

ウィジェットの 適応レイアウト問題

柳田拓人, 野中秀俊
北海道大学 大学院情報科学研究科 知能情報学研究室
http://aiwww.main.ist.hokudai.ac.jp/~takty/

フuzzy制約充足問題としての定式化と解法

グラフィカル・ユーザ・インタフェース (GUI) の生成において、ウィジェットのレイアウト最適化は重要な課題である。さらに、モデルに基づいたユーザ・インタフェース (UI) 設計の分野では、使用する**ウィジェットの選択**も必要とされる。この場合、UIの**論理仕様記述**に基づき、システムが自動的にGUIを生成することになる。論理仕様記述に基づく自動GUI生成システムには、次の2点が要求される。

1. どのウィジェットを使用し、それらをどのように並べるのかを決定すること、
2. そのレイアウトを一定の時間内に完了させること (特に実行時にUI生成を行う場合)

本研究ではこれらの条件を満たすレイアウトを**ウィジェットの適応レイアウト** (flexible widget layout: FWL) と呼び、FWL問題を**フuzzy制約充足問題** (FCSP) として定式化する。そして、与えられた論理仕様記述に基づき、自動的に適当なウィジェットを選択し、それらを配置する手法を提案する。

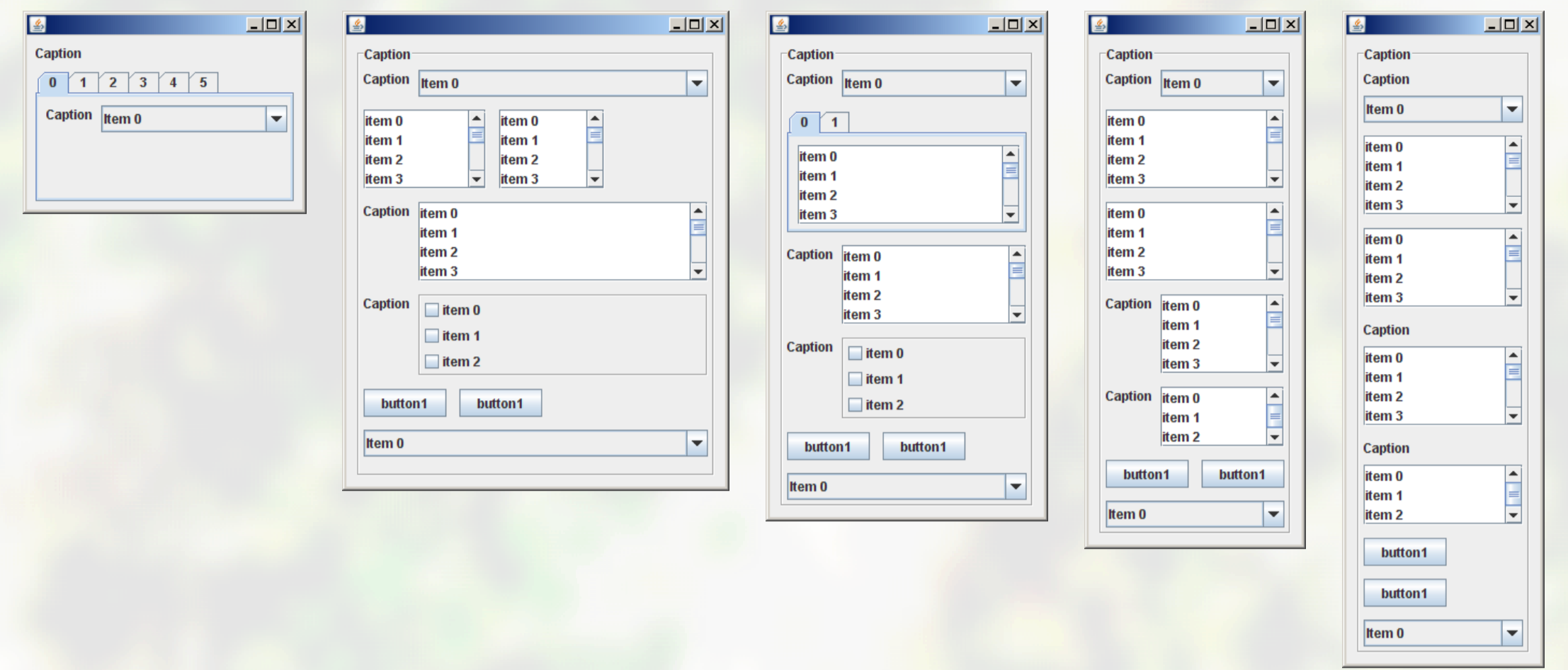


図1 ウィジェットの適応レイアウト

論理仕様記述には、デバイスやプラットフォームを特定しないそれらに共通な**UIの機能** (UI機能) が記述されている。この論理仕様記述におけるUIモデルとして、**選択行為モデル**を採用する。そこでは、個々のUI機能が選択行為 $(s_i = \langle L_i, e_i, t_i, o_i \rangle)$ として表現される。

- L_i 選択肢のリスト ($|L_i|$ は選択肢数)
- e_i 選択する個数 (選択数)
- $t_i \in [1, 10]$ 重要度
- o_i 相反フラグ

グループは関連する選択行為、及び他のグループを子要素とし、自身と子要素との間に包含 (親子) 関係を形成する。これによって、選択行為を葉ノード、グループをルートやその他のノードとするUI機能の木構造が形成される。なお、選択行為とグループにはキャプションとして、それ自体を説明する文字列を付加可能である。

UIの論理仕様記述によって、システムは指定されたUI機能を備えたウィジェットであれば任意に選択することが可能となるため、UIに多様性が生まれる。しかしながら、そのようなウィジェットは一意に定まらないこともあるため、ウィジェット選択が課題となる。

