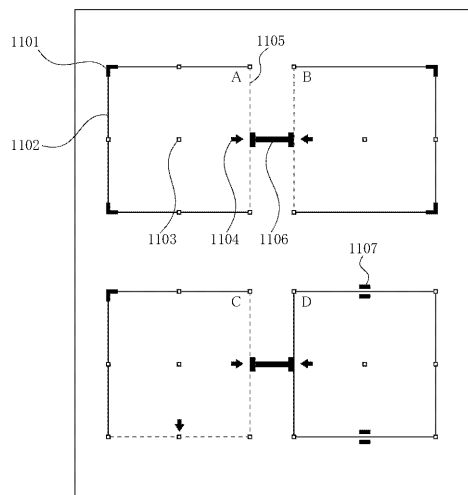
【図 9】



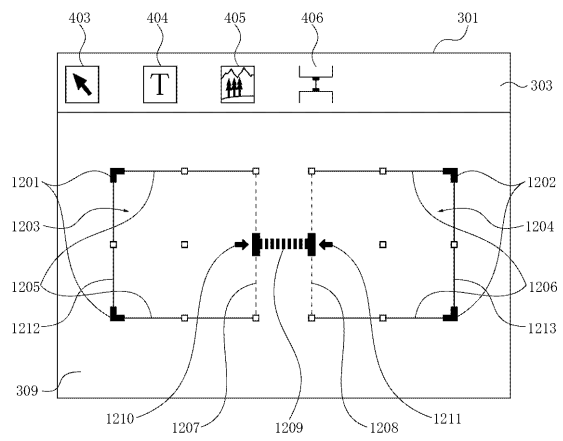
【図 10】



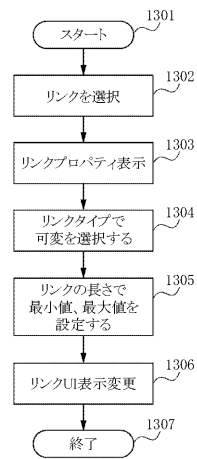
【図 11】



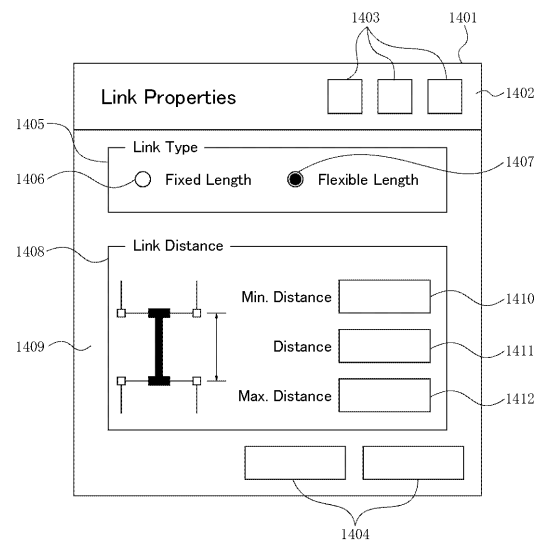
【図 12】



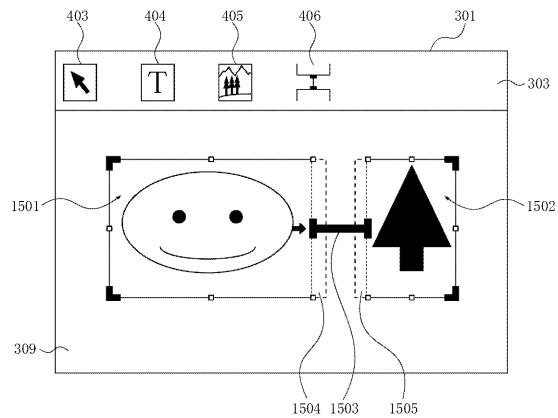
【図 13】



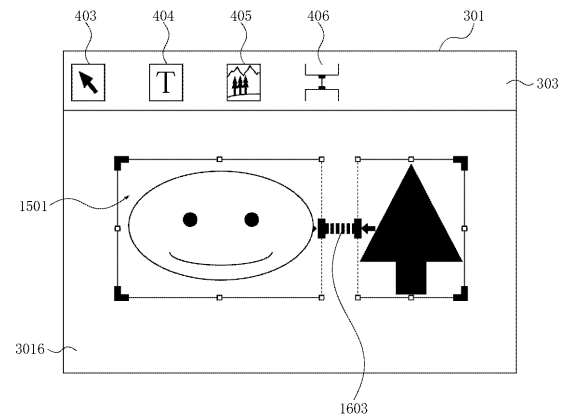
