

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4165888号  
(P4165888)

(45) 発行日 平成20年10月15日(2008.10.15)

(24) 登録日 平成20年8月8日(2008.8.8)

(51) Int.Cl.

F I

G 0 6 F 17/21 (2006.01)

G 0 6 F 17/21 5 3 6

請求項の数 21 (全 66 頁)

(21) 出願番号	特願2004-371886 (P2004-371886)	(73) 特許権者	000001007
(22) 出願日	平成16年12月22日(2004.12.22)		キヤノン株式会社
(65) 公開番号	特開2005-242992 (P2005-242992A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成17年9月8日(2005.9.8)	(74) 代理人	100076428
審査請求日	平成18年6月13日(2006.6.13)		弁理士 大塚 康德
(31) 優先権主張番号	特願2004-24589 (P2004-24589)	(74) 代理人	100112508
(32) 優先日	平成16年1月30日(2004.1.30)		弁理士 高柳 司郎
(33) 優先権主張国	日本国(JP)	(74) 代理人	100115071
前置審査			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 レイアウト制御方法、レイアウト制御装置及びレイアウト制御プログラム

(57) 【特許請求の範囲】

【請求項 1】

指示手段と、指定手段と、位置情報取得手段と、特定手段と、リンク設定手段とを備え、割り当てられたデータを収める複数の部分表示領域から、選択された少なくとも2つの部分表示領域を連結するリンクを当該少なくとも2つの部分表示領域間に設定するレイアウト制御装置におけるレイアウト制御方法であって、

前記指示手段が、前記リンクの設定を指示する指示工程と、

前記指定手段が、前記複数の前記部分表示領域から前記リンクによって関連付けられる少なくとも2つの部分表示領域を指定する指定工程と、

前記位置情報取得手段が、前記指定工程によって指定された前記少なくとも2つの部分表示領域の位置情報を取得する位置情報取得工程と、

前記特定手段が、前記位置情報取得工程によって取得された前記少なくとも2つの部分表示領域の位置情報に基づき、前記リンクによって関連づけられるべき部分表示領域間を特定する特定工程と、

前記リンク設定手段が、前記特定工程によって特定された前記リンクによって関連付けられるべき複数の部分表示領域間に前記リンクを設定するリンク設定工程とを備え、

前記指示工程の1度の指示に従って、前記特定工程は、前記少なくとも2つの部分表示領域の中で基準となる部分表示領域を設定し、前記基準となる部分表示領域の水平方向または垂直方向に交わる位置に配置されている部分表示領域を特定する処理を、前記基準となる部分表示領域を変更することにより複数回数実行することで前記複数の部分表示領域

10

20

間を特定し、前記リンク設定工程は特定された前記複数の部分表示領域間にリンクを設定することで複数のリンクを設定することを特徴とするレイアウト制御方法。

【請求項 2】

前記特定工程は、前記リンクによって関連付けられるべき部分表示領域間として、基準となる部分表示領域と最も距離が短い位置にレイアウトされる他の部分表示領域との間を更に特定することを特徴とする請求項 1 に記載のレイアウト制御方法。

【請求項 3】

前記レイアウト制御装置は更にレイアウト制御手段を備え、

前記レイアウト制御手段が、前記部分表示領域のレイアウトを制御するレイアウト制御工程を更に実行し、

前記割り当てられたデータを収めるために前記部分表示領域のサイズを変更する場合、前記レイアウト制御工程は、前記部分表示領域と前記リンクにより関連付けられたほかの部分表示領域とを連動させて、前記部分表示領域と当該他の部分表示領域とのサイズを変更することを特徴とする請求項 1 または 2 に記載のレイアウト制御方法。

【請求項 4】

前記レイアウト制御工程は、前記部分表示領域のサイズの変更に応じて、前記リンクの長さを変更することを特徴とする請求項 3 に記載のレイアウト制御方法。

【請求項 5】

前記レイアウト制御装置は更に表示手段を備え、

前記表示手段は、前記部分領域および前記リンクを表示する表示工程を実行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制御方法。

【請求項 6】

前記リンク設定工程は、水平方向または垂直方向に複数のリンクを設定することを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載のレイアウト制御方法。

【請求項 7】

前記リンク設定工程は、水平方向および垂直方向に複数のリンクを設定することを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載のレイアウト制御方法。

【請求項 8】

割り当てられたデータを収める複数の部分表示領域から、選択された少なくとも 2 つの部分表示領域を連結するリンクを当該少なくとも 2 つの部分表示領域間に設定するレイアウト制御装置であって、

前記リンクの設定を指示する指示手段と、

前記複数の前記部分表示領域から前記リンクによって関連付けられる少なくとも 2 つの部分表示領域を指定する指定手段と、

前記指定手段によって指定された前記少なくとも 2 つの部分表示領域の位置情報を取得する位置情報取得手段と、

前記位置情報取得手段によって取得された前記少なくとも 2 つの部分表示領域の位置情報に基づき、前記リンクによって関連づけられるべき部分表示領域間を特定する特定手段と、

前記特定手段によって特定された前記リンクによって関連付けられるべき複数の部分表示領域間に前記リンクを設定するリンク設定手段とを備え、

前記指示手段の 1 度の指示に従って、前記特定手段は、前記少なくとも 2 つの部分表示領域の中で基準となる部分表示領域を設定し、前記基準となる部分表示領域の水平方向または垂直方向に交わる位置に配置されている部分表示領域を特定する処理を、前記基準となる部分表示領域を変更することにより複数回数実行することで前記複数の部分表示領域間を特定し、前記リンク設定手段は特定された前記複数の部分表示領域間にリンクを設定することで複数のリンクを設定することを特徴とするレイアウト制御装置。

【請求項 9】

前記特定手段は、前記リンクによって関連付けられるべき部分表示領域間として、基準となる部分表示領域と最も距離が短い位置にレイアウトされる他の部分表示領域との間を

10

20

30

40

50

更に特定することを特徴とする請求項 8 に記載のレイアウト制御装置。

【請求項 10】

前記部分表示領域のレイアウトを制御するレイアウト制御手段を更に備え、

前記割り当てられたデータを収めるために前記部分表示領域のサイズを変更する場合、前記レイアウト制御手段は、前記部分表示領域と前記リンクにより関連付けられたほかの部分表示領域とを連動させて、前記部分表示領域と当該他の部分表示領域とのサイズを変更することを特徴とする請求項 8 または 9 に記載のレイアウト制御装置。

【請求項 11】

前記レイアウト制御手段は、前記部分表示領域のサイズの変更に応じて、前記リンクの長さを変更することを特徴とする請求項 10 に記載のレイアウト制御装置。

10

【請求項 12】

前記部分領域および前記リンクを表示する表示手段を更に備えることを特徴とする請求項 8 乃至 11 のいずれか 1 項に記載のレイアウト制御装置。

【請求項 13】

前記リンク設定手段は、水平方向または垂直方向に複数のリンクを設定することを特徴とする請求項 8 乃至 12 のいずれか 1 項に記載のレイアウト制御装置。

【請求項 14】

前記リンク設定手段は、水平方向および垂直方向に複数のリンクを設定することを特徴とする請求項 8 乃至 13 のいずれか 1 項に記載のレイアウト制御装置。

【請求項 15】

20

割り当てられたデータを収める複数の部分表示領域から、選択された少なくとも 2 つの部分表示領域を連結するリンクを当該少なくとも 2 つの部分表示領域間に設定するレイアウト制御をコンピュータに実行させるレイアウト制御プログラムであって、当該レイアウト制御プログラムが、コンピュータを

前記リンクの設定を指示する指示手段と、

前記複数の前記部分表示領域から前記リンクによって関連付けられる少なくとも 2 つの部分表示領域を指定する指定手段と、

前記指定手段によって指定された前記少なくとも 2 つの部分表示領域の位置情報を取得する位置情報取得手段と、

前記位置情報取得手段によって取得された前記少なくとも 2 つの部分表示領域の位置情報に基づき、前記リンクによって関連づけられるべき部分表示領域間を特定する特定手段と、

30

前記特定手段によって特定された前記リンクによって関連付けられるべき複数の部分表示領域間に前記リンクを設定するリンク設定手段として機能させ、

前記指示手段の 1 度の指示に従って、前記特定手段は、前記少なくとも 2 つの部分表示領域の中で基準となる部分表示領域を設定し、前記基準となる部分表示領域の水平方向または垂直方向に交わる位置に配置されている部分表示領域を特定する処理を、前記基準となる部分表示領域を変更することにより複数回数実行することで前記複数の部分表示領域間を特定し、前記リンク設定手段は特定された前記複数の部分表示領域間にリンクを設定することで複数のリンクを設定する処理を前記コンピュータに実行させることを特徴とするレイアウト制御プログラム。

40

【請求項 16】

前記特定手段は、前記リンクによって関連付けられるべき部分表示領域間として、基準となる部分表示領域と最も距離が短い位置にレイアウトされる他の部分表示領域との間を更に特定する処理を前記コンピュータに実行させることを特徴とする請求項 15 に記載のレイアウト制御プログラム。

【請求項 17】

前記レイアウト制御プログラムは、前記コンピュータを、前記部分表示領域のレイアウトを制御するレイアウト制御手段として更に機能させ、

前記割り当てられたデータを収めるために前記部分表示領域のサイズを変更する場合、

50

前記レイアウト制御手段は、前記部分表示領域と前記リンクにより関連付けられたほかの部分表示領域とを連動させて、前記部分表示領域と当該他の部分表示領域とのサイズを変更する処理を前記コンピュータに実行させることを特徴とする請求項 1 5 または 1 6 に記載のレイアウト制御プログラム。

【請求項 1 8】

前記レイアウト制御手段は、前記部分表示領域のサイズの変更に応じて、前記リンクの長さを変更する処理を前記コンピュータに実行させることを特徴とする請求項 1 7 に記載のレイアウト制御プログラム。

【請求項 1 9】

前記レイアウト制御プログラムは、前記コンピュータを、前記部分領域および前記リンクを表示する表示手段として更に機能させることを特徴とする請求項 1 5 乃至 1 8 のいずれか 1 項に記載のレイアウト制御プログラム。

10

【請求項 2 0】

前記リンク設定手段は、水平方向または垂直方向に複数のリンクを設定する処理を前記コンピュータに実行させることを特徴とする請求項 1 5 乃至 1 9 のいずれか 1 項に記載のレイアウト制御プログラム。

【請求項 2 1】

前記リンク設定手段は、水平方向および垂直方向に複数のリンクを設定する処理を前記コンピュータに実行させることを特徴とする請求項 1 5 乃至 1 9 のいずれか 1 項に記載のレイアウト制御プログラム。

20

【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レイアウト制御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、編集、そして印刷に好適なものである。

【背景技術】

【0 0 0 2】

近年、商品の多品種化で商品ライフサイクルが短くなっていること、インターネットの普及により、これを利用した消費者のカスタマイズサービス指向が強くなっていることなどの要因から C R M (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ものである。

30

【0 0 0 3】

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り、その代表的な手法としてバリエブルプリントが挙げられる。ここ最近では D T P (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エブルプリントシステムが開発され、顧客毎に異なる情報量のコンテンツを最適にレイアウトして表示することが求められるようになった。

40

【0 0 0 4】

従来におけるバリエブルプリントシステムは、ドキュメント上に情報を表示する領域としてコンテナ(帳票フォームではフィールド領域とも呼ばれる)等をレイアウトして、データベースとコンテナとを関連付けることによりレイアウト表示を達成していた。

【0 0 0 5】

しかし、テキストおよびイメージが貼り付けられる部分表示領域であるコンテナのサイズが固定であったため、データベース内のデータがコンテナに挿入されたときに、データ量がコンテナサイズより多いとテキストのオーバーラップやイメージのクリッピングが発生し、またデータ量がコンテナサイズより小さいとコンテナ内に隙間が空いてしまい、い

50

ずれの場合でも表示しようとするテキストやイメージの情報量に応じた最適なレイアウト表示を実現することはできなかった。

【 0 0 0 6 】

その問題を解決するために、情報量に応じてコンテナサイズを可変にする自動レイアウトシステムが提案されている。この自動レイアウトシステムはテキストおよびイメージのコンテナサイズを可変に設定することが可能である。この自動レイアウトシステムにおいて、コンテナのサイズを可変とし、差し込まれるデータ量に応じてコンテナのサイズを大きくするよう変更できるものがある。また、テキストの場合において固定のコンテナサイズ内に入りきらないデータが挿入された場合、テキストのフォントサイズを縮小し、コンテナ内に全てのテキストを表示する技術も存在する。

10

【 0 0 0 7 】

しかし、コンテナのサイズを大きくした場合、ドキュメント上の他のコンテナに重なってしまうという問題が生じる。またフォントサイズを調節する場合はテキストの量が多い場合、フォントサイズが小さくなりすぎるという問題も生じる。これらの問題を解決するために、さらなる自動レイアウトの技術として、あるコンテナのサイズが大きくなった場合、このコンテナに隣接する他のコンテナのサイズを小さくする技術が特許文献 1 の「レイアウトデザイン装置」に開示されている。

【特許文献 1】特開平 1 1 - 3 1 6 7 9 2 号公報

【発明の開示】

【発明が解決しようとする課題】

20

【 0 0 0 8 】

しかしながら、上述の自動レイアウトシステムにおいて、各コンテンツに貼り付けられるデータに合わせてコンテナの 1 例である可変表の行を増やし、それに応じて他のコンテナが移動する位置合わせの技術についてしか考慮していない。つまり、可変表の行が増えればそれだけ他のコンテナが移動し、改ページが行われる帳票のレイアウトデザインを作成するためのものである。しかしながら、上述した One-to-one マーケティングでは、決められた用紙サイズのページにユーザが所望とするコンテナ数をレイアウトすることが望まれている。つまり、従来のような単なる可変表の技術では対応できないという問題がある。そこで、それぞれのコンテナにデータを流し込み、各コンテナのサイズを可変にしつつ、決められた用紙サイズのページに必要な数のコンテナを動的に配置するような、自動レイアウトシステムの需要がある。このような自動レイアウトシステムでは、お互いのコンテナのサイズが可変であるため、各コンテナ同士で押し合いつつレイアウトを動的に変化させなければならない。レイアウトを動的に変化させるためには、各コンテナ間に関連付け（以下、リンク）を設定する必要があるが、リンクはコンテナ間において一つ一つに対して設定する必要があるが、ユーザの操作負担を強いるものとなってしまう。すなわち、複数のコンテナをレイアウトして、それぞれのコンテナを動的にレイアウトするように設定するためには、コンテナ間にリンクを設定しなければならないが、コンテナの数が増えると、そのリンクの数も増えることになり、リンクを一つ一つ設定する作業にはコンテナの数に比例して増大し、ユーザの操作負担は大きくなってしまふ。

30

【 0 0 0 9 】

本発明は上記の課題に鑑みてなされたものであり、複数のコンテナ間に対して、一つの操作でリンクを一括設定することを可能にし、効率のよい操作を実現することを目的とする。

40

【課題を解決するための手段】

【 0 0 1 0 】

上記目的を達成するために、本発明に係るレイアウト制御方法は主として以下の工程を有することを特徴とする。

【 0 0 1 1 】

すなわち、指示手段と、指定手段と、位置情報取得手段と、特定手段と、リンク設定手段とを備え、割り当てられたデータを収める複数の部分表示領域から、選択された少なく

50

とも2つの部分表示領域を連結するリンクを当該少なくとも2つの部分表示領域間に設定するレイアウト制御装置におけるレイアウト制御方法は、

前記指示手段が、前記リンクの設定を指示する指示工程と、

前記指定手段が、前記複数の前記部分表示領域から前記リンクによって関連付けられる少なくとも2つの部分表示領域を指定する指定工程と、

前記位置情報取得手段が、前記指定工程によって指定された前記少なくとも2つの部分表示領域の位置情報を取得する位置情報取得工程と、

前記特定手段が、前記位置情報取得工程によって取得された前記少なくとも2つの部分表示領域の位置情報に基づき、前記リンクによって関連づけられるべき部分表示領域間を特定する特定工程と、

前記リンク設定手段が、前記特定工程によって特定された前記リンクによって関連付けられるべき複数の部分表示領域間に前記リンクを設定するリンク設定工程とを備え、

前記指示工程の1度の指示に従って、前記特定工程は、前記少なくとも2つの部分表示領域の中で基準となる部分表示領域を設定し、前記基準となる部分表示領域の水平方向または垂直方向に交わる位置に配置されている部分表示領域を特定する処理を、前記基準となる部分表示領域を変更することにより複数回数実行することで前記複数の部分表示領域間を特定し、前記リンク設定工程は特定された前記複数の部分表示領域間にリンクを設定することで複数のリンクを設定することを特徴とする。

#### 【0012】

あるいは、上記目的を達成するために本発明に係るレイアウト制御装置は、割り当てられたデータを収める複数の部分表示領域から、選択された少なくとも2つの部分表示領域を連結するリンクを当該少なくとも2つの部分表示領域間に設定するレイアウト制御装置であって、

前記リンクの設定を指示する指示手段と、

前記複数の前記部分表示領域から前記リンクによって関連付けられる少なくとも2つの部分表示領域を指定する指定手段と、

前記指定手段によって指定された前記少なくとも2つの部分表示領域の位置情報を取得する位置情報取得手段と、

前記位置情報取得手段によって取得された前記少なくとも2つの部分表示領域の位置情報に基づき、前記リンクによって関連づけられるべき部分表示領域間を特定する特定手段と、

前記特定手段によって特定された前記リンクによって関連付けられるべき複数の部分表示領域間に前記リンクを設定するリンク設定手段とを備え、

前記指示手段の1度の指示に従って、前記特定手段は、前記少なくとも2つの部分表示領域の中で基準となる部分表示領域を設定し、前記基準となる部分表示領域の水平方向または垂直方向に交わる位置に配置されている部分表示領域を特定する処理を、前記基準となる部分表示領域を変更することにより複数回数実行することで前記複数の部分表示領域間を特定し、前記リンク設定手段は特定された前記複数の部分表示領域間にリンクを設定することで複数のリンクを設定することを特徴とする。

#### 【0013】

あるいは、上記目的を達成するために本発明に係るレイアウト制御プログラムは、割り当てられたデータを収める複数の部分表示領域から、選択された少なくとも2つの部分表示領域を連結するリンクを当該少なくとも2つの部分表示領域間に設定するレイアウト制御をコンピュータに実行させるレイアウト制御プログラムであって、当該レイアウト制御プログラムが、コンピュータを

前記リンクの設定を指示する指示手段と、

前記複数の前記部分表示領域から前記リンクによって関連付けられる少なくとも2つの部分表示領域を指定する指定手段と、

前記指定手段によって指定された前記少なくとも2つの部分表示領域の位置情報を取得する位置情報取得手段と、

前記位置情報取得手段によって取得された前記少なくとも2つの部分表示領域の位置情報に基づき、前記リンクによって関連づけられるべき部分表示領域間を特定する特定手段と、

前記特定手段によって特定された前記リンクによって関連付けられるべき複数の部分表示領域間に前記リンクを設定するリンク設定手段として機能させ、

前記指示手段の1度の指示に従って、前記特定手段は、前記少なくとも2つの部分表示領域の中で基準となる部分表示領域を設定し、前記基準となる部分表示領域の水平方向または垂直方向に交わる位置に配置されている部分表示領域を特定する処理を、前記基準となる部分表示領域を変更することにより複数回数実行することで前記複数の部分表示領域間を特定し、前記リンク設定手段は特定された前記複数の部分表示領域間にリンクを設定することで複数のリンクを設定する処理を前記コンピュータに実行させることを特徴とする。

10

【発明の効果】

【0014】

複数のコンテナ間に対して、一つの操作でリンクを一括設定することを可能にし、効率のよい操作を実現することが可能になる。

【発明を実施するための最良の形態】

【0015】

本発明の好適な実施形態を図面を参照しつつ説明する。

【0016】

20

(システム構成図)

図1Aはコンテナサイズを可変に設定してドキュメントを印刷するシステム100を示している。このシステムにおいて実行するコンテナサイズの制御、コンテナ間のリンク設定は図1Bで詳しく説明されるホストコンピュータ101の制御の下に実行される。図6で記述されるプロセスは、レイアウト制御装置であるホストコンピュータ101内で実行され、システム100上で実行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(本発明のレイアウト制御プログラム)のようにソフトウェアの全体、あるいはその一部分で実行される。特にレイアウト編集や印刷のステップはホストコンピュータ101によるソフトウェアによって実行される。

【0017】

30

ソフトウェアは、例えば、以下に記述されるような記憶装置を含むコンピュータの可読媒体に格納される。ソフトウェアはコンピュータの可読媒体からホストコンピュータにロードされ、実行される。そのようなソフトウェアや媒体に記録されたコンピュータプログラム、あるいはそのプログラムを格納するコンピュータの可読媒体は、コンピュータと協働したドキュメントのレイアウト編集やバリアブルデータ印刷に必要な演算処理や表示制御、レイアウト制御、コンテンツに関する情報処理を実行する手段としてコンピュータの機能を特定する。

【0018】

ホストコンピュータ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、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を含む出力装置と接続する。入力/出力(I/O)インターフェース138はホストコンピュータ101をネットワーク107に接続してシステム100を他のコンピュータ装置(例えば、データベースサーバ117)に接続することができる。そのネットワーク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。

40

【0019】

ホストコンピュータ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、ビデオインターフェース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

50

／Ｏインターフェース１４３を含んでいる。記憶装置１３９は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。ここでは図示されていないが磁気テープドライブもまた記憶装置として使用することができる。

【００２０】

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。ホストコンピュータ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い形成されたコンピュータシステムのオペレーションによって、相互接続バス１３４を介して通信を行うホストコンピュータ１０１のプロセッサユニット１３５からＩ／Ｏインターフェース１４３を利用することができる。図１Ａに示したホストコンピュータ１０１を含んだシステムの例としては、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。

10

【００２１】

レイアウトアプリケーションプログラム１２１は典型的にはハードディスクドライブ１４０に常駐し、プロセッサユニット１３５により読み出され、実行される。プログラム１２１を記憶する記憶装置１３９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、またはこれに呼応してメモリユニット１３６が使用され、アプリケーションプログラム１２１がＣＤ－ＲＯＭ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ＣＤ－ＲＯＭドライブ１４２やＦＤＤ１４１を通じて読み込まれユーザに提供される。

20

【００２２】

あるいはもう一つの方法としてアプリケーションプログラム１２１はネットワーク１０７からユーザによってインストールするようにしてもよい。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無線またはホストコンピュータ１０１とその他のデバイス間の赤外線通信、ＰＣＭＣＩＡ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、他の適当な大きさのコンピュータ可読媒体からホストコンピュータ１０１内にロードすることもできる。但し、これらのコンピュータ可読媒体は、単に関連するコンピュータ可読媒体の規範であり、媒体の種類がこれらに限定される趣旨ではなく、他のコンピュータ可読媒体も使用することも可能である。

30

【００２３】

また、レイアウト編集と名づけられたアプリケーション１２１はバリアブルデータ印刷(ＶＤＰ)を行うためのプログラムモジュールであり、２つのソフトウェアコンポーネントを含んでいる。これらのうち１つめのモジュールは、レイアウトエンジンモジュール１０５であり、これはデータが流し込まれる部分表示領域であるコンテナ(矩形の範囲)のサイズや位置の制限に従ってデータベース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ためのソフトウェアモジュールである。レイアウトエンジン１０５は各部分表示領域(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。

40

【００２４】

２つめのモジュールであるユーザインターフェースモジュール１０３はユーザにドキュメントテンプレートを作成させ、ドキュメントテンプレート内でデータソースとコンテナとを関連付けるメカニズムを提供するモジュールである。ユーザインターフェースモジュール１０３とレイアウトエンジンモジュール１０５はコミュニケーションチャンネル１２３を介して通信することが可能である。ドキュメント作成のためのデータソースは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上にあるデータベース１１９に格納されている。ホストコンピュータ１０１

50



はネットワーク 107 との接続を介してデータベースサーバ 117 と通信することができる。バリアブルデータ印刷を実行する場合、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 か、若しくは他のコンピュータで構成されるファイルサーバ 115 に保存されるドキュメントテンプレートを生成する。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 の記憶装置 139 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。

#### 【0025】

プリントサーバ 109 は直接ネットワークに接続していないプリンタ 113 にネットワーク機能を提供するコンピュータである。プリントサーバ 109 とプリンタ 113 は典型的な通信チャネル 111 を介して接続する。

#### 【0026】

図 2 はレイアウトエンジンモジュール 105 をホストコンピュータ 101 の他にエンジンサーバ 227 にもレイアウトエンジン 225 を設けた構成を示す図である。エンジンサーバ 227 は典型的なコンピュータにより構成され、ファイルサーバ 115 に保存されたドキュメントテンプレートは印刷や他の目的がある際、レイアウトエンジン 225 によってドキュメントを生成するためにデータベース 119 に保存されたデータと結合することができる。そのようなオペレーションはユーザインターフェースモジュール 103 の制御の下、ユーザインターフェース (UI) 画面を介して入力され、特定のレコードのみ印刷するように設定することができる。

#### 【0027】

(アプリケーション構成図)

(メインウインドウ)

図 3 で示されるアプリケーションウインドウ 301 は、ユーザインターフェースモジュール 103 (図 1) によって、操作時にディスプレイ装置 144 に表示される。アプリケーションウインドウ 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 を有しており、これらにより特徴付けされる。

#### 【0028】

メニューバー 302 は、周知の技術として知られているように、メニューオプションの階層の下に拡張される多くのメニューアイテム 304 を有し、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したり、または表示することが可能な多くのツールボタン 305 を有する。オプションのルーラー 308 はワークエリア内のポインター、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。パレット 311 はバリアブルデータライブラリのような追加機能にアクセスするために使われる。パレット 311 は移動、リサイズ、クローズをするためのウインドウコントロール機能を提供するボタン 312 を有する。パレット 311 の表示はオプションで、ワークエリアの前面に表示することもでき、あるいは背面に隠すこともできる。また、このパレット 311 に対する表示制御は、アプリケーションウインドウ 301 の範囲内のみに制限したり、あるいはアプリケーションウインドウ 301 の外側に部分的、あるいはその全体を表示するようにすることもできる。

#### 【0029】

ツールバーエリア 303 には少なくとも、ユーザによる選択が可能な「ボタン」(403 ~ 406) を備える(例えば、図 4 を参照)。以下、ボタン 403 ~ 406 を説明する。

#### 【0030】

選択ツールボタン 403 は、コンテナの辺を選択、移動、サイズ変更、そしてコンテナのロック(固定)、あるいはロック解除をするために使われる。コンテナは、複数のコン

10

20

30

40

50

テナの周りをドラッグする、あるいは複数コンテナを選択する間にCTRLキーを押すことによって、複数のコンテナの選択を可能にする。

【0031】

イメージコンテナツールボタン405は、スタティック或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。

【0032】

テキストコンテナツールボタン404は、スタティック或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。

【0033】

リンクツールボタン406は、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。

【0034】

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として実装される。

【0035】

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各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。

【0036】

(ドキュメントテンプレート)

図3に示したワークエリア306はドキュメントテンプレートのデザインを表示・編集するために使用される。これはユーザに下準備で印刷された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や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を可能にするものである。外部のデータベースがテンプレートにリンクされている場合には、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をそれらのコンテナに表示することができる。ドキュメントの構造とバリアブルデータコンテナの表示は、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するようにすることができる。

【0037】

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に特徴付けられる。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。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、周知の技術とし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における実際のページ数は、バリアブルデータによって変化する可能性があり、もし1ページ内にバリアブルデータを収めることができず、所定の用紙サイズのページに各コンテナを収めるような制約が基本レイアウトになかった場合には、追加のページは自動的に作成される。所定の用紙サイズのページに各コンテナを収めるように制約(後述するアンカーアイコン等により設定可能である)が基本レイアウトに設定されている場合は、ページ内で各コンテナがお互いに押しやり押しされたりしながら、各コンテナサイズと位置が決定され、必要に応じて流し込まれるバリアブルデータが縮小され、レコード毎に動的にレイアウトが決定される。

10

20

30

40

50

## 【 0 0 3 8 】

図 3 において破線で示される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ものであり、任意に設定することが可能なページマージン 3 1 0 である。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を示す図であり、図 4 では複数のコンテナ 4 0 7 及び 4 0 8 が示されており、コンテナの辺 4 1 4 の位置を固定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 によりコンテナ間の相互の関係が規定されている。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。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。アンカーアイコン 4 0 9 が辺に設定されている場合は、その辺が固定となり、その他の 3 辺の各方向に拡大可能である。また、アンカーアイコン 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 は詳細は後述するが、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おり、このリンクに設定されている長さ（範囲指定が可能）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を示している。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。

## 【 0 0 3 9 】

## ( 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/イメージがレコード単位に流し込まれ、描画されるスペース（これを部分表示領域と呼ぶ）であり、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。

## 【 0 0 4 0 】

コンテナは、アンカーアイコン 4 0 9、リンク 4 1 2、スライダー 4 1 3 によりコンテナの変形に関する拘束条件が規定されている。テキストやイメージデータの情報量に応じてその変形が許容されている方向にコンテナスペースの変形が制御されて、テキストデータやイメージデータがそのコンテナ内にレイアウトされる。2 つのコンテナに関して相互に変形が必要となる場合、各コンテナに収納すべきテキストデータやイメージデータの量に応じて、相互のコンテナの変形がバランスするように（各コンテナにストレスの無い状態）に各コンテナの形状が制御される。

