

(19) 日本国特許庁(JP)

(12) 公 開 特 許 公 報(A)

(11) 特許出願公開番号
特開2006-277727
(P2006-277727A)

(43) 公開日 平成18年10月12日(2006. 10. 12)

(51) Int.Cl.	F I	テーマコード (参考)
G O 6 T 11/60 (2006. 01)	G O 6 T 11/60 1 O O A	5 B O O 9
G O 6 F 17/21 (2006. 01)	G O 6 F 17/21 5 3 6	5 B O 5 O
H O 4 N 1/387 (2006. 01)	H O 4 N 1/387	5 C O 7 6
G O 6 F 3/048 (2006. 01)	G O 6 F 3/048 6 5 1 B	5 E 5 O 1

審査請求 未請求 請求項の数 26 O L (全 55 頁)

(21) 出願番号 特願2006-48014 (P2006-48014)	(71) 出願人 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
(22) 出願日 平成18年2月24日 (2006. 2. 24)	(74) 代理人 100076428 弁理士 大塚 康德
(31) 優先権主張番号 特願2005-61063 (P2005-61063)	(74) 代理人 100112508 弁理士 高柳 司郎
(32) 優先日 平成17年3月4日 (2005. 3. 4)	(74) 代理人 100115071 弁理士 大塚 康弘
(33) 優先権主張国 日本国 (JP)	(74) 代理人 100116894 弁理士 木村 秀二
(特許庁注：以下のものは登録商標) 1. L i n u x	(72) 発明者 牧野 順 東京都港区港南3-9-33-810 Fターム(参考) 5B009 NC01 NC02 RC06
	最終頁に続く

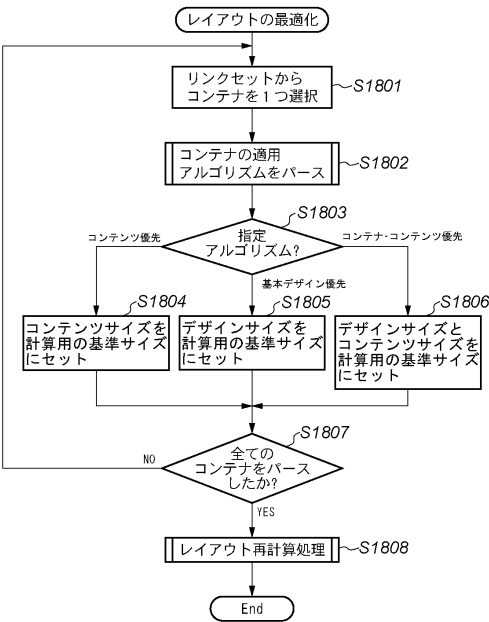
(54) 【発明の名称】 レイアウト処理方法、情報処理装置及びコンピュータプログラム

(57) 【要約】

【課題】 各データ領域毎にレイアウト計算におけるレイアウト調整の仕方を設定可能とし、ユーザの意図を柔軟に反映できるレイアウト処理を実現する。

【解決手段】 データ領域を有するテンプレートのレイアウトを、該データ領域に描画されるコンテンツデータに応じて動的に変更するレイアウト処理方法は、データ領域に適用すべき複数のレイアウト調整アルゴリズムを保持する保持手段から、データ領域に適用すべきレイアウト調整アルゴリズムを取得選択し、データ領域へコンテンツデータを流し込むに際して、取得選択したレイアウト調整アルゴリズムに従ってデータ領域のレイアウトを調整する。そして、調整されたデータ領域にコンテンツデータをレイアウトして出力する。

【選択図】 図 1 8



【特許請求の範囲】

【請求項 1】

データ領域を有するテンプレートのレイアウトを、該データ領域に描画されるコンテンツデータに応じて動的に変更するレイアウト処理方法であって、

複数のレイアウト調整アルゴリズムから、前記データ領域に適用すべきレイアウト調整アルゴリズムを選択する選択工程と、

データ領域へコンテンツデータを流し込むに際して、前記選択工程で選択したレイアウト調整アルゴリズムに従って前記データ領域のレイアウトを調整するレイアウト調整工程と、

前記レイアウト調整工程において調整されたデータ領域に前記コンテンツデータをレイアウトして出力する出力工程とを備えることを特徴とするレイアウト処理方法。 10

【請求項 2】

前記複数のレイアウト調整アルゴリズムの各々は、データ領域と目標形状との差異が小さくなるようにデータ領域のレイアウトを調整するものであり、

前記差異の算出過程がレイアウト調整アルゴリズム毎に異な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】

前記レイアウト調整工程は、リンクで関連付けられた複数のデータ領域の各々のレイアウトを調整する場合に、

前記関連付けられた複数のデータ領域の各々について、前記選択工程で選択されたレイアウト調整アルゴリズムに従って前記差異を算出し、それら算出された差異の合計が最小となるように、該関連付けられた複数のデータ領域の各々のレイアウトを調整す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。 20

【請求項 4】

前記複数のレイアウト調整アルゴリズムは、

データ領域に描画されるコンテンツデータに従ったコンテンツ形状を前記目標形状として前記差異を算出する第 1 アルゴリズムと、

前記レイアウトの設計時におけるデータ領域の形状であるレイアウト形状とを前記目標形状として前記差異を算出する第 2 アルゴリズムと、

データ領域と前記コンテンツ形状との差異である第 1 差異と、データ領域と前記レイアウト形状との差異である第 2 差異との合計を、前記データ領域と目標形状との差異とする第 3 アルゴリズムのうちの少なくとも 2 つを含む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。 30

【請求項 5】

前記レイアウト調整工程は、前記第 3 アルゴリズムによりデータ領域のレイアウトを調整する場合に、

前記データ領域と前記コンテンツ形状との差異である第 1 の差異と、該データ領域と前記レイアウト形状との差異である第 2 の差異との合計を目標形状との差異とすることを特徴とする請求項 4 に記載のレイアウト処理方法。

【請求項 6】

前記レイアウト調整工程は、前記第 1 の差異と前記第 2 の差異の各々に設定される重み値に応じて、データ領域のレイアウトを調整することを特徴とする請求項 5 に記載のレイアウト処理方法。 40

【請求項 7】

データ領域にレイアウト調整アルゴリズムを設定するためのユーザインターフェースを提示する提示工程を更に有し、

前記提示工程により提示された前記ユーザインターフェースを介した入力に応じて、前記データ領域に対して設定されたレイアウト調整アルゴリズムを前記保持手段に保持す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 8】

前記ユーザインターフェースは、予め定められた範囲を単位としてレイアウト調整アルゴリズムを指定可能であり、

前記レイアウト調整アルゴリズムは前記予め定められた範囲の各々を単位として設定され、前記予め定められた範囲には、ドキュメント、該ドキュメントを構成するページ、該ページに含まれるエリア、データ領域が含まれることを特徴とする請求項 7 に記載のレイアウト処理方法。

【請求項 9】

前記選択工程は、前記予め定められた範囲毎の優先順に従って、前記データ領域に適用すべきレイアウト編集アルゴリズムを選択することを特徴とする請求項 8 に記載のレイアウト処理方法。

10

【請求項 10】

前記ユーザインターフェースは、ページ上のデータ領域のレイアウト状態を表示し、表示されたデータ領域から指定されたデータ領域に対してレイアウト調整アルゴリズムをユーザに設定させる機能を有することを特徴とする請求項 7 に記載のレイアウト処理方法。

【請求項 11】

前記ユーザインターフェースは、データ領域に設定されたレイアウト編集アルゴリズムをユーザに視覚的に示すことを特徴とする請求項 7 に記載のレイアウト処理方法。

【請求項 12】

データ領域毎に任意のレイアウト調整アルゴリズムを設定可能とする設定工程を更に有す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。

20

【請求項 13】

データ領域を有するテンプレートのレイアウトを、該データ領域に描画されるコンテンツデータに応じて動的に変更する情報処理装置であって、

複数のレイアウト調整アルゴリズムから、前記データ領域に適用すべきレイアウト調整アルゴリズムを選択する選択手段と、

データ領域へコンテンツデータを流し込むに際して、前記選択手段で選択したレイアウト調整アルゴリズムに従って前記データ領域のレイアウトを調整するレイアウト調整手段と、

前記レイアウト調整手段において調整されたデータ領域に前記コンテンツデータをレイアウトして出力する出力手段とを備えることを特徴とする情報処理装置。

30

【請求項 14】

前記複数のレイアウト調整アルゴリズムの各々は、データ領域と目標形状との差異が小さくなるようにデータ領域のレイアウトを調整するものであり、

前記差異の算出過程がレイアウト調整アルゴリズム毎に異なることを特徴とする請求項 13 に記載の情報処理装置。

【請求項 15】

前記レイアウト調整手段は、リンクで関連付けられた複数のデータ領域の各々のレイアウトを調整する場合に、

前記関連付けられた複数のデータ領域の各々について、前記選択手段で選択されたレイアウト調整アルゴリズムに従って前記差異を算出し、それら算出された差異の合計が最小となるように、該関連付けられた複数のデータ領域の各々のレイアウトを調整することを特徴とする請求項 14 に記載の情報処理装置。

40

【請求項 16】

前記複数のレイアウト調整アルゴリズムは、

データ領域に描画されるコンテンツデータに従ったコンテンツ形状を前記目標形状として前記差異を算出する第 1 アルゴリズムと、

前記レイアウトの設計時におけるデータ領域の形状であるレイアウト形状とを前記目標形状として前記差異を算出する第 2 アルゴリズムと、

データ領域と前記コンテンツ形状との差異である第 1 差異と、データ領域と前記レイアウト形状との差異である第 2 差異との合計を、前記データ領域と目標形状との差異とする

50

第 3 アルゴリズムのうちの少なくとも 2 つを含むことを特徴とする請求項 1 4 に記載の情報処理装置。

【請求項 1 7】

前記レイアウト調整手段は、前記第 3 アルゴリズムによりデータ領域のレイアウトを調整する場合に、

前記データ領域と前記コンテンツ形状との差異である第 1 の差異と、該データ領域と前記レイアウト形状との差異である第 2 の差異との合計を目標形状との差異とすることを特徴とする請求項 1 6 に記載の情報処理装置。

【請求項 1 8】

前記レイアウト調整手段は、前記第 1 の差異と前記第 2 の差異の各々に設定される重み値に応じて、データ領域のレイアウトを調整することを特徴とする請求項 1 7 に記載の情報処理装置。 10

【請求項 1 9】

データ領域にレイアウト調整アルゴリズムを設定するためのユーザインターフェースを提示する提示手段を更に有し、

前記提示手段により提示された前記ユーザインターフェースを介した入力に応じて、前記データ領域に対して設定されたレイアウト調整アルゴリズムを前記保持手段に保持することを特徴とする請求項 1 3 に記載の情報処理装置。

【請求項 2 0】

前記ユーザインターフェースは、予め定められた範囲を単位としてレイアウト調整アルゴリズムを指定可能であり、 20

前記レイアウト調整アルゴリズムは予め定められた範囲の各々を単位として設定され、前記予め定められた範囲には、ドキュメント、該ドキュメントを構成するページ、該ページに含まれるエリア、データ領域が含まれることを特徴とする請求項 1 9 に記載の情報処理装置。

【請求項 2 1】

前記選択手段は、前記予め定められた範囲毎の優先順に従って、前記データ領域に適用すべきレイアウト編集アルゴリズムを選択することを特徴とする請求項 2 0 に記載の情報処理装置。

【請求項 2 2】

前記ユーザインターフェースは、ページ上のデータ領域のレイアウト状態を表示し、表示されたデータ領域から指定されたデータ領域に対してレイアウト調整アルゴリズムをユーザに設定させる機能を有することを特徴とする請求項 1 9 に記載の情報処理装置。 30

【請求項 2 3】

前記ユーザインターフェースは、データ領域に設定されたレイアウト編集アルゴリズムをユーザに視覚的に示すことを特徴とする請求項 1 9 に記載の情報処理装置。

【請求項 2 4】

データ領域毎に任意のレイアウト調整アルゴリズムを設定可能とする設定手段を更に有することを特徴とする請求項 1 3 に記載の情報処理装置。

【請求項 2 5】

請求項 1 乃至 1 2 のいずれかに記載のレイアウト処理方法をコンピュータによって実行するための制御プログラム。 40

【請求項 2 6】

請求項 1 乃至 1 2 のいずれかに記載のレイアウト処理方法をコンピュータによって実行するための制御プログラムを格納したコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、テキストやイメージで構成されたドキュメントの生成、編集、そして印刷のための技術に関するものであり、更に詳しくは、バリエブルデータドキュメントの生成、 50

編集、そして印刷に好適なものである。

【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、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。

【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、顧客のニーズに合った提案を行うものである。One-to-Oneマーケティングの代表的な具体的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】

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、ドキュメント上にコンテナをレイアウトする。コンテナとはコンテンツ (描画内容) を描画するための部分領域 (データ領域) であり、フィールド領域と呼ばれることもある。すなわち、ドキュメント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 (データベースの各コンテンツと各コンテナとを関連付ける) といった作業により、カスタマイズ文書 (ドキュメント) を作成する。本明細書ではこのようなドキュメントをバリアブルデータドキュメントという。

【0005】

このようなバリアブルプリントシステムにおいて、レイアウトされた各コンテナには顧客によって異なるコンテンツを流し込むことが可能である。従って、コンテンツに流し込まれるデータのサイズは可変であり、これに対してコンテナのサイズが固定であると次のような課題が生じる。例えば、コンテナのサイズよりも大きなサイズのテキストデータが流し込まれた場合には全てのテキストをそのコンテナ内に表示出来なくなってしまう。あるいは、コンテナのサイズよりも大きなサイズのイメージデータが流し込まれた場合には、そのイメージの一部が欠落してしまう。このような問題をオーバーフローと呼ぶ。

【0006】

なお、コンテンツがイメージデータの場合には、当該イメージを縮小してコンテナ内に描画することも考えられるが、イメージが極端に小さくなってしまうといった弊害が生じる可能性がある。また、固定サイズのコンテナ内に入りきらないテキストデータが挿入された場合に、テキストのフォントサイズを縮小し、当該コンテナ内に全てのテキストを表示可能にする技術も提案されている。しかしながら、このようにフォントサイズを調節する場合、フォントサイズが小さくなりすぎて、ドキュメント全体のバランスが崩れたり、読みにくくなるという課題が生じる。

【0007】

このような課題を解決するための自動レイアウトの技術として、ある文字領域のサイズが大きくなった場合に、隣接した領域との間隔を保つべく当該隣接する領域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。

【特許文献1】特開平7-129658号公報 (0049、図8)

【発明の開示】

【発明が解決しようとする課題】

【0008】

特許文献1に記載の技術には、テキストの入力に応じて文字領域が拡大されていくこと

が記載されている。しかしながら、文字領域のサイズが大きくなった場合、隣接した領域は間隔を保つべく縮小されてしまうため、入力されるテキスト量が増加するに従い、隣接した領域は縮小され続けてしまうといった問題点があった。

【0009】

更に、特許文献1では、上述したようなデータベースと各レイアウト枠を関連付けてコンテンツデータを流し込むことで、顧客毎にカスタマイズした文書を作成するバリアブルプリントシステムについては考慮されていなかった。

【0010】

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るという問題が生じてしまう。また、そのような場合に、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。また、流し込まれるコンテンツデータのサイズに応じてコンテナのサイズを拡大または縮小させることが考えられる。しかしながら、2つのコンテナが関連付けられた状態で、一方のコンテナサイズが拡大された場合、関連付けられているコンテナは一方のコンテナサイズの拡大に伴って縮小するしかなくなる。そのため、コンテナ間のサイズバランスを考慮したレイアウト処理を行うことができない。

10

【0011】

そこで、関連付けられた複数の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要求される。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を変更する場合、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関連付けられたコンテナのサイズバランスを考慮したレイアウト処理が可能なる。

20

【0012】

上述したOne-to-Oneマーケティングでは顧客のニーズに合わせてカタログやパンフレットを作成することが目的の一つとして挙げられる。このような場合、顧客ごとに異なるパンフレットやカタログを作成することが想定される。

【0013】

しかしながら、多種多様なパンフレットやカタログを作成する場合、上述したコンテンツデータのサイズに応じてコンテナサイズの変化量を調整するレイアウト調整アルゴリズムのみでは、ユーザに最適なレイアウトを生成することができなくなる可能性が高くなる。

30

【0014】

また、上述したようなレイアウト調整アルゴリズムでは、データベースのデータのサイズに応じて自動的にレイアウト計算を行う際に、どのようにコンテナのサイズを変化させるかをユーザが設定することはできなかった。このため、作成するドキュメント内で、あるコンテナはデザインした時のサイズをなるべく維持し、あるコンテナはコンテンツの描画サイズを優先して変形させたいというように、ユーザの意図を柔軟に反映することができない。

【0015】

本発明は上記の課題に鑑みてなされたものであり、上述するような多種多様なコンテンツデータを扱うバリアブルプリントにおいてもユーザが所望とする出力結果を得ることを可能とすることを目的の一つとする。例えば、複数のレイアウト調整アルゴリズムから適用すべきレイアウト調整アルゴリズムを用いてデータ領域のコンテンツデータを調整するレイアウト処理を実現する。

40

【0016】

また、領域毎に任意のレイアウト調整アルゴリズムを設定可能とすることにより、ユーザの意図を柔軟に反映できるレイアウト処理を実現することを目的の一つとする。

【課題を解決するための手段】

【0017】

50

上記の課題を解決するための本発明の一態様によるレイアウト処理方法は、
データ領域を有するテンプレートのレイアウトを、該データ領域に描画されるコンテンツデータに応じて動的に変更するレイアウト処理方法であって、
複数のレイアウト調整アルゴリズムから、前記データ領域に適用すべきレイアウト調整アルゴリズムを選択する選択工程と、
データ領域へコンテンツデータを流し込むに際して、前記選択工程で選択したレイアウト調整アルゴリズムに従って前記データ領域のレイアウトを調整するレイアウト調整工程と、
前記レイアウト調整工程において調整されたデータ領域に前記コンテンツデータをレイアウトして出力する出力工程とを備える。

10

【 0 0 1 8 】

また、本発明によれば、上記レイアウト処理方法を実行する情報処理装置が提供される。

【 発明の効果 】**【 0 0 1 9 】**

本発明によれば、複数のレイアウト調整アルゴリズムから適用すべきレイアウト調整アルゴリズムを用いてデータ領域のコンテンツデータが調整される。このため、多種多様なコンテンツデータを扱うバリエابلプリントにおいてもユーザが所望とする出力結果を得ることが可能となる。

【 発明を実施するための最良の形態 】

20

【 0 0 2 0 】

以下、添付の図面を参照して本発明の好適な実施形態を説明する。

【 0 0 2 1 】**< システム構成 >**

まず、図 1 A および図 1 B を参照して、本実施形態のバリエابلプリントシステムの構成を説明する。図 1 A はバリエابلデータドキュメントを印刷するためのバリエابلプリントシステム 1 0 0 の構成例を示すブロック図である。また、図 1 B は図 1 A に示されたホストコンピュータ 1 0 1 の構成を更に詳細に示すブロック図である。本実施形態で説明されるバリエابلプリント処理は、レイアウト調整装置であ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で構成される)によって実行される。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プログラム 1 2 1 (本発明のレイアウト調整プログラム)はホストコンピュータ 1 0 1 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る。特にレイアウト編集のための処理やバリエابلデータドキュメントの印刷のための処理はホストコンピュータ 1 0 1 で実行されるソフトウェアにより実現される。

30

【 0 0 2 2 】

レイアウト編集アプリケーションプログラム 1 2 1 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 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、ドキュメントのレイアウト編集やバリエابلプリントに好適な装置が提供されることになる。

40

【 0 0 2 3 】

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、入出力インターフェース 1 4 3 を介してキーボード 1 3 2 や、マウス 1 3 3 のようなポインティングデバイス等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 がビデオインターフェース 1 3 7 を介して接続される。更に、ローカルプリンタ 1 4 5 等を入出力インターフェース 1 3 8 を介して接続することも可能である。また、入出力インターフェース 1 3 8 はコンピュータモジュール 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、ネットワークを介してシステム 1 0 0 内の他のコンピュータ装置にホストコン

50

ピュータ１０１を接続することができる。ネットワーク１０７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。

