

GUI構築支援ツール「ペンぎん」

5Q-6

井上 雅裕 山村 由紀子 鈴木 由美子

(株)沖テクノシステムズ ラボラトリ

1 はじめに

最近、Xウィンドウなどのウィンドウシステム上で動作するアプリケーションが多く作られるようになってきた。しかし、テキスト画面での使用を主体としていた従来のアプリケーションに比べ、より使い易いグラフィカルユーザインタフェース(GUI)を構築するためには、Motif^[1]対応のウィジェットなどを用いた面倒で手間のかかるプログラミングが必要となってくる。

本稿では、ウィジェットを画面上で対話的に配置し、それらを管理するプログラムを自動的に生成することができるツール「ペンぎん」について報告する。

ペンぎんはMotifに対応したGUI構築支援ツールで、テストモードを備え、グラフィックデータを扱えるなどの特徴がある。

2 ペンぎんの特徴

ペンぎんには以下の特徴がある。

1. GUI構築支援ツールとして、DevGuide^[2]などがあるが、一つのアプリケーションに対するインタフェースを一つの間接ファイルで管理している。本ツールでは複数の中間ファイルを同時に読み込むことができ、よく使うウィジェットのグループを中間ファイルとして管理しておくことができる。

2. 作成したグラフィカル・ユーザ・インタフェースの構成要素を一覧表示やツリー表現ができ、その情報を画面表示以外にファイルに保存することができる。
3. ベースウィンドウ、プリテンボード、フォーム上に格子線を表示することができ、格子線上の交点に構成要素を位置付けたり、各部品の間隔が等間隔になるように再配置することができる。
4. メニューウィンドウ、ヘルプウィンドウ、ダイアログウィンドウを専用のエディタで作成できる。
5. コールバックエディタにより、コールバックルーチンを作成し、テストモードに切り替えることによりコールバックの動作テストを行なうことができ、実際の操作感覚が確かめられる。
6. ビットマップやビクスマップなどのグラフィックデータを使用することができる。

3 インタフェースの設計の流れ

3.1 インタフェースの作成

画面レイアウトは、グラフィカル・ユーザ・インタフェースの部品(ウィジェット)が格納されているパレット・ウィンドウからマウスを使ってドラッグし、ワークスペース上に配置することによって作成する。

図1にペンぎんを使ったレイアウトの例を示す。

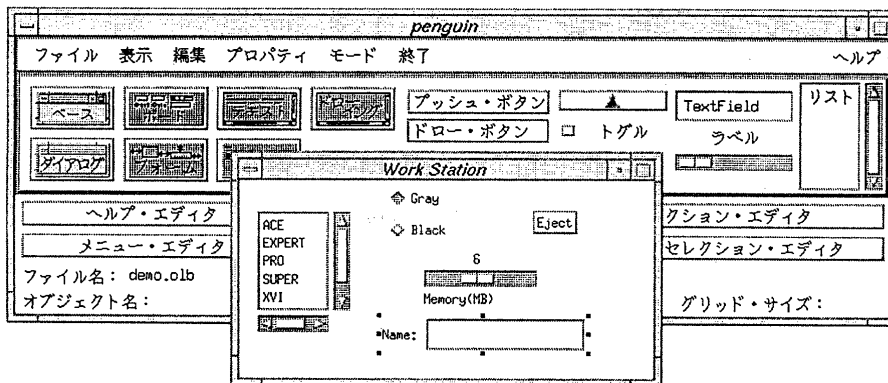


図1. レイアウトの例

GUI Layout Builder: "PENGUIN"
Masahiro INOUE, Yukiko YAMAMURA and Yumiko SUZUKI
Oki Technosystems Laboratory, Inc.

