

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4065545号
(P4065545)

(45) 発行日 平成20年3月26日 (2008. 3. 26)

(24) 登録日 平成20年1月11日 (2008. 1. 11)

(51) Int. Cl.	F I
G 0 6 T 11/60 (2006. 01)	G O 6 T 11/60 1 0 0 A
G 0 6 F 17/21 (2006. 01)	G O 6 F 17/21 5 3 6
H 0 4 N 1/387 (2006. 01)	H O 4 N 1/387

請求項の数 11 (全 95 頁)

(21) 出願番号	特願2004-290545 (P2004-290545)	(73) 特許権者	000001007
(22) 出願日	平成16年10月1日 (2004. 10. 1)		キヤノン株式会社
(65) 公開番号	特開2005-135397 (P2005-135397A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成17年5月26日 (2005. 5. 26)	(74) 代理人	100076428
審査請求日	平成16年10月1日 (2004. 10. 1)		弁理士 大塚 康德
(31) 優先権主張番号	2003905658	(74) 代理人	100112508
(32) 優先日	平成15年10月15日 (2003. 10. 15)		弁理士 高柳 司郎
(33) 優先権主張国	オーストラリア (AU)	(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及び装置とそのプログラム

(57) 【特許請求の範囲】

【請求項 1】

コンテンツデータを配置するための複数の部分領域を含むテンプレートと、各部分領域に配置されるコンテンツデータとに基づいて可変データ文書を生成するレイアウト制御方法であって、

前記複数の部分領域の少なくとも2つの部分領域を結合する第1制約条件を設定する設定工程と、

前記複数の部分領域にコンテンツデータを割当てて割り当て工程と、

前記第1制約条件によって結合された前記少なくとも2つの部分領域に割り当てられるコンテンツデータの所定方向のサイズと前記第1制約条件のサイズとの和である第1の値が、前記部分領域を配置すべき領域の前記所定方向のサイズである第2の値に収まらない場合、前記第1制約条件によって結合された前記少なくとも2つの部分領域に割り当てられるコンテンツデータの所定方向のサイズと前記第1制約条件のサイズとの和である第1の値を、前記部分領域を配置すべき領域の前記所定方向のサイズである第2の値に収め、かつ、前記2つの部分領域のそれぞれについて部分領域のサイズと部分領域に割り当てられるコンテンツデータのサイズとの差分を算出し、当該差分の合計が最小となるように、前記第1制約条件によって結合された前記少なくとも2つの部分領域のサイズを決定する決定工程と、

前記決定工程によって決定された部分領域のサイズに従って前記コンテンツデータを配置する配置工程とを有し、

10

20

前記決定工程は、前記第 1 制約条件によって結合された少なくとも 2 つの部分領域のそれぞれについて、部分領域に割当てられるコンテンツデータのサイズと前記テンプレートに含まれる各部分領域との差分を 2 乗し、当該 2 乗された差分の合計が最小となるように部分領域のサイズを決定することを特徴とするレイアウト制御方法。

【請求項 2】

前記決定工程は、前記第 1 制約条件によって結合された少なくとも 2 つの部分領域のそれぞれについて、少なくとも 1 辺の位置を変更することで前記少なくとも 2 つの部分領域のサイズを決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】

前記設定工程は、前記テンプレートにおける制約条件の 1 つとして、前記部分領域の最大サイズ、最小サイズを設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。

10

【請求項 4】

前記決定工程は、前記第 1 制約条件によって結合された少なくとも 2 つの部分領域のそれぞれについて、部分領域に割当てられるコンテンツデータのサイズと前記テンプレートに含まれる各部分領域との差分を算出し、前記制約条件内において当該差分の合計が最小となるまで、変更可能な辺の位置を変更して、前記少なくとも 2 つの部分領域のサイズを決定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】

前記設定工程は、前記テンプレートにおける複数の部分領域から第 1 部分領域と第 2 部分領域とを選択することに従い、前記第 1 制約条件であるストラットを設定することを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載のレイアウト制御方法。

20

【請求項 6】

コンテンツデータを配置するための複数の部分領域を含むテンプレートと、各部分領域に配置されるコンテンツデータとに基づいて可変データ文書を生成するレイアウト制御装置であって、

前記複数の部分領域の少なくとも 2 つの部分領域を結合する第 1 制約条件を設定する設定手段と、

前記複数の部分領域にコンテンツデータを割当てる割当て手段と、

前記第 1 制約条件によって結合された前記少なくとも 2 つの部分領域に割当てられるコンテンツデータの所定方向のサイズと前記第 1 制約条件のサイズとの和である第 1 の値が、前記部分領域を配置すべき領域の前記所定方向のサイズである第 2 の値に収まらない場合、前記第 1 制約条件によって結合された前記少なくとも 2 つの部分領域に割当てられるコンテンツデータの所定方向のサイズと前記第 1 制約条件のサイズとの和である第 1 の値を、前記部分領域を配置すべき領域の前記所定方向のサイズである第 2 の値に収め、かつ、前記 2 つの部分領域のそれぞれについて各部分領域のサイズと部分領域に割当てられるコンテンツデータのサイズとの差分を算出し、当該差分の合計が最小となるように、前記第 1 制約条件によって結合された前記少なくとも 2 つの部分領域のサイズを決定する決定手段と、

30

前記決定手段によって決定された部分領域のサイズに従って前記コンテンツデータを配置する配置手段とを有し、

40

前記決定手段は、前記第 1 制約条件によって結合された少なくとも 2 つの部分領域のそれぞれについて、部分領域に割当てられるコンテンツデータのサイズと前記テンプレートに含まれる各部分領域との差分を 2 乗し、当該 2 乗された差分の合計が最小となるように部分領域のサイズを決定することを特徴とするレイアウト制御装置。

【請求項 7】

前記決定手段は、前記第 1 制約条件によって結合された少なくとも 2 つの部分領域のそれぞれについて、少なくとも 1 辺の位置を変更することで前記少なくとも 2 つの部分領域のサイズを決定することを特徴とする請求項 6 に記載のレイアウト制御装置。

【請求項 8】

50

前記設定手段は、前記テンプレートにおける制約条件の1つとして、前記部分領域の最大サイズ、最小サイズを設定することを特徴とする請求項6に記載のレイアウト制御装置。

【請求項9】

前記決定手段は、前記第1制約条件によって結合された少なくとも2つの部分領域のそれぞれについて、部分領域に割当てられるコンテンツデータのサイズと前記テンプレートに含まれる各部分領域との差分を算出し、前記制約条件内において当該差分の合計が最小となるまで、変更可能な辺の位置を変更して、前記少なくとも2つの部分領域のサイズを決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】

前記設定手段は、前記テンプレートにおける複数の部分領域から第1部分領域と第2部分領域とを選択することに従い、前記第1制約条件であるストラットを設定することを特徴とする請求項6に記載のレイアウト制御装置。

【請求項11】

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をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、テキスト及び図形から構成される文書の作成、編集及び印刷のためのソフトウェア実現方法及びプロセスに関し、特に、可変データ文書の作成、編集及び印刷に関する。

【0002】

本出願は、参考として全体の内容を本明細書に取り入れられている、2003年10月15日出願のオーストラリア特許出願No. 2003905658に基づいて35U.S.C. § 119の下で優先権を主張する。

【背景技術】

【0003】

[可変データ印刷]

従来、多数の文書を印刷するときに使用される印刷プロセスは、文書の物理的表現を要求するアナログ処理であった。それらは、印刷前に作成されなければならないブロマイド及び／又は印刷プレートを含んでいた。これは、カスタム化された文書を作成するのに高い費用がかかり、多数回にわたる連続した文書印刷は、通常、全く同じ文書の多数のコピーを印刷することを含んでいたことを意味する。

【0004】

カスタム化文書は、印刷前に文書の物理的コピーを準備する必要のないデジタル印刷プロセスを使用して生成されても良い。最近になるまで、デジタルプロセスは高いコストを要していたか、又は従来のアナログ印刷プロセスにより作成される印刷物より品質の上で劣る結果しか出せなかった。

【0005】

しかし、この5年の間に、デジタル印刷の進歩により、高品質のカスタム化文書を作成するための費用は著しく低減された。そのような進歩の結果、可変データ印刷が一層普及してきた。文書の著者が個々の文書のみならず、その著者が意図している読者のために各文書がカスタム化されている何種類もの関連文書をも作成することを可能にする可変データ印刷ソフトウェアが利用できるようになっている。

【0006】

高品質カスタム化印刷を実現可能にするためには、いくつかの問題を解決しなければならない。1つの問題は印刷出力の品質である。ごく最近になって、デジタルプロセスは従来の印刷プロセスの分解能に近づくことができるようになった。もう1つの問題は、特に文書が複数の高分解能画像を含む場合に、高品質の文書をデジタル形態で表現するために

10

20

30

40

50

通常は大量のデータが要求されるということである。これにより、多数のカスタム化文書をコンピュータネットワークを介してデジタルプリンタへ電子的に送信することが実用的でなくなる。第3の問題は、通常、単純な文書と比較して、カスタム化文書の作成及び維持に、より多くの労力と技術が必要になることである。カスタム化文書は、可変データを文書テンプレートに挿入することにより作成される。文書テンプレートは、可変データを挿入するためのスロットを有する文書である。文書テンプレートは、多数の文書に共通するいくつかのデータと、テンプレートの可変データスロットにデータを挿入することにより生成される各カスタム文書をどのようにレイアウトすべきかを定義する情報とを含んでも良い。様々に異なるサイズのデータ又は欠落したデータに対応できるようにページのレイアウトを定義する必要があるため、可変データテンプレートをオーサリングすることは、単純な文書をオーサリングする場合より困難である。加えて、テンプレートとデータとの関係を定義し、それを維持しなければならない。従って、可変データ印刷技術を使用してカスタム化文書を作成するために、データベース技術が要求される場合が多い。

10

【0007】

デジタル印刷ハードウェアの性能は、劇的なコストダウンを伴いつつ、ハイエンドアナログ印刷により作成される結果に匹敵する結果をもたらすレベルまで改善されてきた。PPMLなどの新しい規格は、大量の反復データを送信することなく、複数群の関連文書をネットワークを介してプリンタへ送信することを可能にする。これにより、可変データ印刷に要求されるネットワーク帯域幅は縮小される。

20

【0008】

可変データ文書のオーサリング及び管理の複雑さの問題を容易に解決する方法はない。長年にわたり、ワード処理ソフトウェアにおいては、カスタム化印刷は「メールマージ」という未熟な形態で利用されていたが、このメールマージは個別の出力文書のレイアウトの制御を非常に粗雑な形でしか支援しない。最も進んだ可変データ印刷ソフトウェアによれば、文書ごとに異なる場合もあるデータのサイズ及び形状に合わせて調整される精巧で、融通性に富むレイアウトの作成が可能であるが、それらのアプリケーションは依然として複雑であり、その使用は困難である。

【0009】

[レイアウトモデル]

通常、テキストと図形の双方を含む文書は種類の異なるアイテムの集積として編成されている。これらアイテムはテキスト、図形、又はアイテムの集積であるグループなどの他の種類の抽象オブジェクトであると考えられる。レイアウトアイテムは、通常、矩形の形状を有するか、又は矩形の範囲を有する。アイテムをどのようにしてレイアウトすべきかを定義するためにアプリケーションごとに許可される規則がレイアウトモデルを表す。可変データ印刷アプリケーションでは、それらの規則は、アイテムのサイズが変化することによってレイアウトをどのようにして計算するかを定義しなければならない。

30

【0010】

[階層レイアウトモデル]

可変データ印刷の問題を解決する現在の方法は、通常、レイアウトを定義するために階層モデルを使用する。それらのモデルはHTML文書及びXML文書に使用されるモデルに類似している。そのようなモデルの例は、CSSボックスモデル及びXSL:FOである。そのようなレイアウトモデルにおいては、矩形レイアウトエリアの内側に矩形のアイテムを垂直又は水平に積み重ねることにより、それらのアイテムがレイアウトエリア内にレイアウトされる。

40

【0011】

水平及び垂直に積み重ねるのに加えて、いくつかのアプリケーションにより支援される別の、より一般的な代替方法では、アイテムをページ上の複数行の語句のように縦方向の列又は横方向の行として積み重ねることができる。この形態のレイアウトにおいては、1つの行又は列にはめ込むべきアイテムが多すぎる場合、新たな行又は列が開始され、余剰のアイテムは次の行又は列へ「ラップ (wrap)」される。これはワードプロセッサにおけ

50

るワードラッピングと同様である。アイテムをレイアウトする方法は、ウェブアプリケーションの場合のように、ページ又はレイアウトエリアのサイズが非常に幅広く変化するようなアプリケーションで最も有用である。可変データ印刷アプリケーションの場合には、レイアウトの品質が通常は重要であり、アイテムが次の行又は列へラップすると、レイアウトが好ましく見えなくなってしまうことが多いため、積み重ねは、可変データ印刷アプリケーションにはそれほど有用ではない。

【0012】

階層レイアウトモデルでは、複雑なレイアウトは複数群のアイテムを使用することにより定義できる。1つのグループは、他の複数のアイテムを含み、それ自体がレイアウトである矩形レイアウトアイテムである。1つのグループの中のアイテムは、そのグループの境界を規定する矩形の内側にレイアウトされる。

10

【0013】

レイアウトにおけるアイテムの位置は、例えば、垂直スタックにある1つのアイテムがその垂直スタックを含む矩形の左側に対して常にできる限り遠い位置にあるように左寄せされるような整列オプションにより制御されても良い。グループ内の全てのアイテムが同じ整列状態を有するように、整列は、そのグループと関連付けられても良いし、或はそのグループ内の各アイテムが、対応する関連整列オプションを有していても良い。センタリング、左寄せ及び右寄せを含む様々な整列オプションが一般に支援されている。

【0014】

アイテムの位置は、個別のアイテムに関連付けられるか又はそれを含むレイアウト又はグループと関連付けられるマージンにより制御されても良い。マージンは、2つの隣接するアイテムの間の最小許容距離を定義する

20

〔制約ベースモデル〕

現在の階層レイアウトモデルは、以前に開発された、より一般的な制約ベースモデルより単純に理解できる。制約ベースモデルは、コンピュータ支援デザイン（CAD）アプリケーション、ユーザインターフェース構成及びウィンドウ管理に使用されてきた。ユーザインターフェース構成アプリケーションでは、レイアウトされるべきオブジェクトはボタン又は入力フィールドなどのウィジェット（widgets）である。ウィンドウ管理アプリケーションにおいては、レイアウトされるべきオブジェクトは、コンピュータオペレーティングシステムのためのユーザインターフェースの一部としてデータのビューを表現するために使用されるウィンドウである。

30

【0015】

ユーザインターフェース構成アプリケーションでは、グラフィカルユーザインターフェース（GUI）を様々な異なる画面分解能や、様々な異なるフォント及びテキストサイズの使用などの変更に適応させるために、動的レイアウト制約が使用される。動的レイアウトは、複数のプラットフォームに対してユーザインターフェースを構築するプロセスを単純化する。

【0016】

ウィンドウ管理アプリケーションでは、ユーザは、ウィンドウが追加又は削除されるとき、又はウィンドウがサイズ又は位置を変化させるときに維持されるウィンドウ間の関係を作成するために動的制約を適用することができる。

40

【0017】

CADアプリケーションは非常に一般的な幾何学的制約を有し、レイアウトモデルはより複雑で、使用しにくく、レイアウト計算は文書レイアウトモデルの場合より遅くなる。例えば、CADアプリケーションに使用されるいくつかの制約ソルバー（solvers）は、1次方程式を使用したのでは表現できない、2本の線が平行であるということの指定を支援する。

【0018】

ユーザインターフェース（UI）アプリケーション及びウィンドウ管理アプリケーションに使用される制約ベースレイアウトモデルは、図形編集作業中に1秒間に何度もレイア

50

ウトを更新する必要があるような場合に、頻繁に対話形式で使用されるのに十分な速度を有していなければならない。そのため、レイアウト方法はそのような作業を十分に支援できるほど高速でなければならない。それらのアプリケーションは、通常、レイアウト計算を実行するために特殊化制約ソルバーを使用する。それらのアプリケーションに使用される制約ソルバーは、通常、1次方程式（即ち、1次等式）又は1次不等式、或はその双方として表される線形制約を支援する。通常、それらの制約ソルバーは、解の質を規定する目標関数を更に有する。制約ソルバーは、制約のいずれにも違反せずに目標関数の値を最小（又は最大）にすることを目的としている。目標関数も1次関数である場合、この種の問題を線形プログラムと呼ぶ。この種の問題はよく理解されている最適化問題であり、それらを解決するための周知のアルゴリズムが存在している。通常、この種の問題を解決するために、シンプレックスアルゴリズム（simplex algorithm）と呼ばれるアルゴリズムが使用される。

10

【0019】

1組の制約と、目標関数とにより定義されるレイアウト問題が与えられた場合、その問題が過剰制約であることがあり、その場合には解は存在しない。或は、問題は制約不足であるかもしれない。その場合には、多数の解が存在するか、解は1つになるであろう。レイアウトを計算可能にするためには、レイアウトモデルはレイアウトが厳密に1つの解を有することを保証する必要がある。これを実現する1つの方法は、ユーザに対して解がないこと、又は多くの解がありすぎることを報告し、ユーザに問題を修正させる。問題が過剰制約又は制約不足である理由に関してユーザに意義のある情報を提示するのは容易ではないため、この方法は一般に受け入れられる解決方法ではない。システムが過剰制約である場合、多くの制約は互いに関して同時に矛盾していると考えられる。システムが制約不足である場合には、ユーザが矛盾する制約を追加することをどのようにして防止するかは自明ではないであろう。これは、グラフィカルインターフェースを使用して制約が編集されているときに特に困難である。

20

【0020】

解が存在しないという事態を回避するための周知の方法の1つは、制約階層（constraint hierarchy）と呼ばれる形態をとるように制約の優先順位を規定する。制約に対する解が存在しない場合、解が可能になるまで、制約は逆の優先順位で無視される。

【0021】

制約不足の問題を回避するための周知の方法は、非線形目標関数を使用する。適切な厳密凸関数（convex function）が使用される場合、問題は常に1つの独自の解を有する。線形制約及び2次目標関数を有する最適化問題を解決するための周知の技法が存在するため、通常は2次目標関数を使用される。それらのうちで最も単純な方法はシンプレックスアルゴリズムの変形である。

30

【0022】

可変データ文書の生成に伴うもう1つの問題は、生成中の文書のプレビューの問題である。ワードプロセッサやデスクトップパブリッシングアプリケーションは、印刷に取り掛かる前にユーザが作業内容を全体的に評価するのに補助するために、「プリントプレビュー」機能を使用することが多い。多数の文書をプレビューしなければならない場合、これは煩雑な作業になる。

40

【発明の開示】

【発明が解決しようとする課題】

【0023】

本発明の目的は、可変文書印刷に関連する1つ以上の問題点を実質的に克服するか又は少なくとも軽減することである。

【課題を解決するための手段】

【0024】

本発明の一態様によれば、コンテンツデータを配置するための複数の部分領域を含むテンプレートと、各部分領域に配置されるコンテンツデータとに基づいて可変データ文書を

50

生成するレイアウト制御方法であって、

前記複数の部分領域の少なくとも2つの部分領域を結合する第1制約条件を設定する設定工程と、

前記複数の部分領域にコンテンツデータを割当てて割り当て工程と、

前記第1制約条件によって結合された前記少なくとも2つの部分領域に割当てられるコンテンツデータの所定方向のサイズと前記第1制約条件のサイズとの和である第1の値が、前記部分領域を配置すべき領域の前記所定方向のサイズである第2の値に収まらない場合、前記第1制約条件によって結合された前記少なくとも2つの部分領域に割当てられるコンテンツデータの所定方向のサイズと前記第1制約条件のサイズとの和である第1の値を、前記部分領域を配置すべき領域の前記所定方向のサイズである第2の値に収め、かつ、前記2つの部分領域のそれぞれについて部分領域のサイズと部分領域に割当てられるコンテンツデータのサイズとの差分を算出し、当該差分の合計が最小となるように、前記第1制約条件によって結合された前記少なくとも2つの部分領域のサイズを決定する決定工程と、

10

前記決定工程によって決定された部分領域のサイズに従って前記コンテンツデータを配置する配置工程とを有し、

前記決定工程は、前記第1制約条件によって結合された少なくとも2つの部分領域のそれぞれについて、部分領域に割当てられるコンテンツデータのサイズと前記テンプレートに含まれる各部分領域との差分を2乗し、当該2乗された差分の合計が最小となるように部分領域のサイズを決定することを特徴とするレイアウト制御方法が開示される。

20

【0025】

上記方法を実行するための装置及びコンピュータプログラムを含む、本発明の他の態様も開示される。

【0026】

次に、添付の図面を参照して本発明の少なくとも1つの実施例を説明する。

【発明を実施するための最良の形態】

【0027】

以下、添付図面を参照して本発明の好適な実施の形態を詳しく説明する。

【0028】

1. 概要

30

ソフトウェアアプリケーションとして実現されるのが好ましい可変データ文書作成及び印刷システムを開示する。このアプリケーションは、類似する形態を有するが、コンテンツは異なっている場合もあるような複数の文書を作成させる。これは、文書テンプレートの作成及び編集と、文書テンプレート上のエリアと可変データの関連付けとにより実現される。

【0029】

文書テンプレートは複数のコンテナを含んでいても良く、各々のコンテナはテキスト又は画像データなどのコンテンツを保持するように構成されている。コンテナの位置及びサイズを固定することは可能であるが、ユーザ指定規則に従って、文書ごとにコンテナの寸法又は位置を変化させても良い。それらのコンテナのコンテンツは静的コンテンツ又は可変コンテンツのいずれであっても良い（即ち、データベース等の何らかのデータソースに応じて決まる）。

40

【0030】

ユーザはデータを獲得すべき様々な種類のデータソースを指定してもよい。データソースは複数のデータレコードを含む。アプリケーションは、データを文書テンプレートのコンテナと関連付けるためのメカニズムを提供する。

【0031】

文書テンプレートと、データソースからのデータは、その後、「マージ」されて、データソースと文書テンプレートのコンテナとの関連付けに基づいて複数の文書を生成する。通常、データソースのレコードごとに、又はレコード群ごとに1つの文書が生成される。

50

その後、文書をディスクにセーブするか、又は必要に応じて印刷することができる。

【0032】

これらの構成は従来の技術と比較していくつかの独立した利点を提供する。例えば、従来の技術とは異なり、文書テンプレートを編集するときに、ユーザは常にマージされた文書の内の1つを見ることを選択できる。これは、文書テンプレートとデータが動的にマージされるためである。その結果、（文書テンプレートではなく）文書がどのように見えるかを見るために特殊なプレビューエリアへ移行することが必要である従来の技術とは異なり、テンプレートを編集している間に、最終結果としての文書のうちの1つがどのように見えるかを少なくとも見る事が可能である。

【0033】

コンテナに対する制約（コンテナがどちらに移動し、どちらにサイズを変化するかを判定する）はその場で示され、従来の技術のように別個の画面又はエリアへ移行する必要なく、コンテナ上の様々な位置をクリックすることにより、その場で編集できる。これにより、従来の技術と比較して、文書テンプレートの作成が簡略化される。

【0034】

2. 実施の形態の概要

基本的な実施の形態は、少なくともユーザインターフェースと、レイアウトエンジンとを具備する可変データ文書作成及び印刷アプリケーションである。1つの特定の実施の形態は、プリンタと共にパーソナルコンピュータで実行可能であるソフトウェアである。別の実施の形態では、アプリケーションは、可変データ文書集合の印刷を支援するプリンタ又はプリントコントローラに埋め込まれるプロセッサにおいて実行可能であるソフトウェアとして組み込まれる。更に、別の実施の形態においては、アプリケーションは、視聴者に対してカスタマイズされた文書をサーブすることができるウェブサーバで実行可能であるソフトウェアとして組み込まれる。それらの実施の形態は、適切なハードウェア構造でそのようなソフトウェアが実行される場合も含む。

【0035】

可変データ印刷とは、関連文書の集合を印刷することを表す。非可変データ印刷の場合、文書は個別に印刷され、文書ごとのデータは通常プリンタへ別個に送信される。可変データ印刷においては、複数の関連文書を含むプリントジョブがプリンタへ送信される。通常、可変データプリントジョブにおける文書は、複数の文書で出現する共有要素を含む。このデータは、通常、文書ごとに繰り返されるのではなく、ジョブごとに一度だけプリンタへ送信される。可変データプリントジョブが印刷される時、各文書に共有データが挿入される。これには、文書のページが印刷されるにつれて、ページのレイアウトを計算することが必要であろう。この場合、レイアウト方法はプリンタ又はプリントサーバのソフトウェアの一部になるであろう。より典型的には、各文書のレイアウトはクライアントマシンで計算され、レイアウトはプリントジョブの一部としてプリンタへ送信されるが、この場合、レイアウト方法はクライアントマシンで実行される。

【0036】

3. システムの説明

図1Aは、可変データ文書を印刷するためのシステム100を示す。ここで説明する方法は、図1Bに詳細を示す汎用コンピュータモジュール101の内部で実施されても良く、その場合、説明するプロセスは、コンピュータモジュール101内部で実行され、システム100を介して動作可能であるレイアウト編集アプリケーションプログラム121などのソフトウェアとして全体又は一部を実現されても良い。特に、レイアウト編集及びその結果としての印刷工程は、コンピュータ101により実行されるソフトウェアの命令により実現されても良い。ソフトウェアは、例えば、以下に説明する記憶装置を含むコンピュータ読み取り可能な媒体に格納されていても良い。ソフトウェアはコンピュータ読み取り可能な媒体からコンピュータにロードされた後、コンピュータ101により実行される。そのようなソフトウェア又はコンピュータプログラムが記録されているコンピュータ読み取り可能な媒体は、コンピュータプログラム製品である。このコンピュータプログラム

10

20

30

40

50

製品をコンピュータで使用した結果、文書レイアウト編集及び可変文書印刷のための好都合な装置が得られるのが好ましい。

【0037】

コンピュータモジュール101は、キーボード132などの入力装置及びマウス133などの指示装置と、表示装置144及びオプションであるローカルプリンタ145を含む出力装置とに接続されている。入出力インターフェース138は、ネットワーク接続107を介してコンピュータモジュール101をシステム100の他の計算装置に接続することができる。通常、ネットワーク接続107は、ローカルエリアネットワーク（LAN）又はワイドエリアネットワーク（WAN）である。

【0038】

コンピュータモジュール101は、通常、少なくとも1つのプロセッサユニット135と、例えば、半導体ランダムアクセスメモリ（RAM）及び読み取り専用メモリ（ROM）から形成されているメモリユニット136と、ビデオインターフェース137を含む入出力（I/O）インターフェースと、キーボード132及びマウス133に対するI/Oインターフェース143とを含む。記憶装置139が設けられており、通常はハードディスクドライブ140と、フロッピー（登録商標）ディスクドライブ141とを含む。磁気テープドライブ（図示せず）も使用されて良い。データの不揮発性供給源として、通常、CD-ROMドライブ142が設けられている。コンピュータモジュール101はGNU/Linux又はMicrosoft Windows（登録商標）などのオペレーティングシステムを使用し、コンピュータモジュール101の構成要素135～143は、通常、そのオペレーティングシステムに従って、当業者には知られている、それらの構成要素により形成されるコンピュータシステムの従来通りの動作モードを結果としてもたらしように、相互接続バス134を介して通信する。説明した構成を実施できるコンピュータの例には、IBM-PC及びその互換機、Sun Sparcstations又はそこから派生した類似のコンピュータシステムがある。

【0039】

通常、レイアウト編集アプリケーションプログラム121はハードディスクドライブ140に常駐しており、実行時にはプロセッサ135により読み取られ、制御される。プログラム121及びネットワーク1020から取り出されるデータの間格納は、可能であればハードディスクドライブ140と協調して、半導体メモリ136を使用して実現されても良い。場合によっては、アプリケーションプログラム121は、CD-ROM又はフロッピー（登録商標）ディスクに符号化されてユーザに供給され、対応するドライブ142又は141を介して読み取られても良く、或はネットワーク接続107からユーザにより読み取られても良い。更に、磁気テープ、ROM又は集積回路、磁気光ディスク、コンピュータモジュール101と別の装置との間の無線送信チャンネル又は赤外線送信チャンネル、PCMCIAカードなどのコンピュータ読み取り可能なカード、並びにEメール送信及びウェブサイトなどに記録された情報を含めたインターネット及びイントラネットを含む、他の適切なサイズに規定されたコンピュータ読み取り可能な媒体から、コンピュータモジュール101にソフトウェアがロードされることも可能である。以上挙げた媒体は関連するコンピュータ読み取り可能な媒体の単なる例である。他のコンピュータ読み取り可能な媒体を使用しても差し支えない。

【0040】

レイアウト編集用として名付けられてはいるが、アプリケーション121は可変データ印刷（VDP）を実行するためにも動作し、2つのソフトウェア要素を含む。それらの要素のうちの第1の要素はレイアウトエンジン105であり、これは矩形のエリアの中で所与の制約及びサイズによって、矩形及び行の位置を計算するためのソフトウェア要素である。第2に、ユーザインターフェース103要素は、ユーザに文書テンプレートを構成させ、文書テンプレート内の複数の領域をデータソースと関連付けさせるメカニズムを提供する。ユーザインターフェース103とレイアウトエンジン105は、通信チャンネル123を介して通信する。文書生成のためのデータソースは、通常、一般にデータベースアプリケーションが実行されている別のコンピュータにより形成されるデータベースサーバ1

10

20

30

40

50

17においてホスティングされる (hosted) データベース119である。ホストコンピュータ101は、ネットワーク接続107によってデータベースサーバ117と通信する。可変データ印刷アプリケーション121は、ホストコンピュータ101又は一般に別のコンピュータにより形成されるファイルサーバ115にセーブされることが可能である文書テンプレートを生成する。可変データ印刷アプリケーション121はまた、データとマージされた文書テンプレートにより形成される文書を生成する。それらの文書はホストコンピュータ101にあるローカルファイルシステムにセーブされるか、ファイルサーバ115にセーブされるか、或は直接、印刷のためにプリントサーバ109又はプリンタ113へ送信されることが可能である。プリントサーバ109は、直接にネットワーク接続できないプリンタにネットワーク能力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を介して接続されている。

10

【0041】

図2は、可変データ文書を印刷するための他のシステム例を示す図である。図2は、レイアウトエンジン105の別個のバージョン225を含むエンジンサーバ227が追加されている点を除いて図1に類似している。エンジンサーバ227は別の典型的なコンピュータである。ファイルサーバ115に格納された文書テンプレートは、レイアウトエンジン225による印刷又は他の目的のために文書を生成するためにデータベース119に格納されているデータと組み合わせられることが可能である。そのような動作はユーザインターフェース103を介して要求されても良いし、或は単に特定のレコードが印刷されることを要求するために要求されても良い。

20

【0042】

4. メインウィンドウ

図3は、メニューバー、ツールバー、作業エリア及びフローティングパレットを含むアプリケーションのメインウィンドウの一例を示す図である。

【0043】

図3を参照すると、ユーザインターフェース103は、動作中にビデオディスプレイ144に表示されるアプリケーションウィンドウ301により形成されるグラフィカルユーザインターフェースを含む。ウィンドウ301は、メニューバー302と、いくつかの実施の形態において、その場所から切り離されて画面上の様々な場所へ移動されることができるツールバーエリア303と、作業エリア306と、オプションであるフローティングパレット311と、通常はマウス133の位置又は動きと関連する場所を占めるカーソル/ポインタデバイス313とを有している。

30

【0044】

メニューバー302は、起動されたとき、現在の技術において一般的であるように、メニューオプションの階層を示すように拡張された複数のメニュー項目304を有する。

【0045】

ツールバー303は、各々がアプリケーションの特定のモードに応じて隠蔽されるか又は表示される複数のツールボタン又はウィジェット305を有する。

【0046】

オプションであるルーラー308は、作業エリアにおいてポインタ、ページ、行、マージンガイド、コンテナ又は他のオブジェクトの位置を指示するために使用される。ルーラー308は、例えば、インチ、ミリメートル又は画素などの使用される単位を数値で指示しても良い。

40

【0047】

フローティングパレット311は、可変データライブラリなどの追加機能をアクセスするために使用されることが出来る。パレット311は、そのウィンドウの移動、サイズ設定又は閉鎖を可能にする独自のウィンドウコントロール312を有する。パレット311はオプションとして常時作業エリアの前面にとどまっても良いし、或は他のオブジェクトの背後に隠されても良い。パレット311は、アプリケーションウィンドウ301の範囲内にのみ現れるように制約されていても良いし、或はその一部又は全体がアプリケー

50

ションウィンドウ 301 の外に現れることも可能である。

【0048】

図4は、コンテナ作成の例示、コンテナ間の第1の種類のストラット (struts)、アンカー (anchors) 及びスライダを含む本実施の形態に係る画面、ツール及びアイコンを例示する図である。

【0049】

図4を参照すると、少なくとも次のようなユーザ選択可能なアイコン「ボタン」を有するツールバーエリア303が示されている。

【0050】

・選択ツールボタン403：コンテナの辺を選択、移動、スケーリング、サイズ再設定及びロック／ロック解除するために使用される。コンテナは、コンテナを選択している間にコンテナの周囲に選択ボックスをドラッグするか、又はCTRLキーを押し続けることにより多重選択されることも可能である。

10

【0051】

・イメージコンテナツールボタン405：静止画像又は可変画像を保持するコンテナを作成するために使用される。

【0052】

・テキストコンテナツールボタン404：静止テキスト又は可変テキストを保持するコンテナを作成するために使用される。

【0053】

・ストラットツールボタン406：コンテナ間の距離を制御するために使用される。

20

【0054】

これらのボタンは、当該技術では良く知られているように、コンテキストセンシティブツールチップ (context sensitive tool tips) によって実現されることが可能である。

【0055】

5. 文書テンプレート

作業エリア306は文書テンプレートを見て、そのデザインを編集するために使用される。これにより、ユーザは準備中の文書が印刷された場合にどのように見えるかをデザインし、文書テンプレートとマージされるべき可変データの量及びサイズに基づいて、マージ後の各文書がどのように変化するかを理解することができる。

30

【0056】

テンプレートに外部データソースがリンクされている場合、可変テキスト及び可変画像は、ユーザが作業中に現文書を下見できるようにコンテナ内に表示される。

【0057】

文書の構造及びその可変データコンテナの挙動を記述するビジュアルクルーは、ユーザがカーソルをロールオーバーするか又はコンテナを選択するたびに表示される。

【0058】

作業エリア306は、スクロールバー307と、オプションであるルーラー308と、文書テンプレート309とを特徴としている。文書テンプレート309は複数のページを示すことができる。

40

【0059】

所定の文書テンプレートのページサイズは、当該技術において良く知られているように、ユーザにより指定される。各文書の実際のページ数は、可変データに応じて異なるであろう。1ページに全てが収まらない場合には、データを表示するために追加のページが自動的に作成される。

【0060】

各ページ境界の中に、そのページにおける印刷可能オブジェクトの最大範囲を指示するオプションのページマージン310がある。

【0061】

図4には、文書テンプレート309の1ページに現れることが可能であると思われる複

50

数のオブジェクトの例も示されている。それらはオプションであるアンカーアイコン 4 0 9、非固定辺 4 1 0、ストラット 4 1 2 及びスライダ 4 1 3 を特徴とする複数のコンテナ 4 0 7、4 0 8 である。

【0062】

6. コンテナ (Containers)

コンテナは、文書テンプレートの中の、テキスト、画像及び他のコンテナ又はオブジェクトなどの静的コンテンツ又は動的コンテンツを挿入できるスペースである。コンテナは、ユーザインターフェースに示すように、ポインタ 3 1 3 及びマウス 1 3 3 を介する制御を使用したコンテナの操作により、移動、スケーリング及び再整形されることができであろう。

10

【0063】

更に厳密に言えば、コンテナは設定、視覚的見かけ、並びに対話及び編集の挙動の集積である。以下に、コンテナの定義の全ての部分を示す。

【0064】

・コンテナは、静的及び／又は動的コンテンツを有する。動的コンテンツは、データソースに由来し、異なる文書に対して異なっていることもあるという意味で動的である。動的コンテンツは、印刷には適合しないような、時間の経過に伴って変化するアニメーション又は他のコンテンツを含むことを意図されていない。同様に、静的コンテンツは、このコンテナを使用して生成される全ての文書において同じに見えるが、動的コンテンツのアクションによって、静的コンテンツが文書ごとに異なる場所に配置されることがある。

20

【0065】

・コンテナは、コンテナのコンテンツに適用される背景色、境界、フォントや文体などのテキスト設定のような装飾特徴を有していても良い。

【0066】

・コンテナは、文書を生成するときにデータソースからのデータとマージされても良い。装飾特徴は、通常、静的コンテンツのように印刷済み出力の中に見ることができる。動的コンテンツは、データソースからの特定のデータの見かけに表れる。このようなコンテナの表現は、例えば、印刷されるか又は画面 1 4 4 に表示されるか、或は印刷と表示の双方が実行されることが可能であろう。

【0067】

・コンテナはユーザインターフェース、例えば、コンテナの設定を編集し、見るための対話型グラフィカルユーザインターフェースを更に有していても良い。インターフェース要素は、通常、画面 1 4 4 に現れるが、印刷される文書には表れない。ユーザインターフェース 1 0 3 は、背景色又はフォントなどのコンテナの装飾特徴の一部を表示しても良く、また、コンテナの設定を編集し、見ることを可能にするための特徴を更に追加しても良い。特殊目的用ユーザインターフェースの例によれば、コンテナのサイズ又は位置を表示し、対話によって変更するための境界又は角アイコン、或はコンテナがデータソースからのデータとマージされるときにコンテナの挙動を指示するためのオーバーレイ数、行、アイコン又はテキストを備えることが特徴となっている。

30

【0068】

本実施の形態の一態様によれば、一連の新たな直接操作技法と、コンテナのグラフィカルユーザインターフェース要素を含む表示の方法が開示される。

40

【0069】

6. 1 コンテナ制約

本実施の形態によれば、コンテナは、各文書において関連コンテンツがどのように表示されれば良いかを制御するいくつかの制約を有していても良い。それらの制約は、静的コンテンツ及び動的コンテンツをコンテナと関連付ける手段と共に、ユーザが単一の文書テンプレートからの複数の文書の生成を制御するための第 1 の方法である。制約の一例は、「このコンテナのコンテンツは最大で 4 インチの高さである」というものであろう。別の制約は「このコンテナの左辺は各文書において同じ水平位置に現れなければならない」と

50

いうものであろう。ここで説明するのは、グラフィカルユーザインターフェースを使用してそのような制約を表示し、編集するための一連の方法である。

【0070】

1 ページ上で何らかの定義済み場所を有する画像のように、静的コンテンツの場所を指定するコンテンツプレースホルダ (Placeholders) は、デジタル印刷技術においては良く知られている。以下の説明中、コンテナは位置とサイズを有し、当該技術で知られている方式に類似する方式で表示され、編集されると仮定するが、本実施の形態は可変データ印刷に特定した表示及び編集の方法に集中している。

【0071】

コンテナは、ユーザが文書中のコンテンツのサイズ及び位置を指定することを可能にする。単一の文書テンプレートからいくつかの文書を生じ得るので、コンテナは多数の可能性及び制約を指定し、表示するためのユーザインターフェースを有していなければならない。

