

手書きヒューマンインタフェースの開発 (2)  
— アプリケーション構築支援環境 —

6G-2

荒井俊史, 福永泰, 正嶋博, 葛貫壮四郎  
(株)日立製作所 日立研究所

1. はじめに

ペンだけで全てのオペレーションができる、手書きヒューマンインタフェース(HI)の開発を進めている[1]。今回、UNIXワークステーションに接続可能な入力表示一体化ディスプレイを試作し、UNIX上で手書き文字/手書きイメージを活用したアプリケーション(AP)を作成する基盤を開発した。

手書きHIでは、紙と鉛筆に近い操作感覚が得られるので、文字/図形/イメージを自然に描くことができる。このため、ストロークの形そのものに、なんらかの意味を持たせることができ、それを活用することで、より快適なHIを実現できる可能性がある。例えば、手書き文字認識技術を用いれば、文字を手書きで入力する"手書きワープロ"のようなものも実現できる。さらに、原稿用紙に校正記号を書き込む要領でテキストを編集することも可能である。

しかしAPを作成する立場から見ると、手書きHIを実現するためにはストローク列の意味を認識する必要があるため、キーボード/マウスをベースとしたHIの実現に比べ、より多くの手間を必要とする。ここでは、UNIX上で実現した手書きHIの構築支援環境について報告する。

2. ソフトウェア構成

図1は手書きHI構築支援環境のソフトウェア構成である。手書き文字認識モジュールは、ストロークエコーの応答性を検討した結果、APとは独立したサーバプロセスとして実現した。1文字当たりの認識時間は数百ミリ秒(後述)であるが、それだけでもエコーが遅れると、利用者に大きな違和感を与えることになり、快適さが著しく損なわれる。認識モジュールをサ

ーバプロセスとし、APとサーバが並行動作できるようにすることで、一応の応答性を確保することができた。

また、認識モジュールをサーバ形式で実現したことで、メモリ領域などの資源の有効利用、パターン辞書などの統一的管理も可能になった。

3. ストローク認識サーバ (Sserver)

筆者らが開発を進めてきた手書き文字認識アルゴリズム/パターン辞書[2]を、UNIX上でSserverというサーバプロセスとしてインプリメントした。

Sserverは、クライアントから正規化済みのストローク列のデータを受け取り、それを文字認識した結果(漢字コード)を返送する。クライアントとの通信は、ネットワーク透過なソケットインタフェースを用いる。

認識性能については詳しく述べないが、認識率は、平仮名95%、漢字98%であり、実用上問題がないレベルまで達していると考えている。また、一文字当たりの認識時間(認識開始から結果が表示されるまでの時間)は、日立のワークステーション2050/32E(CPU:M68020, Clock:25MHz)を一人で使用している状況で200~500 msec程度であり、こちらも十分に実用に耐えるレベルと考えている。

4. ストローク処理ライブラリ (Slib)

APにストローク処理の基本機能を提供するため、Slibというライブラリを作成した。Slibは、タブレットからのデータ(ペン座標列)入力、ストローク認識サーバとの通信をサポートする。Slibに含まれる関数の一部を図2に示す。

文字を認識するための関数 SRecogStrokeData は、

アプリケーションプログラム

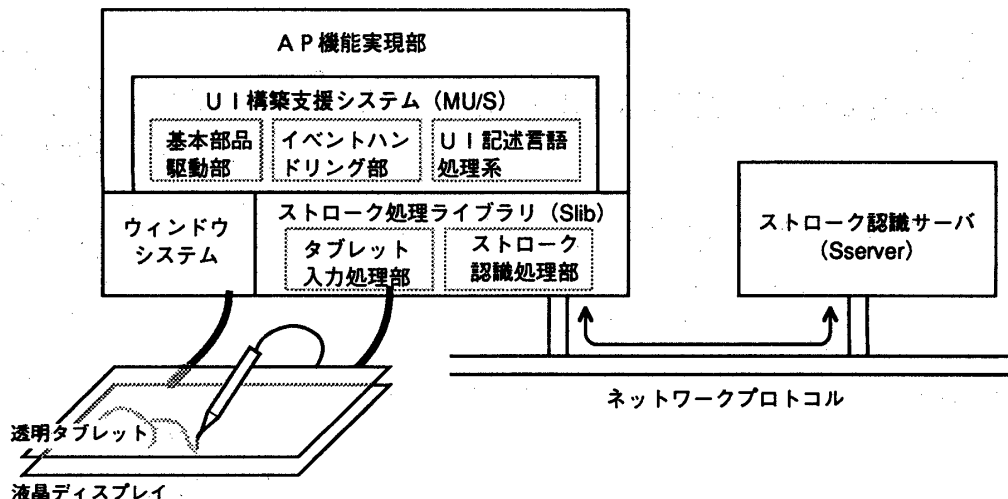


図1 手書きHIのソフトウェア構成

Development of Handwritten Human Interface System(2)  
- Application Developing Environment -  
Toshifumi ARAI, Yasushi FUKUNAGA,  
Hiroshi SHOJIMA, Soushirou KUZUNUKI  
Hitachi Research Lab., Hitachi Ltd.