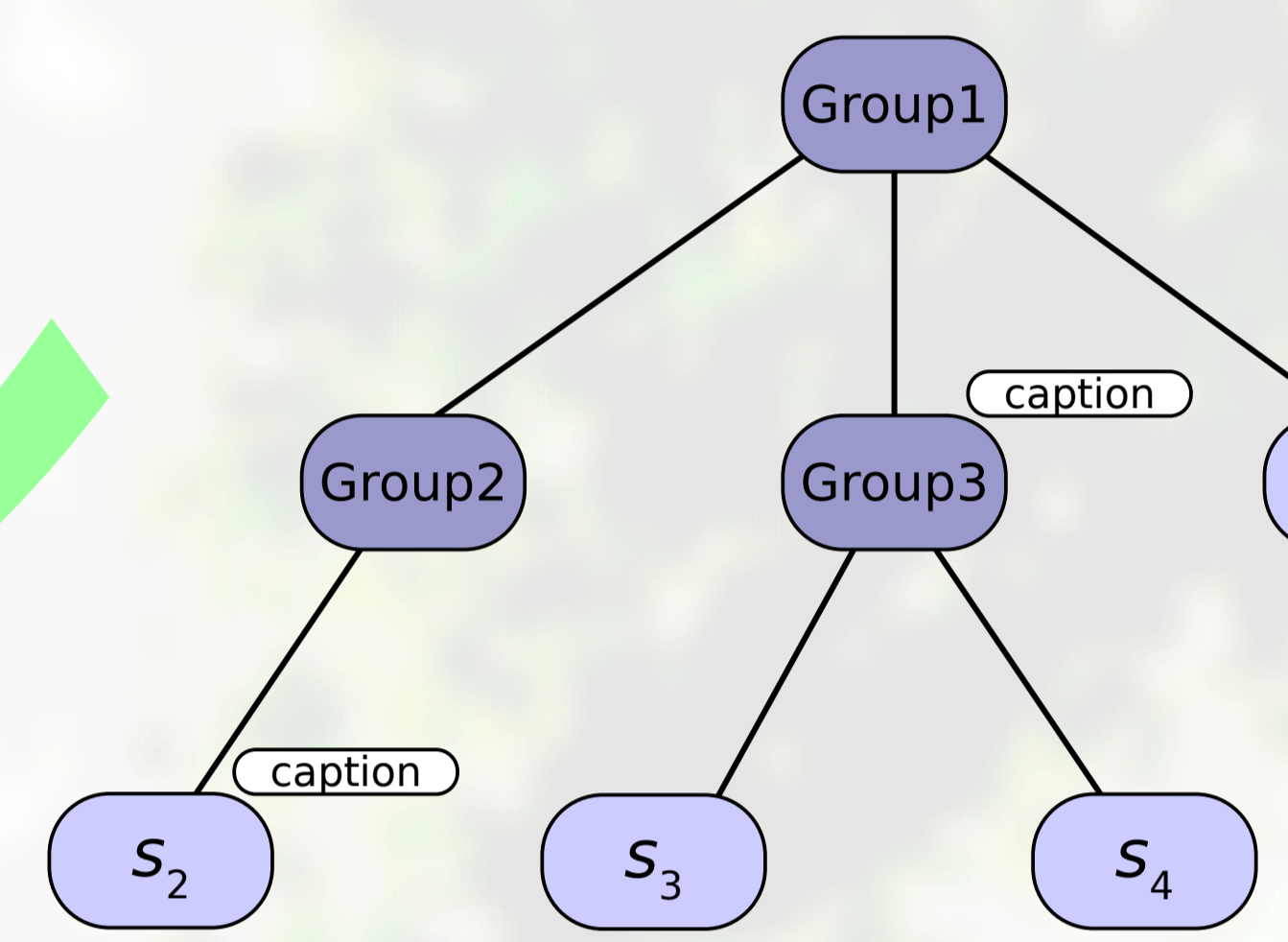


図2 UIの選択行為モデル

ウィジェットの適応レイアウト (FWL) 問題: 使用するウィジェットの候補とそれらのポジショニング (並べ方) の候補の組み合わせの中から、レイアウト可能で出来るだけ最適なものを探索する問題。**ウィジェット候補集合**や**ポジショニング候補集合**は、UI機能やそのグループを含むUIモデルの、論理仕様として書かれた記述に基づいて決定される。

UIモデルにおける選択行為を**ウィジェット**として表現する。ウィジェットの種類に対応した**望ましさ**の指標を $\alpha_i \in [0, 1]$ で表す。UIモデルにおける選択行為はFWLにおいてウィジェットの候補の集合へ対応づけられる。

UIモデルにおけるグループと選択行為のキャプションをFWLではそれぞれ**配列コンテナ**と**ラベル・コンテナ**として表現する。コンテナのポジショニングにはそれぞれ種類に応じた**望ましさ**の指標があり、ウィジェットの場合と同様に $\alpha_i \in [0, 1]$ で表す。UIモデルにおけるグループやキャプションはコンテナとしてFWLにおいてポジショニングの候補の集合へ対応づけられる。

