

フォームシステム FOSTER における
レイアウトの自動調整について

5X-1

平川 直行 趙 巨民 宮尾 淳一 菊野 亨 吉田 典可
広島 大学

1. まえがき

筆者らはフォームシステム FOSTERを開発中である[1],[2]. 本稿では使い易さ向上のために導入した, レイアウト自動調整機能[3]についてその概要を紹介する.

2. レイアウト

二次元平面上に描かれる矩形をウィンドウ w_i と呼ぶ(図1). 各 w_i は基準点 p_i と大きさ s_i をパラメータとしてもち(図2参照), x 軸及び y 軸と平行に配置されるものと仮定する. ここで, 各パラメータは

$$p_i = (x_i, y_i)$$

$$s_i = (L_i, R_i, U_i, D_i)$$

とする. 特に, $L_i = U_i = 0$ の w_i をタイプ A, $L_i = R_i, U_i = D_i$ の w_i をタイプ B と呼ぶ. 本稿ではタイプ A と B に限定して議論する.

ウィンドウの有限集合 $W = \{w_i \mid 0 \leq i \leq n\}$ をレイアウトと定める(図1). 但し, w_0 を外枠のウィンドウと仮定する.

レイアウト W を構成するウィンドウ間の相対位置関係を表すため, レイアウトグラフ $G(W) = (N, E)$ を次の様な重み付き有向グラフと定める.

(1) $N = W$

(2) $E = E_H \cup E_V$, $E_H = \{(w_i, w_j) \mid x_i < x_j\}$, $E_V = \{(w_i, w_j) \mid y_i < y_j\}$. 但し, E は多重集合とする.

(3) 各 $(w_i, w_j) \in E_H$ にはラベル (H, β) , $\beta = x_j - L_j - x_i - R_i$ が, 各 $(w_i, w_j) \in E_V$ にはラベル (V, β) , $\beta = y_j - U_j - y_i - D_i$ がつけられる.

互いに重なりのない2つのウィンドウ w_i と w_j の相対位置関係は, レイアウトグラフ上の節点 w_i と w_j の間の枝に基づい

て, 表1に示す8通りに分類できる. 同表で, 例えば $\beta_{ij} \leq 0$ は w_i から w_j に向かう有向枝が存在し, ラベル中の β の値が0または負であることを, ϕ は有向枝そのものが存在しないことを示す.

表1 分類

パターン番号	ラベル(H, β)	ラベル(V, β)
1	$\beta_{ij} \leq 0$	$\beta_{ij} \geq 1$
2	$\beta_{ji} \leq 0$	$\beta_{ij} \geq 1$
3	ϕ	$\beta_{ij} \geq 1$
4	$\beta_{ij} \geq 1$	$\beta_{ij} \leq 0$
5	$\beta_{ji} \geq 1$	$\beta_{ij} \leq 0$
6	$\beta_{ij} \geq 1$	ϕ
7	$\beta_{ij} \geq 1$	$\beta_{ij} \geq 1$
8	$\beta_{ji} \geq 1$	$\beta_{ij} \geq 1$

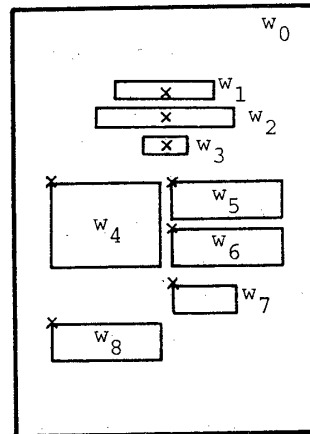


図1 レイアウト

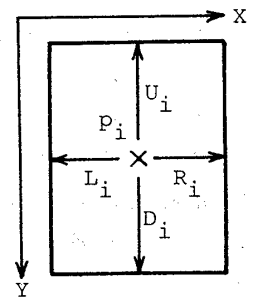


図2 パラメータ

3. 自動調整

FOSTERにおけるレイアウトの自動調整についてその概要を説明する.