## 【 0 0 4 1 】

コンテナはポインティングデバイスとしてのマウス 1 3 3 の動作によりワークエリア 3 0 6 内の位置が特定され（図 3 の 3 1 3）、ユーザインターフェースとしての画面内（図 3）で移動、サイズ調整が施され、新たなコンテナを追加することも可能である。コンテナは、アンカーアイコンや、リンクそしてスライダーによりコンテナ間の相互の関係等の変形に関する種々の条件が設定され、コンテンツを表示する視覚的表現、そしてコンテナ間のインタラクションとコンテナ内の情報の編集機能を有している。ここで、コンテナを定義すると以下になる。

## 【 0 0 4 2 】

( 1 ) コンテナは固定あるいは可変のコンテンツが入力され、可変コンテンツは、データソースからデータをとってきて、異なるドキュメントでは異なるデータに応じてコンテナサイズを可変するという意味でダイナミック（動的）である。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を変更するコンテンツを含むことは印刷に適合していないため意図していない。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。けれども、可変コンテンツとリンクが設定されている場合、可変コンテナの動作によって、固定コンテンツはそれぞれのドキュメントで表示位置が異なるように表示制御される。

【 0 0 4 3 】

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

【 0 0 4 4 】

( 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、典型的に印刷された出力物に対して反映され、装飾が反映された結果は可視化される。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。すなわち、その特定のデータに対して、コンテナに設定されている条件に従い、コンテナのサイズが可変に設定される。可変に設定されたコンテナのレイアウト、およびコンテナ内の特定のデータ表現は、例えば、プリンタ 1 1 3 を介して印刷されるか、ディスプレイ装置 1 4 4 上で表示されるか、その両方により処理することが可能である。

【 0 0 4 5 】

( 4 ) コンテナの設定に関し、システムはユーザインターフェースモジュール 1 0 3 を備え、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 ) を有する。そして、ユーザインターフェースの要素はディスプレイ装置 1 4 4 上に表示される。しかしドキュメントは印刷されない。ユーザインターフェースモジュール 1 0 3 は、背景色やフォントのようなコンテナに関する装飾機能のいくつかをインターフェース画面上に表示させ、これらの装飾機能をコンテナの設定、編集、コンテナ表示に追加することができる。

【 0 0 4 6 】

( 5 ) 更に、ユーザインターフェースモジュール 1 0 3 による設定の例としては、コンテナ間のボーダー、あるいは、コンテナのサイズや位置を対話的に変更、表示するためにコンテナに設定させるコーナー部のアイコンや、あるいはデータソースからデータをマージしたとき、コンテナの動作 ( 例えば、上塗りした数、線、アイコンの設定、テキストの編集 ) を設定する。

( コンテナに対する制約 )

コンテナはそれぞれのドキュメントにおけるコンテンツの内容を表示するために、コンテナをコンテンツと結びつけて表示制御、レイアウト制御するための制約がある。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 ) 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をコントロールする主要な方法である。制約の例は、たとえば、コンテナのコンテンツの高さは、最大値 4 インチとするような制約が挙げられる。もうひとつの制約としては、コンテナのコンテンツの左エッジは、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とするような制約である。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、編集のために設定することが可能である。

【 0 0 4 7 】

イメージがページ上に定義された位置を持っているように、コンテンツの配置もその位置が特定される。コンテナは位置とサイズを持ち、それらはバリアブルデータ印刷において、コンテナ内に挿入されるコンテンツの内容に応じてコンテナのレイアウトが設定された条件の下、コンテンツの内容を表示・編集することを可能にする。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、コンテナはユーザインターフェースモジュール 1 0 3 により与えられるコンテナに対する制約を指定することができる。

## 【 0 0 4 8 】

1つのコンテナを構成する辺は、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。たとえば、矩形のコンテナの左辺は、関連付けられたコンテンツが配置される左側の位置を与え、同様に、コンテナの高さは生成されたドキュメントに関連付けられたコンテンツの高さの制約を与える。

## 【 0 0 4 9 】

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 0 】

( 1 ) コンテナの幅が固定なら、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。

10

## 【 0 0 5 1 】

( 2 ) コンテナの高さが固定なら、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。

## 【 0 0 5 2 】

( 3 ) 距離の制約が固定なら、指定された距離は全てのドキュメントの制約となる。

## 【 0 0 5 3 】

( 4 ) コンテナの左右辺が固定の場合、コンテナの高さあるいは垂直方向の位置は、コンテンツを完全にコンテナ内に収容するために、コンテナサイズは変形が許容されている高さ方向あるいは垂直方向に変えることが可能である。

20

## 【 0 0 5 4 】

( 5 ) コンテナの上下辺が固定の場合、コンテナの幅あるいは水平位置は、コンテンツを完全にコンテナ内に収容するために、コンテナサイズは変形が許容されている幅方向あるいは水平方向に変えることが可能である。

## 【 0 0 5 5 】

( 6 ) コンテナの垂直軸はコンテナの右と左辺の平行で、そして中間に位置される想像上の垂直線である。もしコンテナの垂直軸が固定なら、コンテナの左右辺の水平位置の平均は、すべてのドキュメントで同じである。この制約で、コンテナの幅は変化するかもしれない、左右辺両方が異なったドキュメントで、垂直軸にもっとも遠いかもっとも近いかもしれない、しかし軸は全てのドキュメントで同じ水平位置にある。コンテナの高さと水平位置はこの制約によって影響されない。

30

## 【 0 0 5 6 】

( 7 ) 同様に、もし水平軸が固定なら、コンテナの上そして下辺が垂直に位置されることを制約する、けれども高さは、この制約によって影響されない。

## 【 0 0 5 7 】

( 8 ) 水平、垂直軸両方が固定なら、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、幅・高さはこの制約によって影響されない。

## 【 0 0 5 8 】

( 9 ) コンテナの角、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なら、すべてのドキュメントで同じ場所で、そしてコンテナに関連付けられた同じ場所で表示される。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。

40

## 【 0 0 5 9 】

( 1 0 ) 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。

## 【 0 0 6 0 】

「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

50

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 1 2 1 では、各コンテナ（部分表示領域）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。これにより、ユーザはドキュメントテンプレート 1 8 0 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。

#### 【 0 0 6 1 】

（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種類ある。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。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ボタン 4 0 4、あるいはイメージコンテナツールボタン 4 0 5 をマウス 1 3 3 でクリックし、テンプレート 3 0 9 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 3 0 9 上に作成される。コンテナは、適切なツール 4 0 4、4 0 5 をアクティブにした後に、ドキュメントテンプレート 3 0 9 上でマウス 1 3 3 をクリックすることによって作成される。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。コンテナは自動的に前もって定義され、計算されたスキーマによって作成され、ドキュメントテンプレート 3 0 9 上に配置される。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（部分表示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。

#### 【 0 0 6 2 】

（コンテナの表示方法）

図 5 A ~ 5 D はコンテナを構成する辺に対するルールを例示的に説明する図である。レイアウト編集アプリケーション 1 2 1 は、辺の状態を表現するために、塗りつぶし線（5 0 3）あるいは点線（5 0 4）で辺を描き、アンカー（辺の近くに描画された 5 0 6、5 0 7、5 0 9 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、5 0 2）、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 4 1 3）、拡張アイコン（5 0 5）、そして色を特徴として持っている。

#### 【 0 0 6 3 】

図 5 A ~ 5 D に示すコンテナの表示方法のルールは、つぎの通りである。

#### 【 0 0 6 4 】

1. それぞれの辺を固定するために、塗りつぶし線で描画する。

#### 【 0 0 6 5 】

2. もし幅が固定なら、左と右の辺を塗りつぶし線で描画する。

#### 【 0 0 6 6 】

3. もし高さが固定なら、上と下の辺を塗りつぶし線で描画する。

#### 【 0 0 6 7 】

4. 軸は描画しない。

#### 【 0 0 6 8 】

5. まだ描画されていない全ての辺は、それぞれの辺の近くに拡張アイコンが描画され、点線になる。

## 【 0 0 6 9 】

6. 垂直辺あるいは軸のそれぞれのペアで、もし両者が固定なら、交差点にアンカーが描画される。

## 【 0 0 7 0 】

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、エッジの中央にスライダーが描画される。

## 【 0 0 7 1 】

8. 垂直辺あるいは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、交差点にハンドルが描画される。

## 【 0 0 7 2 】

上述のルール 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 3 】

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 4 】

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。レイアウト編集アプリケーション 1 2 1 は以上のルールに従って、コンテナの表示制御、レイアウト制御を実行することができる。

## 【 0 0 7 5 】

コンテンツの表現は、コンテナの辺に設定されたアンカーアイコン、スライダーアイコン等の設定に従い、サイズを可変にすることが可能なコンテナに挿入されることにより可視化される。コンテナに設定される具体的なアイコン及びコンテナを構成する辺は、以下に説明するように機能する。

## 【 0 0 7 6 】

図 4 の辺 4 1 0 のように、点線はコンテナに挿入されるコンテンツに依存して、ドキュメント内の辺の位置が移動可能であることを意味する。そして、実線 4 1 4 は位置が制限された辺を意味する。

## 【 0 0 7 7 】

アンカーは辺または軸が交差した場所が固定されていることを意味する。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示すアンカーアイコンの例である。

## 【 0 0 7 8 】

スライダーは関係付けられた辺が、その辺の垂直方向に固定されているが、並行移動する可能性があることを意味する。幅・高さのサイズは、レイアウト編集アプリケーション 1 2 1 の制御の下、副次的なダイアログウィンドウに表示され、コンテナサイズとして許容される基本値・最小値・最大値の基本パターンを設定することが可能である。

## 【 0 0 7 9 】

図 5 A において、コンテナ 5 0 1 を構成する実線で示される辺 5 0 3 は固定であり、点線で示される辺 5 0 4 は幅・高さ両方が可変である。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す。また、図 5 B において、コンテナ 5 0 1 を構成する実線で示される辺 5 0 3 は固定であり、同図において、アンカーアイコン 5 0 6 は、辺 5 0 3 が交差する高さ方向および幅方向に変位が拘束されることを示す。

10

20

30

40

50

## 【 0 0 8 0 】

図 5 C では、コンテナ 5 0 1 は、コンテナの拡大あるいは縮小が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に幅方向及び高さ方向に均等に広がるという状態に設定されており、幅・高さ両方が各辺において可変である。また、図 5 D において、コンテナ 5 0 1 は、上辺 5 0 8 に対してスライダ - アイコン 5 0 9 が設定されており、固定されているが、点線で示される辺 5 0 2 は幅及び高さの両方が可変である。この場合、アンカーアイコン 5 0 9 を通る中心軸（垂直軸）を基準として、辺 5 0 2 の左右方向、及び上下方向に変位をすることが可能であり、これによりコンテナのサイズを変えることが可能である。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 1 】

（リンクの設定方法）

図 6 はレイアウト編集アプリケーション 1 2 1 によるリンクの設定方法を示すフローチャートであり、図 7 はユーザインターフェースの画面表示例を示す図である。以下、図 6、図 7 を用いてコンテナにリンクを設定する操作方法について説明をする。まず、リンクをコンテナに設定するためには、リンク設定の対象となるコンテナ（最低 2 つ）を作成する（S 6 0 1）。図 7（A）は、2 つのコンテナ 7 0 1、7 0 2 が作成された状態を示している。

## 【 0 0 8 2 】

次に、前述したリンクツールボタン（図 4 を参照）を選択した状態にする（S 6 0 2）。ここで、図 7（A）のコンテナ 7 0 1 と 7 0 2 は、前述した図 4 のコンテナ 4 0 7 と 4 0 8 と同じ内容のものを示しており、固定されている辺（実線で示される）によりコンテナが構成されることを示している。また、アイコン 7 0 3 と 7 0 4 は、図 4 におけるアイコン 4 0 9 と同じアンカーアイコンであることを示し、7 0 5 はマウスポインタを意味している。

20

## 【 0 0 8 3 】

ステップ S 6 0 2 において、リンクツールボタン 4 0 6 の選択が完了すると、次に、コンテナを選択する（S 6 0 3）。この選択は、リンクを設定する対象となるコンテナの片方（たとえば、7 0 1）をマウス 1 3 3 によりクリックすることでコンテナは選択される。次に、リンクを設定する他方のコンテナ（たとえば、7 0 2）をマウス 1 3 3 によりク

30

## 【 0 0 8 4 】

図 7（B）の 7 0 6 はステップ S 6 0 3 でマウス 1 3 3 をクリックした位置（P 1）とステップ S 6 0 4 でマウスをクリックした位置（P 2）とを結ぶ、マウスポインタ 1 3 3 の概略的な軌跡を示す線分であり（図 6 の S 6 0 5）、この線分が交わるコンテナの辺 7 1 1 及び辺 7 1 2 の間にリンク 7 0 7 が設定され、最終的にリンク 7 0 7 がコンテナ間に設定された状態のインターフェース画面が更新される（S 6 0 6）。リンク 7 0 7 が設定されたことにより、コンテナを表示するための画面表示も自動的に切り替えられる。切り替えられた画面（図 7（C））において、辺 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している。固定の辺（リンク設定前の 7 1 1、7 1 2 が対応）から可変の辺に 7 0 8 が切り替えられたのは、リンク 7 0 7 を設定したことにより、コンテナ 7 0 1、7 0 2 の辺を可変にする必要があるためであり、リンク 7 0 7 を設定したにもかかわらず、全ての辺が固定とすると、コンテナ間の相対的な関係をリンクにより特定するという矛盾を防ぐためであり、レイアウト編集アプリケーション 1 2 1 の制御の下、自動的に行われる処理である。

40

## 【 0 0 8 5 】

また、7 0 9 は図 5 の 5 0 5 と同様に機能し、リンク 7 0 7 を設定したことにより、コンテナ 7 0 1、7 0 2 が変位できる方向をユーザに視覚的に示す識別用の表示である。図 7（C）の例では、左のコンテナ 7 0 1 の右辺と右のコンテナ 7 0 2 の左辺が可変に変化した。これは一例であり、左のコンテナ 7 0 1、右のコンテナ 7 0 2 が、図 4 の 4 1 3

50



で示したスライダーを持つ設定により辺 7 0 8 が可変であることを表示するようにすることもできる。以上の処理はレイアウト編集アプリケーションの制御のもとに実行することができる。

#### 【 0 0 8 6 】

( 可変長のリンク )

図 4 6 のインターフェース画面において、一般的な可変リンク 4 6 0 9 が示されている。同図においては、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 4 6 0 3 とコンテナ 4 6 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 4 6 0 1、アンカーアイコン 4 6 0 2 と固定された辺 4 6 0 5、辺 4 6 0 6 から成り立つ。各コンテナ 4 6 0 3 と 4 6 0 4 の間には可変サイズのリンク 4 6 0 9 があり、それぞれのコンテナを結んでいる。コンテナ 4 6 0 3 とコンテナ 4 6 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 4 6 0 7 と左辺 4 6 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 4 6 1 0、インジケータ 4 6 1 1 が表示され、これは辺 4 6 0 7 と辺 4 6 0 8 が可変であることを示している。

#### 【 0 0 8 7 】

また、図 4 8 はリンク 4 6 0 9 の情報がセットされているダイアログウィンドウ 4 8 0 1 の例を示す図である。このダイアログは典型的にタイトルバー 4 8 0 2、ツールボタン 4 8 0 3、ダイアログウィンドウの開閉を行うボタン 4 8 0 4、各種の情報をセットするエリア 4 8 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長 ( 4 8 0 7 ) のリンクであるか、あるいは固定長 ( 4 8 0 6 ) のリンクであるかの択一的な選択が行える。可変長の場合、リンクの長さの最大値 ( 4 8 1 2 ) そして最小値 ( 4 8 1 0 )、また基準値 ( 4 8 1 1 ) が設定できる。ここで各コンテナ間の距離の基準値 4 8 1 1 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。

#### 【 0 0 8 8 】

図 4 7 は自動レイアウトシステムにおける可変リンクを設定するフローチャートである。マウス 1 3 3 によりリンク 4 6 0 9 をクリックし、選択する ( S 4 7 0 2 )。レイアウト編集アプリケーション 1 2 1 は、マウスの右クリックあるいはキーボードの特定のキーにより選択されたリンク 4 6 0 9 のプロパティダイアログウィンドウ 4 8 0 1 を表示する ( S 4 7 0 3 )。この状態ではリンクサイズは可変ではなく固定であるため、固定 4 8 0 6 がリンクタイプ 4 8 0 5 において選択されている。リンクを固定サイズから可変サイズに変更するために、リンクタイプ 4 8 0 5 においてリンクサイズを可変に設定する可変 4 8 0 7 を選択する ( S 4 7 0 4 )。これによりリンク間隔 4 8 0 8 内に配置されている最大値 4 8 1 2、最小値 4 8 1 0、基準値 4 8 1 1 が有効化され、数値の設定が可能となる。ここで、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 4 8 1 2、最小値を 4 8 1 0、基準値を 4 8 1 1 において設定する ( S 4 7 0 5 )。一般的なダイアログウィンドウ開閉ボタン 4 8 0 4 によって設定を適用すると図 4 6 のリンク 4 6 0 9 のような状態にリンクの UI 表示が変化する。 ( S 4 7 0 6 ) このダイアログウィンドウ 4 8 0 1 の設定情報はメモリに格納される。

#### 【 0 0 8 9 】

図 4 9 は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。レイアウト編集アプリケーション 1 2 1 のレイアウトエンジンモジュール 1 0 5 におけ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。例えば図 4 6 においてコンテナ 4 6 0 3 とコンテナ 4 6 0 4 にそれぞれ違ったサイズのデータが挿入された時、それぞれのコンテナはデータの大きさを最適と考え、コンテナ 4 6 0 3 は挿入されたイメージサイズになる枠 4 9 0 4 (最適コンテナサイズ) に近づこうと右方向へ大きく、同様にコンテナ 4 6 0 4 も挿入されたイメージサイズになる枠 4 9 0 5 (最適コンテナサイズ) に近づこうと左方向へやや大きくサイズを変更しようとする。

#### 【 0 0 9 0 】

しかしコンテナ 4 6 0 3 とコンテナ 4 6 0 4 の間には固定サイズのリンク 4 9 0 3 が設定されているとすると、コンテナ 4 6 0 3 とコンテナ 4 6 0 4 はそれぞれアンカー 4 6 0 1 とアンカー 4 6 0 2 によってそれぞれ左辺 4 6 1 2 と右辺 4 6 1 3 は移動出来ないため、変更しようとするサイズがリンクサイズを上回ってしまう。そしてこのリンクサイズは固定されているためレイアウト計算時に優先的に計算されるためコンテナ 4 6 0 3 (図 4 6) とコンテナ 4 6 0 4 (図 4 6) のサイズが変更されることになる。その結果、コンテナ 4 6 0 3 とコンテナ 4 6 0 4 はデータ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 4 9 のコンテナ 4 9 0 1 とコンテナ 4 9 0 2 のように最適なサイズに比べて枠 4 9 0 4、枠 4 9 0 5 分小さくなってしまう。つまり、リンク 4 9 0 3 のサイズは固定であるためコンテナ 4 9 0 1 とコンテナ 4 9 0 2 は最適サイズを達成していない。

10

#### 【 0 0 9 1 】

図 5 0 は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 4 6 0 3 とコンテナ 4 6 0 4 の間には可変サイズのリンク 4 6 0 9 (図 4 6) が設定されているとすると、コンテナ 4 6 0 3 とコンテナ 4 6 0 4 のサイズが変更される際に、リンクサイズが縮まることでコンテナ 4 6 0 3 とコンテナ 4 6 0 4 のサイズが上記の例より大きくなることができ、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し、あるいはより挿入データサイズ (最適サイズ) に近づけてコンテナの枠を計算することが出来る。その結果が、図 5 0 のコンテナ 5 0 0 1 とコンテナ 5 0 0 2 である。可変リンク 4 6 0 9 はレイアウト計算の結果、可変リンク 5 0 0 3 のサイズとなり、この場合コンテナ 5 0 0 1 とコンテナ 5 0 0 2 は変形が連動して、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。以上の処理は、プロセッサユニット 1 3 5、レイアウト編集アプリケーションの制御のもとに実行することができる。

20

#### 【 0 0 9 2 】

(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、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。このプレビューモードにおいて、実際のレコードが挿入され、前述した優先順位に従って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 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。図 8 はレイアウト計算のフローを示している。

30

#### 【 0 0 9 3 】

まず、プレビューモードを選択する (S 8 0 1)。自動レイアウトシステムでは、コンテナを作成して、そのコンテナ間に関連付けを行い、レイアウトを作成するレイアウトモードと、作成したレイアウトに表示するデータレコード (以下、「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。このプレビューモードにおいて、実際のレコードが挿入され、レイアウトを計算する。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、レコードを挿入してレイアウトを計算する。その際の計算方法も同じである。プレビューモードになったら、プレビューするレコードを選択して挿入する (S 8 0 2)。レコードの挿入を行うと、そのレコードをレイアウトするためにレイアウト計算を行う (S 8 0 3)。そして、処理ステップ S 8 0 3 で計算されたレイアウトを表示し (S 8 0 4)、他のレコードについてもプレビューを行うかどうかを判断する (S 8 0 5)。処理ステップ 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と判断した場合は (S 8 0 5 - No)、処理をステップ S 8 0 7 に進め、プレビューモードを終了する。一方、ステップ S 8 0 5 の判断で、他のレコードについてプレビューを行うのであれば、他のレコードを選択して再度レイアウト計算を行い (S 8 0 3)、プレビュー

40

50

ーを行う (S 8 0 4)。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、処理 S 8 0 4 は印刷時の処理においては必要がないステップである。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。以上の処理は、プロセッサユニット 1 3 5 の制御のもとに実行することができる。

#### 【 0 0 9 4 】

(レイアウト計算方法)

図 9 はレイアウト計算の詳細な処理の流れを説明するフローチャートであり、図 1 0 はそのときのユーザインターフェース画面の表示例を示した図である。本図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。

#### 【 0 0 9 5 】

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求める (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。例えば、図 1 1 を参照すると、ページ上に 4 つのコンテナ A、B、C、D 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。そして、1 1 0 1 はアンカー、1 1 0 2 は固定された辺、1 1 0 3 はコントロール点、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 0 9 6 】

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の集合を選択する (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。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 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 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。また、コンテナ A がテキスト用コンテナである場合は、そのテキストデータも文字数と、コンテナ A のコンテナ属性で指定されているフォントタイプ、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。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 1 1 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 1 2 1 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、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 1 2 1 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

20

30

40

50

## 【 0 0 9 7 】

次に、レイアウト編集アプリケーション 1 2 1 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 ( S 9 0 3 )。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の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 1 2 1 は、ステップ S 9 0 2 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 1 5 0 6 (ここでは固定リンク) の合計サイズを求め、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 1 1 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、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 1 2 1 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。

10

## 【 0 0 9 8 】

レイアウト編集アプリケーション 1 2 1 は、レイアウトの最適化を行い、最適化の計算において、ルール (コンテナに課せられる制限) に違反する場合は、再度ルールを違反しないようにコンテナサイズの計算をする ( S 9 0 4 )。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、コンテナのサイズと位置、可変リンクの場合はリンクの長さなどである。ルールを違反しないようにレイアウトが計算されたら、ステップ S 9 0 2 で選択した集合のレイアウトは完成する。そして、ステップ S 9 0 2 ~ S 9 0 4 のステップをページ上のすべてのコンテナの集合について施し、ページ全体のレイアウトを計算し ( S 9 0 5 )、レイアウト計算の処理を終了する。

20

## 【 0 0 9 9 】

図 1 0 はレイアウト計算時に示されるユーザインターフェース画面を例示する図である。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と、図 1 0 の ( B ) の状態になる。図 1 0 の ( A ) の状態から可変の辺がどのように変位したかを 2 点鎖線で示している。そして、ルールを加味したレイアウト計算が行われる。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 1 0 1 2、1 0 1 3 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 (変形が連動するように) 計算され、且つ、前述したルールに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 ) で示した挿入するコンテンツサイズ ( a 1 b 1 c 1 d 1 , e 1 f 1 g 1 h 1 ) と計算後のコンテンツサイズ ( a 2 b 2 c 2 d 2 , e 2 f 2 g 2 h 2 ) は、左右双方のコンテナにおいて同等な差異をもってコンテナサイズが計算される。以上の処理は、プロセッサユニット 1 3 5、レイアウトエンジンモジュール 1 0 5 の制御のもとに実行することができる。

30

## 【 0 1 0 0 】

( コンテナの優先順位 )

40

( 設定方法 )

図 5 6 は、優先順位の設定処理の流れを説明するフローチャートであり、図 5 7 ~ 図 6 0 のユーザインターフェース画面の例とともに各ステップの処理を説明する。

## 【 0 1 0 1 】

まず、レイアウト編集アプリケーション 1 2 1 は、図 6 0 に示すように、ユーザに対して優先順位を設定するコンテナを選択するためのポップアップメニュー 6 0 0 2 を表示させ、優先順位設定が選択されると、優先順位設定画面を表示する ( S 5 6 0 1 )。図 6 0 において、6 0 0 1 はコンテナである、6 0 0 2 はポップアップメニュー、6 0 0 3 はマウスポインターを示している。このとき、図 6 0 では、優先順位の設定はポップアップメニューで選択されているが、コンテナのプロパティダイアログなどで選択するようにして

50

もよい。

#### 【0102】

次に、レイアウト編集アプリケーション121は、ステップS5601で選択したコンテナと関連付けされているコンテナを一つの集合として認識して(S5602)、優先順位を設定していく(S5603)。本自動レイアウトシステムは、関連付けされたコンテナ同士が、相互にコンテンツのサイズに合わせてレイアウトを計算して、最適化が行われる(レイアウト計算方法を参照)。優先順位は、関連し合うコンテナについて、計算する順番を指定していくものであり、関連付けされているコンテナを一つの集合として認識して、優先順位を設定していく。例えば、図57ではテキストコンテナA、D、FとイメージコンテナB、C、Eがレイアウトされている。リンク(5701)によって、コンテナが関連付けられており、コンテナA、B、C、DとE、Fの2つの集合になっている。ここで、マウスポインタ(5702)でコンテナAを選択して優先順位の設定を行うと、コンテナA、B、C、Dについて優先順位を設定することになる。

10

#### 【0103】

次に、レイアウト編集アプリケーション121は、ステップS5601で表示された優先順位設定画面(図58を参照)で、ユーザが設定した優先順位をコンテナに設定していく(S5604)。図58は優先順位を設定するダイアログユーザインターフェース(UI)画面の例を示す図である。同図において、5801はダイアログ、5802は優先順位を示しているUI、5803は優先順位を設定するコンテナが表示されるリストボックスである。ここには、ステップS5602で一つの集合と認識されたコンテナがすべて表示される。5804は優先順位を変更する矢印ボタン。5805は設定をキャンセルするキャンセルボタン、5806は設定を反映させるOKボタンである。5802のUIで示されるように、リストボックス(5803)で上に表示されているコンテナは優先順位が高く、下に行くにつれて優先順位が低くなる。計算されるレイアウト計算が行われる順番は上から順番になる。図58では、テキストコンテナA、イメージコンテナB、イメージコンテナC、テキストコンテナDの順に計算される。

20

#### 【0104】

図59は、コンテナに設定する優先順位を変更する手順を説明する図である。例えば、図59(A)の5901で示しているように、マウスポインターなどで優先順位を変更したいコンテナ名を選択し、優先順位を上げるのであれば、5902の上矢印ボタン、下げるのであれば、下矢印ボタンを押下する。ここでは、テキストコンテナDを選択して、優先順位を上げる操作を行ったものとする。すると、図59の(B)で示すように、テキストコンテナDの優先順位が一つ上がり、イメージコンテナCの優先順位が一つ下がる。優先順位の設定方法は、この方法に限らず、ここではコンテナA、B、C、Dと4つのコンテナについて優先順位を設定するので、各コンテナについて優先順位1番、2番と番号で指定する方法でもよい。以上の処理は、プロセッサユニット135の制御のもとに実行することができる。

30

#### 【0105】

(表示方法)

図61、図62、図63はそれぞれ設定したコンテナに設定した優先順位を視覚的にユーザに示すユーザインターフェース(UI)画面を例示する図である。図61、62はコンテナのレイアウト表示と優先順位の設定を別ウインドウで表示するUI画面を示している。図61、図62において、6101は自動レイアウトアプリケーションウインドウを示している。このウインドウを構成するページエリア、ツールバーなどは図3と同じであり、その他はここでは省略してある。6102はマウスポインタ、6103、6104は優先順位が設定されているコンテナの集合である。そして、6105は優先順位を表示するプロパティパレットであり、6106はコンテナ名が表示されているリストボックスである。図62においても同様に、6201はマウスポインタ、6202はリストボックスを示している。6105と6106および6202によって、コンテナの優先順位をユーザに視覚的に表示することができる。6106・6202には、優先順位の高い順に上か

40

50

らコンテナ名が表示される。図 6 1 で示しているように、マウスポインタ ( 6 1 0 2 ) をコンテナの集合 ( 6 1 0 3 ) 上に置くと、その集合内のコンテナ ( コンテナ A , B , C , D , E ) が、優先順位の高い順位表示される。ここでは、コンテナ B , A , E , C , D の順で優先順位が高いことを示している。図 6 2 で示すように、マウスポインタ 6 2 0 1 を別のコンテナの集合へ移動させると、リストボックス 6 2 0 2 の表示が更新され、コンテナ F , G , H が優先順位の高い順に表示される、ここではコンテナ F , H , G の順に優先順位が高い状態に設定されている。

