

## 表示一体型タブレットの接続に対するOSの機能拡張の一方式

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

表示一体型タブレットを用いたヒューマンインタフェース (HI) 指向のシステムを構築する場合, (i) ペンの操作方式が単一でないこと, (ii) 扱う対象の管理形態が多様になること, (iii) 応答性能を向上させることに対応しなければならない. そのため, 我々は次の特徴を持った OS アーキテクチャの考察を行った.

- (1) 抽象的な枠組みとして, オブジェクトの概念の導入
- (2) 実現方式として, システムに対する機能拡張

本報告では, 我々が研究・開発を行っている OS/omicron の, HI 研究における設計哲学と, 具体的な実現の一手法について述べる. 特に, 表示一体型タブレットの応答性を向上させる, OS とハイパ OS に対する機能拡張の実現について述べる.

## A Method of Operating System Enhancement for Display Integrated Tablet

Eiichi Hayakawa, Mitarou Namiki and Nobumasa Takahashi

Department of Computer Science,  
Tokyo University of Agriculture and Technology,

Koganei-shi, Tokyo, 184 Japan

This paper describes a design principle for human interface (HI) oriented OS architecture, and enhancement for a display integrated tablet on OS/omicron. Building HI oriented system with a display integrated tablet imposes (i) various interpretations of pen action, (ii) various management styles of objects and (iii) immediate response. Therefore, the OS architecture employs these features:

- (1) object oriented approach for abstraction
- (2) systems enhancement for implementation

Implementation of immediate echo-backing in Hyper-OS and OS is described in detail.

## 1. はじめに

かつてのコンピュータシステムでは、コンピュータの製造コストが高いために、人間の生産性よりも、コンピュータのパワーをいかに無駄なく利用するかがコンピュータシステム的设计者に課せられた課題であった。

しかし、今日では、半導体技術の進歩によって、高性能のコンピュータを安価かつ、大量に生産することが可能になった。それに対して、コンピュータを使う人の人件費は上がり続けている。そして、使う人の時間をどう省くことができるかがコンピュータシステムやその上で動作するアプリケーションの課題となってきた。

このような課題を解決するために、ヒューマンインタフェースに関する研究が行われている。我々の研究グループの中でも、手書きインタフェースに着目して研究を行っている [1]。この研究では、入力デバイスとして、表示一体型タブレットを用いている。表示一体型タブレットは、優れたインタフェースを提供するが、対話性という点でマウスやキーボードと異なっているために、それをサポートするコンピュータシステム的设计や実現に対して、新しい设计哲学を必要とする。

本報告では、我々の HI 指向のコンピュータシステム的设计哲学について述べる。また、表示一体型タブレットを効率よく用いるために、OS に対して行った機能拡張について述べる。

## 2. OS/omicron のヒューマンインタフェースシステム

### 2.1 研究の目標

OS/omicron のユーザの中には、ヒューマンインタフェース研究として、手書きインタフェースシステムの研究 [1, 2, 3, 4] を行っている。これは、扱う対象として文字、図形、数式などを含んだ文書作成システムである。このシステムの入力デバイスとしては、表示一体型タブレットを用いる。このデバイスは、紙に近いインタフェースを提供することができる。我々は、このようなデバイスを用いたアプリケーションを効率よく作成、実行することが可能なコンピュータシステムの構築を目標に設定した。