10

【0072】

コンテナの辺 (稜線) は、関連するコンテンツが文書内で現れる仮想境界を定義する。従って、本明細書において、コンテナの左辺というときには、それは生成されるどの文書にあっても関連するコンテンツが表示されることが可能である場所の最も左側の辺というのと同様であろう。同様に、コンテナの高さというときには、それは生成されるどの文書にあっても関連するコンテンツの高さに対する制約を論じていると理解されるであろう。本願明細書がユーザインターフェース 103 に関連してコンテナの辺又はサイズを論じる場合、この区別が明確にされるであろう。

20

【0073】

以下の説明中、「固定 (fixed)」という用語は、コンテンツの見かけを制御するために使用される何らかの値が全ての文書において同じであることを定義している。

【0074】

・コンテナの幅が固定されている場合、それは関連するコンテンツに許可されている幅が全ての文書において同じであることを意味する。

【0075】

・コンテナの高さが固定されている場合、それは関連するコンテンツに許可されている高さが全ての文書において同じであることを意味する。

30

【0076】

・距離制約が固定されている場合、全ての文書について指定距離が一定である。

【0077】

・コンテナの左辺又は右辺が固定されている場合、それはページに対して全ての文書について、その辺の水平 (横) 方向の位置が同じであるが、コンテナの高さ或は縦方向の位置は変化しても良いことを意味する。例えば、コンテナの左辺が固定されている場合、関連するコンテンツは 1 つの文書ではページの最上部近くに現れ、別の文書ではページの一番下に現れることも可能であるが、左辺は全てのケースにおいて同じ水平位置を有する。

【0078】

・コンテナの上辺又は下辺が固定されている場合、それはページに対して全ての文書について辺の縦方向の位置が同じであるが、コンテナの幅或は水平方向の位置は変化しても良いことを意味する。

40

【0079】

・コンテナの縦軸は、コンテナの左辺と右辺との中間に位置し、それらの辺と平行である想像上の垂直 (縦) 線である。コンテナの縦軸が固定されている場合、そのコンテナの左辺と右辺の水平方向位置の平均は全ての文書について同じである。この制約があると、コンテナの幅は変化でき、従って、様々に異なる文書において、左辺と右辺は共に縦軸から遠ざかるか、又は縦軸に接近する場合もあるが、この縦軸は全ての文書について、同じ水平位置に留まる。コンテナの高さと縦方向位置はこの制約によって影響を受けない。

【0080】

50

・同様に、横軸が固定されている場合、これはコンテナの上辺及び下辺が縦方向に位置する場所を制約するが、高さはこの制約によって影響を受けない。

【0081】

・横軸と縦軸の双方が固定されている場合、それはコンテナの中心点が固定されていることを意味するが、幅と高さはこの制約によって影響を受けない。

【0082】

・コンテナの1つの角、コンテナの1つの辺の中間点、又はコンテナの中心点が固定されている場合、それは全ての文書において、その点が同じ場所に現れ、コンテナに対して同じ場所に現れることを意味する。例えば、コンテナの左上角が固定されている場合、それはそのコンテナのコンテンツが配置されるであろう場所の左上の点が全ての文書について同じであることを意味する。

10

【0083】

・縦方向の辺又は縦軸はページの左側の辺、又は右、又は左ページマージン、又は右ページマージン、或は他の何らかの水平位置に対して固定されていても良い。同様に、横方向の辺又は横軸は、ページの上辺又は下辺又はマージン、或は他の何らかの縦方向位置に対して固定されていても良い。「固定」という用語をこのように微細に区別することは、ページサイズが文書間で異なる場合にのみ重要である。それは、ページサイズが全ての文書で同じであれば、それらの可能性は生成される文書に相違を生み出さないからである。

【0084】

「固定」の反対は「非固定」であり、これは辺、軸、角、中間点又は距離制約が文書ごとに異なっても良いが、特定の文書の集合においては必ずしもそうではないことを意味している。例えば、実際には辺の位置が変化することを阻止する他の外部制約は存在するであろうが、それらの外部制約が適用されなければ、辺の位置は変化することが可能であろうという事実は、辺がそれでも依然として非固定とラベル付けされることを意味している。

20

【0085】

6. 2 コンテナ状態

コンテナの「辺状態 (edge state)」は、先に説明したように「固定」又は「非固定」とすることができる8つ特性、即ち、左辺、右辺、縦軸、上辺、下辺、横軸、幅及び高さの1セットとして定義される。辺は最小又は最大コンテナサイズ設定、或は他のコンテナの位置などの他の制約により影響を受けることもあり得るので、辺状態は辺の位置を完全には制約しない。従って、「辺状態」は、コンテナが持ちうる8つの状態全てのうちの一部のみを表しているにすぎない。

30

【0086】

コンテナのグラフィカルユーザインターフェースは、辺状態を図形表現にマッピングすることと、辺及びコンテナ表現の直接操作を辺状態への調整にマッピングすることとから一部分取り出されるため、多くの実施の形態において辺状態は重大である。

【0087】

コンテナの完全状態は、辺状態以上のものを含むのが好ましい。コンテナは、辺位置を更に制御する幅及び高さに関する最小設定及び最大設定を有していても良い。内側マージンは、関連するコンテンツが表示されうる場所を制御し、これはコンテナの辺を配置できる場所に影響を及ぼすことになる。テキストのフォント、文体、サイズ及び整列の設定、並びに画像のクロッピング (cropping)、スケーリングなどもコンテナの状態の一部となりうると考えられ、コンテンツ位置、サイズ及び見かけに影響を及ぼすであろう。背景色及び境界色、境界線重み、型などの装飾設定も各コンテナの状態の一部である。そのような面は、文書におけるコンテンツの位置には影響を及ぼさないであろうが、見かけには影響する。

40

【0088】

実施の形態によっては、辺状態は固定又は非固定のいずれかをとり得る上述の8つの特性を含む。ページサイズは変化できないので、ここでは「固定」という用語は、それ以上

50

の修飾子を持たない。それらの8つの特性は 2^8 通りの異なる辺状態をもたらし、合わせて256をとり得る。

【0089】

それら256の辺状態の全てが使用される必要はない。例えば、コンテナの左辺が固定され、縦軸が固定され且つ右辺は非固定である場合を考える。縦軸は固定されているので、左辺から縦軸までの距離は全ての文書について右辺から縦軸までの距離と等しくなければならない。左辺も固定されているので、縦軸から左辺までの距離は全ての文書について一定であり、それは右辺も固定されていることを示唆する。

【0090】

いくつかの実施の形態は、各々が独自の挙動を示す辺状態の部分集合をモデル化している。先に説明した辺状態の挙動は、他のいくつかの状態と同一であるため、その辺状態が全ての実施の形態で使用されるわけではない。別の実施の形態はそのような状態を採用し、ユーザインターフェースにおいて、独特のコンテナの挙動をモデル化するのではなく、辺が固定されているか否かをモデル化し、編集する手段として、それらを表現しても良いであろう。

10

【0091】

特定の1つの実施の形態は6つの特有の水平（横方向）制約と、6つの特有の垂直（縦方向）制約との積として形成される36通りのコンテナ状態を使用する。

【0092】

コンテナに対する水平制約は次の通りである。

20

【0093】

1. 左辺、右辺、縦軸及び幅が非固定である；
2. 左辺のみが固定されている；
3. 右辺のみが固定されている；
4. 左辺と右辺が共に固定されている；
5. 縦軸のみが固定されている；及び
6. 幅は固定されているが、左辺、右辺及び縦軸は非固定である。

【0094】

コンテナに対する垂直制約は次の通りである。

【0095】

1. 上辺、下辺、水平軸及び高さが非固定である；
2. 上辺のみが固定されている；
3. 下辺のみが固定されている；
4. 上辺と下辺が共に固定されている；
5. 横軸のみが固定されている；及び
6. 高さは固定されているが、上辺、下辺及び横軸は非固定である。

30

【0096】

「固定」という用語が修飾子を有する場合、例えば、各々の辺又は軸を、ページの左辺又は右辺のいずれかに対して固定することが可能であろうという場合、特有の辺状態は36通りを超えるであろう。

40

【0097】

尚、上記のリストの中で「左辺と右辺が共に固定されている」とラベル付けされた制約は、「左辺、右辺、縦軸及び幅が全て固定されている」及び「左辺及び縦軸が固定されている」、並びに実際にいくつかの他の類似の制約と同じ挙動を有することに注意されたい。「固定」の様々な概念を表現するためにビジュアルキューを使用する実施の形態は、辺状態がより少ない数の「固定」制約を採用すれば、より少ない数のそのようなキューを使用でき、それにより、そうでない場合と比較して生成する視覚的混乱が少なくなるため、一般にはより単純で対称的な制約が好ましい。

【0098】

本開示の一態様は、以下に説明するように、辺状態をコンテナの図形表現及び編集方法

50

と関連付ける。

【0099】

6. 3 コンテナの表示及び編集

6. 3. 1 新たなコンテナを作成する方法

2つのクラスのコンテナ、即ち、テキストコンテナ及びイメージコンテナを説明する。テキストコンテナはテキスト及び／又は埋め込み画像を保持する。イメージコンテナは画像のみを保持する。

【0100】

図4を参照すると、新たなテキストコンテナ及びイメージコンテナは、文書テンプレート309において、マウス133を使用してテキストコンテナツール404又はイメージコンテナツール405をそれぞれクリックし、その後、テンプレート309内で矩形をドラッグすることにより作成できる。

10

【0101】

或は単純に、適切なツール404、405を起動した後に、文書テンプレート309の中をクリックすることによりコンテナを作成しても良い。デフォルトサイズのコンテナが挿入されるか、或は新たなコンテナの寸法を入力するためのダイアログボックス又は他のプロンプトが提供されるかのいずれかである。コンテナによっては、何らかの事前に定義済み又は計算済みスキーマに従って自動的に作成、配置されても良い。その他の代替方式を考案しても良い。

20

【0102】

6. 3. 2 コンテナを表示する方法

先に説明した36通りの辺状態の各々が図形表現へマッピングされるのが好ましい。いくつかの辺状態は、状況によっては1つの表現を共有することもあるため、図形表現の数は36通り未満になる場合もある。

【0103】

図5(A)～(D)は、1つのコンテナに関する第1の辺規則の例を示す図である。

【0104】

アプリケーション121は、辺を実線(アイテム503を参照)又は破線(504を参照)として描画し、更に、辺状態を表現するために、アンカー(辺の上又はその付近に描画され、506、507、509により示すように複数の線、形状及び／又はアイコンから構成されている)、ハンドル(辺又は形状を移動又は修正させるためにその上又はその付近に描画されるコントロールポイント、502を参照)、スライダ(辺の両側に描画される短い平行線、図4を参照、413とラベル付けされたアイテム)、成長アイコン(505を参照)及び色を特徴として有する。

30

【0105】

図5(A)～(D)のコンテナ表示方法の規則は順に次の通りである。

【0106】

1. 固定された辺ごとに、その辺を実線で描画する；
2. 幅が固定されている場合、左辺及び右辺を実線で描画する；
3. 高さが固定されている場合、上辺及び下辺を実線で描画する；
4. 縦軸及び横軸は描画されない；
5. まだ描画されていない全ての辺は破線で描画され、各辺の付近に成長アイコンが描画される；
6. 縦方向の辺及び／又は軸の対ごとに、それらが共に固定されているならば、その交差点にアンカーを描画する；
7. 固定された辺ごとに、その辺の上のどこにもアンカーが描画されていなければ、辺の中心にスライダを描画する；
8. 縦方向の辺及び／又は軸の対ごとに、その場所にアンカー又はスライダが描画されていなければ、その交差点にハンドルを描画する。

40

【0107】

50

規則 1、2 及び 3 は、線が固定又は制約されている場合に、それらが実線で描画されることを保証する。規則 5 は非固定の辺が破線で描画されることを保証する。規則 6、7 及び 8 は、固定された点はアンカーを表示し、固定された辺のいくつかはスライダを表示し、その他はハンドルを表示することを保証する。

【0108】

上記の規則において、辺は一度描画されるだけで良く、従って、ある規則が辺を描画させたならば、その後の規則はその辺を再度描画させない。アイコンを別の方法で描画するか又は省略するほうが好都合である場合、例えば、コンテナが非常に小さく、アイコンが互いに重なり合うか又は表現の他の特徴を分かりにくくする虞れがある場合には、そのようにしても良い。

10

【0109】

非固定の辺が描画される厳密な場所は、コンテナコンテンツの内容によって決まるであろう。先に説明したように、コンテンツが文書テンプレートとマージされ、ユーザーインターフェースに目に見えるように表れることを意味する「ライブプルーフイング (live proofing)」が使用される。それに代わる実施の形態は、全ての文書について平均されたコンテナのコンテンツエリアを使用するか、又はユーザーインターフェースにおいて非固定の辺が配置されるべき場所を決定する他の何らかの手段を使用することができる。

【0110】

これらのコンテナの表現は、コンテナの辺状態を表示する図形的方法を提供する。表現の解釈は次の通りである。

20

【0111】

・破線は、その辺の文書内における位置がコンテナのコンテンツに応じて決まることを意味する。図 4 において、そのような辺は 410 とラベル付けされている。

【0112】

・実線は、その辺が固定されている（辺 414 など）か、又はコンテナの幅又は高さが固定されている（コンテナ 408 では共に固定されている）ために、辺が制約されていることを意味する。

【0113】

・アンカーは、そのアンカーと交わる辺及び／又は軸が固定されていることを意味する。従って、アンカーポイントは、全ての文書において同じ横及び縦方向位置に現れる。そのため、定義上、アンカーは固定されている。図 4 のアイコン 409 は、交わる辺 414 が固定されていることを指示するアンカーアイコンの一例である。

30

【0114】

・スライダは、関連する辺が固定されているが、コンテナはその辺に沿って「縦方向にスライドして」数多くの位置に配置されて良いことを意味する。例えば、図 4 において、スライダ 413 は、コンテナ 408 のコンテンツが文書の中で、その特定の線図に見られる位置の左又は右に現れて良いことを指示している。

【0115】

これらのアイコン又は辺の一部又は全ては、どのツール又はどのコンテナが選択され、ハイライトイングされ、又はその他の方法で起動されるかに応じて描画されるか、又は描画されない。コンテナ辺及びアイコンは、一般に、文書テンプレートのデザインを補助する手段であるので、印刷される文書では描画されない。

40

【0116】

最小幅、最大幅、最小高さ及び最大高さなどの設定は、2 次ダイアログウィンドウに表示されることが可能である。

【0117】

図 5 (A) において、コンテナ 501 の幅及び高さは共に非固定である。固定された辺 503 は実線で表されており、非固定の辺 504 は破線で表されている。成長アイコン 505 は、隣接する辺 504 が非固定であることを示す追加標識又は代替標識である。

【0118】

50

図5（B）において、コンテナ501の幅と高さは共に非固定である。アンカーアイコン506は、そのアイコンと交わる2つの辺503が固定されていることを付加的に又は代替的に示す。

【0119】

図5（C）において、コンテナ501の幅と高さは共に非固定であり、オプションのアンカーアイコン507で示すように、中心点の周囲でコンテナの拡張又は縮小が均等に起こる。

【0120】

図5（D）において、上辺508が固定されていることを除いて、コンテナ501の幅と高さは非固定である。アンカーアイコン509は、中心にアイコンが位置している上辺508が固定されていることを示すと共に、コンテナの左右の辺がアイコンを垂直に通って描画された中心軸線（垂直軸）に関して拡張又は縮小することを示す。

【0121】

6. 3. 3 コンテナ特性をその場で編集する方法

マウス133及びポインタ313によっていずれかの辺503又は504の上又はそれに近接する場所をクリックすることにより、辺503／504は固定状態と非固定状態との間でスワップされ、それに従ってユーザインターフェース301の図形表現が更新される。

【0122】

マウス133とポインタ313との組み合わせを使用して、コントロールポイント502をドラッグすることにより、対応する1つ以上の辺をコントロールポイント502の位置に従わせても良い。マウス133／ポインタ313によって選択場所からコントロールポイント502を解放すると、ドラッグされてきた位置に固定辺はとどまるが、その場合、新たな位置を無効にさせる外部制約は存在せず、一方、非固定の辺はコンテナ内部のコンテンツのサイズ及び形状などのファクタと、コンテナ又は辺に適用され得る他の何らかの制約により判定される計算上の位置に戻る。後述する、非固定の辺及びコンテナの位置を判定するために使用されるアルゴリズムの説明を参照されたい。

【0123】

非固定の辺の中心又は2つの非固定の辺の交差点に位置するコントロールポイントは固定されるか又は全く示されなくても良く、1つの非固定の辺の両端部に位置するコントロールポイントは、その辺と平行な方向に移動するように制限されても良い。

【0124】

更に、コンテナの頂点又は辺の付近をマウス133によってクリックするか、又はその頂点に位置するコントロールポイント又は辺の中央をクリックすることにより、その頂点又は辺にアンカーを追加しても良い。また、アンカーアイコンの付近をクリックするか、或はアンカーを選択し、次にインターフェースでメニューアイテム、ボタン又は他のコントロールを起動することにより、アンカーを除去しても良い。

【0125】

このようにしてアンカーを追加すると、その結果、アンカーと交わる1つ以上の辺又は軸は固定される。

【0126】

外部からの他の制約がない限り、コンテナは少なくとも1つの固定されたポイントを有することが望ましい。それは、この条件がないと、横方向位置又は縦方向位置の定義が不十分であるようなコンテナの作成が可能になると考えられるからである。例えば、固定された辺又は軸を持たないコンテナが有効である外部制約を持たない場合、そのコンテナのコンテンツは、論理上はページ上のどの場所にも出現できることになってしまうであろう。これは、そのようなコンテナのコンテンツが文書のどこに現れるかをユーザが予測することを困難にするであろう。そのため、実施の形態によっては、コンテナをそのような定義の不十分な状態に置くような遷移を許可しないことにより、この可能性を防止している。

【0127】

先に説明したように、コンテナが外部制約を有しているならば、コンテナは対応する固定された辺を持たずに幅又は高さが固定されていても良い。例えば、コンテナの横方向の位置を何らかの水平制約により確定できるのであれば、コンテナは左右の辺が非固定であっても固定された幅を有することが可能であろう。そのような外部制約については以下に説明するが、それらは「ストラット (struts)」と呼ばれる。

【0128】

コンテナに外部制約ストラットが付随しているならば、先に説明した、辺を固定するための方法に類似する方法で辺上をクリックすることにより、幅又は高さを固定できる。この状況では、コンテナに横ストラットが結合されている場合、非固定の左又は右の辺の上又はその付近をクリックすることにより、コンテナの幅が固定され、コンテナに縦ストラットが結合されている場合には、非固定の上又は下の辺の上又はその付近をクリックすることにより、高さが固定される。

10

【0129】

辺、軸、幅及び高さの固定／非固定を操作することは、コンテナの辺状態の間の遷移に相当する。固定された辺又は角の位置を変更するなどの他の操作は、他のコンテナ状態情報を修正する。

【0130】

各コンテナが常に有効辺状態を有することを保証するために、コンテナに制約が追加されるか又はコンテナから制約が除去されるたびに1組の規則が適用され、問題を解決するように他の制約に対して必要な調整が実行されるのが好ましい。新たな制約構成を示すように画面表示を更新することにより、調整は直ちにユーザに対して示される。これにより、ユーザインターフェース103は、コンテナが望ましくない状態になることを防止する。

20

【0131】

規則は各コンテナの横方向寸法及び縦方向の寸法に個別に次のように適用される。

【0132】

1. 外部制約によりいずれかの辺が固定されるか又は作用を受けた場合、その辺と平行な軸は非固定になる。

【0133】

2. 軸が非固定になった場合、少なくとも1つの平行な辺は外部制約により固定されるか又は作用を受ける。

30

【0134】

3. 対向する辺が非固定になり、それらの辺に対する全ての外部制約が除去された場合、それらの辺と平行な軸は固定される。

【0135】

4. 軸が固定された場合、その軸と平行な辺は非固定になり、それらの辺に作用する外部制約は除去される。

【0136】

5. 幅が固定された場合、左右の辺と縦軸は非固定になる。

40

【0137】

6. 高さが固定された場合、上下の辺と横軸は非固定になる。

【0138】

6. 3. 4 代替コンテナユーザインターフェース：別個の制約

コンテナユーザインターフェースの別の(第2の)実施の形態は、コンテナの辺状態を目で見て明確に区別でき、別個に編集可能な制約に分割するように、コンテナを表示し、コンテナを編集させる。

【0139】

この実施の形態では、幅及び高さの制約はコンテナのコンテンツエリアを横切るバーにより指示される。辺は辺の固定又は非固定の性質を示すだけである。

50

【0140】

コンテナ表示方法に従った第2の実施の形態の規則は順に次の通りである。

【0141】

1. 辺又は軸ごとに、その線が固定されていれば、それを実線で描画し、そうでない場合には破線で描画する；

2. 幅が固定されている場合、コンテンツエリアに実線の幅バーを描画し、そうでない場合にはそれを破線で描画する；

3. 高さが固定されている場合、コンテンツエリアに実線の高さバーを描画し、そうでない場合にはそれを破線で描画する；及び

4. 縦方向の辺及び／又は軸の対ごとに、それらが共に固定されていれば、その交差点にアンカーを描画し、そうでない場合にはハンドルを描画する。

10

【0142】

前述したように、第2の実施の形態においては、実線は、既にスライダがなすであろう挙動と同じ挙動を示すため、スライダの必要はない。先の実施の形態は、辺の固定／非固定に加えて、幅及び高さの制約を指示するために実線を使用していたので、これは先の実施の形態には当てはまらない。

【0143】

第2の実施の形態の幅バー及び高さバーは、コンテナに対する幅及び高さの制約をそれぞれ表示し、編集する図形的手段である。バーは辺に似た外観を呈し、固定及び非固定をそれぞれ示すために実線又は破線の見かけを有する。図38には、2つのコンテナ3802及び3803に渡された幅バー3805及び高さバー3806が示されており、その横方向の位置と縦方向の位置は、コンテナ3801のサイズと、結合ストラット3804の長さによりそれぞれ決定される。オプションとして、固定された辺などの他の制約が幅又は高さの固定を無意味にするであろうと考えられるならば、バーは不活動状態(inactive)であるか又は描画されなくても良いであろう。例えば、コンテナの左辺が固定され、右辺が非固定である場合、ユーザが幅を固定してしまうと、コンテナは先に説明した36通りの好ましい辺状態の集合から外れた辺状態をとることになるため、そのような事態を阻止することが望ましいであろう。そのような状況では、幅バーは示されないであろう。

20

【0144】

この第2の実施の形態において、コンテナを編集する場合には、マウス133及びポインタ313を使用して辺、軸、幅バー又は高さバーをクリックして、辺を固定から非固定に、又は非固定から固定に変化させる。辺、ハンドル又はアンカーをドラッグすることは先に説明したように作用し、アンカーをクリックすると、それと交わる非固定の辺又は軸が固定され、交わる線が共に既に固定されている場合には、それらが共に非固定になる。これにより、横方向の位置又は縦方向の位置のいずれかの定義が不十分であるようなコンテナが残された場合、中心軸の一方又は双方を固定することで、この問題を修正できるであろう。例えば、アンカーをクリックしたことにより、コンテナの全ての辺が非固定のままである場合、コンテンツの場所として複数の場所が可能であり、従って、この問題を修正するために、その縦軸と横軸の双方を固定することによりコンテナの中心点を固定することが可能であろう。他の外部制約又は規則により、この過程が不必要になるか、又は1つの軸のみを固定することで十分な場合もあると考えられるであろう。

30

40

【0145】

6. 3. 5 代替コンテナユーザインターフェース：サイズフレーム

別の(第3の)実施の形態においては、コンテナの挙動は、各々が別個の矩形フレームにより表現される最小範囲及び最大範囲の直接操作により制御される。

【0146】

図6A～図6Cは、コンテナに関する第3の例示的規則を示す。

【0147】

図6Aを参照すると、コンテナ601は各頂点と、オプションとして各辺の中央とにコントロールポイント603が配置されている最小範囲フレーム602と、各頂点と、オブ

50

ションとして各辺の中央とにコントロールポイント605が配置されている最大範囲フレーム604とを具備する。

【0148】

最小範囲フレーム602は、常に、横方向と縦方向の双方で最大範囲フレーム604の内側に現れるか、又はそれと一致するように現れる。

【0149】

第3のフレーム606は、コンテナ601の内側のコンテンツの実際の範囲を現す。このコンテンツフレーム606は、常に最大範囲フレームの内側に現れるか、又はそれと一致するように現れると共に、常に最小範囲フレームの外側に現れるか、又はそれと一致するように現れる。コンテンツフレーム606の好ましいサイズは、コンテナ601の内側のコンテンツのサイズ及び形状などの要因、並びにコンテナに適用されうる他の何らかの制約により判定される。

10

【0150】

ユーザがこれら3つのフレームを互いに容易に区別できるようにするために、各フレームとそのコントロールポイントはオプションとして、異なる線重み、型又は色により表現されても良い。

【0151】

1つ或はそれ以上のコンテナを含む文書テンプレートを見るときに視覚的混乱 (clutter) を減少させるために、最小範囲フレーム及び最大範囲フレームと、それらに関連するコントロールポイント603、605及びコンテンツフレーム606は、オプションとして、それらが関連している特定のコンテナが「活動状態 (active)」にされない限り、例えば、コンテナにより占められているエリアの中にポイントを配置することにより、又はマウスによってコンテナをクリックすることにより隠蔽されても良い。

20

【0152】

最小範囲フレーム602と最大範囲フレーム604は、マウス133を使用してそれぞれ対応するコントロールポイント603、605を直接操作することによりリサイズ (サイズを変更) 可能である。最小範囲フレーム602をリサイズすることにより、ユーザは、コンテナ601の内側にどのようなコンテンツが現れるかに関わらず、コンテンツフレーム606の可能最小サイズを決定できる。最大範囲フレーム604をリサイズすることにより、ユーザは、コンテナ601の内側にどのようなコンテンツが現れるかに関わらず、コンテンツフレーム606の可能最大サイズを決定できる。

30

【0153】

最小範囲フレーム602をリサイズするとき、フレームのいずれかの頂点又は辺の位置が最大範囲フレーム604の外側までドラッグされると、コントロールポイント603が解放された後、その頂点又は辺は、最大範囲フレーム604の対応する頂点又は辺と等しくなるように自動的に移動されるか、或は頂点又は辺が最大範囲フレーム604の対応する頂点又は辺に到達した後にそれ以上移動することを停止するか、或は最大範囲フレーム604の対応する頂点又は辺が、ドラッグされている頂点又は辺と一致するか又はその外側に出るように自動的に拡張される。

【0154】

40

最大範囲フレーム604をリサイズするとき、フレームのいずれかの頂点又は辺の位置が最小範囲フレーム602の内側までドラッグされると、コントロールポイント605が解放された後、その頂点又は辺は最小範囲フレーム602の対応する頂点又は辺と等しくなるように自動的に移動されるか、或は頂点又は辺が最小範囲フレーム602の対応する頂点又は辺に到達した後にそれ以上移動することを停止するか、或は最小範囲フレーム602の対応する頂点又は辺が、ドラッグされている頂点又は辺と一致するか又はその内側に来るように自動的に縮小される。

【0155】

最大範囲フレーム604内におけるコンテンツフレーム606の位置は、最大範囲フレーム604内における最小範囲フレーム602の相対位置に従って確定される。

50

【0156】

最小範囲フレーム602は、マウス133によってそれをドラッグすることにより、最大範囲フレーム604内の任意の位置へ移動できる。

【0157】

オプションとして、最小範囲フレーム602は、マウス133によってドラッグされている間に、左、右及び横方向の中心位置合わせ、並びに上、下及び縦方向の中心位置合わせを含めた最大範囲フレーム604内の好ましい位置の選択肢の内の最も近接する位置に「スナップ(snap)」するようにプログラムされても良い。

【0158】

図6Aにおける横(水平)方向の寸法a, b, c及びdは、最大範囲フレーム604、コンテンツフレーム606及び最小範囲フレーム602のそれぞれの縦方向の辺の距離を示す。

10

【0159】

ある特定の時点におけるフレーム内のコンテンツにより判定されるコンテンツフレーム606のサイズが与えられれば、寸法a, b, c及びdは、常に下記の式に従う。

【0160】

$$a : b = d : c$$

図にはラベル付けされていないが、対応する縦方向の寸法にも対応する式が適用される。

【0161】

従って、最大範囲フレーム604内におけるコンテンツフレーム606の位置は式を解いた後の横方向の寸法a, b, c及びdと、対応する縦方向の寸法とに従って決定される。

20

【0162】

図6Bは、最小範囲フレーム602が最大範囲フレーム604の左上に位置しており、従って、コンテンツフレーム606も同様に最大範囲フレームの左上に位置しているような1つの特定のケースを示す。

【0163】

図6Cは、最小範囲フレーム602が横方向と縦方向の双方で最大範囲フレーム604内の中央に位置し、従って、コンテンツフレーム606も横方向と縦方向の双方で最大範囲フレーム604内の中央に位置している別の特定のケースを示す。

30

【0164】

図6D及び図6Eは、コンテナ範囲が使用される可変データ文書を生成する方法620を説明するフローチャートである。この方法620は、アプリケーション121の中で実現され、スタートポイント622の後、ステップ624では文書テンプレートが検索され、ディスプレイ144のGUI301の中に表示される。ステップ626は、通常はマウス133及び対応するポインタ313の動きを介してユーザによりコンテナの描画を検出する。

【0165】

描画されるとき、コンテナは、デフォルトとして、図6A～図6Cに示すような最小コンテナ範囲602及び最大コンテナ範囲604という2つの制約を有する。コンテナの初期状態では、最小コンテナ範囲602は最大コンテナ範囲604と一致している。ステップ634では、いずれの範囲のコントロールポイントもコンテナ範囲を修正するために操作されることができであろう。

40

【0166】

コンテナはテンプレートにおいてレイアウトを形成するために表現され、その後、ステップ628で様々な制約が適用される。ステップ628は、レイアウト中のコンテナの中でここで説明するような数多くの種類の制約の組み込みを含んでも良い。現在論じている実施の形態においては、適用されるべき制約は図6A～図6Cに示すようなコンテナ範囲の制約である。ステップ630で、再びマウス133及びポインタ313を使用して

50

コンテナ範囲が描画され、ステップ632では、コンテナ範囲はレイアウト中のコンテナと関連付けられる。ステップ634では、コンテナ範囲を修正するために、範囲のコントロールポイントが選択、操作されても良い。そのような修正は、範囲の1つ以上の辺をコンテナの辺と一致するように設定すること（図6B）、及び範囲フレームを関連するコンテナに対して移動又は位置決めすることを含んでいても良い。

【0167】

次のステップ636では、ユーザはステップ628に戻ることににより、更に別の制約を適用しても良い。更に別の制約は、最小範囲フレームと最大範囲フレームの双方を構成するための機会を提供する別の範囲フレームを含んでいても良く、或はその他の制約が適用されても良い。例えば、最大範囲フレームと、最大範囲フレームと関連していない別のコンテナとの間にストラットが適用されても良い。それにより、最大範囲フレームと関連するコンテナのサイズを制約し続ける一方で、別のコンテナの変化に従って範囲フレームを動的に移動させることが可能になるであろう。

【0168】

ステップ636の後、ステップ638の操作を経て、ユーザはレイアウトが完成する時点までレイアウトに、更に別のコンテナを追加しても良い。レイアウトが完成したならば、ステップ640はコンテンツのレコードをレイアウトに挿入して文書を生成する。ステップ642は、コンテンツの全てのレコードに関してこのプロセスを繰り返させ、それにより、1組の可変データ文書の作成を可能にする。全てのレコードが消費され、文書が生成されたならば、それらはステップ644で必要に応じてプルーフニングされ（proofed）（11.10-11.13章を参照）且つ／又は印刷される。この方法620はステップ646で終了する。

【0169】

7. イメージコンテナ

イメージコンテナは、写真、イラスト、ロゴ又は線図などの画像を含むことを目的とする特定の種類のコンテナである。

【0170】

イメージコンテナは、含まれるべき画像のスケールリングを制御するための挙動の選択肢のうちの1つを有することができる。それらの選択肢は次の通りである。

【0171】

・「全画像当てはめ」。これにより、画像全体がコンテナの中に嵌り込み、画像の縦横比が維持されるように、画像はスケールアップ又はスケールダウンされる。

【0172】

・「ボックスのサイズへの画像当てはめ」。これにより、コンテナ全体が画像により満たされ、画像の縦横比が維持され、画像の残りの部分は視界から切り取られるように、画像はスケールアップ又はスケールダウンされる。

【0173】

・「伸長当てはめ」。これにより、画像は、元来の画像の縦横比を無視して、横方向と縦方向の双方でコンテナのサイズに厳密に当てはまるようにスケールアップ又はスケールダウンされる。

【0174】

・「スケールリングせず」。これにより、画像にスケールリングは適用されず、コンテナの外側に出る画像の部分はいずれも視界から切り取られる。

【0175】

更に、コンテナの内側にあり、コンテナより小さい画像が横方向の左、中央又は右と、縦方向の上、中央又は下との何らかの組み合わせである特定の方向に整列されるように、イメージコンテナを設定することができる。

【0176】

8. テキストコンテナ

テキストコンテナは、様々なフォント及び段落文体によってフォーマットでき、

10

20

30

40

50

且つテキストコンテナの様々な辺に対して整列又は行端揃えできるようなテキストの範囲を含むことを目的とする特定の種類のコンテナである。

【0177】

テキストコンテナは静止テキスト、可変テキスト又はそれらの組み合わせのいずれかを含む。テキストコンテナに画像などの他のオブジェクトを挿入することも可能であり、それらのオブジェクトはテキストの流れと同じように流れる。

【0178】

静止テキストは、コンテナ内へ直接に文字を打ち込むことにより入力される。可変テキストは、11章で更に詳細に説明するように、ライブラリからコンテナへ可変データオブジェクトをドラッグすることにより追加される。1つのテキストコンテナに2つ以上の可変テキストオブジェクトを表示することができる。

10

【0179】

静止テキストのフォーマットは、個別の文字、語句又は段落全体に適用されることが可能であるが、フォーマットオプションは可変テキストアイテムのインスタンス全体にのみ適用可能である。

【0180】

8.1 テキストフォーマット

テキストフォーマットは次のようなオプションを含む。

【0181】

- ・フォント
- ・フォントサイズ
- ・太字
- ・イタリック体
- ・下線
- ・色
- ・行間隔
- ・強制大文字化
- ・自動ハイフネーション

20

8.2 整列

下記のテキスト整列オプションはいくつかの特定の実施の形態で許可される。

30

【0182】

- ・横方向の整列：左（デフォルト）、横方向の中心位置合わせ、右又は行端揃え
- ・縦方向の整列：上（デフォルト）、縦方向の中心位置合わせ、下

異なる実施の形態においては、垂直方向に行端揃えされたテキストなどの他のオプションが適切であろう。

【0183】

8.3 カラム（列）（Columns）

テキストコンテナの中のテキストは、1つのカラム、又は2つ以上の縦方向のカラムのいずれかに配列されれば良く、「ガター（gutter）」として知られるスペースが、各カラムの左右の辺を、それぞれ隣接するカラムから分離している。

40

【0184】

従来の技術においては、隣接するカラムの間に分割線を引くことによりカラム幅を調整可能にすることが一般的であるが、ガターは事前にプロパティシート又はダイアログボックスで値を変更することにより、或は類似の非直接的方法により設定されている。

【0185】

図7A及び図7Bは、3つのカラムを有するテキストコンテナ701と、マウス133及びポインタ313などの指示デバイスを使用する直接操作によって、カラム幅とガター幅の双方を再設定できる方法とを示す。

【0186】

図7Aを参照すると、テキストコンテナ701は、テキストの3つのカラム702に分

50

割されている。図示されるように、線 703 は隣接するカラムの間の中央分割線を表し、線 704 は隣接するカラム 702 の間のガター境界を表わしている。

【0187】

マウス 133 及びポインタ 313 によって中央分割線 703 を引くことにより、隣接するカラムの幅を調整できる。線引き機能の間に、カラム線引き機能が活動中であることを指示するために、デフォルトポインタの場所に特殊ポインタ 705 が表示される。

【0188】

図 7B に示すように、指示デバイスによってガター境界線 704 を引くことにより、ガターの幅を調整できる。線引き機能の間に、ガター線引き機能が活動中であることを指示するために、デフォルトポインタの場所に特殊ポインタ 706 が表示される。

10

【0189】

同じカラム間スペースの中にある 2 本のガター境界線は、各々のガター境界線と中央分割線との間で等しい間隔が常に維持されるように同時に調整される。

【0190】

更に図 7B に示すように、3 つ以上のカラムが存在している場合には、各々のガターを個別に操作することが可能である。

【0191】

8.4 可変フォントサイズ

様々なボリュームのテキストを収縮又は成長させることによりコンテナの内側にはめ込むため、及びテキスト全体をテキストコンテナの内側にはめ込んだ状態を保ちつつ、外部制約に従ってコンテナのサイズを変化させるために、テキストコンテナ内におけるテキストのフォントサイズを変化させても良い。最小フォントサイズ及び最大フォントサイズは、フォントサイズの可能範囲を制約して、テキストコンテナごとに設定されても良い。全てのテキストをコンテナの中にはめ込むために使用すべき最適フォントサイズを判定するために、2 進検索アルゴリズムを使用できるであろう。フォントサイズが最小サイズに達し、それでもテキストがコンテナの中に収まりきらない場合には、ユーザに対してエラーを示すことができる。

20

【0192】

8.5 コンテナ間におけるフォントサイズ同期

2 つ或はそれ以上のテキストコンテナの間で、あるテキストコンテンツをコンテナのうちのいずれかの中に適正にはめ込むためにそのコンテナのフォントサイズが変更されるか否かに関わらず、全てのコンテナにおいてフォントサイズが等しくなければならないことを指定する制約が作成されても良い。

30

【0193】

この同期のために選択されたテキストコンテナが、制約の追加以前に異なるフォントサイズを有している場合、最初にフォントサイズが平均され、選択された各々のコンテナに適用されるのが好ましい。別の実施の形態は、第 1 の選択されたコンテナのフォントサイズを使用し、そのフォントサイズを他の全てのコンテナに適用する。更に別の実施形態は各コンテナに外部フォントサイズ設定を適用する。

【0194】

図 8 は、文字サイズ同期の動作を示すための画面、ツール及びアイコン例を示す図である。

40

【0195】

図 8 を参照すると、アプリケーションウィンドウ 301 は先に説明したようなツールバーエリア 303 を有する。ツールバーエリア 303 は少なくともフォントセクタ 803 と、フォントサイズセクタ 804 と、フォントサイズ同期制約を追加するためのボタン 805 とを含む。フォントセクタ 803 とフォントサイズセクタ 804 は、テキストコンテナ内におけるテキストの特性を変更するために使用されれば良い。いくつかのテキストコンテナが選択されたときに、フォントサイズ同期ボタン 805 をクリックすると、選択されている全てのコンテナのフォントサイズが同期される。

50

【0196】

文書テンプレート309の中には、複数のテキストコンテナ806, 807が前もって配置されている。コンテナはマウス133を操作して、ポインタ13によってコンテナをクリックするか、又は選択矩形をコンテナの周囲へドラッグするか、或は別の方法により選択されれば良い。コンテナ上にコントロールポイントを示すなど、テキストコンテナが現在選択されていることを指示するために視覚的手段を使用しても良い。