#### 【 0 1 0 6 】

図 6 3 は、図 6 1 、図 6 2 の表示方法とは別の形式による U I 画面を例示する図であり、同図において、6 3 0 1 はページ領域を示しているページマージン、6 3 0 2 はマウスポインタ、6 3 0 3 はコンテナの優先順位を数字により示す表示部、6 3 0 4 は優先順位が設定してあるコンテナの集合である。図 6 3 で示している U I 画面は、マウスポインタ 6 3 0 2 をコンテナ上に移動させると、そのコンテナと関連付けられている各コンテナに設定されている優先順位が数字 6 3 0 3 で表示される。ここでは、マウスポインタをコンテナ B 上に移動させている、するとコンテナ B と関連付けられているコンテナ A , C , D , E に設定されている優先順位が各コンテナ上に表示される、数字が小さいほうが優先順位が高く、( 1 ) が一番優先順位が高く、( 5 ) が最も優先順位が低いコンテナを示す。従って、図 6 3 ではコンテナ B , A , E , D , C の順に優先順位が設定されている。以上の処理は、プロセッサユニット 1 3 5 、レイアウト編集アプリケーションの制御のもとに実行することができる。

#### 【 0 1 0 7 】

( レイアウト計算方法 )

次に、優先順位が設定されたコンテナが含まれる場合のレイアウトの計算処理を図 6 4 のレイアウト計算のフローチャートを参照しつつ説明する。

#### 【 0 1 0 8 】

まず、ステップ S 6 4 0 1 において、プレビューモードを選択する。自動レイアウトシステムでは、コンテナを作成して、そのコンテナ間に関連付けを行ってレイアウト作成するレイアウトモードと、作成したレイアウトに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 1 0 9 】

このプレビューモードにおいて、実際のレコードが挿入され、前述した優先順位に従って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が、実際に印刷する場合においても、レコードを挿入してレイアウトを計算する際の計算方法も同様である。ステップ S 6 4 0 1 の処理でプレビューモードが起動したら、プレビューするレコードがデータベース 1 1 9 に格納されているデータソースから選択されプレビューが実行される ( S 6 4 0 2 ) 。

#### 【 0 1 1 0 】

レコードの選択が行われプレビューが実行されると、レイアウト編集アプリケーション 1 2 1 は、そのレコードをレイアウトするために計算を行う。その際に前述した優先順位がコンテナに設定されているかをチェックする ( S 6 4 0 3 ) 。ステップ S 6 4 0 3 のチェックで、優先順位が設定されているコンテナが存在すると判断された場合 ( S 6 4 0 3 - Y E S ) 、 「 優先順位の設定あり 」 としてレイアウトを計算する ( S 6 4 0 4 ) 。この「優先順位あり」の場合のレイアウト計算方法については、後で詳細に記述する。

#### 【 0 1 1 1 】

ステップ S 6 4 0 3 の判断処理で優先順位が設定されているコンテナが存在しなかった場合 ( S 6 4 0 3 - N O ) 、処理をステップ S 6 4 0 5 に進めてレイアウト編集アプリケーション 1 2 1 は、 「 優先順位なし 」 としてレイアウトを計算する。

#### 【 0 1 1 2 】

そしてステップ S 6 4 0 6 において、レイアウト編集アプリケーション 1 2 1 は、優先順位ありとしてレイアウト計算された結果 ( S 6 4 0 4 ) 、または、優先順位なしとして

10

20

30

40

50

レイアウト計算された結果 ( S 6 4 0 5 ) をレイアウト表示する。そして、ステップ S 6 4 0 7 において、他のレコードについてもプレビューを行うかどうかを判断し、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 ( S 6 4 0 7 - N O ) 、プレビューモードを終了する ( S 6 4 0 9 ) 。他のレコードについてプレビューを行うのであれば ( S 6 4 0 7 - Y E S ) 、他のレコードを選択して再度レイアウト計算を行い ( S 6 4 0 4 、 S 6 4 0 5 ) 、プレビューを行う ( S 6 4 0 6 ) 。

【 0 1 1 3 】

プレビューモードでなく印刷時におけるレイアウト計算の場合は、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、レコードを移動させてレイアウト計算を行うためのステップ S 6 4 0 7 、 S 6 4 0 8 の処理は印刷時におけるレイアウト計算では不要となり、全てのレコードについて印刷が終了した時点で終了する。

【 0 1 1 4 】

( 優先順位が設定されたコンテナに対する具体的なレイアウト計算方法 )

レイアウトの優先順位がコンテナに設定されていた場合のレイアウト計算処理の流れを図 6 5 のフローチャートを参照しつつ説明する。また図 6 6 、 6 7 はそのレイアウト計算処理に従ってユーザに示される U I 画面の表示例を示す図である。

【 0 1 1 5 】

ここで、図 6 6 の 6 6 0 1 はコンテナ A を示し、 6 6 0 3 はコンテナ A に設定されている優先順位を示し、 6 6 0 2 はコンテナ B ( 6 6 0 4 ) とコンテナ C ( 6 6 0 5 ) の間に設定されているリンクを示している。なお、 6 6 0 3 で示される優先順位は数が小さいほうが高いものとする。ここでは優先順位 ( 1 ) 、 ( 2 ) 、 ( 3 ) に従い、コンテナ A 、 B 、 C の順にレイアウト計算がされる。図 6 7 の ( A ) の破線で示される領域 6 7 0 1 はコンテナ A に挿入されるコンテンツのサイズを示している。 6 7 0 2 はコンテナ A のレイアウト計算後のコンテナ A ' 、破線で示される領域 6 7 0 3 はコンテナ B に挿入されるコンテンツのサイズを示している。 6 7 0 4 はコンテナ B のレイアウト計算後のコンテナ B ' を示し、 6 6 0 5 はコンテナ C のレイアウト計算後のコンテナ C ' を示している。

【 0 1 1 6 】

まず、図 6 5 のステップ S 6 5 0 1 において、レイアウト計算するコンテナの集合を求める。そして計算を行う一つの集合を選択する ( S 6 5 0 2 ) 。この場合、コンテナの集合は各コンテナ間がリンク 6 6 0 2 により関連付けされている図 6 6 のコンテナ A 、 B 、 C が該当する。

【 0 1 1 7 】

そして、レイアウト編集アプリケーション 1 2 1 は、レイアウトするコンテンツデータをデータベース 1 1 9 から取得する ( S 6 5 0 3 ) 。ここで記述したコンテンツデータは、コンテンツの内容、サイズ、量などである。プロセッサユニット 1 3 5 は I / O インターフェース 1 3 8 を介してデータベース 1 1 9 より取得したコンテンツデータからコンテンツデータをそのコンテンツを挿入するコンテナに割り当て ( S 6 5 0 4 ) 。

【 0 1 1 8 】

次に、ステップ S 6 5 0 5 において、レイアウト編集アプリケーション 1 2 1 は、優先順位が一番高いコンテナを抽出する。図 6 7 の ( A ) では、コンテナ A の優先順位が一番高いので、コンテナ A が抽出される。そして、抽出されたコンテナ A について、ステップ 6 5 0 4 で割り当てられたコンテンツデータに基づいてレイアウト計算を行う ( S 6 5 0 6 ) 。ここで、図 6 7 ( A ) の 6 7 0 1 はコンテナ A に割り当てられたコンテンツデータのサイズに対応する領域を示す。計算の方法は、現在計算の対象になっているコンテナ A より優先順位が低いコンテナ ( B 、 C ) にルールが違反しない限り影響を与えることが可能である。その代わりに、レイアウト計算の対象となるコンテナより優先順位が高いコンテナ ( 例えば、コンテナ B から見て優先順位の高いコンテナ A ) には影響を与えることができない。図 6 7 ( A ) の場合、コンテナ A のレイアウト計算では、コンテナ A より優先順位の低いコンテナ B 、 C のコンテナサイズに関係なく、計算をすることが可能である。コンテナ B については、コンテナ A のレイアウト計算結果の影響を受けることになるが、コ

10

20

30

40

50

ンテナBのレイアウト計算結果は、優先順位の低いコンテナCに影響を与えることが可能である。最も優先順位の低いコンテナCは、コンテナA、Bのレイアウト計算結果の影響を受けつつ、自己のコンテナに対するレイアウト計算が実行されることになる。

【0119】

次に処理をステップS6507に進め、レイアウト編集アプリケーション121は、ステップ6506で計算されたレイアウト計算の結果より、コンテナのサイズと位置を決定する。そして、ステップS6507で決定したコンテナのサイズと位置を確定する(S6508)。図67(B)の6702は、コンテナAについて、計算が終了してサイズ・位置が確定された後のコンテナを示している。

【0120】

ステップS6509では、レイアウト編集アプリケーション121は、選択された集合内の全てのコンテナA、B、Cについて計算が終了したかどうかを確認する。全てのコンテナA、B、Cについて終了していなければ(S6509-NO)、計算したコンテナの次に優先度の高いコンテナを抽出し(S6510)、そのコンテナについてステップS6506～S6508の処理を実行する。図67の(B)の状態では、コンテナAの計算が終了したあと、次に優先順位が高いコンテナBについてコンテナサイズの計算、位置の決定等の処理が実行される。図67の(B)において、6703はコンテナBに割り当てられたコンテンツデータのサイズ(コンテンツサイズ)に対応する領域を示す。このコンテンツサイズに基づいてコンテナBに対するレイアウト計算が行われるが、コンテナBの優先順位はコンテナAより優先順位が低いため、計算結果は、すでに前ステップでコンテナAのサイズと位置は確定されているために、コンテナAのレイアウト結果に影響を与えることができないことになる。

【0121】

したがって、コンテナBのレイアウト計算はコンテナAに影響を与えない範囲でコンテナサイズを計算しなければならないという拘束条件が課せられることになる。図67(B)の6703で示すように、実際コンテナBに挿入されるコンテンツのサイズはコンテナA'と重なってしまっている。そのため、計算結果が確定したコンテナA'に重複しないようにコンテナBのレイアウトは実際のサイズより縮小して計算されることになる。コンテナBについて計算が終了し、サイズと位置が確定したコンテナB'が図67(C)の6704に示される。

【0122】

コンテナBについてレイアウト計算が終了した後、次にコンテナBより優先度が低いコンテナCが抽出され(図65のステップS6510)、コンテナCについてレイアウト計算が行われる(S6506～S6508)。しかし、図67の例では、コンテナCは集合内で一番優先順位が低いコンテナであり、そのコンテナについて計算するときは、その他全ての優先順位の高いコンテナ(A、B)について、コンテナサイズと位置が確定されているため、コンテナCのレイアウト計算結果を他のコンテナ(A、B)に与えるようにレイアウト計算をすることができない。したがって、一番優先順位が低いコンテナがステップ6510で抽出されたときには、すでにそのコンテナのサイズ・位置も確定されていることになる。一番優先順位が低いコンテナについてもステップ6506～6508で計算を行うが、他の優先順位の高いコンテナの計算結果に従いに実質的には確定してしまっていることになる。そして、集合内全てのコンテナについて、レイアウト計算が終了したら(S6509-YES)、ページ内にまだ計算していない集合があるかを確認する(6511)。まだ計算していない集合がある場合(S6511-NO)はステップ6502へ処理を戻し、レイアウト編集アプリケーション121は、前述した方法でレイアウト計算を行う。全ての集合について計算が終了しているのであれば、レイアウト編集アプリケーション121は、データベースからパリアブルデータのコンテンツを読み込み、計算されたコンテナに該コンテンツを流し込み(S6512)、処理を終了する。以上の処理は、プロセッサユニット135、レイアウト編集アプリケーション(ユーザインターフェースモジュール103、レイアウトエンジンモジュール105)の制御のもとに実行すること

10

20

30

40

50



ができる。

#### 【 0 1 2 3 】

( リンクの自動設定 )

次に、本発明の特徴であるリンクの自動設定に関する内容について説明する。図 1 2 の ( A ) はリンクの自動設定に関する U I 画面を例示する図である。1 2 0 1 はアプリケーションウインドウを示し、1 2 0 2 はコンテナ、1 2 0 3 はコンテナ間に設定されているリンク、1 2 0 4 はマウスポインタを示している。また、本実施形態でコンテナの位置について記述があるが、ドキュメント内左上を原点 1 2 0 5 として、X 方向は右側へ、Y 方向は下側へ増加していくことと定義する ( 図 1 2 の ( B ) を参照 )。コンテナの位置情報はそれぞれコンテナの各コーナー部の位置座標 ( ( X 1、Y 1 )、( X 2、Y 2 ) ( X 3、Y 3 ) 等 ) により記述することができ、ドキュメントテンプレート 3 0 9 にレイアウトされている複数のコンテナ間の位置関係は、基準となるコンテナの位置座標情報と、他のコンテナとの位置座標情報と、を求めることにより相対的な位置関係を求めることが可能になる。

10

#### 【 0 1 2 4 】

図 1 6 はリンクの自動設定の全体的な流れを説明するフローチャートである。まず、ステップ S 1 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、リンク設定を一括に行うコンテナが指定されたかを判断する。コンテナの設定は、ユーザによって任意に指定することが可能であるが、何も指定しなければ ( S 1 6 0 1 - N O )、ページ上の全コンテナが選択されたことになる ( S 1 6 0 3 )。一方、レイアウト編集アプリケーション 1 2 1 は、複数のコンテナが指定されたと判断した場合 ( S 1 6 0 1 - Y E S )、処理をステップ S 1 6 0 2 に進め、プロセッサユニット 1 3 5 はユーザによって選択された複数のコンテナを認識する ( S 1 6 0 3 )。

20

#### 【 0 1 2 5 】

そして、プロセッサユニット 1 3 5 及びユーザインターフェースモジュール 1 0 3 による制御の下、認識したコンテナをユーザに視覚的に示すために U I 画面を変更する ( 1 6 0 4 )。図 1 3 は複数のコンテナ F、G、H を選択する U I 画面の例を示す図である。マウスポインタ 1 3 0 3 等の指示入力手段によって、矩形 1 3 0 2 をドラッグして矩形 1 3 0 2 内に複数コンテナを入れることによって、矩形内のコンテナを複数個選択することが可能である。また、複数個のコンテナの選択はこの方法に限らず、選択しようとするコンテナを個別にマストポインタ 1 3 0 3 によりドラッグして指定するようにしてもよい。

30

#### 【 0 1 2 6 】

図 1 4 A は、図 1 3 で説明した操作によりコンテナ F、G、H が選択されたことをユーザに示すために U I 画面が切り替えられた状態を示す図である。図 1 4 A では、コンテナ F、G、H に対して、選択されたことを示す表示 ( 図 1 4 A の例ではハッチングを付して示している ) 1 4 0 2 が表示される。

#### 【 0 1 2 7 】

図 1 4 B は、図 1 4 A のように矩形領域によりコンテナを複数指定するのではなく、マウスポインタ 1 3 0 3 でコンテナを押下する ( 指定する ) ことによる複数のコンテナの選択の例を示している。この方法では、図 1 4 A で説明したように矩形領域による選択するでは選択することができないコンテナの配置 ( 部分的に選択しないコンテナが含まれるような配置 ) のコンテナを複数選択することができる。

40

#### 【 0 1 2 8 】

選択されたコンテナは図 1 4 A の場合と同様に選択されたことを示す表示 5 1 0 2 が表示される。

#### 【 0 1 2 9 】

説明を図 1 6 に戻し、ステップ S 1 6 0 5 において、レイアウト編集アプリケーション 1 2 1 は、選択された複数のコンテナ間にリンクを一括に設定する。このステップでは、リンクを一括で設定したことによって、レイアウトがリンクに設定されているルールに違反しないようにコンテナの設定を変更するステップも含まれるものとする。複数のコンテ

50

ナが選択された状態で、図 1 4 A に示すようなリンクツールボタン 1 4 0 3 をマウスポインタで押下することによって、図 7 3 のようなリンク作成方向 UI が表示される。本実施例では一括にリンクを設定する設定方向モードとして、以下の 3 つを含む。まず、第 1 リンク設定方向モードは 7 3 0 2 の「X 方向にリンク設定」であり、ユーザによって選択されたことを、レイアウト編集アプリケーション 1 2 1 が認識した場合、X 方向にリンクが一括設定されることとなる。また、第 2 リンク設定方向モードは 7 3 0 3 の「Y 方向にリンク設定」であり、ユーザによって 7 3 0 3 が選択されたことを認識した場合、Y 方向にリンクが一括設定される。第 3 リンク設定方法モードは 7 3 0 4 の「X および Y 方向にリンク設定」であり、ユーザによって 7 3 0 4 が選択されたことを認識した場合、X および Y 方向にリンクを一括設定することとなる。

10

#### 【 0 1 3 0 】

リンクの一括設定は、ダイアログを別途表示して設定の有無をユーザに確認させるようにしてもよい。その場合は、ユーザの確認の入力があったときにリンクの一括設定が行われる。図 1 5 は複数選択されたコンテナ F , G , H 間にリンクが一括で設定された状態を示す図である。リンクの一括設定により、コンテナ F , G , H の間にはリンク 1 5 0 2 が設定される。このリンク設定のための具体的な内容は、後の詳述する。つぎに、本処理は、バリエブルデータを読み込んで、各コンテナに流し込み、自動レイアウトした結果をプレビューするプレビュー処理中にレイアウトの編集が行われている場合は、設定されたリンク 1 5 0 2 の状態を考慮してレイアウト計算を行う ( S 1 6 0 6 )。このレイアウト計算については、図 9 , 1 0 で説明した内容と重複するため説明は省略する。また、単に基本レイアウトの編集集中であり、プレビュー表示や印刷要求を受けていない場合は、ステップ S 1 6 0 6 の処理は行われない。

20

#### 【 0 1 3 1 】

( リンクの一括作成 )

図 1 7 はリンクの一括作成の処理の流れを概略的に説明するフローチャートである。以後の説明では、矩形領域で複数のコンテナが選択された場合を例として処理の内容を説明するが、ただし、ページ内の全コンテナを選択した場合でも、マウスポインタで個別に複数コンテナを選択した場合も同じ手法で処理を行うことは可能である。

#### 【 0 1 3 2 】

まず、ステップ S 1 7 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、図 1 6 のステップ S 1 6 0 2 で認識しているユーザにより選択されたコンテナの位置情報及びコンテナの個数 ( ConNum ) を取得する。次に、レイアウト編集アプリケーション 1 2 1 は、選択されたコンテナの個数を記憶装置 1 3 9 ( 1 4 0 、 1 4 1 ) またはメモリユニット 1 3 6 に格納する ( S 1 7 0 2 )。そして、レイアウト編集アプリケーション 1 2 1 は、選択されたコンテナに対して、X 方向に隣接する位置にコンテナが配置されている場合は X 方向のリンクの作成を行う ( S 1 7 0 3 )。次に、Y 方向に隣接する位置にコンテナが配置されている場合は、レイアウト編集アプリケーション 1 2 1 は、Y 方向のリンクの作成を行う ( S 1 7 0 4 )。最後に、レイアウト編集アプリケーション 1 2 1 は、作成されたリンクによって、コンテナのレイアウトに関するルールが違反しないように、コンテナの設定 ( アンカーの設定等 ) を自動変更する ( 1 7 0 5 )。

30

40

#### 【 0 1 3 3 】

以上の各ステップ ( S 1 7 0 1 ~ S 1 7 0 5 ) の具体的な処理を、以下図 1 8 等の図面を参照しつつ説明する。尚、リンクの一括作成に関する処理は、レイアウト編集アプリケーション 1 2 1 に基づきプロセッサユニット 1 3 5 の全体的な制御の下、実行することが可能である。

#### 【 0 1 3 4 】

( コンテナの位置情報取得 )

図 1 8 はコンテナの位置情報を取得する処理の流れを説明するフローチャートである。まず、レイアウト編集アプリケーション 1 2 1 は、選択したコンテナに任意に順番を設定する ( S 1 8 0 1 )。この処理は、全てのコンテナについて位置情報を取得するために、

50

順番を設定するものである。次に、レイアウト編集アプリケーション 121 は、変数  $i$  に 1 を代入して変数の初期化を行う (S 1802)。この変数  $i$  はコンテナに設定された番号と対応し、この変数  $i$  により各コンテナが対応付けられることになる。そして、 $i$  番目のコンテナについて、プロセッサユニット 135 は、まず、X 方向について選択されているコンテナの位置情報を取得し (S 1803)、次に、Y 方向について位置情報を取得する (S 1804)。そしてプロセッサユニット 135 は、取得した情報を記憶装置 139 (140、141) またはメモリユニット 136 に格納する (S 1805)。次に、レイアウト編集アプリケーション 121 は、変数  $i$  をインクリメントして、次のコンテナの X 方向の位置情報 (S 1803)、Y 方向の位置情報を取得して (S 1804)、データを格納し (S 1805)、選択した全てのコンテナについてデータの取得と格納が終了するまで、以上の処理を繰り返す。選択した全てのコンテナに対して、処理が終了した場合 (S 1807-Y E S)、処理を終了する。

10

#### 【0135】

(リンクの作成)

図 19A, B は X 方向のリンクの作成処理の流れを説明するフローチャートである。リンクの作成は、隣接するコンテナを抽出し、そのコンテナ間にリンクを設定する。ここで隣接するコンテナは、コンテナ間の距離が最小であることを意味しており、最小であるコンテナが複数見つかった場合は、リンクを複数設定する。

#### 【0136】

まず、ステップ S 1901 で、レイアウト編集アプリケーション 121 は、リンク設定の基準となるコンテナを設定し (この場合はコンテナ 2104)、このコンテナ 2104 の左辺の X 座標を基準として、コンテナ間の相対的な位置関係を求め、X 方向の間隔の小さい順にコンテナの番号をリンク作成用に設定する。この順番の設定は、図 21A の 2160 に示すように、コンテナ 2104 の左辺の X 座標を基準として、各コンテナの X 方向の距離の小さい ( $X_1 < X_2$ )、コンテナ 2105 に順番 (2) が設定され、この次に X 方向の距離の小さいコンテナである 2106 に順番 (3) が設定される。以下、同様に選択された全てのコンテナに対して、X 方向の距離に従い、順番が設定される。この場合、同じ X 方向の位置に複数のコンテナがレイアウトされる場合は、Y 方向の距離の関係を求め、Y 方向の距離が小さいコンテナを早い順番に設定する。図 21A の 2150 は、コンテナに対して、リンク設定のための順番付けが完了した状態を示す図である。

20

30

#### 【0137】

なお、本実施例ではコンテナの左辺が左上にあるコンテナに対して 1 番目と順番が設定されるが、これに限ることはなく、例えば右上、右下方向などから順番を設定するようにしても構わない。

次に、処理をステップ S 1902 に進め、レイアウト編集アプリケーション 121 は、変数  $n$  に 1 を代入する。ステップ S 1903 で、レイアウト編集アプリケーション 121 は、X 方向に隣接するコンテナの番号 (ステップ 1901 で設定した順番) を格納する  $ContX[]$  を用意し、この値を 0 で初期化をする。

#### 【0138】

次に、ステップ S 1904 において、レイアウト編集アプリケーション 121 は、X 方向の距離の最小値を格納する  $DistX\_Min$  を用意し、初期値にページ幅を代入する (1904)。

40

#### 【0139】

そして、ステップ S 1905 において、レイアウト編集アプリケーション 121 は、隣接するコンテナをチェックするための変数  $i$  に  $n + 1$  を代入する。これは、ステップ S 1901 で X 方向の距離が小さい順に順番を設定しているため、隣接しており、かつ、リンクを設定する必要があるコンテナは、自身コンテナ (図 21A の場合は、コンテナ 2104 が対応する。) より順番が後のものになる。したがって、ステップ S 1905 の処理は、チェックするコンテナを  $n + 1$  番目からになるように変数を制御するためである。

#### 【0140】

50

次に、処理を図 19 B のフローチャートのステップ S 1 9 0 6 に移行して、n 番目のコンテナと i 番目のコンテナの Y 方向の位置が重なるかどうかをチェックする。本実施形態におけるリンクの一括設定では、作成されるリンクは、コンテナ同士が同じ水平方向または垂直方向にあるということを前提にしている。従って、X 方向のリンクについてはステップ 1 9 0 6 のチェックを行い、コンテナ同士が水平方向にレイアウトされているかを判断する。例えば、図 2 1 A の 2 1 6 0 において、コンテナ 2 1 0 4 と、コンテナ 2 1 0 5 との Y 方向の距離は Y 1 であり、コンテナ 2 1 0 4 と、コンテナ 2 1 0 6 の Y 方向の距離は Y 2 である。

【 0 1 4 1 】

ステップ S 1 9 0 6 の判断で、Y 方向の位置が重なりと判断された場合 ( S 1 9 0 6 - Y E S )、処理をステップ S 1 9 0 7 に進め、レイアウト編集アプリケーション 1 2 1 は、n 番目のコンテナと i 番目のコンテナの X 方向の位置の差を計算して、結果を D i s t X に代入する。

【 0 1 4 2 】

次に、処理をステップ S 1 9 0 8 に進め、レイアウト編集アプリケーション 1 2 1 は、前ステップで計算された D i s t X と、最小の距離が格納されている D i s t X \_ M i n とを比較する。もし、D i s t X が最小値 ( D i s t X \_ M i n ) よりも小さいのであれば ( S 1 9 0 8 - Y E S )、処理をステップ S 1 9 0 9 に進め、レイアウト編集アプリケーション 1 2 1 は、C o n t X [ ] を 0 で初期化し、変数 k に 0 を代入する。

【 0 1 4 3 】

そして、距離が最小であったコンテナの番号を C o n t X [ k ] に代入する ( 1 9 1 0 ) 。

【 0 1 4 4 】

一方、ステップ 1 9 0 8 で D i s t X が最小値よりは小さくなかった場合 ( S 1 9 0 8 - N O )、処理をステップ S 1 9 1 1 に進め、レイアウト編集アプリケーション 1 2 1 は、さらに比較を行い D i s t X と D i s t X \_ M i n が一致するかどうかを判断する ( S 1 9 1 1 ) 。もし一致するのであれば ( S 1 9 1 1 - Y E S )、そのコンテナ番号を C o n t X [ k ] に代入して、k に 1 を加算する。

【 0 1 4 5 】

ステップ 1 9 1 0 を処理した後、あるいは、S 1 9 1 1 で D i s t X が最小値よりも大きいと判断された場合や、ステップ 1 9 0 6 で i 番目のコンテナが Y 方向に重なっていない場合は、レイアウト編集アプリケーション 1 2 1 は、次のコンテナをチェックするために変数 i をインクリメントする ( S 1 9 1 2 ) 。

【 0 1 4 6 】

ステップ S 1 9 1 3 では、レイアウト編集アプリケーション 1 2 1 は、選択されているコンテナすべてについてチェックを行ったかどうかを判断し、まだ行っていない場合は、ステップ S 1 9 0 6 へ進み処理を繰り返す。全てについてチェックが終了した場合は、検出された隣接するコンテナへリンクを設定する処理を行う ( S 1 9 1 4 ) 。この処理については後述する。

【 0 1 4 7 】

そして、リンク設定の処理が終了したら、変数 n をインクリメントし ( S 1 9 1 5 )、選択された全てのコンテナについて、これまでの処理が行われたかどうかを判断し、まだ行われていない場合は、ステップ S 1 9 0 3 に進み処理を繰り返す ( S 1 9 1 6 ) 。全てについて処理が終了していれば処理を終了する。

【 0 1 4 8 】

図 2 0 は、図 1 9 B のステップ 1 9 1 4 のリンク設定処理の流れを説明するフローチャートである。まず、ステップ S 2 0 0 1 において、C o n t X [ 0 ] が 0 であるかどうかを判断する。もし、0 であるならば ( S 2 0 0 1 - Y E S )、隣接したコンテナが検出されなかったことを意味するので、リンクの設定処理は行わずに終了する。

【 0 1 4 9 】

10

20

30

40

50

一方、ContX[0]が0でなかった場合は(S2001-NO)、kに0を代入する(S2002)。そして、n番目のコンテナとContX[k]番目のコンテナ間にX方向のリンクを設定する。リンクを設定したら、kをインクリメントして(S2004)、ContX[k]をチェックする(2005)。ContX[k]が0になるまで、格納されている全てのコンテナについてリンクの設定を行う。ContX[k]が0になったら、リンクの設定処理を終了する。

図21A、BはX方向のリンク設定を説明する図である。図21Aにおいて、2101、2102は選択外のコンテナを示し、2103は複数のコンテナを指定する矩形領域、2104～2111はリンクの一括処理を行うように選択されたコンテナを示し、2112はマウスポインタを示している。コンテナ内の番号は、図19のステップ1901でコンテナに順番を付けられた番号である。番号はX方向の位置の小さい順に設定されており、2104、2105、2106、2107、2108、2109、2110、2111の順に番号が設定されている。ここで、リンク設定ツールボタン2113を押下すると、リンクが設定される。図21Bの2114～2117は設定されたX方向のリンクを示しており、各コンテナのX方向に対して最小距離にあるコンテナ間にリンクが設定されている。