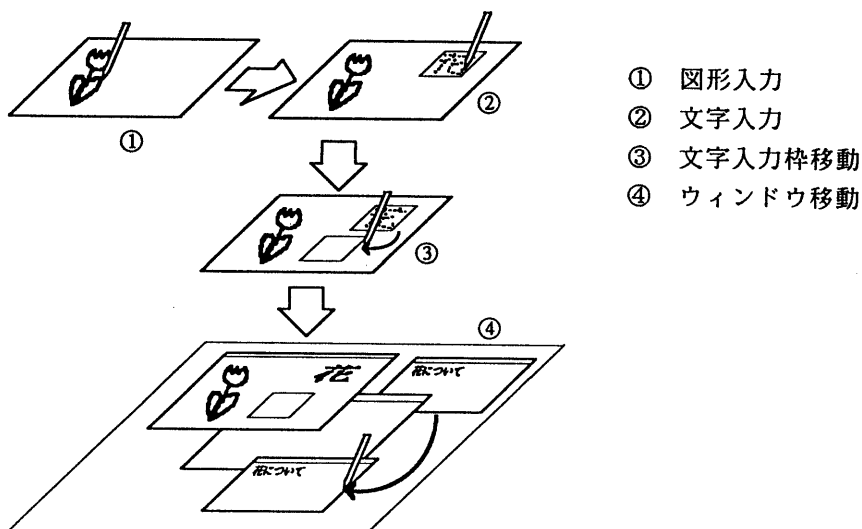


図 1 表示一体型タブレットを用いたアプリケーション例

## 2.2 表示一体型タブレットの問題

このようなシステムで、アプリケーションを動かすときの問題について、例を用いて考察する。

ユーザが、表示一体型タブレット上でペンを用いて、絵を書くことを考えてみる(図1-①)。この場合、紙で書くのと同等のインタフェースにしたい。つまり、書いたものが、ディスプレイにすぐ反映しなければならない。人間は、「紙と筆記具」の手段に長い間慣れしただけであるために、ここでエコバックが遅ければ、そのデバイスは使い勝手の悪いものになってしまう。

次に、ユーザは文字を図と同一画面上で入力したい(図1-②)。ほとんどの図には文字が混在しているためである。文字入力に対して、我々は手書き文字認識システムを採用する事を考えている。これで、システムからキーボードをなくし、システム全体を小型化することが可能である。手書き文字認識システムそのものに対しては、認識に必要な辞書の管理がある。辞書検索が認識系の速度を向上させるファクタとなる。また、誤認識した文字の扱いや、学習の問題がある。また、このような認識システムに対してどのようなアプリケーションインタフェースを設定すべきかという点も問題となる。

また、これらの対象物に対して、編集操作を行いたいことがある(図1-③)。例えば、文字認識の枠を移動することや、書いた文字を消去する、挿入する、図の一部を削除するなどである。このような操作もペンで行いたい。そのためには、対象によってペンの持つ意味を変えていく必要がある。ユーザから見れば、ペンの操作は同じであるが、図と文字では消す手続きは異なっている。また、図と手書き文字では、エコバックの方法が異なってくる。手書き文字では、筆点列をエコバックすればよいが、図では、筆点間を補間する必要が出てくる。つまり、表示一体型タブレットでは、操作は多様であり、システムで用意する単一の操作系ではまかないきれない。

さらに、このようなアプリケーションが、ウィンドウシステム上で複数動作する場合、例えば、手書きワードプロセッサと、図形入力システム、出力レイアウト指定システムでは、ペンの持つ意味は、ウィンドウの移動、ウィンドウ内のアプリケーションの操作、入力など、多様な形態をとる(図1-④)。つまり、タブレットからくるデータは同じだが、扱う対象物によって、その解釈や、操作形態は大きく変わってくるものと考えられる。また、消去や挿入といった編集は校正記号のような記号をタブレットで書くこと(ジェスチャーと呼ばれる)で行いたい、これらはアプリケーションやユーザの好みによって、その形態が異なってくる。

また、ペンでメニューやダイアログの操作などを行うことがあるが、これらは、頻繁に使用されることから、ライブラリとして一貫したものにしたい。また、ダイアログ中では、手書き文字認識システムを用いた文字入力なども行われることから、ダイアログは、文字、図、ジェスチャーなどいくつかの入力対象と、それらへの操作の混合した形態になる。このようなダイアログやメニューに対して、どのようなアプリケーションインタフェースを提供するか、という点も問題になってくる。

このようなアプリケーションを動作させる場合の問題点を次にまとめる。

- (1) ウィンドウ、文字、図など扱う対象によって、ペンの操作方式が変化する。
- (2) 扱う対象が多様化しているために、それらの管理部分が多様化、複雑化する。
- (3) 応答性が使い勝手に影響を与える。

## 2.3 OS/omicrontにおけるユーザインタフェース研究の設計方針

このようなシステムに対して、従来のウィンドウシステムでは対応できない。入力デバイスとして、マウスとキーボードを仮定し、認識系に対する対応がないこと、アプリケーションによって意味が大き

く変化するペンの性質を考慮していないためである。また、従来の OS では、デバイスドライバをカーネル中に静的にリンクしているために、アプリケーションによって、デバイスの処理を特化することは難しい。入出力に関しては、その意味がアプリケーションによって多様に変化することから、OS の管理下において、仮想化することは難しい。

我々は、表示一体型タブレット指向のウィンドウシステム [5] の研究、開発を行っているが、2.2 で挙げた HI のために要求されている機能を、すべてウィンドウシステム内で実現するのは難しい。ウィンドウシステムは OS の実行環境で動作するために、タブレットのエコーバックのように時間制約を持った処理に対して管理することは難しいためである。ウィンドウが扱う資源以外のもの、例えば、ファイルや、プリンタといったデバイスに対しても操作が多様化する事を考えた場合、計算機資源に対するアクセスを OS レベルで統一化しておくことが、ヒューマンインタフェースを向上させる要因になると考えた。