配列コンテナには複数の子要素が含まれ、そのポジショニングとして縦並び、横並び、タブ表示の三つがある。

- a) Vertical alignment
- b) Horizontal alignment
- c) Tab pages

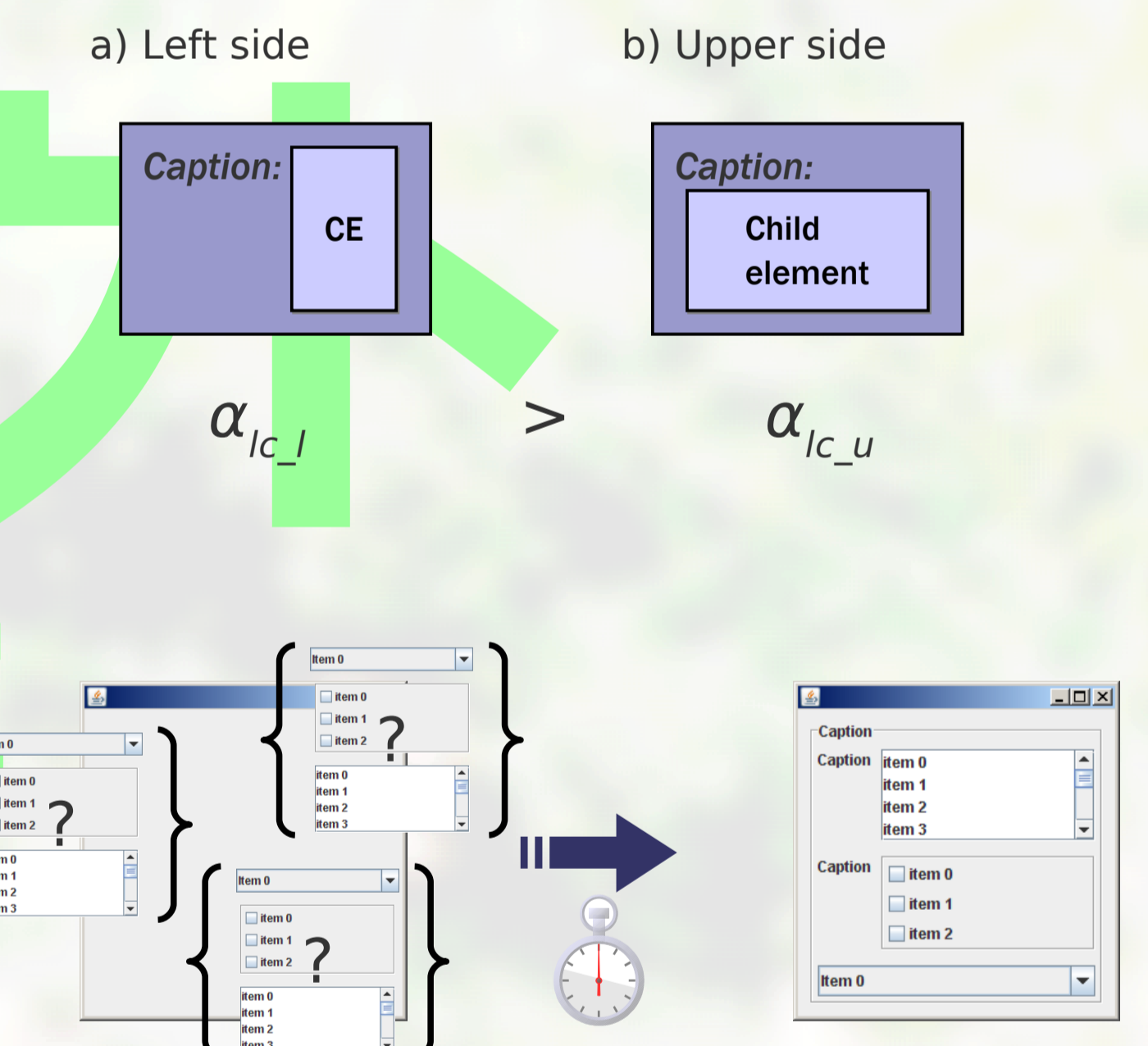
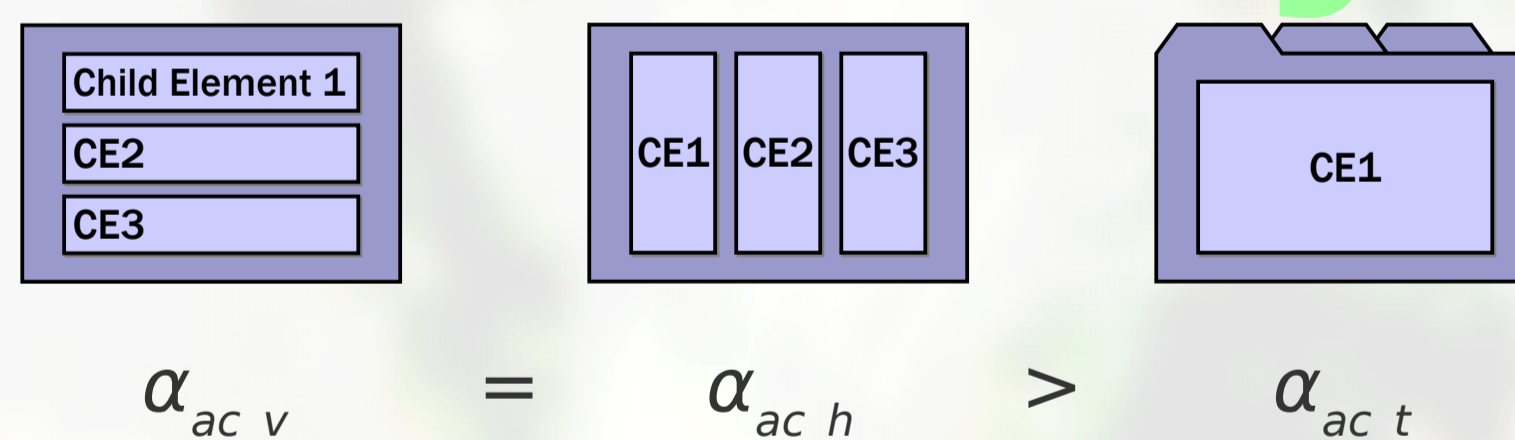
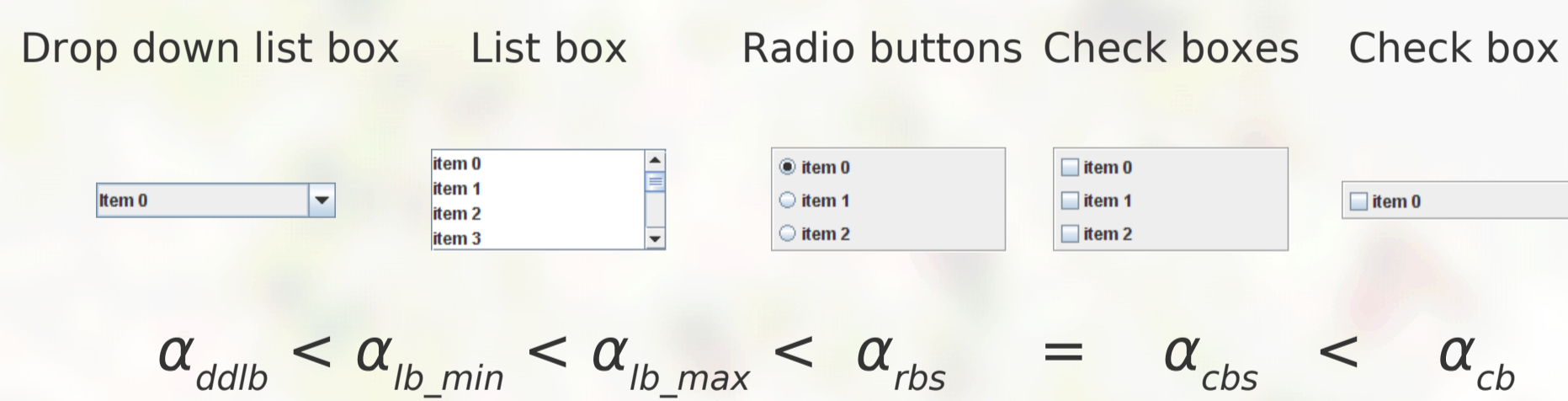