#### 【0150】

なお、図21A、および図21Bのフローチャートにおいては各コンテナのX方向に対して最小距離にあるコンテナ間にリンクを設定することを記載したが、最小距離に限らない場合の処理の流れについて図68に記載する。なお、基本的な処理の流れは上記した図16とほぼ同様であるので異なる点のみを記載する。

#### 【0151】

まず、図68におけるS6801～S6804は同様の処理であるため説明を省略する。レイアウト編集アプリケーション121は、マウス等により、ある特定の指示がされた場合(例えば右クリックなど)、図69に示すようなリンク設定UIをディスプレイ装置144に表示する(S6805)。図69はリンク設定モードを選択するためのUI図であり、リンク設定UI6901には図示するように第1設定モードとしての「基準コンテナと最短コンテナにリンク設定」と、第2設定モードとしての「基準コンテナとXまたはY方向に交わるコンテナに設定」とが記載されており、ユーザがどちらかのリンク設定方法を選択したことを認識することにより、次のステップへと進む。「基準コンテナと最短コンテナにリンク設定」は上述したリンク設定方法のことを指すので、「基準コンテナとXまたはY方向に交わるコンテナに設定」が選択された場合の処理について記載する。ここで、「基準コンテナとXまたはY方向に交わるコンテナに設定」について、図72を用いて具体的に説明する。上述した「基準コンテナと最短コンテナにリンク設定」の場合、図72においてコンテナ7201乃至7203がリンク設定対象であり、コンテナ7201を基準としてリンクを生成する場合、コンテナ7203はコンテナ7201との最短距離に位置していないためリンク7205が設定されることがなかったが、「基準コンテナとXまたはY方向に交わるコンテナに設定」が選択された場合、コンテナ7201とY方向で交わるコンテナ7202に対してリンク7204が設定され、7203に対してリンク7205が設定されることとなる。

#### 【0152】

レイアウト編集アプリケーション121は、図69におけるリンク設定UIにおいて設定方法を選択し、「OK」ボタンが押されたことを認識すると、該選択されたコンテナ間にリンクを設定する(S6806)。なお、S6806～S6809に関しても図16におけるS1605～S1606の処理と同様であるためここでは説明を省略する。リンクの一括作成およびコンテナの位置情報取得についての詳細は、上述した図17および図18の処理と同様であるためここでは省略する。

#### 【0153】

続いて、「基準コンテナとXまたはY方向に交わるコンテナに設定」におけるリンク作成フローを図70Aおよび図70Bを用いて説明する。

## 【 0 1 5 4 】

なお、S 7 0 0 1 ~ S 7 0 0 5 は、S 1 9 0 1 ~ S 1 9 0 3 および S 1 9 0 5 ~ S 1 9 0 6 と同様の処理であるため詳細については省略する。

## 【 0 1 5 5 】

レイアウト編集アプリケーションは、S 7 0 0 5 において、i 番目のコンテナは n 番目のコンテナの Y 方向の位置と重なると判定した場合、n 番目のコンテナと i 番目のコンテナの間に他のコンテナがあるか否かを判定する (S 7 0 0 6)。詳細には、図 1 8 の処理により各コンテナの位置情報を記憶装置 1 3 9 またはメモリユニット 1 3 6 に格納しているので、レイアウト編集アプリケーション 1 2 1 が参照することにより、判定可能である。

10

## 【 0 1 5 6 】

S 7 0 0 6 の判定の結果、n 番目のコンテナと i 番目のコンテナの間に他のコンテナがないと判定した場合、n 番目のコンテナとリンクを用いて関連付けるために S 7 0 0 8 へ進む。しかし、S 7 0 0 6 の判定の結果、間に他のコンテナがあると判定した場合、レイアウト編集アプリケーション 1 2 1 は、レイアウト編集アプリケーション 1 2 1 が i 番目のコンテナと n 番目のコンテナを直接的にリンクを設定できるか否かを判定する (S 7 0 0 7)。詳細には、各コンテナサイズの情報図 1 8 の処理により各コンテナの位置情報を記憶装置 1 3 9 またはメモリユニット 1 3 6 に格納しているので、レイアウト編集アプリケーション 1 2 1 が参照することにより、判定可能である。

## 【 0 1 5 7 】

S 7 0 0 7 は、n 番目のコンテナと i 番目のコンテナが他のコンテナを介することなく直接的にリンクを設定することができる場合に、S 7 0 0 8 へ進むことを指し、具体的には図 7 1 において 5 番目のコンテナと 8 番目のコンテナは Y 方向の位置が重なると判定することができるが、5 番目のコンテナと 8 番目のコンテナの間に 7 番目のコンテナがあり、5 番目のコンテナから 8 番目のコンテナへは 7 番目のコンテナを介して、間接的にしかリンクを設定することができない。このような場合、5 番目のコンテナから 8 番目のコンテナに対してリンクは設定されない。また、4 番目のコンテナと 8 番目のコンテナは Y 方向の位置が重なっており、間には 7 番目のコンテナがある。しかし、8 番目のコンテナの Y 方向は 7 番目のコンテナに対して垂直方向にはみ出しており、4 番目のコンテナから 8 番目のコンテナに対して、間にある 7 番目のコンテナを介することなく直接的にリンクを設定することができるためリンク 7 1 0 2 が生成されることとなる。

20

30

## 【 0 1 5 8 】

なお、S 7 0 0 8 から S 7 0 1 3 の処理は図 1 9 における S 1 9 1 0 ~ S 1 9 1 3 の処理と同様であるため詳細については省略し、図 7 0 B の S 7 0 1 1 の処理については図 1 9 B の S 1 9 1 4 を説明した図 2 0 の処理の流れと同様であるため、ここでは詳細について省略する。

## 【 0 1 5 9 】

図 7 1 は、X 方向のリンク設定を説明する図である。なお、図 7 1 は図 2 1 B と基本的に同じであるため異なる点のみを記載する。図 2 1 B では 5 番目のコンテナと X 方向の最短距離に位置する 6 番目のコンテナにのみリンク 2 1 1 6 が設定されていたが、図 7 1 では、5 番目のコンテナと 7 番目のコンテナは X 方向の最短距離に位置されていないが、リンク 7 1 0 1 が設定されている。これは、図 6 9 のリンク設定 UI により「基準コンテナと X または Y 方向に交わるコンテナに設定」を選択したため、7 番目のコンテナは基準コンテナ (5 番目のコンテナ) の Y 方向と交わっていることによりリンク 7 1 0 1 が設定されることとなる。さらに、図 2 1 B では 4 番目のコンテナと 8 番目のコンテナは X 方向の最短距離に位置していないため、リンクが設定されなかったが、図 7 1 では、リンク 7 2 0 2 が設定されている。これは、両者の間には 7 番目のコンテナが存在するが、8 番目のコンテナは基準となるコンテナ (4 番目のコンテナ) と Y 方向で交わり、さらに 7 番目のコンテナの Y 方向の位置を Y 方向にはみ出しているため、4 番目のコンテナから 8 番目のコンテナに対して直接的にリンクを設定することが可能である。よってリンク 7 1 0 2 が

40

50

設定されることとなる。

#### 【 0 1 6 0 】

図 2 2 A , B は Y 方向のリンクの作成フローを示している。各ステップについては、方向が変化するのみで、X 方向の処理 ( 図 1 9 A 、 B ) とほぼ同じ処理である。

#### 【 0 1 6 1 】

まず、レイアウト編集アプリケーション 1 2 1 は、順番が設定されたコンテナを Y 方向の位置について小さい順番に、Y 方向のリンクを作成するための順番を設定する ( S 2 2 0 1 )。このとき、同じ Y 方向の位置のコンテナがあった場合は、X 方向の距離が小さいコンテナを早い順番に設定する。次に、n に 1 を代入する ( S 2 2 0 2 )。そして、レイアウト編集アプリケーション 1 2 1 は、Y 方向に隣接するコンテナの番号 ( ステップ S 2 2 0 1 で設定した順番 ) を格納する `Cont Y [ ]` を用意し、これを 0 で初期化をする ( S 2 2 0 3 )。また、Y 方向の距離の最小値を格納する `Dist Y _ Min` を用意し、初期値にページ幅を代入する ( S 2 2 0 4 )。次に、隣接するコンテナをチェックするための変数 i に n + 1 を代入する ( 2 2 0 5 )。

10

#### 【 0 1 6 2 】

次に、処理を図 2 2 B のステップ S 2 2 0 6 に移行して、レイアウト編集アプリケーション 1 2 1 は、n 番目のコンテナと i 番目のコンテナの X 方向の位置が重なるかどうかをチェックする ( S 2 2 0 6 )。ステップ 2 2 0 6 で重なりと判断された場合 ( S 2 2 0 6 - Y E S )、レイアウト編集アプリケーション 1 2 1 は、n 番目のコンテナと i 番目のコンテナの Y 方向の位置の差を計算して、その結果を `Dist Y` に代入する ( S 2 2 0 7 )。

20

#### 【 0 1 6 3 】

そして、レイアウト編集アプリケーション 1 2 1 は、前ステップで計算された `Dist Y` と最小の距離が格納されている `Dist Y _ Min` とを比較する ( S 2 2 0 8 )。もし、`Dist Y` が最小値よりも小さいのであれば、処理をステップ S 2 2 0 9 に進め、レイアウト編集アプリケーション 1 2 1 は、`Cont Y [ ]` を 0 で初期化し、変数 k に 0 を代入する。そして、距離が最小であったコンテナの番号を `Cont Y [ k ]` に代入する ( 2 2 1 0 )。

#### 【 0 1 6 4 】

一方、ステップ S 2 2 0 8 で `Dist Y` が最小値よりは小さくなかった場合 ( S 2 2 0 8 - N O )、処理をステップ S 2 2 1 1 に進め、レイアウト編集アプリケーション 1 2 1 は、さらに比較を行い `Dist Y` と `Dist Y _ Min` が一致するかどうかを判断する ( S 2 2 1 1 )。もし一致するのであれば ( S 2 2 1 1 - Y E S )、レイアウト編集アプリケーション 1 2 1 は、そのコンテナ番号を `Cont Y [ k ]` に代入して、k に 1 を加算する ( S 2 2 1 0 )。

30

#### 【 0 1 6 5 】

ステップ 2 2 1 0 を処理した後、あるいは、ステップ 2 2 1 1 で `Dist Y` が最小値よりも大きいと判断された場合、ステップ 2 2 0 6 で i 番目のコンテナが X 方向に重なっていない場合は、レイアウト編集アプリケーション 1 2 1 は、次のコンテナをチェックするために i をインクリメントする ( S 2 2 1 2 )。ステップ S 2 2 1 3 において、レイアウト編集アプリケーション 1 2 1 は、選択されているコンテナすべてについてチェックを行ったかどうかを判断し、まだ全てのコンテナに対してチェックを行っていない場合は、ステップ S 2 2 0 6 へ進み処理を繰り返す。全てのコンテナに対してチェックが終了した場合は、レイアウト編集アプリケーション 1 2 1 は、検出された隣接するコンテナへリンクを設定する処理を行う ( S 2 2 1 4 )。この処理については後述する。

40

#### 【 0 1 6 6 】

そして、Y 方向のリンク設定の処理が終了したら、ステップ S 2 2 1 5 において、変数 n をインクリメントする。そして、選択された全てのコンテナについて、これまでの処理が行われたかどうかを判断し、まだ、行われていなければ、ステップ S 2 2 0 3 に進み処理を繰り返す ( S 2 2 1 6 )。全てについて処理が終了していれば全体の処理を終了する。

50

## 【 0 1 6 7 】

図 2 3 は、図 2 2 B のステップ S 2 2 1 4 における Y 方向のリンクを設定する処理の流れを説明するフローチャートである。まず、ステップ S 2 3 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、Cont Y[0] が 0 であるかどうかを判断する。もし 0 であるならば ( S 2 3 0 1 - Y E S )、隣接したコンテナが検出されなかったことを意味するので、リンクの設定処理は行わずに終了する。一方、0 でなかった場合は ( S 2 3 0 1 - N O )、レイアウト編集アプリケーション 1 2 1 は、変数 k に 0 を代入する ( S 2 3 0 2 )。そして、n 番目のコンテナと Cont Y[k] 番目のコンテナ間に Y 方向のリンクを設定する。リンクを設定したら、レイアウト編集アプリケーション 1 2 1 は、変数 k をインクリメントして ( S 2 3 0 4 )、Cont Y[k] をチェックし、Cont Y[k] が 0 になるまで、格納されている全てのコンテナについてリンクの設定を行う ( S 2 3 0 3 )。Cont Y[k] が 0 になったら ( S 2 3 0 5 - Y E S )、リンクの設定処理を終了する。

10

## 【 0 1 6 8 】

図 2 4 A、B はリンク作成の状態を示す UI 画面の例を示す図である。図 2 4 A において、2 4 0 1 は複数のコンテナの選択を指定する矩形領域、2 4 0 2 ~ 2 4 0 9 は選択されたコンテナ、2 1 1 0 はマウスポインタを示している。コンテナ内の番号は、図 2 2 のステップ S 2 2 0 1 でコンテナに順番を付けた番号である。( 1 ) のコンテナからみて、Y 方向の距離の小さい順に他のコンテナ ( ( 2 ) ~ ( 8 ) ) の順番は設定されており、2 4 0 2、2 4 0 3、2 4 0 4、2 4 0 5、2 4 0 6、2 4 0 7、2 4 0 8、2 4 0 9 の順に番号が設定されている。ここで、図 2 4 B の 2 4 1 5 で示すように、リンク設定ツールボタンを押下すると、リンクが設定される。2 4 1 1 ~ 2 1 1 3 は作成された Y 方向のリンクを示しており、各コンテナの Y 方向に対して最小距離にあるコンテナ間にリンクが設定されている。また、2 1 1 3 のように、コンテナ ( 1 ) からコンテナ ( 4 ) コンテナ ( 5 ) 5 間へ 2 本のリンクが設定されているが、これは Y 方向の最小距離に 2 つのコンテナがあったためである。

20

## 【 0 1 6 9 】

なお、Y 方向にリンクを一括設定する場合にも、上述した X 方向にリンクを一括設定する場合と同様に、図 6 9 のリンク設定 UI による選択により、最小距離に位置しないコンテナに対してもリンクを設定することが可能となる。

30

## 【 0 1 7 0 】

( コンテナの計算のための優先順位の設定 )

コンテナに対してリンクを一括設定する際、コンテナの計算の順番を指定するための優先方向の設定をすることが可能である。この処理を図 5 1、図 5 2、図 5 3 を参照しつつ説明する。ここで、図 5 2 はリンクの一括設定により作成されたリンクとそのコンテナを示す図である。左右に示される 5 3 0 1 はページ幅を示し、5 3 0 2 はリンクの一括設定が行われたコンテナ、5 3 0 3 はリンクを示している。設定される優先順位は、X 方向と Y 方向どちらかの方向を指定して、指定された方向から、コンテナに優先順位が設定される。方向の指定は、図 5 3 のようなダイアログで行われ、ユーザによって指定される。5 3 0 4 はダイアログボックスであり、ユーザインターフェースモジュール 1 0 3 により表示制御される。5 3 0 5 は方向指定を行うラジオボタン、5 3 0 6 は設定をキャンセルするボタン、5 3 0 7 は OK ボタンを示している。方向を指定して、OK ボタンを押下することで優先順位が自動に設定される。

40

## 【 0 1 7 1 】

図 5 1 は優先順位の設定フローの流れを説明するフローチャートである。まず、ステップ S 5 1 0 1 において、前述した図 5 3 のダイアログにより指定された優先方向を判断する。ここで、X 方向が指定されていたならば、レイアウト編集アプリケーション 1 2 1 は、コンテナの左辺の X 座標の位置について、小さい順にコンテナの順番を設定する ( S 5 1 0 2 )。もし、同じ X 座標に複数のコンテナがレイアウトされていた場合、レイアウト編集アプリケーション 1 2 1 は、それらのコンテナの上辺の Y 座標が小さい順に順番を設

50



定する。

#### 【 0 1 7 2 】

一方、優先方向に Y 方向が指定されていたならば、レイアウト編集アプリケーション 1 2 1 は、コンテナの上辺の Y 座標について、小さい順にコンテナの順番を設定する ( S 5 1 0 3 )。もし、同じ Y 座標に複数のコンテナがレイアウトされていた場合、レイアウト編集アプリケーション 1 2 1 は、それらのコンテナの上辺の X 座標が小さい順に順番を設定する。順番の設定が終了したら、その順番に沿って優先順位を設定する ( S 5 1 0 4 )。この優先方向の指定により、コンテナに関する計算の優先順位を設定することができる。

#### 【 0 1 7 3 】

図 5 4 A、B は、X 方向を優先させてコンテナに優先順位を設定する処理の例を示す図であり、図 5 4 A は優先順位が設定される前のコンテナとリンクを示している。ここで、左右の 5 4 0 1 はページ幅を示し、5 4 0 2 はコンテナを示している。そして、5 4 0 3 は全コンテナの左辺の X 座標を示している。コンテナの優先順位は、X 座標の小さい ( ページ上の左から ) 順番に設定される。図 5 4 B は優先順位が設定された後のコンテナとリンクを示している。各コンテナ内の数字が優先順位を示している。X 座標の小さい順に設定されるので、5 4 0 4、5 4 0 5、5 4 0 6、5 4 0 7、5 4 0 8、5 4 0 9、5 4 1 0、5 4 1 1 の順にコンテナの優先順位が設定される。

#### 【 0 1 7 4 】

同様に図 5 5 A、B は、Y 方向を優先させてコンテナに優先順位を設定する処理の例を示す図であり、図 5 5 A は優先順位が設定される前のコンテナとリンクを示している。ここで、左右の 5 5 0 1 はページ幅を示し、5 5 0 2 はコンテナを示している。そして、5 5 0 4 は全コンテナの上辺の Y 座標を示している。コンテナの優先順位は、Y 座標の小さい ( ページ上の上から ) 順番に設定される。図 5 5 B は優先順位が設定された後のコンテナとリンクを示している。各コンテナ内の数字が優先順位を示している。Y 座標の小さい順に設定されるので、5 5 0 4、5 5 0 5、5 5 0 6、5 5 0 7、5 5 0 8、5 5 0 9、5 5 1 0、5 5 1 1 の順に優先順位が設定される。

#### 【 0 1 7 5 】

( リンク設定の修正 )

一括設定されたコンテナ間のリンクに関して、ユーザは、例えば、キーボード 1 3 2、またはマウス 1 3 3 の指示入力手段を介して、修正の指定をすることができる。この場合、指定されたリンクに関して、リンクの属性 ( 可変、固定 ) の変更や、リンクの設定をキャンセルすることが可能である。この変更や設定のキャンセルはユーザインターフェースモジュール 1 0 3、プロセッサユニット 1 3 5 の制御の下に実行することが可能である。また、コンテナをそれぞれ指定することにより、個別にリンクの設定を指定することができる。これにより、ユーザは、一括設定されたリンクの他、新たに追加してリンクの設定を行うことが可能になる。

#### 【 0 1 7 6 】

( コンテナに関する設定の自動変更 ( ページ上の一番外側にある辺をロック ) )

図 1 7 のステップ 1 7 0 5 にあたる、コンテナのアンカーの設定を変更する処理について説明する。ここで行う処理は、リンクを作成したコンテナの辺が、ページ内にレイアウトされているコンテナの最も外側に位置する辺と同じ位置であったときは、その辺の位置を自動的にロック ( 固定 ) するものである。同じコンテナで横方向の辺とこれに交差する縦方向の辺がロックされた場合は、自動的にその辺により形成されるコーナー部にアンカーが設定される。

#### 【 0 1 7 7 】

図 2 5 は、アンカーの設定に関する全体的な処理の流れを説明するフローチャートである。ステップ S 2 5 0 1 において、コンテナの左辺をロックして、ステップ S 2 5 0 2 において、コンテナの右辺をロックする。更に、ステップ S 2 5 0 3、ステップ S 2 5 0 4 において、コンテナの上辺、下辺をロックする。各辺をロックする詳細な処理の流れは図

10

20

30

40

50

26、図28、図30、図32のフローチャートに基づき説明する。

【0178】

(左辺のロック)

図26は、図25におけるステップS2501のコンテナの左辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップS2601において、レイアウト編集アプリケーション121は、レイアウトされている全コンテナの左辺の最小X座標を取得する。そして、取得した最小X座標をMin\_Xに代入する(S2602)。次に、ステップS2603において、変数iに1を代入し、ステップ2604において、変数kに0を代入する。

【0179】

さらに、レイアウト編集アプリケーション121は、Lock\_Cont[]を用意し、0で初期化をする(S2605)。ここで、変数iはコンテナの番号を示しており、変数kはLock\_Cont[]の添字を示す変数であり、Lock\_Cont[]は、ロック対象のコンテナの番号を格納する配列である。

【0180】

次に、ステップS2606において、レイアウト編集アプリケーション121は、i番目のコンテナの左辺のX座標とMin\_Xが等しいかどうかをチェックする。等しいのであれば、処理をステップS2607に進め、iをLock\_Cont[k]に代入する。そして、kおよびiをインクリメントする(S2608, S2609)。

【0181】

ステップ2606で等しくなかった場合は(S2606-NO)、Lock\_Contに代入せずに、iをインクリメントする(S2609)。そして、すべてのコンテナについて処理が終了したかどうかを確認し(S2610)、終了していなければ、ステップS2606へ進み処理を繰り返す。終了していれば、処理をステップS2611に進め、kに0を代入する。そして、処理をステップS2612に進め、Lock\_Cont[k]の値をチェックする。もし、値が0ならば、ロックを設定せずにこの処理を終了する。値が0でないならば、Lock\_Cont[k]番目のコンテナの左辺をロックする(S2613)。そして、変数kをインクリメントする(S2614)。この処理をLock\_Cont[k]の値が0になるまで、ステップ2613と2614を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内に一番左にある辺がロックされる。複数の辺が見つかった場合は、見つかった複数の辺すべてがロックされる。

【0182】

図27はコンテナの左辺のロック状態を示す図であり、2701は複数のコンテナの選択を指定する矩形、2702, 2703, 2704はリンク、2705はロックされた辺を示し、2706は最小のX座標を示している。この図で示されるように、図26の処理を行うと、最小のX座標にあるコンテナの左辺がロックされる。

【0183】

図28は図25におけるステップ2502の右辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップS2801において、レイアウト編集アプリケーション121は、レイアウトされている全コンテナの右辺の最大X座標を取得する。そして、ステップS2802において、取得した最大X座標をMax\_Xに代入する。次に、ステップS2803において、変数iに1を代入し、また、ステップS2804において、変数kに0を代入する。さらに、ステップS2805において、Lock\_Cont[]を用意し、0で初期化をする。ここで、変数iはコンテナの番号を示しており、kはLock\_Cont[]の添字を示す変数であり、Lock\_Cont[]は、ロック対象のコンテナの番号を格納する配列である。

【0184】

次に、ステップS2806でレイアウト編集アプリケーション121は、i番目のコンテナの右辺のX座標とMax\_Xが等しいかどうかをチェックする。等しいのであれば(S2806-YES)、処理をステップS2807に進め、iをLock\_Cont[k]

10

20

30

40

50

に代入する。そして、変数  $k$  および変数  $i$  をインクリメントする (S 2 8 0 8 , S 2 8 0 9)。ステップ S 2 8 0 6 で等しくなかった場合は (S 2 8 0 6 - NO)、Lock\_\_Cont に代入せずに、処理をステップ S 2 8 0 9 に進め、変数  $i$  をインクリメントする。  
【0185】

そして、すべてのコンテナについて終了したかどうかを確認し (S 2 8 1 0)、終了していなければ (S 2 8 1 0 - NO)、ステップ S 2 8 0 6 へ進み処理を繰り返す。終了していれば (S 2 8 1 0 - YES)、処理をステップ S 2 8 1 1 に進め、変数  $k$  に 0 を代入する。

【0186】

そして、処理をステップ S 2 8 1 2 に進め、レイアウト編集アプリケーション 1 2 1 は、Lock\_\_Cont[ $k$ ] の値をチェックする。もし、値が 0 ならば、ロックを設定せずにこの処理を終了する。一方、値が 0 でないならば (S 2 8 1 2 - NO)、処理をステップ S 2 8 1 3 に進め Lock\_\_Cont[ $k$ ] 番目のコンテナの右辺をロックする。

【0187】

そして、ステップ S 2 8 1 4 において、変数  $k$  をインクリメントし、Lock\_\_Cont[ $k$ ] の値が 0 になるまで、ステップ S 2 8 1 3 とステップ S 2 8 1 4 の処理を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内に一番右にある辺がロックされる。複数の辺が見つかった場合は、見つかった複数の辺すべてがロックされる。

【0188】

図 2 9 はコンテナの右辺のロック状態を示す図である。同図において、2 9 0 1 は複数のコンテナの選択を指定する矩形、2 9 0 2 , 2 9 0 3 はリンク、2 9 0 4 はロック、2 9 0 5 は最大の X 座標を示している。図 2 9 で示されるように、図 2 8 の処理を行うと、最大の X 座標にあるコンテナの右辺がロックされる。

【0189】

(上辺のロック)

図 3 0 は図 2 5 におけるステップ 2 5 0 3 の上辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップ S 3 0 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、レイアウトされている全コンテナの上辺の最小 Y 座標を取得する。そして、ステップ S 3 0 0 2 において、取得した最小 Y 座標を Min\_\_Y に代入する。次に、ステップ S 3 0 0 3 において、変数  $i$  に 1 を代入し、ステップ S 3 0 0 4 において、変数  $k$  に 0 を代入する。さらに、ステップ S 3 0 0 5 において、レイアウト編集アプリケーション 1 2 1 は、Lock\_\_Cont[] を用意し、0 で初期化をする。ここで、変数  $i$  はコンテナの番号を示しており、 $k$  は Lock\_\_Cont[] の添字を示す変数であり、Lock\_\_Cont[] は、ロック対象のコンテナの番号を格納する配列である。

【0190】

次に、処理をステップ S 3 0 0 6 に進め、レイアウト編集アプリケーション 1 2 1 は、 $i$  番目のコンテナの上辺の Y 座標と Min\_\_Y が等しいかどうかをチェックする。等しいのであれば (S 3 0 0 6 - YES)、処理をステップ S 3 0 0 7 に進め、変数  $i$  を Lock\_\_Cont[ $k$ ] に代入する。そして、変数  $k$  および変数  $i$  をインクリメントする (S 3 0 0 8 , S 3 0 0 9)。

【0191】

ステップ 3 0 0 6 で等しくなかった場合は (S 3 0 0 6 - NO)、レイアウト編集アプリケーション 1 2 1 は、Lock\_\_Cont に代入せずに、変数  $i$  をインクリメントする (S 3 0 0 9)。そして、処理をステップ S 3 0 1 0 に進め、すべてのコンテナについて処理が終了したかどうかを確認し、終了していなければ (S 3 0 1 0 - NO)、ステップ 3 0 0 6 へ進み処理を繰り返す。処理が終了していれば (S 3 0 1 0 - YES)、処理をステップ S 3 0 1 1 に進め、変数  $k$  に 0 を代入する。そして、処理をステップ S 3 0 1 2 に進め、レイアウト編集アプリケーション 1 2 1 は、Lock\_\_Cont[ $k$ ] の値をチェックする。もし、値が 0 ならば (S 3 0 1 2 - YES)、ロックを設定せずにこの処理を

10

20

30

40

50

終了する。値が0でないならば(S 3 0 1 2 - NO)、処理をステップS 3 0 1 3に進め、Lock\_\_Cont[k]番目のコンテナの上辺をロックする。そして、処理をステップS 3 0 1 4に進め、変数kをインクリメントし(3 0 1 4)、Lock\_\_Cont[k]の値が0になるまで、ステップS 3 0 1 3とS 3 0 1 4を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内に一番上にある辺がロックされる。複数の辺が見つかった場合は、見つかった複数の辺すべてがロックされる。

#### 【0 1 9 2】

図3 1はコンテナの上辺のロック状態を示す図である。同図において、3 1 0 1は複数のコンテナの選択を指定する矩形、3 1 0 2, 3 1 0 3はリンク、3 1 0 4は最小のY座標を示している。この図で示されるように、図3 0の処理を行うと、最小のY座標にあるコンテナの上辺がロックされる。

10

#### 【0 1 9 3】

図3 2は図2 5におけるステップ2 5 0 4の下辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップS 3 2 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、レイアウトされている全コンテナの下辺の最大Y座標を取得する。そして、ステップS 3 2 0 2において、取得した最大Y座標をMax\_\_Yに代入する。次に、ステップS 3 2 0 3において、変数iに1を代入し、ステップS 3 2 0 4において、変数kに0を代入する。さらに、ステップS 3 2 0 5において、Lock\_\_Cont[]を用意し、0で初期化をする。ここで、iはコンテナの番号を示しており、kはLock\_\_Cont[]の添字を示す変数であり、Lock\_\_Cont[]は、ロック対象のコンテナの番号を格納する配列である。

20

#### 【0 1 9 4】

次に、処理をステップS 3 2 0 6に進め、レイアウト編集アプリケーション1 2 1は、i番目のコンテナの下辺のY座標とMax\_\_Yが等しいかどうかをチェックする。等しいのであれば、処理をステップS 3 2 0 7に進め、変数iをLock\_\_Cont[k]に代入する。そして、ステップS 3 2 0 8において、変数kおよび変数iをインクリメントする(S 3 2 0 8, S 3 2 0 9)。

