7K−1　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

河又恒久、宮島靖、早川栄一、並木美太郎、高橋延匡
(東京農工大学　工学部　電子情報工学科)

1．はじめに

表示一体型液晶タブレット（以下液晶タブレット）は、液晶ディスプレイに透明タブレットを重ねたデバイスであり、入力は、スタイラスペン（以下ペン）を使用して行われる。液晶タブレットは、入出力を同じ場所で行うため、紙とペンの感覚で計算機を扱えることが最大の特徴である。

したがって、液晶タブレットでは、紙とペンを使用するのと同じユーザインタフェース（以下UI）のアプリケーションプログラム（以下AP）が作成できる。

本稿では、マルチウィンドウ環境で、APがペンの入力を扱えるような「未」ウィンドウシステム（以下「未」）の設計と実現について述べる。

2．従来のシステムの問題点

従来のウィンドウシステムは、入出力デバイスとして、ビットマップディスプレイとマウスを使用したものが大半である。マウスは、画面上のマウスカーソルを間接的に移動させて入力するデバイスであり、フリーハンドで文字や図形を入力するのは非常に困難な作業である。このため、マウスは、主にポインティングに使用されている。

これに比べて、液晶タブレットでは、直接ペンを出力画面に触れて入力するため、紙とペンを使用している感覚で、文字や図形を描画できる。

実際に我々の研究室では、原稿用紙を模範したウィンドウにペンで入力できる原稿用紙ワープロ[1]、ペンを用いて、フリーハンドで描いた図形を整形する図形入力システム[2]などについて研究を行っている。これらは、ペンを使った入力でなくては困難なものである。

ペン入力で得られた筆点列は、対象とするウィンドウが、文字入力のウィンドウなら文字を表すし、図形入力のウィンドウなら図形を表す。この入力の多様性が、マウスにはない特徴である。

マルチウィンドウ環境でペン入力を扱えるようにするには、従来のマウス用のシステムをペン用に拡張するアプローチが考えられる。しかし、この場合は、マウスで行えることをペンで代行しただけになり、上で述べたペンの多様性に対応できず、ペンをポインティングにしか使用できない。また、ウィンドウデザインなどもマウスでの操作用に作られているため、ペン用に変更する必要がある。これらの問題は、初めからペンを指向したウィンドウシステムを作成することで解決する。

したがって、我々は、液晶タブレット用に「未」ウィンドウシステムを作成することにした。

3．「未」ウィンドウシステムの設計方針

液晶タブレットでは、マウスのシステムにないUIのAPを作成できる可能性がある。その一例として、ジェスチャが挙げられる。例えば、シェル[3]において、異なるディレクトリ間でファイルを移動する場合、マウスでは、ドラッグしてアイコンをウィンドウまで引っ張っていくが、ペンでは、アイコンを囲んで、矢印で移動先を指定することができる（図1）。

このように、ペンでは、マウスのシステムと異なるUIを作成できるため、ウィンドウシステムのUI管理も従来のものとは全く異なる。例えば、メニューはそれほど使われなくなるだろうし、ダイアログは、手書き入力が主体となるだろう。しかし、現在、液晶タブレット用のUIについて、最適なものが存在していないため、これから、試行錯誤して最善のものを追求していく必要がある。したがって、「未」では、ペンを使用したUIを研究するための基盤となることを方針とする。この方針をもとに、「未」は、ウィンドウカーネル（以下カーネル）とユーザインタフェースマネージャ（以下UIマネージャ）の二層に分ける。前者は、マルチウィンドウ管理を行い、後者は、ペン入力の解釈や、ウィンドウの操作体系などのUI管理を行う。これにより、マルチウィンドウ管理の部分を変更することなく、UI管理の部分を再構築することが可能となっている。

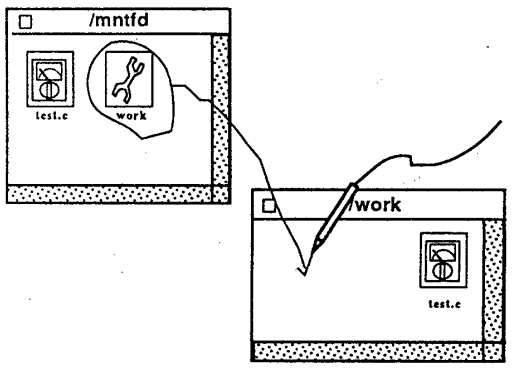

図1．ファイル移動のジェスチャ例

"HITSUJI" Window System on a Disply Integrated LCD Tablet
Tsunehisa KAWAMATA, Yasushi MIYAJIMA, Eiichi HAYAKAWA, Mitarou NAMIKI, Nobumasa TAKAHASHI
Tokyo University of Agriculture and Technology

4．「未」ウィンドウシステム初版の設計

カーネルについては、すでに設計、実現した［4］。ここでは、UIマネージャの設計について述べる。

4．1 ウィンドウデザインと操作系

「未」初版では、ウィンドウデザインは、図2のように固定とした。デザインは、ウィンドウのオープン時につけるかつけないかを指定できる。

また、ウィンドウの操作は、図3のように、移動、リサイズ、内容のスクロール、クローズを行える。

これらは、マウス用のシステムを模範しているが、将来は、液晶タブレット用のデザインを研究し、採用する。

4．2 入力データ

APへ渡る入力データは、1ストローク（ペンで描き始めてから終わるまでの筆点列）とする。ペンでの入力は、マウスのように1点ではなく、1ストロークを単位としている。したがって、APへ1点1点渡すよりも、1ストロークずつ渡したほうがよいと考えた。