図3 FWLのイメージ



変数集合: $X = X_W \cup X_P \cup X_D$ はそれぞれ、ウィジェット、ポジショニング、ダイアログの候補集合とし、変数 $x_{Wi} \in X_W, x_{Pj} \in X_P, x_{Dk} \in X_D$ の値はそれぞれどの候補が選択されているのかを表す。

ウィジェット候補変数のドメイン: ウィジェット w_i とその最小サイズ (w_{wi}, h_{wi}) からなるタプル集合とする。

$$D_{x_{w_i}} = \{ \langle w_i, w_{w_i}, h_{w_i} \rangle \mid w_i \in W_i \}$$

例: $D_{x_{w1}} = \{ \langle \text{check_box}, 210, 18 \rangle, \langle \text{radio_buttons}, 210, 36 \rangle, \langle \text{drop_down_list_box}, 210, 18 \rangle \}$

ポジショニング候補変数のドメイン: ポジショニング p_j 、その最小サイズ (w_{pj}, h_{pj}) 、各子要素に対する許容最大サイズ $(W_{pj,k}, H_{pj,k})$ からなるタプル集合とする。

$$D_{x_{p_j}} = \{ \langle p_j, w_{p_j}, h_{p_j}, M_{p_j} \rangle \mid p_j \in P_j \}$$

$$M_{p_j} = \langle W_{p_{j,1}}, H_{p_{j,1}}, W_{p_{j,2}}, H_{p_{j,2}}, \dots, W_{p_{j,K_{p_j}}}, H_{p_{j,K_{p_j}}} \rangle$$

ダイアログ変数のドメイン: ダイアログのクライアント領域サイズ (W_d, H_d) を子要素の許容最大サイズとするタプルのみ。

$$D_{x_{D}} = \{ \langle W_d, H_d \rangle \}$$

望ましさを表す制約: ダイアログ変数を除く個々の変数には望ましさを表現する**単項制約**が付加される。

Satisfaction degree = desirability of candidate

包含関係を表す制約: 任意の二つの変数には包含関係を表現する**2項制約**が付加される。

$$C_{con_{P_{j,k}}}(v_1, v_2) = \begin{cases} 1 & \text{if } w_{p_{j,k}} \leq W_{p_{j,k}} \text{ and } h_{p_{j,k}} \leq H_{p_{j,k}} \\ 0 & \text{otherwise} \end{cases}$$

問題生成後、最悪制約充足度に基づいたドメインの枝刈りを行いながら、前方チェック・アルゴリズムを反復適用し、より良い変数への割り当てを求める。そして、変数割り当て、すなわち選択された候補に基づいて、実際の位置とサイズを決定し、ウィジェットを配置する。

提案システムをノートPC (Pentium M 1.10 GHz CPU, 512 MB メイン・メモリ, Windows XP Professional edition) 上にJava 6を用いて実装した。提案システムは、例題のレイアウトを250 ms以内に終わることが可能である。これはGUIの実時間で生成という課題に対して、実用上十分な速度と言える。