以上の考察から、ヒューマンインタフェース指向の OS を構築するための設計方針として、次の二点を定めた。

(1) OS アーキテクチャとして、操作対象への統一的なアクセスを行うために、オブジェクトの概念を導入すること。

統一化のための抽象的な枠組としてに、各対象物をデータと対象物ごとの操作手続きを含んだオブジェクトとして扱う。オブジェクトの操作インタフェースを統一することによって、同じ操作名で動作が異なるインタフェースを提供することが可能である。OS は、ウィンドウシステムやデバイスドライバなどが、資源をオブジェクトとして操作、管理し、ユーザに対して提供するための枠組みを与える。例えば、図 1 で手書きワードプロセッサウィンドウと、図形入力ウィンドウでは、ペンデータの入力という操作は同じだが、エコーバックの動作やペンデータの解釈が異なる。

(2) 実現方式として、アプリケーションを効率よく実行させるために、デバイスドライバを含めたシステムに対して機能拡張が行えること。

アプリケーションの多様性に対応しつつ、ユーザに対する応答性を確保するために、OS の機能として、デバイスドライバに対して、アプリケーション指向の拡張機能を導入する。

本報告では、実現の一方式として、表示一体型タブレットを動かすための OS の機能拡張を行った。

### 3. 表示一体型タブレットのための OS の機能拡張

#### 3.1 目的

表示一体型タブレットは、液晶による表示部と、タブレットとペンを用いた入力からなり、タブレットが透明になるので、タブレットを通して表示部を見ることが可能な点を特徴としている。これによって、ユーザに「紙にペンを用いて書く」という動作に近いインタフェースを提供することが可能である。しかし、このデバイスをシステムでサポートする場合、システムのオーバーヘッドから、入力データを出力に反映させるのは難しいと考えられる。そこで、システムを拡張して、実用的な速度で表示一体型タブレットをサポートすることを目的とする。

### 3.2 表示一体型タブレットの使用形態

表示一体型タブレットの使用形態としては、

- (1) ペンで書いた部分の座標データを入力する。
- (2) その座標データを元に、データの表示を行う。
- (3) 座標データに付属するデータ、例えば、そのときの時間などを入力する。

の3点を設定した。特に、利用者に対して「紙とペン」のインタフェースを提供するためには、ペンで書いた部分のデータ表示を高速に行う必要がある。

ここで、我々は、システムの問題を明らかにするために、拡張を次の二つの方法で行った。

方法1：ソフトウェア構成に変更を加えずに、接続を行う。

方法2：方法1で発生した問題を解決するために、ソフトウェアを拡張する。

この拡張は、当研究室で開発した OS/omicon 第2版システムで行った。このシステムは、ハードウェア層にハイパ OS「江戸」と呼ぶ仮想マシンを搭載し、OS/omicon 第2版はこの上で動作する。表示一体型タブレットは、タブレット入力として RS-232C を用いているために、システムを変更せずに接続することができた。

### 3.3 タブレットの接続

図2に示した構成で、ペンで書いた部分をそのまま画面へ表示するプログラムを動かした。このとき、ペンで書いたエコーバックに遅れが生じた。これは、ペンで書いた後を、エコーバックの筆点が追いかけて行く形になり、使いにくかった。この原因は、次の三つであると考えられる。

(1) 描画までのコンテキストスイッチに時間がかかる  
アプリケーションがエコーバック処理を行うには、ハイパ OS、OS の層を通らなければならない。このときのコンテキストスイッチによるオーバーヘッドがかかっている。

(2) 画面の再描画に時間がかかる  
ハイパ OS では、OS ごとに仮想フレームメモリ (Virtual Frame Memory 以下 VFM) と呼ぶ表示用のフレームメモリを持っている。これは、各 OS に対して個々の表示空間を与える。ハイパ OS では、VFM に対して書込みが発生すると、フレームメモリに対して再書換えを行う。これのオーバーヘッドがある。

(3) 筆点列の割込みをすべて待ち行列に確保している

ハイパ OS では、ハードウェア割込みを必ずエミュレーションするという方針から、割込みを待ち行

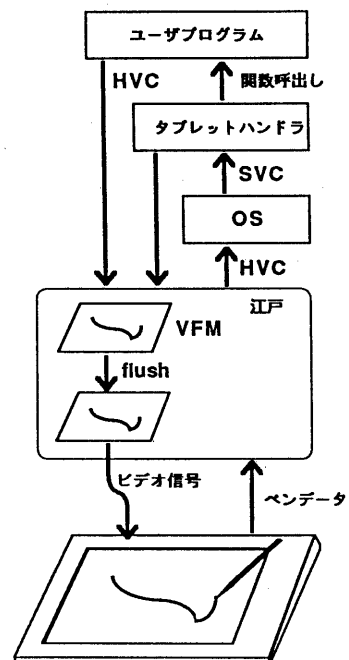


図2 タブレットの接続形態

列に格納して、必ず発生させるようにしている。そのために OS への割込みエミュレーションよりも、ハードウェア割込みが多く発生した場合、割込みが遅れて発生することがある。