#### 【0 1 9 5】

ステップS 3 2 0 6で等しくなかった場合は(S 3 2 0 6 - NO)、Lock\_\_Contに代入せずに、変数iをインクリメントする(S 3 2 0 9)。そして、すべてのコンテナについて処理を終了したかどうかを確認し(S 3 2 1 0)、終了していなければ(S 3 2 1 0 - NO)、ステップ3 2 0 6へ進み処理を繰り返す。終了していれば(S 3 2 1 0 - YES)、処理をステップS 3 2 1 1に進め、変数kに0を代入する。そして、処理をステップS 3 2 1 2に進め、レイアウト編集アプリケーション1 2 1は、Lock\_\_Cont[k]の値をチェックする。もし、値が0ならば、リンクを設定せずにこの処理を終了する。値が0でないならば(S 3 2 1 2 - NO)、Lock\_\_Cont[k]番目のコンテナの下辺をロックする(S 3 2 1 3)。そして、処理をステップS 3 2 1 4に進め、変数kをインクリメントし、Lock\_\_Cont[k]の値が0になるまで、ステップS 3 2 1 3とS 3 2 1 4を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内に一番下にある辺がロックされる。複数の辺が見つかった場合は、見つかった複数の辺すべてがロックされる。

30

40

#### 【0 1 9 6】

図3 3はコンテナの下辺のロック状態を示す図である。同図において、3 3 0 1は複数のコンテナの選択を指定する矩形、3 3 0 2, 3 3 0 3はリンク、3 3 0 4は最大のY座標を示している。この図で示されるように、図3 2の処理を行うと、最大のY座標にあるコンテナの下辺がロックされる。

#### 【0 1 9 7】

図3 4はアンカーの設定の状態を示す図である。ここで、3 4 0 1はページ端を示し、3 4 0 2はコンテナ、3 4 0 3, 3 4 0 4はリンク、3 4 0 5, 3 5 0 6, 3 4 0 7はロックされた辺を示している。上述の処理でページ内の外側に位置する辺3 4 0 7がロック

50

されたときに、同じコンテナの縦と横の辺がロックされて、コンテナの角が固定になったときは、3408で示されるようにアンカーが自動的に設定される。

#### 【0198】

(フレキシブルなリンクの設定)

図35, 36は、上述のリンクの一括作成処理で、作成されるリンクが固定長さではなく、最大・最小値をもつフレキシブルなリンクとして設定するための処理の流れを説明するフローチャートである。ここで、リンクを設定するコンテナを選択する操作は上述の内容と同様なので重複を避けるためここでは説明を省略するが、フレキシブルなリンクの最大、最小値の設定処理は、図20及び図23のフローが変更になることになるので、ここではその変更点のみを説明する。

10

#### 【0199】

図35は、図20で示したX方向のリンク作成のフローチャートにおいて、作成されるリンク長をフレキシブルに設定するための処理の流れを説明するフローチャートである。まず、ステップS3501において、レイアウト編集アプリケーション121は、変数kに0を代入する。そして、ステップS3502において、ContX[k]の値をチェックする。もし値が0であるならば(S3502-YES)、リンク作成処理は終了する。一方、値が0でなければ(S3502-NO)、処理をステップS3503に進め、n番目のコンテナと、ContX[k]番目のコンテナの間にX方向のリンクを設定する。そして、ステップS3504において、作成されたリンクの長さをLink\_Distに代入する。ステップS3505では、Link\_Distに係数1.2を乗算し、その値を作成したリンクの最大値に設定する。そして、Link\_Distに係数0.8を乗算し、その値を作成したリンクの最小値に設定する(S3506)。ステップS3507において、変数kをインクリメントして、リンクを設定するコンテナがある限り、ステップS3503からS3506までの処理を続ける。

20

#### 【0200】

同様に図36は、図23で示したY方向のリンク作成のフローチャートにおいて、作成されるリンク長をフレキシブルに設定するための処理の流れを説明するフローチャートである。まず、ステップS3601において、変数kに0を代入する。そして、ステップS3602で、ContY[k]の値をチェックする。もし値が0であるならば、リンク作成処理は終了する。値が0でなければ(S3602-NO)、処理をステップS3603に進め、n番目のコンテナと、ContY[k]番目のコンテナの間にY方向のリンクを設定する。そして、ステップS3604において、作成されたリンクの長さをLink\_Distに代入する。ステップS3605では、Link\_Distに係数1.2を乗算し、その値を作成したリンクの最大値に設定する。そして、Link\_Distに係数0.8を乗算し、その値を作成したリンクの最小値に設定する(S3606)。ステップS3607において、変数kをインクリメントして、リンクを設定するコンテナがある限り、ステップS3603からS3606までの処理を続ける。

30

#### 【0201】

以上説明した図35、図36の処理により、作成されたリンクの長さの120%を最大値、80%を最小値として設定することが可能になる。ただし、リンク長の最大、最小値は上述の係数に限定されるものではなく、例えば、図48のUI画面を介してユーザによって任意に指定することができるものとする。

40

#### 【0202】

(コンテナに関する設定の自動変更(ユーザの指定した範囲にある辺をロック))

次に、コンテナに関する設定の自動変更として、ユーザから入力された範囲にある辺をロックする処理について説明する。

#### 【0203】

図37は、本ロック処理の全体的な処理の流れを説明するフローチャートであり、フローを示している。まず、ステップS3701において、レイアウト編集アプリケーション121は、ユーザによるポインティングデバイス(132、133)の入力により指定さ

50

れたX, Y方向の範囲を、指定範囲情報として保持する変数「V a l u e \_ X , V a l u e \_ Y」に代入する。ここでの範囲とは、ページ内の一番外側の辺の座標から、どのくらい内側までをロックの対象とするかを定める値である。そして、ステップS 3 7 0 2 ~ S 3 7 0 5の処理によりコンテナの各辺をロックする。以下、各辺のロックについて具体的に説明する。

#### 【 0 2 0 4 】

( 左辺のロック )

図 3 8 は、図 3 7 におけるステップS 3 7 0 2のコンテナの左辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップS 3 8 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、レイアウトされている全コンテナの左辺の最小X座標を取得する。そして、取得した最小X座標をM i n \_ Xに代入する( S 3 8 0 2 )。次に、ステップS 3 8 0 3において、変数iに1を代入し、ステップS 3 8 0 4において、変数kに0を代入する。

#### 【 0 2 0 5 】

さらに、レイアウト編集アプリケーション1 2 1は、L o c k \_ C o n t [ ]を用意し、0で初期化をする( S 3 8 0 5 )。ここで、変数iはコンテナの番号を示しており、変数kはL o c k \_ C o n t [ ]の添字を示す変数であり、L o c k \_ C o n t [ ]は、ロック対象のコンテナの番号を格納する配列である。

#### 【 0 2 0 6 】

次に、ステップS 3 8 0 6において、レイアウト編集アプリケーション1 2 1は、i番目のコンテナの左辺のX座標からM i n \_ Xを減算して、その値が指定範囲情報のX座標であるV a l u e \_ X以下であるかをチェックする。V a l u e \_ X以下であれば( S 3 8 0 6 - Y E S )、i番目のコンテナは、ロック対象として指定された範囲内であるので、変数iをL o c k \_ C o n t [ k ]に代入する( S 3 8 0 7 )。そして、変数kおよび変数iをインクリメントする( S 3 8 0 8 , S 3 8 0 9 )。

#### 【 0 2 0 7 】

ステップ3 8 0 6でV a l u e \_ X以下でない場合は( S 3 8 0 6 - N O )、L o c k \_ C o n t に代入せずに、変数iをインクリメントする( S 3 8 0 9 )。そして、すべてのコンテナについて処理が終了したかどうかを確認し( S 3 8 1 0 )、終了していなければ( S 3 8 1 0 - N O )、ステップS 3 8 0 6へ進み処理を繰り返す。終了していれば( S 3 8 1 0 - Y E S )、処理をステップS 3 8 1 1に進め、変数kに0を代入する。そして、処理をステップS 3 8 1 2に進め、L o c k \_ C o n t [ k ]の値をチェックする。もし、値が0ならば、ロックを設定せずにこの処理を終了する。値が0でないならば( S 3 8 1 2 - N O )、処理をステップS 3 8 1 3に進め、L o c k \_ C o n t [ k ]番目のコンテナの左辺をロックする。そして、処理をステップS 3 8 1 4に進め、変数kをインクリメントし、L o c k \_ C o n t [ k ]の値が0になるまで、ステップS 3 8 1 3とS 3 8 1 4の処理を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内に一番左からユーザが指定した範囲内にあるコンテナの左辺はすべてロックされる。

#### 【 0 2 0 8 】

図 3 9 はコンテナの左辺のロック状態を示す図であり、3 9 0 1は複数のコンテナの選択を指定する矩形、3 9 0 2 , 3 9 0 3 , 3 9 0 4はリンク、3 9 0 5はロックされたコンテナの辺、3 9 0 6は最小のX座標、3 9 0 7はロック対象となる指定範囲の境界線、3 9 0 8はV a l u e \_ Xの間隔を示している。この図で示されるように、図 3 8の処理を行うと、最小のX座標3 9 0 6から、V a l u e \_ X 3 9 0 8で指定された範囲内にあるコンテナの左辺がロックされる( 3 9 0 5 )。この場合、複数の辺が該当する場合は、該当する複数の辺すべてがロックされる。

#### 【 0 2 0 9 】

( 右辺のロック )

図 4 0 は、図 3 7 におけるステップS 3 7 0 3のコンテナの右辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップS 4 0 0 1において、レイ

10

20

30

40

50

アウト編集アプリケーション 121 は、レイアウトされている全コンテナの右辺の最大 X 座標を取得する。そして、取得した最大 X 座標を `Max__X` に代入する (S 4 0 0 2)。次に、ステップ S 4 0 0 3 において、変数 `i` に 1 を代入し、ステップ S 4 0 0 4 において、変数 `k` に 0 を代入する。

#### 【0210】

さらに、`Lock__Cont[]` を用意し、0 で初期化をする (S 4 0 0 5)。ここで、変数 `i` はコンテナの番号を示しており、変数 `k` は `Lock__Cont[]` の添字を示す変数であり、`Lock__Cont[]` は、ロック対象のコンテナの番号を格納する配列である。

次に、ステップ S 4 0 0 6 において、`Max__X` から `i` 番目のコンテナの右辺の X 座標を減算して、その値が指定範囲情報の X 座標である `Value__X` 以下であるかどうかをチェックする。`Value__X` 以下であれば (S 4 0 0 6 - YES)、`i` 番目のコンテナは、ロック対象として指定された範囲内であるので、変数 `i` を `Lock__Cont[k]` に代入する (S 4 0 0 7)。そして、変数 `k` および変数 `i` をインクリメントする (S 4 0 0 8, S 4 0 0 9)。

#### 【0211】

ステップ S 4 0 0 6 で `Value__X` 以下でない場合は (S 4 0 0 6 - NO)、`Lock__Cont` に代入せずに、変数 `i` をインクリメントする (S 3 8 0 9)。そして、すべてのコンテナについて処理が終了したかどうかを確認し (S 4 0 1 0)、終了していなければ (S 4 0 1 0 - NO)、ステップ S 4 0 0 6 へ進み処理を繰り返す。終了していれば (S 4 0 1 0 - YES)、処理をステップ S 4 0 1 1 に進め、変数 `k` に 0 を代入する。そして、処理をステップ S 4 0 1 2 に進め、`Lock__Cont[k]` の値をチェックする。もし、値が 0 ならば、ロックを設定せずにこの処理を終了する。値が 0 でないならば (S 4 0 1 2 - NO)、処理をステップ S 4 0 1 3 に進め、`Lock__Cont[k]` 番目のコンテナの右辺をロックする。そして、処理をステップ S 4 0 1 4 に進め、変数 `k` をインクリメントし、`Lock__Cont[k]` の値が 0 になるまで、ステップ S 4 0 1 3 と S 4 0 1 4 の処理を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内に一番右からユーザが指定した範囲内にあるコンテナの右辺はすべてロックされる。

#### 【0212】

図 4 1 はコンテナの左辺のロック状態を示す図であり、4 1 0 1 は複数のコンテナの選択を指定する矩形、4 1 0 2, 4 1 0 3, 4 1 0 4 はリンク、4 1 0 5 はロックされたコンテナの辺、4 1 0 7 は最大の X 座標、4 1 0 6 はロック対象となる指定範囲の境界線、4 1 0 8 は `Value__X` の間隔を示している。この図で示されるように、図 4 0 の処理を行うと、最大の X 座標から、`Value__X` 4 1 0 8 で指定された範囲内にあるコンテナの右辺がロックされる (4 1 0 5)。この場合、複数の辺が該当する場合は、該当する複数の辺すべてがロックされる。

#### 【0213】

(上辺のロック)

図 4 2 は、図 3 7 におけるステップ S 3 7 0 4 のコンテナの上辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップ S 4 2 0 1 において、レイアウトされている全コンテナの上辺の最小 Y 座標を取得する。そして、取得した最小 Y 座標を `Min__Y` に代入する (S 4 2 0 2)。次に、ステップ S 4 0 0 3 において、変数 `i` に 1 を代入し、ステップ S 4 2 0 4 において、変数 `k` に 0 を代入する。

#### 【0214】

さらに、`Lock__Cont[]` を用意し、0 で初期化をする (S 4 2 0 5)。ここで、変数 `i` はコンテナの番号を示しており、変数 `k` は `Lock__Cont[]` の添字を示す変数であり、`Lock__Cont[]` は、ロック対象のコンテナの番号を格納する配列である。

次に、ステップ S 4 2 0 6 において、`i` 番目のコンテナの上辺の Y 座標から `Min__Y` を減算して、その値が `Value__Y` 以下かどうかをチェックする。`Value__Y` 以下であれば (S 4 2 0 6 - YES)、`i` 番目のコンテナは、ロック対象として指定された範囲内であるので、変数 `i` を `Lock__Cont[k]` に代入する (S 4 2 0 7)。そして、

変数  $k$  および変数  $i$  をインクリメントする (S 4 2 0 8 , S 4 2 0 9 )。

【 0 2 1 5 】

ステップ S 4 2 0 6 で  $Value\_Y$  以下でない場合は (S 4 2 0 6 - NO)、 $Lock\_Cont$  に代入せずに、変数  $i$  をインクリメントする (S 4 2 0 9)。そして、すべてのコンテナについて処理が終了したかどうかを確認し (S 4 2 1 0)、終了していなければ (S 4 2 1 0 - NO)、ステップ S 4 2 0 6 へ進み処理を繰り返す。終了していれば (S 4 2 1 0 - YES)、処理をステップ S 4 2 1 1 に進め、変数  $k$  に 0 を代入する。そして、処理をステップ S 4 2 1 2 に進め、 $Lock\_Cont[k]$  の値をチェックする。もし、値が 0 ならば、ロックを設定せずにこの処理を終了する。値が 0 でないならば (S 4 2 1 2 - NO)、処理をステップ S 4 2 1 3 に進め、 $Lock\_Cont[k]$  番目のコンテナの上辺をロックする。そして、処理をステップ S 4 2 1 4 に進め、変数  $k$  をインクリメントし、 $Lock\_Cont[k]$  の値が 0 になるまで、ステップ S 4 2 1 3 と S 4 2 1 4 の処理を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内の一番上からユーザが指定した範囲内にあるコンテナの上辺はすべてロックされる。

10

【 0 2 1 6 】

図 4 3 はコンテナの上辺のロック状態を示す図であり、4 3 0 1 は複数のコンテナの選択を指定する矩形、4 3 0 2 , 4 3 0 3 はリンク、4 3 0 4 は最小の  $Y$  座標、4 3 0 5 はロック対象となる指定範囲の境界線、4 3 0 8 は  $Value\_Y$  の間隔を示している。この図で示されるように、図 4 2 の処理を行うと、最小  $Y$  座標から、 $Value\_Y$  4 3 0 8 で指定された範囲内にあるコンテナの上辺がロックされる (4 3 0 6)。この場合、複数の辺が該当する場合は、該当する複数の辺すべてがロックされる。

20

【 0 2 1 7 】

(下辺のロック)

図 4 4 は、図 3 7 におけるステップ S 3 7 0 5 のコンテナの下辺をロックする処理の流れを具体的に説明するフローチャートである。まず、ステップ S 4 4 0 1 において、レイアウトされている全コンテナの下辺の最大  $Y$  座標を取得する。そして、取得した最大  $Y$  座標を  $Max\_Y$  に代入する (S 4 4 0 2)。次に、ステップ S 4 4 0 3 において、変数  $i$  に 1 を代入し、ステップ S 4 4 0 4 において、変数  $k$  に 0 を代入する。

【 0 2 1 8 】

さらに、 $Lock\_Cont[]$  を用意し、0 で初期化をする (S 4 4 0 5)。ここで、変数  $i$  はコンテナの番号を示しており、変数  $k$  は  $Lock\_Cont[]$  の添字を示す変数であり、 $Lock\_Cont[]$  は、ロック対象のコンテナの番号を格納する配列である。

30

次に、ステップ S 4 4 0 6 において、 $Max\_Y$  から  $i$  番目のコンテナの下辺の  $Y$  座標を減算して、その値が  $Value\_Y$  以下かどうかをチェックする。 $Value\_Y$  以下であれば (S 4 2 0 6 - YES)、 $i$  番目のコンテナは、ロック対象として指定された範囲内であるので、変数  $i$  を  $Lock\_Cont[k]$  に代入する (S 4 4 0 7)。そして、変数  $k$  および変数  $i$  をインクリメントする (S 4 4 0 8 , S 4 4 0 9)。

【 0 2 1 9 】

ステップ S 4 4 0 6 で  $Value\_Y$  以下でない場合は (S 4 4 0 6 - NO)、 $Lock\_Cont$  に代入せずに、変数  $i$  をインクリメントする (S 4 4 0 9)。そして、すべてのコンテナについて処理が終了したかどうかを確認し (S 4 4 1 0)、終了していなければ (S 4 4 1 0 - NO)、ステップ S 4 4 0 6 へ進み処理を繰り返す。終了していれば (S 4 4 1 0 - YES)、ステップ S 4 4 1 1 に進め、変数  $k$  に 0 を代入する。そして、処理をステップ S 4 4 1 2 に進め、 $Lock\_Cont[k]$  の値をチェックする。もし、値が 0 ならば、ロックを設定せずにこの処理を終了する。値が 0 でないならば (S 4 4 1 2 - NO)、処理をステップ S 4 4 1 3 に進め、 $Lock\_Cont[k]$  番目のコンテナの下辺をロックする。そして、処理をステップ S 4 4 1 4 に進め、変数  $k$  をインクリメントし、 $Lock\_Cont[k]$  の値が 0 になるまで、ステップ S 4 4 1 3 と S 4 4 1 4 の処理を繰り返す。この処理により、リンクを設定したコンテナの辺の中で、ページ内の一番下からユーザが指定した範囲内にあるコンテナの辺がすべてロックされる。この場合、

40

50



複数の辺が該当する場合は、該当する複数の辺すべてがロックされる。

【0220】

図45はコンテナの下辺のロック状態を示す図であり、4501は複数のコンテナの選択を指定する矩形、4502、4503はリンク、4504は最大のY座標、4505はロック対象となる指定範囲の境界線、4508はValue\_Yの間隔を示している。この図で示されるように、図44の処理を行うと、最大Y座標から、Value\_Yで指定された範囲内にあるコンテナの下辺がロックされることになるが、ここでは、選択した範囲内にロック対象のコンテナがないのでロックはされていない。

【0221】

以上説明したとおり、本実施形態によれば、複数のコンテナ間に対して、一つの操作でリンクを一括設定することが可能になり、効率のよい操作を実現することが可能になる。

10

【0222】

(他の実施形態)

なお、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0223】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

20

【0224】

プログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0225】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

30

【0226】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【図面の簡単な説明】

【0227】

【図1A】本発明の実施形態にかかるコンピュータシステムの構成を示す図である。

40

【図1B】ホストコンピュータの具体的な構成を説明する図である。

【図2】レイアウトエンジンモジュールをホストコンピュータ101の他にエンジンサーバ227にもレイアウトエンジン225を設けた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ユーザインターフェース画面を示す図である。

【図4】コンテナ間のリンク、アンカーやスライダーを持つ典型的なコンテナを示す図である。

【図5A】コンテナを構成する辺に対するルールを例示的に説明する図である。

【図5B】コンテナを構成する辺に対するルールを例示的に説明する図である。

【図5C】コンテナを構成する辺に対するルールを例示的に説明する図である。

50

- 【図 5 D】コンテナを構成する辺に対するルールを例示的に説明する図である。
- 【図 6】リンクの設定方法を示すフローチャートである。
- 【図 7】リンク作成時のユーザインターフェースの画面表示例を示す図である。
- 【図 8】レイアウト計算の処理の流れを説明するフローチャートである。
- 【図 9】レイアウト計算の詳細な処理の流れを説明するフローチャートである。
- 【図 10】図 9 のフローに対応するユーザインターフェース画面の表示例を示した図である。
- 【図 11】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 12】リンクの自動設定に関する UI 画面を例示する図である。
- 【図 13】複数のコンテナ F, G, H を選択する UI 画面の例を示す図である。 10
- 【図 14 A】図 13 で説明した操作によりコンテナ F, G, H が選択されたことをユーザに示すために UI 画面が切り替えられた状態を示す図である。
- 【図 14 B】マウスポインタ 1303 でコンテナを押下する（指定する）ことによる複数のコンテナの選択の例を示す図である。
- 【図 15】複数選択されたコンテナ F, G, H 間にリンクが一括で設定された状態を示す図である。
- 【図 16】リンクの自動設定の全体的な流れを説明するフローチャートである。
- 【図 17】リンクの一括作成の処理の流れを概略的に説明するフローチャートである。
- 【図 18】コンテナの位置情報を取得する処理の流れを説明するフローチャートである。
- 【図 19 A】X 方向のリンクの作成処理の流れを説明するフローチャートである。 20
- 【図 19 B】X 方向のリンクの作成処理の流れを説明するフローチャートである。
- 【図 20】図 19 B のステップ S 1914 のリンク設定処理の流れを説明するフローチャートである。
- 【図 21 A】X 方向のリンク設定を説明する図である。
- 【図 21 B】X 方向のリンク設定を説明する図である。
- 【図 22 A】Y 方向のリンクの作成処理の流れを説明するフローチャートである。
- 【図 22 B】Y 方向のリンクの作成処理の流れを説明するフローチャートである。
- 【図 23】図 22 B のステップ S 2214 のリンク設定処理の流れを説明するフローチャートである。
- 【図 24 A】リンク作成の状態を示す UI 画面の例を示す図である。 30
- 【図 24 B】リンク作成の状態を示す UI 画面の例を示す図である。
- 【図 25】アンカーの設定に関する全体的な処理の流れを説明するフローチャートである。
- 【図 26】図 25 におけるステップ S 2501 の左辺をロックする処理の流れを具体的に説明するフローチャートである。
- 【図 27】コンテナの左辺のロック状態を示す図である。
- 【図 28】図 25 におけるステップ S 2502 の右辺をロックする処理の流れを具体的に説明するフローチャートである。
- 【図 29】コンテナの右辺のロック状態を示す図である。
- 【図 30】図 25 におけるステップ S 2503 の上辺をロックする処理の流れを具体的に説明するフローチャートである。 40
- 【図 31】コンテナの上辺のロック状態を示す図である。
- 【図 32】図 25 におけるステップ S 2504 の下辺をロックする処理の流れを具体的に説明するフローチャートである。
- 【図 33】コンテナの下辺のロック状態を示す図である。
- 【図 34】アンカーの設定の状態を示す図である。
- 【図 35】最大・最小値をもつフレキシブルなリンクを設定するための処理の流れを説明するフローチャートである。
- 【図 36】最大・最小値をもつフレキシブルなリンクを設定するための処理の流れを説明するフローチャートである。 50

【図 3 7】ユーザの指定した範囲にある辺をロックする処理の全体的な処理の流れを説明するフローチャートである。

【図 3 8】図 3 7 におけるステップ S 3 7 0 2 のコンテナの左辺をロックする処理の流れを具体的に説明するフローチャートである。

【図 3 9】コンテナの左辺のロック状態を示す図である。

【図 4 0】図 3 7 におけるステップ S 3 7 0 3 のコンテナの右辺をロックする処理の流れを具体的に説明するフローチャートである。

【図 4 1】コンテナの左辺のロック状態を示す図である。

【図 4 2】図 3 7 におけるステップ S 3 7 0 4 のコンテナの上辺をロックする処理の流れを具体的に説明するフローチャートである。

10

【図 4 3】コンテナの上辺のロック状態を示す図である。

【図 4 4】図 3 7 におけるステップ S 3 7 0 5 のコンテナの下辺をロックする処理の流れを具体的に説明するフローチャートである。

【図 4 5】コンテナの下辺のロック状態を示す図である。

【図 4 6】自動レイアウトシステムにおける典型的な可変リンクによるコンテナの配置を説明する図である。

【図 4 7】自動レイアウトシステムにおける可変リンクを設定するフローを示している。

【図 4 8】リンク 4 6 0 9 の情報がセットされているダイアログウィンドウの例を示す図である。

【図 4 9】自動レイアウトシステムにおける固定リンクによるレイアウト結果を示す図である。

20

【図 5 0】自動レイアウトシステムにおける可変リンクによるレイアウト結果を示す図である。

【図 5 1】優先順位の設定フローの流れを説明するフローチャートである。

【図 5 2】リンクの一括設定により作成されたリンクとそのコンテナを示す図である。

【図 5 3】自動優先順位設定のダイアログボックスの表示例を示す図である。

【図 5 4 A】X 方向を優先して優先順位が設定される前のコンテナとリンクを示す図である。

【図 5 4 B】X 方向を優先して優先順位が設定された後のコンテナとリンクを示す図である。

30

【図 5 5 A】Y 方向を優先して優先順位が設定される前のコンテナとリンクを示す図である。

【図 5 5 B】Y 方向を優先して優先順位が設定された後のコンテナとリンクを示す図である。

【図 5 6】優先順位の設定処理の流れを説明するフローチャートである。

【図 5 7】コンテナの集合を単位として優先順位を設定する内容を説明する図である。

【図 5 8】優先順位を設定するダイアログユーザインターフェース (UI) 画面の例を示す図である。

【図 5 9】コンテナに設定する優先順位を変更する手順を説明する図である。

【図 6 0】優先順位を設定するモードへ移行するためのポップアップメニューの例を示す図である。

40

【図 6 1】優先順位の設定をユーザに示す UI 画面の例を示す図である。

【図 6 2】優先順位の設定をユーザに示す UI 画面の例を示す図である。

【図 6 3】優先順位の設定をユーザに示す UI 画面の例を示す図である。

【図 6 4】優先順位が設定されたコンテナのレイアウトの計算処理の流れを説明するフローチャートである。

【図 6 5】レイアウトの優先順位がコンテナに設定されていた場合のレイアウト計算処理の流れを説明するフローチャートである。

【図 6 6】図 6 5 のレイアウト計算処理で表示される UI 画面の例を示す図である。

【図 6 7】図 6 5 のレイアウト計算処理で表示される UI 画面の例を示す図である。

50

【図 6 8】リンクの自動設定において、最小距離に限らない場合の処理の流れを説明するフローチャートである。

【図 6 9】リンクの自動設定方法を選択するための UI 画面の例を示す図である。

【図 7 0 A】リンクの自動設定において、最小距離に限らない自動リンク設定方法が選択された場合の、リンク作成処理の流れを説明するフローチャートである。

【図 7 0 B】リンクの自動設定において、最小距離に限らない自動リンク設定方法が選択された場合の、リンク作成処理の流れを説明するフローチャートである。

【図 7 1】図 7 0 A , B の処理による X 方向のリンク設定を説明する図である。

【図 7 2】図 7 0 A、B の処理によるリンクの自動設定において、最小距離に限らない自動リンク設定方法が選択された場合の、具体例を示す図である。

10

【図 7 3】リンクの自動設定処理におけるリンク生成方向を選択するための UI 画面の例を示す図である。

【符号の説明】

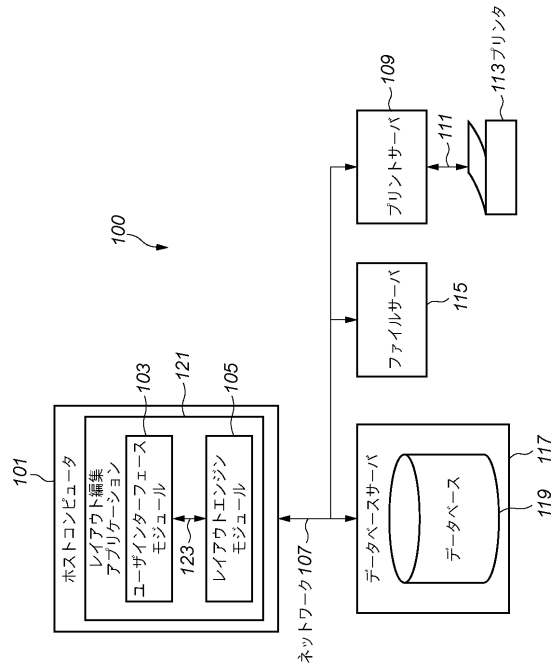
【 0 2 2 8 】

- 1 0 1    ホストコンピュータ
- 1 0 3    ユーザインターフェースモジュール
- 1 0 5    レイアウトエンジンモジュール
- 1 1 9    データベース
- 1 2 1    レイアウト編集アプリケーションプログラム
- 1 3 2    キーボード
- 1 3 3    マウス
- 1 4 3    I/O インタフェース
- 1 4 4    ディスプレイ装置
- 3 0 1    アプリケーションウインドウ
- 3 0 3    ツールバー
- 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

