

K-003

GUI定義とスキン定義を組み合わせたレイアウトの一方式 A Proposal of Layout Based on Skin and GUI Components

岡本 啓嗣[†]
Hirotsugu OKAMOTO

北村 操代[†]
Misayo KITAMURA

小島 泰三[†]
Taizo KOJIMA

1. はじめに

近年、スキン機能により、ユーザの好みに画面をカスタマイズできるアプリケーションが増加している。スキンは画面上の表示部品とその配置を定義する。スキンは各アプリケーションの主要画面に対して定義する必要があり、アプリケーション及びその主要画面の数が多い場合、その定義の作成が煩雑となる。一方、複数のアプリケーションの主要画面に対し、共通的に利用できるスキンを実現できれば、新たなアプリケーションが追加された場合にも、既存のスキンを利用でき、新たに作成する必要はない。例えば、携帯電話で新たなアプリケーションをダウンロードした場合にもユーザの好みを反映した画面を即座に提供できる。本稿では、このようなスキン機能の実現方法について提案する。

2. スキンの適用によるレイアウト変更

アプリケーションの表示画面のカスタマイズは、レイアウト定義に対してスキン定義を適用することによって行われる。レイアウト定義は、GUI部品の配置情報を記述したものであり、スキン定義は、アプリケーションの表示画面の変更定義である。スキン定義には、個々のGUI部品の縁に飾りを付加する部品デコレーションとキャラクター画像などの新たなGUI部品を配置する画面デコレーションの2種が含まれる。図1にレイアウト例を示す。図中(a)は、レイアウト定義のみ、(b)は、そのレイアウト定義にスキン定義を適用している。部品デコレーションにより、ボタン(BTN)等のGUI部品には縁取りが施され、画面デコレーションにより、画面の枠飾りや、キャラクター画像が追加されている。

適切な画面を得るためには、GUI部品間の相対的な位置関係が変わるようなGUI部品の移動や、GUI部品を小さくする等の方法によるレイアウト変更が必要となる。図1中、上段のようなGUI部品の配置の場合、BTN1とTEXT1は干渉するが、GUI部品を少し横方向へ移動させることで解決できる。しかしながら、図1下段のような場合、LABEL1も存在するため、横方向への移動だけでは画面からはみ出してしまふ。この場合、位置関係や大きさを変更することが必要となる。本稿では、相対的な位置関係を保持した単純な移動によるレイアウトの変更を小さなレイアウト変更と呼び、相対的な位置関係の変更や、サイズの変更などをともなうレイアウト変更を大きなレイアウト変更と呼ぶ。携帯電話等の環境では、アプリケーションが表示されるウィンドウサイズが固定であるため、本稿ではGUI部品を画面内にレイアウトすることを前提とする。そのため、大きなレイアウト変更は重要な役割を担う。

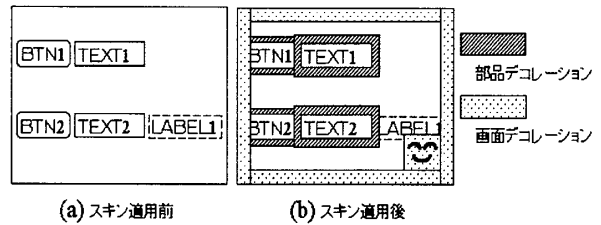


図1: スキン適用による干渉の例

3. プロパティ優先度

本稿では、レイアウト定義のGUI部品やスキン定義に含まれる個別要素の各プロパティごとに優先度(プロパティ優先度)を設定し、その値を基に、スキン適用時のレイアウトを計算する方式を提案する。プロパティ優先度を用いて、例えばサイズを変更すべきでないGUI部品を判別することができる。レイアウト計算では、干渉しているGUI部品を探し、まず小さなレイアウト変更を試みる。これによって干渉が解決できない場合、次に大きなレイアウト変更を行う。その際、GUI部品の位置やサイズ等のプロパティの操作が必要となるが、操作対象のプロパティとして、プロパティ優先度が低いものを選択する。

本方式を用いたレイアウト定義、スキン定義の例をそれぞれ表1、表2に示す。レイアウト定義では、各GUI部品のプロパティと各プロパティ毎の優先度を設定する。表1において、ガイドとは一連のGUI部品の並びを表現するプロパティである。例えば、横方向に関するガイドの値が同じGUI部品は、横方向に並ぶ。そのプロパティ優先度は、一連のGUI部品群において、直前のGUI部

表1: レイアウト定義

ID	種類	テキスト	画像	可視性		位置		サイズ		ガイド	
				x	y	幅	高さ	横	縦		
0	ラベル	LABEL	—	true	5	5	60	24	1	1	—
				10	8	10	7	8	—	—	—
1	テキストボックス	BOX	—	true	65	5	90	30	1	—	—
				10	8	5	10	—	—	—	—
2	ボタン	BTN	—	true	5	40	36	24	2	1	—
				10	8	10	7	8	—	—	—
3	ピクチャ	—	pic.png	true	100	50	50	50	—	—	—
				10	5	10	—	—	—	—	—

プロパティ優先度

表2: スキン定義

	縁取りの幅				優先度
	左	右	上	下	
ラベル	0	0	0	0	—
ボタン	3	3	2	2	3
テキストボックス	3	3	3	3	5

[†]三菱電機(株)先端技術総合研究所
Advanced Technology R&D Center, Mitsubishi Electric Corp.