【0197】

2つ或はそれ以上のテキストコンテナ806, 807を選択し、同期ボタン805を起動することにより、制約が追加される。他の実施の形態では、フォントサイズ同期制約は、プルダウンメニュー、キーボード指令又はその他の手段により適用される。

10

【0198】

オプションとして、制約はユーザインターフェースにおいてテキストコンテナ806, 807の上、又はその付近にあるアイコン808又は他の図形表現を介して指示されても良い。更に、オプションとして、どのテキストコンテナが互いに制約によりリンクされているかを指示するために、リンクされているテキストコンテナのそれぞれ対応するアイコンの間に接合線809が引かれても良い。

【0199】

他の実施の形態においては、選択されたオブジェクトの様々な特性を表示するために、ビデオディスプレイ画面144の別個のエリアが設けられても良く、このエリアは、選択されたテキストコンテナにフォントサイズ同期制約が適用されていることを指示し、且つ他のどのコンテナがその制約によりリンクされているかを識別するために使用されても良い。

20

【0200】

フォントサイズ制約によりリンクされているテキストコンテナのうちの1つ以上を選択し、当初、その制約を適用するために使用された指令又はシーケンスを繰り返すことにより、制約を削除しても良い。これは、同期ボタン805を起動するか、制約によりリンクされているテキストコンテナのうちの1つ以上を選択して、その目的のために特別に規定された追加指令又はシーケンスを適用するか、或は代表的アイコン808又は接合線809をクリックして、キーボード132の「削除(delete)」キーなどの指令又はシーケンス、又はディスプレイ144に示されるメニュー指令の中からの指令又はシーケンスを適用することにより実現されれば良い。

30

【0201】

異なるフォントサイズを有するテキストを既に含む2つ以上のテキストコンテナに対してユーザがフォントサイズ制約を適用した場合、最大フォントサイズ、最小フォントサイズ又は平均フォントサイズのいずれかが全てのテキストコンテナに適用される。或は、ユーザはポップアップダイアログ、又はそれに類似する手段によってフォントサイズを選択することをプロンプトされることが可能である。

【0202】

例えば、フォントサイズセクタ804から新たなフォントサイズを選択することにより、ユーザが既にフォントサイズ制約によりリンクされているコンテナのうちのいずれかに対して手動操作でフォントサイズを変更した場合、制約によりリンクされている全てのコンテナにその新たなフォントサイズが適用される。

40

【0203】

8.6 内側マージンの自動適用

テキストコンテナの内側マージンは、テキストコンテナが可視境界及び／又は可視背景を有しているか否かに基づいて自動的に設定される。

【0204】

図9は、内側マージンの自動適用の動作を示すための画面、ツール及びアイコン例を示す図である。

【0205】

50

図9を参照すると、アプリケーションウィンドウ301は、前述したようなツールバーエリア303を有する。このツールバーエリア303は、少なくとも境界セクタ903と、背景色セクタ904とを含む。

【0206】

テキストコンテナ905は可視境界又は背景色を持っておらず、コンテナの辺とコンテナの内側のテキストとの間にマージンも存在していない。

【0207】

テキストコンテナ906は可視境界を有し、コンテナの辺とコンテナの内側のテキストとの間に4つの辺全てでマージン907を有する。

【0208】

図9に示すようなシステムの状態に先立つ事象において、テキストコンテナ905に類似する状態にあったテキストコンテナ906は、可視境界又は背景も、内側マージンも持っておらず、マウス133によってコンテナ906をクリックするか、その周囲に選択矩形をドラッグするか、又は別の方法により、テキストコンテナ906が選択される。テキストコンテナ上にコントロールポイントを示すなど、テキストコンテナが現在選択されていることを指示するために視覚的手段が使用されても良い。

【0209】

次に、境界セクタ903を使用することにより可視境界が適用されるか、或は背景色セクタ904を使用して背景色が適用される。

【0210】

可視境界又は背景がテキストコンテナに適用される時点で、内側マージンが設定されていない場合、4つの辺の全てでテキストコンテナに所定の内側マージン907が自動的に適用される。内側マージンは、テキストコンテナの背景色とテンプレートの主要な周囲背景色との視覚的区別がつく場合には、必ず自動的に追加されても良い。

【0211】

コンテナに対して内側マージンが自動的に設定されるであろうと思われるもう1つの状況は、コンテナの背景色がページ色と異なる場合である。この場合、コンテンツをコンテナの辺まで流すためではなく、コンテンツとページ背景との視覚的分離を図るために、内側マージンが追加されるであろう。コンテナの背景色が完全に透明である場合には、定義上、そのコンテンツはページと同じ背景の上に描画され、内側マージンが追加される必要はない。

【0212】

8. 7 コンテナ間におけるテキストの流れ

テキストのボリュームが先行するコンテナのサイズを超えるような場合、テキストをそのコンテナから次のコンテナへ流すために、2つ或はそれ以上のテキストコンテナを順次リンクさせても良い。

【0213】

テキストがそのようにリンクされた全てのコンテナの組み合わせエリアに収まりきらない場合には、異なるコンテナのフォントサイズの間の関係を維持しようと試みつつ、テキストをはめ込み可能にするような、コンテナごとのフォントサイズを見出すことが必要である。

【0214】

1組のリンクされるコンテナに対してこれを実現する望ましい1つの方法は、テキストがちょうど収まりきるまで、又はリンクされた全てのコンテナがそれぞれ対応する最小フォントサイズより小さいフォントサイズに到達するまで（到達した時点で、操作が完了できないであろうということを示すエラー又は他の標識がユーザに示される）、コンテナごとにフォントサイズをスケーリングするために2進検索アルゴリズムを使用する。尚、一方のコンテナが他方のコンテナより先に最小フォントサイズに到達した場合、その時点で必ずしもアルゴリズムの終了にはならないことに注意する。そのような状況においては、アルゴリズムはコンテナ間のフォントサイズ関係を維持しようと試みるが、そうなること

10

20

30

40

50

を保証はしない。

【0215】

これを実現する別の方法は、先に説明した方法に類似した方法を使用するが、リンクされたコンテナの内のいずれかがその最小フォントサイズより小さいフォントサイズに到達した場合にエラー又は他の標識によって終了する。この代替方法はリンクされているコンテナの間の相対フォントサイズを維持するが、2進検索アルゴリズムの場合より多くのケースでエラーを生じ、解を生成できないと思われる。この代替方法は、リンクされている全てのコンテナの間のフォントサイズ同期も望まれている場合にテキストの流れを実現するために使用されるであろう。

【0216】

フォント収縮によってテキストの流れを実現する別の方法は、テキストがリンクされているコンテナの集合の中にはめ込まれるまで、又は集合の中の第1のコンテナがその最小フォントサイズに到達するまで、単純に第1のコンテナのフォントサイズを縮小してゆく。第1のコンテナがその最小フォントサイズに到達した場合、テキストが収まりきるか、又は全てのコンテナが最小フォントサイズになるまで、第2、第3のコンテナと順に収縮させることにより操作を続ける。それでもテキストが完全に収まらなければ、何らかの方法により、おそらくはエラー通知によってそれを指示する。この方法では、コンテナの順序付けが重要であり、それはテキストが流れる順序として定義される。従って、テキストがコンテナAで始まり、コンテナBへ続くならば、Aは第1のコンテナであり、Bは第2のコンテナである。

【0217】

9. ガイド

ガイドは、コンテナ及び他のガイドの位置決めを補助するためにページの幅又は高さいっぱいにならって示される縦線又は横方向の線である。固定及び非固定（浮動）の2種類のガイドがある。ガイドはストラット（struts: 後述する距離制約）を介して他のガイド又はコンテナに接続されても良い。ガイドは文書には現れず、デザインを補助する手段として文書テンプレート309にのみ現れる。ガイドは、ルーラエリアから作業エリアへ指示デバイスをドラッグすることにより作成されれば良い。

【0218】

9. 1 固定ガイド

固定ガイドはページに固定されるか、印刷可能エリアなどのページのいくつかの部分に固定されている。固定ガイドは、全ての文書について同じ位置を有するという意味で固定されている。そのようなガイドはページの特定の縁部に対して、又はページの印刷可能部分に対して、例えば、左端に対して固定されていても良い。この例では、ページがリサイズされる場合（例えば、異なるサイズの用紙に印刷される場合）、ガイドは左端から同じ距離に留まるが、他の端からは必ずしも同じ距離をとらないであろう。

【0219】

9. 2 非固定ガイド

非固定ガイド又は浮動ガイドは異なる文書において異なる位置を有し、文書内のデータが変化した場合、例えば、文書テンプレートがデータソースからのレコードとマージされる場合にも異なる位置を有する。この例では、データが文書テンプレートとマージされると、1つ以上のコンテナはデータソースからのデータを保持し、そのため、非固定ガイドの位置はストラットの作用によって判定されることが可能になるであろう。そのようなデータが文書テンプレートとマージされるまで、非固定ガイドの位置はわかっていないか、或は編集及びデザインを目的として初期（おそらくは任意の）位置が与えられるであろう。実施の形態によっては、浮動ガイドは、データソースからのデータが文書テンプレートとマージされるときに、ユーザにより編集されるか又はレイアウトエンジン105により変更されることが可能である位置を常に有する。

【0220】

9. 3 ガイドの実施の形態

特定の実施の形態においては、固定ガイドと非固定ガイドは、共に、作業スペース、ページ境界又はページ印刷可能エリアの縁部にあるルーラまで拡張できる実線により表現される。固定ガイドと非固定ガイドを識別するために、色が使用される。或は、固定ガイドは実線により表現され、非固定ガイドは破線又は点線により表現されても良い。

【0221】

図12を参照すると、非固定ガイド1204は、ルーラ308の水平方向セクションからマウス133によってドラッグすることにより、前もってGUI301に作成されている。第1のコンテナ1201は非固定の辺1202を有する。辺1202から非固定ガイド1204までストラット1203が作成されている。第2のコンテナ1205は、コンテナ1205の上辺をガイド1204に直接に整列させることにより、ガイド1204に直接に接続されている。第3のコンテナ1206はストラット1207によりガイド1204に結合されている。第4のコンテナ1208もストラット1209によりガイド1204に結合されている。

10

【0222】

コンテナ1201の高さの変化させる結果、非固定の辺1202が動くにつれて、ストラット制約1203を維持するために、ガイド1204はその縦方向の位置を変化させる。その結果、コンテナ1205、1206及び1208はガイド1204に対する制約を維持するために移動する。同様に、データを文書テンプレートとマージした結果、コンテナのサイズが決定又は変更されると、ストラットの作用によって非固定ガイドはそれらの制約の中のコンテンツに基づく位置をとるであろう。

20

【0223】

コンテナとガイドとの間にストラット制約を追加する方法は、本明細書の別の章で説明される、コンテナ間にストラットを追加する方法に類似している。

【0224】

例えば、図12のコンテナ1205のように、コンテナの辺がガイドに直接に付着されるような制約を追加するのに好ましい1つの方法は、コンテナ又はコンテナの辺をガイドから事前に定義された距離の中の、視覚的にガイドに「スナップ」するような位置までドラッグするか、或は同様にガイドをコンテナの辺から事前に定義された距離の中の、コンテナの辺にスナップするような位置までドラッグするという方法である。コンテナ、辺又はガイドがスナップ状態にある間にドラッグ操作から解放されると、制約が作成される。

30

【0225】

10. ストラット (Struts)

ストラットは、コンテナの特定の辺と、ガイド及びマージンなどの他のオブジェクトとの間の一定の距離を維持するために使用される種類の制約である。

【0226】

各ストラットは同じ文書テンプレートの中の2つの異なるコンテナに属する厳密に2つの平行な辺に適用されるか、1つのコンテナの1つの辺と平行なガイドに適用されるか、又は2つの平行なガイドの間に適用されるのが好ましい。他の実施の形態はストラットが結合できるものを制限しても良く、或は1つのコンテナの幅又は高さの制約を指定する方法として、ストラットにコンテナの辺同士を結合させるか、又はストラットにより、辺又はガイドを平行なページマージン又は辺と結合させても良い。

40

【0227】

ストラットは、ストラットにより付着される辺がいずれも、配置に関して他方の辺に対する優先順位を持たないという意味で対称である。

【0228】

ストラットの長さは視覚的に又は数値により表現されるか、或はその組み合わせにより表現されれば良い。実施の形態によっては、全てのストラットがその現在長さを視覚的に図形表現（ストラットアイコン）により表示するが、他の特性はダイアログボックス内の数又は視覚的設定として目に見える。

【0229】

50

図4を参照すると、第1のコンテナ407の辺410はストラット制約により第2のコンテナの辺411に結合されている。ストラット制約は、オプションとしてアプリケーションの様々なモード、例えば、プレビューモードにおいては隠蔽されても良いストラットアイコン(strut icon)412により指示されている。

【0230】

ストラットアイコンは、通常、正規の動作モードでは隠蔽されており、マウス133と関連するポインタ313がストラット、或はストラットに結合されたコンテナ又は辺又はガイドの付近にあるときに現れる。その付近の範囲には、ストラットに直接に結合されている、又は任意の数のストラットを介して間接的に結合されている他のストラット、コンテナ、辺又はガイドが含まれる。

10

【0231】

ストラットは1次元の長さを持つだけであるので、その次元のストラット位置をストラットが結合されている辺の現在位置から取り出すことができる。そこで、辺410と辺411との間にストラット412が描画される。縦方向のストラットの位置は様々な方法により計算できる。1つの方法は結合されている辺の中間点を平均し、その平均にストラットの中間点を配置する。従って、ストラット412の中間点は、辺410及び411の中間点の厳密に中間の位置にある。

【0232】

コンテナのコンテンツは文書間で異なる場合もあり、従って、文書が異なれば、コンテナの非固定の辺はページに対して様々な位置にあって良い。ストラット制約の作用は、辺又はガイドの間の距離を維持することであり、そのため、ストラットにより結合された辺又はガイドは文書の中で限られた可能位置範囲を有することになる。

20

図4を参照すると、例えば、ユーザがマウス133を使用してユーザインターフェースを介して1つの辺を操作した場合に、コンテナ407の第1の辺410が動けば、第1の辺410と、それに結合された第2の辺411との間のストラット制約(412)を維持するために、第2の辺411は「押し出される」か又は「引っ込められる」であろう。ユーザは結合されている辺又はガイドを押し出す又は引っ込めるなどのストラットの動作を認知するであろうが、ユーザインターフェースにおける実際の動作は文書を生成するときに使用される動作と同じである。

【0233】

別の実施の形態では、ストラットツールが起動されるが、ストラットを出現させるための、先に説明した方法は、ストラットツールが起動される度に全てのストラットが出現するように変形されても良い。或は、ストラットがポインタの上に直接重ねられたときにのみストラットは出現し、他のストラットはその時点では全く現れない。

30

【0234】

図4に示すように、ストラット412により結合された辺410及び411は互いに隣接し、対向しているが、辺が互いに隣接しているか、又は対向しているか、或はそのいずれでもないかに関わらず、別個のコンテナの任意の2つの平行な辺にストラットを適用することができる。

【0235】

図11は、図6A～図6Cを参照して先に説明したコンテナの例の第3の規則を使用して図示された、コンテナ間のストラットを示す。第1のコンテナ1101は最小範囲フレーム1103と、最大範囲フレーム1105と、コンテンツフレーム1104とを有する。ストラット1106は、コンテンツフレーム1104の対応する辺に結合されている。最小範囲と最大範囲が等しくなるような第2のコンテナ1102が示されている。従って、コンテナ1102はこの構成では固定されたサイズを有する。ストラット1106はコンテナ1102全体に作用し、その結果、コンテナ1102は水平方向に押し出されるか又は引っ込められる。

40

【0236】

従って、第1のコンテナのコンテンツフレームのいずれかの辺と、第2のコンテナの最

50

大範囲フレーム又はコンテンツフレームのうちのいずれかの平行な辺、又は平行なガイド、マージン又は他のオブジェクトとの間にストラットを作成できるであろう。

【0237】

10.2 コンテナ間にストラットを作成する

図10を参照すると、まずマウス133及びポインタ313を使用して、ツールバーエリア303のストラットツールボタン406をクリックすることによりストラットツールを起動することによって、新たなストラットを作成できる。次に、2つのコンテナをリンクするために2つの操作のうち的一方を実行できるであろう。第1に、第1のコンテナ1001の内側にある任意のポイント1003でマウス133を押して保持し、ポインタ313を第2のコンテナ1002の内側にある任意のポイントで終わるようにドラッグし、そのポイントでマウス133から指を放すことにより、経路1004をドラッグする。これは図10ではポインタ313の位置により示されている。或は、図10でポインタ313の位置により示されるように、まず、第1のコンテナ1001の内側にある任意のポイント1003でマウス133をクリックし（即ち、押して、放し）、次に第2のコンテナ1002の内側の任意のポイントで2度目のクリックを実行することにより、コンテナ1001及び1002の間にストラットを形成しても良い。

10

【0238】

本実施の形態においては、上述のストラット作成プロセス、並びに結合すべき2つのコンテナを選択することによって、経路1004の開始ポイント及び終了ポイントの、対応するコンテナの様々な辺に対する近接度に基づいてコンテナの結合される辺が選択される。例えば、開始ポイント1003は第1のコンテナ1001の左側の辺1005より右側の辺1006に近く、従って、右側の辺1006が結合される。同様に、第2のコンテナ1002の左側の辺1007が結合される。

20

【0239】

別の実施の形態では、選択される辺は2つのコンテナについて平行な隣接する辺の中で最も近接している1対の辺、例えば、図10に示すようなコンテナ1001及び1002の場合では、辺1006及び1007であっても良い。

【0240】

更に別の実施の形態においては、選択される辺は、ポインタ313が第1のコンテナの内側から外側へ移動するときポインタの経路が最初に交わる辺と、ポインタ313が第2のコンテナの内側から外側へ移動するときポインタの経路が最初に交わる辺であってもよい。

30

【0241】

ストラットの第1のポイント1003が規定された後、ユーザが適切な第2の辺を選択するのを助けるために、ポインタが第2の辺になりうる辺に接近するように動くにつれて、その辺の線の種類を変えるか、又は辺の付近にアイコンを表示することにより、第2の辺になりうる辺を図式的に示しても良い。第1のポイントにより規定されたコンテナ以外のコンテナの平行な辺を含めて、有効なストラットを作成しうる辺のみが指示されるようにしても良い。

【0242】

更に、ストラットが非対称である場合、例えば、ストラットが何らかの理由により結合される辺の各々を異なる方法で処理すべき場合などには、上述のストラット作成プロセスはそのことを図式的に指示することができる。例えば、開始ポイント1003は第1のコンテナ1001の内側にあり、終了ポイントは第2のコンテナ1002の内側にあるため、第1のコンテナの辺1006はストラットの操作により辺1007とは異なる方法で処理されることになり、ストラットはそのことを図式的に指示できるであろう。

40

【0243】

10.3 ガイド間にストラットを作成する

ガイド間へのストラットの作成は、先に説明したコンテナの辺の間へのストラットの作成に類似している。ガイドはコンテナのような幅や高さを持たないため、可能なユーザイ

50

ンターフェースの数は少なくなる。別の方法によれば、ストラットツール 406 を選択し、次に、第 1 のガイドを選択し、指示デバイス（マウス）133 によりドラッグして、そのようにして作成されたストラットを第 2 のガイドに付着させる。

【0244】

別の実施の形態では、2 つのガイドを選択し、次に、それらのガイドの間に方向性を持たないストラットを作成するストラット作成ボタン（アイコン）を選択する。方向性を有するガイドは、別の実施の形態により、どちらが駆動する辺で、どちらが駆動される辺であるかを識別する方法としてガイドが選択された順序を考慮することによって、作成されても良い。

【0245】

10

10.4 ガイドとコンテナとの間にストラットを作成する

1 つの実施の形態では、コンテナの 1 つの辺の上へガイドをドラッグすることにより、通常はそのガイドと辺との間に「0」の距離を指定する距離制約が作成される。この距離制約はストラットにより表現されても良く、従って、（ガイド又はコンテナを構成している間に、或は編集中に）ガイドをドラッグするそのような方法で実際にストラットが作成されるであろう。

【0246】

コンテナの辺の間にストラットが作成されるときの方法に類似する方法（ツールを選択し、辺又はガイドをクリックし、ドラッグし、放す）により、ガイドとコンテナの辺との間にストラットを作成できることが望ましい。

20

【0247】

10.5 マージンに至るストラットの作成

ページマージンとガイド又はコンテナの辺との間にも距離制約が存在することが望ましいであろう。この目的のためにストラットを使用することができる。実施の形態によっては、ガイドとコンテナの辺がページの縁部に対して位置を固定されており、それがストラットと同じ効果を有するために、ストラットが不必要である場合もある。所望のページマージンに対して固定されたガイドが作成された場合、他のガイド及びコンテナの辺をストラットを介してそれにリンクできるので、その場合にもページマージンとガイド又はコンテナの辺との間のストラットは不要である。従って、ストラットは、ページマージンをガイド又はコンテナの辺にリンク可能にする必要はない。

30

【0248】

別の実施の形態においては、ガイド及びコンテナの辺はページマージンに対して固定される能力を持たないであろう。この状況では、ページマージンに対する距離制約を表現するためにストラットを使用することが望ましいであろう。

【0249】

10.6 ストラットとコンテナとのインタラクション（Interactions）

ストラットは、コンテナと数多くの方法でインタラクトする。ストラットの最も単純な動作は、文書の中でコンテンツがどこに配置されるかを判定する方法として、コンテナの辺及びガイドが位置決めされうる場所を制限することである。しかし、ストラットをコンテナに結合すると、そのコンテナを許可されている状態の集合の中に保持したままにするように、コンテナの状態、特にその辺状態も変更されるであろう。

40

【0250】

特定の実施の形態において、ストラットをコンテナへ取り付けるのを統制する規則は次の通りである。

【0251】

1. 2 つの固定辺の間にストラットが結合される場合、選択された辺同士は非固定になる。

【0252】

2. 2 つの固定辺の間にストラットが結合される場合、選択された第 2 の辺は非固定になり、ストラットの向きに対応する次元におけるコンテナの長さが固定されている場合、

50

ストラットに対して垂直な2つの辺は非固定であり、その次元におけるコンテナの長さは固定されたままである。例えば、選択された第2の辺が、幅が固定のコンテナの縦方向の辺であれば、そのコンテナの2つの縦方向の辺は非固定であるので、そのコンテナは横方向には移動自在であるが、その幅は固定されたままである。

【0253】

3. ストラットに付着された2つのコンテナ辺及び／又はガイドが固定になると、そのストラットは除去される。

【0254】

4. 固定されたコンテナ辺又はガイドがユーザにより移動されると、それに付随するストラットに結合された他方の辺又はガイドが不変のままとなるように、ストラットの長さ設定が変更される。

10

【0255】

11. 可変データ

ユーザインターフェース103により、ユーザはデータソースを可変文書テンプレートと関連付け、データソースをテンプレートとマージするときに有効文書を生成するためにテンプレートをデザインすることができる。マージングは、バックグラウンドにおいて対話形式で実行されるか、又はソフトウェアアプリケーション121のレイアウトエンジン105コンポーネントによりオンデマンドで実行される。

【0256】

データソースは可変文書テンプレートと対話形式で関連付けられ、レイアウトエンジン105は文書をマージし、ユーザがマージされた文書を対話形式でナビゲートする間にオンデマンドで文書のページをレイアウトするのが好ましい。別の実施の形態は、マージングプロセスの間にインタラクティブユーザインターフェースを必要とせずにデータをテンプレートとマージし、そのマージされた文書を印刷しても良い。別の実施の形態は、バックグラウンドにおいてマージされた文書を生成しつつデータをテンプレートとマージし、マージされた文書のユーザによるナビゲーションに関して非同期的に現れるメッセージにより、マージングプロセス中の問題をユーザに警告しても良い。

20

【0257】

11.1 データソースの選択

可変文書テンプレートをデータソースとマージするためには、適切なデータソースをテンプレートと関連付けることが必要である。そのような関連付けを実現するための1つの方法は、図13に示すような、ユーザインターフェース103の一部を形成するデータソース選択ウィンドウを使用する方法である。

30

【0258】

図13は、UIアプリケーション103によりディスプレイ144に提示されるGUIウィンドウ例を示す図である。これによりユーザは、データベース、ファイル、複数のデータベースの結合体、又は他の何らかの情報源であると考えられるデータソースを対話形式で選択できる。

【0259】

図14は、図13の画面でXMLファイルが選択された例を示す図である。こうしてソースが選択されると、そのソースは、明示されて又は暗黙のうちに関連付けが解除されるまでテンプレートと関連付けられる。

40

【0260】

11.2 データフィルタリング

特定のマージング動作に対して、データソースの中のデータの全てが有効であるとは限らないということも起こりうる。そのような事態が起こった場合、ユーザが使用されるべきデータの範囲を限定するために実行可能な方法は数多くある。1つの実施の形態では、アルゴリズムック(algorithmic)データフィルタリングにより、使用されるべきデータの種類と量を制限する。例えば、所望の地理的領域と関連するいくつかのレコードのみが文書の生成に使用されるように、ある範囲の郵便番号が選択されても良いであろう。その

50

ようなアルゴリズムックフィルタリングは当該技術においては良く知られている。別の実施の形態では、データソースがテンプレートと関連付けられた後、ユーザインターフェースを介して、ユーザは使用されるべきデータを選択し、マーキングする。

【0261】

図17は、メニューアイテムの選択を介して、データソースが可変文書テンプレートと関連付けられた後、ユーザインターフェース103のGUIを介してユーザがデータフィルタリング動作にアクセスできる好ましい方法を示す。

【0262】

11.3 データソーティング

フィルタリングに関連して、関連データを見出すためにデータ集積を1つのシーケンスに分類する能力がある。1つの実施の形態では、ユーザはデータソースのユーザが選択する「キー」変数に基づいて、ソースのレコードのソーティングを選択する。全てのレコードは、選択されたキー変数で、アルファベットの昇順に並べ替えられる。例えば、キーが変数「Name」であった場合、ソート動作が実行された後、名前が文字Aで始まるレコードは名前が文字Bで始まるレコードより前に現れるであろう（英文テキストの場合）。適切なユーザインターフェースの一部が図15に示されており、これはデータソースを選択するプロセスの一部としてソーティングが実行できることを示している。このソーティングは、データソースが関連付けられた後でも、図17に示すようなメニュー選択により指定できる。

【0263】

別の実施の形態では、様々に異なる種類のデータに対して、数値又は言語特定方法などのユーザ選択可能なソーティング方法を実行できる。別の実施の形態は、選択されたキー変数に格納されている情報の種類に基づいてソーティング方法を自動的に選択するが、自動的に選択された方法の選択肢が不十分である場合、これはおそらくオーバーライドメカニズムを伴うであろう。

【0264】

11.4 レコードナビゲーション (Record Navigation)

データソースは、各々が関連情報の集積である複数のレコードを含む。例えば、データベースは顧客を記述したいくつかのレコードを含むか、又は製品のレコードを含むと考えられるであろう。データベース内のレコードを1つのシーケンスとして順序付け、ユーザによりそれをナビゲートできれば有用である。

【0265】

図18は、データソースの変数及びデータソースをナビゲートするための方法を示すユーザインターフェースの一例を示す図である。好ましいナビゲーション方法は、図18に符号1805で示すように、ポインタ313及びマウス133により起動できる1組のGUIボタンを表示する。各々のボタンはレコードのシーケンスに沿って1つずつ前後に動くか、シーケンスの始まり又は終わりまで進むか、或は何らかのレコード数だけ進む。1つの方法は、ポインタ313／マウス133又はキーボード132により起動される、それらのオプション（図17に示す）を示すメニューを更に含み、加えて、ユーザは特定の番号のレコードへナビゲートすることができる。

【0266】

特定のレコードへナビゲートしたならば、ソフトウェアアプリケーション121は、その現レコードを何らかの方法によりディスプレイ画面144に表示する。これは、そのレコードを可変文書テンプレートとマージし、シーケンス中の現レコードの索引を表示しても良いであろう。

【0267】

11.5 変数

各データソースは、データソース内のレコードに適用される複数の名前付き変数から構成できる。例えば、顧客データソースは、顧客名及び顧客住所に関する変数を有しているであろう。レコードごとに値が異なるため、それらは変数であるが、変数の名前はあらゆる

るレコードに対して同じであっても良い。

【0268】

ユーザがデータソース内のレコードを検査し、可変文書テンプレートと関連付けられるべき変数を選択することができるように変数を表示する方法を有するのにも有用である。

【0269】

変数を表示する好ましい方法は、図18に示すように、自由浮動ウィンドウの中に名前及び現在値により縦に各変数を列挙する方法である。現在値は、現レコードを検査し、そのレコードに関するデータソース中の各変数値を求めることにより決定される。現レコードは、レコードナビゲーションを使用して先に概要を述べたように選択されても良い。尚、全ての値又は変数名が表示される必要はないことに注意されたい。ウィンドウの中に嵌め込むには情報が多すぎる場合に、ユーザが各々の変数又は値を見られるようにするためのナビゲーションメカニズムがあっても良い。別の実施の形態は、別のウィンドウの一部である（言い換えれば、サブウィンドウ）類似のディスプレイを使用しても良い。文書テンプレートと関連するデータソースが存在しなければ、図16に示すように、ウィンドウは変数なしになるであろう。

【0270】

変数は数種類のデータを含んでいても良い。1つの変数はテキスト値を格納しているか、又は画像を格納している。以下の説明中、それらをそれぞれテキスト変数及びイメージ変数と呼ぶ。

【0271】

データソースの変数は1種類の値を有し、従って、1つの変数はテキスト又は画像のいずれかを表示し、双方を表示することはないであろう。別の実施の形態は、データソースの変数にレコードごとに異なる種類のデータを格納させる。例えば、1つのレコードでは、製品記述変数はテキスト記述を含み、同じデータソースの別のレコードでは、同じ変数がイメージ値を有することが起こりうるであろう。

【0272】

11.6 テキスト変数 (Text Variables)

テキスト変数は、データソースの各レコードの中でテキスト値を含む。アプリケーション121は、図18のラベル1801により示すように、データソースの変数のリストの中の変数の名前の次に識別ラベルを配置することにより、変数がテキスト変数であることをユーザに報知する。別の実施の形態は、テキスト1802により示すように、変数の型を報知する方法として、変数の名前の付近に変数の値を表示しても良いであろうが、そのような方法単独では、状況によってイメージデータがテキストデータと同じように見えてしまうことも起こりうるため、あいまいさがないとは保証されない。アプリケーション121は、この技法を変数の型を識別する第1の方法としてではなく、単に変数のデータを示すための補助手段として使用する。

【0273】

11.7 イメージ変数 (Image Variables)

イメージ変数は、データソースの各レコードの中でイメージ値を含む。値は圧縮フォーマット又は伸張フォーマットのイメージデータであっても良く、或はファイル名又はイメージ資源の場所を規定するための他のアクセス方法であっても良い。アプリケーション121は、図18のラベル1803で示すように、データソースの変数のリストの中の変数の名前の次に識別ラベルを配置することにより、変数がイメージ変数であることをユーザに報知する。別の方法は、テキスト1804で示すように、変数の名前の付近に変数の値を表示するが、（例えば）画像ファイル名がテキストデータと混同されることもありうるため、この方法が正確であるとは保証されない。

【0274】

11.8 コンテナへのドラッグ・アンド・ドロップ

どのような変数が存在するかがユーザにわかるように、データソースの変数が見えている間にユーザに可変文書テンプレートを構成させるための方法を有するのは有用である。

また、所定の変数から適切な型のコンテナを作成できると有用である。

【0275】

アプリケーション121は、ユーザインターフェース103を介して、ユーザにマウス133及びポインタ313を使用して変数リスト(図19に示す)の変数を直接に操作させることにより、適切な型のコンテナを作成する。ユーザは1つの変数(例えば、1902と記されている矩形エリアのうちの1つ)を選択し、それをマウス133を使用してデザイン面までドラッグし、選択肢1906を解放する。これは、適切な型のコンテナがその変数によって作成され、初期設定されるべきであることを報知する手段として使用される。例えば、このようにしてデザイン面へドラッグされたテキスト変数(ラベル1901により示される)は、現レコードにおけるその変数の値(テキスト1903によっても示される)を表示するテキストコンテナを生成する。イメージ変数1904がデザイン面へドラッグされた場合には、ファイル名1905及び/又はユーザにより採用される設定に応じてスケールリング、クロッピング及び整列が行われた画像自体を表示するイメージコンテナが表示される。

10

【0276】

そのようにして生成されるテキストコンテナの初期状態は、ユーザが構成できるが、別の方法は、ユーザが選択肢を解放した場所をヒントとして使用して、コンテナをページに固着する。

【0277】

その場所はデザインウィンドウの外側又はページの外側(無効である場合)など、無効である場合もあると考えられるため、その場所をコンテナの厳密な場所として使用できず、その場所はヒントにはなり得るであろう。コンテナの中心がその場所に置かれても良く、又はコンテナの角の1つが場所に近接して位置決めされるか、或は完全にページの中に入るなどの他の制約に違反することなく、コンテナはその場所にできる限り近接して配置されても良く、又はコンテナは単純にデフォルト位置又は無作為の位置に配置され、場所が無視されても良い。マウスをドラッグした結果として新たに作成されたオブジェクトを位置決めする方法には、数多くの可能な実施の形態が存在する。

20

【0278】

更に、アプリケーション121は、そのようにしてデザイン面へドラッグされた変数に関して現レコードのコンテンツの全てを十分に格納できるサイズになるように、コンテナを作成する。従って、長文のテキストは大きなテキストコンテナを生成する。別の方法はドラッグされた変数のデータに対してメジアンサイズを事前に計算し、そのメジアンサイズを表示するのに十分なサイズのコンテナを生成しても良いであろう。他の方法はこの方法の変形を利用して、最大のレコードのデータ、又は最小のデータ、或は他の何らかのアルゴリズムを格納するのに十分なサイズのコンテナを作成する。

30

【0279】

別の方法によれば、変数をデザインスペースへドラッグし、次に、その変数の値を保持するコンテナについて矩形を描画する。更に別の方法は変数を選択し、次に、初めにドラッグ操作を実行することなく、そのコンテナに対して矩形を描画する。別の代替方法では、変数表現の上でマウスを2度クリックし、デザインの中で無作為に又はアルゴリズムに基づいて選択された場所に対応するコンテナを出現させる。

40

【0280】

別の実施の形態は、データソースがデザインと関連付けられたならば、直ちにデータソース中の全ての変数をデザインの中でコンテナとして出現させ、ユーザは、この後、それらのコンテナを必要に応じて位置決めするか又は削除する。このようにして作成されるコンテナの初期位置とサイズは、データソース中のデータの順序付けと、各変数におけるコンテンツの相対的サイズによって決まるであろう。そのような実施の形態においては、データソースを既存のデザインと関連付けても、特定の変数を表現するコンテナが既に存在する場合には、その変数に対するコンテナは作成されないであろう。この方法は同じ変数を表す複数のコンテナの作成を容易にしないため好ましくない。

50

【0281】

特定の実施の形態は、データソース変数と既に作成されていたコンテナとの連携をも支援する。これによりユーザは、まずデータソースを関連付ける必要なくデザインを作成することができる。そのような実施の形態は、融通性に富む作業の流れを支援するために、データソースの関連付け、コンテナの作成、及び変数とコンテナとの連携を更に実行しても良い。

【0282】

そのような方法により、既存のコンテナのコンテンツをマウス133とポインタ313の組み合わせを使用して実現されるドラッグ・アンド・ドロップメカニズムによって修正することもできる。その場合、コンテナへ変数をドラッグすることで、そのコンテナが表示する内容が再度割り当てられるか、又はコンテナが保持する内容が追加されることもある。例えば、静止テキスト「Dear」を含むコンテナが構成されており、そこで、「Last Name」と名づけられた変数がコンテナ内へドラッグされると、この結果、動作モード及びコンテナに対する設定に応じて、それらのデータの全てを表示するコンテナが得られるか、又は最後の変数の値のみを表示するコンテナが得られることになる。

10

【0283】

11.9 複数の文書の表示及びナビゲーション

文書は、文書テンプレートをデータソースからのデータとマージしたことによる生成物であるので、1回のマージ操作の結果として多数の文書が生成されても良い。そのような文書をナビゲートし、表示できる方法は数多くある。

20

【0284】

1つの実施の形態は文書を表示し、ナビゲートするいくつかの方法を含み、データソースの各レコードがオンデマンドで文書テンプレートとマージされて、所定のマージされた文書に関して文書テンプレートの制約と、データの双方のビューを生成する「ライブブルーフィング (live proofing)」(以下に説明する)に依存している。そのような実施の形態の場合、ユーザ設定及び／又は生成される文書の性質に応じて、いくつかの可能なユーザインターフェースが存在する。それらのユーザインターフェースの概要は次の通りである。

【0285】

・各文書は、ワード処理文書又はスプレッドシートと同様に、縦方向及び／又は横方向に編成された1方向に流れる文書として表示され、スクローリングメカニズム、指示デバイス及び／又はキーボードによりナビゲートされる。図18のボタン1805のように、データソースのレコードに沿ってナビゲートするために別個のメカニズムが使用されても良い。

30

【0286】

・各レコードはマージされると、多数のページを有する文書を生成できる。それらのページは縦方向及び／又は横方向に編成され、表示されれば良い。データソースのレコードに沿ってナビゲートするために、別個のメカニズムが使用されることが可能である。

【0287】

・文書テンプレートが各レコードとマージされたときに、ごくわずかのページしか生成しない場合、いくつかの文書が縦方向及び／又は横方向に表示されても良い。例えば、各文書が1ページ分の長さしかない場合には、全ての文書を1つの線形シーケンスとして配列し、ウィンドウイングシステムを使用してそれらを表示し、文書をナビゲートするためにスクローリングメカニズムを採用することが可能である。これは1つの文書内と、データソースのレコード間の双方で、ナビゲーションを空間ナビゲーションにマッピングする。

40

【0288】

・各文書が1つ或はそれ以上のページを有する場合、それらは横方向に編成され、その後、後続する各文書を先のレコードを表す文書の下方にレイアウトすることにより、データソースのレコードを網羅するために縦方向を使用し、それにより、1つの次元を使用し

50

て文書のスペースをナビゲートしても良い。横方向と縦方向の役割を入れ替えても、同様の効果が得られるであろう。

【0289】

上記の各々において、レコード間及びページ及び／又は文書の間をナビゲートするための、ボタン、メニューなどの他の方法も存在するであろう。

【0290】

別の実施の形態においては、文書テンプレートをマージされた文書とは別個に表示することができる。マージされた文書は「プリントプレビュー」でのみ生成され、そこから編集することは不可能であり、単純に1つのレコードをテンプレートとマージした結果を表示している。更に別の方法では、文書テンプレートと、非対話型プリントプレビューと、文書テンプレートの編集及びマージされた文書のプレビューの双方が可能であるライブプルーフリング (live proofing) のプレビューとを見ることができる。これらの作業モードの他の組み合わせも可能である。

10

【0291】

11. 10 プルーフリング (Proofing)

可変データが使用される画面上ジョブ及び印刷ジョブにおいては、マージされている全範囲にわたるレコードを表すレコード、特に、可変データレコードの集合からの実際の極端なケースを効率良く見つけ出すことにより、著しい生産性の向上を得ることができる。こうしてユーザは、マージされた最終的な文書がどのように見えるかを理解できる。

