

GUI プログラムを対象としたリエンジニアリング

4 J-9

小泉 昌紀* 中島 震* 杉山 高弘**

* 日本電気(株) C&C システム研究所

** 日本電気(株) ソフトウェア生産技術開発本部

1 はじめに

近年、既存プログラムを新規開発するよりも保守改造する要求が急速に増大している。特に、グラフィックユーザインタフェース(GUI)は、アプリケーションソフトウェアのかんりの部分を占め、複雑に既存プログラムを修正する必要がある分野である。しかし、一般的にGUIプログラムはライブラリ体系が複雑かつ莫大なため理解しにくく、ソースプログラムと画面レイアウトの関係が直観的にわかりにくいいため修正にかかる工数が大きくなるという問題があった。

GUIプログラムから、ボタン、メニューなどのGUIオブジェクトの属性とそれらの関係などの画面レイアウト情報を抽出できれば、この情報をGUIビルダ上で容易に変更できる。しかし、既存のリバース技術ではGUIプログラムから画面レイアウト情報に変換する過程は機械化できないため、現状では人手に頼らなければならない。

本稿では、GUIプログラムからGUIオブジェクトに関する情報を抽出して中間言語として表し、それを評価することにより画面レイアウト情報を生成する方式を提案する。この方式により、関数呼び出しや制御文を含んだGUIプログラムから近似的な画面レイアウト情報を半自動的に抽出することができる。本稿では、GUIライブラリは「鼎[2]」、GUIビルダは「ゆず[3]」を例として方式を考察する。「鼎」は、X-Toolkitの上位層にあるC言語ライブラリであり、「ゆず」は、画面上で対話的にGUIオブジェクトの画面レイアウトを編集し、鼎プログラムを自動生成する機能を持つGUIビルダである。

2 GUIプログラムのリバース

2.1 対象とする問題の定義

入力 GUIプログラムは、GUIオブジェクトを表示する画面レイアウト部分と、コールバック関数本体などを記述したアプリケーション部に分解できる。画面レイアウト部分では、画面上にボタン、メニューなどのGUIオブジェクトを生成し、それらの間にサブメニュー、ポップアップなどの関係を張る。「鼎」では、これを次のGUIオブジェクト操作関数を用いて記述する。

1. GUIオブジェクト生成関数群
GUIオブジェクトを生成または属性を設定する関数で、引数は属性(x座標など)と属性値のペアである。
2. GUIオブジェクトを関係付ける関数群
サブメニュー関係などを表現する

例題の鼎プログラムを図1に示す。このプログラムは、図2に示すようにフォーム上に3つのボタンを水平方向に連続して表示するプログラムの一部である。繰り返し文から呼ばれる関数makebutton中で鼎ライブラリ関数CxMakeManagedWidgetを呼び出してGUIオブジェクトを生成する。本関数

Reengineering for GUI Programs.

Masaki KOIZUMI, Shin NAKAJIMA, and Takahiro Sugiyama
NEC Corporation

```
Widget form,button[3];
main()
{
  int i,x,y;
  y=0;
  makeform();
  for (i=0; i<3; i++){
    x=scale(i);
    makebutton(i,x,y);
  }
}
makebutton(i,x,y)
int i,x,y;
{
  button[i]=CxMakeManagedWidget(
    "button",buttonWidgetClass,form,
    XtNx,x, XtNy,y,
    XtNwidth,80, XtNheight,50,
    NULL);
}
```

図 1: 例題鼎プログラム

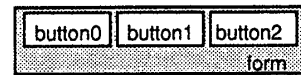


図 2: 例題鼎プログラムの出力

ではパラメータとしてオブジェクト名、クラス名、親オブジェクト名、x,y座標、幅、高さを指定する。

出力 画面レイアウト情報とは、GUIオブジェクトの属性とそれらの間の関係を表したデータ形式である。「ゆず」では、このデータ形式はS式で記述されており、1,2のそれぞれに対応した構文を持つ。この構文の引数は属性と属性値のペアから成り、属性値は定数で明示しなくてはならない。本稿では、このデータ形式を画面レイアウト情報と呼ぶ。図1のプログラムに対応した、画面レイアウト情報を図3に示す。

2.2 問題点

GUIプログラムから画面レイアウト情報を得るためには、入力のGUIオブジェクト操作関数中の属性値を抽出し、画面レイアウト情報の形式に変換すればよい。しかし、プログラム中で常に属性値が明示されているとは限らないため、以下

```
(load-widget "button0" 1 "form"
  'x "0" 'y "0" 'width "80" 'height "50"
)
(load-widget "button1" 2 "form"
  'x "80" 'y "0" 'width "80" 'height "50"
)
(load-widget "button2" 3 "form"
  'x "160" 'y "0" 'width "80" 'height "50"
)
```

図 3: 画面レイアウト情報

の解析をする必要がある。

1. 中間変数
2. ユーザ定義関数中にある GUI オブジェクト操作関数
3. 制御構造中にある GUI オブジェクト操作関数

これらの解析は、従来のリバースエンジニアリング技術を用いても機械化できなかった。従来のリバース技術 [1] は、ファイル依存関係 (C 言語では include 関係)、関数呼び出し関係、制御フロー (if, while など) などプログラム構造に関する情報は解析できるが、GUI オブジェクトに関する情報を持たないため、GUI オブジェクトに特化した解析を行なうことができない。したがって、利用者がリバースツールが出力した情報をもとにして、画面レイアウト情報を新たに作成する必要があった。