品との関連の強さを表す。一方、スキン定義では、部品デコレーションと画面デコレーション、及びそれらの各プロパティ毎の優先度を設定する。画面デコレーションの定義方法はレイアウト定義と同様である。表2においては、スキン定義のうち、部品デコレーションについてのみ示している。部品デコレーションでは、GUI部品の種類ごとに、その上下左右の縁取りの幅を設定しており、この4つの値の組をプロパティとし、これに対する優先度を設けている。なお、ここではプロパティ優先度は10段階で表現している。

4. レイアウト処理例

本方式における処理の概略は以下のとおりである。

1. 画面左上のGUI部品から順に、干渉を起こしているGUI部品の重畳領域のサイズが、縦 \geq 横であるものに関して、横方向へ小さなレイアウト変更を試みる。
2. 横方向への小さなレイアウト変更では干渉を解決できなかったGUI部品に対し、プロパティ優先度を基に大きなレイアウト変更を行う。
3. 重畳している領域のサイズが、縦 $<$ 横であるものに関して1、2と同様の処理を縦方向に対して行う。

ここでは特に、大きなレイアウト変更時に操作対象となるGUI部品とその操作内容を決定する方法の一例を図2に示す。図2において、リストLISTに画面内に収まりきれない一連のGUI部品が登録されている。例えば図1(b)下段のBTN2、TEXT2、LABEL1等である。変更するプロパティ決定のために、LIST中のGUI部品に対し、各プロパティの基準値 T を設ける。プロパティ優先度を P 、可視性、位置、サイズ、ガイド、部品デコレーションの各プロパティを表現する文字を v 、 l 、 s 、 g 、 d とする。例えば、位置プロパティの基準値を T_l 、GUI部品 i のサイズのプロパティ優先度を P_{si} と表す。図中if文では、まず重要性の非常に低いGUI部品があればそれを非表示にしてレイアウト変更を行う。switch文においては、基準値の最も小さいプロパティを変更する操作を行う。本処理の適用例を図3に示す。画面右下のピクチャ部品は可視性のプロパティ優先度が低く、中央付近の「次へ」のボタンは位置のプロパティ優先度が低い。この結果、レイアウト変更により、ピクチャ部品は非表示、ボタンは大きく移動している。なお、ここで示した処理は一例であり、大きなレイアウト変更もしくはレイアウト変更方法全体に対し、いくつかのバリエーションが考えられる。

5. おわりに

本稿では、複数のアプリケーションで共通的に利用できるスキンを定義して、画面レイアウトを行う方式を提案した。本方式では、レイアウト定義とスキン定義の各プロパティ毎にプロパティ優先度を設定し、スキン適用時にGUI部品が干渉した場合、この値を基に、干渉を解決するようレイアウト変更を行う。この方式に基づいた処理例とレイアウトの例を示した。

現在、本方式を[1]で示されるような携帯電話応用システムへ適用することを検討している。このシステムで

```
while 干渉がある
```

```
   $S_v = \min_{i \in LIST}(P_{vi})$ .
```

```
  ( $T_l$ 、 $T_g$ 、 $T_d$ も同様)
```

```
   $P_{si} \neq N$  ( $N$ はプロパティ優先度の上限値)なる
```

```
  GUI部品 $i$ に対し、
```

```
   $T_s = \text{ave}_{i \in LIST}(P_{si})$ 
```

```
  if  $T_v < \alpha$  ( $\alpha$ は予め決定された閾値)
```

```
  then
```

```
     $P_{vi} = T_v$ であるGUI部品 $i$ を非表示化。
```

```
     $i$ をLISTから削除
```

```
  else
```

```
     $T_v$ 、 $T_l$ 、 $T_s$ 、 $T_d$ を比較
```

```
  switch
```

```
  case :  $T_l$ が最小
```

```
     $P_{li} = T_l$ であるGUI部品 $i$ を移動。
```

```
     $i$ をLISTから削除。break
```

```
  case :  $T_s$ が最小
```

```
     $P_{si} \neq N$ であるGUI部品 $i$ を、プロパティ優先度の逆比でサイズ縮小。break
```

```
  case :  $T_g$ が最小
```

```
     $P_{gi} = T_g$ であるGUI部品 $i$ の直前で改行。
```

```
     $i$ をLISTから削除。break
```

```
  case :  $T_d$ が最小
```

```
     $P_{di} = T_d$ であるGUI部品 $i$ の部品デコ
```

```
    レーションを非表示化。break
```

```
  end switch
```

```
  LIST中のGUI部品に対し小さなレイアウト変更
```

```
end while
```

図2: 変更プロパティ決定処理の例

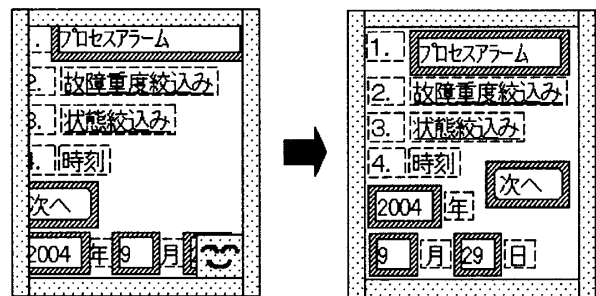


図3: スキン適用時の干渉解決の例

は、アプリケーションの画面数が多く、個々の画面に対してスキンを定義するのは非常に効率が悪い。本方式によって、一つのスキンで効率よく画面をカスタマイズすることができる。今後、処理方法を洗練し、システムへ実適用する予定である。

参考文献

- [1] 北村操代, 轟木伸俊, 秋吉正徳, 小島泰三, “インタラクティブなロジックの分割による携帯電話応用システム”, 電気学会論文誌 C Vol.125, No.3, pp.502-511 (2005).