\* UNIXオペレーティングシステムは、米国AT&T社ベル研究所が開発したソフトウェアであり、AT&T社がライセンスしています。

Sserver ヘデータを送った後、認識の終了を待たずに呼び出し側に制御を戻す。これは、エコーの応答性を考慮し、APがブロックされる時間をできるだけ短くするためである。認識結果は、認識の終了を確認した後、SGetRecogResult で読み出す。認識の終了は、SGetSserverFd が返すファイル記述子を使って調べられる（読み出し可能かどうかを select で調べる）。また、ペン入力があるかどうかは SGetTabletFd で得られるファイル記述子で調べられる。APは複数のファイル記述子について同時に調べられるので、ストロークを入力/エコーしながら、認識の終了も調べることができる。

ファイル記述子で入力を調べるのは、APへのインタフェースとしては比較的レベルなものである。敢えてそうした理由は、他の入力（マウス/キーボード、ウィンドウシステムからのイベントなど）との親和性を考慮し、それらと同時に入出力のチェックができるようにするためである。

5. ユーザインタフェース構築支援システム (MU/S)

Slib で提供される APインタフェースは比較的レベルなものであり、描画に関する支援もないため、手軽にAPを作るという訳にはいかない。そこで、AP作成の手間をより少なくするため、ユーザインタフェース (UI) 構築支援システム - MU/S - を提供する。

UIを構成するための基本的な部品（メニュー、アイコン、テキスト表示フィールドなど）は、あらかじめ MU/S に組み込まれているため、対象とする計算機の描画系について特別な知識がなくともUIを作成できる。また MU/S は、オブジェクト指向に基づきUIを記述するための言語処理系を備えており、画面上での構成部品の組み合わせや各種イベント（マウス入力、キー入力など）発生時の動作を容易に定義することができる [3, 4]。

図3の画面は、手書きHIで、テキストとイメージが混在したメールを読み書きしているところである。ユーザは、ペンでメニューを選択したり、文字やイメージを入力したりできる。このプログラムのUIは36個の部品で構成され、MU/S が提供する言語（約700行）で記述されている。MU/S を用いると、簡単な記述で色々なUIを実現できるために、使い勝手評価用のプロトタイプングなども簡単にできるようになった。

SInitialize	ストローク処理環境の初期化
SDestroy	ストローク処理環境の終了
SOpenTablet	タブレット使用開始
SCloseTablet	タブレット使用終了
SGetTabletFd	ペン入力用ファイル記述子を返す
SGetEvent	ペン入力のイベントを読み込む
SOpenSserver	認識サーバと接続
SCloseSserver	認識サーバとの接続を切る
SGetSserverFd	サーバ通信用ファイル記述子を返す
SNormalizeStrokeData	ストロークデータの正規化
SRecogStrokeData	認識開始（サーバにデータを送る）
SGetRecogResult	認識結果を読み出す

図2 Slibの関数のリスト (一部)

MU/S は日立のウィンドウシステム上で実現されているが、Xウィンドウ上で widget を用いたバージョンも作成中である。

6. おわりに

手書きHIの開発支援環境を開発した。今後は、それをベースとして手書きHIの使いがたの評価を進める予定である。例えば、文字入力領域の大きさ/位置/個数などが文字入力の効率や快適さに与える影響、校正記号のようなストロークの形によるコマンド入力の有用性などは、多くの実験結果を基に検討して必要があると考えている。

参考文献

- [1] 福永ほか：手書きヒューマンインタフェースの開発 (1) - ハードウェアに関する考察 -、情処第41回全国大会、1990年
- [2] 正嶋ほか：手書きマンマシンインタフェースの開発、信学技報 Vol.88 No.351、1989年
- [3] 谷ほか：メタユーザインタフェースを有するユーザインタフェース構築支援システム、情処論文誌第30巻 第9号、1989年
- [4] 荒井ほか：メタレベルユーザインタフェースを構築支援できるユーザインタフェース管理システム、情処研報 Vol.90, No.3, 1990年

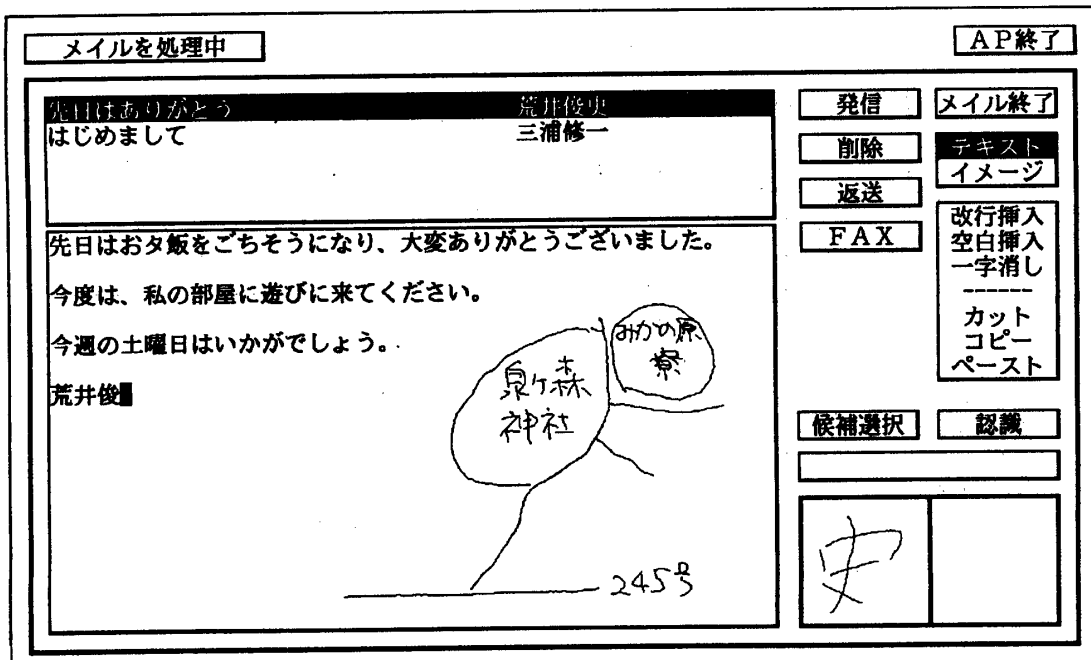


図3 手書きHIの実行例