### 3.4 ハイバ OS と OS の機能拡張

前節の方法では、表示一体型タブレットは既存の入出力を用いているために、ハードウェア層の上にあるハイバ OS 層が、ペンのデータや画面データに適した処理を行っていない点が問題である。そこで、ハイバ OS と OS のインタフェースを表示一体型タブレットに適した形に拡張した。その全体構成を図 3 に示す。

#### (1) ハイバ OS 内部でエコーバックを行う

アプリケーション層において、エコーバックの制御を行なうと、ペンデータの出力、描画、そして、VFM の再描画のオーバーヘッドがかかる。また、ペンデータは割込みによってハードウェアから通知されるが、ハイバ OS 層で割込みをエミュレートしているために、そのオーバーヘッドも加わる。そこで、ペンデータのエコーバックが必要な場合は、ハイバ OS 層で直接エコーバックを行うように拡張する。また、メニュー選択のように、ペンをポインティングデバイスとして使う用途もある。そこで、エコーバックの on/off をユーザ側から指定できる SVC を用意した。

#### (2) フレームメモリへの直接描画を行う

画面の再描画に問題があるために、VFM へ直接描く方法では転送のオーバーヘッドがあるために、遅くなると考えられる。エコーバックの応答性の問題から、ハードウェアのフレームメモリへ直接描画を行う手法を採用した。VFM と フレームメモリ間のコピーによって、データをつぶしてしまうという点については、アプリケーション側で再描画を行う領域を限定することによって解決した。

#### (3) 筆点データをブロック単位で転送する

表示一体型タブレットから送られる筆点データは、必ず 5 バイト単位で構成されている。このデータは通常、1 バイトずつ OS 側のリングバッファに渡されるが、これでは効率が悪い。そこで、OS とハイバ OS の間のリングバッファを 5 バイト単位にして、データ受信の仮想割込みも 5 バイト単位にした。

またデータエラーリカバリもハイバ OS 層で行うようにした。

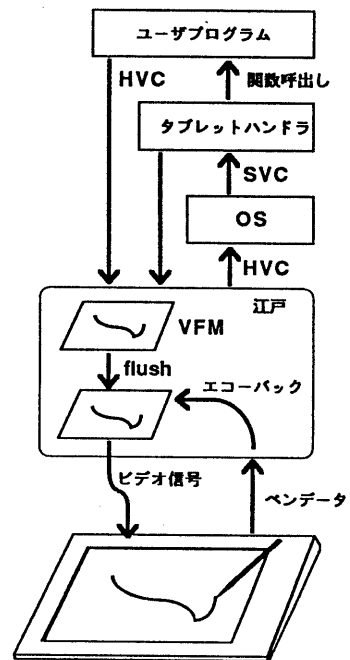


図 3 拡張時のタブレットの接続形態

これに対応して、OS の RS-232C ハンドラを次のように変更した。

#### (1) リングバッファのデータをすべて読み込む

応答性を上げるために、一回の SVC でリングバッファにあるすべてのデータを得るようにした。

(2) リングバッファのデータをすべて捨てる

メニューやダイアログを実現する場合、あるデータを読み込んだ時点で、残りのデータをすべて捨てる必要がある。この機能を追加した。

### 3.5 実現

前節の機能を実現するために、ハイバ OS の RS-232C ハンドラ内部にエコーバックルーチンを組込んだ。また、外部の座標と、内部の座標を一致させるために、次の SVC 命令をハイバ OS に追加した。

```
edo_tablet_set_origin(int タブレットx, int タブレットy, int VFMx, int VFMy,  
                      int スケールファクタx, int スケールファクタy);  
edo_tablet_echo(int フラグ);
```

edo\_tablet\_set\_origin SVC は、タブレットと液晶表示部の原点合わせを行う。これは、タブレットと液晶の解像度が異なるために必要である。原点合わせは、VFM 座標に対応するタブレットの位置を与えることで決定する。これによって、アプリケーションによるエコーバックと、システムエコーバックを合わせることができる。