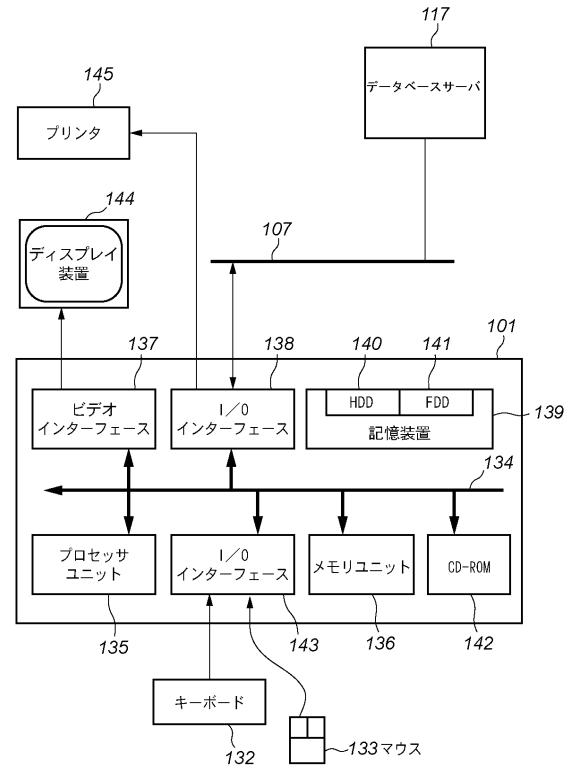
20

30

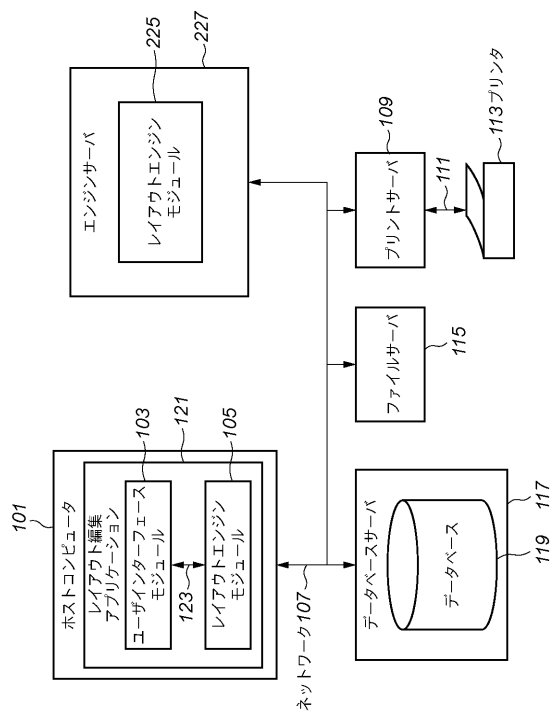
【図 1 A】



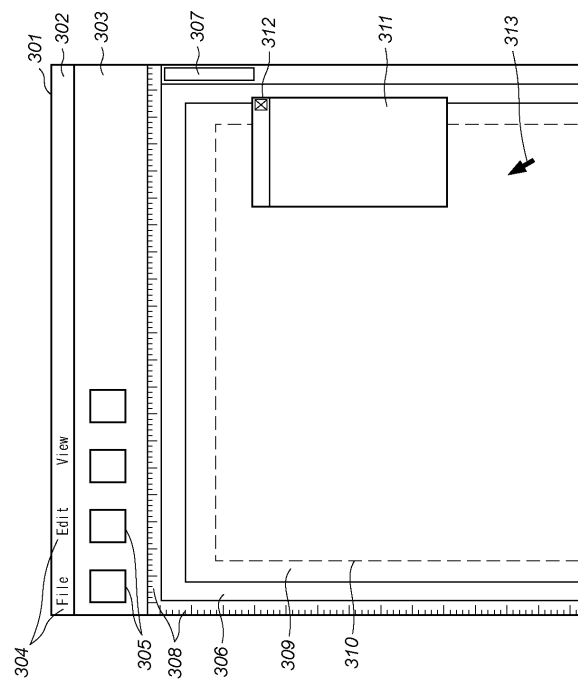
【図 1 B】



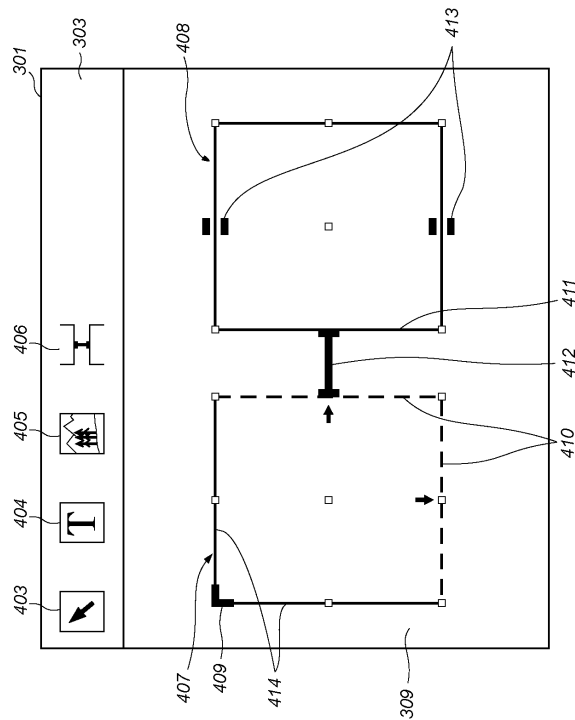
【図 2】



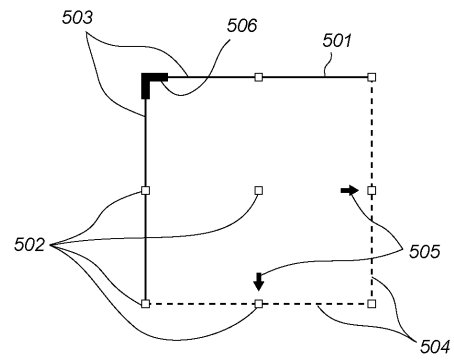
【図 3】



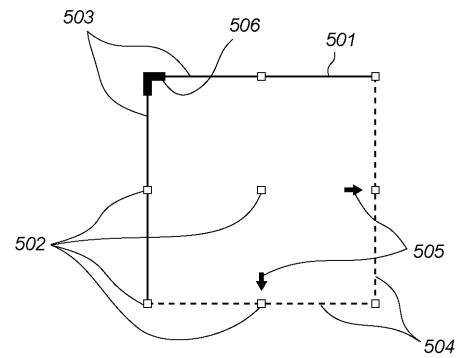
【図 4】



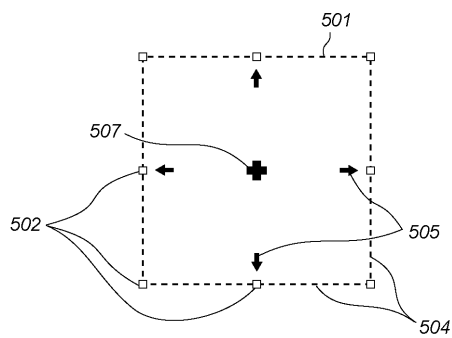
【図 5 A】



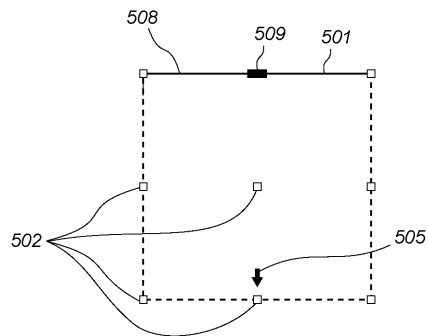
【図 5 B】



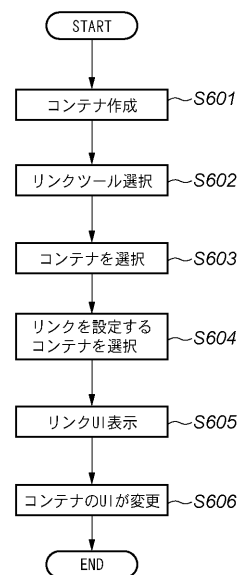
【図 5 C】



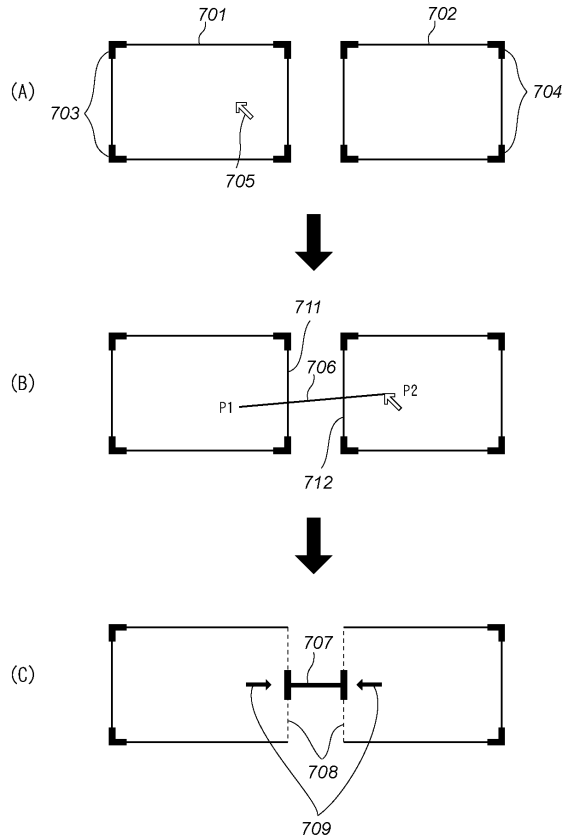
【図 5 D】



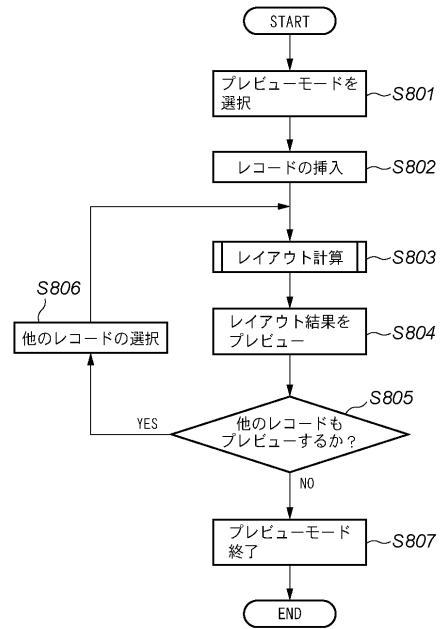
【図 6】



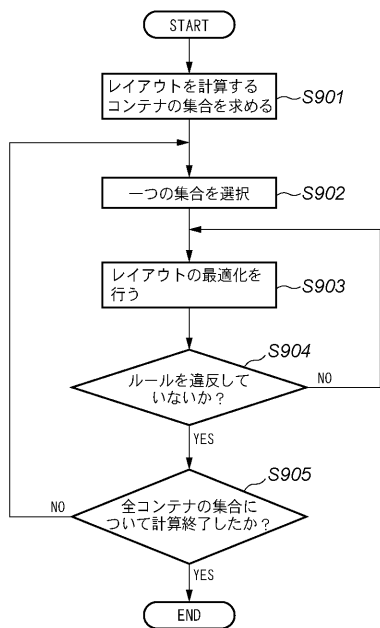
【図 7】



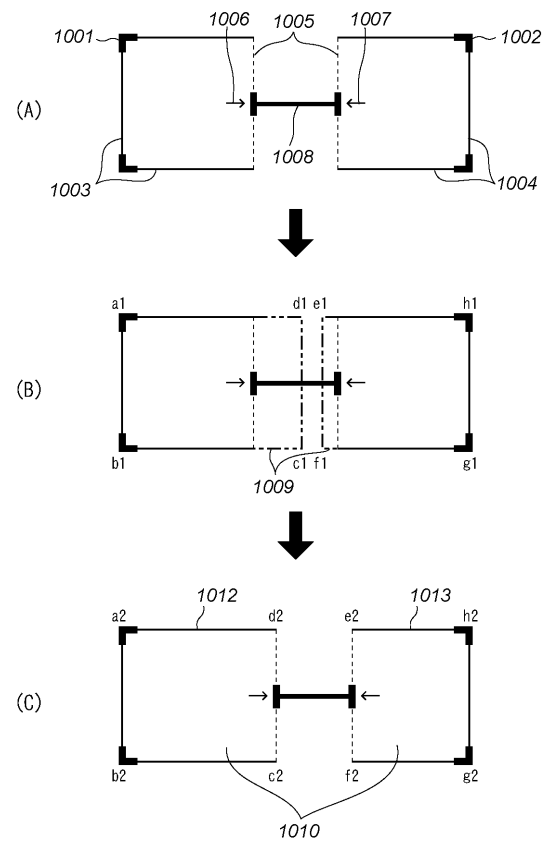
【図 8】



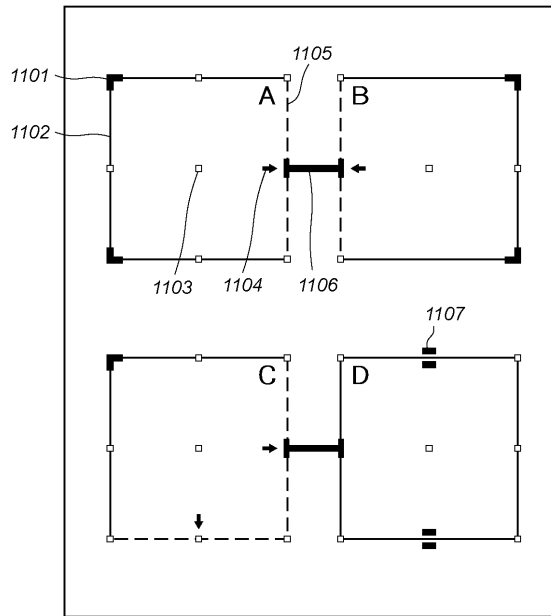
【図 9】



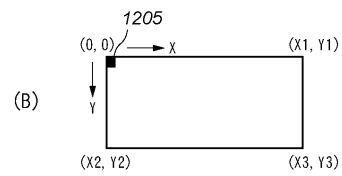
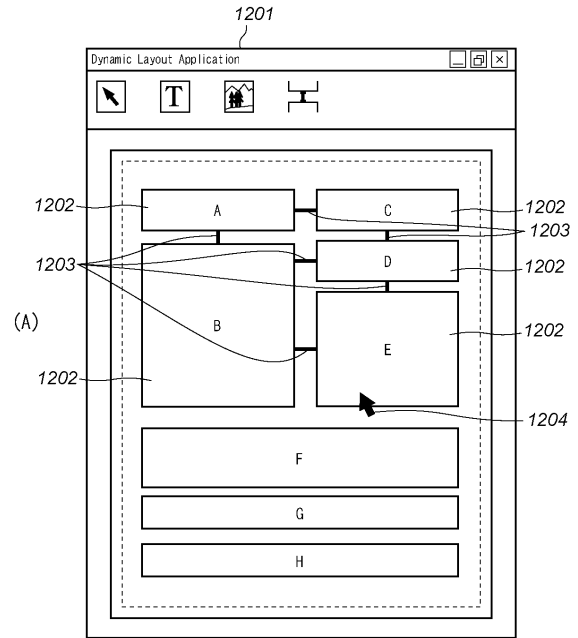
【図 10】



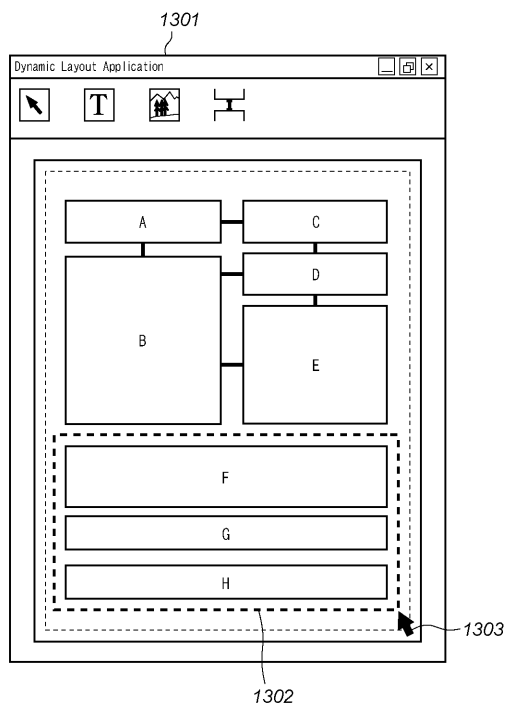
【 図 1 1 】



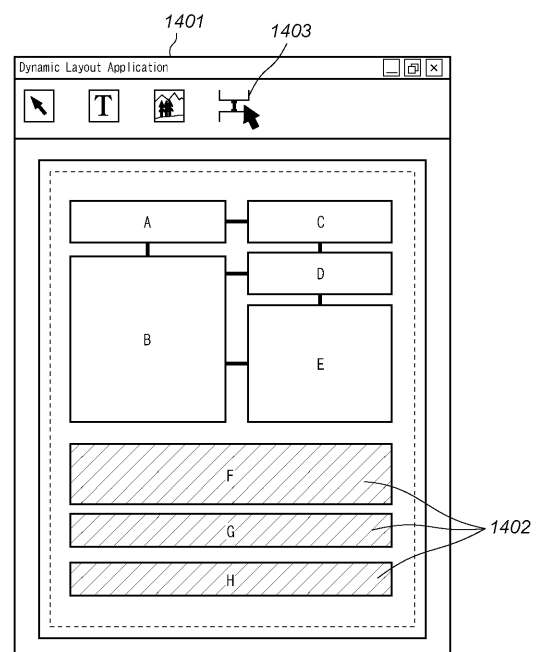
【 図 1 2 】



【 図 1 3 】

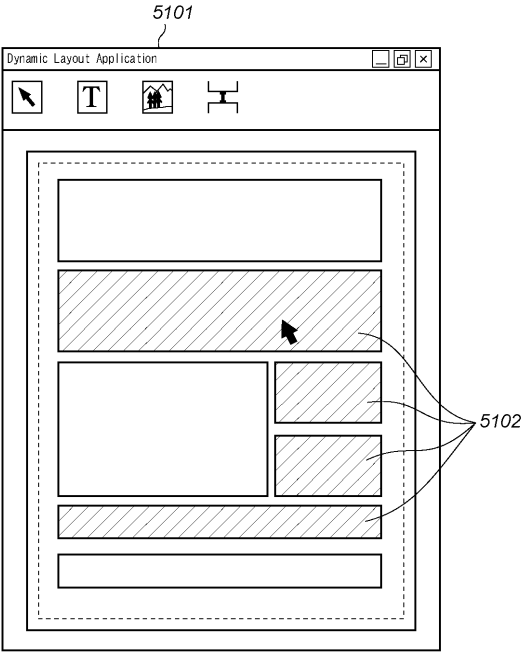


【 図 1 4 A 】

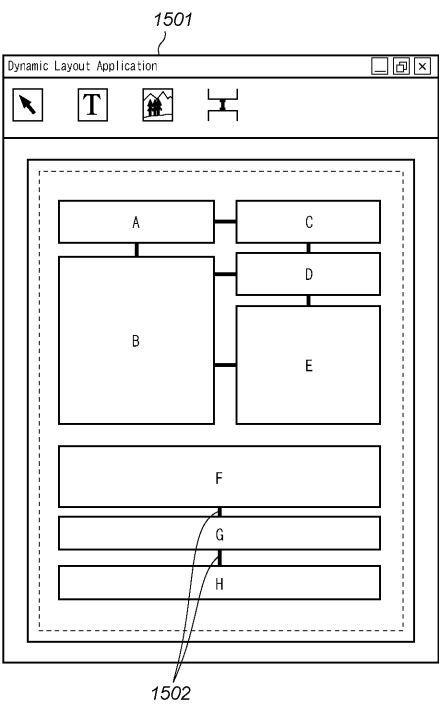




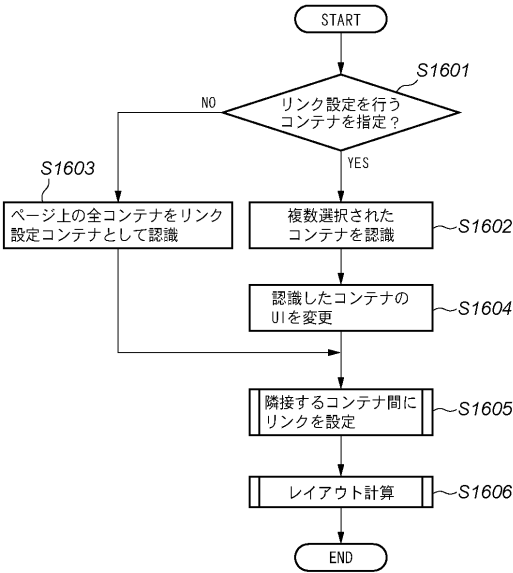
【図 14 B】



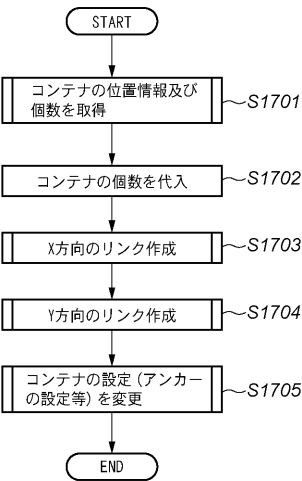
【図 15】



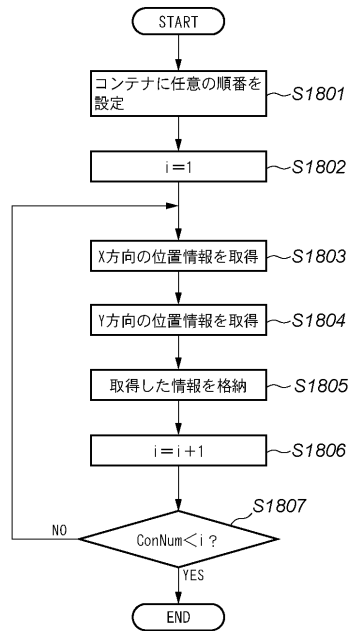
【図 16】



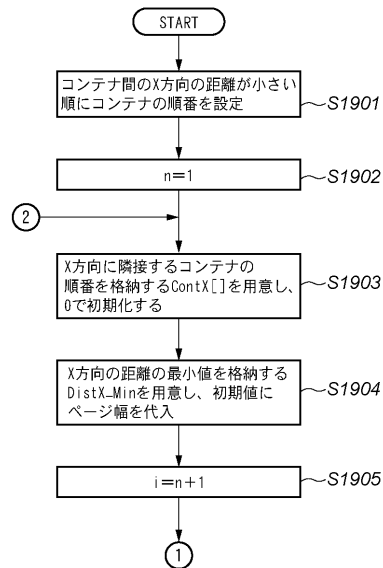
【図 17】



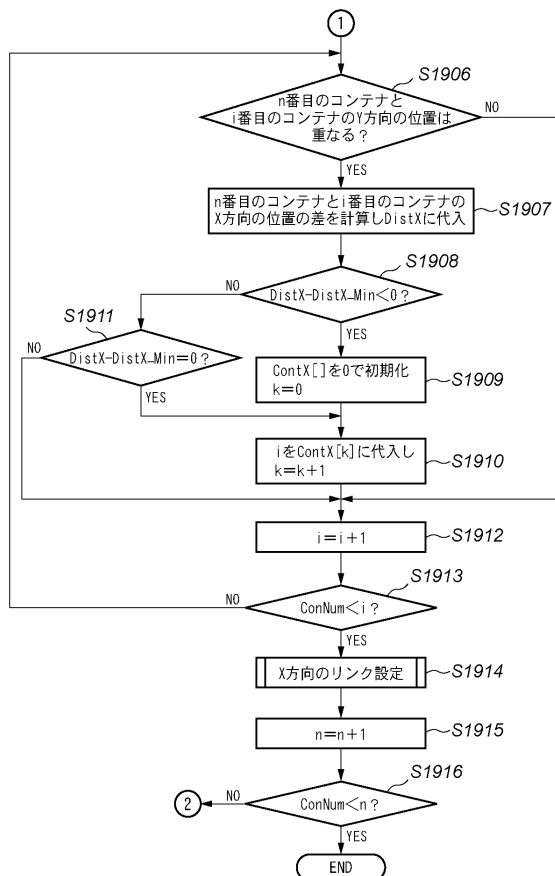
【図 18】



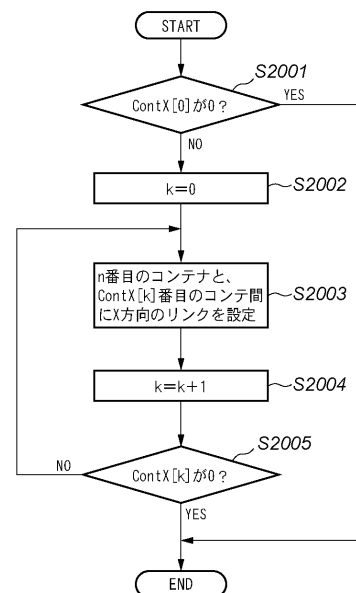
【図 19 A】



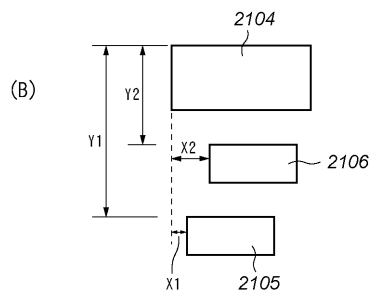
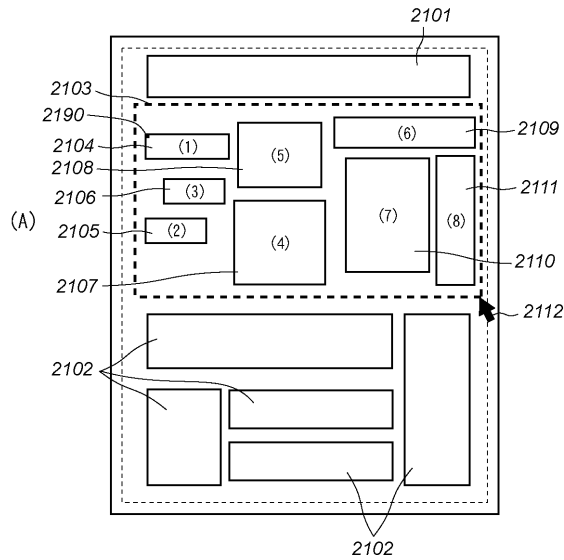
【図 19 B】