部品としては、プッシュボタン、スケール、リスト、テキストフィールド、アローボタン、トグルボタンなど Motif でサポートされているウィジェットを用意している。

また、属性設定ウィンドウ上でマウスとキーボードを使ってウィジェットのリソースを変更することができ、見た目の変化をすぐに確認することができる。

図2に属性設定ウィンドウの例を示す。

さらに、ビットマップファイルやピクスマップファイルなどのグラフィックデータをサポートしている。ボタン、ラベル、メニューなどラベルを使う場所に表示することができ、対応する表示ルーチンも自動生成される。

3.2 インタフェースの編集/統合

コールバックエディタでは、イベント発生時に行なう処理をコーディングしておくことができ、コールバックルーチンに対するコメントなどを記述しておくこともできる。そして、各ウィジェットには、コールバックルーチンのエントリ名を記述する箇所があり、コールバックエディタによって作成したコールバックルーチンの呼び出しを指定することができる。

ヘルプエディタでは各ウィジェットに対する説明文を、メニューエディタではサブメニューの内容を記述し、ダイアログエディタでは各種ダイアログのレイアウト設定などを行なう。

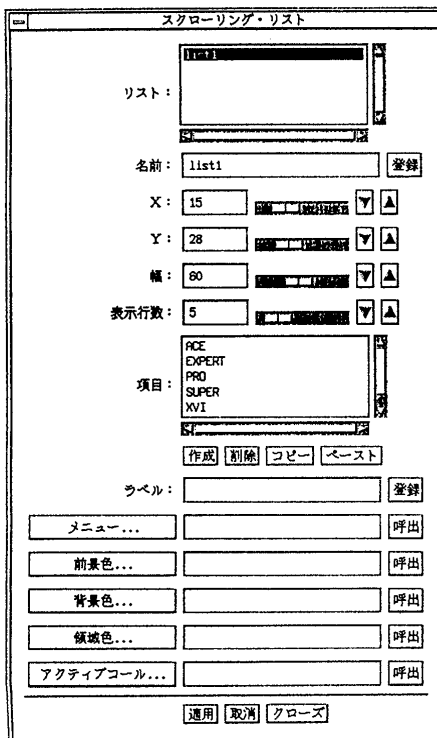


図2. 属性設定ウィンドウの例

構築したグラフィカル・ユーザ・インタフェースは中間ファイルとして保存できる。保存した中間ファイルは再びロードして、インタフェースに変更を加えることもでき、複数の中間ファイルを同時に読み込むことができる。これによって、インタフェースをモジュール化し、それを組み合わせて画面設計をすることができる。

3.3 インタフェースの確認/テスト

テストモードに切り替えると、ウィジェットを実際に操作してそのフィーリングを確かめることができる。コールバックルーチンを登録しているなら、コールバックルーチンが呼び出されるときに、起動される要因となるイベント名とルーチン名が表示される。また実体テストボタンをクリックすると、インタフェースのソースを自動生成し、それをコンパイルして、サブプロセスとして起動される。

3.4 ソースコードの自動生成

レイアウトが完成したら、中間ファイルから Motif ツールキットに対応した C 言語のソースファイルを生成する。生成されたソース・ファイルとユーザが開発したアプリケーションの本体をリンクすることによって、グラフィカル・ユーザ・インタフェースを使用したアプリケーションを開発することができる。

4 おわりに

グラフィカル・ユーザ・インタフェースを画面上で対話的に構築し、C 言語のソース・プログラムを自動生成するためのツール「べんぎん」について報告した。

べんぎんはテストモードを備え、ビットマップやピクスマップなどのグラフィックデータをサポートしている。

グラフィカル・ユーザ・インタフェースをコーディングしなくても作成できるので、デザイン (ルック & フィール) に時間をかけることができる。

今後の課題として、C 言語のみでなく他の言語、例えば Common Lisp のプログラムを自動生成することがある。

参考文献

- [1] Open Software Foundation, "OSF/Motif Programmer's Reference Release 1.1", Prentice-Hall, Inc 1991.
- [2] Sun Microsystems, Inc., "日本語 DevGuide 1.2 ユーザーズ・ガイド", 日本サン・マイクロシステムズ株式会社 1991.