【0292】

20

可変画像の様々な異なるサイズ及び可変テキストの様々な異なる長さにより、マージされた文書がどのように影響されるかを見るために、複数のレコードをプレビューし、プルーフ印刷することは有用である。印刷すべき最も有用なサンプルレコードは、

- ・レコードの大半がどのように印刷されるか；及び
- ・表示すべきコンテンツがほとんどないか又は大量にあるレコード

を含む。

【0293】

この原理のより複雑な拡張形態は、

- ・テキストが余りにも小さく縮小されたか又はその最小サイズに到達したとき；
- ・何らかの可変データテキストが表示されていないとき；
- ・画像が余りにも大きくスケールアップされた時（印刷品質が不十分になる）；及び
- ・画像の縦横比が余りにも大きくひずんだとき

30

を含む。

【0294】

可変データを取り扱う場合、文書のレコード又はバージョンの数は非常に多くなり、自動化プルーフリングメカニズムがなければ、ユーザは、マージされた文書の全ての見かけが受け入れられるものであることを検査するために、各レコードを手動操作で検討しなければならないであろう。例えば、データベースのレコードの内のただ1つが好ましくない見かけをもたらすコンテンツを含むということは起こりうる事態であり、そのレコードを発見することは困難である。これは、レコードの数が非常に多い可変データジョブの場合に特に当てはまる。

40

【0295】

可変データ印刷ジョブをプレビューし、プルーフ印刷する効率の良い方法が開示される。これはどの文書の最終的な見かけも受け入れられるものであり、正しく印刷されることを保証する。画面上にプレビューされ且つ／又はプルーフ印刷されるために、最終的に印刷される他の全てのレコードを代表する限られた数のレコードが選択される。それらのレコードは、各レコードのマージ後の見かけを解析し、所定の1組の規則を使用して、文書の見かけに最も重大な影響を及ぼすレコードを判定することにより選択される。

【0296】

選択的プレビュー及びプルーフリングは、極端なケースを含めた出力結果のプロファイ

50

ルを提供する。これによりユーザは、ジョブ全体を印刷したときに驚くことはない。最短の表示サイズ及び最長の表示サイズを有するレコードである極端なケースを示すことは有益であり、メジアン表示サイズ、最も極端なケース及びメジアンケースと最も極端なケースとの中間にあるいくつかのレコードを含む「代表的な」ケースの短いリストを含むように、経済的な方法を一般化することができる。

【0297】

11. 11 ライブプルーフイング (Live Proofing)

ライブプルーフイングは、可変文書テンプレートとマージされた可変データを対話形式及びオンデマンドで表示するプロセスである。

【0298】

図20は、可変文書テンプレートとマージされたデータソースコンテンツのライブプルーフイングのためのユーザインターフェースを示す図で、アイテム2010、2012、2014、2016、2018、2020を有するデータソースのレコードがテンプレートとマージされており、印刷した場合にそのレコードのどの部分を表示するためのバー(2002)を含むグラフィカルユーザインターフェース2000を示す。可変文書テンプレート内で定義されている制約を見えるようにするために余分の境界2004及び色2006が追加されているが、それらのアーティファクトは印刷されることを意図されていない。

【0299】

ライブプルーフイングでは、テンプレートのデザインにおける誤りがインタラクティブナビゲーションの場合より早く発見されると思われるため、ライブプルーフイングは別個のマージングプロセスより有利である。

【0300】

アプリケーション121の特定の実施の形態は、ライブプルーフイングを使用している。別の実施の形態は、オプションとしてライブプルーフイングを使用しても良く、或はライブプルーフイングを全く使用しなくても良い。

【0301】

11. 12 選択的プルーフイング (Selective Proofing)

選択的プルーフイングは、平均文書を典型的に表しているか、又は少なくとも平均文書に類似している可変データ印刷アプリケーション121の選択されたレコードのみを見る方法である。

【0302】

ユーザが選択的プルーフイングの間に、評価すべき多数の種類のアイテムを指定できると有用である。次に示す尺度は、個々のコンテナ、コンテナをリンクするストラット、ページ、データソースのレコード又はデータソースの変数に適用できるであろう。以下の選択的プルーフイングの説明の中では、これらのものをまとめて「デザインアイテム (design items)」と呼ぶ。「マージサイズ」は、データソースからのレコードが可変文書テンプレートとマージされた後のデザインアイテムのサイズ尺度を表す。様々なサイズ尺度については後に説明する。

【0303】

選択的プレビュー又は選択的プルーフイングが起動されると、ユーザによる対話形式でデザインアイテムが選択されるか、又は優先設定によりデザインアイテムが選択される。いずれのレコードも可変文書テンプレートと別個にマージされる。レコードごとに、選択されたデザインアイテムのマージサイズが記録(格納)される。それらのマージサイズを比較して、選択されたデザインアイテムのマージサイズが、

- ・最小であるようなレコード；
- ・集合のメジアン、平均又は他の何らかの中央尺度であるようなレコード；及び
- ・最大であるようなレコード

が見出される。

【0304】

最も関連性のあるレコードを発見するために、アプリケーション 121 は単に文字の数ではなく、各レコードの実際の表示サイズを検査する。例えば、「w」は「i」より多くのスペースをとり、コンテナのサイズ及び文書の全体的な見かけに、より大きな影響を及ぼすであろう。

【0305】

それらのレコードが識別されたならば、それらは画面上でのプレビュー及び／又は印刷のためにユーザに提示される。

【0306】

可変データ印刷ジョブにおける全てのレコードが正しく表示されるという信頼性を更に高めるために、ユーザはマージされた文書からの追加のレコードを更にプレビューするか又はプルーフ印刷することもできる。選択されたデザインアイテムに関して最も異なるマージサイズを有するレコードが、最初に又は最も目立ってユーザに対し提示されるようにデザインアイテムを1つのシーケンスとして順序付けるために、集合の中央からの偏移を使用するのが好ましい。

【0307】

11.12.1 最小コンテンツサイズ及び最大コンテンツサイズ

デザインアイテムのサイズの有用な尺度は、その最小又は最大の幅、高さ又はコンテンツ面積を考慮することである。個別のコンテナの場合、それらの尺度を計算するのは簡単である。コンテナをリンクするストラットの場合には、幅と高さの双方ではなく、いずれか一方が重要であり、従って、面積は重要な尺度ではない。個別のページの場合、コンテナの幅の和、又はコンテナの高さの和、又はコンテナの面積の和を使用しても良い。レコードの場合には、そのレコードを表示するために必要とされる全てのページにおけるそれらの尺度の和を使用しても良い。変数の場合、その変数の値を全体として、又は一部で表示する全てのコンテナを使用しても良い。それらの尺度を選択し、合計するための他の方法も使用できる。

【0308】

11.12.2 典型的サイズ (Typical Size)

コンテンツのサイズのもう1つの有用な尺度は、所定のデザインアイテムに対して典型的サイズを考慮することである。典型的サイズは、平均幅又は平均高さ、或は平均面積、又は面積の和の平均、又は幅及び高さの二乗の和の平均、或は他の可能性であれば良い。別の尺度は上記の各々について平均ではなく、メジアンを使用する。他の統計的方式を使用しても差し支えない。

【0309】

特定のデザインアイテムの典型的インスタンスを位置付けるために、これらの尺度のいずれか又は全てを使用しても良い。例えば、平均面積に最も近い面積を有する特定のコンテナに対して、レコードの場所が位置付けられても良い。

【0310】

11.12.3 ホワイトスペースサイズ (White-space Size)

ホワイトスペースサイズは、デザインアイテムがどれほど典型的にマージ文書の集合の中に入っているかを表すもう1つの有用な尺度である。ホワイトスペースは多くの方法により定義できる。例えば、コンテナが最小サイズを有し、コンテンツがそのサイズを満たさないような場合に、ホワイトスペースは、コンテナの制約により規定されるようなコンテナの面積と、特定のレコードとマージされたときのそのコンテナの中のコンテンツの面積との面積差として定義されても良い。或は、ホワイトスペースは、幅又は高さなどの一次元のみにおけるコンテナのサイズとそのコンテナのコンテンツのサイズとの差として定義されても良い。

【0311】

ホワイトスペースを最大にすることは、コンテンツサイズを最小にすることに類似しているが、それと同じではなく、逆に、ホワイトスペースを最小にすることは、コンテンツサイズを最大にすることに類似しているが、それと同じではない。最小寸法などのコンテ

ナ制約が計算の上である役割を果たすために、それらは同じではないのである。例えば、最大のコンテンツを有するレコードを発見することは、最小のホワイトスペース面積を有するレコードを発見することと必ずしも同じではない。

【0312】

11. 12. 4 最も異なるサイズ (Most Different Size)

可変データ印刷において特に関心を持たれるのは、異常な文書の場所を規定することである。このために、異常度の尺度が要求される。そのような尺度は多くの方法により定義できるであろう。そのような尺度の一例は、あるデザインアイテムに関してメジアンサイズからの最大偏差を計算することである。サイズは幅、高さ、面積、幅と高さの和、幅と高さの二乗の和などの何らかの尺度に関して定義される。このメジアンサイズは、データソースのレコードごとに問題となる各デザインアイテムを検査し、そのアイテムのサイズを求め、そのようにして判定された全てのサイズのメジアンを求めることにより計算できる。次に、各デザインアイテムのサイズとメジアンサイズとの最大の差を計算し、それにより、最大偏差を有するデザインアイテムを求めることができる。

10

【0313】

図21A及び図21Bは、各文書ごとのコンテナの幅及び高さを使用して、平均文書及び最も異なる文書を計算する方法2100を説明するフローチャートである。この方法2100は、アプリケーション121のサブモジュールとして実行されても良い。

【0314】

この処理はステップ2101で始まる。まず2102で、第1のレコードから始めて文書中の各コンテナの位置及び寸法を計算し、ステップ2103、2104及び2105により形成されるプロセスループを使用しながら、データソースの全てのレコードを通して繰り返される。ステップ2106では、全ての文書について幅と高さを加算し、次に、それを文書の数で除算することにより、文書テンプレートにおける各コンテナの平均幅及び平均高さを計算する。こうして文書テンプレートのコンテナごとの幅と高さの平均値が求められるとステップ2107で、第1のレコードから始めて、ステップ2108、2113及び2116により形成されるループを経て全てのレコードを通して繰り返される。ステップ2108、2111及び2115により規定される別のループにより定義されて、文書中のコンテナごとに、ステップ2109でコンテナの幅と平均幅との差を計算し、ステップ2110でコンテナの高さと平均高さとの差を計算する。ステップ2110はそれらの値の各々を二乗し、それらを合計して、文書に対応する数値を出す。結果を二乗することは、値が平均値より離れていればいるほど、それらの値がより大きく強調されるということを意味する。

20

30

【0315】

上記の数値が計算されたならば、所定のコンテナ又は所定の文書に関して、そのコンテナが平均サイズにどれほど近いかを知ることが可能になる。コンテナが最低の値を有していれば、そのコンテナは平均に最も近い。逆に、コンテナが最高の値を有していれば、そのコンテナは平均から最も離れている。

【0316】

ユーザが最も平均的な文書又は最も平均から外れた文書を知りたい場合には、各文書についてコンテナごとの値がステップ2112で合計される。このようにして形成されたりストはステップ2114で分類され、そこで方法2100は終了する。

40

【0317】

最低の値を有する文書は、そのコンテナの各々を平均した文書に最も近い。最高の値を有する文書は、平均した文書に最も似ていない。そこでユーザの要求に応じて、それらの文書のうち幾つの文書が、5つの最悪文書、又は10の最悪文書であるかのように、ユーザに示すことができるであろう。

【0318】

他の方法では、コンテナの幅及び高さとは異なる特性に対して上記の計算を実行する。有用であると思われる他の特性には、コンテンツのサイズ、フォントサイズ（フォントが

50

拡張、収縮するコンテナの場合)、文書のホワイトスペースの面積、ストラットの長さ又はコンテナ辺の位置等がある。

【0319】

11.13 プレフライトチェック (Pre-Flight Check)

先に説明した種類のサイズ尺度は、典型的なデザインアイテム、非典型的デザインアイテム又は問題を含むデザインアイテムを発見するために使用されても良い。「プレフライトチェック」は、ハードコピー印刷のためのジョブを開始する前に、マージされた文書の問題点又は他の特徴を発見しようとする自動化印刷前プロセスを記述するために可変データ印刷の分野で使用される用語である。

【0320】

このチェックは、全てのレコードが許容しうる程度に印刷され、余分なことを何も実行しないことを判定するほど単純なものであり、問題がなかったことを報告するか、又は何らかの方法により問題のあるレコードに注意を向ける。

【0321】

プレフライトチェックはユーザの要求に応じて実行され、検査で問題が発見されなかったか、又は問題が発見されたかに関してユーザに警告するのが好ましい。問題が発見されたならば、現れた第1の問題を、その問題の性質の何らかの視覚的方法及びテキストに基づく方法による説明と共にユーザに対し表示することができる。

【0322】

別の実施の形態は、全ての問題点を発見し、調査のためにそれらをユーザに対し列挙することができる。この検査は、ユーザが可変文書テンプレートを編集している間にバックグラウンドで実行され、絶えず更新している問題のリストをウィンドウ又はサブウィンドウに提示するようにしても良い。リストが空であれば、問題が発見されなかったことを示すエントリがそれと置き換えられても良い。このバックグラウンド検査は、ソフトウェアがアイドル状態にあるときにのみ実行されても良く、絶えず実行されても良く、或は他のいくつかの時点又は状況の組み合わせに対して実行されても良い。

【0323】

12. レイアウト方法の概要

この開示の1つの態様は、ページ上にアイテムをレイアウトする方法である。このレイアウトは、レイアウトされるべきアイテムの集積と、そのレイアウトの中でアイテムがどこに配置されるべきかを定義する規則又は制約の集合とを含む。いくつかのレイアウト方法が説明される。それらは次のものを含む。

【0324】

1. レイアウトを定義するためのモデル。レイアウトモデルはレイアウトの中に出現できるアイテムの型及び特性と、それらのアイテムがいかにしてレイアウトされるべきかを定義するための許可された規則又は制約とを定義する。コンピュータ101のワーキングメモリにレイアウトを格納するためにデータ構造が使用されても良い。以下に、いくつかのレイアウトモデルを詳細に説明する。

【0325】

2. レイアウトを作成し、編集する手段。これは、レイアウトを作成するために使用できる処理の集合を含む。それらの処理は、レイアウトモデルに従ってデータ構造を構築するために呼び出されることが可能であるソフトウェア機能として実現されても良い。これらについては以下に詳細に説明する。

【0326】

3. レイアウト中のアイテムの位置及び寸法を計算する手段。アイテム及び規則の集合により定義されるレイアウトが与えられると、このレイアウト計算方法は規則に従ってアイテムをいかにレイアウトすべきかを指定する。この方法はコンピュータ101で実行されるソフトウェア105により形成されても良い。レイアウトを計算する手段については以下に詳細に説明する。

【0327】

10

20

30

40

50

特定のレイアウトモデルに従ってレイアウトを作成し、編集する手段、及びそのレイアウトモデルに関してアイテムの位置を計算する方法は、先に述べたレイアウトエンジン105であるソフトウェアコンポーネントとして一体に実現される。尚、レイアウトエンジン105は、ユーザインターフェース103を含まないことに注意されたい。先に図1Aを参照して説明したように、レイアウトエンジン105と共に、数多くの異なるユーザインターフェースを使用できるであろう。

【0328】

13. レイアウトモデルの概要

レイアウトモデルは、基本レイアウトモデルと、基本モデルのいくつかの拡張とを含む。この基本レイアウトモデルは、レイアウトアイテムの基本特性と、レイアウトを定義するために使用できる規則とを定義する。基本レイアウトモデルを拡張するために使用されるいくつかの追加規則も定義される。レイアウト方法が単純になるため、規則の何らかの組み合わせを使用すると好都合である。レイアウト方法のユーザの理解が簡単である他の組み合わせも好都合である。

10

【0329】

基本レイアウトモデルと追加規則については以下に説明する。規則の特定の組み合わせを含むいくつかの実施の形態を説明する。

【0330】

14. レイアウトの作成及び編集の概要

レイアウトは、通常、文書における1つのページの一部と関連している。作成及び編集の作業は、最初は空レイアウトとして作成されるレイアウトを操作すると想定され、ユーザインターフェースソフトウェア103により呼び出される機能である。作業は、データを文書テンプレートと組み合わせることにより文書を生成している間に、ユーザによる直接入力なしにサーバソフトウェアにより実行されても良い。

20

【0331】

レイアウトを作成又は編集するために使用される操作は、ソフトウェアのユーザにより実行される操作に直接対応していないのが好ましい。ユーザによる単一のアクションは、通常、ユーザインターフェースソフトウェア103により、又は好ましくはレイアウトエンジン105により直接支援される操作よりも高いレベルのインターフェースをレイアウトエンジン105に提供するソフトウェアの別のレイヤにより、複数のレイアウト作成動作に変換される。

30

【0332】

レイアウトを作成し、編集する手段は、少なくとも、

1. アイテムを追加するための動作；
2. アイテムを除去するための動作；
3. 規則を追加するための動作；及び
4. 規則を除去するための動作

を含む。

【0333】

アイテム又は規則を修正するための他の動作が追加されても良い。編集動作の厳密な形態は、使用される特定のレイアウトモデルによって決まる。数多くの異なる形態が実施可能である。それらの動作については以下に詳細に説明する。

40

【0334】

レイアウトにアイテム及び規則を追加するとき、全ての規則を満足できるアイテムの配置は取り得ないという意味で、矛盾している規則の組み合わせを追加することは容易である。そのようなレイアウトを過剰制約付き (over-constrained) という。逆に、所定の規則の組み合わせが数多くの可能解を許すこともあり得る。例えば、全ての規則を満足するアイテムの配置が数多く取り得る場合が考えられる。そのようなレイアウトを制約不足 (under-constrained) という。どのレイアウトも、確実に、厳密に1つの解を有するのが望ましい。従って、レイアウトを作成するために使用される動作に加えて、アプリケーシ

50

ョン121は厳密に1つの解が存在することを保証する手段を更に含む。これはレイアウト作成手段及びレイアウトを計算する手段に組み込まれている。

【0335】

ユーザが解を持たないレイアウト又は2つ以上の解を有するレイアウトを作成することを防止するのが望ましいため、編集動作と関連して、一貫性及び一意性の検査が実行されても良い。そのような検査は、編集動作が終了するたびに、その動作が有効であるか否かを試験するために実行される。有効でなければ、変更は許可されず、直ちにユーザに対して適切なフィードバックが与えられる。

【0336】

一貫性及び／又は一意性の検査は、編集動作の一部として組み込まれていても良い。或は、レイアウト計算により、レイアウト作成動作及び編集動作を使用して作成される各取り得るレイアウトごとに、単一の十分に定義された解が計算されることが保証される。

【0337】

レイアウト解の存在及び一意性を保証する手段については以下に詳細に説明する。

【0338】

15. レイアウト計算の概要

レイアウトモデルごとに、レイアウトアイテムの位置及び寸法を計算する、対応する方法が存在する。それらの位置及び寸法は、問題に対して好都合である何らかの2次元座標系において定義されれば良い。

【0339】

レイアウトアイテムの位置及び寸法を計算するためにいくつかの方法を使用できるであろう。それらは、

1. ツリートラバーサル (tree traversal) 方法；
2. シンプレックス (simplex) 方法；
3. 2次目標関数を伴う変形シンプレックス方法；及び
4. グラフ利用レイアウト (graph based layout)

を含む。

【0340】

ツリートラバーサル方法は、基本レイアウトモデルとほぼ同様の限定レイアウトモデルに適用される。基本モデルの様々な拡張には、その他の方法が適用される。方法2、3及び4は様々な最適化問題を解くことと同等である。

【0341】

最適化問題は制約の集合と、最小化又は最大化されなければならない目標関数とから構成されている。アプリケーション121においては、レイアウトを定義する規則のうちのいくつかは制約を表現し、規則のうちのいくつかは目標関数を定義するために使用される。

【0342】

シンプレックス方法は、特定のクラスの最適化問題を解くための周知の方法である。このシンプレックス方法は、この開示の一部として説明されるレイアウトモデルの一部と共に使用されるのに適している。

【0343】

シンプレックス方法は1次目標関数のみを支援するが、レイアウトが一意的な解を有することを保証するためには、厳密に凸関数 (convex function) を有することが望ましい。線形制約及び2次目標関数により最適化問題を解くための技法は良く知られているため、通常は2次目標関数を使用される。それらの中で最も単純なのがシンプレックスアルゴリズムの変形である。それらの方法は最適化問題を解く技術においてよく知られており、それらの内のどの方法を使用しても、この開示の範囲から逸脱することにはならないであろう。

【0344】

16. レイアウトモデルの詳細な説明

16.1 基本レイアウトモデル

アプリケーション121の一態様は、アイテムの集積をレイアウトする方法を含む。通常、アイテムは、規則又は制約の集合に従って矩形のスペースの中にレイアウトされる。

図22は、アイテム及び制約の集合の一例を示す図である。

【0345】

基本レイアウトモデルは、そのようなレイアウトを定義するために記述される。基本レイアウトモデルは、レイアウトの基本構造と、レイアウトの可変性の限界とを定義するために使用される。例えば、基本レイアウトモデルはレイアウトアイテム間の基本的関係を定義し、アイテムの最小サイズ及び最大サイズを設定し、アイテムの位置に対する制限を設定するために使用される。基本レイアウトモデルの拡張は、レイアウトを更に精密に調整するために使用される。基本レイアウトモデルが有している利点は次の通りである。

【0346】

1. 固定オブジェクト及び可動オブジェクトなどの物理的相似に関してレイアウトデザイナーが容易に理解できる；
2. グラフィカルユーザインターフェースを介して容易に表現、操作できる；
3. 十分に定義された動作の集合を試用して操作されることが可能であるソフトウェアの単純表現を有する。

【0347】

基本レイアウトモデルにおいては、図22に示すように、レイアウトはボックスと呼ばれる1つ或はそれ以上の矩形のレイアウトアイテム2201と、横方向の辺2215及び縦方向の辺2216により定義され、内側にボックスが配置されるべきレイアウト矩形と、0個或はそれ以上の規則(rules)2207、2211を含む。

【0348】

基本レイアウトモデルのアプリケーションの中には、レイアウト矩形が無いものもあるが、典型的には、アイテムがレイアウトされる矩形スペースは、文書のページ、或はページの一部を表しており、各アイテムは、グラフィックイメージ或はテキストボックスを表している。

【0349】

レイアウトにおける各ボックスのサイズと場所は、その境界を規定する矩形の4つの辺の位置により定義される。サイズと場所は、通常はミリメートルのような絶対測定単位と関連しているレイアウト単位で指定されても良い。規則(rules)2207、2209、2211及び2213は、アイテム又はレイアウト矩形の辺の間の関係を定義する。規則は、規則2211、2213のように、同じアイテムの対向する辺の関係を定義しても良いし、規則2209のように1つのアイテムの辺と、別のアイテムの辺との関係を定義しても良いし、或は規則2207のようにアイテムの辺とレイアウト矩形の辺との関係を定義しても良いし、又は規則2217のように、レイアウト矩形の対向する辺同士の間を定義しても良い。この説明において、辺はアイテム又はオブジェクトの左、右、上又は下を意味すると定義されている。

【0350】

基本レイアウトモデルでは、各ボックスの辺は「整列マーク」又は単に「マーク」と呼ばれる抽象レイアウトアイテムと関連付けられる。レイアウトエリアの辺もマークと関連付けられる。従って、基本レイアウトモデルは2種類のレイアウトアイテム、即ち、ボックスとマークとを支援する。

【0351】

「整列マーク」という用語は、互いに重ね合わせて印刷されるべき複数の画像を整列させるために、プリンタにより使用される位置合わせマークを抽象化、一般化したものである。この整列マークには、水平(横)と垂直(縦)の2種類がある。水平マークは、ページ又はレイアウトにおける縦方向の位置を表現し、不定又は無限の長さの水平線であると考えられても良い。垂直マークは、ページにおける水平方向の位置を表現し、概念上は不定又は無限の長さの垂直線である。

【0352】

整列マークは一般に、印刷されることを意図されておらず、単にボックス間の位置決め及び関係を定義するために存在している。この整列マークはユーザインターフェースにより様々な方法により表示されても良い（又は全く表示されなくても良い）。

【0353】

レイアウトは2次元座標系と関連付けられる。選択された座標系における各マークの位置又は座標を判定するのは、レイアウトエンジン105のジョブである。垂直マークは横方向の座標を有し、水平マークは縦方向の座標を有する。座標系は、1つの指定された垂直マークと、1つの指定された水平マークとに座標を割り当てることにより指定されるのが好ましい。他の全てのマークの座標は、指定マークに対するそれらのマークのオフセットにより決定される。レイアウトが一定のサイズである場合、座標系を指定するために使用されるマークは、通常、レイアウト矩形の2つの辺である。

10

【0354】

ボックスは、通常、何らかのテキスト又は図形と関連付けられる矩形エリアである。ボックスは、レイアウトを計算するときに、コンテナを表現するために使用されても良い。各ボックスは、そのボックスの辺を定義する4つのマークと関連付けられる。ボックスは本質的には4つのマークの関係である。基本レイアウトモデルの拡張においては、追加マークがボックスと関連付けられても良い。基本レイアウトモデルでは、ボックスの位置及び寸法を決定するために使用されるレイアウト規則は、それらのボックスの辺と関連付けられるマークに関して完全に記述されるが、モデルの拡張においては、追加規則がボックスと関連付けられる。

20

【0355】

ボックス及びマークは、文書に現れるオブジェクトの形状及び位置に直接には対応しない形状及び位置を表現するためにも使用される。例えば、ボックスは、レイアウトエンジン105により計算されたサイズとは異なることもあるアイテムの理想寸法を表現するためにも使用される。マークは他のアイテムの配置に対する制限を定義するために使用されても良い。例えば、マークはページマージンを表す位置に配置されても良く、レイアウトエンジン105がページマージンの外側にアイテムを配置しないことを保証するために規則が追加されても良い。

【0356】

30

基本モデルでは、各規則は縦オフセット2209又は横オフセット2213を表現し、指定サイズ2207又は未知の負ではないサイズ（破線により示す）2213のいずれかを有していれば良い。固定オフセット規則は、レイアウト中の2つのマークの間の縦又は横方向のオフセットのサイズ及び方向を指定する。未知オフセット制約は、2つのマークの間のオフセットの方向のみを指定する。例えば、オフセット2211は、ボックス2219の辺2221が常に同じボックスの辺2223の上方に位置していなければならないことを指示する。未知オフセット制約のサイズは、レイアウト方法により計算される必要がある値を表す。

【0357】

従って、基本モデルには、固定オフセット規則及び非負オフセット規則という2種類の規則がある。どの型のオフセット規則も、1対のマークの相対位置に対する制約を定義する。

40

【0358】

固定オフセット規則は、第1のマークの位置と第2のマークの位置との間のオフセットが指定値を有していなければならないことを指示する。例えば、マークm及びnが与えられた場合、固定オフセット規則は式

$$\text{pos}(n) - \text{pos}(m) = d$$

により定義されるfixed (m, n, d) により表される制約である。式中、dは非負数である。ストラット412は、レイアウトエンジン105において固定オフセット規則として表現される。また、コンテナが固定幅を有する場合、これは、コンテナの左右の辺を表現す

50

るマークに関連する固定オフセット制約を追加することにより、レイアウトエンジン 105 に対して指示される。同様に、コンテナが固定高さを有する場合、これは、レイアウトエンジン 105 において、コンテナの上下の辺を表現するマークの間の固定オフセット制約により表される。

【0359】

距離 d は、一方のマークから他方のマークへのオフセットを表し、 $pos(u)$ はマーク u の位置を表す。水平マークの位置はレイアウト座標系の原点から、そのマークまでの縦方向の距離である。垂直マークの位置は、レイアウト座標系の原点からマークまでの横方向の距離である。レイアウトを定義するのに好都合であるどのような座標系が使用されても良い。固定オフセット規則では、2つのマークは同じ向きでなければならない。この点に関して、マークは共に水平マークでなければならないか、又は共に垂直マークでなければならない。

10

【0360】

尚、この説明で使用される意味での用語「固定」は、レイアウト計算方法に対する入力であり、レイアウトエンジン 105 により変更されることができない値を表すことに注意されたい。

【0361】

非負オフセット規則は、2つのマークの間のオフセットが負ではないことを指定する制約である。マーク m とマーク n との間の非負規則は、 $non-negative(m, n)$ により表される制約であり、不等式

20

$$pos(n) \geq pos(m)$$

により定義される。非負オフセット規則では、2つのマークは同じ向きでなければならない。例えば、マークは共に水平マークでなければならないか、又は共に垂直マークでなければならない。

【0362】

基本レイアウトモデルにおいては、水平規則は垂直制約からは独立しており、そのため、レイアウトを計算する問題は2つの別個の問題に分割できる。図 23 は、図 22 に示すレイアウトに対応する垂直オフセットのみを示している。この実施の形態では、距離は論理単位で指定されている。図 23 において、固定オフセット規則は、論理単位によるオフセットのサイズによってラベル付けされている。この場合、単純にするため、各論理単位は 1 mm に対応している。論理単位はアプリケーションに都合の良いどのようなサイズであっても良いが、通常、1 論理単位は独立してアドレッシング可能な最小のプリンタドットより小さいであろう。図 23 に示す垂直オフセット規則は、図 24 に示すような有向グラフの形で表現されても良い。図 24 において、各頂点 2401 は水平マークを表し、各矢印 2403 は規則を表す。固定オフセット規則（例えば、2402）は実線で示され、非負オフセット規則（例えば、2405）は破線で示されている。図 24 では、例えば、ルートノード 2404 と末端ノード 2406 との間の矢印 2402 は、図 23 からの固定オフセット規則 2217 を表し、図 22 のレイアウト矩形の高さを定義している。更に、非負オフセット規則は、固定オフセット規則により確定される境界の中で可変である。例えば、 h_1 の値が増加すると、共に h_1 の末端ノード 2408 から下がって来る h_2 の値及び h_3 の値も対応して増加する。

30

40

【0363】

通常、レイアウトエリアのサイズは固定されている。これは、レイアウトエリアの対向する辺に関連するマークを固定オフセット規則によって結合することにより指定される。アプリケーションによっては、レイアウトエンジン 105 がレイアウトエリアの辺と、レイアウト中のアイテムとの関係に基づいて、レイアウト矩形のサイズを計算することにも意味がある。通常、印刷可能なアイテムを表現するボックスがレイアウトエリアの中に入るように制約されることを保証するためにレイアウトに規則が追加されるが、実施の形態によっては、これが適用されない場合もあり、レイアウトエリア矩形が要求されない場合もある。基本レイアウトモデルのいくつかのアプリケーションにおいては、レイアウト矩

50

形は欠落していてもよい。

【0364】

16.2 基本レイアウトモデルの代替表現

別の実施の形態では、基本レイアウトモデルは先に説明した方法とは異なるが、それと等価の方法で表現される。この場合、基本レイアウトモデルはただ1つの種類の規則、即ち、最小オフセット規則を使用して表現される。

【0365】

最小オフセット規則は、第1のマークと第2のマークとの間の最小許容オフセットを指定する。マークmとマークnとの間の最小オフセット規則は、 $\min(m, n, d)$ により表され、不等式

$$\text{pos}(n) - \text{pos}(m) \geq d$$

により定義される。式中、dは最小許容オフセットを表す数である。この表現においては、数dは正、負又は0のいずれであっても良く、最小オフセット規則により関連付けられる2つのマークは同じ向きでなければならない（即ち、共に水平マークであるか、又は共に垂直マークである）。

【0366】

別の等価の実施の形態では、最小オフセット規則の代わりに最大オフセット規則が使用されても良い。 $\max(m, n, d)$ により表される最大オフセット規則は不等式

$$\text{pos}(n) - \text{pos}(m) \leq d$$

により定義される。式中、dは最大許容オフセットを表す数である。

【0367】

これら2種類の規則のうち的一方のみが要求されることを理解するために、等式

$$\max(m, n, d) \equiv \min(m, n, -d)$$

があるために、どの最大オフセット規則も等価の最小オフセット規則と置き換えできるとに注意する。

【0368】

基本レイアウトモデルの代替表現は初めに説明した表現と等価であることを理解するために、等式

$$\text{nonnegative}(m, n) \equiv \min(m, n, 0)$$

からわかるように、どの非負オフセット規則も最小オフセット規則の特殊ケースであり、等式

$$\begin{aligned} \text{fixed}(m, n, d) &\equiv \min(m, n, d) \text{ 及び } \max(m, n, d) \\ &\equiv \min(m, n, d) \text{ 及び } \min(m, n, -d) \end{aligned}$$

からわかるように、どの固定オフセット規則も2つの最小オフセット規則と等価であることに注意されたい。

【0369】

逆に、どの最小オフセット規則も、特別なマークt、固定オフセット規則及び非負オフセットを使用して、等式

$$d \geq 0 \text{ であれば、} \min(m, n, d) \equiv \text{fixed}(m, n, d) \text{ 及び } \text{nonnegative}(t, n)$$

$$d < 0 \text{ であれば、} \equiv \text{fixed}(t, m, -d) \text{ 及び } \text{nonnegative}(n, t)$$

により示されるように表現されることが可能である。式中、tはmに対してnの許可位置の限界をマーキングするために使用される特別なマークである。d=0である特別なケースにおいては、tはmと置き換えられても良く、固定オフセット規則は冗長である。

【0370】

従って、最小オフセット、最大オフセット、固定オフセット及び非負オフセットを指定することを可能にする数多くの同値の表現がある。最小オフセット規則は、レイアウトを表現するために使用されるデータ構造及びそのデータ構造を操作するためのソフトウェアを簡略化するので、これらの全ての種類の規則を表現するために使用できるであろう。

【0371】

基本レイアウトモデルに準拠したレイアウトを表現するために有向グラフを使用するこ

10

20

30

40

50

とも好ましく、その場合、各グラフの頂点は1つのマークに対応し、グラフの各辺は辺のソース頂点により表されるマークから辺の目的頂点により表されるマークまでの最小許容オフセットを表す数でラベル付けされている。従って、与えられるどのレイアウトに対しても、レイアウト中の各々のマークがグラフの対応する頂点により表され、各最小オフセット規則が対応する有向辺により表されるようなグラフ表現を生成できるであろう。基本レイアウトモデルでは、各ボックスはその辺に対応するマークにより完全に記述されるため、表現においてはボックスは無視されても良い。

【0372】

コンピュータのメモリにおいてグラフを表現する方法は数多く存在し、適切などの表現を使用しても差し支えない。一般に、表示及び編集を目的とする場合には、アイテム及び制約から構成されるレイアウトをグラフ表現以外の形態で格納するのが好都合であり、そのため、通常、インタラクティブアプリケーションにおいて、レイアウトは必ずしもグラフの形態で直接に表現されるとは限らない。グラフ表現は、グラフの辺及び頂点を必要に応じて数及び他の情報でラベル付けする手段を含んでいても良い。

【0373】

図33Aから図33Cは、固定オフセット規則及び非負オフセット規則のみを使用した場合、及び最小オフセット規則のみを使用した場合に、コンテナの最小高さ及び最大高さをどのように表現できるかを示す図である。図33Aでは、40単位の最小許容高さ、100単位の最大高さを有するコンテナ3300が示されている。コンテナ3300はレイアウトエンジン105において、マーク3301と関連付けられた上辺及びマーク3303と関連付けられた下辺を有するボックスにより表現されている。図33Bの有向グラフは、4つの頂点3305、3307、3309及び3311を有する。頂点3305はマーク3301を表し、頂点3307はマーク3303を表す。他の2つの頂点3309及び3311は、マーク3301に対してマーク3303に許可される移動の範囲を定義する特別のマークを表す。それらのマークの位置は、グラフ中の実線の辺3317により表される固定オフセット規則を使用して、マーク3301に対して固定されている。グラフ中の破線の辺3319により表される2つの非負オフセット規則は、マーク3303の位置を2つの特別のマーク3311及び3309の間にあるように制約するために使用される。図33Cの第2の有向グラフは2つの頂点3313及び3315のみを使用して同じ制約を表すが、頂点3313はマーク3301を表し、頂点3315はマーク3303を表す。2つの辺3319は、最小オフセット規則を表すために使用される。頂点3313から頂点3315への辺は、最小値40を有する最小オフセット規則を表す。頂点3315から頂点3313への辺は、(最小オフセット-100)を有する最小オフセット規則を表す。これは100の最大値を有する頂点3313から3315への最大オフセット規則と等価である。

【0374】

16.3 形状規則

可変データ印刷アプリケーションにおいて、ボックスの幅と高さの関係を定義することが可能であるのが好都合である。基本レイアウトモデルでは、これを実行する方法はない。

【0375】

例えば、ボックスはレイアウトに挿入されるべき画像に対応し、画像のサイズは利用可能スペースにはめ込まれるように調整される必要があると考えて良いであろう。そのような場合、ボックスの縦横比を指定し、レイアウトエンジン105にレイアウト中の他のアイテムのサイズに基づいてボックスの最良のサイズを判定させることが可能であることが望ましい。別の例として、ボックスは高さ及び幅が指定されていないテキストのブロックの境界規定ボックスに対応していても良く、従って、高さ及び幅は共にレイアウトエンジン105により判定される必要がある。

【0376】

基本レイアウトモデルは、ボックスの幅と高さの関係を定義する形状規則と呼ばれる追

10

20

30

40

50

加の規則を可能にし、且つ縦横比規則及びテキスト規則という 2 種類の形状規則を含めるように拡張されても良い。

【0377】

縦横比規則は、ボックスが特定の縦横比を有していなければならないことを指定する。テキスト規則は、ボックスがテキストの特定のブロックに対して境界規定ボックスとして整形されるべきであることを指定する。テキストのレイアウトは専用モジュールにより処理される。テキストレイアウトの例については後に説明する。

【0378】

形状規則の厳密な挙動は、レイアウトを計算するために使用される方法によって決まるため、その挙動については以下に更に詳細に説明する。それらの規則は、アイテムの位置及び寸法を計算するために選択される方法に応じて、様々に異なる実施の形態において様々に異なる挙動を有するであろう。

16.4 等オフセット規則 (Equal Offset Rules)

レイアウトを指定するときに設定されていると好都合である別の種類の制約は、2つの距離が等しいことを指定可能であることである。これは、同じ向きを有するマーク m 及び n と、同じ向きを有するマーク s 及び t に対して $\text{equal}(m, n, s, t)$ により表される等オフセット規則を使用して指定されることが可能である。等オフセット規則 $\text{equal}(m, n, s, t)$ は式

$$\text{offset}(m, n) = \text{offset}(s, t)$$

により定義される。式中、 $\text{offset}(a, b)$ は、

$$\text{offset}(a, b) = \text{pos}(b) - \text{pos}(a)$$

となるような、第 1 のマーク a から第 2 のマーク b までのオフセットを表す。

【0379】

基本レイアウトモデルは、等オフセット規則の追加により拡張されることが可能である。レイアウトモデルにおいて等オフセット規則が許可されると、過剰に制約されたレイアウトを作成するのが非常に容易になる。

【0380】

等オフセット規則は、レイアウトエンジン 105 において好適な方法より効率の点で劣る、より一般的なレイアウト方法を要求するため、全ての実施の形態で支援されるわけではない。レイアウトエンジン 105 は、様々なオフセットを等しくするための代替手段として、等オフセット規則の代わりに以下に定義する距離最小化規則を含むのが好ましい。