【００２４】

また、図１Ｂに示すように、ホストコンピュータ１０１は少なくとも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を含んでいる。格納デバイス１３９は、プログラム等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１４０やフロッピー（登録商標）ディスクドライブ１４１を含む。なお、図１Ｂには示されていないが、磁気テープドライブ等も格納デバイス１３９として使用可能である。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（もちろん、ＣＤ－ＲＯＭによってコンピュータプログラムを提供してもよい）。

10

【００２５】

ホストコンピュータ１０１は、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１３５～１４３を利用する。この利用は、ＧＮＵ／ＬＩＮＵＸやマイクロソフトウインドウズ（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なされる。或は、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、上記利用はなされる。すなわち、上述した１３５～１４３で示される各構成は、バス１３４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１０１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。

20

【００２６】

なお、図１Ｂに示したホストコンピュータ１０１の例としては、ＩＢＭ互換ＰＣやＳＵＮのSparcstation、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。

【００２７】

< レイアウト編集アプリケーションの概要 >

本実施形態では、レイアウト編集アプリケーションプログラム１２１はハードディスクドライブ１４０に常駐し、プロセッサ１３５により実行や読み込みがコントロールされるものとする。レイアウト編集アプリケーション１２１のプログラムの媒介記憶装置とネットワーク１０７からフェッチされるデータはハードディスクドライブ１４０に呼応して半導体メモリ１３６を使用する。

30

【００２８】

一つの例では、レイアウト編集アプリケーション１２１のエンコードされたプログラムは、ＣＤ－ＲＯＭやフロッピー（登録商標）ディスク上に格納され、対応するドライブ１４２や１４１を通じて読み込まれ、ハードディスクドライブ１４０にインストールされる。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プログラム１２１はネットワーク１０７からホストコンピュータ１０１内に読み込まれて、ハードディスクドライブ１４０にインストールされてもよい。さらにソフトウェアは、磁気テープまたはＲＯＭまたは集積回路、光磁気ディスク、または、ホストコンピュータ１０１とその他のデバイス間における赤外線等の無線通信、ＰＣＭＣＩＡカードのようなコンピュータ可読カードからインストールされてもよい。或は、そのソフトウェアは、Ｅメール通信やＷＥＢ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１０１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

【００２９】

図１Ａにおいて、本発明のレイアウト編集アプリケーション１２１はコンピュータにバリアブルプリント（バリアブルデータプリント（ＶＤＰ）ともいう）を行わせるものである。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、すなわちレイアウトエンジン１０５とユーザインターフェース１０３、を含んでいる。レイアウトエンジン１０５は、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

50

位置の制限にしたがって、データベース 119 に格納されているバリアブルデータから 1 レコードずつ読み込み、読み込んだデータとコンテナの制限とから、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ソフトウェアコンポーネントである。また、本実施例では、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データを描画し、バリア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。ただし、本発明はこれに限るものではない。例えば、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、図示省略したプリンタドライバに描画情報を出力するようにしてもよい。この場合、プリンタドライバがバリアブルデータドキュメントのイメージ描画処理を行い、印刷データを生成することになる。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とを関連付けるメカニズムを提供する。ユーザインタフェース 103 とレイアウトエンジン 105 はコミュニケーションチャンネル 123 を介して通信する。

10

20

30

【0030】

図 1C は本実施形態によるバリアブルデータプリントの概略を説明する図である。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モジュール 103（以下、ユーザインタフェース 103 と記載する）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 が配置される。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。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（図 1 ではデータベース 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内に記述され、該ドキュメントテンプレートは、HDD 140 またはメモリ 136 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（バリアブルデータ）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。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の各コンテナ 181 ~ 183 に、関連付け情報で関連付けられたバリアブルデータをデータソース 190 から読み込む。そして、読み込んだバリアブル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。なお、レイアウト調整の方法については後述する。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、ビデオディスプレイ 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 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。

【0031】

ドキュメント生成のためのデータソース（190）は、例えば、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 117 上の一般的なデータベース 119 であってもよい。この場合、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 と通信し、データソースを取得できる。また、レイアウト編集アプリケーション 121 によって生成された、バリアブルデータプリントのためのドキュメントテンプレート（180）は、ホストコンピュータ 101 或い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。図 1C で上述したように、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 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バリアブルデータ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、あるいはブ

40

50

リント 1 1 3 に送信されて印刷される。プリントサーバ 1 0 9 はネットワークと直接には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ためのコンピュータである。プリントサーバ 1 0 9 とプリンタ 1 1 3 は一般的な通信チャンネル 1 1 1 を介して接続される。

【 0 0 3 2 】

< 他のシステム構成例 >

図 2 は、図 1 A と類似のブロック図であるが、エンジンサーバ 2 2 7 が追加されている。エンジンサーバ 2 2 7 に格納されているレイアウトエンジン 2 2 5 は、レイアウトエンジン 1 0 5 の分離バージョンである。エンジンサーバ 2 2 7 には一般的なコンピュータが用いられる。レイアウトエンジン 2 2 5 は、印刷やその他の目的に応じてバリアブルデータドキュメントを生成するために、ファイルサーバ 1 1 5 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 1 1 9 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フェース 1 0 3 を介して要求される。

10

【 0 0 3 3 】

< 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 1 2 1 について説明する。

【 0 0 3 4 】

[メインウィンドウ]

ユーザインタフェース 1 0 3 は、操作時に図 3 に示されるようなアプリケーションウィンドウ 3 0 1 によって形成されたユーザインタフェース画面をビデオディスプレイ 1 4 4 に表示させる。このウィンドウ 3 0 1 は、メニューバー 3 0 2、ツールバー 3 0 3、ワークエリア 3 0 6 とオプションのパレット 3 1 1 を有する。メニューバー 3 0 2 とツールバー 3 0 3 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である。また、ワークエリア 3 0 6 はマウス 1 3 3 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 3 1 1 はオプションであり、カーソル / ポインタデバイス 3 1 3 はマウス 1 3 3 が指し示す位置を表す。

20

【 0 0 3 5 】

メニューバー 3 0 2 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 3 0 4 を持つ。

【 0 0 3 6 】

ツールバー 3 0 3 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 3 0 5 を持つ。

30

【 0 0 3 7 】

ルーラー 3 0 8 はオプションであり、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。

【 0 0 3 8 】

パレット 3 1 1 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 3 1 1 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 3 1 2 を持つ。パレット 3 1 1 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 3 1 1 はアプリケーションウィンドウ 3 0 1 の範囲内のみに表示されることを制限される、あるいはアプリケーションウィンドウ 3 0 1 の外側にその一部或いは全体を表示することを許される。

40

【 0 0 3 9 】

ツールバー 3 0 3 には図 4 に示されるような、ユーザ選択可能な『ボタン』が配置されている。

(1) 選択ツールボタン 4 0 3 : 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、コンテナの周り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、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。

(2) 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

50

を持つコンテナを作成するために使われる。

(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持つコンテナを作成するために使われる。

(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われる。

【0040】

レイアウト編集アプリケーション 121 の図 3 に示し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。また、基本レイアウト内の少なくとも 1 つのコンテナが後述する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、後述する制約の範囲内で変動することになる。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り、可変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。

10

【0041】

[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。また、ドキュメントテンプレートを単にテンプレートということもある。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できる。

20

【0042】

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。

【0043】

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 (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。そして、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。

30

【0044】

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 1C のドキュメントテンプレート 180 を表示しているものである。

40

【0045】

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。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アブルデータによって変化する可能性が有る。これは、ドキュメントテンプレート内に可変表のようにバリアブルデータ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、1 ページ内にフィットできないバリアブル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。

【0046】

50

それぞれのページ内に示される境界線 3 1 0 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。

【 0 0 4 7 】

図 4 は 1 ページのドキュメントテンプレート 3 0 9 上に表示され得るオブジェクトの例を示す図である。このようなオブジェクトとしては、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、固定されている辺 4 1 1、4 1 4、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。アンカーアイコン 4 0 9 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。なお、本願で記述するオブジェクトとはドキュメントテンプレート上に表示され得るもの全てを指し、各オブジェクトを明確に区別する場合は、それぞれの名称で記述することにより明確にする。アンカーアイコン 4 0 9 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 4 0 9 は、コンテナ 4 0 7 の左上の角に設定されている。そのため、コンテナ 4 0 7 はバリアブルデータが流し込まれ、バリアブルデータの画像サイズもしくはテキスト量が多い場合に、右方向及び下方向に拡大可能であることを示している。例えば、アンカーアイコン 4 0 9 がいずれか 1 つの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 4 0 9 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。リンク 4 1 2 についての詳細は後述するが、コンテナ 4 0 7 とコンテナ 4 0 8 が関連付けられていることを示している。そして、このリンク 4 1 2 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 4 0 8 が右方向に移動可能であることが示されている（なお後述する可変サイズリンクの場合は、流し込まれるデータサイズによりリンクの長さが設定された範囲内で可変となる）。スライダー 4 1 3 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。

10

20

【 0 0 4 8 】

[コンテナ]

ここで、コンテナについて説明する。コンテナは、ドキュメントテンプレート内にバリアブルデータファイルから固定あるいは可変のテキスト/イメージが流し込まれ、描画されるスペース（これを部分領域またはデータ領域と呼ぶ）である。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。また、コンテナに流し込まれるデータをコンテンツまたはコンテンツデータ（バリアブルデータまたはフィールドデータ）と呼ぶ。

30

【 0 0 4 9 】

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集動作をもっている。下記は本実施形態によるコンテナの定義である。

【 0 0 5 0 】

（ 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。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印刷には適していないため、ここでは意図していない。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、固定コンテンツが可変コンテンツとリンクにより関連付けられている場合、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。

40

【 0 0 5 1 】

（ 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、各コンテナに設定可能であるが、あるコンテナと同じコンテナ属性

50

であるという設定を行うことも可能である。

【 0 0 5 2 】

(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、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か、その両方が可能である。

【 0 0 5 3 】

(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。

10

【 0 0 5 4 】

[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世代をコントロールする主要な方法である。制約の一つの例は『このコンテナのコンテンツの高さは最大 4 インチです』である。また、別の制約の例は『コンテナのコンテンツの左エッジ (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』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、編集するためのいろいろな方法である。

20

【 0 0 5 5 】

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。よって、以下の説明では、バリアブルデータ印刷に特化した方法における表示・編集に焦点を合わせる。

30

【 0 0 5 6 】

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ーフェースが利用される。

【 0 0 5 7 】

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。したがって、コンテナの左辺を論じるとは、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。同様に、コンテナの高さを論じるとは、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、ユーザインターフェース 1 0 3 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。

40

【 0 0 5 8 】

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。

【 0 0 5 9 】

(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

50

は、全てのドキュメントで同じになる。

【0060】

(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。

【0061】

(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。

【0062】

(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。ただし、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、そのコンテンツは、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。

【0063】

(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。

【0064】

(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。

【0065】

(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。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ることはない。

【0066】

(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。ただし、コンテナの幅・高さはこの制約によって影響されない。

【0067】

(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。

【0068】

(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。

【0069】

『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、バリア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、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。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た。こ

10

20

30

40

50

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。

【0070】

[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持つ。イメージコンテナは、イメージだけを持つ。

【0071】

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 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、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とによって、当該ドキュメントテンプレート 309 上に作成される。 10

【0072】

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。なお、コンテナの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る。また、コンテナを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。 20

【0073】

30

[コンテナの表示方法]

図 5 の (A) ~ (D) はコンテナの辺に関する表示ルールを例示している。

【0074】

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(アイテム 503)あるいは点線(504)を用いて辺を表すとともに、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、修正するために領域の辺上あるいは近傍に描画されたコントロール点、502)、スライダー(辺の両側に描画された短い並行線、図 4 の 413)、拡縮アイコン(505)、色を用いる。 40

【0075】

40

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。

- (1) それぞれの辺を固定するために、実線で描画する。
- (2) 幅が固定の場合は、左と右の辺を実線で描画する。
- (3) 高さが固定の場合は、上と下の辺を実線で描画する。
- (4) 軸は描画しない。
- (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縮アイコンが描画され、それらの辺を点線で描画する。
- (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。
- (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、エッジ 50

の中央にスライダーが描画される。

(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。

【 0 0 7 6 】

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。

【 0 0 7 7 】

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。

【 0 0 7 8 】

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。なお、ユーザインターフェースにおける可変の辺の描画位置を別の手段を用いて決定してもよい。例えば、すべてのドキュメントで平均化されたサイズのコンテナ (平均化されたコンテンツエリア) を用いて可変の辺を描画する位置が決定されてもよい。

【 0 0 7 9 】

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。

(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、固定されている (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 4 0 9 は、辺 4 1 4 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。

(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 4 1 3 はコンテナ 4 0 8 に流し込まれたコンテンツデータが、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。例えば、コンテナ 4 0 8 と関連付けられている (リンク設定されている) コンテナ 4 0 7 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 4 0 7 のサイズが小さくなる。そのため、コンテナ 4 0 8 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 4 0 7 のサイズが大きくなる場合は、逆にコンテナ 4 0 8 は右方向にスライドしてレイアウトされることになる。

【 0 0 8 0 】

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 8 1 】

なお、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。

【 0 0 8 2 】

図 5 の (A) で、コンテナ 5 0 1 は、幅・高さの両方が固定されていない (可変である

10

20

30

40

50

）。固定された辺 5 0 3 は実線で表現され、可変の辺 5 0 4 は点線で表現されている。拡大アイコン 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。

【 0 0 8 3 】

図 5 の (B) において、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 が、交差している両方の辺 5 0 3 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。

【 0 0 8 4 】

図 5 の (C) において、コンテナ 5 0 1 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 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を示している。すなわち、コンテナ 5 0 1 はアンカーアイコン 5 0 7 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 5 0 7 の位置が常にコンテナ 5 0 1 の中心点となるようにレイアウト調整される。

10

【 0 0 8 5 】

図 5 の (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・右辺は、アンカーアイコン 5 0 9 を通って垂直な中心軸（垂直軸）の周りを、拡大・縮小する。

【 0 0 8 6 】

[リンク]

20

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、図 4 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 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。図 6 はリンクの設定方法を示したフローチャートである。また図 7 の (A) ~ (C) はリンク設定時のユーザインターフェース (UI) の遷移例を示している。図 6 , 図 7 (A) ~ (C) 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。なお、本願に記載されているフローチャートの各ステップの処理は、情報処理装置（ホストコンピュータ）が有するプロセッサ 1 3 5 によって実行されることとなる。

30

【 0 0 8 8 】

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画面のワークエリア 3 0 6 上に編集すべく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ーフェースの遷移例を示している。

40

【 0 0 8 9 】

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、前述したリンクツールが選択状態（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）になったかを判断する。リンクツールが選択上体でなければ、必要に応じて他の各種処理を実行し（ステップ S 6 0 9 ）、ステップ S 6 0 2 に戻る。

【 0 0 9 0 】

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。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 つのコンテナのうち的一方（コンテナ 7 0 1 と

50

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 121 のユーザインターフェース 103 は第 1 のコンテナが指定されたことを認識し(ステップ S603)、選択されたコンテナを特定する情報をメモリ 136 に保持する。また、以降のマウ斯卡ソルの移動に応じた軌跡を画面に表示するようにする(ステップ S604)。例えば、図 7 の(B)における線分 706 は、図 7 の(A)の状態におけるクリック位置と現在のマウ斯卡ソル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。

【0091】

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 702)までマウスポインタ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 103 は第 2 のコンテナが指定されたことを認識する(ステップ S605)。そして、レイアウト編集アプリケーション 121 は、ステップ S604 で保持した第 1 のコンテナと、ステップ S605 で指定が認識された第 2 のコンテナとの間にリンクを設定する(ステップ S606)。

10

【0092】

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 UI 707 が表示される(ステップ S607)。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 S608)。すなわち、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、点線で示される。図 7 の(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。

20

【0093】

なお、図 7 の(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(A)~(D)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 の(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、たとえば、右コンテナが図 4 の 413 で示したスライダーを持つ設定に変化してもかまわない。

30

【0094】

<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)]

本実施形態のレイアウト編集アプリケーションは、レイアウトモードとプレビューモードに分けられる。レイアウトモードは、ユーザインターフェース 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モードである。プレビューモードは、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モードである。このプレビューモードにおいて、実際のレコードが有するコンテンツデータが挿入され、レイアウトを計算する。ただし、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。なお、コンテンツデータを流し込むべきコンテナに優先順位が設定されている場合は、該優先順位に従ってレイアウト計算を行うこととなる。優先順位に従うレイアウト計算の一例として、優先順位の高いコンテナに流し込まれるコンテンツデータは、できる限りオリジナルサイズ(元のデータサイズ)で表示することが考えられる。

40

【0095】

図 8 はレイアウト計算のフローを示している。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 1

50

21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よう決定する（ステップS802）。各コンテナへのフィールドデータの挿入が決定されると、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）。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。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する（ステップS804）。レイアウト編集アプリケーション121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判断する（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と判断した場合は、プレビューモードを終了する（ステップS807）。他のレコードについてプレビューを行うのであれば、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）。

10

【0096】

なお、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。したがって、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（印刷すべく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。

20

【0097】

[レイアウト計算方法（詳細）]

次に、上記ステップS803によるレイアウト計算の詳細について説明する。図9は本実施形態によるレイアウトの優先順位を設定しない場合のレイアウト計算の方法を示したフローチャートである。また、図10の（A）～（C）はそのときのUI表示例を示した図である。図9はレイアウト計算の処理方法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方法に相当する。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。

【0098】

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求める（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。例えば図11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。したがって、コンテナA、Bが集合1、コンテナC、Dが集合2となる。すなわち、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はコントローラー、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。

30

40

【0099】

次に、レイアウト編集アプリケーション121は、ステップS901で求め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。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。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断する。この判断は、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断できる。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。コンテナ

50

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。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の文字数と、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が計算できる。尚、文字属性とは、フォントタイプ、フォントサイズ、文字ピッチ、行ピッチなどである。ここで、テキスト用コンテナの場合は、コンテナ A の縦横比の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）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断する。すべて流し込めると判断された場合は、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、すべて流し込めないと判断された場合はサイズ変更を行う。図 11 の例では、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。

10

【0100】

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）。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。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）の合計サイズを求める。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 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）との差を求める。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。

20

【0101】

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、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に計算をする（ステップ S904）。ここで記述したルールとは、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。そして、ステップ S902 ~ 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を計算する（ステップ S905）。

30

【0102】

図 10 の（A）～（C）は優先順位の設定なしでのレイアウト時の UI 例である。

40

【0103】

図 10 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。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。図 10 の（B）は図 10 の（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サイズを表している。そして、レイアウト計算が行われる。図 10 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 10 の

50

(C)で示されるように、図10の(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。

【0104】

図10の(A)~(C)のような横方向のみのレイアウト計算の場合は同等の差異をもってレイアウト計算を行う処理だけで問題は生じない。しかしながら、図31のように縦方向および横方向を考慮したレイアウト計算を行わなければならない場合、同等の差異をもってレイアウト計算を行う処理だけでは対応できなくなる可能性がある。

横方向と縦方向の2次元的なレイアウトを扱う図31の場合、図9のレイアウト計算方法を用いて、まず横方向に調整を行い、後に縦方向に調整を行う。すなわち、コンテナ3101と3102の間で、上述した手順で設定値Dに収まるようレイアウト調整を行い、その後、コンテナ3102と3103の間で設定Gに収まるようレイアウト調整を行うこととなる。

【0105】

しかし、横方向に調整を行い、後に縦方向に調整を行うという調整方法では、ユーザが所望とする基本レイアウトからずれたレイアウト結果になることが想定される。

【0106】

そのため、各コンテナについて、基本パターン(レイアウト設計時のコンテナサイズやコンテナの縦横比など)からなるべくずれないようにする。そのために、例えば、以下の方法が考えられる。基本パターンの幅とレイアウト調整後の幅の差の2乗と、基本パターンの高さでレイアウト調整後の高さの差の2乗との和を求め、これをすべてのコンテナについて算出する。そして、レイアウト編集アプリケーション121は、各コンテナの2乗の和(横方向+縦方向)の合計値が最小となるように、それぞれのコンテナの幅と高さを算出してレイアウト調整を行う。このように最小2乗法を用いてレイアウト調整を行うことにより、ユーザが当初設定した基本レイアウトから大きく外れることがなく、全体としてバランスのとれた最終レイアウトを決定することができる。