3 GR 表現を用いたリエンジニアリング方式

提案するリエンジニアリング方式の枠組みを図4に示す。中間表現として、GUI オブジェクトの関係を表現した GR (GUI Objects Relation) 表現を用いる。本方式は、GUI プログラムの構文解析を行ない、構文解析木から GR 表現を抽出し、GR 表現の情報を対話的に補い、GR 表現から画面レイアウト情報を生成する。GUI ビルダはこの画面レイアウト情報を基に GUI オブジェクトを視覚的に表示する。

3.1 GR 表現

GR 表現は、GUI オブジェクトを表すノードと、それらの関係を表現したリンクから成る。

例題プログラムに対する GR 表現を図5に示す。フォームと3つのボタンをノード G1-G4 で表現する。各ノードにはクラスリンクを用いてオブジェクトのクラスを明示する。G2-G4 と G1 は親子関係にあり、親子関係リンクが張られている。各ノードには属性リンクを介して属性との関係が張られている。G2 と G3 の間、G3 と G4 の間には、水平的に隣的位置にあることを示す位置制約が張られている。

3.2 GR 表現の抽出

1. GR 表現の構造抽出

構文解析木をプログラムの実行順にたどり、以下の情報に注目することにより GR 表現のノードとリンクを自動的に生成する。

- GUI オブジェクト生成関数からノード生成
- GUI オブジェクト関連付け関数からリンク生成

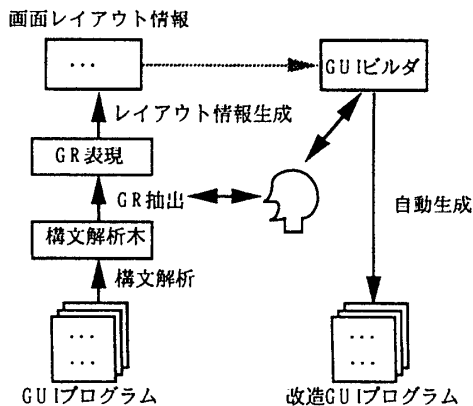


図4: 方式の概要

ただし、以下の実行時に決定する情報は利用者に質問することにより、対話的に値を決定する。

- 繰り返し文の実行回数
 - 条件分岐がある時はどのパスを選択するか
2. 位置制約の付与
プログラム中で、GUI オブジェクトの位置属性を表現するのに具体的な評価値を用いている場合は、GUI オブジェクト間の位置制約を利用者が与える。位置制約として、上下左右関係などを用意している。例題では、G2, G3, G4 の x 座標属性が具体的な評価値でないため、利用者が、G3 は G2 の右に配置することを表現した位置制約 `right(G2, G3)` を与える。システムはこの制約をより詳細な記述に展開し、GR 表現の対応するリンクに付加する。

3.3 画面レイアウト情報生成

1. 制約評価
画面レイアウト情報では、属性値は全て定数でなければならない。そこで、GR 表現中の位置制約を評価することにより、定数値を求める。
2. デフォルト値埋め込み
画面レイアウト情報では、必要な属性値を全て与えなければならないが、GR 表現中でこれらの値が与えられているとは限らない。GUI オブジェクトの種類、属性に応じてデフォルト値を埋め込む。

例題では、図3に示す画面レイアウト情報を生成する。

4 おわりに

関数呼び出し、制御文などを含む GUI プログラムから、GUI オブジェクトに関する中間表現 GR 表現を抽出し、GR 表現中の位置制約を評価することにより近似的な画面レイアウト情報を半自動的に生成する方式を提案した。GUI プログラミング環境「鼎」を用いて記述されたプログラムを、GUI ビルダ「ゆず」の内部形式へ半自動的に変換する例を通じて本方式の見通しを得た。現在、本方式に基づくプロトタイプシステムを実装中である。

本稿で提案した GR 表現は「ゆず」のデータ形式を直接反映しており、GUI オブジェクトの個数と属性値が静的に確定する部分しか表現できない。広範な GUI プログラムを扱えるように GR 表現を拡張することが今後の課題である。

参考文献

- [1] Y.Chen, M.Y.Nishimoto, and C.V.Ramamoorthy, The C Information Abstraction System, IEEE Trans. on Soft.Eng., Vol.16, No.3, pp.325-334, 1990.
- [2] 暦本純一他, エディタを部品としたユーザインタフェース構築基盤: 鼎, 情報処理, Vol.31, No.5, pp.602-611, May.1990.
- [3] 垂水浩幸, 杉山高弘他, 「鼎」上の対話的 X ウィンドウユーザインタフェース構築環境「ゆず」, 第13回 UNIX シンポジウム, Jul.1989.

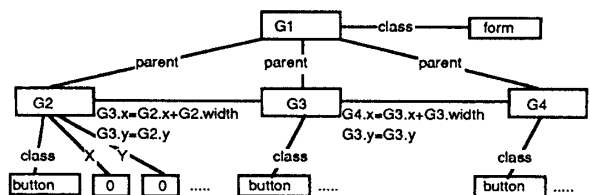


図5: GR 表現