【0381】

16.5 ボックスの中心を含む規則

基本レイアウトモデルでは、ボックスの辺のみがレイアウト規則に関与している。ボックスの中心間の関係を定義することも望ましい。基本レイアウトモデルの拡張においては、各ボックスは常に 2 つの追加マークと関連付けられる。それらは、共にボックスの中心を通る垂直マーク及び水平マークから構成されている。

【0382】

等オフセット規則を可能にするモデルにおいては、等オフセット規則を使用してマークを各ボックスの中心と関連付けることができ、従って、ボックスの中心を含む規則を支援するためにモデルへの追加は不要である。

【0383】

16.6 固定中心規則 (Fixed Center Rules)

ボックスの中心を含む規則の特別なケースとして、別の実施の形態では、各ボックスの中心の縦方向位置又は横方向位置をページに対して、或はページ又はレイアウトエリアの 1 つの辺に対して固定させるような基本レイアウトモデルの拡張がある。この特別なケースは、等オフセット規則の必要性を回避するので、より単純なレイアウト計算方法を使用できる。

【0384】

固定中心規則は、1 つのボックスの対向する辺と関連付けられる 1 対のマークの関係で

10

20

30

40

50

あると考えることができ、従って、ボックスの中心を通る追加のマークをレイアウトに追加する必要はない。

【0385】

16.7 オフセット最大化規則及びオフセット最小化規則

以上説明した基本モデルレイアウト規則及び追加規則は、アイテムの厳密なサイズ及びアイテムの間隔などの精密な制約を指定するため、並びにアイテムの位置及びサイズの可変性に対する制限を指定するために有用である。ここまで説明した規則によれば、過剰制約又は制約不足であるレイアウトの定義を容易に実現することができる。レイアウトを、より精密なレベルでコントロールするための良い戦略は、基本モデル規則、並びに等オフセット規則及び固定中心規則などの他の精密な規則を使用して、制約不足レイアウトを定義し、アイテムのサイズ及び位置の優先性を指定するための追加の、更に融通性に富む規則を追加することである。この目的のために、最小化規則及び最大化規則という2種類の規則を更に使用しても良い。

【0386】

ここまで定義された規則は、レイアウト方法が従わなければならない制約を定義している。先に定義された規則とは異なり、最小化規則及び最大化規則はレイアウト方法の目標関数を定義する。

【0387】

$\text{minimizeoffset}(m, n)$ により表されるオフセット最小化規則は、マーク m からマーク n までのオフセットをできる限り小さく（即ち、負に）すべきであることをレイアウトエンジン105に指示する。この規則の挙動はレイアウトを計算する方法によって異なるため、以下に詳細に定義される。

【0388】

$\text{maximizeoffset}(m, n)$ により表されるオフセット最大化規則は、マーク m からマーク n までのオフセットをできる限り大きく（即ち、正に）すべきであることをレイアウトエンジン105に指示する。この規則の挙動はレイアウトを計算する方法によって異なるため、以下に詳細に定義される。

【0389】

オフセット最小化規則及びオフセット最大化規則は、1次目標関数を算入するのに有用であり、従って、シンプレックス方法を使用する計算に適するレイアウトモデルに含まれても良い。1次目標関数を使用される場合、最小化規則又は最大化規則は、それぞれ、1つの1次項を目標関数に与える。この場合、規則 $\text{minimizeoffset}(m, n)$ は、 $\text{maximizeoffset}(m, n)$ と等価であるため、2種類の規則のうちの一方のみが必要である。従って、これらの種類の規則を含むレイアウトモデルのデータ表現は、それらの種類の規則の内の一方を支援するだけで良い。例えば、データ構造はオフセット最小化規則のみを支援しても良く、各オフセット最大化規則は等価のオフセット最小化規則を使用して表現される。

【0390】

16.8 距離最小化規則 (Minimize Distance Rules)

オフセット最大化規則及びオフセット最小化規則の使用により、2つのマークの間の理想の離間が無限であるようなレイアウトがもたらされる場合があり、従って、これらの規則が常に十分に定義されたレイアウトを提供するとは限らない。この問題を回避するために、オフセット最大化規則及びオフセット最小化規則は、別の種類の規則と置き換えられても良い。或は、 $\text{minimizedist}(m, n)$ により表される距離最小化規則と呼ばれる別の種類の規則を使用して、距離を指定する。例えば、2つのマーク m 及び n の間のオフセットの絶対値はできる限り小さくなければならない。距離最小化規則は同じ向きを有するマークの間にのみ適用される。特定の実施の形態においては、距離最小化規則を使用する代わりに、以下に説明する、好適オフセット規則と呼ばれる別の種類の規則が使用される。

【0391】

尚、距離最小化規則は、固定オフセット規則、オフセット最小化規則及びオフセット最大化規則並びに非負オフセット規則を使用して、下記の等式を使用して近似されることが

可能であることに注意する。

【0392】

$\text{minimizedist}(m, n) \equiv \text{minimizeoffset}(m, t)$ 及び $\text{minimizeoffset}(n, t)$ 及び $\text{nonnegative}(m, t)$ 及び $\text{nonnegative}(n, t)$

この式中、 t はレイアウトに追加され、他の規則のいずれにも含まれない特別のマークである。この近似は厳密ではないこともあり、使用される厳密なレイアウト計算方法によって左右される。特に、レイアウト方法により計算されるアイテムの厳密な配置は選択される目標関数によって決まる。

【0393】

尚、オフセット最小化規則も距離最小化規則を使用して、下記の等式を使用して近似されることが可能であることに注意する。

【0394】

$\text{minimizeoffset}(m, n) \equiv \text{minimizedist}(t, m)$ 及び $\text{fixed}(t, n, d)$

式中、 t はレイアウトに追加された特別のマークであり、 d は m と n の間の所望の最大距離よりはるかに大きい距離を表す大きな正の数である。この近似は厳密ではなく、アイテムの厳密な配置はレイアウト計算方法の厳密な実施の形態によって決まるが、固定オフセット規則及び非負オフセット規則を可能にするレイアウトモデルに距離最小化規則のみを追加することにより、オフセット最大化規則及びオフセット最小化規則に類似する機能性を提供できることが判明している。

【0395】

実施の形態によっては、距離最小化規則は規則の強さを示す特別の値を含んでいても良い。そのような規則は $\text{minimizeoffset}(m, n, s)$ で表されるが、 m 及び n は規則により関連付けられるマークであり、 s は規則の強さを表す正の数である。この場合、2つ以上の距離最小化規則がマークの位置に影響を及ぼしているとき、より強い規則が、弱い規則より大きな影響を位置に対して与える。

【0396】

16.9 好適オフセット規則

特定の実施の形態においては、距離最小化規則を使用する代わりに、「好適オフセット規則」と呼ばれる別の種類の規則が使用される。 $\text{preferred}(m, n, d)$ により表される好適オフセット規則は、マーク m からマーク n までの好ましいオフセットが d であることを指定する。好適な規則は下記の等式によって距離最小化規則により表現されることができ

【0397】

$\text{preferred}(m, n, d) \equiv \text{fixed}(m, t, d)$ 及び $\text{minimizedist}(t, n)$

式中、 t は m の位置に対する n の好ましい位置を表すためにレイアウトに追加された特別のマークである。

【0398】

逆に、距離最小化規則は下記の等式によって好適オフセット規則を使用して表現されることができる。

【0399】

$\text{minimizedist}(m, n) \equiv \text{preferred}(m, n, 0)$

従って、基本レイアウトモデルの何らかの拡張に距離最小化規則又は好適オフセット規則のいずれかを追加すると、その結果、等価モデルが得られる。好適オフセット規則は、通常、一般に指定されるレイアウト関係に対して距離最小化規則より少ない数のマーク及び少ない数の規則を要求する表現であるため、距離最小化規則の代わりに使用される。

【0400】

好適オフセット規則は、必要に応じて、規則の強さを示す特別の値を更に含んでいても良い。そのような規則は、 $\text{preferred}(m, n, d, s)$ と表され、 m 及び n は規則により関連付けられるマークであり、 d は m から n までの好ましいオフセットを表す数であり、 s は規則の強さを表す正の数である。この場合、2つ以上の好適オフセット規則がマークの位

10

20

30

40

50

置に影響を及ぼしているとき、より強い規則が弱い規則より大きな影響を位置に対して与える。mの位置に対するnの好ましい位置を表すためにレイアウトに追加された特別のマークをtとすると、強さを有する好適オフセット規則は、下記の等式によって強さを有する距離最小化規則と等価である。

【0401】

$\text{preferred}(m, n, d, s) \equiv \text{fixed}(m, t, d)$ 及び $\text{minimizedist}(t, n, s)$

及び $\text{minimizedist}(m, n, s) \equiv \text{preferred}(m, n, 0, s)$

16.10 好適レイアウトモデル

望ましいレイアウトモデルの1つは、先に定義した下記の種類の規則を追加するために拡張された基本レイアウトモデルを含む。

10

【0402】

1. ボックスの中心を横方向又は縦方向に固定するための規則；
2. テキスト及び画像を含むボックスを支援するための形状規則；及び
3. 好適オフセット規則

このレイアウトモデルは、ユーザが十分に理解できるほど単純である一方で、一般に必要とされる全てのレイアウト特徴を支援するのに十分な融通性を有し、単純なデータ表現を有するほど十分に単純であり、レイアウトアイテムの位置及び寸法を計算するための高速の方法を支援することが可能である。

【0403】

16.11 レイアウト例

20

図37Aは、図4に示すレイアウト例が、ボックス、マーク及び規則を使用してレイアウトエンジン105の一実施の形態においてどのように表現されるかを示す図である。レイアウトエリアの境界は、マーク3701、3703、3705及び3707により表されている。レイアウトエリアは、通常、テンプレート中の1つのページの一部を表す。レイアウトエリアの高さは、固定オフセット規則3709により表されている。レイアウトエリアの幅は、固定オフセット規則3701により表されている。図4に示す2つのコンテナは、レイアウトエンジン105において、図37Aに示される2つのボックス3702及び3704により表されている。マーク3701及び3705は、原点マークとして指定されている。

【0404】

30

第1のボックス3702の4つの辺は、マーク3713、3715、3717及び3719により表されている。ボックス3702の左上角は、ページ150上の、レイアウトエリアの左辺から右側へ150論理単位離れ、レイアウトエリアの上から200単位下だった固定位置を有する。これは2つの規則3735及び3737により表されている。規則3735は、ボックス3702の左辺がレイアウトエリアの左辺から150単位右にあることを保証し、規則3737はボックス3702の上辺がレイアウトエリアの上から200単位下だった位置にあることを保証する。ボックス3702の高さと幅は固定されておらず、従って、レイアウトエンジン105により計算される。最小オフセット規則3744は、ボックスの最小幅が120論理単位であることを指示する。規則3742は、ボックスの最小高さが100論理単位であることを指示する。好適オフセット規則3741は、ボックス3702の好ましい高さが550論理単位であることをレイアウトエンジン105に指示する。別の好適オフセット規則3743は、ボックス3702の好ましい幅も550論理単位であることをレイアウトエンジン105に指示する。ボックスの好ましい幅及び高さは、ボックスが表現しているコンテナのコンテンツから決定される。最小オフセット規則3739は、ボックス3702の下辺が、マーク3703により表されるレイアウトエリアの底辺の下方にあってはならないことをレイアウトエンジン105に指示する。

40

【0405】

第2のボックス3704の4つの辺は、マーク3721、3723、3725及び3727により表されている。ボックス3704の上辺及び下辺の位置は、レイアウトエリア

50

に対して固定されている。上辺の位置はマーク 3725 により表されている。上辺は、固定オフセット規則 3731 により、レイアウトエリアの上から 200 論理単位下がった位置で固定されている。同様に、ボックスの下辺は規則 3733 により、レイアウトエリアの上から 750 論理単位下がった位置で固定されている。

【0406】

ボックス 3704 の幅は、固定オフセット規則 3729 により 550 論理単位に固定されているが、ボックスの左右の辺の位置は固定されておらず、レイアウトエンジン 105 により計算されなければならない。最小オフセット規則 3747 は、マーク 3723 により表されるボックスの右辺がマーク 3707 により表されるレイアウトエリアの右辺を過ぎるほど出てはならないことをレイアウトエンジン 105 に指示する。

10

【0407】

更に別の最小オフセット規則 3748 及び 3749 は、可変辺がレイアウトエリアの内側に留まることを保証する。この例では、それらの規則は冗長であり、実施の形態によっては、レイアウトを計算する前に冗長規則は取り除かれても良い。冗長規則の検出は、規則を一時的に取り除き、プッシュ動作を使用して、影響を受けるマークをプッシュすることにより規則違反が起こる可能性があるか否かを判定することにより実行されても良い。プッシュ動作を使用して規則違反になる可能性がなければ、その規則は冗長である。

【0408】

ストラット 412 は固定オフセット規則 3745 により表されている。この規則は、2 つのボックス 3702 及び 3704 の距離が 200 論理単位でなければならないことをレイアウトエンジン 105 に指示する。

20

【0409】

図 37B は、図 37A に示す水平オフセット規則を表すグラフである。垂直オフセット規則を表すために同様のグラフ（図示せず）を使用できるであろう。頂点 3751 は、レイアウトエリアの左辺に対応するマーク 3705 を表す。頂点 3761 は、レイアウトエリアの右辺に対応するマーク 3707 を表す。頂点 3753 は、ボックス 3702 の左辺に対応するマーク 3713 を表す。頂点 3755 は、ボックス 3702 の右辺に対応するマーク 3715 を表す。頂点 3757 は、ボックス 3704 の左辺に対応するマーク 3721 を表す。頂点 3759 は、ボックス 3704 の右辺に対応するマーク 3723 を表す。

30

【0410】

固定オフセット規則 3735 は、グラフの辺 3767 及び 3769 により表される 1 対の最小オフセット規則により表されている。最小オフセット規則 3744 は、グラフの辺 3771 により表され、最大オフセット規則 3746 はグラフの辺 3773 により表されている。固定オフセット規則 3745 は、グラフの辺 3775 及び 3777 により表されている。固定オフセット規則 3729 はグラフの辺 3779 及び 3781 により表されている。最小オフセット規則 3747 はグラフの辺 3783 により表されている。

【0411】

図 37C は、図 37B に示すグラフを格納するために使用されるメモリ構造を示す。グラフの各頂点は、少なくともその頂点が表現するマークの位置に加えて、頂点から出ている辺を表す隣接度リストを含む、例えば、3791 のようなデータ構造により表される。隣接度リストは、各々が 1 つの辺を表し、目的頂点までの、ポインタ 3793 のような少なくとも 1 つのポインタと、ソース頂点から目的頂点までの最小許容オフセット（例えば、3794）とを含む複数のレコードから構成されている。隣接度リストのサイズは変化するため、それらは連係リストとして格納される。隣接するレコードの間のリンクは、図 37C には明示して示されていない。

40

【0412】

好適オフセットは、図 37D に示すような別個のグラフに格納される。好適オフセットは、辺 3795 のようなグラフ中の 1 つの辺によりそれぞれ表される。好適オフセットグラフを格納するためのデータ構造は、最小オフセット規則を格納するために使用されるデ

50

ータ構造に類似している。2つのグラフにおける対応する頂点はポインタ（明示せず）により連係されている。実施の形態によっては、各辺が逆方向の辺を有することを保障するために、各グラフに特別の辺が追加される。これは、グラフを順方向又は逆方向に効率良くトラバースすることを可能にするためである。

【0413】

17. レイアウトの作成及び編集の詳細な説明

17. 1 レイアウトを作成及び編集するための基本動作

レイアウトを作成及び編集する最も単純なアプローチは、任意の種類のアイテムを追加又は除去させることと、支援されるレイアウトモデルによって許可された任意の種類の規則を追加又は除去させることである。このアプローチにおいて、除去されるアイテムを含む任意の規則を自動的に除去することによって1つのアイテムがレイアウトから除去されるとき、最小限の一貫性チェックしか行なわない場合は、レイアウトデータ構造が矛盾することを防止する必要がある。このアプローチにおいては、全ての規則を満たすアイテムの位置及び寸法を決定できるという保証はない。

【0414】

また冗長な情報を含むレイアウトを回避することは有用である。これは、レイアウトの複雑性を減少でき、これにより単純化されてユーザが理解しやすくなり、また冗長な情報の除去処理がユーザにフィードバックを提供することによってユーザにとってより明確なレイアウトを実現できるためである。また冗長な情報は、レイアウトを表すデータ構造のサイズを大きくし、レイアウト計算速度を低下させる。

【0415】

基本レイアウトモデルによる規則を含むレイアウトを表すために説明される有向グラフ構造（図24）は、数種類の冗長な情報を許可しない手段を自動的に提供する。このグラフは、同一の発信元頂点及び目的頂点を有する1つ以上の有向辺を許可しない。これにより、1つ以上の最小オフセット規則が、同一のオフセットに適用されるのを防止する。（同一の順序で、即ち、同一の第1マーク及び同一の第2マークを有する）一対の同一マークの間に1つ以上の最小オフセット規則が存在する場合、最も負の最小オフセット値を有する規則のみが必要であるため、規則の内の1つは冗長となる。

【0416】

最小オフセット規則を追加する動作は、新しく追加された規則を有する同一の発信元頂点及び目的頂点で既に表された任意の既存の最小オフセット規則を自動的に置換する動作であることが好ましい。或は、最小オフセット規則を既存のレイアウトに追加する動作は、同一のオフセットに対してより小さい（即ち、より負の）最小オフセット値を有する最小オフセット規則が既に存在する場合、空命令となる。

【0417】

特定の実施の形態において、この動作は、基本レイアウトモデルによって支援される全種類の規則を追加及び除去するために支援される。これらの規則は、最小オフセット規則を使用して全て表された場合でも、固定オフセット規則、非負のオフセット規則、及び最大オフセット規則を備える。例えば、固定オフセット規則を追加する動作は、2つの対応する最小オフセット規則を追加することによって実現される。動作は、任意のマークの位置を固定又は非固定にするために提供される。マークの位置を固定することは、同一方位のマーク及び原点マークの間に、固定オフセット規則を追加することと同様である。また、動作は、任意のボックスの中心の横方向位置又は縦方向位置を独立して固定するためにも提供される。また、多数のレイアウト作成及び編集動作を組み合わせても良い。

【0418】

ボックスの中心の横方向の位置は、様々な理由によって効果的に固定されても良い。例えば、ボックスの2つの縦辺の横方向の位置がレイアウトエリアに固定される場合、ボックスの横方向の位置及び幅は完全に決定される。そのため、ボックスの中心の横方向位置も固定される。この場合、規則が冗長であるため、横方向位置をレイアウトに対する固定ボックスの中心とする（即ち、固定中心規則を追加する）動作は、空命令となる。ボック

スの中心の縦方向の位置を固定することについても同様である。

【0419】

1つの実施の形態において、好適オフセット規則及び最小距離規則を追加及び除去する動作も同様に支援される。本実施の形態において、最小距離規則は、同様の好適オフセット規則を使用して表される。好適オフセット規則は、有向グラフで表されてもよく、このグラフでは、各頂点がマーク及び各好適オフセット規則を、好適オフセット規則によって関連付けられるマークに対応する頂点を接続する辺によって表す。有向グラフが同一の発信元頂点及び目的頂点を有する単一の辺を許可するだけでは、同一の2つのマークに関する1つ以上の好適オフセット規則を有するレイアウトを表すことは不十分となる可能性があるが、好適オフセット規則が強度を有する特定の実施の形態においては、同一の2つのマークに関する任意の好適オフセット規則は、単一で等価の好適オフセット規則によって、以下の等式を介して置換されることができる。

10

【0420】

$\text{preferred}(m, n, d, s)$ 及び $\text{preferred}(m, n, d', s') \equiv \text{preferred}(m, n, D, S)$

式中、 $D = (sd + s'd') / (s + s')$ であり、

$S = s + s'$ である。

【0421】

この等式は、好適レイアウト計算が行なわれる場合に有効となる。

【0422】

17. 2 レイアウト作成及び編集に基づく例

20

レイアウトを作成及び編集する場合、過剰に制約されたレイアウトを作成するのを回避するのが望ましい。このようなレイアウトは、文書を生成するのに有用でないためである。矛盾する規則を回避する処理は、ユーザに対してフィードバックを提供することによって、作成されるレイアウトに関するユーザの理解をより深くすることもできる。レイアウトの作成中、一貫性のない規則を回避することによって、誤りを自動的に防ぐことにより、ユーザの仕事を減少させることもできる。

【0423】

例示に基づいた編集は、過剰に制約されたレイアウトを回避するために、所定の実施の形態において使用されても良い。例示に基づいた編集において、レイアウトを作成及び編集するための動作は、全ての制約に従ったレイアウトの例示上で常に実行されている。なお、最小オフセット規則、最大オフセット規則、最小距離規則、及び好適オフセット規則は、レイアウトに制約を課さず、また、過剰に制約されたレイアウトを回避するために無視されてもよい。可変データ印刷アプリケーションでは、ユーザが文書のプレビューを示す例示的なレイアウトを構築することを許可することによってこの処理が行なわれる。プレビューは、文書テンプレートと実データとを組み合わせた文書の1つの表示であっても良い。また、プレビューは、文書テンプレートに挿入されると予測されるデータに単に類似しているサンプルデータを含んでも良い。或は、アイテムのサイズ及び位置のみが、テンプレートから生成される実際の文書において予測されるデータとほぼ同一であっても良い。

30

【0424】

40

新たに作成されたアイテムの位置及び寸法は、アイテムが作成されるときに特定されるのが好ましく、そのため、アイテムの例示的な位置は常に知られている。これらの値を保存するために、レイアウトを表すために使用されるグラフの各頂点は、対応するマークの位置にラベル付けされる。更に、規則がアイテムの現在の位置及び寸法と矛盾しない場合は、規則が追加されるだけでも良い。例えば、含まれるマークの現在の位置が最小オフセット規則に従う場合、最小オフセット規則がレイアウトに追加されるだけでも良い。同様に、固定オフセット規則が追加される場合、オフセット値は、規則によって関連付けられたマークの間のオフセットと等しくなければならない。ボックスの中心の横方向位置又は縦方向位置が固定される場合、ボックスの位置及び寸法は既に知られており、横方向の中心又は縦方向の中心は、規則の追加動作が行なわれた時のボックスの中心の実際の位置に

50

基づいて固定される。

【0425】

ユーザは、インターフェース103を介して、レイアウトに追加される規則を必要とするレイアウトへの変更を指示しても良い。ユーザによって示される例示的なレイアウトが追加される規則を満たさない場合、もし可能であれば、レイアウトは自動的に調整される。それによって新たに追加される規則は、レイアウトと一貫性を有する。例えば、ユーザが固定幅のコンテナの幅を変更させる場合、ダイアログボックスを介して、又はボックスの辺をドラッグすることによって、アイテムの位置を修正し、同様に規則を更新する。これは、いくつかの既存の規則の除去、含まれるマークの位置の変更、及び新たな規則の追加等のいくつかのレイアウト動作を必要とする。例えば、マークの位置を変更する場合、アプリケーション121は、まず、マークの位置が変更された場合に侵害されるであろう任意の規則を除去し、マークの位置を更新し、新たなマークの位置と一貫性のある新たな規則を追加する。

10

【0426】

レイアウトを編集及び作成する場合、任意の制約を侵害せずに他のマーク又はレイアウト原点に関連する1つ又は複数のマークを移動する必要があることがある。これは、移動されるマークが多くの制約によって影響される場合に、非常に複雑な動作となる可能性がある。レイアウトエンジン105は、マーク集合を、任意の制約を侵害することなく横又は縦方向に所定の距離だけ移動させ、必要に応じて、元の集合には存在しない他のマークを移動して要求された結果に到達する「プッシュ (push)」と呼ばれる動作を提供することが望ましい。

20

【0427】

17.3 プッシュ動作

このプッシュ動作は、ユーザインターフェース103からの入力に応じてレイアウトエンジン105によって実現され、マーク集合を、任意の制約を侵害することなく、横又は縦方向に所定の距離だけ移動させ、必要に応じて、元の集合には存在しない他のマークを移動して要求された結果に到達する。

【0428】

このプッシュ動作は、全て同じ方位であるマーク集合上で動作し、プッシュグラフと呼ばれるグラフによって説明されても良い。レイアウト内の各マークに割り当てられた位置と共にマーク及び最小オフセット規則の集合によって定義された例示的なレイアウトが存在する場合、対応するプッシュグラフが定義されても良い。プッシュグラフは、有向グラフであり、このグラフにおいては各頂点がレイアウト内のマークと対応し、各辺が、実際のオフセットが最小オフセット規則によって許可される最小値と同等である最小オフセット規則と対応する。

30

【0429】

レイアウトに対するプッシュグラフは、その辺が、レイアウトを形成するマーク以外を直接プッシュできるマークを示す有効グラフである。第1のマークは、第1のマークの座標値が増加することによって第1マークと第2マークとの間の最小オフセット規則を侵害する場合、第2のマークを（正方向に）直接プッシュできる。この動作は、これらのマーク間のオフセットが既に許可されている最低値である場合に実現される。なお、全ての基本モデル規則は、レイアウトを作成又は編集する場合、同等の最小オフセット規則と置換されることとする。

40

【0430】

また、マークを正方向に（即ち、マークの座標値を増加させて）移動する場合、プッシュグラフ内の対応する頂点から到達可能な頂点に対応する任意のマークは、任意の最小オフセット規則を侵害することを回避するために、同一の距離だけ移動されなければならない。

【0431】

またマークを負方向に（即ち、マークの座標値を減少させて）移動する場合、逆プッシュ

50

ュグラフにおいて対応する頂点から到達可能な任意のマークは、任意の最小オフセット規則を侵害することを回避するために、同一の距離及び寸法だけ移動されなければならない。この逆プッシュグラフは、辺の方向が逆であることを除いてプッシュグラフと同一である。

【0432】

有向グラフ内の到達可能な頂点を決定する様々な方法は、上記を実現するために使用されても良い。

【0433】

プッシュ動作は、任意の最小オフセット規則を侵害することなくマークグループを移動することによって機能する。マークグループの許可された移動の範囲を計算するために、
「緩和 (slack)」と呼ばれる量が、各最小オフセット規則に対して定義される。任意の
最小オフセット規則を $\min(m, n, d)$ とすると、規則に関する緩和は、 $\text{offset}(m, n) - d$ によって得られる。この緩和は、規則が侵害される前にマーク m が（正方向に）移動可能な最大距離であり、また、規則が侵害される前にマーク n が負方向に移動可能な最大距離（即ち、座標が減少可能である最大量）である。最小オフセット規則に対する緩和は、規則が例示的なレイアウトを侵害しない場合、常に非負の数である。プッシュグラフの各辺は、緩和「0」を有する規則に対応する。

【0434】

マーク集合の最大許容正移動は、 m が集合内に存在し、 n が集合内に存在しない全ての最小オフセット規則 $\min(m, n, d)$ の緩和値の最小値である。これは、任意の他のマークを
集合内で移動することなく、また任意の最小オフセット規則を侵害することなく、マークが正方向に移動可能な最大距離である。マーク集合が負方向に移動可能な最大距離は、 m が集合内に存在し、 n が集合内に存在しない形式 $\min(n, m, d)$ の全ての最小オフセット規則の緩和値の最小値と同様に定義できる。マーク集合の所定方向の移動を制限する最小オフセット規則が存在しない場合、最大距離は無限となる（即ち、マークが移動できる距離が無制限となる）。

【0435】

プッシュ動作は、移動される現マーク集合と、現集合内のマークをプッシュするべき残りの距離であるプッシュ距離との経過を追跡する。マーク集合及びプッシュ距離は、動作のステップが実行される度に更新される。プッシュ動作は、マークを1つ或はそれ以上の増分で移動させる。各増分での移動距離は、現マーク集合の最大許容移動から決定される。集合内のマークによってプッシュされることができ追加のマークは、各増加の前に集合に追加され、そのため、マークの差分集合は、各増分だけ移動される。

【0436】

図25は、プッシュ動作の第1の実施の形態に含まれる方法2500を示すフローチャートである。このプッシュ動作は、レイアウトに対する原点が保存されるステップ2501で開始される。プッシュ動作が座標系を定義するマークの位置を変更可能であるため、このマーク及びその位置が記憶される。そのため、座標系は、後でステップ2513において再格納されることができる。ステップ2501の後、プッシュ動作はステップ2502に進み、プッシュする残りの距離が0と比較される。プッシュする残りの距離が0である場合、何も実行されず、動作はステップ2513に進み、座標系の原点が再格納される。これは、レイアウト内の全てのマークを適切な値で移動することによって実行され、ステップ2501で保存された原点マークの元の位置を再格納する。ステップ2502でテストされるプッシュ距離が0でない場合、動作はステップ2503に進み、プッシュグラフがレイアウト内のマークの現位置に基づいて更新される。その後、動作はステップ2505に進む。ステップ2505において、移動するマーク集合は、現集合内に存在する任意のマークがプッシュ方向にプッシュ可能な任意の追加マークを追加することによって更新される。例えば、対応する頂点が、現集合内に存在する任意のマークに対応する頂点からプッシュグラフ（又は、移動距離が負の場合は逆プッシュグラフ）内で到達可能である任意のマークは、現マーク集合に追加される。動作はステップ2507に進み、現在の増

10

20

30

40

50

分の距離が計算される。増分の距離は、残りのプッシュ距離より少ない距離であり、現マーク集合の所定方向における最大許容移動距離である。この距離は、プッシュ距離が正の場合、常に正である。移動距離が計算された後、動作は、ステップ2509に進み、現集合内の全てのマークがステップ2507で計算された距離だけ移動される。ステップ2509の後、動作はステップ2511に進み、残りのプッシュ距離がそこから移動する距離を減算することによって計算される。ステップ2511の後、動作は、ステップ2502に戻り、ステップ2511で計算された距離がテストされ、必要に応じて、他の反復を行うかどうか判定される。

【0437】

図25で説明されたプッシュ動作のバージョンは、決して失敗することはないが、原点マークが動作中に移動される場合、動作の挙動は非直観的となる。例えば、マークグループが右にプッシュされる場合、右に移動するマークの代わりに他のマークが左に移動する可能性がある。この挙動は、ステップ2513において原点を再格納することによって起こる。実際、この挙動は有用であることもある。例えば、ボックスの幅を10単位だけ広くする場合、右辺が10単位も移動できない場合であっても、可能であればボックスの右辺を10単位右にプッシュすることにより、ボックスの幅を10単位広くする。これに関連して、右辺が3単位しか右に移動できないが左辺が7単位だけ左に移動できる場合、プッシュ動作は、自動的に左辺を左に移動させる。その副作用により、他のアイテムも移動しても良い。

【0438】

また、原点マークを移動させないプッシュ動作の他のバージョンも有用である。これは図26の方法2600で説明される。このプッシュ動作の実施の形態は、冗長なステップ2501、2513が省略されることを除いて、図25と同様の方法で実施されるが、追加のテスト、即ちステップ2505とステップ2507との間に挿入されたステップ2606を含む。ステップ2606のテストは、現マーク集合が原点マークを含むかどうかを判定する。マーク集合が原点マークを含む場合、マークが原点マークを移動しなければ必要なだけプッシュされないため、動作は終了し、失敗する。

【0439】

図27A、27B、27C、27D、27E及び27Fは、ともにプッシュ動作の機能例を提供する。これらの図は、2つのコンテナ、即ち、最小幅15単位を有する可変幅コンテナ2701と幅20単位を有する固定幅コンテナ2704とで形成されたレイアウトを示す。コンテナ2701は、マークA、Bと関連付けられた左辺及び右辺を有するボックスによって表される。コンテナ2704は、マークC、Dによって表される左辺及び右辺を有するボックスによって表される。また、これらの図は、マークEで表されたページ2707の辺を示す。

【0440】

図27Aは、プッシュ動作が開始する前のレイアウトの初期状態を示す図である。アイテム2701は、最小オフセット規則2702によって表された15単位の最小幅を有する。アイテム2704は、固定オフセット規則2705によって表された20単位の固定幅を有する。2つのアイテム2701、2704は、固定オフセット規則2703によって表される6レイアウト単位の長さのストラットによって接続される。アイテム2704と、最小長さ15単位を有するマークEで表されるページの辺2707との間には、追加の最小オフセット規則2706が存在する。この規則は、アイテム2704がページの辺2707に15単位より近接することを防止する。プッシュ動作を呼び出す前の実際の距離は、20単位である。プッシュ動作のステップは、マークAを15単位右にプッシュすることに関して図示される。また、レイアウトに関するプッシュグラフ2709は、図27Bに示される。グラフ2709は、固定オフセット規則2703によって辺で両方向に接続されるマークB、Cを表す頂点を示す。また、マークB、Cを表す頂点は、固定オフセット規則2705によって両方向に接続される。

【0441】

図27Cは、ループの第1の反復結果を示す図である。第1の反復において、ステップ2503において、プッシュグラフ2709が計算される。ステップ2505において、プッシュグラフから、マークAが他のマークをプッシュせず、マークA自身が移動可能であることが判定される。ステップ2507において、マークAが最小オフセット規則2702を侵害することなく移動可能である最大距離が3単位であることが決定される。これは必要な距離より少ないので、マークAはステップ2509で3単位移動されて、図27Bに示されるレイアウトとなる。ステップ2511で計算される残りの距離は、このとき12単位である。ループの第2の反復において、ステップ2503では、図27Dに示されるプッシュグラフ2711が計算される。このグラフにおいて、規則2702内の緩和がこのとき0であるため、マークAはマークBをプッシュし、その結果、マークAはマークC、Dもプッシュする。

10

【0442】

図27Eは、ループの第2の反復結果を示し、このときマークB、C及びDの全てがマークAによってプッシュされ、そのため4つのマーク全てが同時に移動しなくてはならないこと判定する。ステップ2507において、マークA、B、C及びDが、マークDとマークEによって表されたページ2707の辺との間の第2の最小オフセット規則2706を侵害することなく、5単位の最大距離を移動可能であることが判定される。残りの距離は12単位であるが、マークDは5単位しか移動することができないため、マークA、B、C及びDは、それぞれ5単位ずつ移動する。ステップ2511において、残りの距離は7単位と計算される。ループの第3の反復において、ステップ2503では、図27Fに示されるプッシュグラフ2713が計算される。このグラフにおいて、全てのマークはマークAによってプッシュされる。

20

【0443】

図26に示されるプッシュ動作の第2のバージョンにおいて、テスト2606は、原点マークEが移動するマーク集合に含まれると判定し、動作が停止する。図25に示されるプッシュ動作の第1のバージョンにおいて、ループの第3の反復は、ステップ2509において残りの7単位だけ移動される全てのマークが対象となり、次のステップ2511において、このとき残りの距離は0と計算され、そのため、ステップ2502のテストによって処理がステップ2513に進み、全てのマークがこのケース、即ち7単位だけ左に移動することによって原点マークEの位置が再格納される。この例では、プッシュ動作の両方のバージョンが同一の結果を作成する。またこの例では、マークAは必要な15単位の代わりに、全体で8単位だけプッシュされるだけである。

30

【0444】

17.4 固定中心を有するプッシュ動作

固定中心規則を組み込むレイアウトモデルにおいて、プッシュ動作は、固定中心を認識されるように修正される必要がある。ボックスの中心の横方向の位置が固定される場合、ボックスの左辺及び右辺は、常に逆方向に同量だけ移動しなければならない、このようなマークは互いに「対向される (opposed)」と呼ばれ、「対向 (opposing)」マークと呼ばれる。同様に、ボックスの中心の縦方向位置が固定の場合、ボックスの上辺及び下辺は、常に逆方向に同量だけ移動しなければならない。

40

【0445】

プッシュ動作を固定中心の組み込みに適合させるために、プッシュされるマーク集合は、2つのマーク集合によって置換される。1つのマーク集合、即ち前方集合は、正方向にプッシュされ、他のマーク集合、即ち対向集合は、負方向にプッシュされる。ステップ2505で、マークは次のように、2つのマーク集合それぞれに追加される。所定のマークが2つの集合のうちの1つに含まれるマークに対向する場合、対向マークは、対向マークにプッシュされる任意のマークと同様に他の集合に追加される。

【0446】

図32は、ステップ2505等を使用して固定中心規則が許可された場合、2つのマーク集合を判定する方法3200を詳細に示すフローチャートである。図32において、プ

50

ッシュする初期のマーク集合は垂直マークであり、これらは右にプッシュされることとする。また、座標系は右に増加することとする。同様に、マークは他の方向及び方位にプッシュされる。ステップ3201において、現マーク集合によってプッシュされる全てのマークは、（固定中心規則は無視して）前方集合に追加される。ステップ3201は、前方マーク集合を表す頂点集合から到達可能な全ての頂点を見出すことによって、プッシュグラフを使用して計算されても良い。更に、任意の2つの対向マークが前方集合に含まれる場合、原点マークもその集合に追加される。ステップ3201に続いてステップ3203では、現集合内の任意のマークに対向する全てのマークは、対向マークの集合に追加される。また更に、任意の2つの対向マークが対向集合に含まれる場合、原点マークもその対向集合に追加される。原点マークを両方の集合に追加することにより、ステップ2606のテスト後、プッシュ動作が効果的に終了される。ステップ3203に続いて、ステップ3205では、対向集合内のマークによって左にプッシュされる全てのマークが、対向集合に追加される。これは、マークが負方向でプッシュされる（即ち、これらの座標が低下する）ことによって逆プッシュグラフが使用されることを除いて、ステップ3201と同一の方法で行なわれる。ステップ3205に続いて、ステップ3207では、対向集合内のマークに対向する任意のマークは、前方集合に追加される。ステップ3207は、ステップ3203の逆である。ステップ3207に続き、ステップ3209では、テストが実行されて、ステップ3207で新たなマークが前方集合に追加されるかどうかをチェックする。新たなマークが追加される場合、計算はステップ3201に戻り、ステップ3201、3203、3205及び3207が、2つの集合の両方に追加される新たなマークがなくなるまで繰り返される。

【0447】

ステップ2507において、移動距離は、2つの集合に関して独立して計算され、2つの距離のうち短い方（即ち、大きさが小さい方）が移動距離となる。ステップ2509において、マークは同一の距離だけ、反対方向に移動される。1つの集合内のマークは、正方向に移動され、他の集合のマークは、負方向に移動される。

【0448】

17. 5 形状規則を有するプッシュ動作

形状規則が水平マーク及び垂直マークと関連しているため、プッシュ動作を、形状規則を有する組み合わせレイアウトモデルと適合させることは、潜在的に複雑である。問題は、マークをプッシュする場合、同一方位のマーク間の複雑なインタラクションが形状規則及び他の規則の組み合わせによって行なわれることである。特に、単一の解を有さない循環依存を作成する可能性もある。この問題を回避するために、拘束はレイアウトモデルに挿入され、形状規則が、縦辺に対応する最大1つのマーク及び横辺に対向する最大1つのマークが最小オフセット規則内に実現するボックスに適用されるだけとなる。つまり、形状規則ボックスの対向する辺の各対のうち少なくとも1つは、最小オフセット規則が存在しない状態であってはならない。この拘束は、形状規則によるマーク間のいかなる複雑な関係も防止する。この拘束によって、プッシュアルゴリズムは修正されても良く、それによって、形状規則を含むボックスの任意の辺がステップ2509において移動する場合は常に形状規則が適用される。形状規則は、ボックスの自由辺を移動することによって適用され、自由辺が、任意の最小オフセット規則には含まれない辺となる。