【0107】

なお、コンテナ間には後述する可変リンクが設定されているため、流し込まれるコンテンツデータのサイズに応じて後述するようにリンクサイズは変更する。本願記載のレイアウト編集アプリケーションはこのようなレイアウト調整ユニットを備えることにより、割り当てられたデータサイズに応じて複数の部分表示領域(コンテナ)の配置位置を調整することが可能となる。

【0108】

また、上記記載ではレイアウト調整アルゴリズムの一例として最小2乗法を用いることを挙げたが、3乗や4乗など所定の操作によりレイアウト調整アルゴリズムを切り替えることを可能にしても構わない。なお、具体的なレイアウト計算方法は、図32から図34を用いて後述する。

【0109】

[長さ可変のリンクの設定]

図12では可変リンクの設定のためのユーザインターフェースを表している。図4と同様にアプリケーションウィンドウ301とツールバー303がある。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。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の間にはリンク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。

【0110】

また、図14は、リンク設定手段におけるユーザインタフェース画面であり、リンク1

10

20

30

40

50

209の情報をセットするためのダイアログウインドウ1401の例である。このダイアログは、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。このダイアログウインドウ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。リンクタイプが可変の場合にはリンクの長さの最小値(Min.Distance1410)、最大値(Max.Distance1412)、ならびに基準値(Distance1411)が設定できる。図14のダイアログ1401は、たとえば、図6、図7の(A)~(C)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1411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。

10

【0111】

図13は、本実施形態のユーザインターフェース103による可変リンクの設定手順を説明するフローチャートである。例えば図11のコンテナAとコンテナBの間に図6、図7の(A)~(C)で説明した手順でリンクを張ると、まず固定サイズのリンクが張られる。そして、このリンクを選択して図13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11)の状態から可変サイズのリンク1209(図12)へと遷移させることができる。

20

【0112】

まず、マウスにより所望のリンク(たとえばリンク1106)を選択状態とし、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。すると、レイアウト編集アプリケーション121のユーザインターフェース103は、リンクプロパティの表示指示の入力として認識する(ステップS1301)。リンクプロパティの表示指示を認識すると、選択状態のリンク(以下、対象リンクという)に対応したプロパティダイアログウインドウ1401(図14)が表示される。次に、ユーザインターフェース103は、リンクプロパティを表示する(ステップS1302)。なお、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、マウスの右クリックあるいはキーボードの特定のキーの操作等、いかなるものであってもよい。

30

【0113】

ステップS1302で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あり、Link Type1405においては、固定長を示すFixed Length1406が選択されている。

【0114】

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、Link Type1405においてリンクサイズを可変に設定するためのFlexible Length1407を選択する。これによりLinkDistance1408内に配置されているMax. Distance1412、Min. Distance1410、Distance1411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Max.Distance1412に、最小値をMin. Distance1410に、現在の値をDistance1411に設定することになる。

40

【0115】

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。ユーザインターフェース103はこの指示を検出すると、ステップS1303からステップS1304以降へ処理を進め、当該対象リンクに上記設定状態を反映させる。

【0116】

すなわち、まず、ステップS1304において、レイアウト編集アプリケーション12

50

1 は、対象リンクが固定サイズか可変サイズかを判定する。固定サイズが指定されていればステップ S 1 3 0 8 へ進み、対象リンクを「固定サイズ」に設定し、ステップ S 1 3 0 7 で対象リンクの表示状態を「固定リンク」を表す「実線」とする。

【0117】

一方、ステップ S 1 3 0 4 において、可変サイズが指定されていれば、ステップ S 1 3 0 5 へ進み、レイアウト編集アプリケーション 1 2 1 は、対象リンクを「可変サイズ」に設定する。そして、ステップ S 1 3 0 6 にて、上記ダイアログウインドウ 1 4 0 1 によって設定された対象リンクの現在値（基準値）、最大値、最小値を登録する。その後、ステップ S 1 3 0 7 で対象リンクの表示状態を「可変リンク」を表す「点線」とする。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 表示が変化する（ステップ S 1 3 0 6）。以上のダイアログウインドウ 7 0 1 の設定情報はメモリに格納される。

10

【0118】

なお、Distance 1 4 1 1 に設定される現在の値には、現在のレイアウトとして配置されているコンテナの間の距離がデフォルト値として自動的に入力されるようにしてもよい。

【0119】

図 1 5 は固定サイズのリンクを使用した場合であり、図 1 2 に記載されている基本レイアウトにコンテンツを流し込んだレイアウト結果を示している。なお、図 1 2 のリンク 1 2 0 9 は可変リンクとなっているが、固定リンクが設定されているものと想定する。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。

【0120】

例えば図 1 2 においてコンテナ 1 2 0 3 とコンテナ 1 2 0 4 にそれぞれ違ったサイズのイメージデータ（1 5 0 4 および 1 5 0 5）が挿入された場合を考える。

20

【0121】

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 1 2 0 3 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サイズを変更しようとする。同様にコンテナ 1 2 0 4 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。

【0122】

しかしコンテナ 1 2 0 3 とコンテナ 1 2 0 4 はアンカー 1 2 0 1 とアンカー 1 2 0 2 によってそれぞれ左辺 1 2 1 2 と右辺 1 2 1 3 の移動ができず、上記のようにサイズを変更しようとすると両者の間隔を狭めるしかない。しかしながら、コンテナ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維持されるため、コンテナ 1 2 0 3 とコンテナ 1 2 0 4 のサイズが変更されることになる。

30

【0123】

その結果、コンテナ 1 2 0 3 とコンテナ 1 2 0 4 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出来ず、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。すなわちリンク 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（図 1 5 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）。

【0124】

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。この場合、上記の例でコンテナ 1 2 0 3 とコンテナ 1 2 0 4 の間には図示の通り可変サイズのリンク 1 2 0 9 が設定されている。したがって、コンテナ 1 2 0 3 とコンテナ 1 2 0 4 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 2 0 3 とコンテナ 1 2 0 4 のサイズを図 1 5 の例より大きくすることができる。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出来る。図 1 6 はこの結果を示しており、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。なお、この場合コンテナ 1 2 0 3 とコンテナ 1 2 0 4 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。

40

50

【 0 1 2 5 】

[複数種類のレイアウト計算方法を用いた自動レイアウト処理]

< 全体フロー >

次に、本実施形態による、複数種類のレイアウト計算方法を用いた自動レイアウト処理について説明する。

【 0 1 2 6 】

本実施形態では、複数のコンテナを有するドキュメントテンプレートのレイアウトを、これら複数のコンテナの各々に流し込まれるコンテンツデータに応じて動的に変更する自動レイアウト処理を説明する。図 1 7 は本実施形態による自動レイアウト処理の全体的な処理の流れを示すフローチャートである。まず、ステップ S 1 7 0 1 でレイアウトを計算するコンテナの集合を求める。ここで言うコンテナの集合とは、リンクで直接或いは間接に相互に結ばれているコンテナ全てを要素とする集合体のことであり、リンクセットともいい、前述したようにリンクにより関連付けられているコンテナ群を指す。このように、ステップ S 1 7 0 1 では、レイアウト全体に存在するリンクセットを見つける処理が行なわれる。

10

【 0 1 2 7 】

次に、ステップ S 1 7 0 2 ではステップ S 1 7 0 1 で求めたリンクセットから 1 つのリンクセットが選択される。そしてステップ S 1 7 0 3 において、ステップ S 1 7 0 2 で選択された 1 つのリンクセットに関してレイアウトの最適化を行う。ステップ S 1 7 0 3 におけるレイアウトの最適化とは、レイアウト計算を行うということに等しく、この処理については「レイアウトの最適化フロー」にて、図 1 8 を参照して詳細に説明する。

20

【 0 1 2 8 】

次にステップ S 1 7 0 4 ではルールを違反していないかについてチェックを行っている。ここで言うルールとはオフセットルールのことであり、オフセットルールとはレイアウトの制約のことである。レイアウトの制約とは、コンテナや可変リンクの最大値・最小値、またページ枠の位置などを対象としたものであり、レイアウト計算を行う上で破ってはいけないルールである。言い換えると、レイアウト計算はオフセットルールを破らない範囲内で行われることになる。ステップ S 1 7 0 4 においてルールを違反していると判定された場合はステップ S 1 7 0 3 に戻り、再度レイアウトの最適化を行うこととなる。ルールを違反していないと判定された場合は、ステップ S 1 7 0 4 からステップ S 1 7 0 5 へ進む。ステップ S 1 7 0 5 では、ステップ S 1 7 0 1 で見つけた全てのリンクセットについて上記処理（レイアウトの最適化）を終了したかどうかについて判断される。未処理のリンクセットがあればステップ S 1 7 0 2 へ戻り、別のリンクセットを選択し、レイアウトの最適化を行っていく。もし全てのコンテナ集合について計算が終了した場合は、レイアウト全体フローを終了する。

30

【 0 1 2 9 】

こうしてレイアウト全体フローを完了すると、図 8 のステップ S 8 0 4 で示したようにプレビューとして表示したり、プリント出力のためのデータとして外部へ出力したりすることになる。

【 0 1 3 0 】

[レイアウト最適化フロー]

図 1 8 は本実施形態によるレイアウト最適化処理を説明するフローチャートである。以下、同図を参照して本実施形態の最適化レイアウト処理を説明する。

40

【 0 1 3 1 】

まず、ステップ S 1 8 0 1 で、レイアウト編集アプリケーションはステップ S 1 7 0 2 で選択されたリンクセット（集合）からコンテナを 1 つ選択する。詳細には、マウス等を用いて選択されたコンテナを入出力インタフェース 1 4 3 を介して認識することにより、ドキュメントテンプレート上のどのコンテナが選択されたかを認識することが可能となる。なお、S 1 8 0 1 により選択されたコンテナは一時的にメモリ 1 3 6 に保持されることとなる。

50

【0132】

次にステップS1802において、レイアウト編集アプリケーション121は、S1801により選択されたコンテナに適用するレイアウト調整アルゴリズムを取得する。これは、メモリ136或いは格納デバイス139によって保持されているアルゴリズム情報をパース（解析）することにより取得される。レイアウト調整アルゴリズム情報は図25等により後述するユーザインターフェースを用いて設定され、例えば図30（後述）に示されるようにメモリ136或いは格納デバイス139に保持される。ここではそのようなレイアウト調整アルゴリズム情報を探索し、ステップS1801で選択したコンテナに対してレイアウト計算を行うために適用すべきレイアウト調整アルゴリズムとして決定する。なお、ステップS1802の処理については、「適用アルゴリズムのパースフロー」にて、図19を参照して詳細に説明する。 10

【0133】

ステップS1803において、レイアウト編集アプリケーション121は、ステップS1802におけるレイアウト調整アルゴリズムの探索結果に応じて処理を分岐する。ステップS1801で選択されたコンテナに適用すべきレイアウト調整アルゴリズムが「コンテンツ優先」とであると判定した場合、処理はステップS1803からステップS1804に進む。ステップS1804において、レイアウト調整アルゴリズムは、ステップS1801で選択されたコンテナに挿入されるコンテンツ（データベースからの挿入データ）のサイズがそのコンテナの計算用の基準サイズであると設定し、メモリ136に一時的に保持する。例えばイメージであれば、当該イメージデータが有するサイズを計算用の基準サイズとする。また、テキストデータの場合は、文字数と、該テキストデータを流し込むべきコンテナの属性（テキストデータを反映する際のフォントタイプ、フォントサイズなどの文字属性）に基づいて、計算用の基準サイズが決定される。 20

【0134】

又、レイアウト編集アプリケーション121は、ステップS1801で選択したコンテナに対する適用レイアウト調整アルゴリズムが「コンテナ・コンテンツ優先」と判定した場合は、処理をステップS1805に進める。ステップS1805では、ステップS1801で選択されたコンテナ自身がデザインされた時（基本レイアウト時）のコンテナサイズがコンテナの計算用の基準サイズとして設定される。そして、当該コンテナに挿入されるコンテンツ（データベースからの挿入データ）のサイズと共に、設定された計算用の基準サイズをメモリ136に一時的に保持する。 30

【0135】

又、レイアウト編集アプリケーション121は、ステップS1801で選択されたコンテナに対する適用レイアウト調整アルゴリズムが「基本デザイン優先」と判定した場合、処理をステップS1806に進める。ステップS1806では、ステップS1801で選択されたコンテナのデザインサイズ（基本レイアウト時のコンテナサイズ）をそのコンテナの計算用の基準サイズとして設定し、メモリ136に一時的に保持する。なお、本願では計算用の基準サイズの変わりに目標形状ということもある。

【0136】

ステップS1804～S1806の終了後、ステップS1807に進み、全てのコンテナの適用レイアウト調整アルゴリズムをパースしたかどうかチェックされる。パースを行っていないコンテナがある場合は、ステップS1801へ戻り、当該リンクセットから別のコンテナが選択され、選択されたコンテナの適用レイアウト調整アルゴリズムがパースされる。リンクセット内の全てのコンテナについてパースが終了したと判定された場合はステップS1808に進み、レイアウト再計算処理が行われる。レイアウト再計算処理とは、適用されたレイアウト調整アルゴリズムに従ってそれぞれのリンクセット内のレイアウト計算を行う処理である。ステップS1808の処理については後の「レイアウト再計算処理フロー」にて詳しく説明をする。 40

図18の処理により、レイアウト編集アプリケーションは、複数のレイアウトアルゴリズムからコンテナに適用すべきアルゴリズムを選択して、コンテナ毎に選択されたアルゴ 50

リズムに従ってレイアウト調整を実行することが可能となる。

【0137】

〔適用レイアウト調整アルゴリズムのパスフロー〕

図19は、ステップS1801で選択されたコンテナに対して、ステップS1802で行われる適用レイアウト調整アルゴリズムのパスを示すフローチャートである。まず、ステップS1901においてレイアウト編集アプリケーション121は、ドキュメント指定アルゴリズムをコンテナの適用レイアウト調整アルゴリズムに設定する。つまり、デフォルトの設定としてドキュメントに設定されているレイアウト調整アルゴリズムを、とりあえずそのコンテナの適用アルゴリズムとして設定する。例えば、図30のようなレイアウト調整アルゴリズム情報においては、ドキュメントの階層に設定されている「コンテンツ優先」というレイアウト調整アルゴリズムが設定されることになる。

10

【0138】

次に、ステップS1902において、レイアウト編集アプリケーション121は、コンテナに直接指定されているレイアウト調整アルゴリズムがあるかをチェックする。詳細には、メモリ136にコンテナ属性が保持されており、該メモリ136を参照することにより、直接指定されているレイアウト調整アルゴリズムがあるか否かを判定することが可能となる。

【0139】

レイアウト編集アプリケーション121が、S1902においてコンテナに直接指定されているレイアウト調整アルゴリズムがあると判定した場合、ステップS1903に進み、その指定アルゴリズムを当該コンテナに設定し、本パス処理を終了する。

20

【0140】

当該コンテナに直接指定されているレイアウト調整アルゴリズムがない場合は、処理はステップS1904に進む。ステップS1904では、レイアウト編集アプリケーション121は、当該コンテナを含むエリア（ある一定の範囲）に対してレイアウト調整アルゴリズムが指定されているかをチェックする。ここで言うエリアとは、ユーザが任意に選択した範囲や、図18で説明したリンクセット（集合）を指すことができる。レイアウト編集アプリケーション121が、エリアに指定されているレイアウト調整アルゴリズムがあると判定した場合、ステップS1905に進み、その指定アルゴリズムを該エリア内の当該コンテナに設定し、パス処理を終了する。なお、エリアに対してレイアウト調整アルゴリズムが指定されていた場合、これらの情報についてもメモリ136等に保持されているため、レイアウト編集アプリケーション121が、メモリ136を参照することにより、S1904は判定可能となる。

30

【0141】

エリアに指定されているレイアウト調整アルゴリズムがない場合はステップS1906に進む。ステップS1906では、レイアウト編集アプリケーション121は、当該コンテナに流し込まれるコンテンツデータを含むレコードにおいて、使用するレイアウト調整アルゴリズムが指定されているかをチェックする。例えば、レコード1ではこのレイアウト調整アルゴリズム、レコード2ではこのレイアウト調整アルゴリズムというようにレコード単位でレイアウト計算に使用するレイアウト調整アルゴリズムを変更することが出来る。なお、レコード単位に使用するレイアウト調整アルゴリズムが指定されている場合、これらの情報についてもメモリ136等に保持されている。このため、レイアウト編集アプリケーション121が、メモリ136を参照することにより、ステップS1906の判定が実行可能となる。

40

【0142】

レコードに指定されているレイアウト調整アルゴリズムがあればステップS1907に進み、現在のレコードの指定アルゴリズムをコンテナに設定し、パス処理を終了する。

【0143】

レコードに指定されているレイアウト調整アルゴリズムがない場合はステップS1908に進む。ステップS1908では、レイアウト編集アプリケーション121は、当該コ

50

ンテナを含むページ内で使用されるレイアウト調整アルゴリズムが指定されているかをチェックする。ここで言うページ内とはページ上にある全てのコンテンツに適用するための設定であり、ページ単位に異なるレイアウト調整アルゴリズムを指定することが出来る。なお、ページ単位に使用するレイアウト調整アルゴリズムが指定されている場合、これらの情報についてもメモリ136等に保持されているため、レイアウト編集アプリケーション1212が、メモリ136を参照することにより、S1906は判定可能となる。ページに指定されているレイアウト調整アルゴリズムがあればステップS1909に進み、現在のページの指定アルゴリズムをコンテンツに設定し、パース処理を終了する。ステップS1908においてページに指定されているレイアウト調整アルゴリズムがないと判定された場合は、S1902、S1904、S1906、S1908の判定処理により、全てにおいてNoと判定されたこととなる。この場合、レイアウト編集アプリケーション121は、S1901により設定したドキュメント指定アルゴリズムをドキュメント内のコンテンツに適用して、フローを終了する。

10

【0144】

以上のパース処理により、図30に示す階層構造の、下位階層に設定された状態が優先的にコンテンツに設定されていくことになる。

【0145】

[再計算処理フロー]

図20は本実施形態によるレイアウト再計算処理(ステップS1808)を示すフローチャートである。

20

【0146】

まず、ステップS2001で、レイアウト編集アプリケーション121は、マークを選択する。マークとはコンテンツ枠線のことで、コンテンツのどの辺を動かすのかを決めるために選択する。なお、ここで選択される枠線は、コンテンツの移動可能な辺である。

【0147】

そして、ステップS2002において、レイアウト編集アプリケーション121は、テンションの計算を行う。テンションとは計算用の基準サイズと現在のコンテンツサイズの差に対応する。なお、計算用の基準サイズと現在のコンテンツサイズの差の計算方法はレイアウト調整アルゴリズム毎に異なる。レイアウト計算においては、この差を縮めていく(テンションを小さくしていく)ことによってレイアウトを決定していく。また、計算用の基準サイズは上述した通り、適用されるレイアウト調整アルゴリズムに応じて変わるものであり、メモリ136に一時的に保持されている。さらに、現在のコンテンツサイズについても一時的にメモリ136に保持されているため、レイアウト編集アプリケーションはS2002において、メモリ136を参照することによりテンションの計算を行うことが可能となる。

30

【0148】

ステップS2003では、レイアウト編集アプリケーション121は、ステップS2002で計算したテンションに基づいてマークを移動する。なお、移動した後の位置情報(座標値など)およびマークの移動処理が終了したことは、メモリ136に保持される。

【0149】

ステップS2004において全てのマークを選択したかをチェックしている。全てのマークとは全てのコンテンツのエッジ(枠)のことであり、リンクセット内における全コンテンツのテンション計算を行い、マークの移動が行われたか否かについてチェックする。S2003により、マークの移動処理が行われたものについては、移動処理終了であることがメモリ136に保持されているため、レイアウト編集アプリケーションがメモリ136を参照することにより判定可能である。なお、未選択のマークが存在するならば、ステップS2001に進み、未選択のマークを選択してステップS2002以降の処理を繰り返す。全てのマークを選択し終わっていた場合はフローを終了する。

40

【0150】

[各種レイアウト調整アルゴリズムの計算方法]

50

次に、ステップ S 2 0 0 2 , S 2 0 0 3 で実行されるいくつかのレイアウト調整アルゴリズムの計算方法の一例を説明する。なお、以下では一方向（横方向）にコンテナをレイアウト調整しているが、縦方向にレイアウト調整する場合も同様である。また 2 方向（縦横方向）にコンテナをレイアウト調整する処理に拡張できることが明らかであり、一つのアルゴリズム計算方法としては上述した最小 2 乗法を用いるものが挙げられる。また、コンテナのレイアウトを調整するためのレイアウト調整アルゴリズムは本実施形態で示されたものに限られるものではない。

