

AutoView のコールバックルーチン作成負担軽減機能

2 T-4

吉山 正治, 笠原 宏
東京電機大学 工学部

1 はじめに

グラフィカルユーザーインターフェース (GUI) の専門的な知識を持たない End-User にとり、GUI を作成するのは非常に困難である。その原因として、GUI を作成するために用意されている ToolKit 等の使用方法が困難、GUI の動作等を記述するためのコールバックルーチンの記述が複雑、等があげられる。

現在これらの問題を解決するためのさまざまな GUI 構築ツールなどが登場してきている。しかしこれらの多くは GUI プログラマの開発負担の低減を目的としたものあり、GUI の専門的な知識を持たないユーザにとり、これらのツールは使用することさえも困難である。現在我々はこのような GUI の専門知識を持たない End-User を対象とした UI 構築ツール、AutoView[1] を作成している。

本論文ではコールバックルーチンの作成負担軽減機能を AutoView 上にインプリメントしたので報告する。

2 AutoView の概要

AutoView は GUI の専門知識を有しない End-User を対象とした UI 構築ツールである。ここでいう End-User を以下のように定義する。

1. GUI の使いかたは知っているが、オブジェクトの階層、構造等は知らない。
2. GUI に依存しない一般的なプログラミング知識がある。

現在の発表されている UI 構築ツールの多くは上述のようなユーザを対象としたものではない。そこでこのような End-User を対象とした UI 構築ツール AutoView を XView[2] を用い OpenWindow 上に作成した。図 1 に AutoView の基本画面を示す。ユーザは左側に用意された作成アイテムの中から、作成したいオブジェクトを選

びだし右側に用意されたキャンバス上にそのオブジェクトを配置する。

この AutoView が持つ End-User を支援する機能として以下のものがある。

1. 作成するオブジェクトの親子関係を考慮せずオブジェクトを作成できる。
2. オンラインヘルプにより機能等を確認できる。
3. 作成オブジェクトのサンプルの提示。
4. エラールーチンのバンドリング。
5. プログラムの保守性の向上。

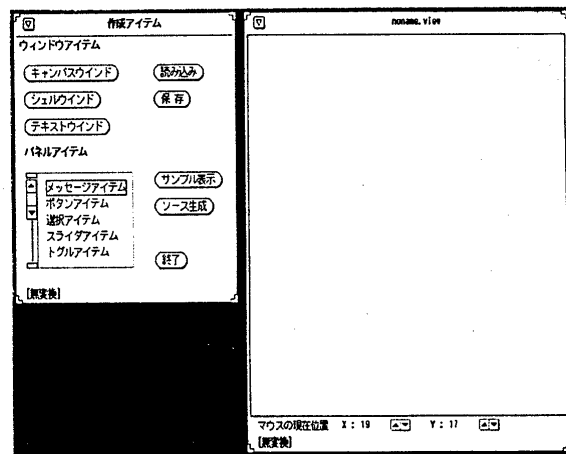


図 1: AutoView

3 コールバックルーチン作成負担軽減機能

AutoView で作成されたオブジェクト群に対してその動作を規定するコールバックルーチンを結合しなければならない。しかしコールバックルーチンはその記述が非

常に複雑である。そこで以下に述べるいくつかのコールバックルーチン作成負担軽減機能を AutoView に実装した。

3.1 呼び出し関数の用意

あらかじめ各オブジェクトから呼び出される関数がある程度用意しておく。ユーザはそのリストの中から自分の目的と一致する関数を選択する。

例えばキャンバスウィンドウを作成した場合、グラフィックエディタ、イメージの表示、グラフ作成等の関数を用意しておきそこから選択するようにする。

呼びだし関数リストは機能の異なるそれぞれのオブジェクトごとに用意する。

呼びだし関数指定時の図を 2 に示す。

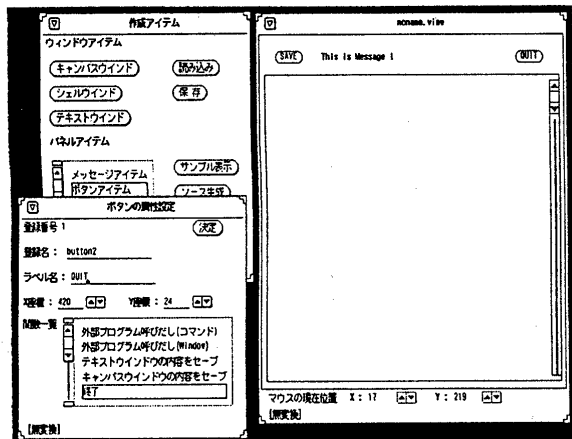


図 2: 呼び出し関数の指定

3.2 呼びだし関数の再利用

3.1 のリスト中にユーザが目的とする関数が存在しない場合が数多く考えられる。この場合ユーザがリスト中の関数を手直しすることにより新たに独自の関数として利用できることが多い。そこでリスト中にある関数を容易に変更を行えるよう、以下の基準で呼びだし関数の作成を行なった。

1. 関数の機能を明らかにする。
2. ウィンドウに依存する関数が使われている場合コメントを挿入する。
3. 専用の利用簡単なライブラリを用意する。

図 3 にリスト中の関数の内容を示す。

```

/*
 * 関数名:CanvasProc
 * 機能 :キャンバスで発生したイベントを受けとる
 *
 * $Log$
 * $Date$
 */
/*補助関数: int event_x(event):マウスのX座標
 *          int event_y(event):マウスのY座標
 *          void line(x1,y1,x2,y2):(x1,y1)-(x2,y2)にラインを引く
 *          void box(x1,y1,x2,y2):(x1,y1)-(x2,y2)のボックスを書く
 *          void circle(x,y,r):(x,y)を中心とし半径rの内を書く
 */
void CanvasProc(window,event)
Xv_Window window;
Event *event;
{
switch(event_action(event))
{
case LOC_MOVE/*
break;
case LOC_DRAG:
break;
case ACTION_SELECT/*マウスの左ボタンを押した時の処理*/
case MS_LEFT:
break;
case MS_MIDDLE/*マウスの中ボタンを押した時の処理*/
break;
case ACTION_MENU/*マウスの右ボタンを押した時の処理*/
case MS_RIGHT:
break;
}
}

```

図 3: リスト中の関数

3.3 既存のプログラムの利用

OpenWindows 上では X-Window, SunView などの様々なアプリケーションが用意されている。ユーザが新しいアプリケーションを作成しなくとも、このように既に用意されているアプリケーションを利用することにより目的とする機能を実現できることがある。そこで呼びだしアプリケーション名を指定するだけでアプリケーションを呼び出せる機能も AutoView に実装した。

4 おわりに

今回、UI 構築ツール AutoView 上にコールバックルーチン作成負担軽減機能をインプリメントした。これにより AutoView の機能が向上し End-User の GUI アプリケーションの作成負担の軽減をはかることがさらに期待できる。今後は従来からあるこのようなツールとの比較および検討を行い、生産性の比較検討を行なうつもりである。

参考文献

- [1] 吉山、笠原「AutoView:エンドユーザを対象とした UI 構築ツール」,平成 4 年電気学会電子・情報・システム部門大会,B-2-6
- [2] Dan Heller, XView Programming Manual, O'Reilly & Associates, Inc (1990)