また、edo\_tablet\_echo SVC は、エコーバックの on/off を指定する。

OS に関しては、前節の機能を RS-232C ドライバに組込んだ。また、アプリケーションプログラマの負担を減らすために、言語Cのライブラリ関数を用意した。

## 4. 考察

我々は、第3章に述べた形態で、表示一体型タブレットの応答性を上げた。応答性を上げる手法については、[6] では、認識とエコーバック部分と、アプリケーション部分をプロセスとして分離し、ネットワークを介してプロセス間通信を行いつつ、負荷分散を行う方法を提示している。

これに対して、我々は、マルチプロセッサを用いることを考えている。この場合、プロセッサの性質としてアノニマスにする必要はない。むしろ、タブレットのエコーバック専用プロセッサや、文字認識専用プロセッサとして扱い、専用プロセッサの集合体としてシステムを構築することを考えている。このように作成することによって、ネットワークを用いずとも、システムのパフォーマンスを上げることが可能であり、マシンをポータブルにすることができる。システムのパーソナル化に対して、有効なアプローチであると考えられる。

本報告では、ハイバ OS 内のデバイスドライバに対して、関数レベルで変更を行った。これから、タブレットを用いたアプリケーションが増えてくるに従って、ユーザ定義のエコーバックを登録したり、それらのルーチンを動的に変更したりするようになると思われる。このようなドライバへの拡張は、扱う粒度によって、その実現形態を変える必要がある。速度が必要であれば、関数レベルでの拡張も必要となる。また、それ以外にも、プロセス、プロセッサ、マシンといったレベルでの拡張方法が考えられる。これらを、拡張機能に合わせて使い分けていく必要がある。

また、本方式のような関数レベルでの機能拡張では、組込むモジュールのデバッグの問題がある。これに対しては、デバッグ時には、ユーザレベルで行い、そのルーチンの呼出しは、RPC のメカニズムを

用いる方法がある。プロセスが異なっている場合は、速度的な問題は生じるが、メモリ空間が保護されているために、システムを破壊することはない。デバッグが完了した時点で、デバイスに対して組込めば、組込みモジュールの安全性を高めることができる。また、組込み後のデバッグに関しては、ハイパ OS を用いた OS デバッグ [7] を用いることが可能である。

この場合、ドライバをオブジェクトとして扱い、インタフェースを統一することによって、内部の実現形態は異なっても、モジュールとして同等に扱うことができる。ドライバの変更はモジュールの入替えだけであり、アプリケーション自体の変更がほとんど不要になる。

## 5. おわりに

本報告では、OS/omicron におけるユーザインタフェース研究の設計方針について述べるとともに、表示一体型タブレットを接続するための、OS の拡張の一方式について述べた。現在、このシステム構成で文字認識システム JOLIS-2E、図形入力システム、打碁システムが稼働している。

本システムは、表示一体型タブレットに関して、プリミティブな機能しか用意していない。第 2 章で述べた機能を実現するためには、システムの改良が必要である。また、本システムでは、ハイパ OS として「江戸」を用いたが、「江戸」はデバイスドライバの拡張性に問題がある。我々は、これに代って軽量のコンテキストを持ち、ドライバを動的に追加可能なハイパ OS である「礎」を開発した [8]。今後は、OS/omicron を「礎」に移植し、その上で、OS の動的な機能追加に関する実験を行う予定である。

## 謝辞

手書きインタフェース研究の立場からご意見を頂いた東京農工大学工学部電子情報工学科 中川正樹助教、曾谷俊男、福島英洋、村瀬敦史、風間信也の各氏に感謝する。

## 参考文献

- [1] 曾谷他：手書きユーザインタフェース，第 30 回プログラミングシンポジウム，1990.
- [2] 曾谷他：オンライン手書き文字認識系の使い勝手に関する考察，第 32 回 プログラミングシンポジウム，1991.
- [3] 風間他：手書きユーザインタフェースの研究 ～ その図形への応用 ～，情報処理学会 第 42 回全国大会，1991.
- [4] 村瀬他：手書きによる数式認識システム，情報処理学会 ヒューマンインタフェース研究会資料，36-1，1991.
- [5] 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 (HITSUJI) の設計と実現，情報処理学会 オペレーティングシステム研究会資料，52-6，1991.
- [6] 荒井他：手書きヒューマンインタフェースの構築支援環境，情報処理学会 ヒューマンインタフェース研究会資料，35-7，1991.
- [7] 清水他：Hyper-OS による OS のデバッグ環境の研究，情報処理学会 第 42 回全国大会，1991.
- [8] 並木他：ビルディング・ブロック・システムのための共有ソフトウェアバス～礎～，情報処理学会 オペレーティングシステム研究会資料，51-2，1991.