【 0 1 5 1 】

図 2 1 の (A) ~ (C) はコンテンツ優先のレイアウト調整アルゴリズムを説明する図である。図 2 1 の (A) に示されるように、コンテナ 2 1 0 1 とコンテナ 2 1 0 2 が存在し、それらはリンク 2 1 0 3 によって関連付けされている。それぞれのコンテナにはデータベースから挿入されるデータ（コンテンツ）が存在し、コンテナ 2 1 0 1 にはコンテンツ 2 1 0 4、コンテナ 2 1 0 2 にはコンテンツ 2 1 0 5 が挿入されている。現在の各コンテナの幅サイズ（コンテンツ 2 1 0 4 およびコンテンツ 2 1 0 5 を挿入する前のコンテナの幅サイズ）はコンテナ 2 1 0 1 が 5、コンテナ 2 1 0 2 が 7、各コンテンツサイズの幅サイズはコンテンツ 2 1 0 4 が 8、コンテンツ 2 1 0 5 が 4 である。

10

【 0 1 5 2 】

コンテンツ優先のレイアウト調整アルゴリズムではこれらの情報を元に、レイアウト計算を行う。コンテンツ優先のレイアウト調整アルゴリズムでは上述した通りコンテンツサイズを計算用の基準サイズとみなし、コンテナサイズをレイアウト調整していく。レイアウト計算では現在のコンテナサイズ（コンテナ 2 1 0 1、コンテナ 2 1 0 2）と計算用の基準サイズ（コンテンツ 2 1 0 4、コンテンツ 2 1 0 5）との差（テンション）を縮めるよう現在のコンテナをレイアウト調整する。そして、このレイアウト調整によって、計算用の基準サイズに近づけようとする。

20

【 0 1 5 3 】

図 2 1 の (B) はテンションの計算処理を図示している。図 2 1 の (B) に示されるように、コンテナ 2 1 0 1 のテンション T_1 は $8 - 5 = 3$ 、コンテナ 2 1 0 2 のテンション T_2 は $7 - 4 = 3$ である。よってこのリンクセットのテンションの合計は 6 である。

【 0 1 5 4 】

このテンションを減らすようにマークの移動を行った結果が図 2 1 の (C) に示されている。ここではコンテナ 2 1 0 1 のサイズは 8、コンテナ 2 1 0 2 のサイズは 4 となっており、ともに計算用の基準サイズ（コンテンツ 2 1 0 4 のサイズ 8 とコンテンツ 2 1 0 5 のサイズ 4）を達成している。つまりレイアウト計算によるレイアウト結果のテンションは 0 となる。よって、コンテンツ優先のレイアウト調整アルゴリズムを使用した場合、コンテンツの大きさが優先され、コンテナサイズが決まることになる。

30

【 0 1 5 5 】

図 2 2 の (A) ~ (C) はコンテナ・コンテンツ優先のレイアウト調整アルゴリズムを説明する図である。なお、デザインとはユーザがレイアウト作成モードにおいてコンテナやリンクを作成していたときの状態（基本レイアウト）のことを指す。デザインサイズとはそれらのオブジェクトを作成したとき（デザイン時）のサイズである。

40

【 0 1 5 6 】

図 2 2 の (A) に示されるように、コンテナ 2 2 0 1 とコンテナ 2 2 0 2 が存在し、それらはリンク 2 2 0 3 によって関連付けされている。それぞれのコンテナにはデータベースから挿入されるデータ（コンテンツ）が存在し、コンテナ 2 2 0 1 にはコンテンツ 2 2 0 4、コンテナ 2 2 0 2 にはコンテンツ 2 2 0 5 が挿入されている。現在の各コンテナの幅サイズ（コンテンツ 2 2 0 4 およびコンテンツ 2 2 0 5 が流し込まれる前のサイズ）はコンテナ 2 2 0 1 が 5、コンテナ 2 2 0 2 が 7 であり、各コンテンツサイズの幅サイズはコンテンツ 2 2 0 4 が 8、コンテンツ 2 2 0 5 が 4 である。また、デザイン時のコンテナサイズ 2 2 0 6 と 2 2 0 7 が存在する。

【 0 1 5 7 】

50

つまり、図 2 2 の (A) の例では、ユーザは基本レイアウトとしてコンテナ 2 2 0 1 およびコンテナ 2 2 0 2 の幅サイズが 6 として設定されている。そして、例えば最初のレコードにおけるコンテンツデータを流し込んだ結果、レイアウト計算が行われコンテナサイズが現在のサイズ (コンテナ 2 1 0 1 が 5、コンテナ 2 2 0 2 が 7) に変更されている。
【 0 1 5 8 】

この状態で、次にコンテナ・コンテンツ優先のレイアウト調整アルゴリズムを用いてレイアウト計算を実行するレコードを流し込もうとしている状態が図 2 2 の (A) である。
【 0 1 5 9 】

コンテナ・コンテンツ優先のレイアウト調整アルゴリズムではコンテンツサイズおよび現在のコンテナサイズおよびデザイン時 (基本レイアウト時) のコンテナサイズを考慮してレイアウト計算を行う。コンテナ・コンテンツ優先のレイアウト調整アルゴリズムでは上述した通りコンテンツサイズに加え、デザインサイズを加えたものを計算用の基準サイズとみなし、コンテナサイズをレイアウト調整していく。つまり、レイアウト計算では、現在のコンテナサイズ (コンテナ 2 2 0 1、コンテナ 2 2 0 2) とコンテンツサイズ (コンテンツ 2 2 0 4、コンテンツ 2 2 0 5) との差 (テンション)、そして現在のコンテナサイズ (コンテナ 2 2 0 1、コンテナ 2 2 0 2) とデザインサイズ (コンテンツ 2 2 0 6、コンテンツ 2 2 0 7) との差を縮めるよう現在のコンテナをレイアウト調整させることによって、計算用の基準サイズに近づけようとする。

【 0 1 6 0 】

図 2 2 の (B) はテンションの計算処理を図示している。各コンテナのテンションは次のとおりである。

【 0 1 6 1 】

コンテナ 2 2 0 1 に関しては、コンテンツサイズと現在のコンテナサイズとのテンション (T_1) が $| 8 - 5 | = 3$ 、デザインサイズと現在のコンテナサイズとのテンション (T_3) が $| 5 - 6 | = 1$ となる。また、コンテナ 2 2 0 2 に関しては、コンテンツサイズと現在のコンテナサイズのテンション (T_2) が $| 4 - 7 | = 3$ 、デザインサイズと現在のコンテナサイズとのテンション (T_4) が $| 6 - 7 | = 1$ となる。よってこのリンクセットのテンションの合計は $3 + 1 + 3 + 1 = 8$ である。

【 0 1 6 2 】

このテンションを減らすようにマークの移動を行った結果が図 2 2 の (C) に示した状態である。

【 0 1 6 3 】

ここではコンテナ 2 2 0 1 のサイズは 7、コンテナ 2 2 0 2 のサイズは 5 となっている。よってレイアウト結果時のテンションは、コンテナ 2 2 0 1 に関しては、コンテンツサイズとのテンション (T_1) が $| 8 - 7 | = 1$ 、デザインサイズとのテンション (T_3) が $| 6 - 7 | = 1$ となる。また、コンテナ 2 2 0 2 に関しては、コンテンツサイズとのテンション (T_2) が $| 4 - 5 | = 1$ 、デザインサイズとのテンション (T_4) が $| 6 - 5 | = 1$ となる。よってこのリンクセットのテンションの合計は $1 + 1 + 1 + 1 = 4$ である。この場合、テンション 0 の状態を達成できなかったが、これ以上、テンションが減らす計算方法がないため、レイアウト計算は終了される。

【 0 1 6 4 】

このように、コンテンツサイズ、デザインサイズは一般的に異なるサイズであり、コンテンツサイズおよびデザインサイズの両者を満足させるためにコンテナをレイアウト調整させる際は、概してテンションが 0 にならない。従って、テンションが最小となるようにマークを移動し、それをレイアウト結果とする。よって、コンテナ・コンテンツ優先のレイアウト調整アルゴリズムを使用した場合、コンテンツの大きさとデザイン時のコンテナサイズの双方を吟味したレイアウト結果を得ることが出来る。

【 0 1 6 5 】

図 2 3 の (A) ~ (C) は基本デザイン優先のレイアウト調整アルゴリズムを説明する図である。本実施形態において、基本デザイン優先とは、デザイン時のサイズのみを計算

10

20

30

40

50

用の基準サイズとしてレイアウト計算する方法である。

【0166】

図23の(A)に示されるように、コンテナ2301とコンテナ2302が存在し、それらはリンク2303によって関連付けされている。それぞれのコンテナにはデータベースから挿入されるデータ(コンテンツ)が存在し、コンテナ2301にはコンテンツ2304、コンテナ2302にはコンテンツ2305が挿入されている。現在の各コンテナの幅サイズ(コンテンツ2304およびコンテンツ2305が流し込まれる前の各コンテナの幅サイズ)はコンテナ2301が5、コンテナ2302が7、各コンテンツサイズの幅サイズはコンテンツ2304が8、コンテンツ2305が4である。また、デザイン時(基本レイアウト)のコンテナサイズ2306と2307が存在する。なお、デザイン時のコンテナサイズ2306および2307は6である。

10

【0167】

基本デザイン優先のレイアウト調整アルゴリズムではこれらのうち、現在のコンテナサイズとデザインサイズの情報を元に、コンテナのレイアウト調整を行う。よってレイアウト計算では現在のコンテナサイズ(コンテナ2301、コンテナ2302)と計算用の基準サイズ(デザインサイズ2306、デザインサイズ2307)との差(テンション)を縮めるよう現在のコンテナをレイアウト調整させることによって、計算用の基準サイズに近づけようとする。

【0168】

図23の(B)はテンションの計算を行う状態を図示している。現在、各コンテナのテンションはコンテナ2301の計算用の基準サイズとのテンション(T_3)が $6 - 5 = 1$ となる。そしてコンテナ2302の場合は、計算用の基準サイズとのテンション(T_4)が $7 - 6 = 1$ となる。よってこのリンクセットのテンションの合計は $1 + 1 = 2$ である。

20

【0169】

図23の(C)は、上記のテンションを減らすようにマークの移動を行った結果を説明しており、コンテナ2301、2302、コンテンツ2304、2305の調整後の状態が示されている。ここではコンテナ2301のサイズは6、コンテナ2302のサイズは6となっている。よってレイアウト結果時のテンションは、コンテナ2301は計算用の基準サイズとのテンションが $6 - 6 = 0$ 、そしてコンテナ2302は計算用の基準サイズとのテンションが $6 - 6 = 0$ となる。よってこのリンクセットのテンションの合計は $0 + 0 = 0$ である。つまりこのレイアウト結果では計算用の基準サイズが達成されている。

30

【0170】

なお、基本デザイン優先のレイアウト調整アルゴリズムを使用した場合は、コンテンツの大きさに関わらず、ユーザのデザイン時の大きさが優先され、コンテナサイズが決まることになる。

【0171】

しかし、このレイアウト調整アルゴリズムはコンテンツの大きさとは全く関係がないため、レイアウト計算における可変要素がなくなってしまう。つまり、コンテナは常にデザイン時の大きさになってしまう。よって基本デザイン優先のレイアウト調整アルゴリズムはリンクセット内で他のレイアウト調整アルゴリズムと共に用いられることにより効果を発揮する。

40

【0172】

図24の(A)~(C)は基本デザイン優先のレイアウト調整アルゴリズムとコンテンツ優先のレイアウト調整アルゴリズムが1つのリンクセット内で別々のコンテナに設定されている場合のレイアウト計算処理について説明する図である。

【0173】

図24の(A)に示されるように、コンテナ2401とコンテナ2402が存在し、それらはリンク2403によって関連付けされている。又、コンテナ2401には基本デザイン優先が、コンテナ2402にはコンテンツ優先がそれぞれ設定されている。それぞれのコンテナにはデータベースから挿入されるデータ(コンテンツ)が存在し、コンテナ2

50

4 0 1 にはコンテンツ 2 4 0 4、コンテナ 2 4 0 2 にはコンテンツ 2 4 0 5 が挿入されている。現在の各コンテナの幅サイズはコンテナ 2 4 0 1 が $|a - b| = 5$ 、コンテナ 2 4 0 2 が $|e - h| = 7$ 、各コンテンツサイズの幅サイズはコンテンツ 2 4 0 4 が $|a - d| = 8$ 、コンテンツ 2 4 0 5 が $|g - h| = 4$ である。また、デザイン時のコンテナサイズ 2 4 0 6 と 2 4 0 7 が存在する。なお、デザイン時のコンテナサイズは 2 4 0 6 および 2 4 0 7 とともに 6 である。

【0 1 7 4】

図 2 4 の (B) に示されるように、コンテナ 2 4 0 1 に設定されているレイアウト調整アルゴリズムは基本デザイン優先である。そのため、レイアウト計算では現在のコンテナサイズ (コンテナ 2 4 0 1) とデザインサイズ (デザインサイズ 2 4 0 6) との差 (テンション T_1) が使用される。またコンテナ 2 4 0 2 にはコンテンツ優先のレイアウト調整アルゴリズムが設定されているため、現在のコンテナサイズ (コンテナ 2 4 0 2) とコンテンツサイズ (コンテンツ 2 4 0 5) との差 (テンション T_2) が使用される。これらのテンションを小さくさせ、現在のコンテナをレイアウト調整させることによって、計算用の基準サイズに近づけようとする。

10

【0 1 7 5】

図 2 4 の (B) において、コンテナ 2 4 0 1 のテンションはコンテナ 2 4 0 1 のデザインサイズ (6) と現在のコンテナサイズ (5) とのテンション $T_1 = |b - c| = 1$ となる。また、コンテナ 2 4 0 2 のテンションは、現在のコンテナサイズ (7) とコンテンツサイズ (4) とのテンション $T_2 = |e - g| = 3$ となる。よってこのリンクセットのテンションの合計は $T_1 + T_2 = |b - c| + |e - g| = 4$ である。

20

【0 1 7 6】

図 2 4 の (C) は、上記のテンションを減らすようにマークの移動を行った結果を示している。ここではコンテナ 2 4 0 1 のサイズは $|a - b| = 7$ 、コンテナ 2 4 0 2 のサイズは $|e - h| = 5$ となっている。よってレイアウト結果において、コンテナ 2 4 0 1 のテンションはデザインサイズ (6) と現在のコンテナサイズ (7) のテンション $T_1 = |b - c| = 1$ となる。また、コンテナ 2 4 0 2 のテンションは現在のコンテナサイズ (5) とコンテンツサイズ (4) とのテンション $T_2 = |e - g| = 1$ となる。よってこのリンクセットのテンションの合計は $T_1 + T_2 = 2$ である。この場合、テンション 0 の状態を達成できなかったが、これ以上、テンションが減らないため、レイアウト計算は終了される。こうして、図 2 4 の (C) におけるコンテナ 2 4 0 1 のサイズ (7) はデザイン時のサイズ (6) に、図 2 4 の (C) におけるコンテナ 2 4 0 2 のサイズ (5) は挿入されたコンテンツのサイズ (4) にそれぞれ近づいてレイアウトされた。

30

【0 1 7 7】

このように、基本デザイン優先のレイアウト調整アルゴリズムはその他のレイアウト調整アルゴリズムと共用した場合に効果を発揮する。また各コンテナに様々なレイアウト調整アルゴリズムが適用できることで、ユーザの求める多様なレイアウトのニーズに応えることができる。なお、上記では基本デザイン優先とコンテンツ優先の組み合わせを説明したが、他の組み合わせも可能であることはいうまでもない。また、リンクセットに 3 つ以上のコンテナが存在する場合も、上記 2 つのコンテナの場合の計算処理に準じて行えばよい。

40

【0 1 7 8】

続いて、各レイアウトアルゴリズムの計算方法について説明する。なお、これのレイアウトアルゴリズムは可変コンテナに対して設定される。固定コンテナは、どのようなレイアウトアルゴリズムが適用されてもコンテナサイズの変更が一切生じないため、本願のレイアウトアルゴリズムを適用してもレイアウト変更処理が実行されない。なお、固定コンテナのレイアウト位置は、可変コンテナとリンクで接続された場合、変更される可能性が生じる。

【0 1 7 9】

[第 1 レイアウトアルゴリズム (コンテンツサイズを考慮)]

50

図 3 2 を用いて、第 1 レイアウトアルゴリズムを説明する。なお、第 1 レイアウトアルゴリズムは、コンテンツサイズを計算用の基準サイズとして計算するため、「コンテンツ優先レイアウトのアルゴリズム」とする。図 3 2 において、3 2 0 1 ~ 3 2 0 3 は、デザイン時（コンテナ等を含むテンプレート作成時）の基本コンテナである。3 2 0 5 ~ 3 2 0 7 は、各コンテナに挿入されるコンテンツである。また、3 2 0 4 は、各コンテナを関連付けるリンクであり、図 3 2 のリンクは、固定リンクとする。また、3 2 0 8 ~ 3 2 1 0 は、第 1 レイアウトアルゴリズム（コンテンツ優先レイアウトのアルゴリズム）を適用した結果を示す。

【 0 1 8 0 】

3 2 0 1 ~ 3 2 0 3 で示すコンテナ A、B、C は固定リンク 3 2 0 4 にて関連付けられている。各コンテナおよびリンクの水平方向のサイズは図示した通りであり、水平方向に合計 5 7 というサイズを持っている。このレイアウトでは、コンテナ A の左辺とコンテナ C の右辺の位置が固定になるように設定されているとする。そのため、水平方向の合計 5 7 という値は固定である。このレイアウトに 3 2 0 5 ~ 3 2 0 7 で示したコンテンツ A'、B'、C' が各コンテナへ挿入されると、テンション（レイアウトの負荷）が発生する。

【 0 1 8 1 】

それぞれのコンテンツは実サイズを持っている。仮に、各コンテンツの実サイズは図示した通りとする（A' は 1 7、B' は 1 3、C' は 2 7）。なお、コンテンツがテキストの場合、計算用の基準サイズは、データの流し込み先であるコンテナに設定されたフォントサイズや縦横サイズの変可 / 固定、最大値 / 最小値等に基づいて決定される。また、コンテンツがイメージの場合、当該イメージデータのサイズ（縦横画素数）が実サイズである。このコンテンツをコンテナに挿入した後にすべてのコンテンツを実サイズでレイアウトするためには、 $17 + 5 + 13 + 7 + 27 = 69$ の水平サイズが必要になるが、このレイアウトは水平サイズが 5 7 で固定になっている。したがって $69 - 57 = 12$ がこのレイアウトにかかるテンションということになる。

【 0 1 8 2 】

次に、上記テンションを各コンテナに分配する。本願のレイアウトアルゴリズムでは、このテンションが各コンテナにおいてつり合うように計算するものであり、そのときに用いる手法の一例が最小 2 乗法である。まず、第 1 レイアウトアルゴリズムにおける各コンテナのテンション T の計算方法は、

$$T = | \text{レイアウト結果のサイズ} - \text{コンテンツサイズ} | \quad \dots (1)$$
となる。

なお、テンションの値は必ずしも正になるとは限らないので、絶対値を用いる必要がある。

【 0 1 8 3 】

そして、(1) 式を用いて各コンテナにて算出された T の合計が現在対象としている集合の合計テンションとなる。ここで、図 3 2 を用いて説明する。各コンテナの計算後のサイズを a、b、c とすると、各コンテナにて発生するテンションを適切に分配するためには、

$$t = (a - 17)^2 + (b - 13)^2 + (c - 27)^2 \quad \dots (2)$$

という計算処理を行うことが望ましい。なお、t は、各コンテナにおけるテンションがつり合う値を算出するための変数である。この変数 t が最小値となった場合、各コンテナのテンションがつり合っている状態である。