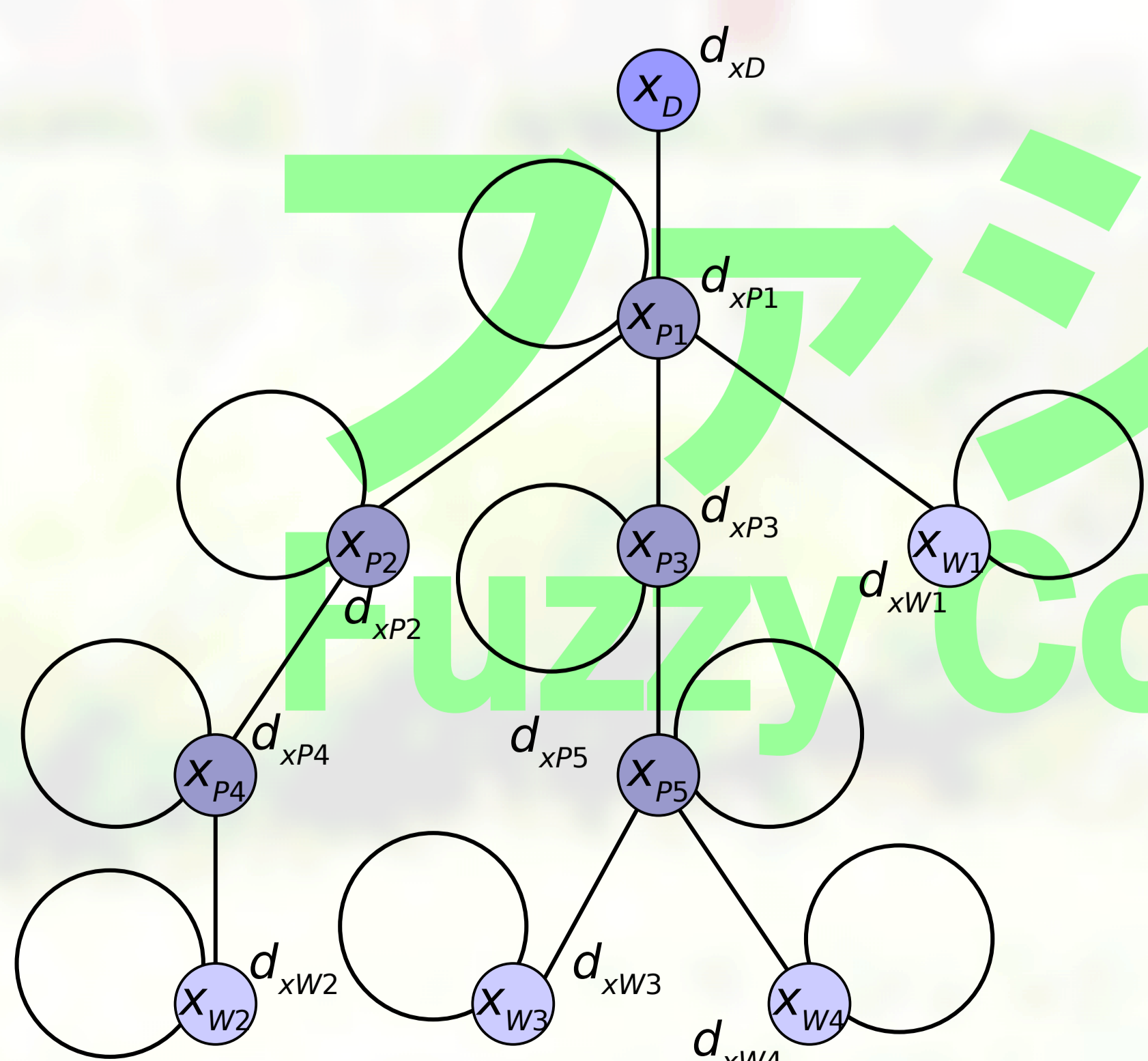
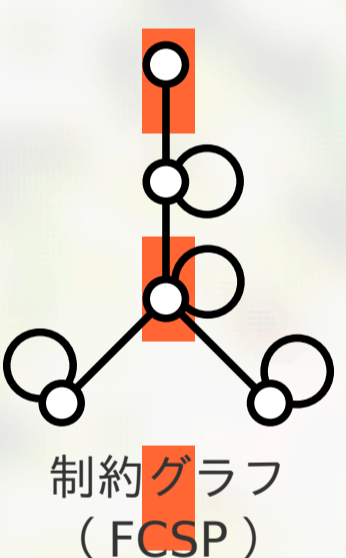


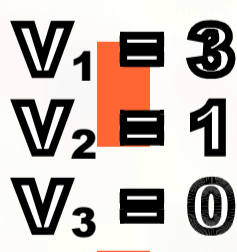
図4 FWLを表現する制約グラフ



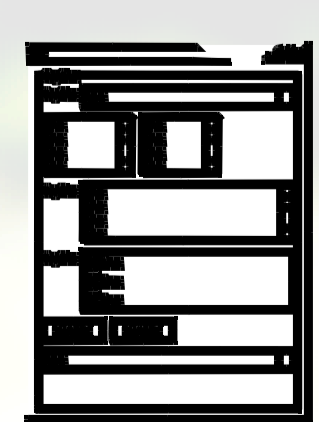
論理仕様記述 (AIDL)



制約グラフ (FCSP)



変数割り当て (導出された解)



実際のGUIレイアウト