

ペンウィンドウシステムのためのプログラム開発支援系

2G-2

平井孝史, 河又恒久, 早川栄一, 並木美太郎, 高橋延匡

(東京農工大学)

1. はじめに

ウィンドウシステムにおけるアプリケーションプログラム (AP) の開発は、従来のシステムのものとは比べ、画面表示に関する設定とユーザインタフェース (UI) の多様化にともなって、困難かつ面倒なものになっている。

現在、我々の研究室においては、ペンコンピュータを指向したペンウィンドウシステムである“未” [1] が稼動している。“未”上で動く AP の開発は、従来のウィンドウシステム上で動く AP の開発時における問題点に加え、UI の設計など、ペンコンピュータであることの問題点も挙げられる。

本稿では、ペンウィンドウシステムのためのプログラム開発支援系の設計方針と設計について述べる。

2. 設計方針

(1) 「紙とペン」の感覚で作業をする

紙とペンによる作業は人間が今まで行ってきた自然な行動である。ペンコンピュータは入力デバイスがペンであるので、そのような「紙とペン」に近い自然な感覚で作業ができると考えられる。ペンウィンドウシステムのプログラム開発時においても「紙とペン」による自然な感覚で作業が行えるようにする。

(2) 入力操作にペンを用いる

「紙とペン」の感覚で AP の作成を行うことができるようにしたいため、ペンを用いて作成環境の入力操作を行う。例えば、直線を引いたり、筆点列データを入力したりする場合である。

(3) ジェスチャを扱う

ジェスチャはペンコンピュータ特有のインタフェースであり、本システムでもこれを扱う。

(4) 視覚的な作成環境にする

画面に何か表示させるとき、その位置などは実行してみなければわからない。画面表示に関することは、設定時に実際に表示させて、それを直接編集でき、それがそのまま実行時に反映されるものがあるとよい。そこで、画面表示に関して、視覚的に作業が行うことができるようにする。

3. 設計

3.1 画面設定

画面設定には、どんなウィンドウをどこに表示するかというウィンドウ設定と、ウィンドウに実際に表示する部分（これをキャンバスと呼んでいる）に何を表示するかというキャンバス設定がある。これらの設定は、プログラムのソースコードとして記述され、コンパイルし実行しなくては目に見ることはできない。またその設定値の決定は、勘に頼ることがほとんどであり、その結果、自分の思っていたものと違うことが多い。画面設定は、勘に頼ることなく、目で見て設定できるようなものがあるとよい。

そこで、ウィンドウ設定は実際にウィンドウを操作変更していくことによって行い（図 1）、キャンバス設定は実際にキャンバスに描画オブジェクトを配置していくことによって行う。つまり、視覚的に画面設定ができるようにする。

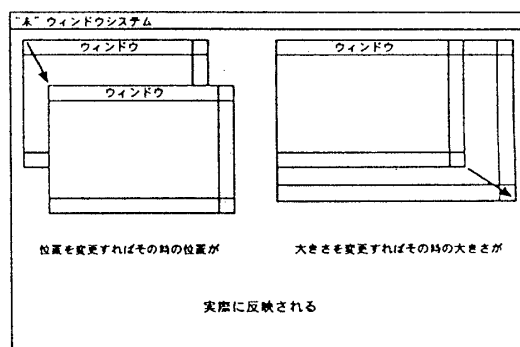


図 1 ウィンドウの設定

またこのとき、ウィンドウの操作や描画オブジェクトの配置はペンによって行い、描画オブジェクトのコピーや削除などのキャンバス設定における編集はジェスチャによって行う。ジェスチャによる編集は、メニュー選択のコマンド式のものに比べ、ステップ数がないので煩わしさが少ないと考えている（図 2）。