また、筆点列は、開始点だけを通常の座標値とし、あとは、開始点からの差分座標とする。これは、ウィンドウシステムに複数の座標系があり、これらの変換の際、有利であると考えたからである。

入力データは、次に述べるジェスチャを除いて、開始点の存在するウィンドウに渡す。

4．3 ジェスチャの扱い

普通のデータ入力とジェスチャは、ウィンドウシステムで区別しなければならない。なぜなら、例えば、あるウィンドウ上で×を描いた場合、その中に×を描いたのか、何かを削除しようとしたのかAPでは判断できないからである。

ウィンドウシステムで区別する場合も、自動的に判別するのは、ほとんど不可能である。そこで、ウィンドウシステム内にモードを設けることにする。「未」初版では、ペンの横にあるボタンを押すか、ジェスチャを示す領域を選んだとき、ジェスチャモードとする。

ジェスチャモードの場合、入力データは、すべて同じ場所で管理しなければならない。これは、システムでジェスチャの一貫性を保つためである。今回は、シェルがこれを行う。

4．4 認識系の扱い

現在、我々の研究室では、手書き文字認識、手書き図形認識、手書き数式認識の研究が進められているが、ウィンドウシステムを含めたAPと認識系のインタフェースが決定していない。そのため、APは、ウィンドウシステムから筆点列をもらい、それを自分で認識するか、各認識系にかける必要がある。「未」初版では、APが認識系へデータを渡しやすくするため、認識系で使用するストロークと同じデータ構造を採用した。

5．おわりに

「未」初版は、カーネル、UIマネージャとも、C言語で作成され、OS／omicron上で稼動している。カーネルは、10000ステップ、UIマネージャは、5200ステップとなっている。

「未」初版により、APがペン入力を扱うための基盤ができた。これからは、認識系を含めた、ウィンドウシステムとAPとのインタフェース（プログラミングインタフェース）を決定する必要がある。

参考文献

［1］曽谷他：“手書きユーザインタフェース”，情処学第30回プログラミングシンポジウム報告集，pp．1－10，1990

［2］風間他：“手書きインタフェースの研究　～　その図形への応用　～”，情処学第42回全大，2P－5

［3］宮島他：“ビジュアルインタフェースとテキストインタフェースを統合したシェルの設計”，情処学第37回ヒューマンインタフェース研究会報告集，37－5，1991

［4］河又他：“ユーザインタフェース研究用ウィンドウシステム「未」の設計と実現”，情処学第52回オペレーティングシステム研究会報告集，52－6，1991

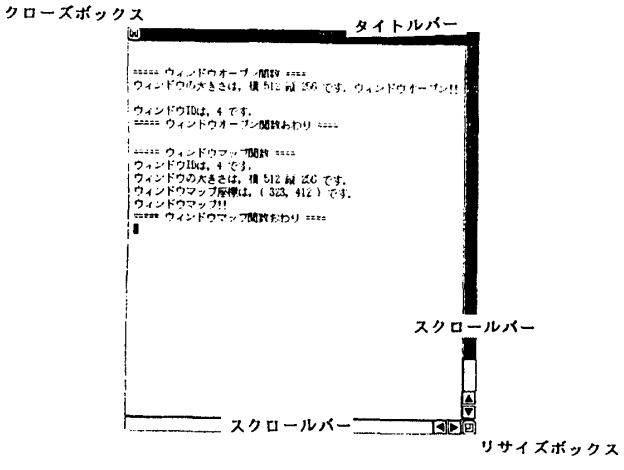

図2．ウィンドウのデザイン

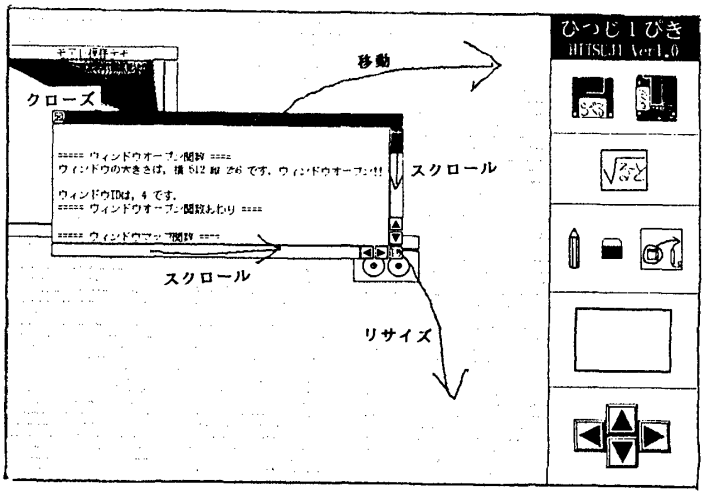

図3．ウィンドウの操作