【0449】

17. 6 文書テンプレートを編集するためのプッシュ動作の使用

ユーザが文書テンプレートを編集する間、GUI301は、プッシュ動作を使用して例示的なレイアウトを修正する。図34は、GUI301がユーザにコンテナの固定辺を移動させる例示的な方法3400を示す。ステップ3401で、ユーザは、例えばマウス133やポインタ313で辺を選択してドラッグすることによって辺が移動されるべきであることを示す。ステップ3401に続きステップ3402において、アプリケーション103は、移動される辺に対応するマークの位置を非固定とする。辺の位置は、通常、辺を表すマークと辺の同一の方位を有する原点マークとの間のオフセットを固定する固定オフ

セット規則を追加することによって固定される。辺を非固定とするために、固定オフセット規則は除去され、辺の位置が変更される。ステップ 3 4 0 2 に続きステップ 3 4 0 3 において、余剰の制約はレイアウトエンジン 1 0 5 に追加され、許可された辺の位置を制限する。辺が固定される場合、辺は移動できないため、辺の移動上の任意の追加制約の必要がない。しかし、レイアウトエンジン 1 0 5 が辺の位置を変更するのに使用される場合、通常、辺の移動範囲は制約を受ける。例えば、レイアウトエンジン 1 0 5 は、辺はレイアウトエリアの辺を超えて移動できないと示されても良い。更なる例として、辺がコンテナの左辺の場合、レイアウト矩形の左辺を表すマークと、コンテナの辺を表すマークとの間に非負のオフセット制約が追加されても良い。また、コンテナの最小幅及び最大幅が適用されても良い。文書テンプレートは、レイアウトエンジン 1 0 5 の各コンテナの最小幅及び最大幅、最小高さ及び最大高さを、それぞれ独立して格納するのに使用されても良い。コンテナの左辺及び右辺を表すマーク間に、最小オフセット制約及び最大オフセット制約を追加することによって、ユーザがコンテナの辺をドラッグする間、コンテナの最小幅及び最大幅の設定は、レイアウトエンジン 1 0 5 によって適用可能となる。

【0 4 5 0】

ステップ 3 4 0 3 に続いて、アプリケーション 1 2 1 は、ステップ 3 4 0 5 において辺を移動するためのプッシュ動作を含む。このプッシュ動作は、辺の移動によって任意の制約が侵害されていないかを確認し、レイアウト規則で許可される範囲に移動を制限する。プッシュ動作に続いて、ステップ 3 4 0 7 では、ステップ 3 4 0 3 に追加される制約は除去され、ステップ 3 4 1 1 において、辺の位置が新たな位置に固定される。ステップ 3 4 1 1 の後、アプリケーション 1 2 1 は、レイアウトエンジン 1 0 5 を使用して、以下に説明する図 2 8、2 9、3 0 A、及び 3 0 B に示す通り、ステップ 3 4 1 3 でレイアウトを再計算する。ステップ 3 4 1 3 は、全ての最小距離規則及び好適オフセット規則が適用されることを確認する。ステップ 3 4 1 1 は、ステップ 3 4 1 3 が行なわれる場合に、レイアウトエンジン 1 0 5 が辺を他の場所に移動するのを防止するのに必要である。最後に、ステップ 3 4 1 3 でレイアウトが再計算された後、アプリケーション 1 2 1 は、ディスプレイ 1 4 4 を更新し、辺の位置の変更結果を示して、更新が完了する。図 3 4 のステップは、ユーザがコンテナの辺をマウスでドラッグするとき、ドラッグ中に即時フィードバックを提供することを繰り返しても良い。

【0 4 5 1】

図 3 5 は、文書テンプレートを編集するためにプッシュ動作が使用される可能性のある他の方法 3 5 0 0 を示す。方法 3 5 0 0 はステップ 3 5 0 1 で開始し、ユーザは固定幅コンテナの幅に変更することを要求する。コンテナの位置は可変であり、左辺及び右辺は固定されないこととする。これは、キーボード 1 3 2 を使用して、GUI 3 0 1 によって表示されるプロパティダイアログに新たな幅を入力することによって実施される。1 つの実施の形態において、マウス 1 3 3 の右ボタンを使用してコンテナ上をクリックすることによって、パレット 3 1 1 と同様のコンテキストメニューが表示される。「プロパティ・・・」アイコンがコンテキストメニューから選択されて、プロパティダイアログを表示させても良い。ステップ 3 5 0 1 の後、ステップ 3 5 0 3 において、コンテナの現在の幅と新たな幅との間の差分が計算される。ステップ 3 5 0 3 の後ステップ 3 5 0 5 において、コンテナの幅は固定から非固定に変更される。コンテナの幅は、コンテナの左辺及び右辺を表すマーク間に固定オフセット規則を追加することによって固定される。このように、ステップ 3 5 0 5 は、固定オフセット規則を除去することを含む。ステップ 3 5 0 5 の後ステップ 3 5 0 7 において、コンテナはレイアウトに追加され、コンテナの辺の移動を制限する。各コンテナの最小幅及び最大幅は、文書テンプレート内に存在しても良い。これらの制約は、規則をレイアウトに追加することによって適用される。最小値は、コンテナの左辺及び右辺を表すマークに関する最小オフセット規則を追加することによって適用され、最大値は、コンテナの左辺及び右辺を表すマークに関する最大オフセット規則を追加することによって適用される。

【0 4 5 2】

ステップ3507からステップ3509に進み、プッシュ動作は、幅の差分だけ、コンテナの右辺に対応するマークを右に移動するのに使用される。幅の変更が負である、即ちより狭い幅のユーザの変更要求を表す場合、マークは負の距離だけプッシュされるため、右ではなく左にプッシュされる。次のステップ3511では、残りの必要な幅の変更が計算される。右辺を計算された距離だけプッシュする場合にプッシュ動作がステップ3509において成功すると、コンテナの幅は、正規のサイズに変更完了され、残りの距離は0となるであろう。ステップ3509においてプッシュ動作が成功しない場合、現在の幅は必要な幅に等しくならず、残りの距離は0とならないであろう。ステップ3511に続きステップ3513において、左辺はプッシュ動作を利用してステップ3511で計算されたサイズの残りの変更によって、左にプッシュされる。幅の残りの変更が負である場合、左辺は負の距離分、左にプッシュされるであろう。つまり、辺は右にプッシュされる。

10

【0453】

ステップ3511で計算された距離だけ左辺をプッシュするのがステップ3513で成功した場合、幅は要求された幅と同じになる。成功しない場合は、幅は要求された幅に可能な限り近い値になる。ステップ3513の後、ステップ3507で追加された制約は、ステップ3515で除去される。また幅は、コンテナの左辺及び右辺を表すマークに関する固定オフセット制約を追加することによって、ステップ3517で新たな幅で固定される。ステップ3517に続いて、レイアウトはステップ3519において再計算される。またディスプレイは更新され、ステップ3521で新たに計算されたレイアウトを反映する。その結果、コンテナの幅の変更動作が完了する。レイアウトの計算処理は、以下により詳細に説明される。図35に示される例において、図26で概要を示されたプッシュ動作の第2のバージョンが使用されることとする。図25で概要を示されたバージョンが使用される場合、ステップ3511、3513は不要となるであろう。

20

【0454】

18. レイアウト計算の詳細な説明

18.1 単純なアルゴリズムを使用したレイアウト計算

1つの実施の形態において、許可された規則は、等価オフセット規則、縦横比（アスペクト）規則、最小オフセット規則、及び最大オフセット規則と共に基本モデルの規則である。ここで、基本モデル規則は線形不等化制約と同等であり、等価オフセット規則は線形制約であり、最小オフセット規則及び最大オフセット規則は1次目的関数を定義するのに使用される。このモデルでは、規則は線形プログラムを定義するため、単純アルゴリズム又は線形問題を解決する任意の他の方法を使用してレイアウト計算が実行される。

30

【0455】

この実施の形態において、目的関数は、最小オフセット規則によって関連付けられる一対のマーク間の各オフセットを加算し、この合計から最大オフセット規則によって関連付けられる一対のマーク間の各オフセットを減算することによって計算される。規則が強度を有する場合、オフセットは、加算又は減算の前に、まず対応する規則の強度によって乗算される。

【0456】

残念ながら、レイアウトが単純な方法への重要な修正をすることなく、単純な方法を使用して計算可能なモデルにテキスト規則を含ませる方法は知られていない。

40

【0457】

18.2 修正された単純な方法

他の実施の形態において、許可された規則は、等価オフセット規則、縦横比規則、最小オフセット規則、及び好適オフセット規則と共に基本モデルの規則である。この実施の形態において、最小オフセット規則及び等価オフセット規則は、線形制約に変換され、最小距離規則及び好適オフセット規則は、2次目的関数を定義するのに使用される。このモデルにおいて、制約は1次等式及び1次不等式であり、目的関数は2次関数である。これらの種類の問題は、2次最適化の当業者にとって知られている方法を使用して解決されても良い。単純な方法の修正バージョンは、本モデルでレイアウトを計算するのに使用されて

50

も良い。

【0458】

単純な方法による通り、2次プログラミングへの重要な修正を行わず、2次プログラミングを使用して様々な幅及び高さを有するテキストボックスを許可するモデルを扱うための既知の方法は存在しない。

【0459】

18.3 グラフに基づいたレイアウト計算

他の実施の形態において、使用されるレイアウトモデルは、全ての基本モデル規則、好適オフセット規則、及び形状規則を許可する。このモデルは、固定中心規則が許可されないことを除いて、好適レイアウトモデルと同一である。この実施の形態において、基本規則は、最小オフセット規則を使用して表され、有向グラフとして格納される。好適オフセット規則は、別個の有効グラフとして格納される。形状規則は別に格納される。この方法は、縦横比規則及びテキスト規則の両方を組み込ませるとい、単純な方法及び2次プログラミング方法より優れた利点を有し、他の形状規則も同様に組み込むように修正されることが可能である。

【0460】

更に、本実施の形態では、各コンテナはボックスに関連付けられている。いくつかのコンテナに関しては、更なる別個のボックスがコンテナに関連付けられ、コンテナの理想的なサイズ及び／又は形状を表すのに使用されても良い。最小オフセット規則は、レイアウトエンジン105に対して、理想的な形状を表すボックスのサイズ及び形状に可能な限り近似して、印刷又は表示されたコンテナの形状を表すボックスを作成するように命令するのに使用されても良い。コンテナの理想的なサイズ及び形状は、関連付けられたボックスに形状規則を適用することによって定義される。レイアウト計算方法は、コンテナのサイズ及び形状を、理想的なサイズ及び形状に可能な限り近づけて作成するよう動作する。理想的なサイズ及び形状では、レイアウト内の利用可能なスペースに当てはまらないことがあるため、コンテナを正確に理想のサイズ及び形状に作成することは、常に可能であるとは限らない。例えば、制約が理想的なサイズ又は形状を妨げる場合もある。

【0461】

最小オフセット規則は、マークの取りえる位置を制限する制約を定義するのに使用され、好適オフセット規則は、レイアウト計算方法によって最小化される目的関数を定義するのに使用される。また形状規則は、計算方法を動的に導くのに使用される。目的関数を定義するために、好適オフセット規則はそれぞれエネルギー値に関連付けられる。好適オフセット規則のエネルギーは、好適オフセットと実際のオフセットとの差分の2乗の2分の1である。総エネルギーと呼ばれる目的関数は、全ての好適オフセット規則のエネルギーの合計である。強度が好適オフセットに関連する場合、エネルギーに強度を乗算する。好適オフセット規則は、自然長を備え且つ圧縮又は伸長時に張力を有するバネとみなされても良い。好適オフセット規則の張力は、実際のオフセットと好適オフセットとの差分を乗算された規則の強度である。規則の強度は、バネの強さに類似している。レイアウトエンジン105は、全ての好適オフセット規則の張力を平均化して総エネルギーを最小化することによってレイアウトを計算する。

【0462】

図28は、レイアウト計算に含まれる主なステップを示す図である。レイアウト計算方法2800は、総エネルギーが最小化されるまで、いかなる最小オフセット規則も侵害することなくマークを移動する（即ち、位置を変更する）。方法2800は、ステップ2801で開始し、レイアウトエンジン105が、いかなる基本モデル規則も侵害することなく移動可能で、目的関数の値を減少させるマークのグループを探索する。ステップ2803では、テストが実行されて、探索が成功したかどうかを調べる。成功した場合はステップ2805に進み、さもなければレイアウト計算は完了し、方法2800は停止する。ステップ2805において、マークは適当な方向に適当な距離を移動されて、総エネルギーを減少させる。ステップ2805の後ステップ2801に戻り、移動する他のマークのグ

ループを探索する。移動するマークのグループを選択する方法及びマークの移動距離を計算する方法は、以下により詳細に説明される。図 28 に説明される処理が終了することを確認するため、マーク位置の小さな変更、及び／又はマーク位置の小さな変更による総エネルギーの若干の変更は無視される。1つの実施の形態において、位置の変更は、論理単位の整数に拘束されても良い。

【0463】

図 29 は、任意の水平マークのグループが移動する（ステップ 2903）前に、垂直マークが最初に移動する（ステップ 2901）ことを除いて、図 28 に示されるのと同等の手順 2900 を示す。ステップ 2905 のテストは、ステップ 2803 のテストと同一の目的で行なわれる。本開示の範囲を逸脱せずに総エネルギーを減少させる方法を構成するための多数の可能な方法が存在する。

10

【0464】

図 30A は、ステップ 2901-A の 1つの実施の形態においてステップ 2901 が実施されて良いかどうかをより詳細に示す図である。ステップ 3001 において、垂直マークに関するプッシュグラフは、計算される。これはどのマークが所定のマークによってプッシュされることが可能であるかを判定するのに使用される。いくつかの実施の形態において、このグラフは規則を表すグラフとは別のデータ構造であっても良い。他の実施の形態において、2つのデータ構造は組み合わせても良い。ステップ 3003 において、第 1 の垂直マークは、垂直マーク及び水平最小オフセット規則を表すグラフから第 1 の頂点を検索することによって検索される。マークが探索される可能性のある多数の可能な順序が存在する。1つの実施の形態において、マークは、水平最小オフセット規則を表すグラフに対応する頂点が格納される順序で探索されても良い。ステップ 3003 の後、方法はステップ 3005 に進み、ステップ 3003 で選択されたマークによって右にプッシュされることができるマークが計算される。ステップ 3005 は、いかなる最小オフセット規則も侵害することなく、0 ではない任意の距離だけ右に全て移動できるマークグループを判定する。このグループは、ステップ 3003 で選択されたマークと、マークを選択することによってプッシュ可能となる全てのマーク（即ち、選択されたマークに対応する頂点からプッシュグラフ内で到達可能な頂点に対応する全てのマーク）とを備える。ステップ 3007 は、いかなる最小オフセット規則も侵害することなくステップ 3005 で判定されたマークグループが右に移動可能な最大距離を判定する。ステップ 3007 の計算は、プッシュ動作のステップ 2507 の一部で行なわれる計算と同一である。ステップ 3007 の後、ステップ 3009 は、グループ内のマークが移動して総エネルギーを減少する距離を計算する。これはステップ 3007 で計算された距離より大きくはない。ステップ 3009 は、以下により詳細に説明される。移動距離は、論理単位の整数に拘束される。ステップ 3009 の後、方法はステップ 3011 に進み、マークがステップ 3009 で計算された距離だけ移動される。マークが変更された位置を有する可能性があるため、プッシュグラフは更新されて新たな位置の影響を反映する必要がある可能性がある。ステップ 3009 で計算された距離がステップ 3007 で計算された最大許可距離と同等である場合、プッシュグラフは変更が必要となるであろう。これはステップ 3011 に続くステップ 3013 において実行される。また、右へのマーク移動が総エネルギーを減少させないことも可能である。この場合、ステップ 3009 で計算された距離は 0 となり、マークは実際に、ステップ 3011 において位置が変更されることはないであろう。両ステップ 3011, 3013 は、無効な空命令となり、いくつかの実施の形態においてスキップされるであろう。ステップ 3015, 3017, 3019, 3021 及び 3023 は、移動方向が逆であることを除いて、ステップ 3005, 3007, 3009, 3011 及び 3013 とそれぞれ同等である。これらのステップでは、逆プッシュグラフは、プッシュグラフの代わりに使用される。ソフトウェア技術の当業者は、プッシュグラフとして使用される有向グラフデータ構造を設計することによって、同一のプッシュグラフを逆方向に横移動でき、それによって別個の逆プッシュグラフが必要なくなることを理解するであろう。ステップ 3023 の後、テストが行なわれ、全ての頂点がステップ 3003 において検査され

20

30

40

50

たかどうかが判定される。検査されていない場合はステップ 3 0 0 3 に戻り、次のマークが選択される。全ての頂点が検査された場合はステップ 2 9 0 1 が完了し、計算処理がステップ 2 9 0 3 に進む。

【0 4 6 5】

ステップ 2 9 0 3 は、水平マーク及び垂直最小オフセット規則が垂直マーク及び水平最小オフセット規則に代わると考えられることを除いてステップ 2 9 0 1 と同一である。

【0 4 6 6】

図 3 1 は、ステップ 3 0 0 9 で実行される計算方法の詳細を示す。ステップ 3 1 0 1 では、総エネルギーの偏導関数 (partial derivative) は、現マーク集合の現在の位置からの位置変更に関して計算される。総エネルギーの偏導関数は、張力の合計と同等である。偏導関数は、現集合内のマークと、現集合にないマークとを関連付ける好適オフセット規則の張力の合計と同等である。つまり、総計は、 m が現集合内に存在するマークであり、 n が現集合には存在しないマークである好適 (n, m, d, s) 形態の規則を超えている。なお、好適 (m, n, d, s) 規則は、任意のマーク m 及び n に関して、好適 $(n, m, -d, s)$ 規則と同等である。エネルギーは、張力の合計を可能な限り 0 に近似することによって最小化される。これはステップ 3 1 0 3 でマークを以下の式に示す距離 δ だけ移動することによって達成される。

【0 4 6 7】

$$\delta = -D / S$$

式中、 D は張力の合計であり、 S は現集合に存在するマークを現集合には存在しないマークに関連付ける好適オフセット規則の強度の合計である。ステップ 3 1 0 5 において、テストが行なわれて、現マーク集合に関して考察される移動方向で目的関数が減少するかどうかを判定する。これは δ が正であるかどうかのテストと同等である。マークを現在の方向に移動することによってエネルギーを減少できない場合、移動距離はステップ 3 1 0 7 で 0 に設定される。マークを現在の方向に移動することによってエネルギーを減少できる場合、移動距離はステップ 3 1 0 9 で計算される。移動距離は、 δ よりも少なく、ステップ 3 0 0 7 で計算された値である。マークの移動は、論理単位の整数に拘束されても良く、そのため、ステップ 3 0 0 7 で計算される距離は、論理単位の整数に丸められる。これにより、レイアウト計算の終了を確認する。

【0 4 6 8】

原点マークに関する同一の考察は、上記のプッシュ動作に適用するのと同様に、レイアウト計算に適用する。これに関して、ステップ 3 0 0 5 が、水平座標系の原点を定義する指定された垂直マークが移動されるマーク集合に含まれても良いことを除いて、常に移動可能であるマーク集合を見出す場合には、原点マークはレイアウト計算によって無視されても良い。この場合、ステップ 3 0 0 7 は 0 ではない正距離を常に生成し、ステップ 3 0 1 1 は、原点マークの位置を変更する。この場合、原点マークの位置は、ステップ 3 0 0 1 の前に格納され、原点マークの位置は、レイアウト内の全ての垂直マークを、計算中に原点マークが移動された距離をマイナスした分移動することによって再格納されるであろう。他の実施の形態において、原点マークはステップ 3 0 0 7 において考察される。これに関して、原点マークが移動されるマーク集合に含まれる場合、ステップ 3 0 0 7 で計算される距離は 0 である。この場合、レイアウト計算を行なうときに、原点は保存又は再格納されなくても良い。

【0 4 6 9】

形状規則は、非固定幅及び非固定高さの両方を有するコンテナにのみ使用されれば良い。このようなコンテナに関して、コンテンツを含むコンテナの一部は、2つのボックスによって、レイアウトエンジン 1 0 5 内に表されている。1つのボックスは、コンテナに含まれる表示又は印刷されたテキスト又は画像の境界を表す。このボックスの境界は、レイアウトエンジンによって計算される。他のボックスは、そのボックスの理想的な形状を表す。このボックスは、形状規則によって制御され、一般にユーザに対しては印刷又は表示されない。2つのボックスは、1つの水平マーク及び1つの垂直マークを共有する。例え

ば、2つのボックスの左辺及び上辺の位置は常に同等である。左辺又は右辺が共有されるかどうか、又は上辺又は下辺が共有されるかの差はない。残りの2つの縦方向の辺は、最小距離規則によって関連付けられており、同様に、残りの2つの横方向の辺は、最小距離規則によって関連付けられている。最小距離規則は、2つのボックスが可能な限り近似したサイズで形成されることをレイアウトエンジンに示す。

【0470】

形状規則は、このような関連する一対のボックスのそれぞれの辺に関連する任意のマークを移動する場合に考慮される。共有される形状規則を有するボックスに属するマークは、ステップ3003において無視及びスキップされても良い。しかし、これらの非共有マークは、形状規則を有さない関連ボックスの辺を表す任意のマークが移動される場合、ステップ3009の前に調整される。水平マークに適用される同等のステップが、同様に適用される。非共有マークは、形状規則に関連して調整され、エネルギー機能を最小化する。この処理は、垂直マークと水平マークとの間でエネルギーが平均化されることを確認する。

10

【0471】

形状規則が縦横比規則である場合、エネルギーを最小化することは容易である。ディスプレイボックスの幅及び高さがW及びHであり、理想的なボックスの幅及び高さがw及びhであるとき、最小エネルギーは、 $W+H=w+h$ の場合に到達される。縦横比規則によって定義される高さとの割合がrであるとき、理想的な幅及び高さは以下の通りに計算できる。

20

【0472】

$$w=(1+r)^{-1}(W+H) \quad \text{及び} \quad h=r(1+r)^{-1}(W+H)$$

形状規則がテキスト規則である場合は、理想的なボックスの形状に関する単純な公式は存在しない。テキストレイアウトエンジンは、計算を行なってテキストを矩形スペース内にレイアウトする。テキストレイアウトエンジンは、理想的なボックスの寸法を計算するのに使用される。テキストレイアウトエンジンは計算を行なって、テキストのブロックを所定幅を有するスペースにレイアウトし、テキストのレイアウトされたブロックの高さを迅速に計算する。テキストレイアウトエンジンは、ワードラップ計算を組み込み、テキストの長い行を複数の行にラップする。テキストレイアウトエンジンは、テキストに関するフォーマット情報に従ってテキストをレイアウトする。フォーマット情報は、段落のスタイル、例えば、各文字に対して使用されるボールド又は下線等のフォントのサイズやスタイルを含んでも良い。

30

【0473】

1つの実施の形態において、理想的な幅はテキストの最も広い幅の行に設定され、理想的な高さはディスプレイボックスの幅と同等の幅を有する、テキストレイアウトエンジンによって計算されたレイアウトされたテキストブロックの理想的な高さに設定される。これは、幅が最も長い行の幅よりも狭い場合、又はディスプレイボックスがテキストを含むほど十分に高くない場合、エネルギーペナルティを追加するという効果を有している。

【0474】

他の実施の形態において、テキストレイアウトエンジンは、最も低いエネルギー値となる幅が見出されるまで、各幅に対する高さを計算して異なる幅にテキストをレイアウトするのに使用される。テキストの最も幅広行がディスプレイボックスと同等に幅広くない場合には、理想的な高さはテキストの最も幅広行の幅に設定されても良く、理想的な高さは、行がラップされない場合にレイアウトエンジンによって計算される高さに設定されても良い。ラップされてディスプレイボックスに当てはめられるべきテキストが長い行を含む場合、探索ストラテジー (search strategy) は、最も少ないエネルギーを有するレイアウトとなる幅を見出すために採用される。多数の可能なストラテジーは、本開示の範囲を逸脱せずに理想的な幅を探索するために工夫されても良い。特に2値化探索ストラテジーは採用されても良い。

40

【0475】

50

他の実施の形態において、テキストが、ラップされてディスプレイボックスに当てはめられるべきテキストの長い行を含む場合、理想的なボックスの寸法は、テキストがレイアウトされてディスプレイボックスに当てはめることができるかどうかに基づいて計算される。テキストがディスプレイボックス内に当てはまらない場合、理想的なボックスの幅及び高さは、ディスプレイボックスより大きく形成される。テキストがディスプレイボックス内に当てはまる場合は、理想的なボックスの幅及び高さはディスプレイボックスより小さく形成される。この戦略は、ディスプレイボックスがテキストを含むのに必要なサイズよりも大きい又は小さいレイアウトに対して、ペナルティを追加する効果を有する。本開示の範囲を逸脱することなく、理想的なボックスの正確な寸法を計算する多数の可能な方法が存在する。使用される可能性のある1つの特別な方法は、ディスプレイボックスの現在の幅と同等の幅を有してレイアウトされる場合、テキストの高さをまず計算し、次にレイアウトされたテキストの高さとディスプレイボックスの高さとの差分の2分の1を、ディスプレイボックスの幅及び高さに加算し、理想的なボックスの寸法を得る。

10

【0476】

18.3.1 単純な1次元レイアウトの例

図39は、18.3章及び以下で説明されるレイアウトメカニズムの使用例を示す。3つのボックス3901は、A、B及びCに特定される。各ボックス3901は、マーク3905によって左辺、上辺、右辺、下辺で定義される。ページの左辺及び右辺は、これらマーク間の固定距離を有する2つの垂直マーク(3903, 3904)として定義される。コンテナのいくつかの辺と、ページのエッジとの間の距離を特定する固定オフセット規則3902が存在する。規則の詳細は、以下の通りである。

20

【0477】

- ・ページの幅は50単位に固定される。

【0478】

- ・PA、即ちコンテナAの好ましい幅は22単位である。

【0479】

- ・PB、即ちコンテナBの好ましい幅は16単位である。

【0480】

- ・PC、即ちコンテナCの好ましい幅は13単位である。

【0481】

- ・(図39で「a」と名づけられる)ボックスAの左辺は、ページの左から1単位の位置である。

30

【0482】

- ・(「m」と名づけられる)ボックスAの右辺は、自由に移動する。

【0483】

- ・(「b」と名づけられる)ボックスBの左辺は、左辺「a」の6単位だけ右に位置する。

【0484】

- ・ボックスBの右辺は、右辺「m」の1単位左に位置する。

【0485】

- ・ボックスCの左辺は、右辺「m」の2単位左に位置する。

40

【0486】

- ・(「c」と名づけられる)ボックスCの右辺は、ページの右辺の1単位左に位置する。

【0487】

ボックスA、B及びCの現在の幅をWA、WB、及びWCによってそれぞれ表す。便宜上、ページの左辺において0で開始してページの右方向に増加する座標系が使用される。また、記号「a」、「b」、「c」及び「m」は座標系内の対応するマークの場所を意味する。

【0488】

50

好適オフセット規則のエネルギーは、好適オフセットと実オフセットの差分の2乗の2分の1である。このレイアウトに関する総エネルギーの目的関数は、全ての好適オフセット規則のエネルギーの合計である。そのため、目的関数は、以下の通りである。

【0489】

$$\begin{aligned} E(A, B, C) &= 1/2(WA - PA)^2 + 1/2(WB - PB)^2 + 1/2(WC - PC)^2 \\ &= 1/2(WA - 22)^2 + 1/2(WB - 16)^2 + 1/2(WC - 13)^2 \\ &= 1/2[(WA - 22)^2 + (WB - 16)^2 + (WC - 13)^2] \\ &= 1/2[((m - a) - 22)^2 + ((m - b - 1) - 16)^2 + ((c - m - 2) - 13)^2] \\ &= 1/2[(m - a - 22)^2 + (m - b - 17)^2 + (c - m - 15)^2] \end{aligned}$$

なお、各ボックスの幅を見出すために、そのボックスの最も左のマーク位置は、最も右のマーク位置から減算され、非負の数を得る。これによって、WBは、「m」とボックスBの右辺との間に1単位の間隔が存在するため(m - b - 1)となる。またWCは、「c」が「m」の右にあり、ボックスCの左辺とマーク「m」との間に2単位の間隔が存在するため、(c - m - 2)となる。

【0490】

目的関数を最小化するために、「m」は移動可能な唯一のマークである(実際、「m」及びそれに近い2つのマークは移動しても良く、そのうちの任意のマークは等式中の制御変数として扱われ、そのため「m」と名づけられた1つのマークが単純化のために使用される)。この場合、変数mに関するエネルギー関数の偏導関数を単純に見出し、それによって0である偏導関数を解くことで十分である。

【0491】

$$\begin{aligned} E(A, B, C) &= 1/2[(m - a - 22)^2 + (m - b - 17)^2 + (c - m - 15)^2] \\ \delta E / \delta m &= (m - a - 22) + (m - b - 17) + (m - c - 15) \end{aligned}$$

第3項で負である偏導関数がmに関連するため、この項(m - c + 15)の正負逆の符号について言及する。

【0492】

$$\begin{aligned} \delta E / \delta m &= m - a - 22 + m - b - 17 + m - c - 15 \\ &= 3m - a - b - c - 22 - 17 + 15 \\ &= 3m - a - b - c - 24 \end{aligned}$$

このとき、0であるこの値を解くことによって最小エネルギーを見出すことができる。a, b及びcが、1, 7及び49の値をそれぞれ有することは知られている。

【0493】

$$\begin{aligned} 0 &= 3m - 1 - 7 - 49 - 24 \\ 3m &= 1 + 7 + 49 + 24 \\ &= 81 \\ m &= 27 \end{aligned}$$

1つの変数のみが存在することにより、これは手順を終了させる。そのため、所定の配列の最も低いエネルギーは、マーク「m」がページの左辺から27単位右に位置するときである。

【0494】

18.3.2 単純な2次元レイアウト例

図40Aは、レイアウトメカニズムの他の使用例を示す図である。この場合、移動しても良い水平マーク及び垂直マークが存在するため、この例は2次元の張力を呼び出す。

【0495】

マーク4002によって境界を設定された既知の固定サイズの矩形ページに示されたA, B及びCとラベル付けされた3つのボックス4001が存在する。固定オフセット規則4003, 4004, 4005及び4006は、A及びCのそれぞれ3つの辺の場所を判定する。Aの最も下の横方向の辺は移動しても良いが、固定オフセット規則によって(「m」と名づけられた)Bの上辺と接続される。そのため、これら2つのマークは、同一の距離だけ共に移動されるのみである。同様に、(「n」と名づけられた)Bの右辺及びC

の左辺は、移動しても良いが、両辺の間に固定オフセット規則が存在するため、同一の距離だけ共に移動されるのみである。

【0496】

ボックスBの幅及び高さが増減しても良いため、形状規則はボックスBに適用される。選択される規則は、ボックスBの幅に対する高さの割合が0.5に可能な限り近づけること（これは $RB=0.5$ とラベル付けされた矢印によって示される）を特定する縦横比規則である。ボックスAは22単位の好ましい高さを有し、ボックスCは36単位の好ましい幅を有する。ボックスA及びCは（規則4005, 4006によって）2次元的に自由に変化することはないため、両ボックスとも形状規則を有さない。

【0497】

規則の詳細は、以下の通りである。

【0498】

- ・ページの幅は50単位に固定される。

【0499】

- ・ページの高さは36単位に固定される。

【0500】

- ・PA、即ちコンテナAの好ましい高さは22単位である。

【0501】

- ・RB、即ちコンテナBの好ましい高さとの割合は0.5である。

【0502】

- ・PC、即ちコンテナCの好ましい高さは36単位である。

【0503】

- ・（「a」と名づけられる）ボックスAの上辺は、ページの上から2単位の位置である。

【0504】

- ・ボックスAの左辺は、ページの左辺から1単位右の位置である。

【0505】

- ・ボックスAの右辺は、ページの左辺から18単位右の位置である。

【0506】

- ・（「m」と名づけられる）ボックスBの上辺は、移動しても良い。

【0507】

- ・ボックスAの下辺は、上辺「m」の2単位上の位置である。

【0508】

・（「b」と名づけられる）ボックスBの左辺は、ページの左辺から1単位右の位置である。

【0509】

・（「d」と名づけられる）ボックスBの下辺は、ページの下辺から3単位上の位置である。

【0510】

- ・（「n」と名づけられる）ボックスBの右辺は、移動しても良い。

【0511】

- ・ボックスCの左辺は、「n」の2単位右に位置する。

【0512】

- ・ボックスCの下辺は、ページの下辺から上に3単位である。

【0513】

- ・（「c」と名づけられる）ボックスCの右辺は、ページの右辺から2単位左である。

【0514】

- ・ボックスCの上辺は、ボックスCの下辺から上に16単位である。

【0515】

HA, WA, HB, WB, HC及びWCを、それぞれボックスAの高さ及び幅、ボックスBの

10

20

30

40

50

高さ及び幅、ボックス C の高さ及び幅とする。

【0516】

PHB及びPWBを、それぞれボックス B の好ましい高さ及び幅とする。ボックス B の好ましい高さ及び幅は、レイアウトアルゴリズムへの固定された入力ではなく、これらは、ボックス B に関して幅及び高さに関連する縦横比規則である形状規則を使用することによって、所定の段階において計算される。これらの値は、ボックス B の高さ及び幅の重み付けされた平均として計算される。

【0517】

$$PWB = (WB + HB) / (1 + RB) = 2/3(WB + HB)$$

$$PHB = RB(WB + HB) / (1 + RB) = 1/3(WB + HB)$$

10

この構成のエネルギーを表す目的関数は、水平方向及び垂直方向の張力の 2 乗の合計の 2 分の 1 として得られても良い。

【0518】

$$\begin{aligned} E(A, B, C) &= 1/2(HA - PA)^2 + 1/2(HB - PHB)^2 + 1/2(WB - PWB)^2 + 1/2(WC - PC)^2 \\ &= 1/2[(HA - PA)^2 + (HB - PHB)^2 + (WB - PWB)^2 + (WC - PC)^2] \\ &= 1/2[(m - 2 - a - PA)^2 + (d - m - PHB)^2 + (n - b - PWB)^2 + (c - n - 2 - PC)^2] \end{aligned}$$

(実際にエネルギーの計算を必要とすることなく) このエネルギーを減少させるためには、水平次元及び垂直次元の張力の絶対値の合計がもはや減少されなくなるまで、両次元の目的関数の偏導関数を減少させることで十分である。これは最も低い張力を有する構成を作成する。張力の絶対値は使用され、負及び正の張力が停止状態において、同様に扱われる。偏導関数は、以下の通りである。

20

【0519】

$$\delta E / \delta n = m - 2 - a - PA + m - d + PHB = 2m - 59 + PHB$$

$$\delta E / \delta n = n - c + 2 + PC + n - b - PWB = 2n - 11 + PWB$$

偏導関数をそれぞれ 0 に設定することによって最小化するため、以下の関係が得られる。

【0520】

$$m = 1/2(2 + a + PA + d - PHB) = 1/2(59 - PHB)$$

$$n = 1/2(c - 2 - PC + b + PWB) = 1/2(11 + PWB)$$

30

この例示的な目的のため、ボックス B の初期の幅及び高さをそれぞれ 15 単位及び 10 単位とし、それによって $m = d - 10 = 33 - 10 = 23$ 及び $n = b + 15 = 16$ となる。

【0521】

$$WB = 15$$

$$HB = 10$$

$$m = 23$$

$$n = 16$$

ボックス B の好ましい幅及び高さは、以下の通り計算されても良い。

【0522】

$$PWB = 2/3(WB + HB)$$

$$= 2/3(15 + 10)$$

$$= 50/3$$

$$= 16 \frac{2}{3}$$

$$= 16 \quad (\text{但し、整数単位に丸めた場合})$$

$$PHB = 1/3(WB + HB)$$

$$= 1/3(15 + 10)$$

$$= 25/3$$

$$= 8 \frac{1}{3}$$

$$= 8 \quad (\text{但し、整数単位に丸めた場合})$$

40

50

水平張力及び垂直張力の絶対値の合計は計算され、それによって、レイアウトメカニズムが終了すべき時を判定することができる。この場合、水平張力TXは、ボックスB及びCの現在の幅とこれらの好ましい幅との差分の合計である。同様に、垂直張力TYは、ボックスA及びBの現在の高さとの好ましい高さとの差分の合計である。

【0523】

$$\begin{aligned} TX &= (WC - PC) + (WB - PWB) \\ &= (c - n - 2 - 36) + (n - b - PWB) \\ &= c - 38 - b - PWB \quad (n \text{ を消去した場合}) \\ &= 48 - 38 - 1 - PWB \\ &= 9 - PWB \\ &= 9 - 16 \\ &= -7 \end{aligned}$$

10

$$\begin{aligned} TY &= (HB - PHB) + (HA - PA) \\ &= (d - m - PHB) + (m - 2 - a - 22) \\ &= d - PHB - 2 - a - 22 \quad (m \text{ を消去した場合}) \\ &= 33 - PHB - 2 - 2 - 22 \\ &= 7 - PHB \\ &= 7 - 8 \\ &= -1 \end{aligned}$$

張力が負であるということは、マークが左又は上に移動してこれらの張力を減少することが必要である可能性があることを示す。

20

【0524】

全ての張力は、垂直張力及び水平張力の絶対値の合計である。

【0525】

$$\begin{aligned} T_{TOTAL} &= |TX| + |TY| \\ &= |-7| + |-1| \\ &= 7 + 1 \\ &= 8 \end{aligned}$$

レイアウトメカニズムは、この全ての張力がもはや減少されなくなるまで、水平次元及び垂直次元において、張力を減少させるために進む。

30

【0526】

まず、垂直マークは水平方向に移動される。Bの好ましい幅は、初期段階で以下のように計算される。

【0527】

$$\begin{aligned} PWB &= 2/3(WB + HB) \\ &= 16 \end{aligned}$$

移動されても良い垂直マークのみが「n」であり、固定オフセット規則によって（2単位右に）接続されているマークである。両マークは、水平張力の最小化を試行するような方法で移動される。これにより、他のマーク位置が（n+2）として演繹されるため、「n」が移動されるべき場所は判定されるだけで十分である。関係によって初期段階に説明した通り、「n」の場所は計算され、nに関する目的関数の偏導関数を減少させる。

40

【0528】

$$\begin{aligned} n &= 1/2(11 + PWB) \\ &= 1/2(11 + 16) \\ &= 13 \frac{1}{2} \\ &= 13 \quad (\text{但し、整数単位に丸めた場合}) \end{aligned}$$

マーク「n」は場所13に移動されて水平張力を減少させ、ボックスBの幅も12に変更する。

【0529】

$$WB = 12$$

50

$$HB=10$$

$$m=23$$

$$n=13$$

ボックス B の好ましい幅及び高さは、形状規則によって変更される。

【0530】

$$PWB=2/3(WB+HB)$$

$$=2/3(12+10)$$

$$=44/3$$

$$=14 \frac{2}{3}$$

$$=14 \quad (\text{但し、整数単位に丸めた場合})$$

10

$$PHB=1/3(WB+HB)$$

$$=1/3(12+10)$$

$$=7 \quad (\text{但し、整数単位に丸めた場合})$$

このとき、全ての張力は、再計算されて減少されたかどうかを調べる必要がある。減少されていない場合、レイアウト手順は停止する。新たな張力は、以下の通りである。

【0531】

$$TX=9-PWB$$

$$=9-14$$

$$=-5$$

$$TY=7-PHB$$

$$=7-7$$

$$=0$$

20

$$TTOTAL=|TX|+|TY|$$

$$=|-5|+|0|$$

$$=5$$

全ての張力が 8 から 5 に減少したため、手順は継続する。

【0532】

このとき、水平マークは垂直方向に移動される。この場合、「m」及びそれに接続するマークのみが移動する。接続されるマークが (m-2) に移動するため、「m」が移動されるべき場所を計算することで十分である。前記関係によって決定された通り、目的関数の偏導関数は m に関連して最小化される。