入力として, 標準レイアウト W_0 (図1) と初期レイアウト W_1 (図3) が与えられる. W_1 に対し, 先ず $G(W_0)$ に出来るだけ

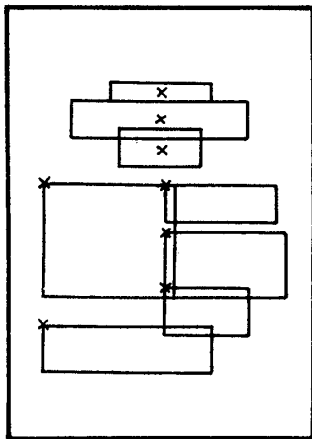


図3 初期レイアウト

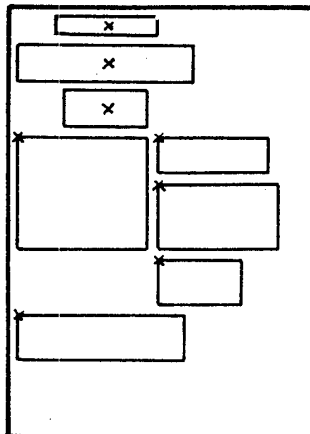


図4 重なりほぐし

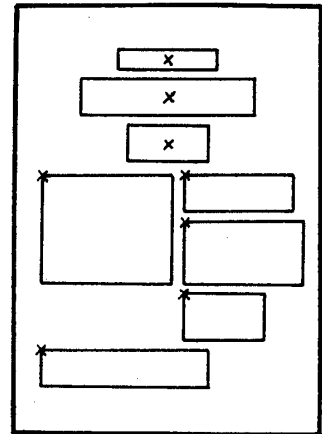


図5 均衡化

忠実にブロック間の重なりを除去したレイアウト W'_1 (図4) を求める。引き続き、 W_0 上の各ウィンドウの位置関係を出来るだけ反映する様に、再配置したレイアウト W''_1 (図5) を求める。

以降、 W_1 から W'_1 を求める手順を4. で、 W'_1 から W''_1 を求める手順を5. で、それぞれ述べる。

4. 重なりほぐし (図3, 4 参照)

ウィンドウ間の重なりを除去は表1のパターンに基づき、表2に従って行うものとする。但し、標準レイアウト W_0 上でパターン3, 6のウィンドウは重なり除去後もそれぞれパターン3, 6に限る。

次に、重なりほぐしアルゴリズムについて直観的に説明する。x方向、及び、y方向について重なりを取除きつつ、外枠のウィンドウ w_0 の左上の方向にコンパクションを行う。具体的には、レイアウトグラフ $G(W_1)$ 上の有向枝のラベル (正確には β の値) を更新して行く。重なりを取除くウィンドウ w_i の順序は基準点 p_i の値に基づく (x方向なら x_i の小さい順、y方向な

ら y_i の小さい順とする)。

この重なりほぐしアルゴリズムの時間計算量は $O(n^2)$ である。

5. 均衡化

ウィンドウ間の均衡化は、標準レイアウト W_0 上で隣接するウィンドウ間の距離の比が保たれる様に行う (図1, 5 参照)。

次に、均衡化アルゴリズムについて説明する。x方向、及び、y方向の均衡化の2つのステップからなる。レイアウト W'_1 上で基準点のx座標の値の大きいウィンドウから順に、x方向に移動する。次に、y方向についても同様に移動する。

この均衡化アルゴリズムの時間計算量もやはり $O(n^2)$ である。

6. むすび

FOSTERはPC-9801を用いて、MS-DOS (Ver. 2.11) 上でLattice Cでプログラム開発を行っている。自動調整アルゴリズムをFOSTERに組み入れて、性能評価を行うことについて現在検討中である。

文献

[1] Sugihara, K. et al. : " An approach to the design of a form language , " Proc. 1984 IEEE Workshop on Visual Languages, pp.453-478 (1982).
 [2] Kikuno, T. et al. : " Advanced layout processing in form system FOSTER, " Proc. 1986 IEEE Workshop on Languages for Automation, to appear.
 [3] 平川直行 : " レイアウトの自動調整アルゴリズム, " ECS Lab. Hiroshima Univ. Tech. Rep. No. 86-06 (1986).

表2 重なり除去

標準レイアウト 上でのパターン	重なり除去 後のパターン
1	1, 7
2	2, 8
4	4, 7
5	5, 8
7	1, 4, 7
8	2, 5, 8