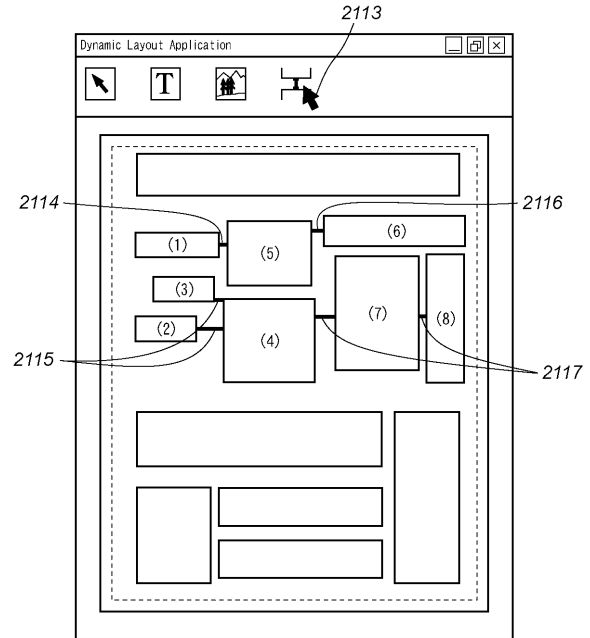
【図 20】



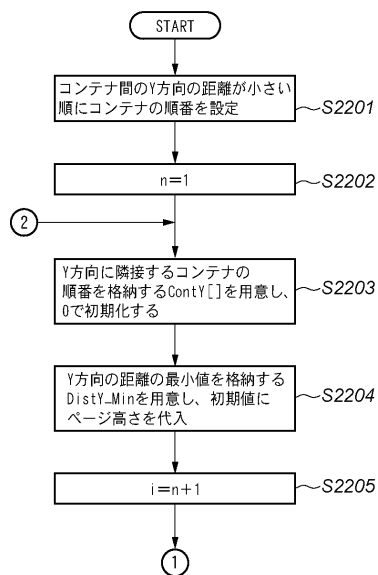
【図 2 1 A】



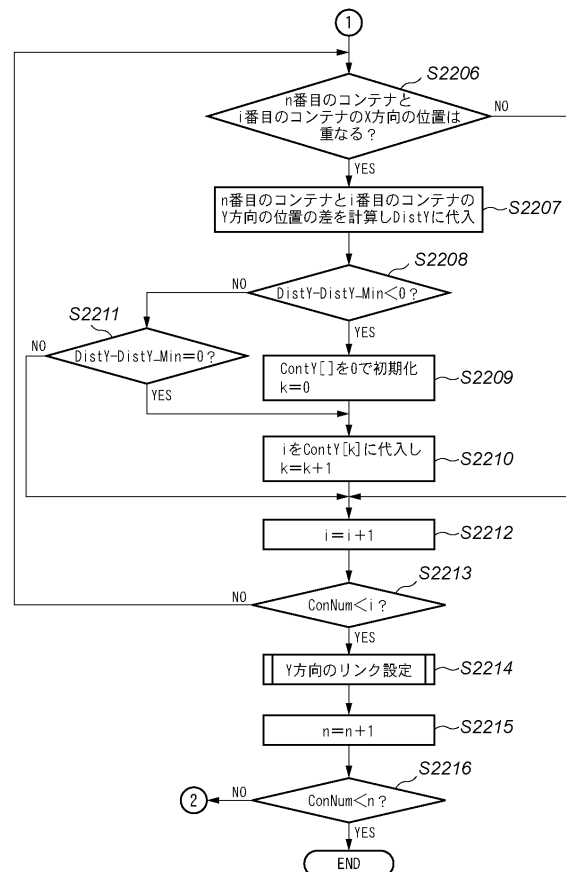
【図 2 1 B】



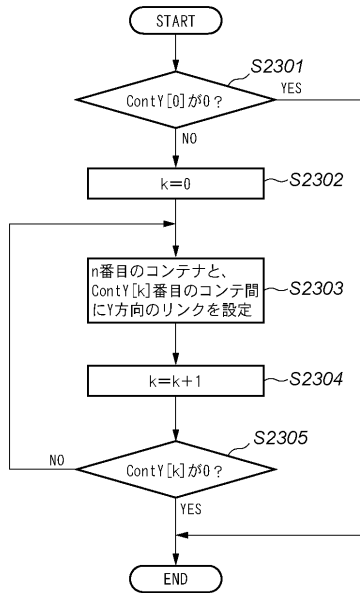
【図 2 2 A】



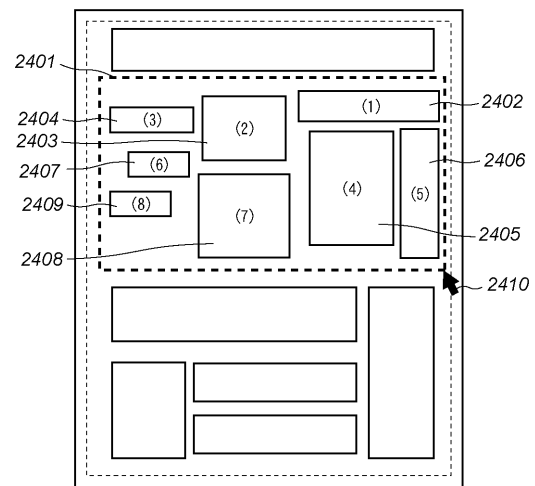
【図 2 2 B】



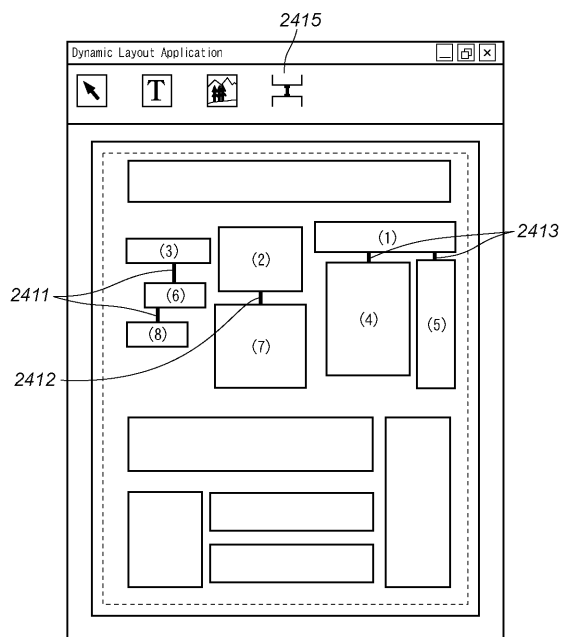
【図 2 3】



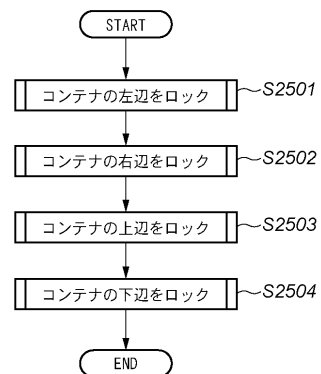
【図 2 4 A】



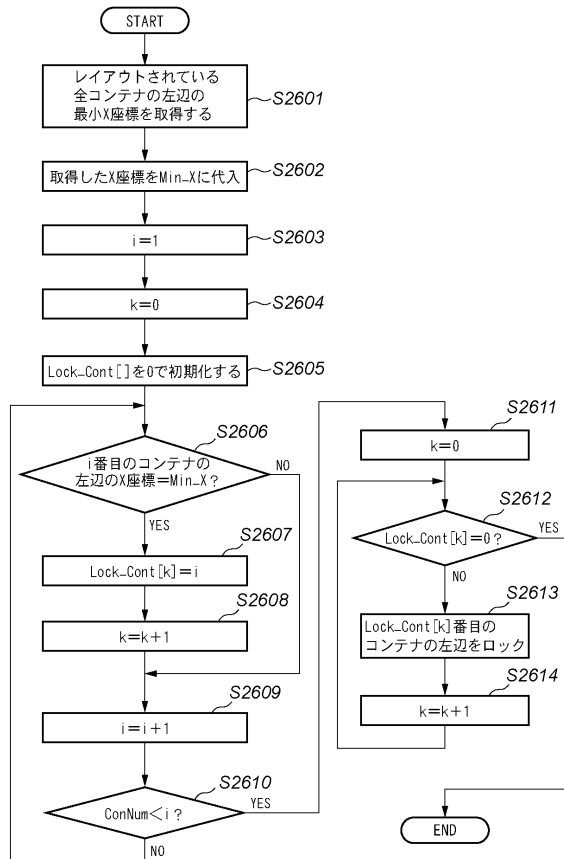
【図 2 4 B】



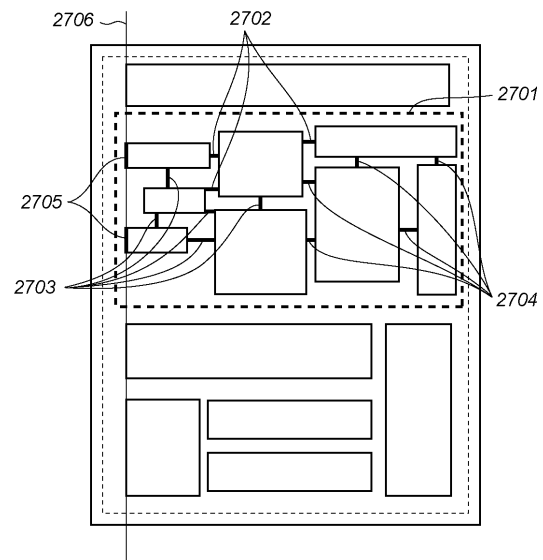
【図 2 5】



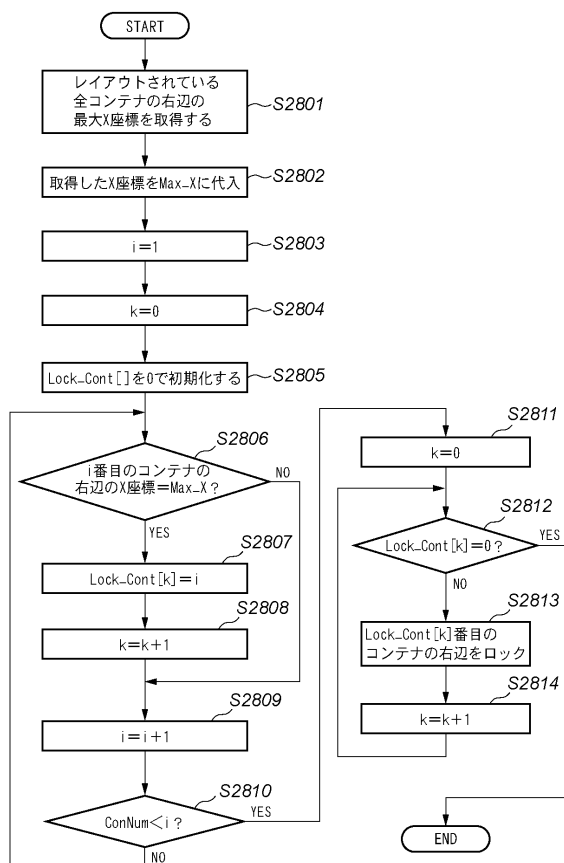
【図 26】



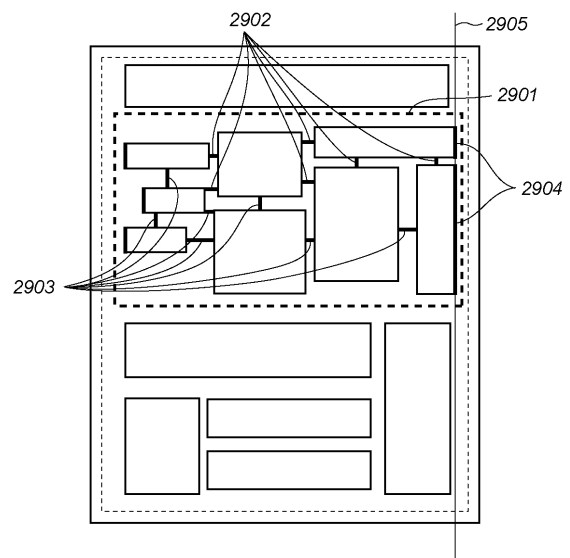
【図 27】



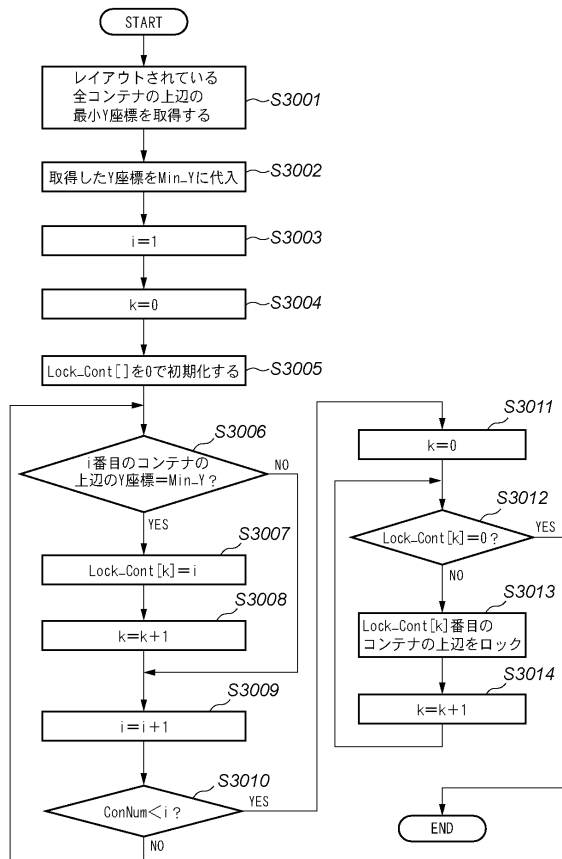
【図 28】



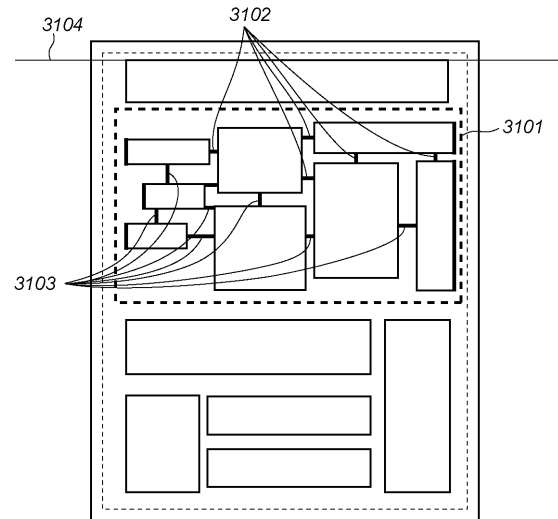
【図 29】



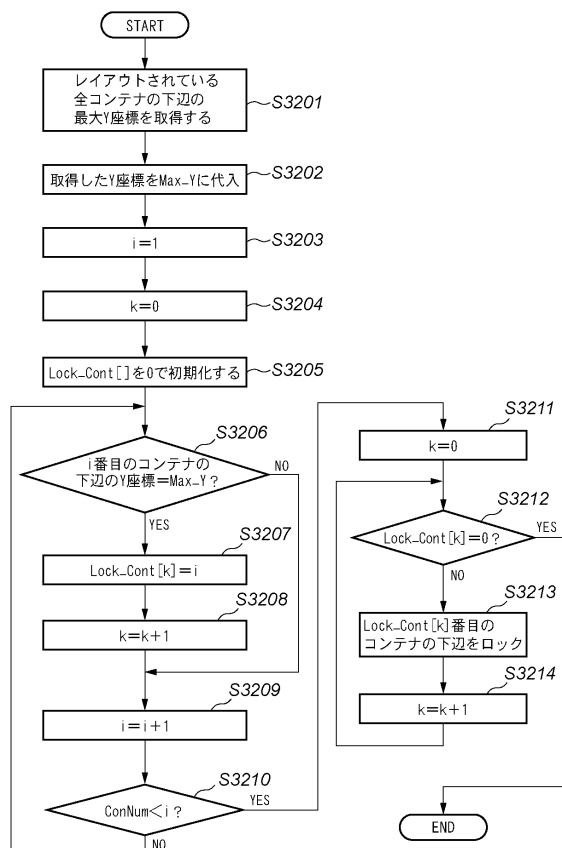
【図 30】



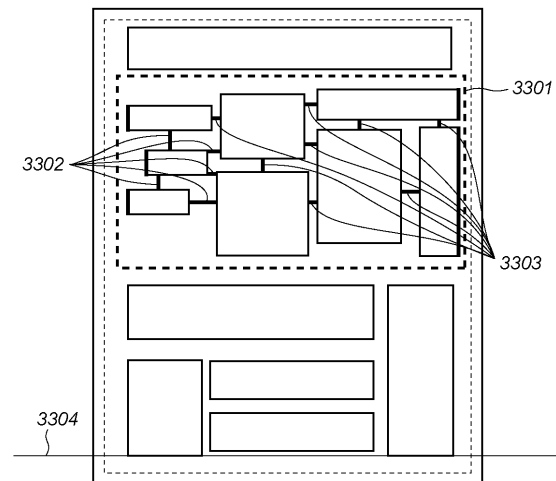
【図 31】



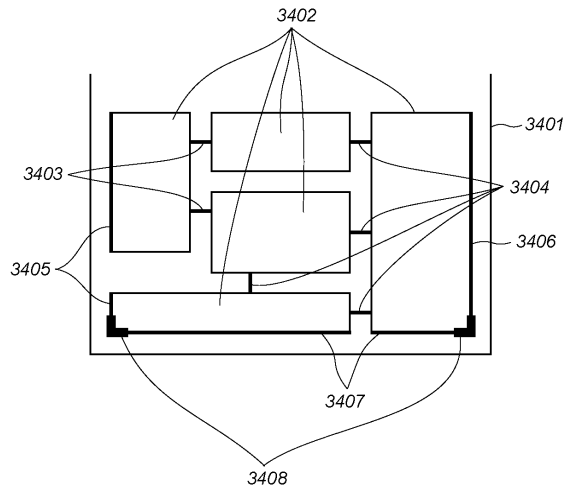
【図 32】



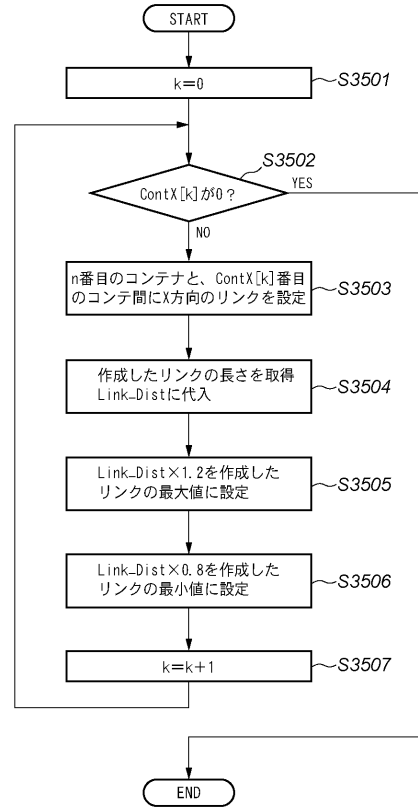
【図 33】



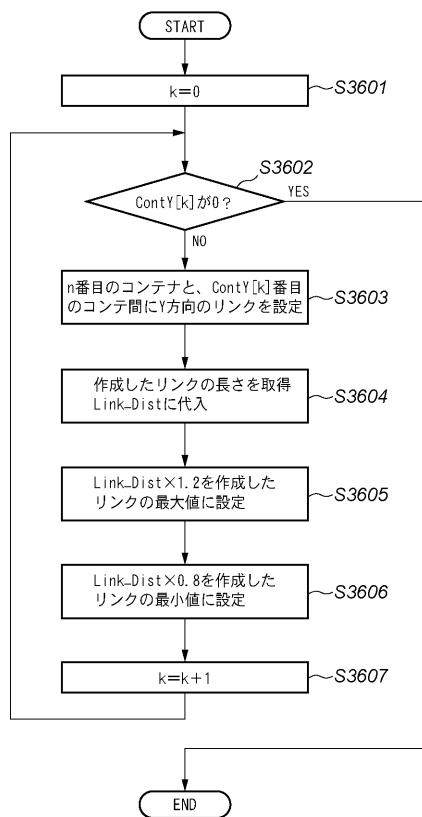
【図 3 4】



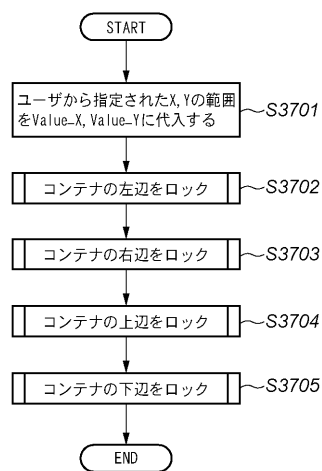
【図 3 5】



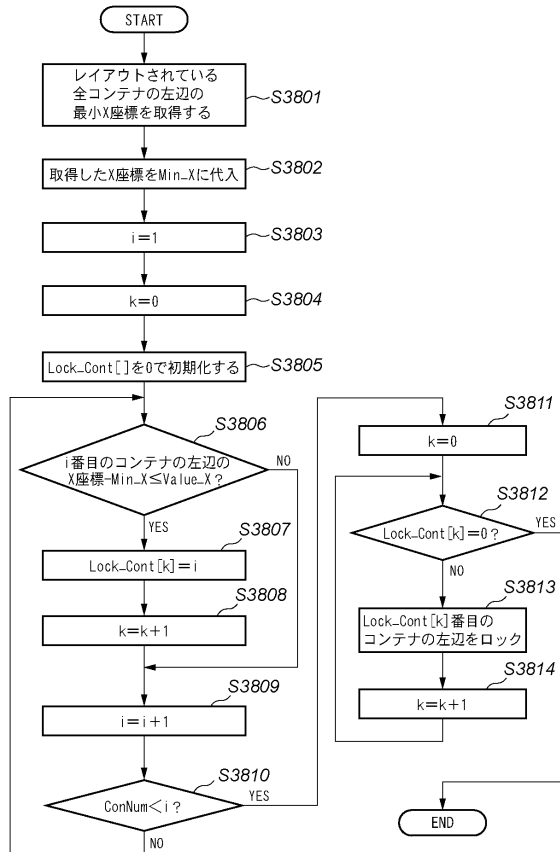
【図 3 6】



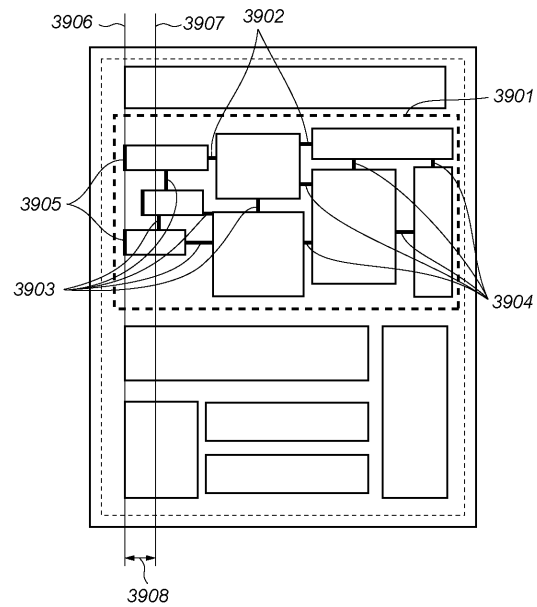
【図 3 7】



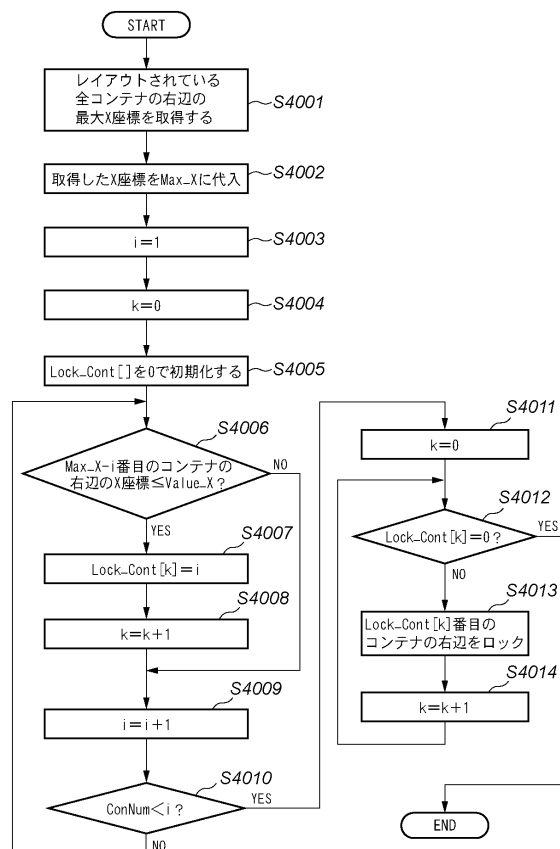
【図 38】



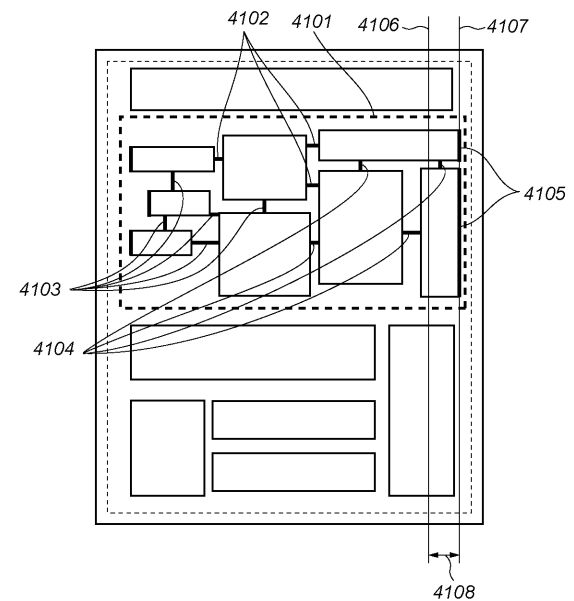
【図 39】



【図 40】

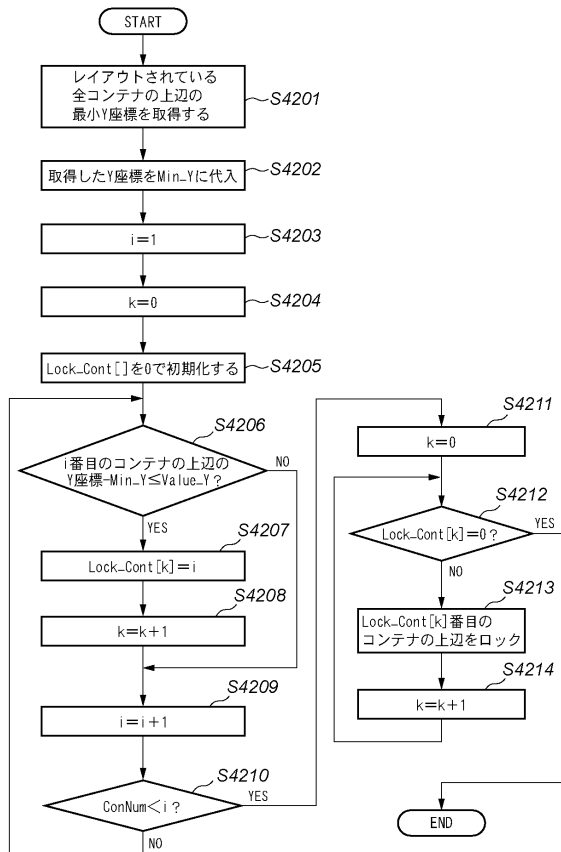


【図 41】

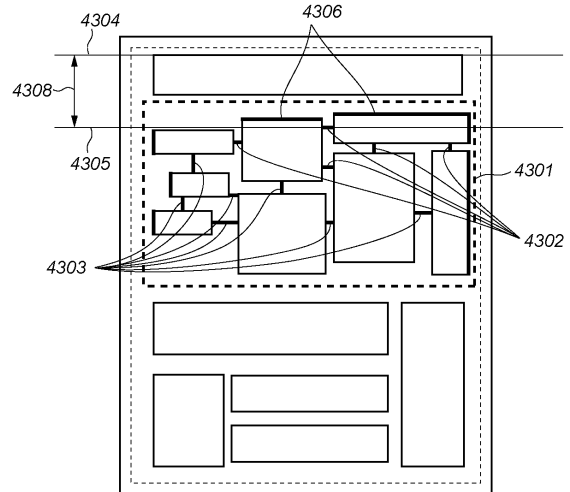




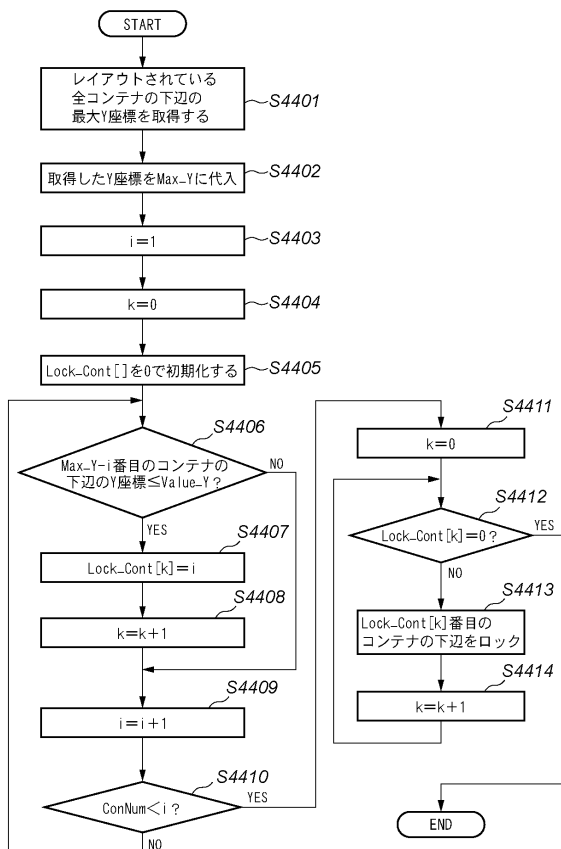
【図 4 2】



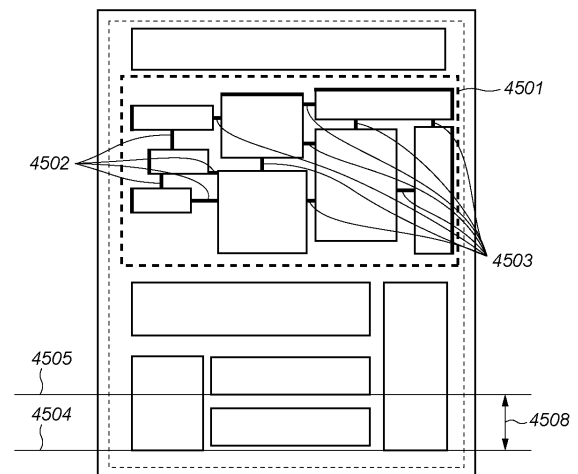
【図 4 3】



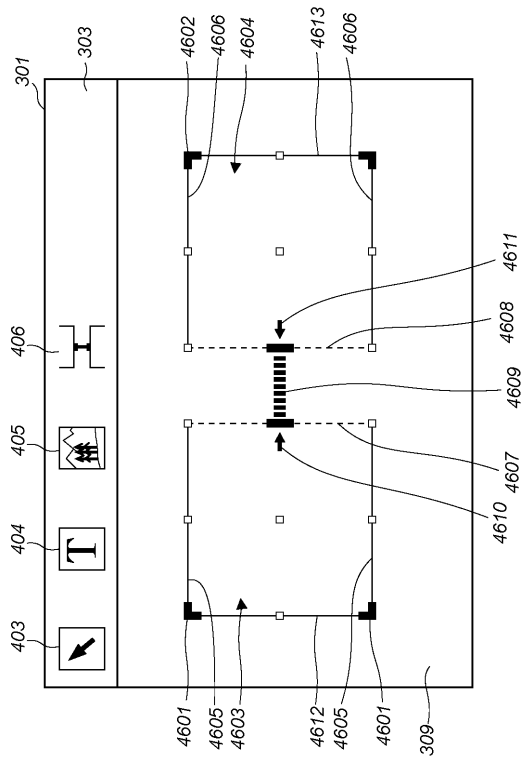
【図 4 4】



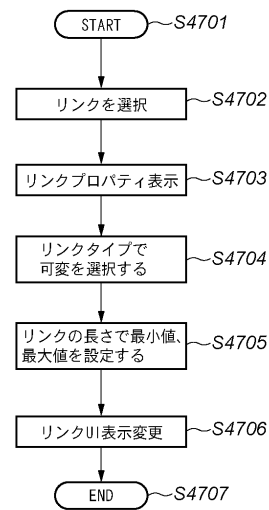
【図 4 5】



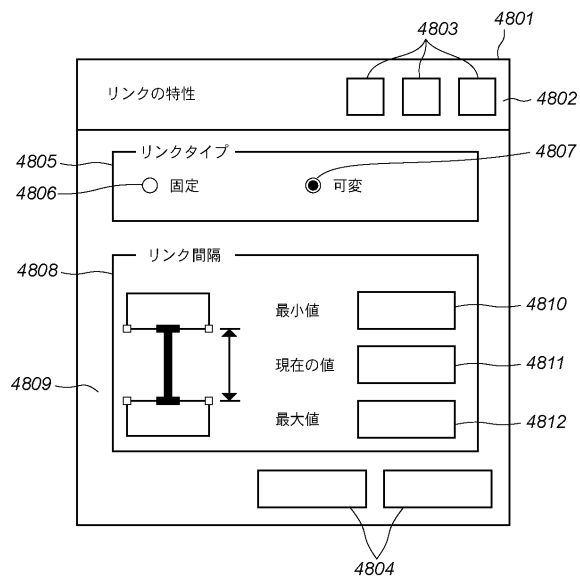
【図 46】



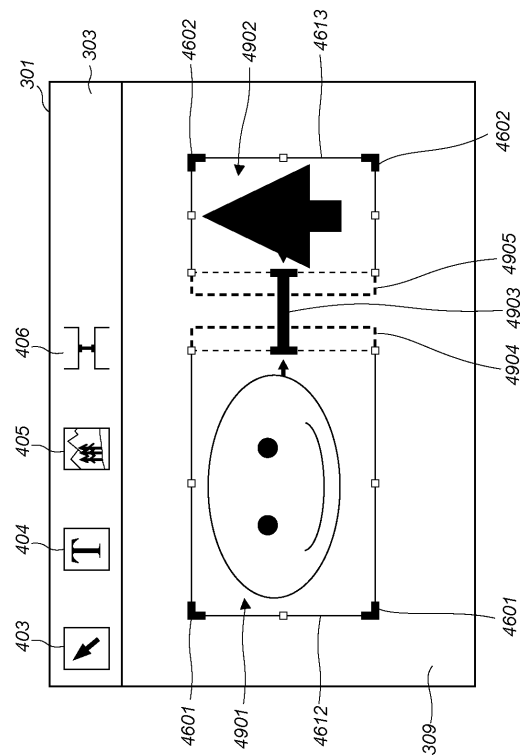
【図 47】



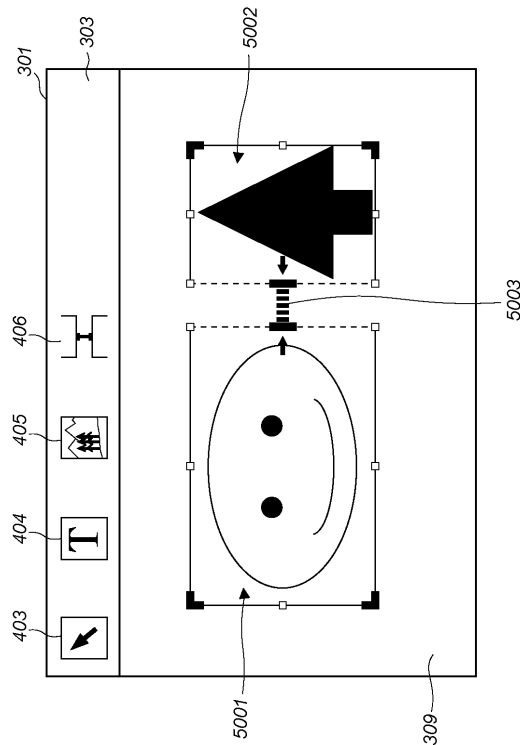
【図 48】