30

【0533】

$$m=1/2(59-PHB)$$

$$=1/2(59-7)$$

$$=26$$

マーク「m」は場所 26 に移動されて垂直張力を減少させる。固定オフセット規則によって接続されるマークは、場所 24 に移動する。

【0534】

$$WB=12$$

$$HB=7$$

$$m=26$$

$$n=13$$

40

このとき、ボックス B の好ましい幅及び高さ及び全ての張力は、再計算される。

【0535】

$$PWB=2/3(WB+HB)$$

$$=2/3(12+7)$$

$$=38/3$$

$$=12 \frac{2}{3}$$

$$=12 \quad (\text{但し、整数単位に丸めた場合})$$

$$PHB=1/3(WB+HB)$$

50

$$\begin{aligned}
 &= 1/3(12+7) \\
 &= 6 \frac{1}{3} \\
 &= 6 \quad (\text{但し、整数単位に丸めた場合}) \\
 TX &= 9 - PWB \\
 &= 9 - 12 \\
 &= -3 \\
 TY &= 7 - PHB \\
 &= 7 - 6 \\
 &= 1 \\
 T_{TOTAL} &= |TX| + |TY| \\
 &= |-3| + |1| \\
 &= 4
 \end{aligned}$$

10

全ての張力が5から4に減少されたため、手順は継続する。垂直マークは、水平方向に再度移動され、張力を減少させるよう試行する。

【0536】

$$\begin{aligned}
 n &= 1/2(11 + PWB) \\
 &= 1/2(11 + 12) \\
 &= 11 \frac{1}{2} \\
 &= 11 \quad (\text{但し、整数単位に丸めた場合})
 \end{aligned}$$

マーク「n」は場所11に移動する。ボックスBの幅は、10単位となる。

20

【0537】

$$\begin{aligned}
 WB &= 10 \\
 HB &= 7 \\
 m &= 26 \\
 n &= 11
 \end{aligned}$$

ボックスBの好ましい幅及び高さ及び全ての張力は、再計算される。

【0538】

$$\begin{aligned}
 PWB &= 2/3(WB + HB) \\
 &= 2/3(10 + 7) \\
 &= 34/3 \\
 &= 11 \frac{1}{3} \\
 &= 11 \quad (\text{但し、整数単位に丸めた場合}) \\
 PHB &= 1/3(WB + HB) \\
 &= 1/3(10 + 7) \\
 &= 5 \frac{2}{3} \\
 &= 5 \quad (\text{但し、整数単位に丸めた場合})
 \end{aligned}$$

30

$$\begin{aligned}
 TX &= 9 - PWB \\
 &= 9 - 11 \\
 &= -2 \\
 TY &= 7 - PHB \\
 &= 7 - 5 \\
 &= 2
 \end{aligned}$$

40

$$\begin{aligned}
 T_{TOTAL} &= |TX| + |TY| \\
 &= |-2| + |2| \\
 &= 4
 \end{aligned}$$

全ての張力が4未満には減少されないため、手順はここで停止する。或は、水平マークも、手順が停止する前に移動されても良い。

この例示的な最終結果は、図40Bに示す通りである。尚、この場合、ボックスA及びCは重畳しても良い。これは望ましい効果である。重畳しない場合、それを回避するため、他の制約を追加することを必要としても良い。例えば、最小オフセット規則を追加して、

50

ボックスCの左辺をボックスAの右辺の右側に保持することが可能であろう。

【0539】

この例は、1つの形状規則及び2つの好適オフセット規則を示すのみであるが、実際のアプリケーションは、各ボックスに対して形状規則を採用しても良い。また、形状規則は、ボックスに表示されるコンテンツによって異なる。例えば、縦横比規則及びテキスト形状規則は、両方とも使用されても良い。各規則は、その形式において理想的なボックス形状を計算する。

【0540】

18.4 固定中心を有するグラフに基づくアルゴリズム

レイアウトエンジン105において、固定中心規則の使用が許可され、そのため、レイアウト計算が修正されて、固定中心規則を正確に操作することが必要となる。ステップ2901（図29）のレイアウト計算に対する修正は、図30Bでステップ2901-Bとして示され、プッシュ動作で固定中心規則を支援するための修正と同様となる。図30Bは図30Aとほぼ同一であるが、レイアウトが計算される場合に固定中心が考察される方法を示す。ステップ3001, 3003, 3005, 3009, 3013, 3015, 3019, 3023, 3025は、図30A, 図30Bと共通である。

【0541】

図30Bにおいて、プッシュされるマークを計算する場合、ステップ3005, 3015は、対向マークを含まなくてはならない。1つの方向に移動する単一のマーク集合を計算するのではなく、2つのマーク集合が計算される。即ち、右に移動する1つの集合及び左に移動する他の集合である。これはプッシュ動作に関して同一である。図30Bにおいて、ステップ3008は、ステップ3007を置換して、最大移動距離が2つのマークセットに関して独立して（固定中央規則は無視して）計算される。次に、2つの値の最小（の大きさ）は、マークが移動しうる最大距離である。ステップ3011はステップ3012によって置換され、ステップ3021はステップ3022によって置換され、その差は両マーク集合が移動されることである。ステップ3012において、ステップ3003で選択されるマークを含む集合は右に移動し、第1の集合内のマークに対向するマークを含む集合は左へ移動する。ステップ3022において、ステップ3003で選択されたマークを含む集合は左へ移動され、第1の集合内のマークに対向するマークを含む集合は、右に移動する。同一の考察が、プッシュ動作の修正に関して適用して固定中心を含む。これに関連して、任意の2つの対向マークが同一の集合内にある場合、原点マークは両方の集合に追加されるべきである。原点マークを移動することは、全ての固定マークが移動され、それによって固定中心を有する全てのボックスも移動されることを要求するため、原点マークが含まれる場合には、論理的には、同一の方位の対向マークの全ての対は両方の集合に含まれるべきである。もちろん、原点マークと対向マークの全ての対とを含む集合は移動されるべきではないため、これらを実際に作成する必要はない。ステップ3008は、原点マークが両方の集合に含まれる場合、最大距離を0に設定する。そのためステップ3009, 3012及び3013は、空命令となり、スキップされる。ステップ3018はステップ3017を置換し、ステップ3008のように、最大距離は、ステップ3003で選択されたマークを含む両方の集合、また、選択されたマークを含む集合内のマークに対向する全てのマークに関して計算される。ステップ3018は、同様に、原点マークが2つの集合のいずれかに含まれる場合、最大距離を0に設定する。

【0542】

19. 文書の印刷

図36は、文書テンプレートから文書を作成及び印刷するステップに含まれる方法3600ステップを示す。方法3600はステップ3601で開始され、アプリケーション121がデータベースに接続し、且つデータベースカーソルを必要なソースデータを備えるテーブルの始めに設定する。ユーザは図14と同様に、ダイアログを介して接続するデータベースを特定しても良い。ステップ3601に続いてステップ3603で、次のレコードがデータベーステーブルからフェッチされ、カーソルが次のレコードを指すように更新

される。文書集合内の各文書が1つのレコードに対応するため、新たな文書が開始され、ページカウンタが文書テンプレートの開始状態にリセットされる。ステップ3605では、アプリケーション121が生成されている文書に関する新たなページを成立させ、ページに関するレイアウトが供給され、ページに適用される。ステップ3607では、現レコードのデータが現ページ上のレイアウト内の各コンテナに関する理想的なサイズを計算するのに使用される。ステップ3607で計算された理想的なサイズは、その後、ステップ3609でレイアウト規則を調整するのに使用される。このような動作は、好適オフセット規則の値の変更とステップ3607で計算された理想的なサイズに基づいた縦横比規則の更新とを含む。更に、テキストの幅も高さも固定されない可変テキストコンテナに関して、高さ及び幅が、ステップ3603でフェッチされたデータベースレコード内のフィールドからのテキストに基づいたテキスト規則と関連して決定される。このようなテキスト規則もステップ3609で更新される。ステップ3609の後、現ページに関するレイアウトがステップ3611で計算される。レイアウトが計算された後、ページは、ステップ3613で描画及び印刷される。ページが印刷された後、テストがステップ3615で実行され、現文書の全てのページの印刷が完了したかどうか判定する。印刷が完了していない場合、方法3600はステップ3605に戻り、次のページの計算を開始する。現文書に関する文書テンプレートからの全てのページの印刷が完了している場合はステップ3617に進み、他のテストが実行されて、現レコードがデータベーステーブルの最後のレコードであるかどうかを判定する。テーブルの全てのレコードが処理された場合、方法3600は完了し、さもなければ、処理はステップ3603に戻り、テーブルの次のレコードがフェッチされることによって次の文書が印刷可能となる。

10

20

【0543】

20. 可能なテキスト形状の予備計算

1つの方法は、レイアウトメカニズムの速度を増加させる。例えば、18章で説明されたグラフに基づいたレイアウト方法は、レイアウトに含まれるであろう任意のテキストの全ての形状を予備計算することになっている。レイアウト計算を実施する前に形状を計算することによって、テキスト形状の計算が既に完了して後続の高速な検索のために結果が格納されているため、レイアウトメカニズムは早く進む。いくつかのテキストの可能な形状を計算する方法は、図41(A)～(K)の例によって図示される。

【0544】

この方法は、以下の方法で機能する。

【0545】

1. 図41(A)のように(各語4102が境界ボックス4103内に示される)、必要な語の間に適切なスペースを設けて、端から端までテキストの語を水平に配置する。この全体の語の設定の幅及び高さは、これら語を囲む最も小さいボックス4101として決定される。境界ボックス4103のサイズは、メモリデータ構造に格納される。これら語は、必要に応じて別のフォントを使用して測定されても良い。尚、小さい間隙は、明確化するため、境界ボックス4103と語ボックス4102との間に図示されるが、この間隙は実際の実施の形態では存在しないであろう。

【0546】

2. 境界ボックスの幅は得られ、使用される座標系の1単位だけ減少される。その後、テキストは、幅制約としての新たな幅を用いて再度レイアウトされる。テキストは、この幅制約とテキスト言語のテキスト流れの規則とに要求されるように後続の行に流れる。高さ制約は存在しないため、必要なだけ多数の行が使用されても良い。図41(B)に示す通り、新たな境界ボックスはその結果から決定される。この新たな境界ボックスも、メモリデータ構造に保持される。語間のスペース4105は、テキスト言語の自然スペーシング規則に従う。例えば英語では、行の端のスペースは、境界ボックスの計算を実行することなくカウントされない。いくつかの言語に関して、継続する語又は語間のスペースの定義は、この例から変化しても良い。

【0547】

30

40

50

3. ステップ2は、より狭い幅の境界ボックスにおいてテキストがレイアウトされなくなるまで繰り返される。図41(C)～図41(K)はこの処理を例示する。なお、例えば語「Brevity」等の各語は、その中にオプションとしてハイフネーション区切り4104を有し、図41(H)～(K)に示す通り、語を区切るのを許可する。英語でこの処理を行なう場合、規則はハイフン4106を追加する。このような動的に挿入される句読点のサイズ及び配置は、通常、テキスト言語の規則に基づいている場合でも、任意の境界ボックスの計算に含まれる。

【0548】

4. 計算されたテキスト形状は、格納された境界ボックスである。これらの形状は、幅又は高さ或はその両方に関する制約が与えられるテキスト部分のサイズを知る必要がある場合、レイアウトメカニズムで使用されても良い。また、これらの形状は、テキストのいくつかの取り得るレイアウト間で迅速に決定するように、所定の次元において、何が次に大きい又は次に小さい形状であるかを知るのに使用されても良い。

【0549】

この説明された実施の形態は、既知の最小サイズの語間のスペースを使って左から右に書かれた言語によって明確に使用されることになっている。別の構成は、これらの規則に従わない言語に関して形成されても良い。例えば、いくつかのアジアの言語では、記号を上から下に縦に書き、これらの列は右から左に書かれる。このような言語に関しては、レイアウト処理は、垂直方位及び水平方位の役割を修正して入れ替える。また、スペース文字のサイズは、より小さい値又は0に減少されても良い。そのため、文字はステップ1で垂直に積み重ねられる。その後、ステップ2で、徐々に幅が広くなる境界ボックスを形成するために繰り返される処理によって、1単位分の高さ（文字の高さであっても良い）が減少されるであろう。同様に、各言語のハイフネーション規則は、テキストの言語に従うであろう。

【0550】

21. 表の作成例

図42A～図42Cは、18章で説明されたグラフに基づくレイアウトメカニズムを使用して表を構成するいくつかの方法を示す。表は、セルに分割され、オプションとして各セルがテキスト又は画像等のいくつかのコンテンツを保持する矩形エリアとして定義される。

【0551】

図42Aは、表を定義する1つの方法を示す。ガイド4203は、行位置及び列位置を定義するのに使用される。コンテナ4201はガイド間に挿入され、コンテンツが挿入される表内でセルとして機能する。ストラット4202は、ガイド4203とコンテナ4201の対応する隣接辺との間に設けられ、隣接セルのコンテンツ間のマージンを定義している。このようなストラットは、セルが互いに同じ高さにされることを意図する場合、長さ0を含む任意の必要な長さを有しても良い。

【0552】

ガイドは、非固定（浮動するガイド）であっても良く、そのため、各セルのコンテンツはガイドをプッシュしてもよく、それによって列サイズ及び行サイズは、異なるサイズのコンテンツを収容するように調整される。或は、ガイドはページに固定されて、行又は列を所定のサイズ又は場所に拘束しても良い。また浮動ガイドは、ストラットをガイド間に挿入することによって、行サイズ及び列サイズを制御するために使用される。これによって（ガイドがページに固定されないため）、表全体がページ上で移動可能となるが、それにも係わらず、行サイズ及び列サイズはこの技術によって制御される。

【0553】

なお、表内のコンテナの外側辺は、ページに固定されるが、これは必須ではない。他のより良い方法は、ガイドを使用して表の最も外側の境界を定義しても良く、これらのガイドとコンテナの外側辺との間にストラットを挿入して、これらの辺を同時に移動しても良い。

【0554】

図42Bは、表を定義する他の方法を示す。この例ではガイドは使用されない。むしろ、ストラット4205は、隣接するコンテナ4204の辺間に直接接合され、行及び列のマージンサイズを特定する。このようなマージンを特定するストラット4205は、最上位のコンテナと左端のコンテナとの間でのみ必要とされる。表内の他のコンテナは、長さ0のストラット4206を使ってこれらの制約にリンクされる。

【0555】

図42Bに示される例において、上列のコンテナの下辺は、これら下辺が互いに正確に配列されるように、高さ0のストラット4206によってリンクされる。同様に、下列のコンテナの上辺は、これらが配置されるように、高さ0のストラットによってリンクされる。同時に、これは、1つの垂直ストラット4205のみが、2つの列のコンテナ間のマージンサイズを定義し且つ制御するために必要とされることを意味する。特に、表のマージンを変更することが、多数のストラットを図42Aの例のように変更することが要求されるよりも、1つのストラットを好ましい又は実際の距離に変更することだけ含むことを意味しているので、図42Bの例は、非常に有用である。同様の論理は、左端の行の右辺をリンクし、更に右端の行の左辺をリンクする、幅0のストラット4206に関して保持される。

【0556】

図42Aに示すように図42Bの例も、ページに固定された表の外側辺を有する。表の場所及びサイズをこのように固定する代わりに、内側辺が配列されるように、長さ0のストラットを使用してコンテナの全ての外側辺を配列することができる。これによって表は、多数の固定外側辺の場所を1つ1つ変更する必要なく、ページ上の必要な全ての場所へ移動できる。このアプローチは、18.3章に記載のレイアウト方法を使用して生成されても良い。

【0557】

第3の選択肢は、ガイドを使用して、表の外側辺と、表の内側辺を配列する長さ0のストラットとを定義しても良い。同様な結果に到達するガイド及びストラットの他の存続可能な組み合わせが多数存在する。

【0558】

なお、表の内側辺を配列するいくつかの方法は、従来の表の概念を形成するのに必要である。図42Cは、ガイドもストラットも内部辺を配列するのに使用されない場合、何が起こるかを示す。コンテナ4207は、表セルとして再度使用され、ストラット4208は、隣接コンテナ間の距離を定義するのに使用される。しかし、この例は、行及び列の内側辺が離間していないため、斜め方向に対向するコンテナを重畳させない状態にする方法が欠如している。

【0559】

これにより、コンテナ及びコンテンツの興味深い存続可能な構成が実現され、18.3章及びそれ以降で説明したレイアウト方法が（レイアウト方法の全ての従来例が可能であるとは限らない）このような構成を生成可能となる。しかし、従来の表の概念に関して、図42A及び図42Bに示される例は、行及び列を正しく配列するのに、より適している。

【0560】

上記の構成は、コンピュータ及びデータ処理産業に適用され、特に、同一のフォーマット及びレイアウトを本質的に有する多数の文書が可変データで再生成される状況に適用される。この例は、全員が異なる名前、住所、その他の個人的詳細情報を有する多数の人物に関する対応表の作成を含む。他の例は、テキスト及び画像を含む固有のコンテンツによって特徴付けられる異なる休日の行先に関する広告パンフレットの作成のためであるが、これによって、全てのパンフレットは、広告を出す組織の商標又は外観を示す共通レイアウトに従って作成される。多数の他の例も、適用される。

【0561】

10

20

30

40

50

前述の事項は、本発明のいくつかの実施の形態を説明するものであり、本発明の範囲から逸脱することなく変形及び／又は変更が可能であり、この実施の形態は実証するものであって制限するものではない。

【図面の簡単な説明】

【0562】

【図1A】本実施の形態に係る可変データ印刷のためのコンピュータシステム構成を説明する図である。

【図1B】図1Aのコンピュータモジュールの概略ブロック図である。

【図2】本実施の形態に係る可変データ印刷のための別のコンピュータシステム構成を説明する図である。

10

【図3】メニューバー、ツールバー、作業エリア及び浮動パレットを含むアプリケーションメインウィンドウの一例を示す図である。

【図4】本実施の形態に係るコンテナ作成、コンテナ間の第1の種類のストラット、アンカー及びスライダを例示する態様を説明するための画面、ツール及びアイコンを例示する図である。

【図5】本実施の形態に係る第1のコンテナ規則の例を示す図である。

【図6A】、

【図6B】、

【図6C】本実施の形態に係る第3のコンテナ規則を示す図である。

20

【図6D】、

【図6E】コンテナ範囲の生成を説明するフローチャートである。

【図7A】、

【図7B】3つの列を有するテキストコンテナと、指示デバイスを使用する直接操作により列幅及びガター幅の双方のサイズを再設定する方法とを説明する図である。

【図8】文字サイズ同期の動作を示すための画面、ツール及びアイコン例を示す図である。

【図9】内側マージンの自動適用の動作を示すための画面、ツール及びアイコン例を示す図である。

【図10】距離制約を説明するために、2つのコンテナの辺の間にストラットを追加する方法を示す図である。

30

【図11】第2の例のコンテナの間のストラットを含むユーザインターフェースのいくつかの特徴動作を示すための画面、ツール及びアイコンを示す図である。

【図12】非固定のガイドの動作を説明する画面例を示す図である。

【図13】、

【図14】データソースの選択方法を示す図である。

【図15】データソースのフィルタリングのためのユーザインターフェースを示す図である。

【図16】、

【図17】データソースを通してナビゲートするための方法を示すユーザインターフェース例を示す図である。

40

【図18】データソースの変数及びデータソースを通してナビゲートするための方法を示すユーザインターフェースの一例を示す図である。

【図19】データソースの変数と、可変文書テンプレートとを関連付けるためのユーザインターフェース例を示す図である。

【図20】可変文書テンプレートにマージされたデータソースコンテンツのライブブルーフィングのためのユーザインターフェース例を示す図である。

【図21A】、

【図21B】選択的ブルーフィングで使用するための、文書ごとにコンテナの幅及び高さを用いた、平均文書と最も異なる文書とを計算する方法を説明するフローチャートである。

50

【図 2 2】レイアウトアイテムと、レイアウト方法で入力として使用される制約とを示すレイアウト例を示す図である。

【図 2 3】図 2 2 の例に対応する縦方向の制約を説明する図である。

【図 2 4】図 2 3 の縦方向の制約を表す有向グラフを示す図である。

【図 2 5】プッシュ動作を実行する処理を説明するフローチャートである。

【図 2 6】プッシュ動作の別の実施形態に関連する処理を説明するフローチャートである。

。

【図 2 7 A】、

【図 2 7 B】プッシュ動作の一例の開始位置を示す図である。

【図 2 7 C】、

【図 2 7 D】プッシュ動作の一例の中間段階を示す図である。

【図 2 7 E】、

【図 2 7 F】プッシュ動作の一例の結果を示す図である。

【図 2 8】グラフ利用レイアウト計算におけるハイレベルステップを示すフローチャートである。

【図 2 9】グラフ利用レイアウト計算の 1 つの実施の形態におけるハイレベルステップを示すフローチャートである。

【図 3 0 A】グラフ利用レイアウト計算が、垂直マークを移動させることによりエネルギー関数の値をいかにして減少させるかを説明するフローチャートである。

【図 3 0 B】固定中心規則の処理を含むグラフ利用レイアウト方法の別の実施形態を示すフローチャートである。

【図 3 1】グラフ利用レイアウト計算において、マークのグループを移動させるべき距離がいかにして計算されるかを示すフローチャートである。

【図 3 2】マークの集合が右側に押されるときに、押される順方向マークの集合及び逆方向マークの集合を計算する処理を示すフローチャートである。

【図 3 3 A】、

【図 3 3 B】、

【図 3 3 C】基本モデルの様々な異なる表現を使用して、コンテナの最小高さ及び最大高さがどのようにして指定されるかを説明する図である。

【図 3 4】レイアウト及び及び他の辺との相互作用によるコンテナの辺位置の修正を示すフローチャートである。

【図 3 5】文書テンプレートを編集するためのプッシュ動作の別の例を示すフローチャートである。

【図 3 6】可変データ文書を生成、印刷する方法を示すフローチャートである。

【図 3 7 A】、

【図 3 7 B】、

【図 3 7 C】、

【図 3 7 D】図 4 に対応するレイアウト例及びレイアウトの様々な制約を格納する方式を示す図である。

【図 3 8】第 2 のコンテナ規則でコンテナ制約を表現するための高さバー及び幅バーの使用を示す図である。

【図 3 9】1 つの次元で動作するレイアウトエンジンの使用例を示す図である。

【図 4 0 A】、

【図 4 0 B】2 つの次元で動作するレイアウトエンジンの使用例を示す図である。

【図 4 1】コンテナ内におけるテキストのレイアウト、及びそれがコンテナの形状からどのような影響を受けるかを示す図である。

【図 4 2 A】、

【図 4 2 B】、

【図 4 2 C】表を構築する方法を説明する図である。

10

20

30

40

【図 1 A】

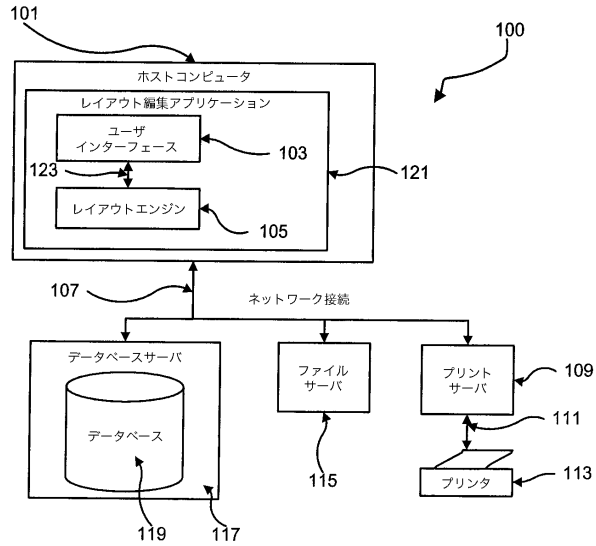


Fig. 1A

【図 1 B】

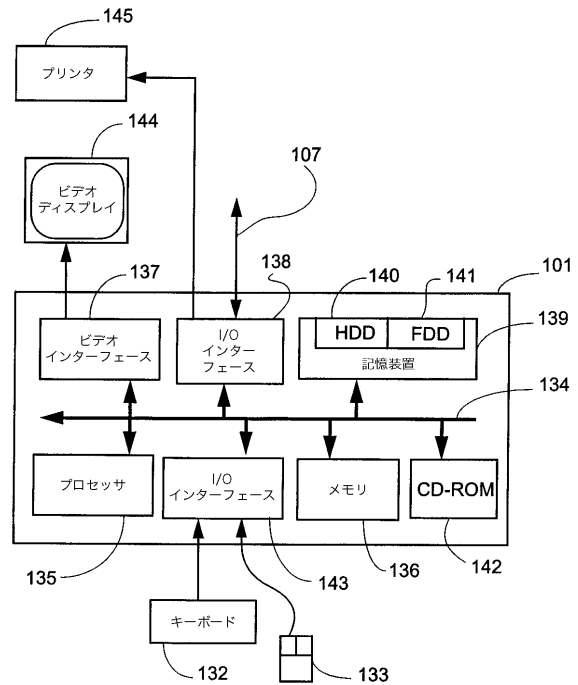


Fig. 1B

【図 2】

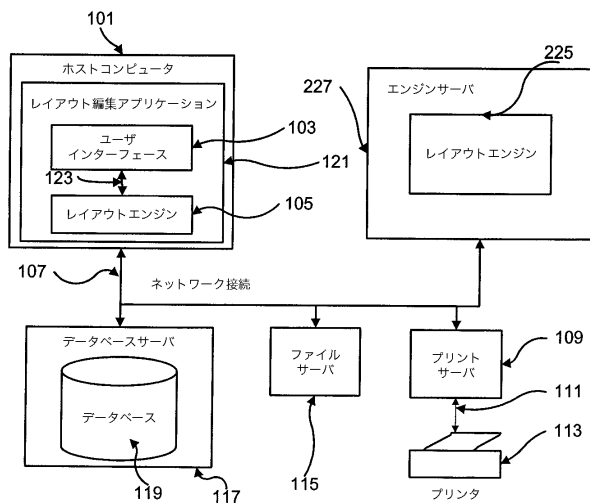


Fig. 2

【図 3】

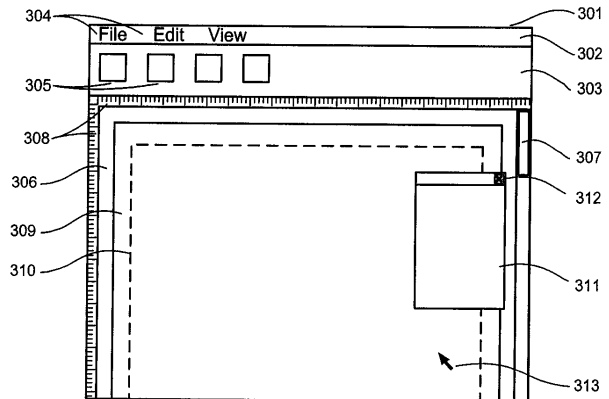


Fig. 3

【図 4】

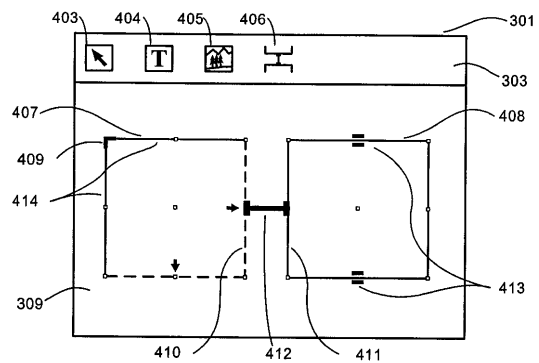
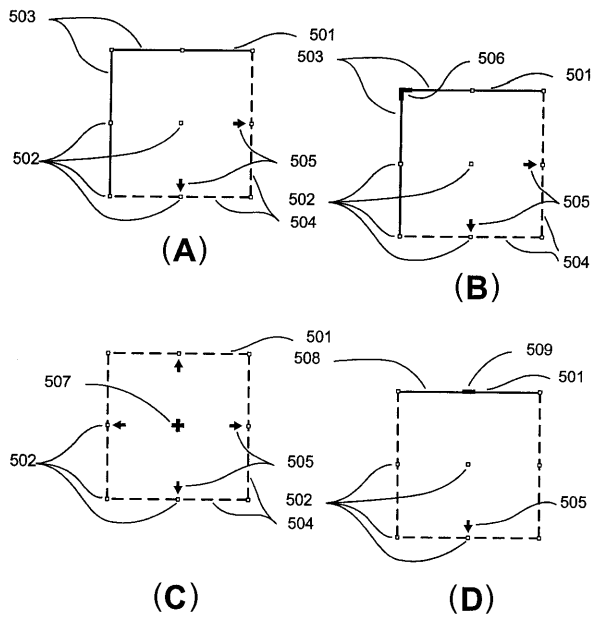


Fig. 4

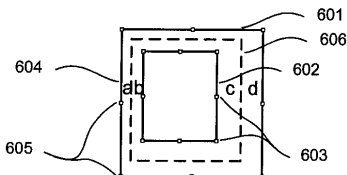
【図 5】

Fig. 5



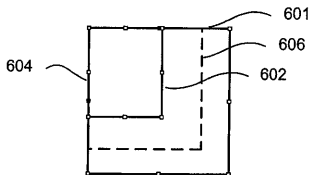
【図 6 A】

Fig. 6A



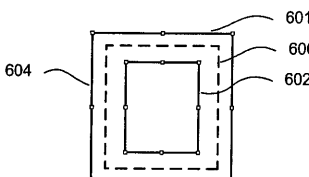
【図 6 B】

Fig. 6B



【図 6 C】

Fig. 6C



【図 6 D】

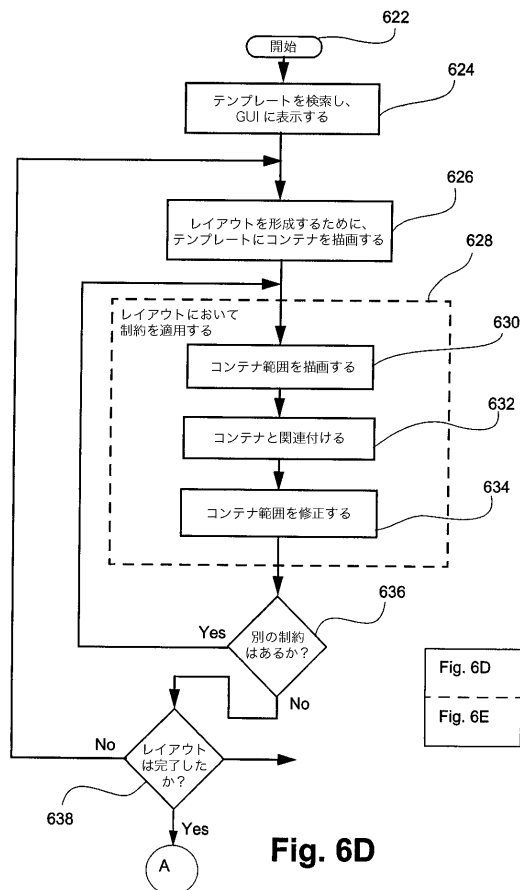


Fig. 6D

【図 6 E】

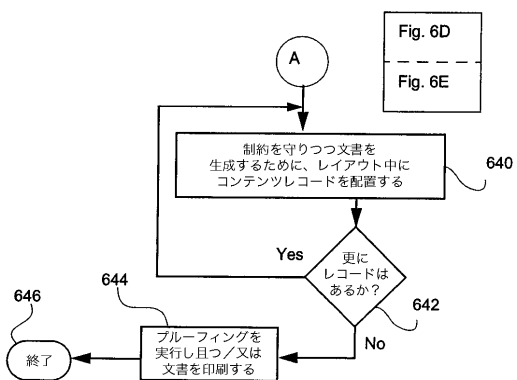
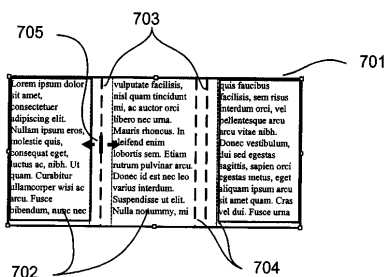


Fig. 6E

【図 7 A】

Fig. 7A



【図 7 B】

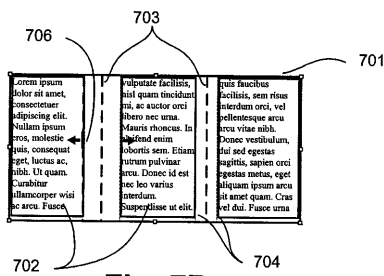


Fig. 7B

【図 8】

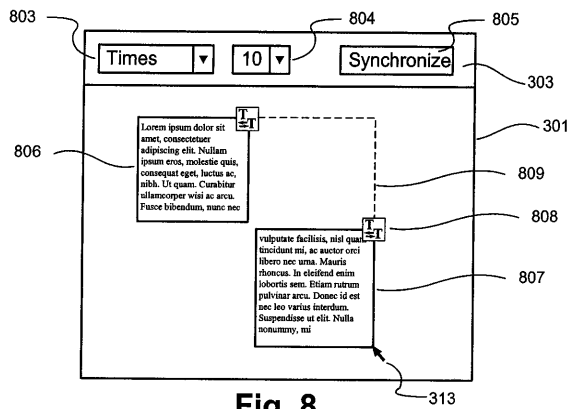


Fig. 8

【図 9】

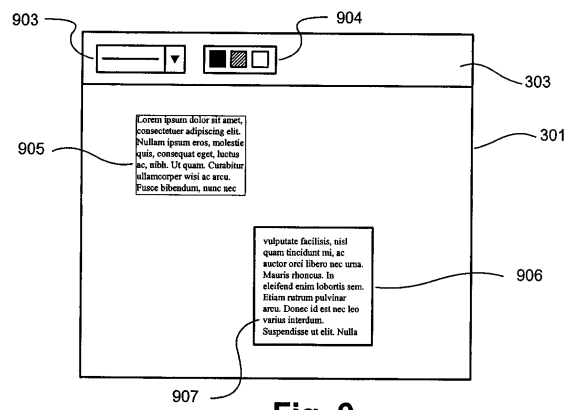


Fig. 9

【図 10】

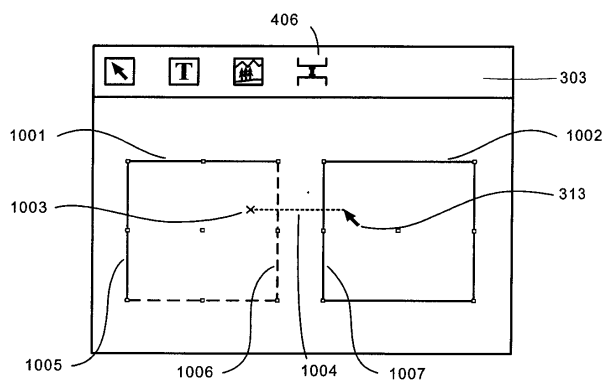


Fig. 10

【図 11】

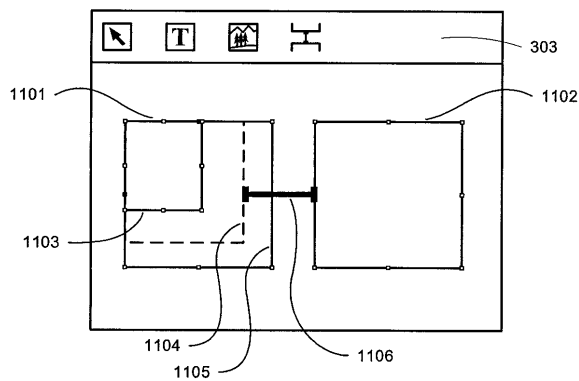


Fig. 11

【図 12】

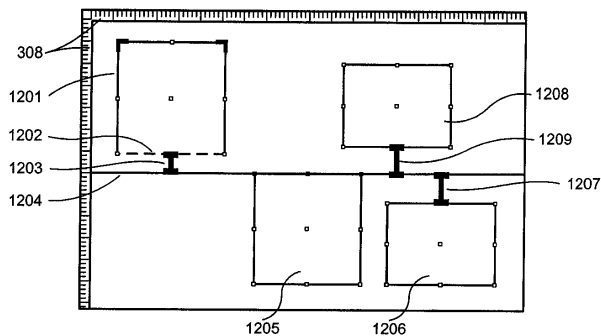


Fig. 12

【図 13】

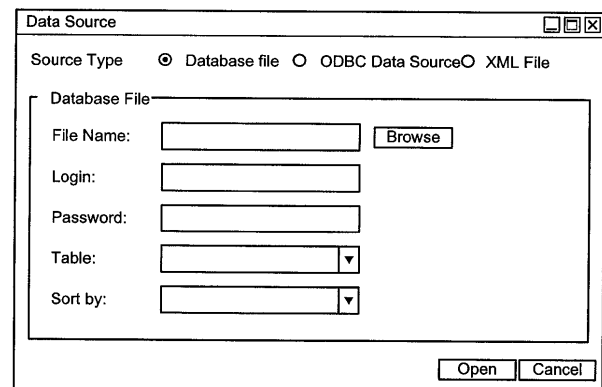


Fig. 13

【図 14】

The dialog box is titled "Data Source". It has three radio buttons for "Source Type": "Database file", "ODBC Data Source", and "XML File". The "XML File" option is selected. Below this, there is a section labeled "Database File" containing the following fields:

- File Name: C:\data\address.xml (with a "Browse" button next to it)
- Login: (empty text box)
- Password: (empty text box)
- Table: address (dropdown menu)
- Sort by: [none] (dropdown menu)

At the bottom right, there are "Open" and "Cancel" buttons.

Fig. 14

【図 15】

The dialog box is titled "Data Source". It has three radio buttons for "Source Type": "Database file", "ODBC Data Source", and "XML File". The "XML File" option is selected. Below this, there is a section labeled "Database File" containing the following fields:

- File Name: C:\data\addresses.xml (with a "Browse" button next to it)
- Login: (empty text box)
- Password: (empty text box)
- Table: addresses (dropdown menu)
- Sort by: [none] (dropdown menu)

A dropdown menu is open for the "Sort by" field, showing the following options: name, address, city, state, and zip. At the bottom right, there are "Open" and "Cancel" buttons.

Fig. 15

【図 16】

The dialog box is titled "Variable Data Library". It has a "Connected to:" label followed by a text box and a "Browse" button. Below this, there are three buttons: "Add", "Edit", and "Delete". At the bottom, there is a status bar with navigation icons and the text "Record 0/0".

Fig. 16

【図 17】

The image shows a menu from a software application. The menu bar includes "File", "Edit", "Database", and "Window Help". The "Database" menu is open, showing the following options:

- Variable Data Library... CTRL+L
- First Record
- Previous Record
- Next Record
- Last Record
- Go to Record
- Filter ...
- Sort ...