【 0 1 8 4 】

図 3 2 のレイアウトは、単純なレイアウトであり、かつ、優先度の設定はされていないため、式 (2) を用いなくとも変数 t が最小になるのは、

$$| a - 17 | = | b - 13 | = | c - 27 | \quad \dots (3)$$

と表すことができる。

さらに、計算後のレイアウトのサイズより、

$$a + 5 + b + 7 + c = 57 \quad \dots (4)$$

10

20

30

40

50

と表すことができる

【0185】

(3) および (4) により、各コンテナの計算後のサイズが求められる。ここでは、 $a = 13$ 、 $b = 9$ 、 $c = 23$ と算出された。よって、コンテンツと計算結果の差分を算出すると、各コンテナの差分は4となる。つまり、各コンテナにかかるテンションは4であり、計算後の各コンテナ(3208 ~ 3210) A"、B"、C"のサイズは図示したとおりになる。

【0186】

しかし、図32の場合、式(3)、(4)により算出しているため、式(2)に記載した最小2乗法を用いなくてもレイアウト結果を算出することができる。これは図32のレイアウトが単純なレイアウトであることに起因する。つまり、生成されたテンプレートによっては、式(2)の最小2乗法を使うことが望ましい。その例を図33を用いて説明する。

【0187】

図33のようなレイアウトにおいて、最小2乗法を用いた算出方法および結果と上述した最小2乗法を用いない場合の算出方法および結果を記述する。なお、図33では、仮にコンテナAに対してサイズ15、コンテナBに対してサイズ25、コンテナCに対してサイズ5のコンテンツが入力されるとする。現状のコンテナサイズは、コンテナAおよびBのサイズが10、コンテナCのサイズが9とする。本説明では、第1レイアウトアルゴリズムを対象としているので計算用の基準サイズはコンテンツサイズとなる。また、3301から3302の距離を23とする。なお、図33では、コンテナAの左辺とコンテナBの右辺の位置が固定となるように設定されていると仮定するため、3301から3302の距離は固定となる。各コンテナを関連付ける固定リンク3307のサイズは1とする。

【0188】

(1) 最小2乗法を用いない場合

最小2乗法を用いない場合、各コンテナについてレイアウト算出結果のコンテナサイズが、計算用の基準サイズより大きくなるか否かを判別する。図33の場合、各コンテナのレイアウト位置と制約条件(3301と3302が固定)、さらにコンテンツサイズを考慮する必要がある。コンテナAは、左辺が固定であり、右辺が可変であるため右方向へのサイズ変更および移動が可能である。さらに、コンテナA3303およびコンテナB3304に挿入されるコンテンツサイズの合計値(=35)が上述した制約条件(3301 ~ 3302の距離=23)よりも大きくなる。つまり、コンテナAおよびBに流し込まれた各コンテンツデータは、縮小しなければならないと判断できる。よって、レイアウト計算後のコンテナAのサイズaは、計算用の基準サイズ(=15)よりも小さくなると判別できる。同様にコンテナBについても解析することにより、コンテナBも、計算用の基準サイズ(=25)よりも小さくなると判別できる。

【0189】

一方、コンテナCは、現状のレイアウト位置や制約条件、コンテンツサイズからだけでは、レイアウト計算後のコンテナサイズが予測できない。よって、図33のレイアウトでは、コンテナCの計算後のサイズが計算用の基準サイズ(コンテンツCのサイズ=5)よりも大きくなる場合と、小さくなる場合について、レイアウト計算を行う必要がある。

【0190】

まず、コンテナCの計算結果のサイズが計算用の基準サイズよりも大きくなる場合($c > 5$)を想定する。また、各コンテナのテンションの計算式は、|レイアウト結果 - 計算用の基準サイズ|を用いる。なお、各コンテナのレイアウト結果の値を、それぞれa、b、cとする。さらに、図33では、3306が軸となっている。よって、図33のレイアウトの場合、本レイアウトエンジンは、軸3306を境にして左側にあるコンテナA及びコンテナCにて発生するテンションと右側にあるコンテナBにて発生するテンションが等しくなった時点の情報に基づいてレイアウト結果を算出する。以上を踏まえて計算式を作成すると、次のようになる。

10

20

30

40

50

$$|a - 15| + |c - 5| = |b - 25| \quad (\text{条件: } c > 5) \quad \dots (5)$$

【0191】

図33のレイアウトでは、コンテナAのレイアウト結果のサイズと1つの固定リンクの和が、コンテナCのレイアウト結果のサイズと2つの固定リンクのサイズと等しくなる。さらに、3301から3302の距離は固定である。以上を踏まえた結果、

$$a + 1 = 1 + c + 1 \quad \dots (6)$$

$$1 + a + 1 + b + 1 = 23 \quad \dots (7)$$

という制約条件が発生する。

【0192】

上記条件を踏まえて、(5)を計算処理すると、

$$15 - a + c - 5 = 25 - b$$

$$15 - a + (a - 1) - 5 = 25 - (-a + 20)$$

$$a = 4$$

となる。

【0193】

さらに、算出結果と式(6)、(7)を用いると、 $b = 16$ 、 $c = 3$ と算出される。しかし、式(5)の条件は、 $c > 5$ となっている。つまり、算出結果 $c (= 3)$ が式(5)の条件を満たしていないため、式(5)は破綻している。

【0194】

続いて、まず、コンテナCの計算結果のサイズが計算用の基準サイズよりも小さくなる場合を想定する。この場合も、軸3306を境にして左側と右側にて発生するテンションを等しくする必要があるので、式(5)を用いることが可能であるが、条件が異なる。その式を(8)に示す。

$$|a - 15| + |c - 5| = |b - 25| \quad (\text{条件: } c < 5) \quad \dots (8)$$

ここで式(6)、(7)を踏まえて、(8)を計算処理すると、

$$15 - a + 5 - c = 25 - b$$

$$15 - a + 5 - (a - 1) = 25 - (-a + 20)$$

$$a = 5.3$$

となる。

【0195】

さらに、算出結果と式(6)、(7)を用いると、 $b = 14.6 \dots$ 、 $c = 4.3 \dots$ と算出される。この場合、式(8)の条件($c < 5$)を満たしているため、式(8)により算出された値がレイアウト結果となる。なお、本願では、小数点第1位以下を切り捨てた値をレイアウト結果とするが、これに限られる必要はない。

【0196】

(2) 最小2乗法を用いた場合

続いて、最小2乗法を用いた場合について説明する。最小2乗法を用いた場合も図33を用いて説明するが、基本的な条件は同じである。まず、最小2乗法にてレイアウト計算を行うための計算式を作成する。

$$t = (a - 15)^2 + (b - 25)^2 + (c - 5)^2 \quad \dots (9)$$

さらに、式(6)、(7)を用いて(9)を変形すると、

$$t = (a - 15)^2 + \{(-a + 20) - 25\}^2 + \{(a - 1) - 5\}^2 \quad \dots (10)$$

となる。

【0197】

本レイアウトアルゴリズムでは、変数 t を最小にすることを目的としている。式(10)は、 a についての2次関数であり、傾きが正の2次関数における最小値は、頂点の値である。つまり、傾きが0の線が式(10)と接する値に該当する。よって、傾きを0として、式(10)を a にて微分すると、

$$0 = 2(a - 15) - 2(-a - 5) + 2(a - 6)$$

という計算式が成立する。この計算式を解くと $a = 5.3$ と算出される。さらに、算出結

10

20

30

40

50

果 a を式 (6)、(7) に代入すると $b = 4.3$ 、 $c = 14.7$ と算出される。

【 0 1 9 8 】

つまり、最小 2 乗法を用いて算出したレイアウト結果も最小 2 乗法を用いずに算出したレイアウト結果も同じとなる。しかし、最小 2 乗法を用いない場合、場合分けをして計算処理をするため、レイアウト計算に関する処理の負担が増加する。一方、最小 2 乗法を用いた場合は、場合分けの必要がないため容易にレイアウト結果を算出することが可能となる。このように、図 3 3 のような複雑なレイアウトについてレイアウト結果を算出する場合に、最小 2 乗法を用いる意義がある。また、図 3 3 では、軸 3 3 0 6 が 1 つであったため、2 つの場合分けでレイアウト結果を算出することができたが、軸が増加するに従い、場合分けの条件が増加するため、最小 2 乗法の有効性が増す。

10

【 0 1 9 9 】

また、コンテンツ優先レイアウトのアルゴリズムを用いることにより、出来るだけ各コンテナサイズをコンテンツサイズに近づくようにレイアウト処理を実行する。このため、コンテンツ優先レイアウトのアルゴリズムは、コンテンツの実サイズを用いてレイアウト処理を行いたい場合に効果を発揮する。

【 0 2 0 0 】

[第 2 レイアウトアルゴリズム (基本コンテナサイズとコンテンツサイズを考慮)]

続いて、図 3 4 を用いて、第 2 レイアウトアルゴリズムについて説明する。なお、初期条件は図 3 2 と同様である。3 4 0 1 ~ 3 4 0 3 は、デザイン時のコンテナである。3 4 0 5 ~ 3 4 0 7 は、各コンテナに挿入されるコンテンツである。3 4 0 4 は、各コンテナを関連付けるリンクであり、固定リンクとする。また、3 4 0 8 ~ 3 4 1 0 は、第 2 レイアウトアルゴリズムを図 3 4 に適用することによって算出されるレイアウト計算結果である。また、第 2 レイアウトアルゴリズムは、基本コンテナサイズとコンテンツサイズを考慮しているため、「コンテナ・コンテンツ優先レイアウトアルゴリズム」とする。

20

【 0 2 0 1 】

まず、第 2 レイアウトアルゴリズムにおける計算用の基準サイズは、各コンテナのサイズと各コンテナに入力されるコンテンツサイズの和を 2 で除算した値である。

よって、コンテナ A における計算用の基準サイズは「 1 6 」、コンテナ B における計算用の基準サイズは「 1 1 . 5 」、コンテナ C における計算用の基準サイズは「 2 3 . 5 」となる。

30

【 0 2 0 2 】

各コンテナのテンション T の計算方法は、

$$T = |\text{レイアウト結果のサイズ} - \{(\text{コンテンツサイズ} + \text{基本コンテナサイズ}) / 2\}| \dots (11)$$

である。

そして、(1 1) 式を用いて各コンテナにて算出された T の合計が現在対象としている集合の合計テンションとなる。ここで、図 3 4 を用いて説明する。

【 0 2 0 3 】

各コンテナの計算後のサイズを a、b、c とすると、各コンテナにて発生するテンションを適切に分配するためには、

40

$$t = (a - 16)^2 + (b - 11.5)^2 + (c - 23.5)^2 \dots (12)$$

を用いて計算処理することが望ましい。

図 3 4 のレイアウトは、単純なレイアウトであり、かつ、優先度が設定されていないため、変数 t が最小になるのは、

$$|a - 16| = |b - 11.5| = |c - 23.5| \dots (13)$$

と表すことができる。

さらに、計算後のレイアウトのサイズより、

$$a + 5 + b + 7 + c = 57 \dots (14)$$

と表すことができる。

【 0 2 0 4 】

50

(13) および (14) により、コンテンツおよび基本コンテナを考慮した場合における各コンテナの計算後のサイズが求められ、 $a = 14$ 、 $b = 9.5$ 、 $C = 16.5$ と算出される。よって、計算後の各コンテナ (3408 ~ 3410) A”、B”、C” の各サイズは図示したとおりになる。

【0205】

第1レイアウトアルゴリズムと同様に図34のようなレイアウトの場合、最小2乗法を用いなくてもレイアウト結果を算出することは可能である。しかし、図33のようなレイアウトに対して第2レイアウトアルゴリズムを適用する場合、やはり最小2乗法を用いることが望ましい。

【0206】

なお、最小2乗法を用いた計算方法は、計算用の基準サイズに{(コンテンツサイズ + 基本コンテナサイズ) / 2} の計算結果を代入して、コンテンツ優先レイアウトのアルゴリズムと同様に処理すればよいため、詳細な説明は省略する。

【0207】

この第2レイアウトアルゴリズムを用いることにより、基本コンテナサイズおよびコンテンツサイズの両者を考慮したレイアウト結果を算出できるため、テンプレートの作成者の意図およびコンテンツ作成者の意図を反映することができる。

【0208】

[第3レイアウトアルゴリズム(デザイン時のコンテナサイズを考慮)]

本願では、上述した第1、第2レイアウトアルゴリズムの他に第3レイアウトアルゴリズムがある。この第3レイアウトアルゴリズムは、第3レイアウトアルゴリズムが設定されたコンテナとリンクにて関連付けられているコンテナに第1または第2レイアウトアルゴリズムが設定されていた場合、各コンテナに流し込まれたコンテンツサイズがデザイン時の基本コンテナサイズに近づくようにレイアウト変更を実行する。そのため、第3レイアウトアルゴリズムは、基本デザイン優先レイアウトアルゴリズムとする。第3レイアウトアルゴリズムが、1つの集合に含まれる全てのコンテナに対して設定されている場合、各コンテナに流し込まれるコンテンツサイズは基本コンテナサイズに一致するようにサイズ変更が行われる。つまり、第1レイアウトアルゴリズムや第2レイアウトアルゴリズムのような計算処理を行わなくてもレイアウト結果を算出することは可能である。しかし、1つの集合の中に第3レイアウトアルゴリズムと他のレイアウトアルゴリズムが設定された場合には、計算処理が必要となる。この計算処理について図33を用いて説明する。

【0209】

図33の初期条件は上述しているので省略する。ここでは、コンテナA、Bに対して第1レイアウトアルゴリズムが設定されており、コンテナCに対して第3レイアウトアルゴリズムが設定されているとする。なお、最小2乗法を用いた場合の有効性は既に上述してある。よって、その有効性を生かすために、本説明でも最小2乗法を用いた場合に絞って説明する。

【0210】

コンテナA、Bの計算用の基準サイズは各コンテナに流し込まれるコンテンツサイズとなる(コンテナAの計算用の基準サイズは15、コンテナBの計算用の基準サイズは25)。一方、コンテナCには、第3レイアウトアルゴリズムが設定されているため、計算用の基準サイズは基本コンテナサイズとなる(コンテナCの計算用の基準サイズは9)。その結果、各コンテナのテンションがつり合うように分配するための変数 t を算出する式は次のように記載される。

$$t = (a - 15)^2 + (b - 25)^2 + (c - 9)^2 \quad \dots (15)$$

【0211】

式(15)に対して、式(6)、(7)を代入すると

$$t = (a - 15) + \{(-a + 20) - 25\}^2 + \{(a - 1) - 9\}^2 \quad \dots (16)$$

(16)を式(10)と同様の理由により、 a にて微分すると、

$$0 = 2(a - 15) - 2(-a - 5) + 2(a - 10)$$

10

20

30

40

50

a = 7 . 5 ... (2 6)

さらに、算出結果と (6)、(7)を用いることにより、b = 1 2 . 5、c = 6 . 5 と算出される。

【 0 2 1 2 】

よって、1つの集合において、例えばコンテンツ優先レイアウトのアルゴリズムと基本デザイン優先レイアウトのアルゴリズムを適用することで、コンテナ毎にコンテンツサイズまたは基本コンテナサイズに近づくようにレイアウト結果を算出することができる。つまり、コンテナ毎にテンプレート作成者またはコンテンツサイズ作成者の意図を反映したレイアウト結果を算出することが可能となる。

【 0 2 1 3 】

このように基本デザイン優先レイアウトのアルゴリズムは1つの集合において他のレイアウトアルゴリズムと組み合わせて用いることにより、その効果を発揮する。

【 0 2 1 4 】

[U I 設定例 (その 1)]

図 2 5 の (A) ~ (G) は、上述したような各種のレイアウト調整アルゴリズムを設定するためのユーザインターフェース (U I) 例を示す図である。

【 0 2 1 5 】

まず、ユーザの所定の操作により、レイアウト調整アルゴリズムを適用する範囲を指定するためのダイアログ 2 5 0 0 が表示される。ダイアログ 2 5 0 0 において選択されたものにより、ダイアログ 2 5 0 1 ~ 2 5 4 1 の対応するダイアログが表示されることとなる。

【 0 2 1 6 】

図 2 5 の (A) では、図 2 5 の (F) のダイアログ 2 5 0 0 において「ドキュメント」が選択された場合、ドキュメント毎に適用するレイアウト調整アルゴリズムを選択するためにダイアログ 2 5 0 1 が表示される。ユーザはダイアログ 2 5 0 1 を用いてドキュメント内のコンテナに対するレイアウト調整アルゴリズムを選択することができる。ダイアログ 2 5 0 1 にはコンテンツ優先のレイアウト 2 5 0 2、コンテナ・コンテンツ優先のレイアウト 2 5 0 3、基本デザイン優先 2 5 0 4 のオプションが存在し、ユーザはいずれかのオプションを選択する。ダイアログ 2 5 0 1 には、図 2 5 の (B) ~ (E) に示される「指定なし」のオプションが存在しないが、これはドキュメントのデフォルト値として使用するべきレイアウト調整アルゴリズムを決定しておく必要があるためである。ただしこれは以下に説明するように、適用ポリシーによって決まることであり、ここでは便宜上指定なしのオプションを選択肢から外している。ダイアログ 2 5 0 1 において選択されたレイアウト調整アルゴリズムはドキュメント全体に影響するものであるが、実際のレイアウト計算時にドキュメント内の全てのコンテナに適用されるかは、適用ポリシーによって決まる。適用ポリシーについては後述の「適用ポリシー順序」において詳しく説明する。

【 0 2 1 7 】

図 2 5 の (B) では、ダイアログ 2 5 0 0 において「ページ」が選択された場合、アルゴリズムを適用すべきページやページの範囲を指定するためのページ設定画面を表示する。そのページ設定画面にて、ページやページ範囲を設定すると、ページ毎に適用するレイアウト調整アルゴリズムを選択するためにダイアログ 2 5 1 1 が表示される。

【 0 2 1 8 】

ユーザはダイアログ 2 5 1 1 を用いてページ内のコンテナに対するレイアウト調整アルゴリズムの選択が可能である。

ダイアログ 2 5 1 1 には指定なし 2 5 1 2、コンテンツ優先のレイアウト 2 5 1 3、コンテナ・コンテンツ優先のレイアウト 2 5 1 4、基本デザイン優先 2 5 1 5 のオプションが存在し、ユーザはいずれかのオプションを選択することができる。

ダイアログ 2 5 1 1 で選択されたレイアウト調整アルゴリズムは、当該設定ダイアログを開いたときに指定されていたページに対して影響する。図 2 5 の (B) では指定なしが選択されている。

10

20

30

40

50

【0219】

なお、実際のレイアウト計算時にページ内の全てのコンテナに当該ダイアログ2511の設定が適用されるかは、後述の適用ポリシーによって決まる。

【0220】

図25の(C)では、ダイアログ2500において「エリア」が選択された場合、エリア毎に適用するレイアウト調整アルゴリズムを選択するためにダイアログ2521が表示される。なお、エリアに対してレイアウト調整アルゴリズムを選択する場合には、ダイアログ2500にて、エリアを選択した後に表示されるドキュメントテンプレート上でエリアの指定を行わなければならない。

【0221】

ユーザはダイアログ2521を用いてエリア内のコンテナに対するレイアウト調整アルゴリズムを選択することができる。ダイアログ2521には指定なし2522、コンテンツ優先のレイアウト2523、コンテナ・コンテンツ優先のレイアウト2524、基本デザイン優先2525のオプションが存在し、ユーザはいずれかのオプションを選択することができる。

【0222】

ダイアログ2521で選択されたレイアウト調整アルゴリズムは、当該設定ダイアログを開いたときに指定されていた領域内のコンテナに影響するものである。図25の(C)では指定なしが選択されている。なお、実際のレイアウト計算時にエリア内の全てのコンテナに当該ダイアログ2521の設定が適用されるかは、後述の適用ポリシーによって決まる。

【0223】

図25の(D)では、ダイアログ2500において「レコード」が選択された場合、レコードを選択するためのレコード選択画面が表示される。そのレコード選択画面にて指定されたレコード毎に適用するレイアウト調整アルゴリズムを選択するためにダイアログ2531が表示される。ユーザはダイアログ2531を用いてレコード内のコンテナに対するレイアウト調整アルゴリズムを選択することができる。ダイアログ2531には指定なし2532、コンテンツ優先のレイアウト2533、コンテナ・コンテンツ優先のレイアウト2534、基本デザイン優先2535のオプションが存在し、ユーザはいずれかのオプションを選択できる。ダイアログ2531で選択されたレイアウト調整アルゴリズムは当該設定ダイアログを開いたときに指定されていたレコード内の全コンテナに影響するものである。