【図 14】



【図 15】

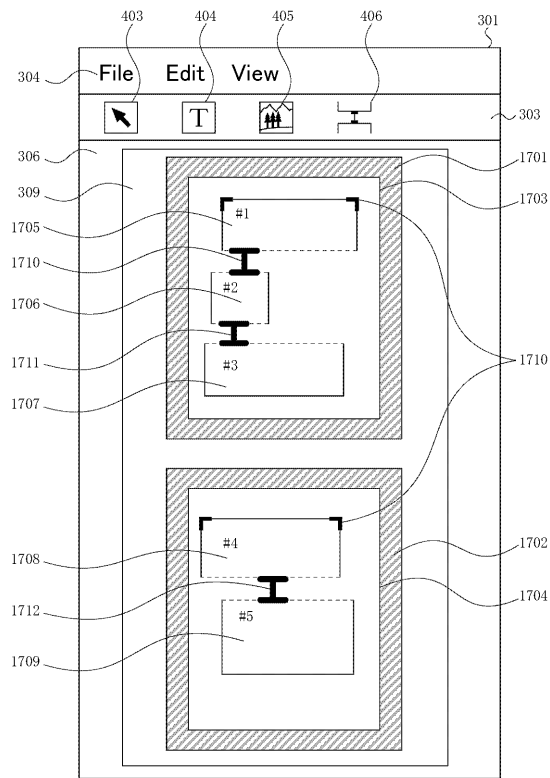


【図 16】

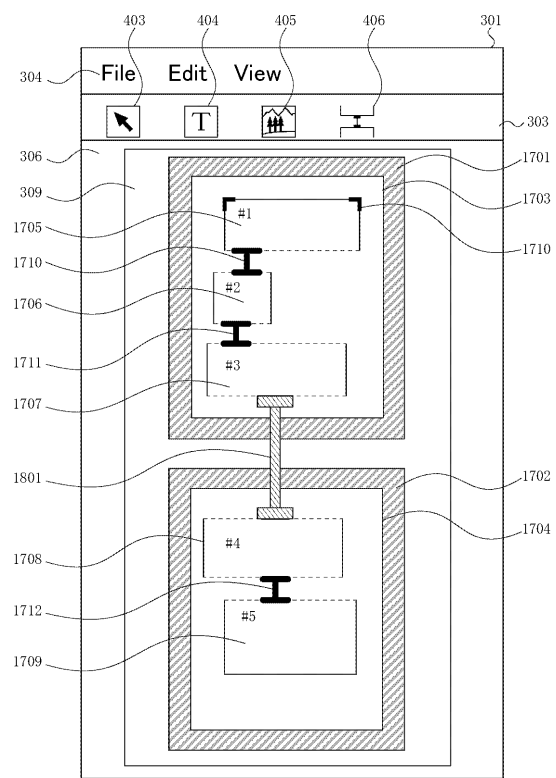




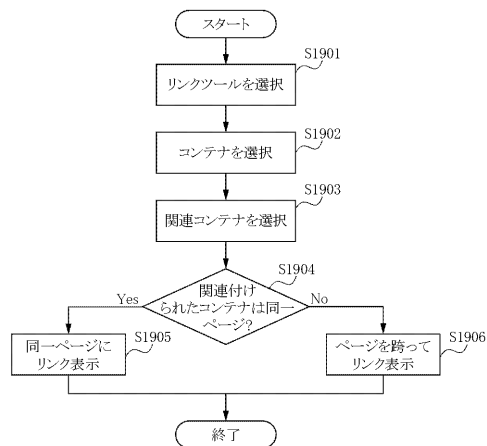
【図 17】



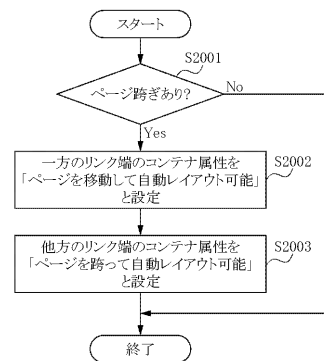
【図 18】



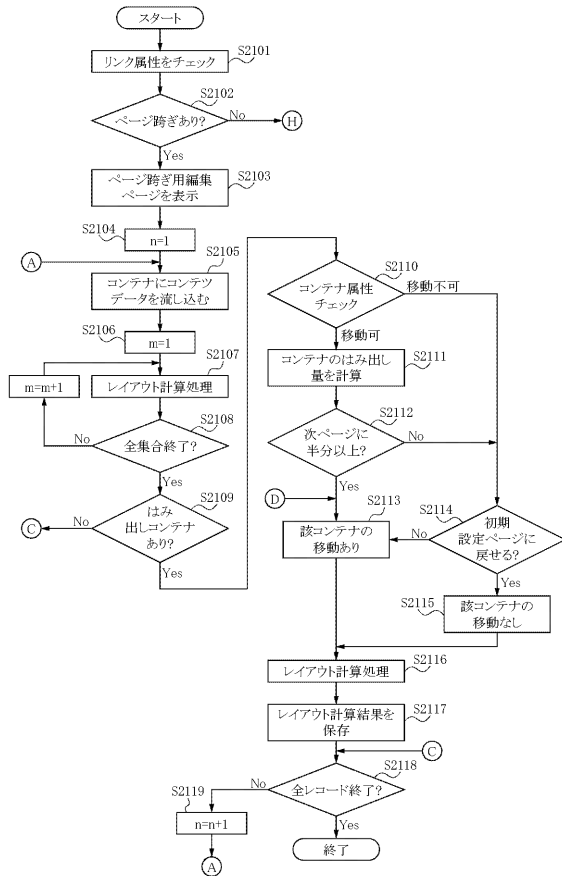
【図 19】



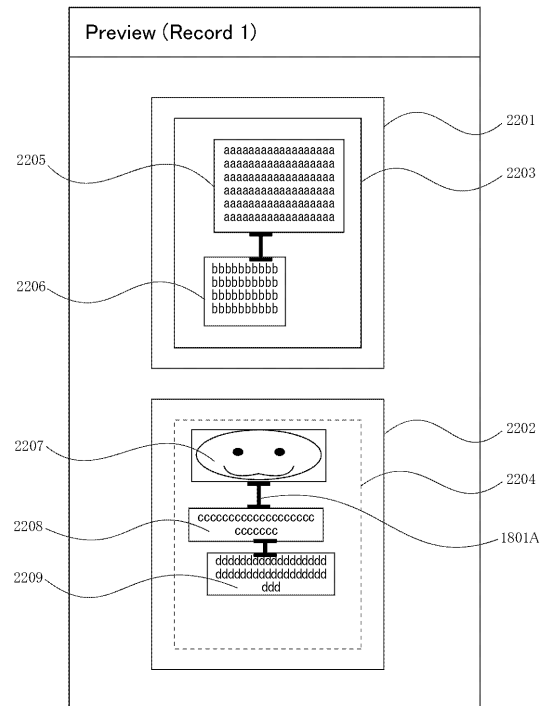