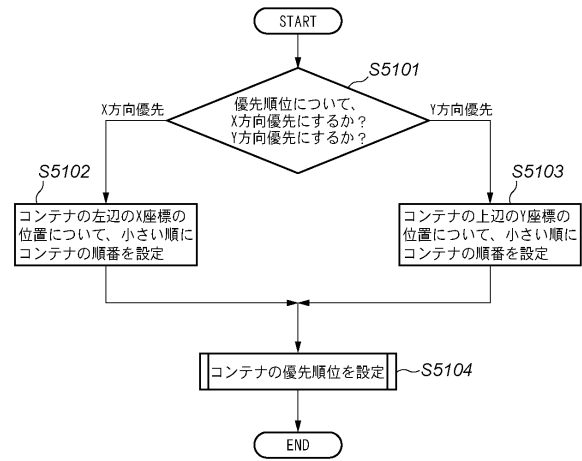
【図 49】



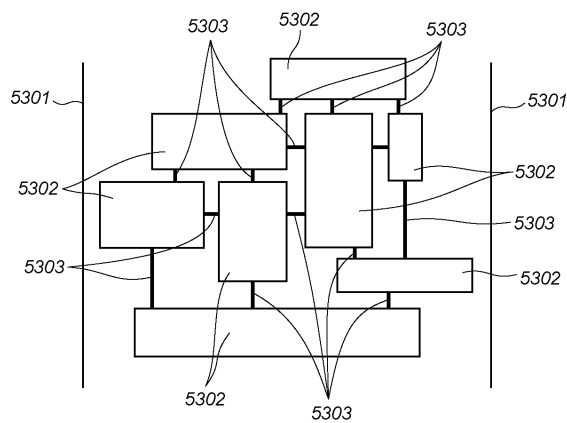
【図 50】



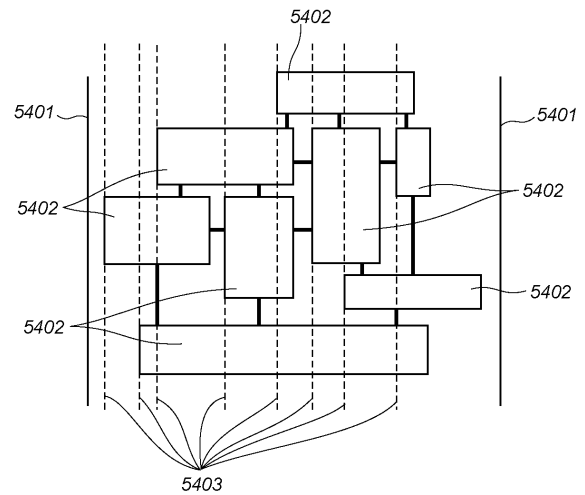
【図 51】



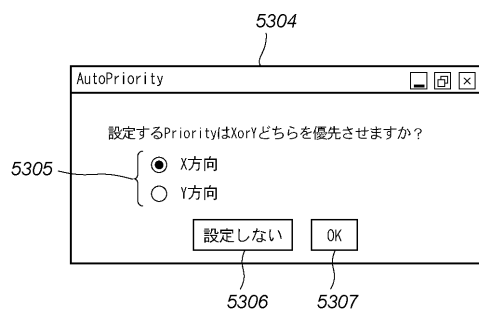
【図 52】



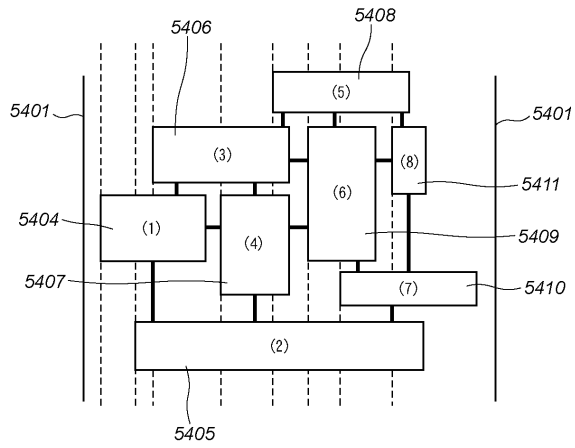
【図 54 A】



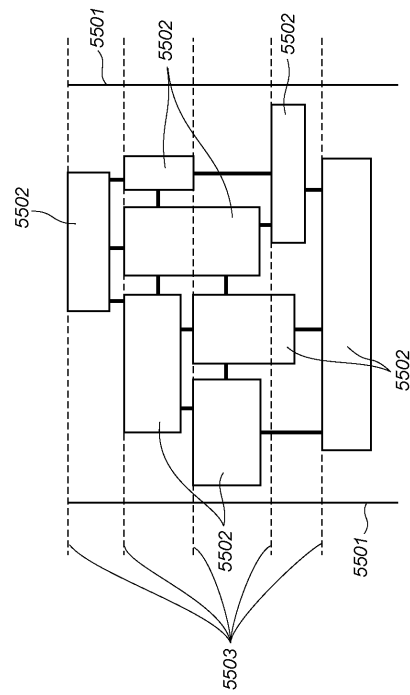
【図 53】



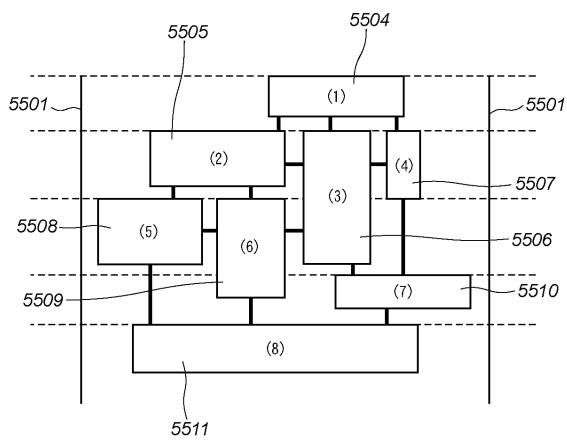
【図 5 4 B】



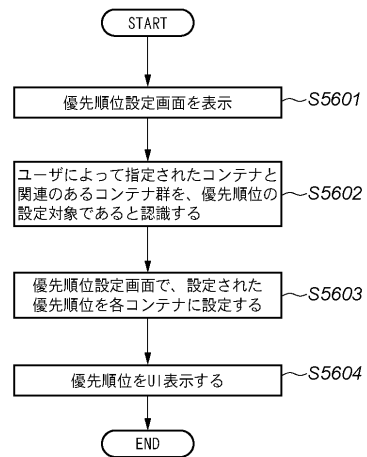
【図 5 5 A】



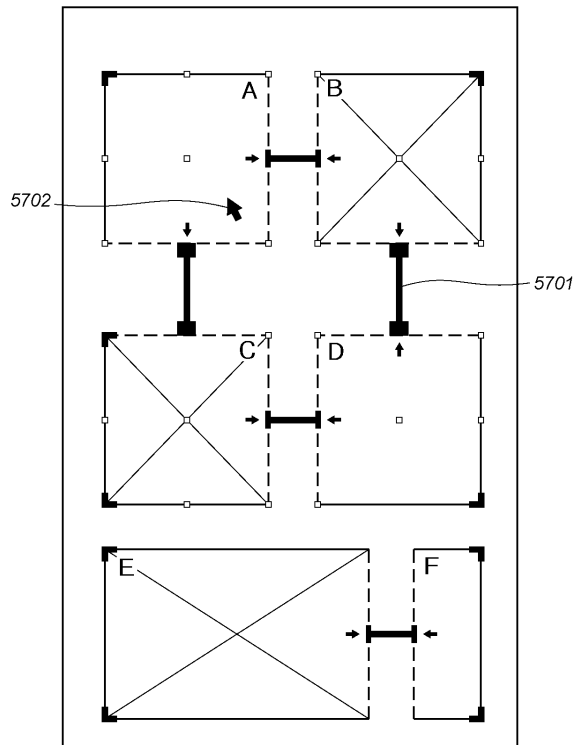
【図 5 5 B】



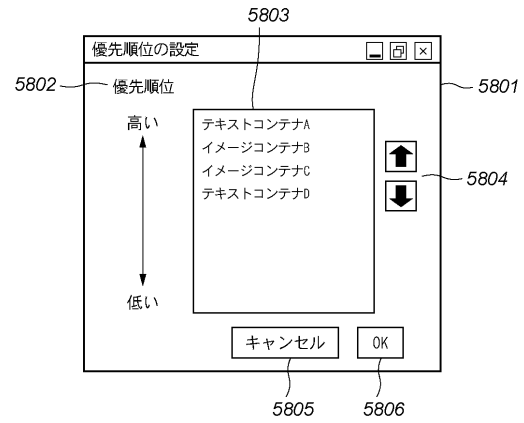
【図 5 6】



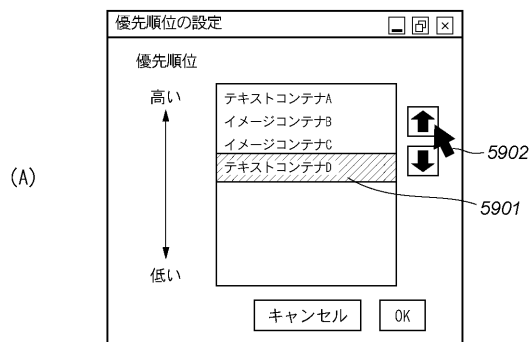
【図 57】



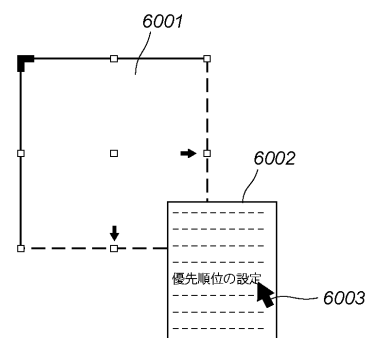
【図 58】



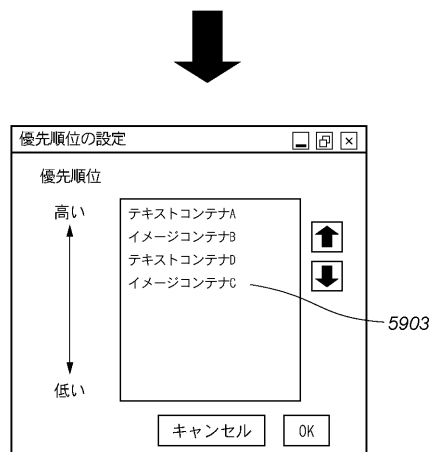
【図 59】



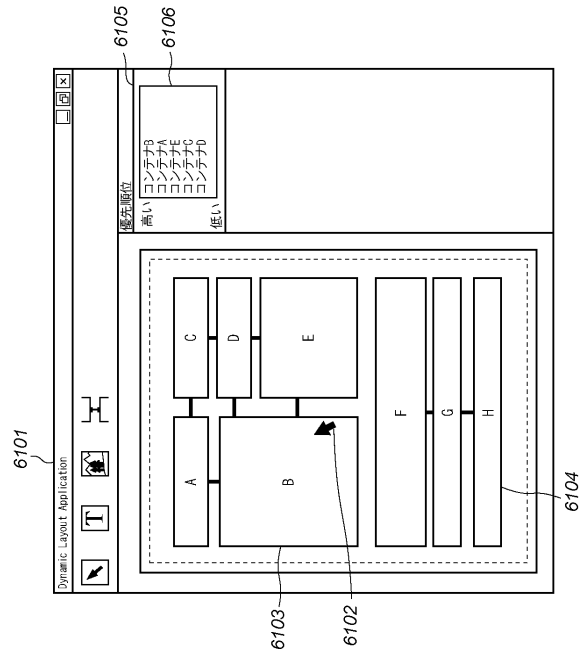
【図 60】



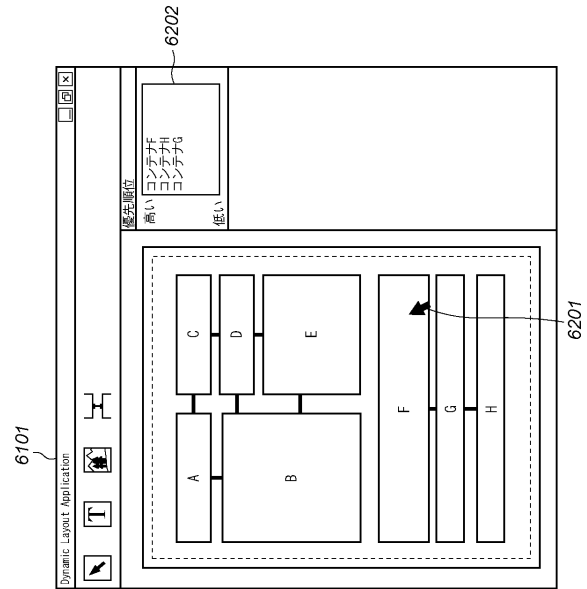
(B)



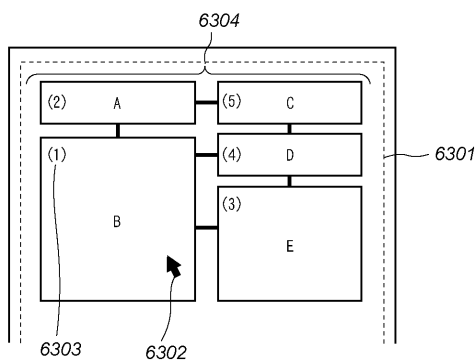
【図 6 1】



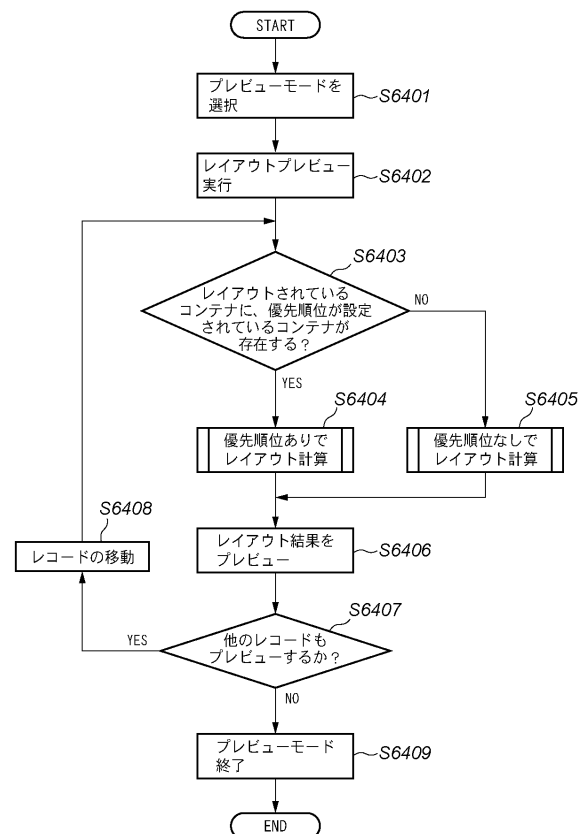
【図 6 2】



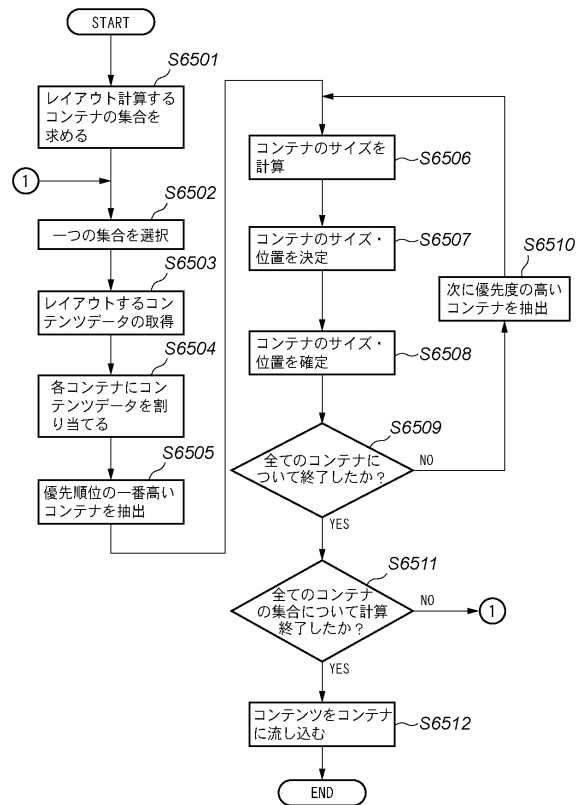
【図 6 3】



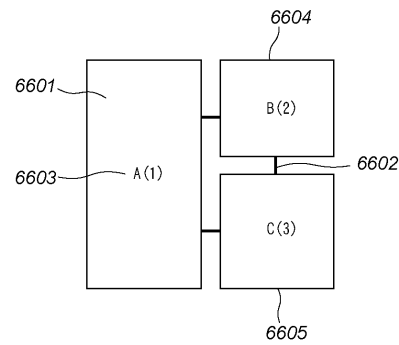
【図 6 4】



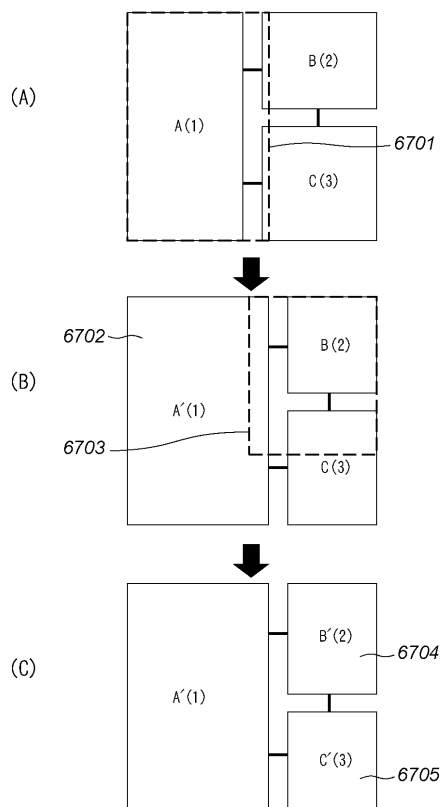
【図 65】



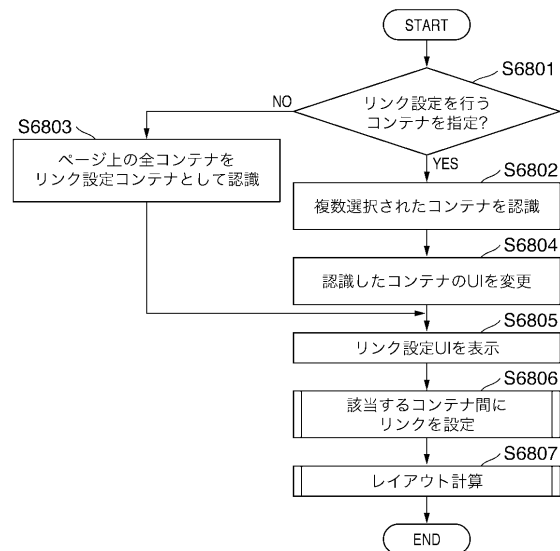
【図 66】



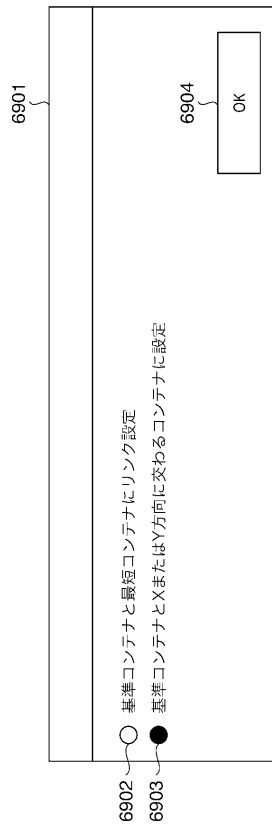
【図 67】



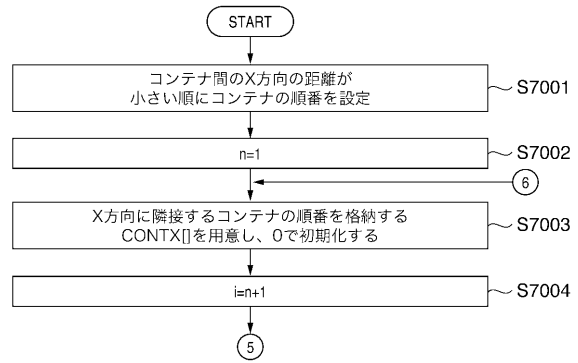
【図 68】



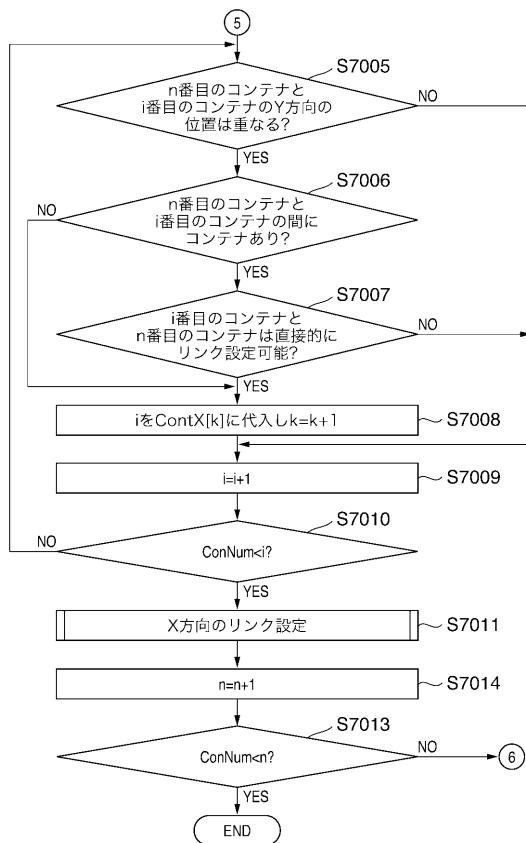
【図 69】



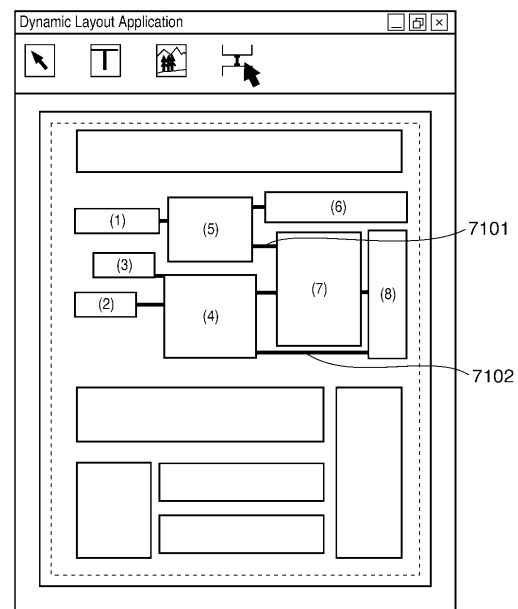
【図 70 A】



【図 70 B】

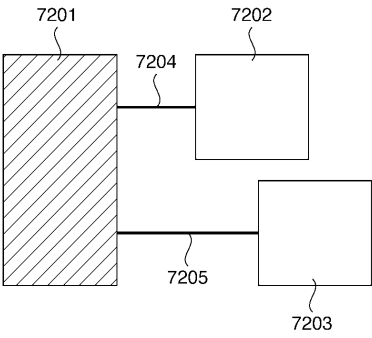


【図 71】

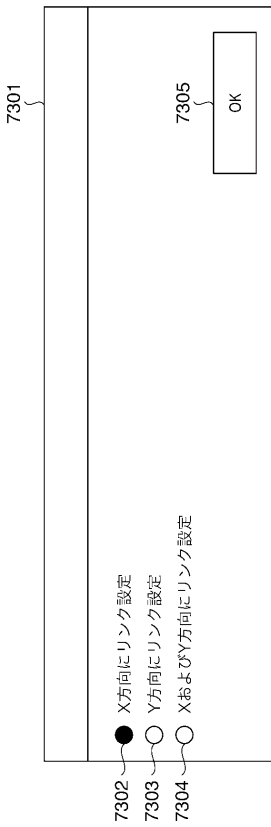




【図 7 2】



【図 7 3】



---

フロントページの続き

- (72)発明者 原 健太  
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 小橋 和文  
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 牧野 順  
神奈川県横浜市港北区菊名6-20-42

審査官 浜岸 広明

- (56)参考文献 米国特許第05845303(US, A)

- (58)調査した分野(Int.Cl., DB名)  
G06F 17/20 - 17/26