【0224】

図25の(D)では指定なしが選択されている。なお、実際のレイアウト計算時にレコードに関係する全てのコンテナに適用されるかは、適用ポリシーによって決まる。

【0225】

図25の(E)では、ダイアログ2500において「コンテナ」が選択された場合、コンテナを選択するためのコンテナ選択画面が表示される。そのコンテナ選択画面にて選択されたコンテナ毎に適用するレイアウト調整アルゴリズムを選択するためにダイアログ2541が表示される。ユーザはダイアログ2541を用いて任意のコンテナに対するレイアウト調整アルゴリズムを選択することができる。ダイアログ2541には指定なし2542、コンテンツ優先のレイアウト2543、コンテナ・コンテンツ優先のレイアウト2544、基本デザイン優先2545のオプションが存在し、ユーザはいずれかのオプションを選択できる。ここで選択されたレイアウト調整アルゴリズムは設定ダイアログを開いたときに指定されていたコンテナに影響するものである。

【0226】

図25の(E)では指定なしが選択されている。実際のレイアウト計算時において、指定されたレイアウト調整アルゴリズムが当該コンテナに適用されるかは、後述の適用ポリシーによって決まる。なお、ダイアログ2500において「コンテナ群」が選択された場合、コンテナ群を選択するためのコンテナ群選択画面が表示される。そのコンテナ群選択

10

20

30

40

50

画面にて選択されたコンテナ群毎に適用するレイアウト調整アルゴリズムを選択するために、図 25 の (G) のダイアログ 2551 が表示される。なお、ダイアログ 2551 についての選択処理等は上記と同等であるため詳細な説明は省略する。また、コンテナ群とは、本願におけるリンクにて関連付けられた集合を指す。

【 0 2 2 7 】

以上のように、図 25 の (A) ~ (E) (G) (*1) で指定されたレイアウト調整アルゴリズムは、ドキュメント、ページ、エリア、レコード、コンテナ等の各範囲に設定されたレイアウト調整アルゴリズムを示すレイアウト調整アルゴリズム情報としてメモリ 136、格納デバイス 139 等により保持される。そのため、レイアウト編集アプリケーションはメモリ 136 または格納デバイス 139 を参照することにより図 19 のフローチャートにおける判断処理を実行することができる。

10

【 0 2 2 8 】

以上のような図 25 の (A) ~ (G) (*1) の UI によってレイアウト調整アルゴリズムを設定すると、図 30 に示すような階層構造で設定されたレイアウト調整アルゴリズムがメモリ 136 または格納デバイス 139 に格納されることになる。

【 0 2 2 9 】

[適用ポリシー]

適用ポリシーは適用レイアウト調整アルゴリズムのパスフロー (図 19) で説明したように、ドキュメントの設定、ページの設定、レコードの設定、エリアの設定、コンテナの設定の順番で適用されていく。つまり、レイアウト調整アルゴリズムを適用する単位の小さいもの程、優先順位が高く設定されており、優先してレイアウト計算に使用される。また、図 30 の階層構造でいえば、階層の低い程、設定内容の優先順位が高く、優先してレイアウト計算に使用されることになる。

20

【 0 2 3 0 】

図 26 ではあるドキュメントにおいて、各単位で設定されたレイアウト調整アルゴリズムにより、最終的に各コンテナの設定がどのようなになるのかを示している。

この設定内容は図 30 の階層構造から得られるものである。図 30 に示すように、このドキュメントには「コンテンツ優先」が設定されている。よってこのドキュメント内の全てのコンテナは「コンテンツ優先」のレイアウト調整アルゴリズムがデフォルトとして設定される。

30

【 0 2 3 1 】

次にページ単位では、図 30 を参照することにより第 1 ページ 2611 に「コンテナ・コンテンツ優先」が、第 2 ページ 2621 と第 3 ページ 2631 には「コンテンツ優先」が設定されていることが分かる。第 1 ページ 2611 にはこれ以上に細かい単位でのレイアウト計算設定がないので、第 1 ページ内の全てのコンテナ (コンテナ 2612、コンテナ 2613、コンテナ 2614) の適用レイアウト調整アルゴリズムは「コンテンツ優先」となる。

【 0 2 3 2 】

第 2 ページ 2621 内では、複数のレコードについてレイアウト調整アルゴリズムの設定がなされている。なお、第 2 ページでは 1 つのページ内に複数のレコードのコンテンツデータを流し込むマルチレコード機能が採用されている。つまり、第 2 ページにおけるリンクセット 2622 には 1 レコード目、リンクセット 2623 には 2 レコード目、リンクセット 2624 には 3 レコード目のコンテンツデータが流し込まれることとなる。

40

【 0 2 3 3 】

そして、第 1 レコードおよび第 3 レコードが流し込まれるリンクセット 2622 および 2624 には「コンテンツ優先」が、リンクセット 2623 には「コンテナ・コンテンツ優先」が設定されている。

【 0 2 3 4 】

第 2 ページ 2621 では、レコード単位よりも細かい単位での設定が存在しない。従って、リンクセット 2623 内の全コンテナに「コンテナ・コンテンツ優先」が適用される

50

こととなるが、リンクセット 2 6 2 2 , 2 6 2 4 内の全コンテナには「コンテンツ優先」が適用される。

【 0 2 3 5 】

次に第 3 ページ 2 6 3 1 にはユーザが指定した選択エリア 2 6 3 7 が存在し、その選択エリア 2 6 3 7 には「基本デザイン優先」が指定されている。さらに、選択エリア内のコンテナ 2 6 3 5 には直接、選択エリア 2 6 3 7 とは別のレイアウト調整アルゴリズム (「コンテンツ優先」) が指定されている。よって、選択エリア 2 6 3 7 内のコンテナ 2 6 3 4、コンテナ 2 6 3 6 には「基本デザイン優先」が、コンテナ 2 6 3 5 には「コンテンツ優先」が適用されることになる。又、第 3 ページ 2 6 3 1 に存在する選択エリア 2 6 3 7 外のコンテナ 2 6 3 2、コンテナ 2 6 3 3 には、当該ページに設定された「コンテンツ優先 (2 6 3 9)」が適用されることになる。

【 0 2 3 6 】

以上の説明からも分かるように、図 3 0 において下流において設定されたレイアウト調整アルゴリズムが優先的に適用されることとなる。

【 0 2 3 7 】

[UI 設定例 (その 2)]

図 2 7 はレイアウト調整アルゴリズムを設定するための別のユーザインターフェース例を示す図である。例えば、図 3 0 に示すようなアルゴリズム情報に基づいて各コンテナに設定されているレイアウト調整アルゴリズムが取得され、図 2 7 に示すように複数種類の網掛けを用いることにより各コンテナに設定されているレイアウト調整アルゴリズムが明示される。このような UI によれば、各コンテナに設定されたレイアウト調整アルゴリズムが一目瞭然であり、また、各コンテナへのレイアウト調整アルゴリズムの指定を容易に行える。

【 0 2 3 8 】

タイトルバー 2 8 0 2 とツールバー 2 8 0 3 を持つ典型的なウィンドウ 2 8 0 1 において、メインウィンドウ 2 8 0 4 にはあるページ 2 8 0 5 とそこにレイアウトされたコンテナが表示されている。またページ内のマージンエリア 2 8 0 6 の線も表示されている。そしてパレット 2 8 1 1 が存在し、パレット 2 8 1 1 を用いてメインウィンドウ上のコンテナのレイアウト調整アルゴリズムを指定することが出来る。例えば、コンテンツ優先のレイアウト調整アルゴリズムをコンテナ B に適用したい場合は、マーク 2 8 1 2 をコンテナ B 内へドラッグアンドドロップする操作を行う。同様に、コンテナ・コンテンツ優先のレイアウト調整アルゴリズムの場合はマーク 2 8 1 4 を、基本デザイン優先のレイアウト調整アルゴリズムの場合はマーク 2 8 1 8 を用いる。各コンテナの表示では、設定されているレイアウト調整アルゴリズムの種類に応じてコンテナの網掛けの種類を異ならせているので、各コンテナがどのレイアウト調整アルゴリズムを使用するかを一目で判断できる。

【 0 2 3 9 】

もちろん、エリアを指定してレイアウト調整アルゴリズムを設定することもできる。例えば、コンテナ C, D, E のみが内包される矩形をマウス操作により UI 上で描画することで、コンテナ C, D, E が選択状態となる。そして、選択状態となった複数のコンテナのいずれかへ所望のレイアウト調整アルゴリズムに対応するマーク (2 8 1 2, 2 8 1 4, 2 8 1 8) をドラッグアンドドロップする。このような操作により選択エリア内の全コンテナに一括してレイアウト調整アルゴリズムを設定できる。なお、エリアの指定により図 3 0 の階層構造におけるエリアの部分が、変更後の構造に応じて変更されることになる。また、あるエリアに属するコンテナについてレイアウト調整アルゴリズムの指定を行ったり、あるエリアに属するコンテナを別のエリアに属するようにエリア指定を行うと、当該コンテナが属していたエリアのくくりは解除されるものとする。

【 0 2 4 0 】

また、図 2 7 の UI において、コンテンツ優先のレイアウト調整アルゴリズムが指定されたコンテナはリストボックス 2 8 1 3 内にリスト表示される。また、コンテナ・コンテンツ優先のレイアウト調整アルゴリズムが指定されたコンテナはリストボックス 2 8 1 7

内にリスト表示され、基本デザイン優先のレイアウト調整アルゴリズムが指定されたコンテナはリストボックス 2819 内にリスト表示されている。また、コンテナ・コンテンツ優先のレイアウト調整アルゴリズムにはデザインパワー 2815 とコンテンツパワー 2816 を入力するエディットボックスが存在するが、これについては後述の「アルゴリズム優先度」にて詳しく説明する。ユーザはこのような画面を見ながら、各コンテナに対してレイアウト調整アルゴリズムを簡便に決めていくことが出来る。

【0241】

図 28 は図 27 の UI を表示するための処理を示すフローチャートである。まずステップ S2701 においてレイアウト編集アプリケーション 121 は、ユーザからのレイアウト調整アルゴリズム設定画面の表示指示を認識して、図 27 に示すようなレイアウト調整アルゴリズム設定画面を表示する。 10

【0242】

次に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ているレイアウト調整アルゴリズムを検索する (S2702)。詳細には、図 30 のような階層情報がメモリ 136 等に保持されているため、レイアウト編集アプリケーション 121 がメモリ 136 を参照することにより S2702 の処理を実現することが可能となる。

【0243】

ステップ S2703 では、レイアウト編集アプリケーション 121 は、各コンテナについて設定されているレイアウト調整アルゴリズムに基づいて、各コンテナの網掛け、各リスト 2813, 2817, 2819 への表示を行う。こうして図 27 に示すようなインターフェース表示が完成する。また、図 27 の UI を用いて各コンテナ毎に設定されたレイアウト調整アルゴリズムの情報は、アルゴリズム情報としてメモリ 136、格納デバイス 139 等により保持される。 20

【0244】

[アルゴリズム優先度]

コンテナ・コンテンツ優先のレイアウト調整アルゴリズムは、現在のコンテナサイズとコンテンツサイズとの差に加え、現在のコンテナサイズとデザイン時のコンテナサイズ (デザインサイズ) からのテンションを小さくしていくことでレイアウト結果を導き出す。よってそれぞれのテンションに重み値を設定することで、コンテンツサイズとデザインサイズのどちらにより近くレイアウトするかを決めることが出来る。 30

【0245】

図 29 の (A) ~ (D) ではコンテナ 2901 およびコンテナ 2902 に対してコンテナ・コンテンツ優先のレイアウト調整アルゴリズムが適用される場合において、優先度が設定されることによりどのようにレイアウト計算されるのかについて説明している。

【0246】

図 29 の (A) に示されるように、コンテナ 2901 とコンテナ 2902 が存在し、それらはリンク 2903 によって関連付けされている。それぞれのコンテナにはデータベースから挿入されるデータ (コンテンツ) が存在し、コンテナ 2901 にはコンテンツ 2904、コンテナ 2902 にはコンテンツ 2905 が挿入されている。

【0247】

現在の各コンテナの幅サイズはコンテナ 2901 が $|a - b| = 7$ 、コンテナ 2902 が $|e - h| = 6$ 、各コンテンツサイズの幅サイズはコンテンツ 2904 が $|a - d| = 8$ 、コンテンツ 2905 が $|g - h| = 5$ である。また、デザイン時のコンテナサイズ 2906 と 2907 が存在する。 40

【0248】

コンテナ 2901 および 2902 には「コンテナ・コンテンツ優先」が設定されている。そのため、レイアウト計算では現在のコンテナサイズ (コンテナ 2901) とデザインサイズ (デザインサイズ 2906) との差 (テンション T_1) と、現在のコンテナサイズ (コンテナ 2901) とコンテンツサイズ (コンテンツ 2904) との差 (テンション T_3) が使用される。 50

【0249】

またコンテナ2902には「コンテナ・コンテンツ優先」が設定されているため、現在のコンテナサイズ（コンテナ2902）とコンテンツサイズ（コンテンツ2905）との差（テンション T_2 ）と、現在のコンテナサイズ（2902）とコンテンツサイズ（コンテンツ2905）との差（テンション T_4 ）が使用される。これらのテンションの合計を小さくさせ、現在のコンテナをレイアウト調整させることによって、計算用の基準サイズに近づけようとする。

【0250】

図29の（B）はテンションの計算を行う状態を図示しているものである。コンテナ2901のテンションには、現在のコンテナサイズ（7）とコンテンツサイズ（8）とのテンション $T_1 = |b - d| = 1$ と、現在のコンテナサイズ（7）とデザインサイズ（5）とのテンション $T_3 = |b - c| = 2$ が存在する。

【0251】

また、コンテナ2902には、現在のコンテナサイズ（6）とコンテンツサイズ（5）とのテンション $T_2 = |e - g| = 1$ と、デザインサイズ（8）と現在のコンテナサイズ（6）のテンション $T_4 = |f - e| = 2$ が存在する。

【0252】

ここで、図29の（D）に示すように、プロパティ2931とプロパティ2934において、それぞれコンテナA、コンテナBに設定しているコンテナ・コンテンツ優先のレイアウト調整アルゴリズムにおける、デザインパワーとコンテンツパワーを設定できる。ここで言うパワーとは重み値と同等の意味であり、あるサイズに近づこうとする力である。例えばデザインサイズに近づこうとする力、あるいはコンテンツサイズに近づこうとする力を指す。ここでは通常のパワー（重み値）を100%としてあり、それよりも値が大きければパワーが強くなる、つまりそのサイズに近づこうとする力が増す。逆にそれよりも小さい場合パワーは弱くなる、つまりそのサイズに近づこうとする力が弱まることになる。これをレイアウト計算の中ではテンションに相乗することで実現している。つまり、このパワーの値（重み値）をベースにテンションの比重を変化させて、コンテナのレイアウト調整を実行する。

【0253】

各プロパティ内では、コンテナAではデザインパワー2932は100%、コンテンツパワー2933は50%、コンテナBでも同様にデザインパワー2935は100%、コンテンツパワー2936は50%が設定されている。つまり、図29の（A）～（D）のリンクセットにおいては、コンテナサイズを基本レイアウト作成時のデザインサイズに近づこうとする力が、コンテンツサイズに近づけようとする力よりも2倍強いこととなる。

【0254】

よってこのリンクセットのテンションの合計は、

$$\begin{aligned} & 1/2 \times (T_1 + T_2) + T_3 + T_4 \\ &= 1/2 \times (|b - d| + |e - g|) + |b - c| + |f - e| \\ &= 1/2 \times 2 + 2 + 2 \\ &= 5 \end{aligned}$$

である。

【0255】

図29の（C）は、このようにして計算されるテンションを減らすようにマークの移動を行った結果を示している。図29の（C）では、コンテナ2901のサイズは $|a - b| = 6$ 、コンテナ2902のサイズは $|e - h| = 7$ となっている。よってレイアウト結果時のテンションは、

$$\begin{aligned} & 1/2 \times (T_1 + T_2) + T_3 + T_4 \\ &= 1/2 \times (|b - d| + |e - g|) + |b - c| + |f - e| \\ &= 1/2 \times 4 + 1 + 1 \\ &= 4 \end{aligned}$$

となる。この場合、テンション 0 の状態を達成できていないが、これ以上、テンションが減らなかったため、レイアウト計算を終了する。

【 0 2 5 6 】

以上のように、上記の例では、コンテナ A、コンテナ B においてコンテンツサイズよりもデザインサイズにやや近い形でレイアウトされた。このようにレイアウト調整アルゴリズムに優先度（パワーまたは重み値ともいう）が適用できることで、ユーザの求める多様なレイアウトのニーズに応えることができる。

【 0 2 5 7 】

以上説明したように、上記実施形態によれば、自動レイアウトシステムにおいて、各コンテナに様々なレイアウト計算方法のレイアウト調整アルゴリズムの設定が可能になる。このため、複数のコンテナが関連付けられており、相互に影響を与えながらレイアウトを変化させるような場合においても、それぞれのコンテナに対してユーザが所望とするレイアウト調整アルゴリズムによるサイズ計算を行うことにより、ユーザが意図した様々なレイアウトを作成することが可能になる。さらに、設定したレイアウト調整アルゴリズムを視覚的に示すことが可能になるため、ユーザは自動レイアウトした後のレイアウト結果を予測しやすくなる。

【 0 2 5 8 】

また、上記実施形態において、複数種類の計算方法アルゴリズムのそれぞれは、データ領域と目標形状との差異が小さくなるようにデータ領域をレイアウト調整するものであり、差異の算出過程が各レイアウト調整アルゴリズムで異なるものである。これにより、柔軟にユーザの意図を反映したレイアウト調整を可能としている。また、関連付けられた複数のコンテナについてレイアウト調整を行う場合にも、各レイアウト調整アルゴリズムを適用して得られた差異（テンション）の合計を小さくするようにコンテナをレイアウト調整すればよい。そのため、複数コンテナ間のネゴシエーション処理への適用が容易である。

【 0 2 5 9 】

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 CPU や MPU）が記憶媒体に格納されたプログラムコードを読み出し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 6 0 】

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 6 1 】

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。

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】

10

20

30

40

50

【 0 2 6 2 】

【図 1 A】実施形態によるバリアブルプリントシステムの構成を示す図である。

【図 1 B】図 1 A におけるホストコンピュータの構成の概略を示す図である。

【図 1 C】バリアブルデータプリントの概要を説明する図である。

【図 2】実施形態によるバリアブルプリントシステムの他の構成を示す図である。

【図 3】メニューバー、ツールバー、ワークエリア、フローティングパレットを含む、実施形態によるユーザインターフェースの例を示す図である。

【図 4】ユーザインター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】実施形態によるコンテナルールを説明する図である。

【図 6】実施形態によるリンクの作成処理を説明するフローチャートである。

10

【図 7】リンク作成時のユーザインターフェースの遷移例を示す図である。

【図 8】実施形態のレイアウトエンジンによるレイアウト計算処理を示すフローチャートである。

【図 9】実施形態のレイアウトエンジンによるレイアウト計算処理を示すフローチャートである。

【図 1 0】実施形態によるレイアウト計算処理を説明するための図である。

【図 1 1】実施形態のレイアウト計算時における、コンテナの集合について説明した図である。

【図 1 2】可変リンクによってコンテナを配置したときのユーザインターフェースにおける表示例を示す図である。

20

【図 1 3】実施形態のユーザインターフェースにより可変リンクを設定するための処理を説明するフローチャートである。

【図 1 4】実施形態によるリンクの設定ダイアログウインドウを示す図である。

【図 1 5】固定リンクによるレイアウト結果例を示す図である。

【図 1 6】可変リンクによるレイアウト結果例を示す図である。

【図 1 7】本実施形態による自動レイアウト処理の全体的な処理の流れを示すフローチャートである。

【図 1 8】本実施形態によるレイアウト最適化処理を説明するフローチャートである。

【図 1 9】ステップ S 1 8 0 1 で選択されたコンテナに対して、ステップ S 1 8 0 2 で行われる適用アルゴリズムのパスを示すフローチャートである。

30

【図 2 0】本実施形態によるレイアウト再計算処理を示すフローチャートである。

【図 2 1】コンテンツ優先のレイアウト調整アルゴリズムを説明する図である。

【図 2 2】コンテナ・コンテンツ優先のレイアウト調整アルゴリズムを説明する図である。

【図 2 3】基本デザイン優先のレイアウト調整アルゴリズムを説明する図である。

【図 2 4】基本デザイン優先のレイアウト調整アルゴリズムとコンテンツ優先のレイアウト調整アルゴリズムが 1 つのリンクセット内で別々のコンテナに設定されている場合のレイアウト計算処理について説明する図である。