【図 20】



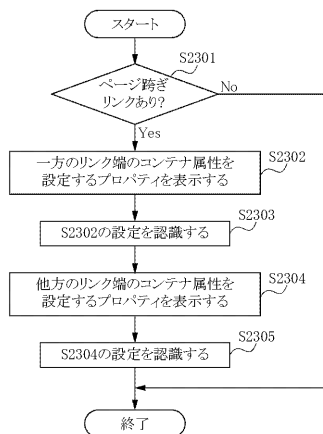
【図 2 1】



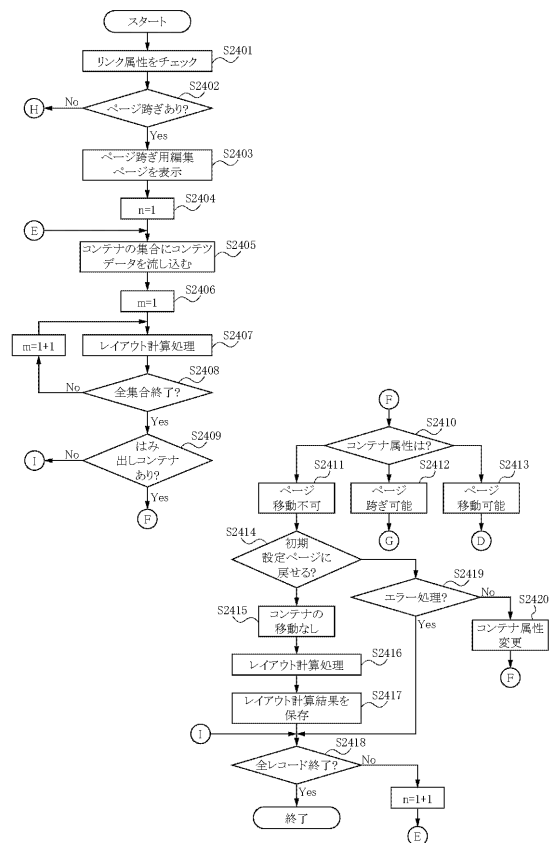
【図 2 2】



【図 2 3】



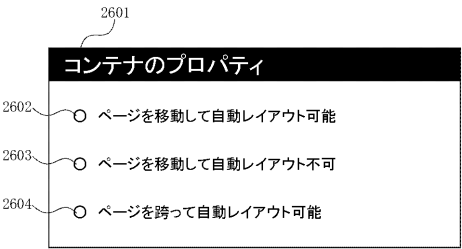
【図 2 4】



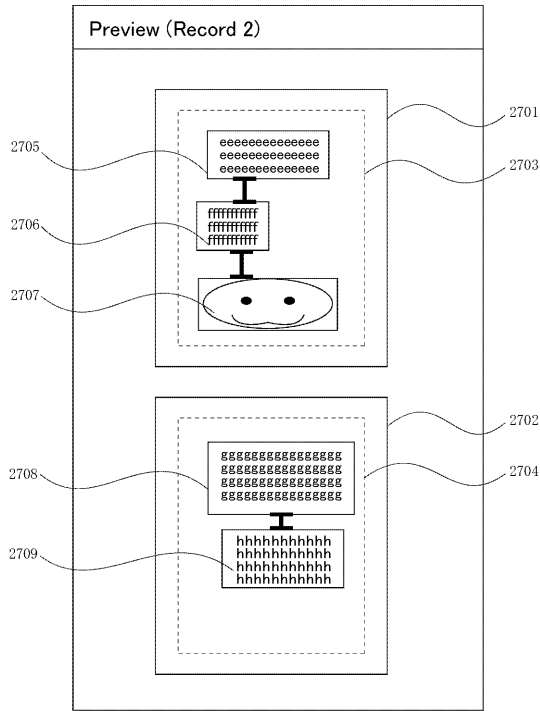
【図 25】

	レイアウト 方法	コンテナ 種	可変/ 固定	可変範囲	コンテナ サイズ	初期配置 ページ	フォント サイズ	フォント 変更範囲	集 合
コンテナ1705	移動不可	テキスト	可変	縦3〜8 横8〜10	縦3 横8	1701	12	12〜15	1
コンテナ1706	移動不可	テキスト	可変	縦3〜8 横5〜8	縦5 横5	1701	10	8〜10	1
コンテナ1707	移動可能	イメージ	可変	縦5〜10 横5〜12	縦4 横10	1701	—	—	1
コンテナ1708	移動可能	テキスト	可変	縦3〜12 横5〜12	縦6 横10	1702	10	8〜12	1
コンテナ1709	移動不可	テキスト	可変	縦4〜10 横4〜12	縦6 横10	1702	8	8〜12	1