Fig. 17

【図 18】

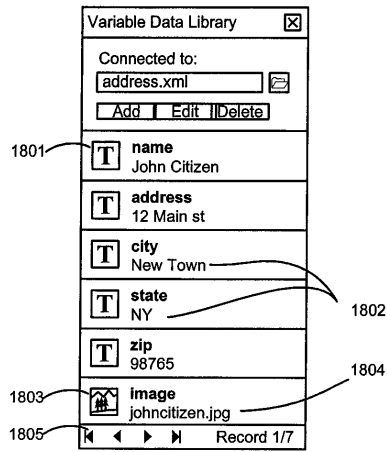


Fig. 18

【図 19】

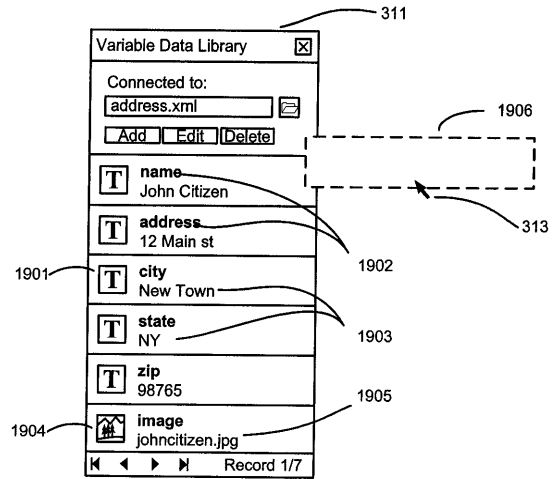


Fig. 19

【図 20】

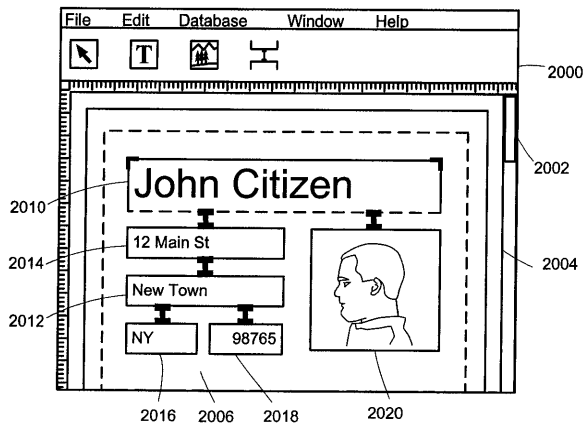


Fig. 20

【図 21 A】

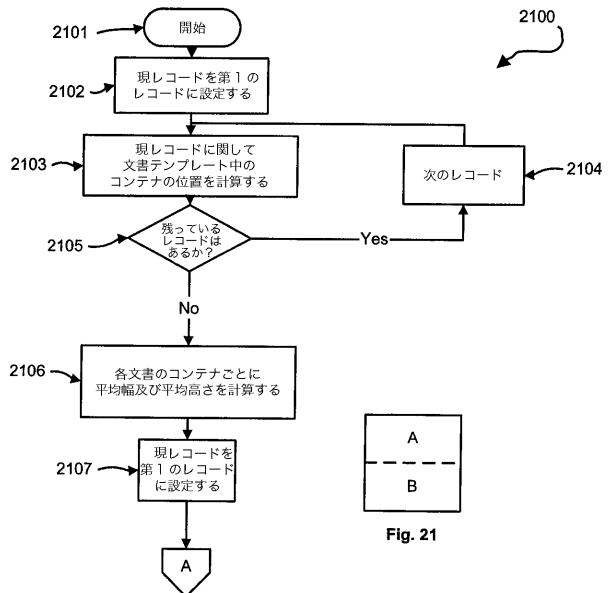


Fig. 21A

【図 2 1 B】

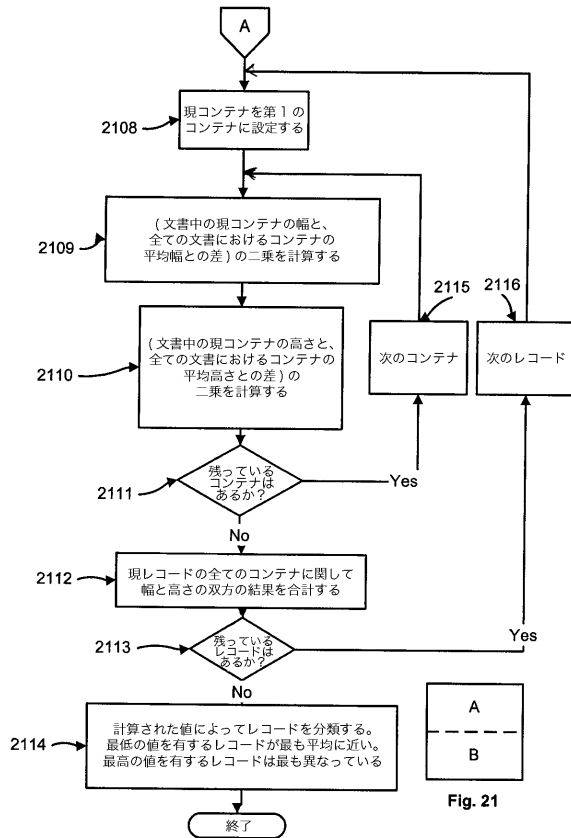


Fig. 21B

【図 2 2】

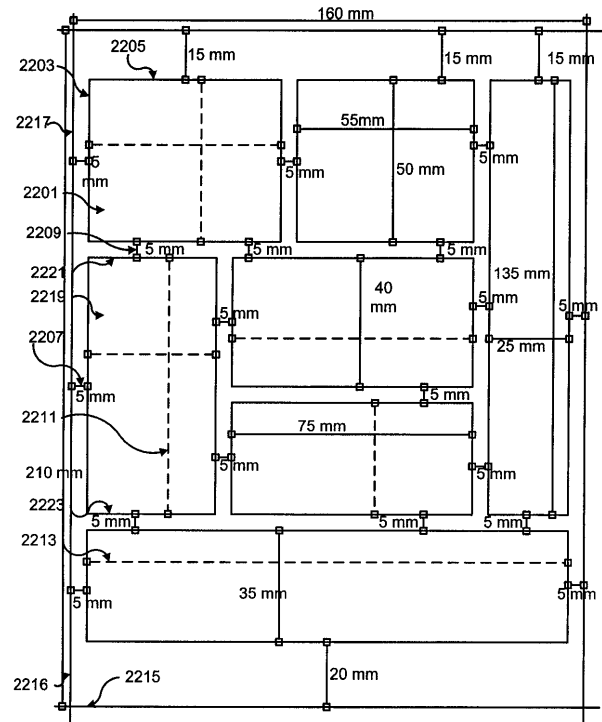
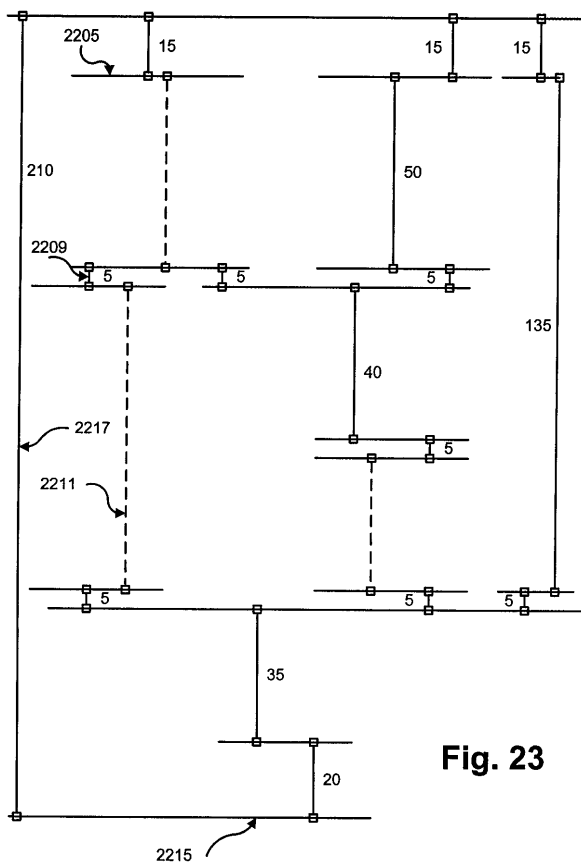


Fig. 22

【図 2 3】



【図 2 4】

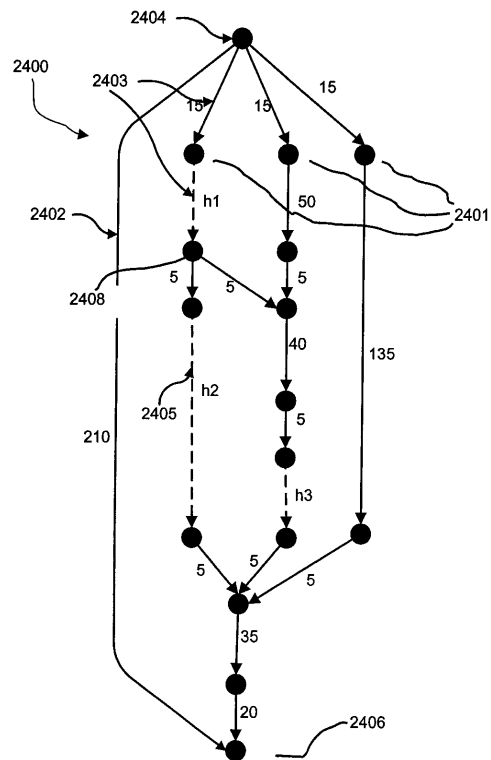


Fig. 24

【図 25】

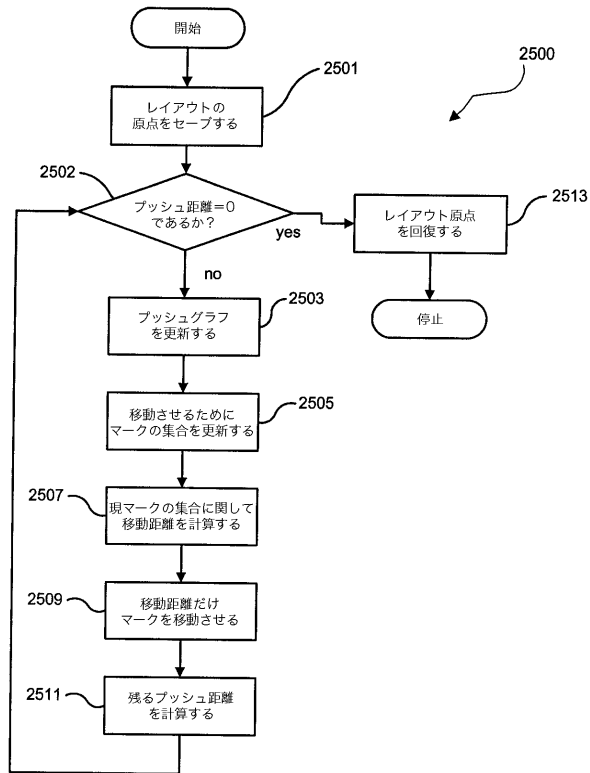


Fig. 25

【図 26】

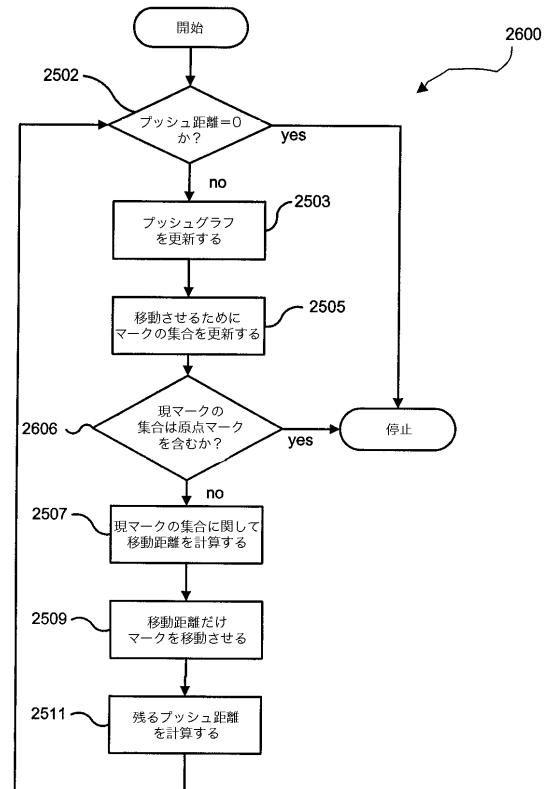


Fig. 26

【図 27 A】

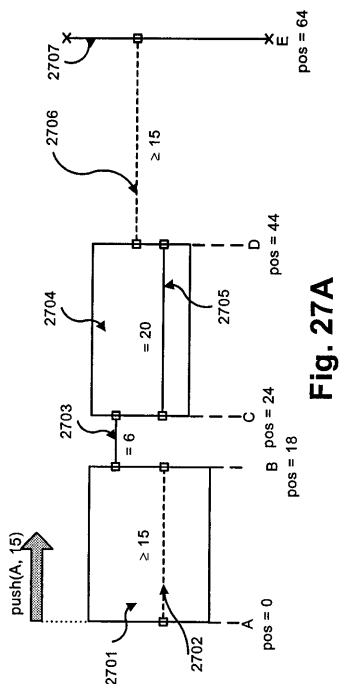


Fig. 27A

【図 27 B】

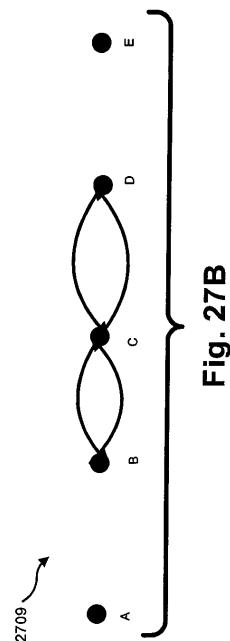
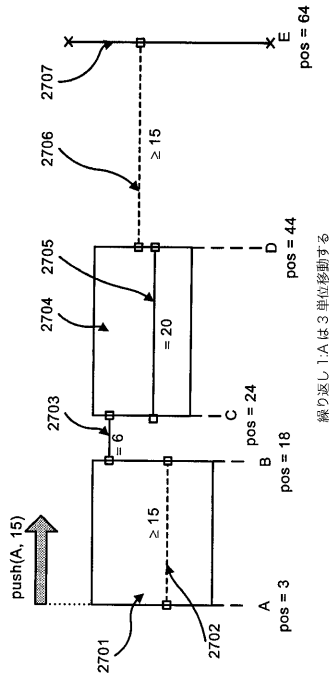
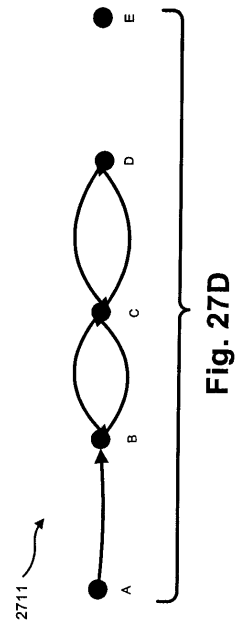


Fig. 27B

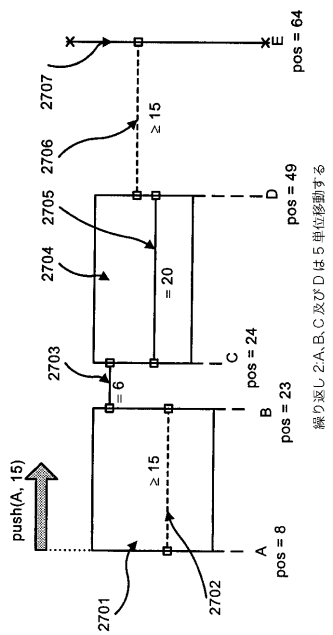
【図 27C】



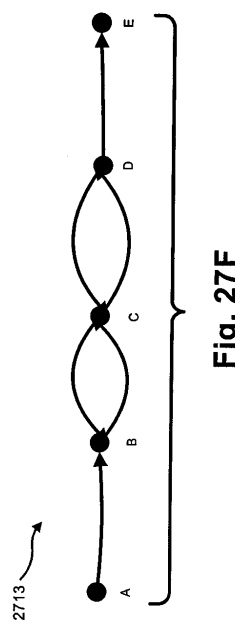
【図 27D】



【図 27E】



【図 27F】



【図 28】

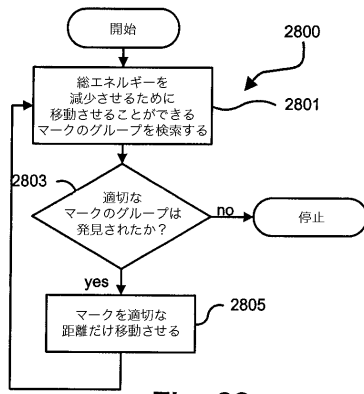


Fig. 28

【図 29】

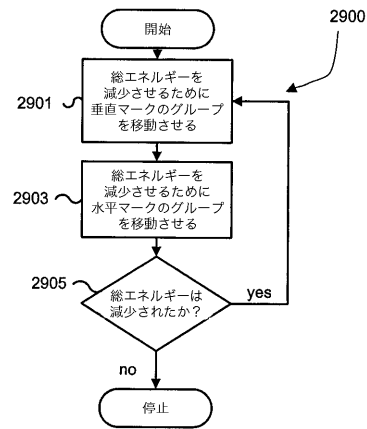


Fig. 29

【図 30 A】

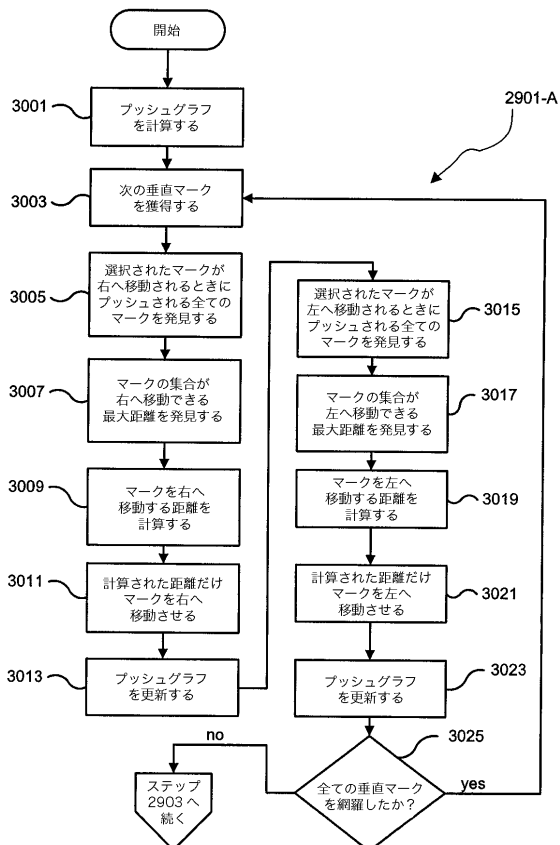


Fig. 30A

【図 30 B】

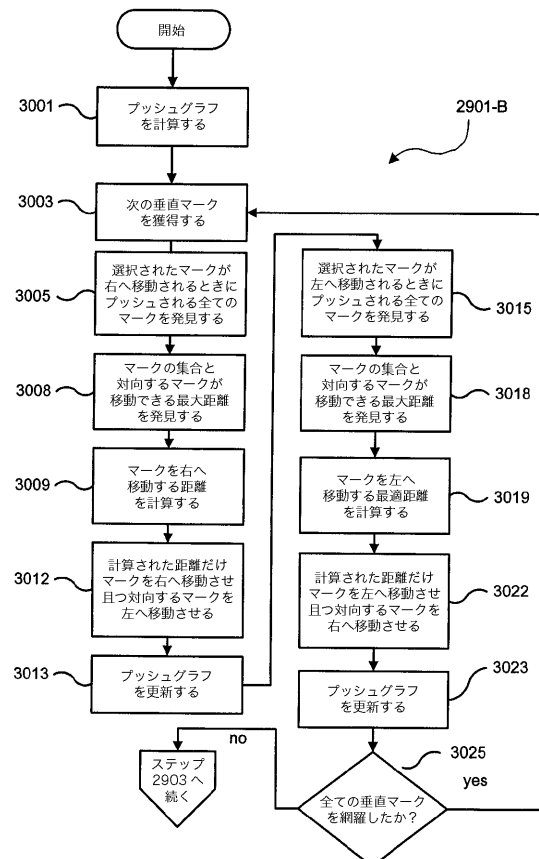


Fig. 30B

【図 3 1】

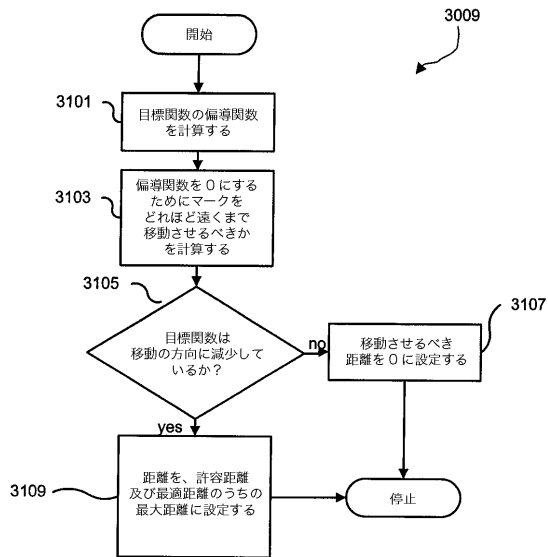


Fig. 31

【図 3 2】

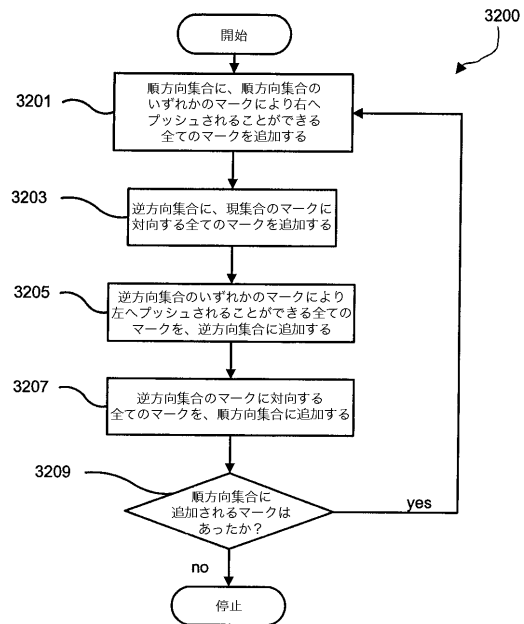


Fig. 32

【図 3 3 A】

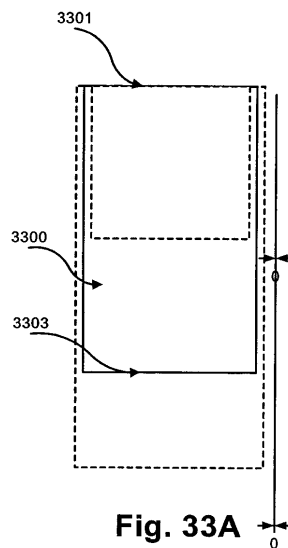


Fig. 33A

【図 3 3 B】

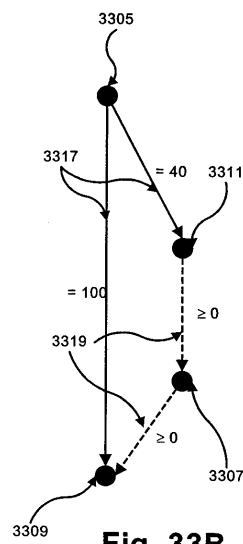


Fig. 33B

【図 3 3 C】

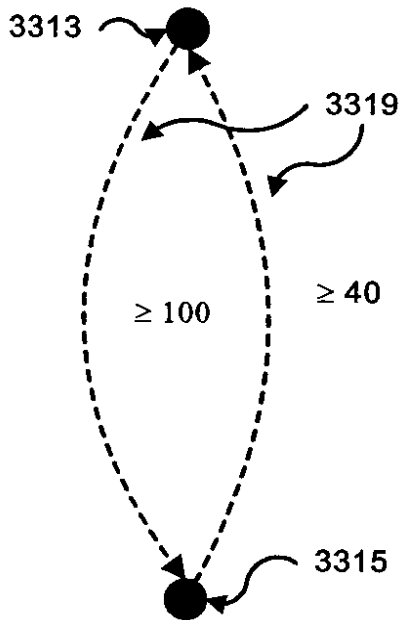


Fig. 33C

【図 3 4】

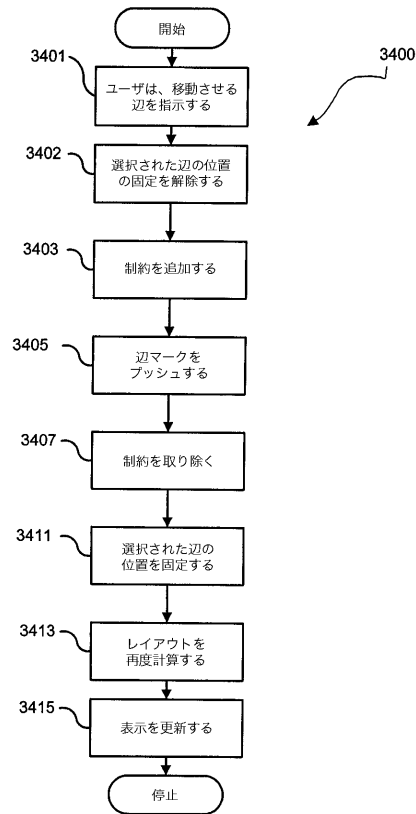


Fig. 34

【図 3 5】

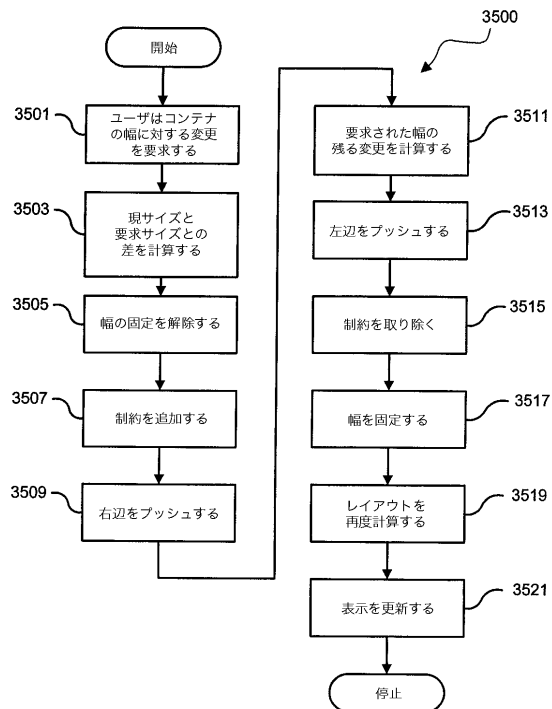


Fig. 35

【図 3 6】

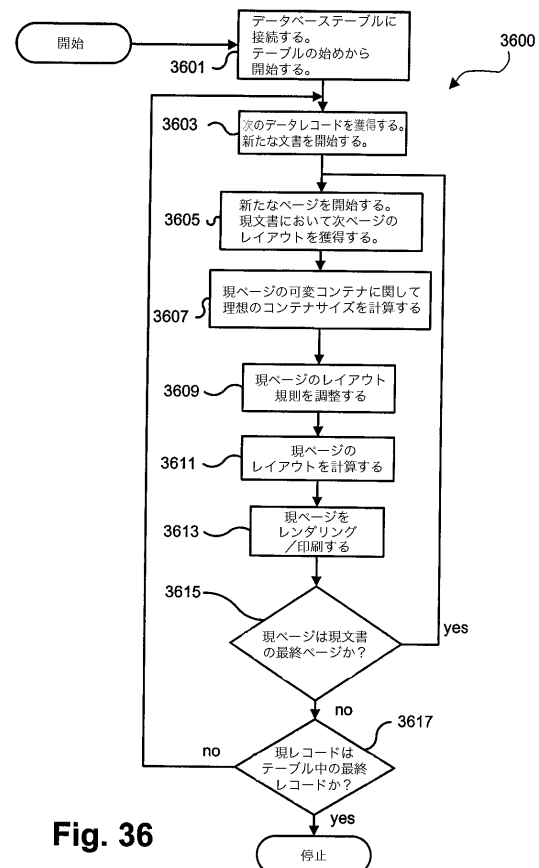
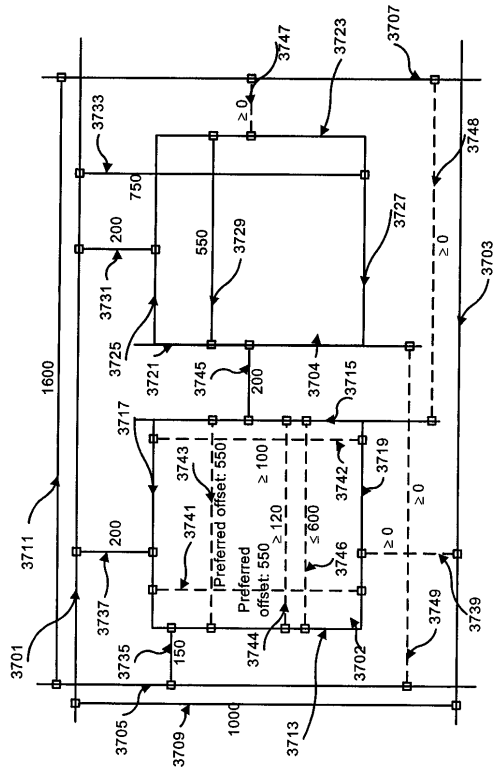


Fig. 36

【図 37 A】



【図 37 B】

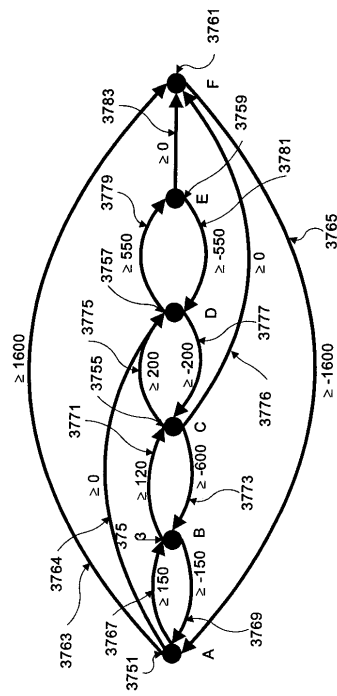


Fig. 37A

Fig. 37B

【図 37 C】

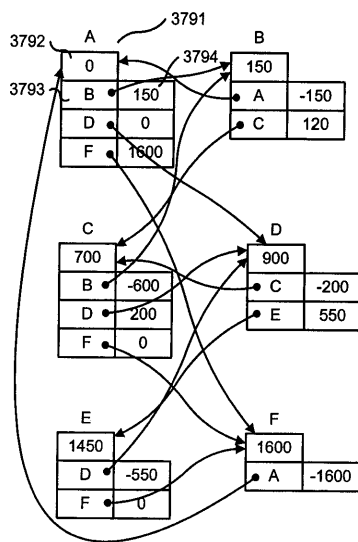


Fig. 37C

【図 37 D】

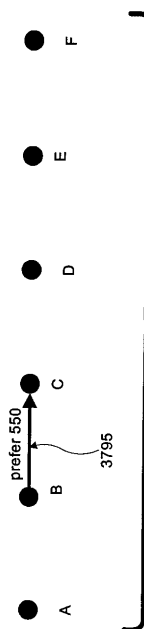


Fig. 37D

【图 3 8】

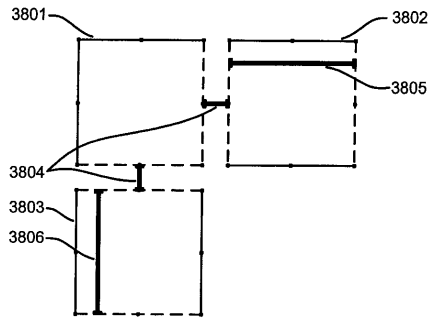


Fig. 38

【図 3 9】

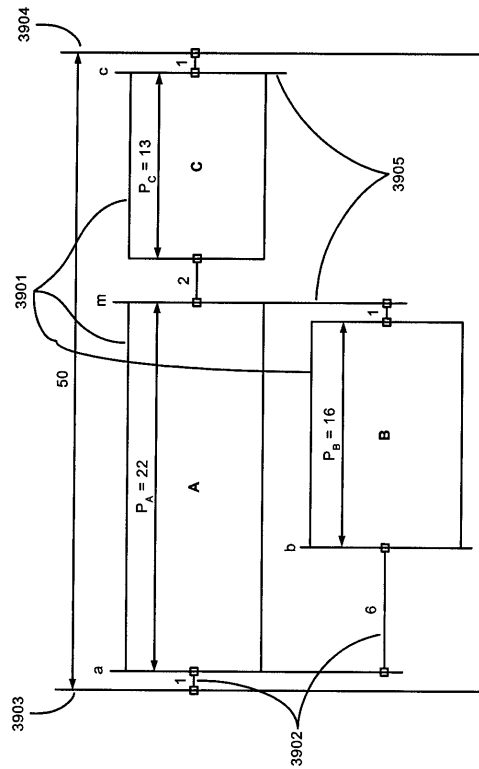


Fig. 39

【図 4 0 A】

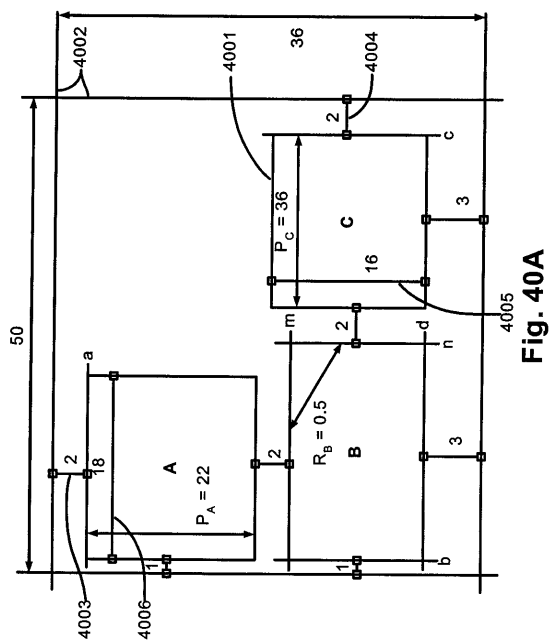


Fig. 40A

【図 4 0 B】

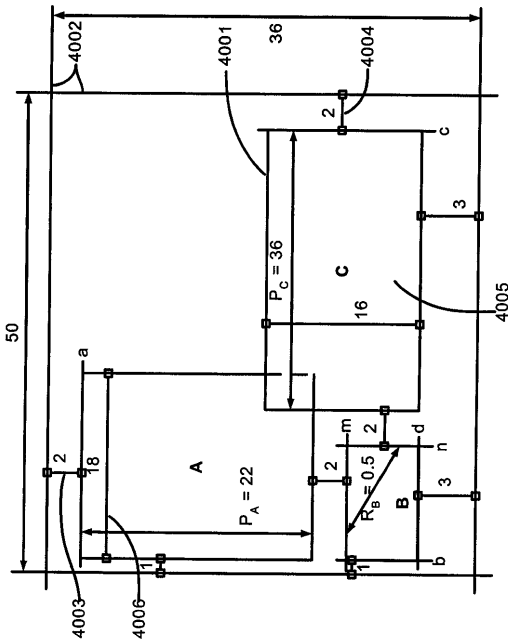
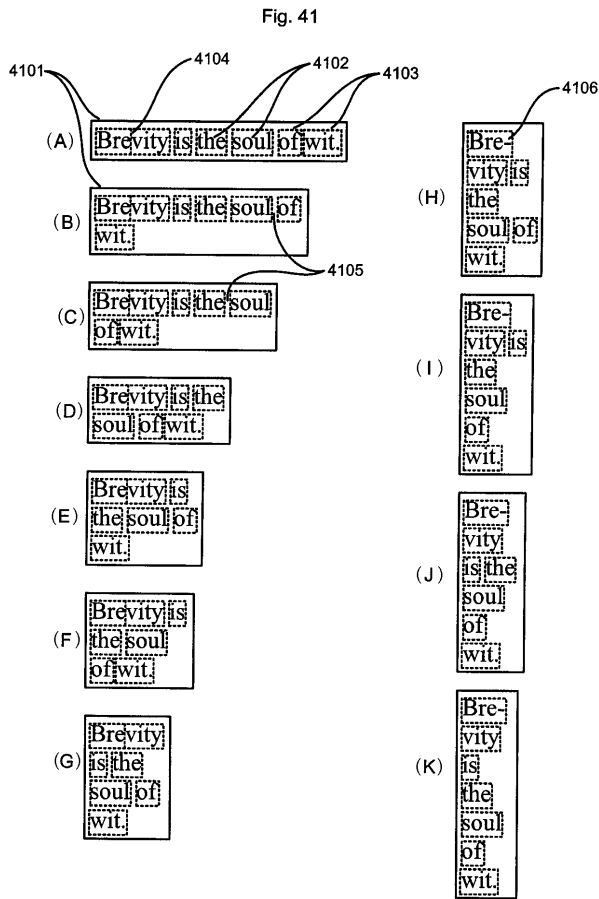


Fig. 40B

【図 4 1】



【図 4 2 A】

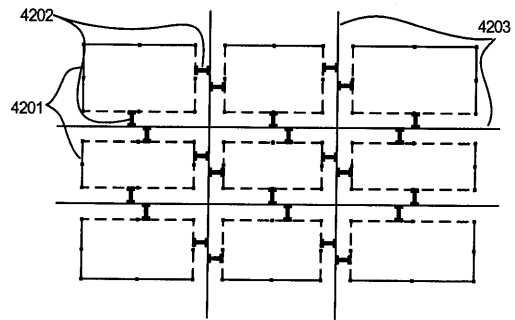


Fig. 42A

【図 4 2 B】

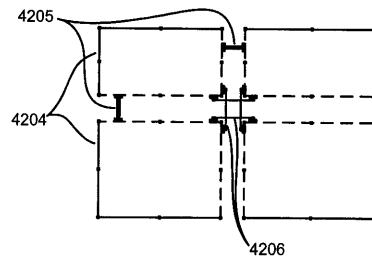


Fig. 42B

【図 4 2 C】

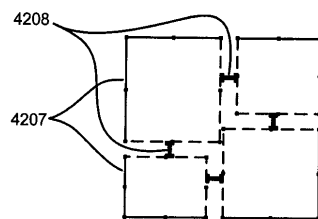


Fig. 42C

フロントページの続き

(72)発明者 アラン バレブ トニソン

オーストラリア国 2113 ニュー サウス ウェールズ州, ノース ライド, トーマス
ホルト ドライブ 1, キヤノン インフォメーション システムズ リサーチ オーストラ
リア プロプライエタリー リミテッド 内

審査官 酒井 伸芳

(56)参考文献 特開平06-203130 (JP, A)

特開平06-096177 (JP, A)

特開2001-076163 (JP, A)

特開平11-250271 (JP, A)

特開2003-241359 (JP, A)

特開2001-101179 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06T 11/60-11/80

B41J 21/00

G06F 3/12

G06F 17/21

H04N 1/387