【図 2 5】各種のレイアウト調整アルゴリズムを設定するためのユーザインターフェース (UI) 例を示す図である。

40

【図 2 6】各単位で設定されたレイアウト調整アルゴリズムにより各コンテナの設定がどのようなになるのかを説明する図である。

【図 2 7】レイアウト調整アルゴリズムを設定するための別のユーザインターフェース例を示す図である。

【図 2 8】図 2 7 の UI を表示するための処理を示すフローチャートである。

【図 2 9】コンテナ・コンテンツ優先とコンテンツ優先が 1 つのリンクセット内で別々のコンテナに設定されている場合のレイアウト計算例を示す図である。

【図 3 0】アルゴリズム情報の格納例を示す図である。

【図 3 1】レイアウト最適化処理を説明するための図である。

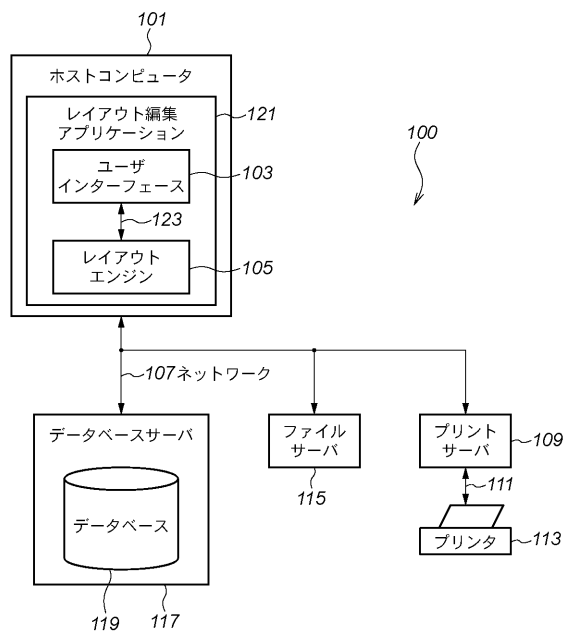
【図 3 2】コンテンツ優先レイアウトのアルゴリズムを説明する図である。

50

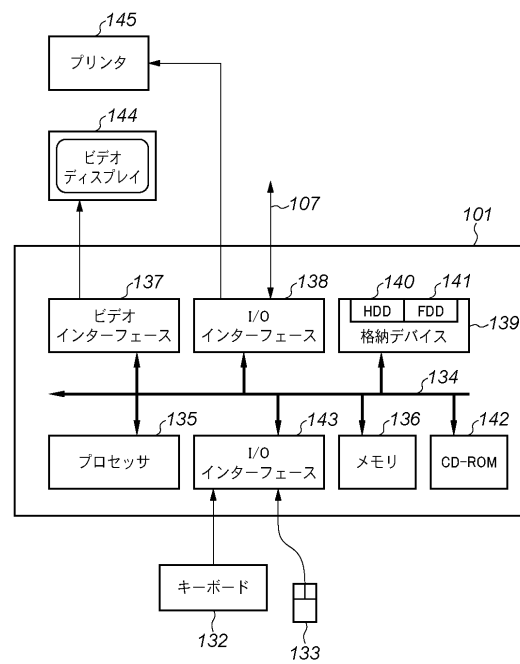
【図 3 3】最小 2 乗法を用いたレイアウト算出方法および結果と最小 2 乗法を用いない場合のレイアウト算出方法および結果を説明するためのレイアウト例を示す図である。