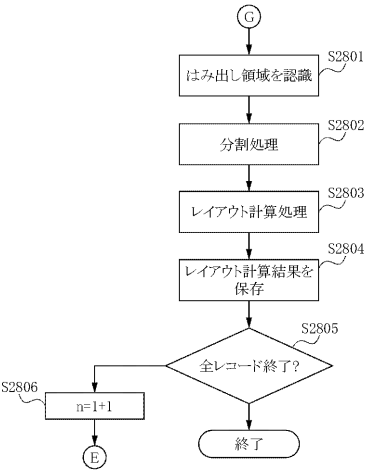
【図 26】



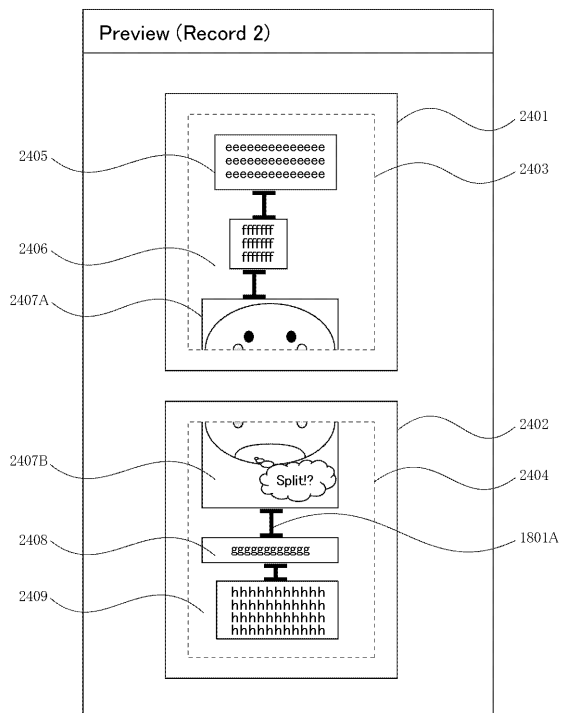
【図 27】



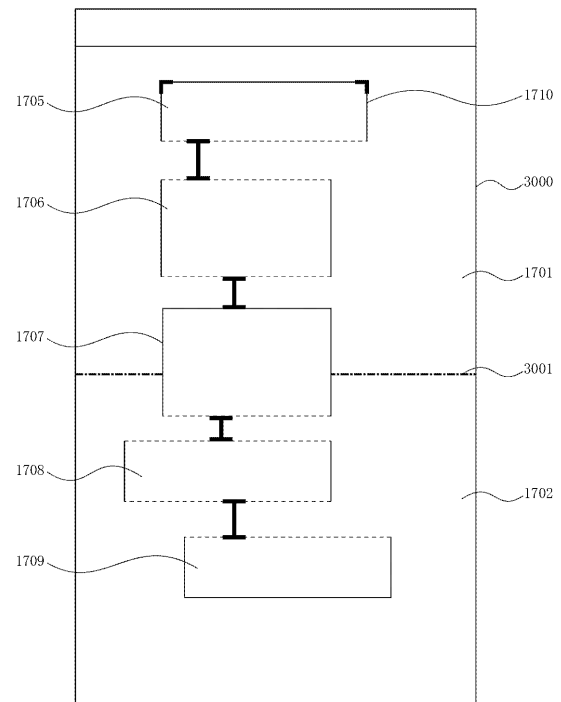
【図 28】



【図 29】



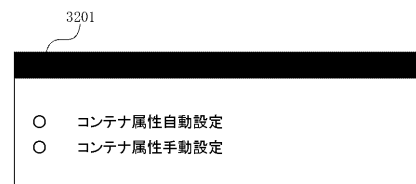
【図 30】



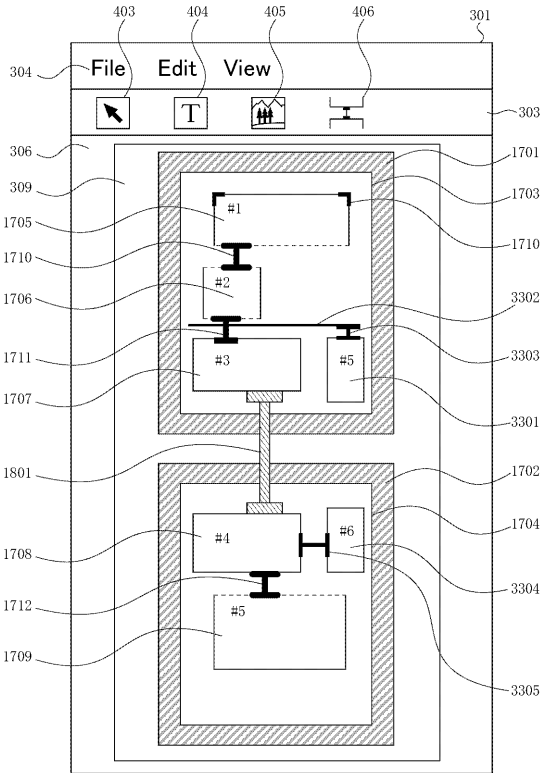
【図 31】

	関連コンテナ	ページ跨ぎON/OFF	可変/固定
リンク1710	1705-1706	OFF	固定
リンク1711	1706-1707	OFF	固定
リンク1801	1707-1708	ON	固定
リンク1712	1708-1709	OFF	固定

【図 32】



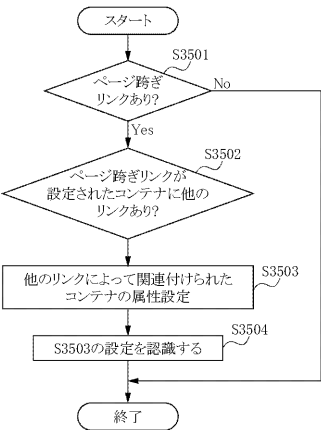
【図 3 3】



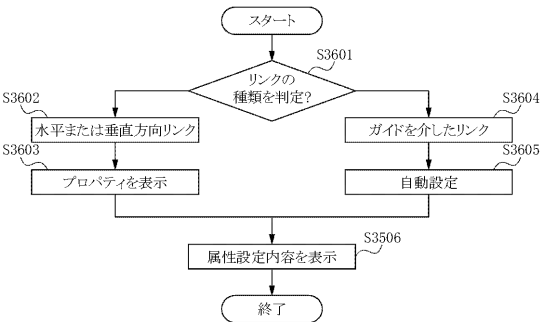
【図 3 4】

	関連コンテナ	ページ跨ぎON/OFF	可変/固定	ガイド
リンク1710	1705-1706	OFF	固定	なし
リンク1711	1706-1707	OFF	固定	なし
リンク1801	1707-1708	ON	固定	なし
リンク1712	1708-1709	OFF	固定	なし
リンク3303	1707-3301	OFF	固定	あり
リンク3305	1708-3304	OFF	固定	なし

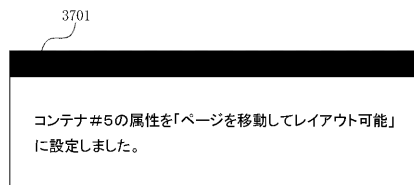
【図 3 5】



【図 3 6】



【図 37】



【図 38】

ページ跨ぎリンク設定プロパティ	
<input type="radio"/> 可変リンク <input checked="" type="radio"/> 固定リンク	
固定リンクサイズ <input type="text"/>	
可変リンクサイズ	
Min.Distance	<input type="text"/>
Distance	<input type="text"/>
Max.Distance	<input type="text"/>

---

フロントページの続き

(56)参考文献 米国特許第05845303(US,A)  
特開2003-271325(JP,A)  
特開2004-220561(JP,A)  
特開2002-297572(JP,A)

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