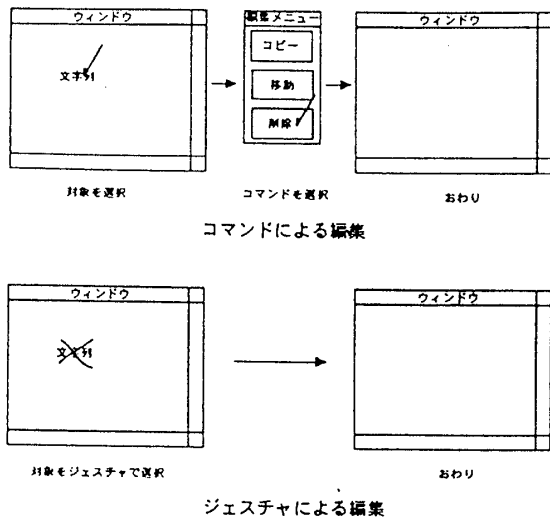


図2 コマンドによる編集とジェスチャによる編集

3.2 イベント

“未”は、プログラミングスタイルとしてイベント駆動方式を採用している。そのため、キャンパスのある領域にあるイベントが起きたらある関数を実行するといったことを設定するイベント登録が必ず必要になる。またこのとき、イベント領域は目に見えないため、実際にはイベント領域と同じ形で同じ大きさの描画オブジェクトをイベント領域にあわせて表示し、イベント領域を可視化することが多い。つまり、一つのイベント登録に関して、イベントの種類、イベント領域、イベントが起きたとき実行される関数名と、必要ならばそれを可視化するための描画オブジェクトの三つもしくは四つのことを設定する必要がある。

これらの設定はプログラムのソースコード上でやっているため、イベント領域とそれを視覚化する描画オブジェクトに関しては、画面設定のところでも述べたものと同じ問題が生じ、さらにこの二つは常に同じ位置、同じ形、同じ大きさである必要があるため、常に対応をとっていなければならない問題が生じる。

また、イベントの種類には様々なものがあり、各々が特有の設定値を持っている。これらの設定値を一つ一つ正確に覚えているのは大変であり、また覚えていなくとも AP 作成のたびにマニュアルを参照するのは効率が悪い。

そこで、イベント登録に関しても画面設定同様視覚的に行えるようにする。イベント領域は、それを配置していくことによって視覚的に設定できるようにし、イベントの種類の設定は、実際にイベント領域にそのイベントを起こすことによって行う。実行する関数名は、ソースコードが存在する場合はソースコードを表

示したところからペンで関数名を選択することによって設定し(図3)、ソースコードが存在しない場合は関数名をシステムが自動生成するかもしれないがユーザーが入力する。

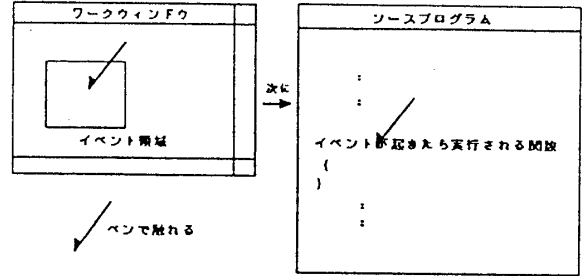


図3 実行する関数名の設定

3.3 ソースコードの出力

作業してきた結果は C のソースコードとしてテキスト形式で出力する。そうすることで、作業終了後流用が利くと考えたためである。次のように生成される。

- ・ウィンドウの設定は、そのウィンドウに関して、一つの関数とする
- ・キャンパスの設定は、そのキャンパス (ウィンドウ) に関して、一つの関数とする
- ・イベント登録は、登録したキャンパス (ウィンドウ) に関して、一つの関数とする

つまり、各設定を一つの関数にしたものである。各設定が独立した処理として論理的に独立しているため、そのまま素直に分ければわかりやすいと考えた(図4)。

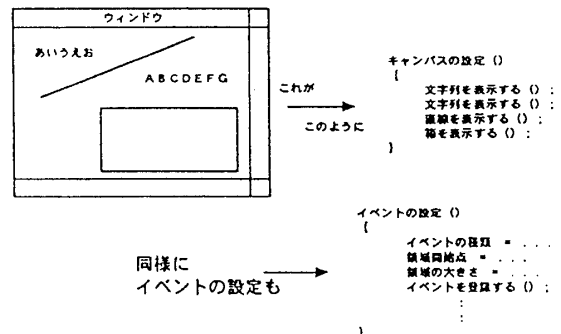


図4 出力するソースコード

4. おわりに

本稿では、開発支援系の設計方針と設計について述べた。今後は、プロトタイピングを行う予定である。

参考文献

[1]河又 他：表示一体型液晶タブレットを用いた“未”ウィンドウシステムの設計と実現，情報処理学会ヒューマンインタフェース研究会，45-17，1992