【図 3 4】コンテナ・コンテンツ優先レイアウトアルゴリズムを説明する図である。

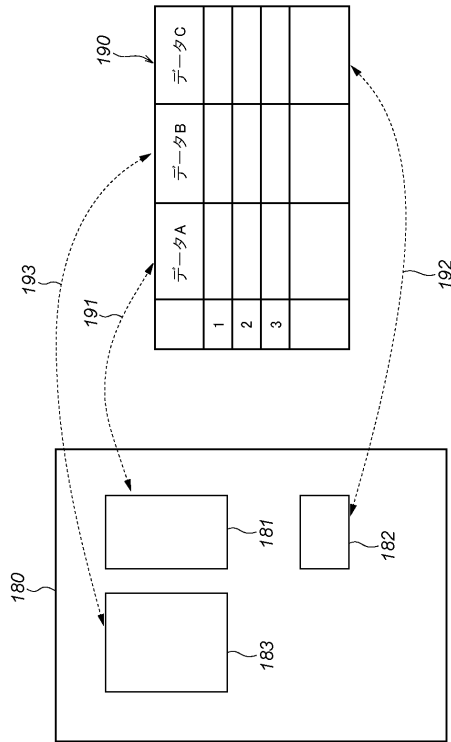
【図 1 A】



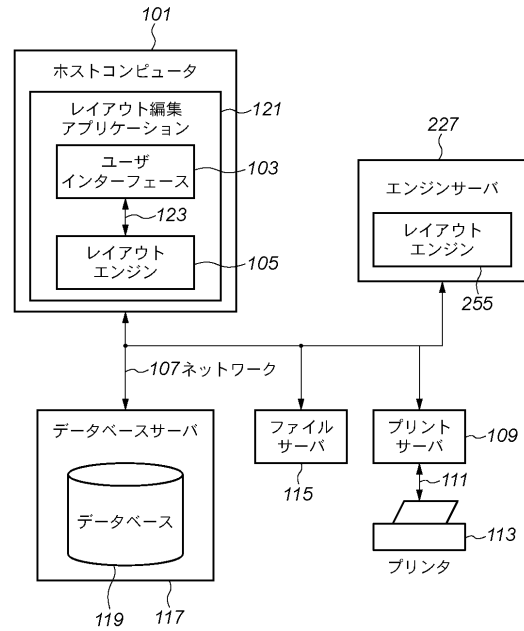
【図 1 B】



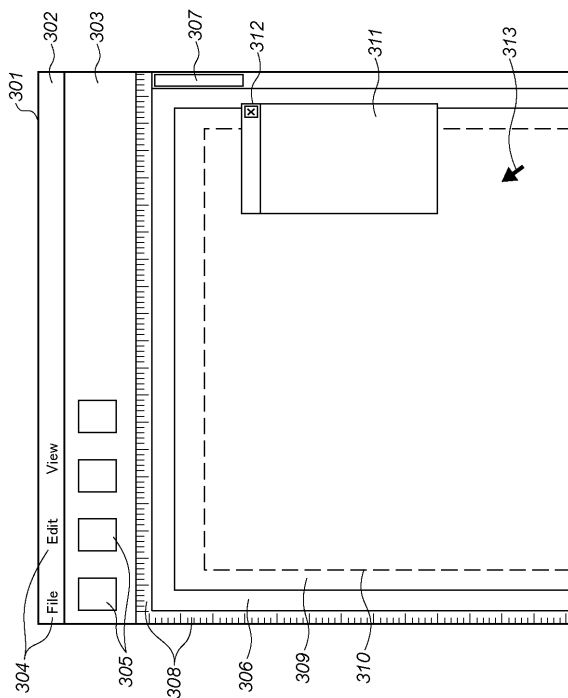
【図 1 C】



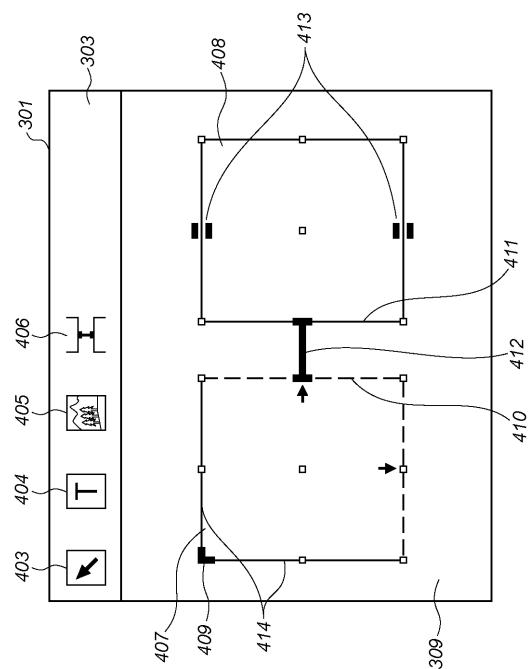
【図 2】



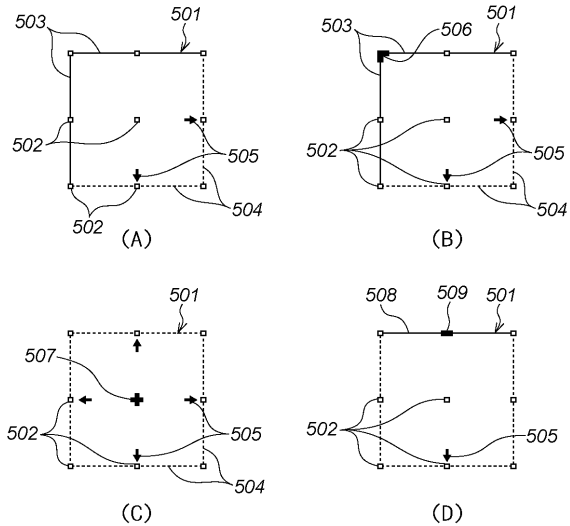
【図 3】



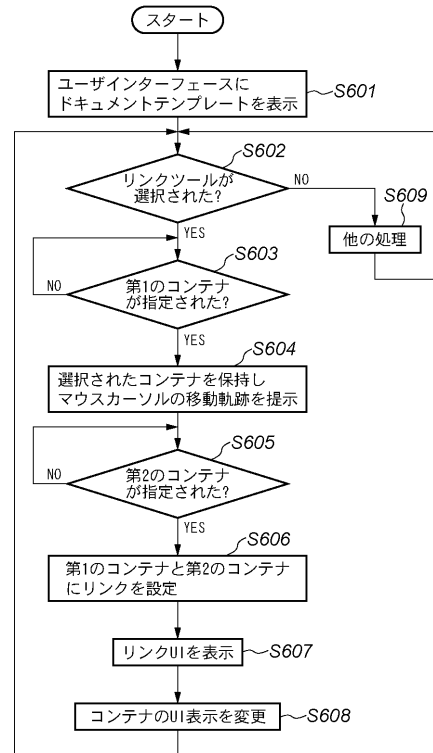
【図 4】



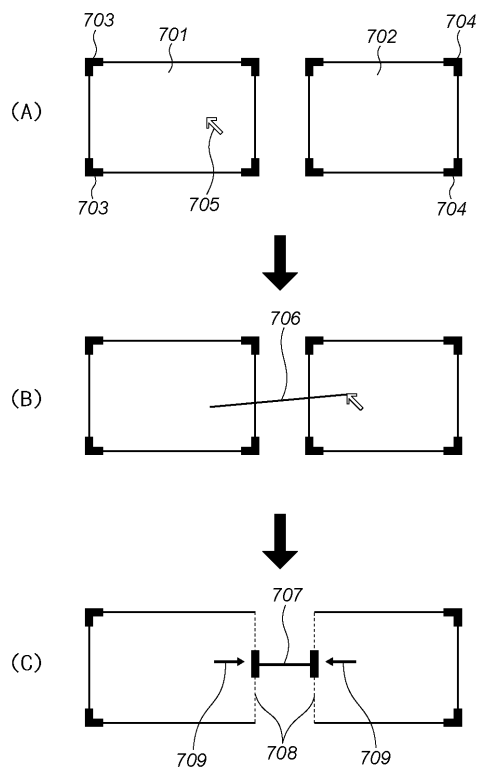
【 図 5 】



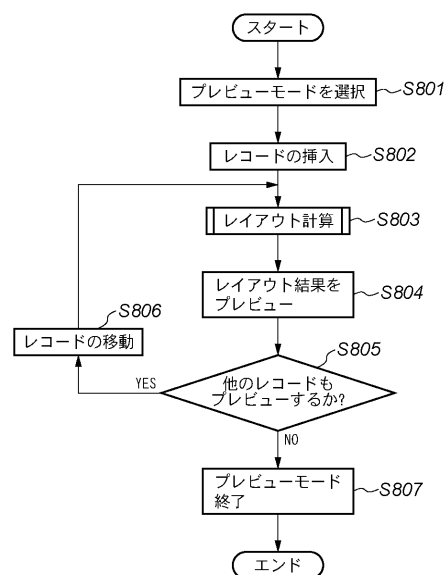
【 図 6 】



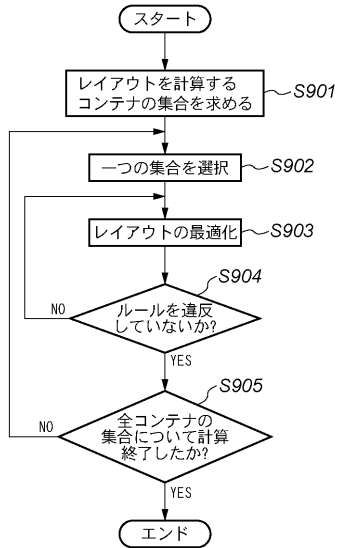
【 図 7 】



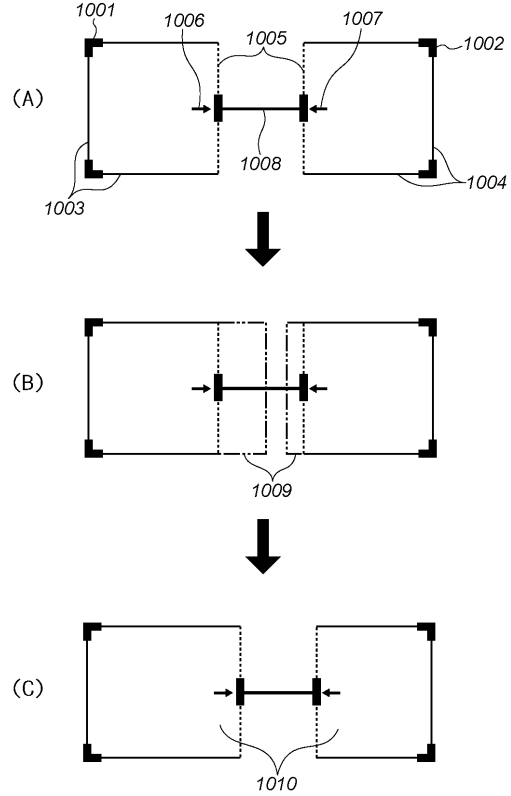
【 図 8 】



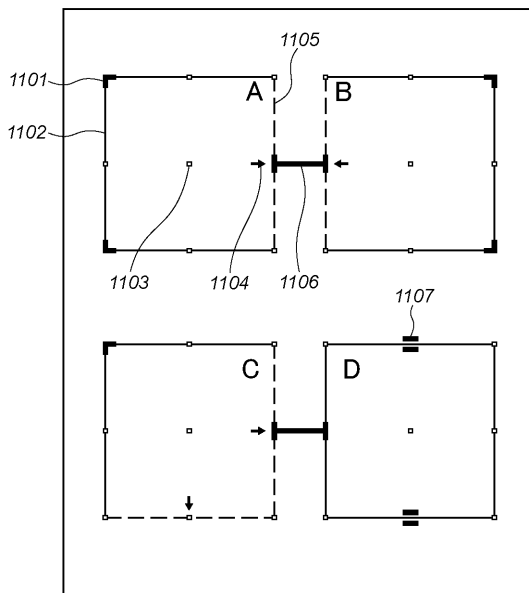
【図 9】



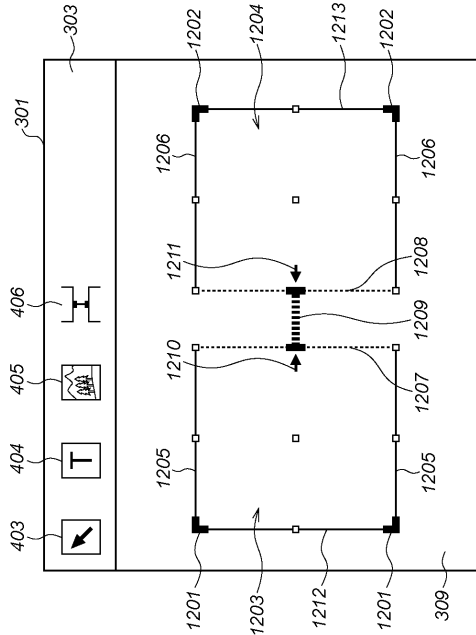
【図 10】



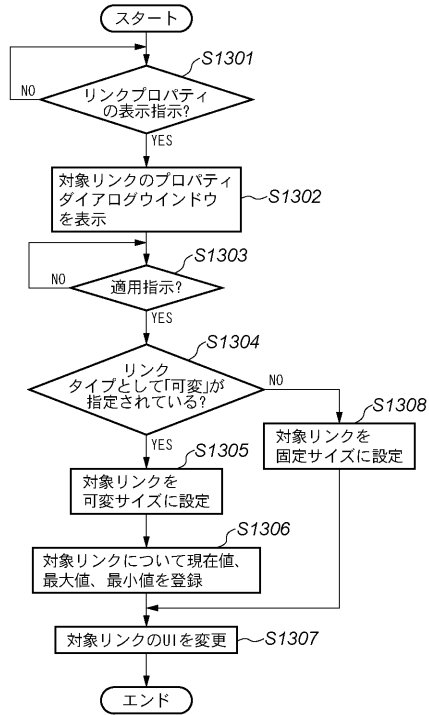
【図 11】



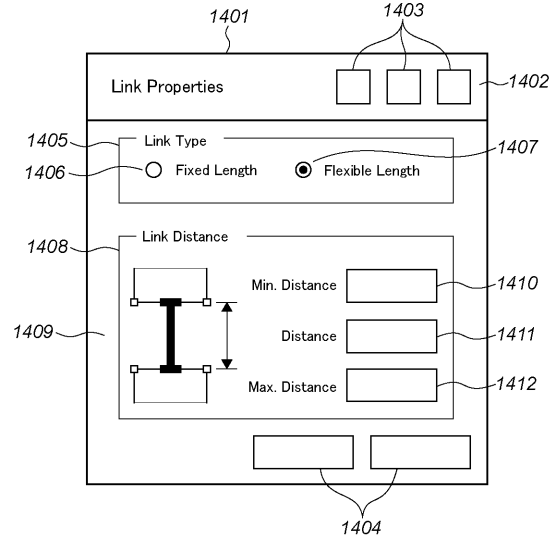
【図 12】



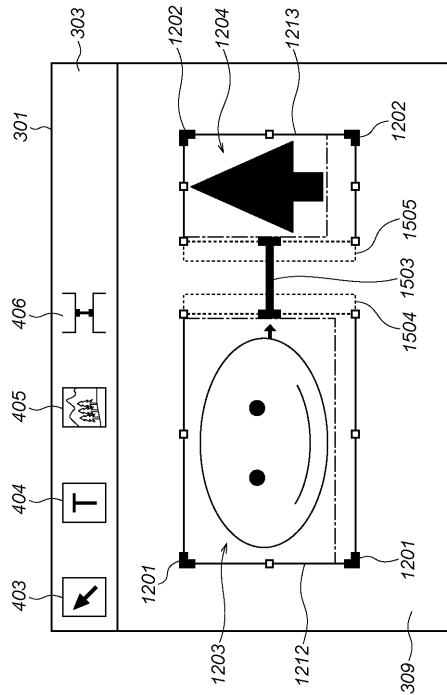
【図 13】



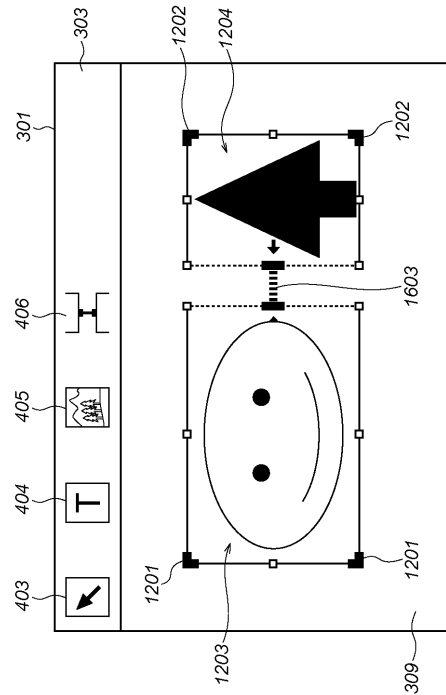
【図 14】



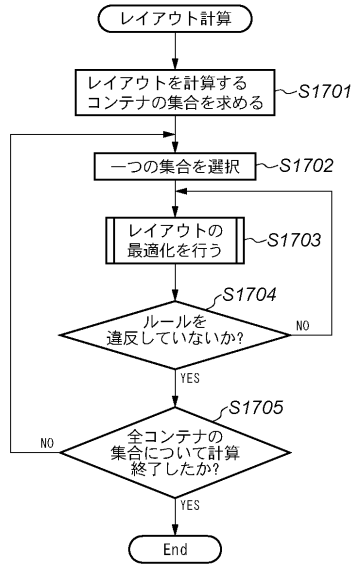
【図 15】



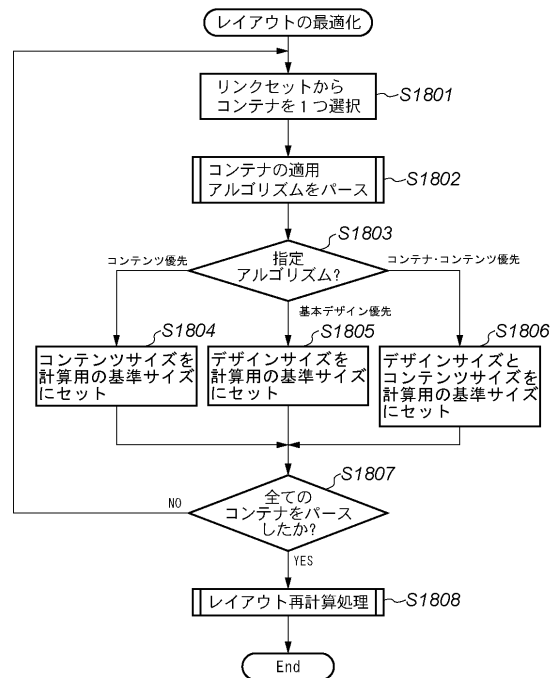
【図 16】



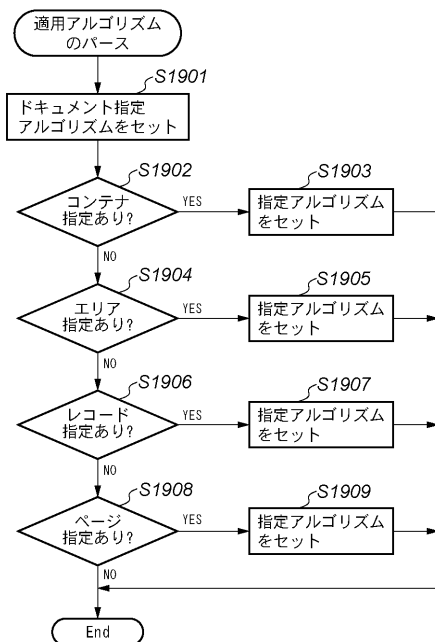
【図 17】



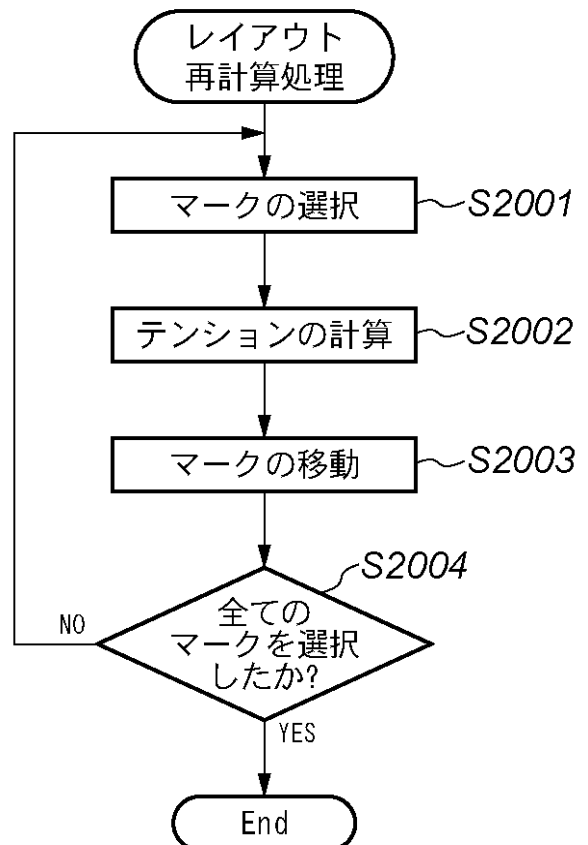
【図 18】



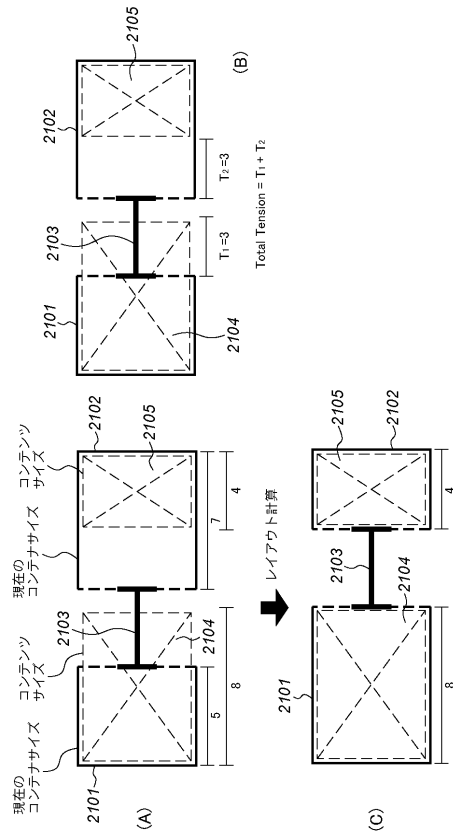
【図 19】



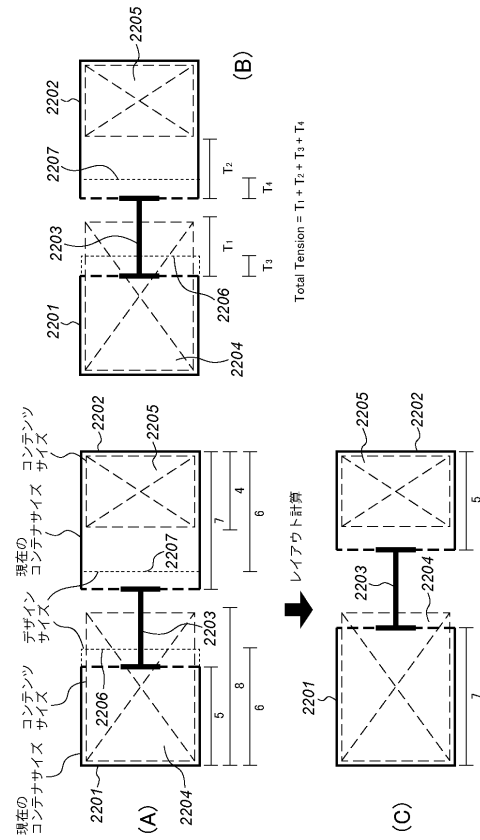
【図 20】



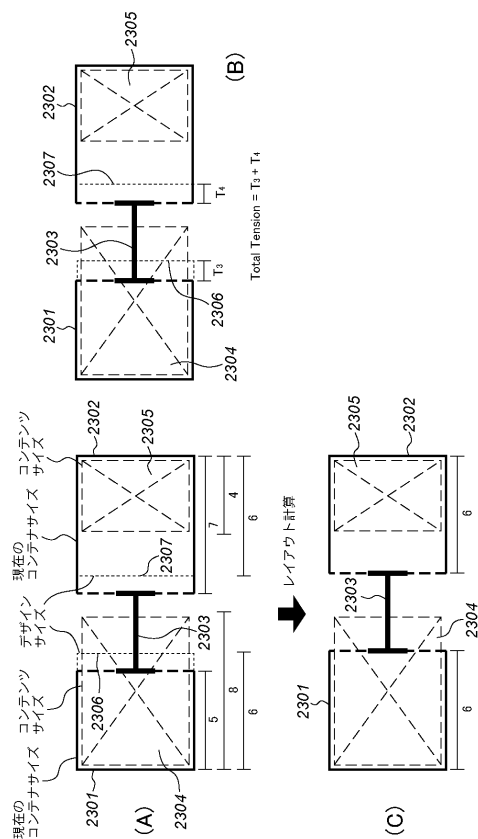
【図 2 1】



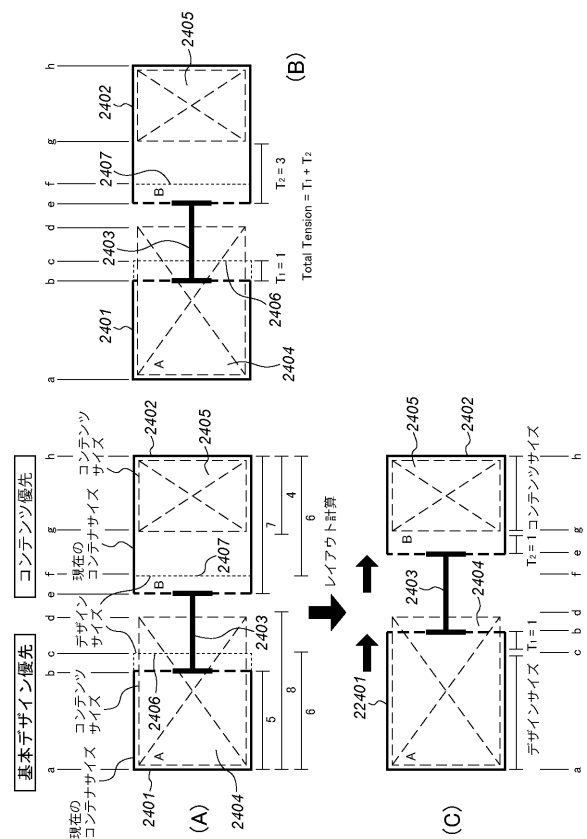
【図 2 2】



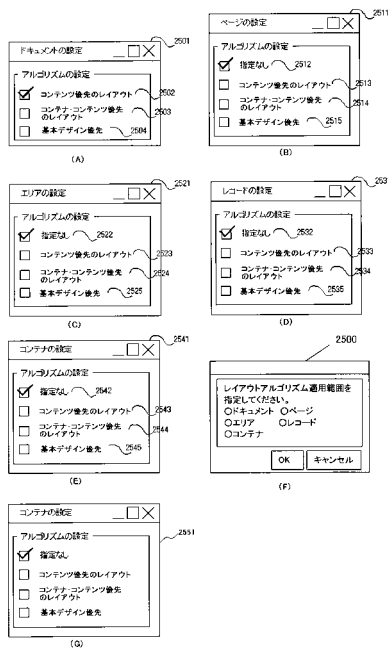
【図 2 3】



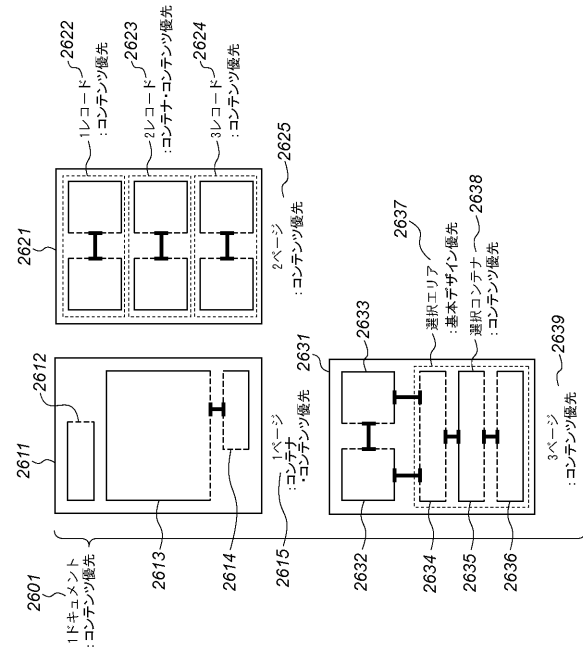
【図 2 4】



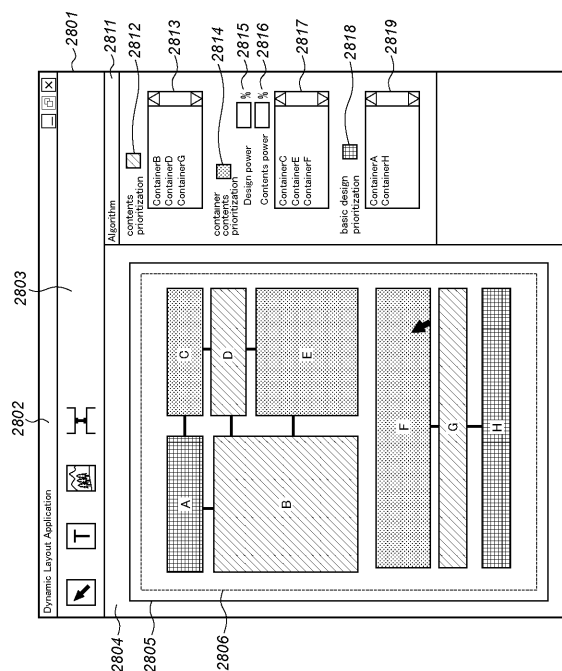
【図 25】



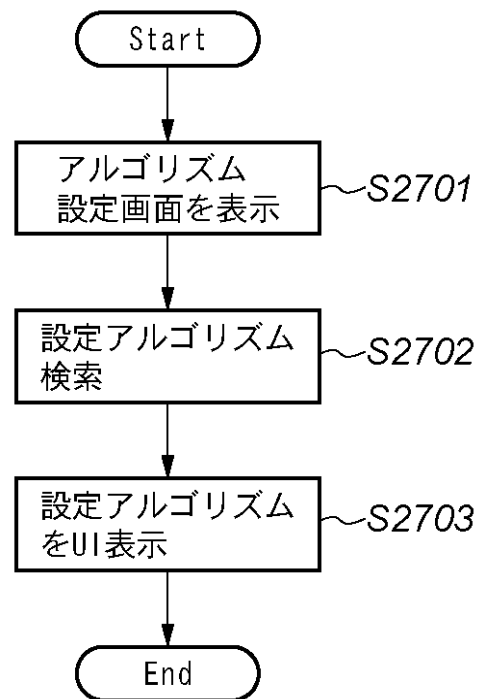
【図 26】



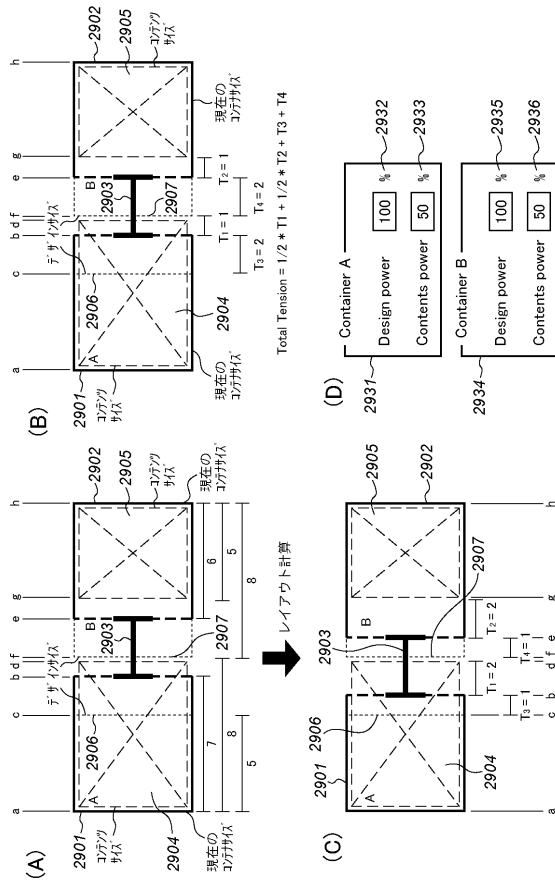
【図 27】



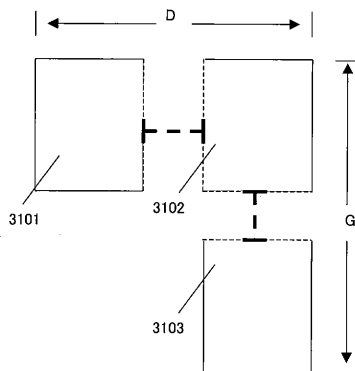
【図 28】



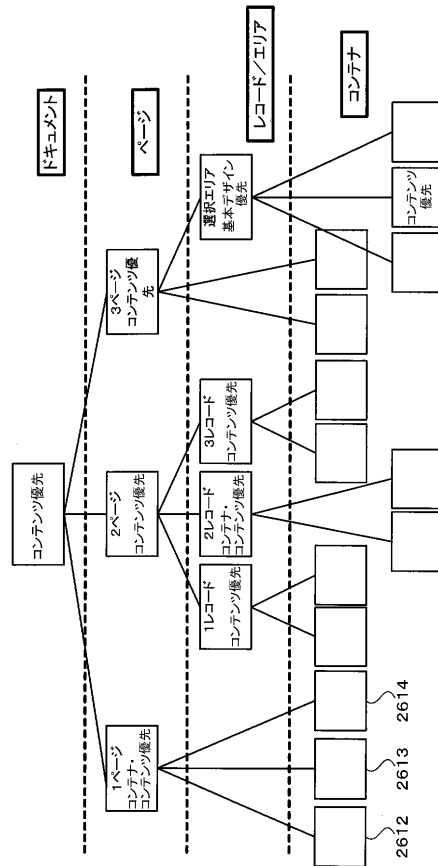
【図 29】



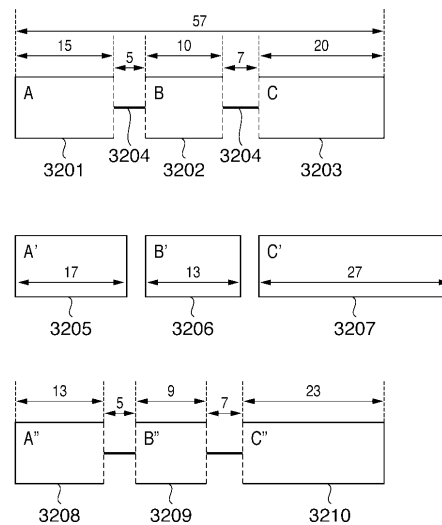
【図 31】



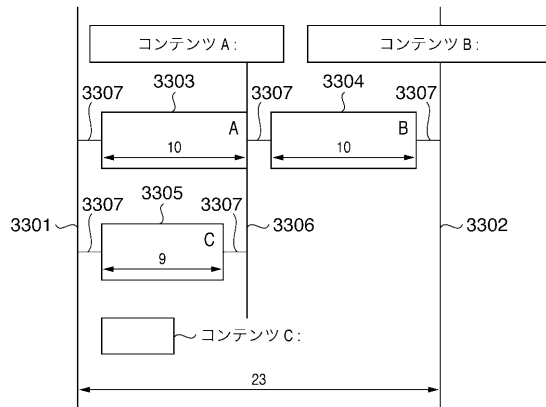
【図 30】



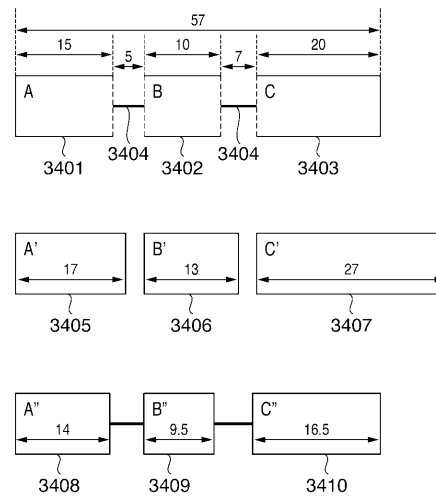
【図 32】



【図 3 3】



【図 3 4】



フロントページの続き

F ターム(参考) 5B050 AA10 BA06 BA16 BA18 CA07 EA12 EA19 FA02 FA08 FA13
5C076 AA17 AA19 AA21 AA22
5E501 AA02 AA03 AA04 AA14 BA03 CA02 DA12 EB11 FA06 FA14
FA44