

2K-1

OS/omicron におけるビジュアルとテキストを

統合したシェルの設計

宮島靖, 河又恒久, 小松徹, 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

OSの主たるインタフェースはコマンドインタプリタまたはシェルと呼ばれる。シェルにはキーボードから文字列を打ち込むことによって操作を行うテキストタイプのシェルの他に、Bit Map 指向のシステムにおいて、アイコンやウィンドウを主にポインティングデバイスによって操作を行うビジュアルシェルがある。この論文では双方のシェルの長所と短所を述べ、新たに設計したビジュアルシェルの特徴を中心に概略を述べる。

2. ビジュアルシェルとテキストシェルの利点と欠点

筆者が現在までに使用したことのあるシェルは以下の通りである。

・ビジュアルシェル

- (1) Finder (Mac OS, Macintosh)
- (2) SX-Shell (Human68k, X68000)

・テキストシェル

- (1) csh (UNIX, Sun-3)
- (2) COMMAND.COM (MS-DOS, PC-9801)

これらのシェルの使用経験から、それぞれのタイプのシェルの利点と欠点をまとめてみた。

	ビジュアルシェル	テキストシェル
利点	<ul style="list-style-type: none"> ・長い名前や名前を知らないファイルを扱うのが楽 ・任意の複数ファイルに対する操作が可能 ・深い階層のファイルを扱うのが楽 	<ul style="list-style-type: none"> ・打ち込みやすい簡単なファイル名はすばやく入力できる ・キーボードから手を放さなくてよい
欠点	<ul style="list-style-type: none"> ・ポインティングデバイスに手をのぼすのが面倒 ・アイコン指定のためにマウスをかなり動かさなければならぬ場合がある ・コマンド起動時の引数の指定が面倒 	<ul style="list-style-type: none"> ・長い名前や名前を知らないファイルは入力が大変 ・任意の複数ファイルに対する操作が一度にできない ・深い階層間を行き来するのが大変

表1 既存のシェルの利点と欠点

3. シェルの設計方針

(1) ビジュアルシェルとテキストシェルの両方を提供する

ビジュアルシェルの利点がテキストシェルの欠点になり、ビジュアルシェルでの欠点がテキストシェルの利点になっているのならば、ビジュアルシェルとテキストシ

ェルの両方をシステムに組み入れればよいのではないかと考えた。しかし、どちらのシェルを使えば楽に操作ができるかということは操作内容と操作対象によって異なる。

(2) ビジュアルシェルとテキストシェルの処理を連動させる

両方のシェルを同時に動かすとすると、その結果が両方のシェルに反映される必要がある。特にテキストシェルで操作を行った場合に、ビジュアルに反映されなくては困るだろう。例えば、ファイルをコピーするのに、ビジュアルにはそれが反映されていないのでは、一貫性がなく、誤操作の原因にもなる。

(3) 両方のシェルに統一の内部表現を定める

ビジュアルシェルとテキストシェルを連動させるために、操作の内容を両方のシェルに共通の内部表現で表す。そして、それぞれのシェルは内部表現を見てそれぞれのシェルの形態で画面に反映させるのである。

このような構造になっているため、もし他の形式のシェルをのせたいと思っても簡単にのせることが可能である。新たなシェルを作るには、ユーザからの入力を内部表現に直す部分と、内部表現を解釈してそのシェルの表現形式になおす部分さえ作ってやればよい。これによって、いろいろな形式のシェルを作って、ユーザインタフェース研究のための実験を行うことが可能である。

4. シェルの設計

4.1 操作対象

ビジュアルシェルで操作の対象となるものは、現段階ではファイルである。ファイルは扱うモデルとしては単純であり利用度が高く、初期段階の操作対象としては適当である。

ここでいうファイルとは、下に個々のファイルを持つディレクトリファイルと、ディレクトリの下にある個々のファイルの両方のことを指している。これらのファイルをこのビジュアルシェルでは、

- ・ディレクトリ、ディレクトリ下の個々のファイル → アイコン
- ・開いているディレクトリ → ウィンドウ

という対応付けを行っている。(図1)

ディレクトリを表すアイコンを開くと、新たにディレクトリウィンドウが開き、その中のファイルをアイコンで表示する。

4.2 操作内容

このシェルではアイコンを用いて以下のファイル操作が可能である。

- ・ディレクトリを開く

A Design of the Integrated Shell of Visual and Textual Interface
Yasushi MIYAJIMA, Tsunehisa KAWAMATA, Tooru KOMATSU,
Mitarou NAMIKI and Nobumasa TAKAHASHI
Tokyo University of Agriculture and Technology

- ・ディレクトリの移行
- ・ディレクトリの作成
- ・ファイルの移動
- ・ファイルのコピー
- ・ファイルの消去
- ・ファイルの名前変更
- ・ファイルの実行

4.3 シェルの全体構成

シェルの全体構成を図2に示した。主要な部分の簡単な説明を行う。

・イベント管理部

ウィンドウシステムから来る入力デバイスイベントを管理している部分である。イベントが発生した場合に、イベントの内容に応じて必要な処理先に制御を移してやる動きをする。

・ビジュアルシェル

イベント管理部から入ってきた入力データを解釈して、内部表現におおす解析部と、内部表現を解釈してそれをビジュアルに表現することをを行うビジュアル表現部がある。

・テキストシェル

イベント管理部から入ってきた一連のキーデータを解釈して、内部表現に直す解析部と、内部表現を解釈してそれを文字列としてテキストシェル画面に表示することをを行うテキスト表現部がある。

・シェルカーネル

シェルカーネルで解釈され内部表現に表されたものを保存している。そしてその内部表現に応じてファイルシステム管理部、タスクフォース管理部、ダイアログ管理部などを呼ぶ。

また、内部表現として表されたコードを用いて、ビジュアルカーネル内のビジュアル表現部とテキストカーネル内のテキスト表現部を駆動する。ビジュアルとテキストのシンクロの仕掛けはここにある。

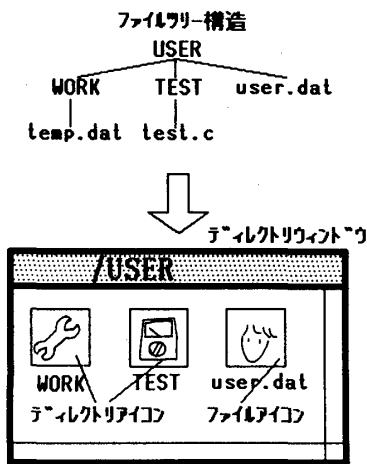


図1 ビジュアルへの対応付け

4.4 実現環境

今回作成するシェルは、研究室で使用している OS/omicon のユーザインタフェース構築の目的で設計している。よって、シェルの実現は OS/omicon 上で行う。

また、シェルはシェル自身でウィンドウを開き、その上で実行されるユーザアプリケーションもまたウィンドウを開く。このためウィンドウシステムは欠かせないものである。ウィンドウの提供には、“未”ウィンドウシステムを用いることにした。このウィンドウシステムもまた OS/omicon のユーザインタフェース構築のために設計されたウィンドウシステムであり、シェルを始めとする各種アプリケーションの動作環境の土台となる存在である。ビジュアルシェルでは、ウィンドウシステムからのイベントを受け取り、それらの処理を行った後、その結果を画面に反映させるためにウィンドウシステムを使用して描画を行う。

5. おわりに

今回作成するシェルは、その土台となるウィンドウシステムも含めて、ユーザインタフェースの研究のために設計した。よってこれらは、これらを利用して多種多様なインタフェースを構築してもらうための骨組みに過ぎない。

参考文献

- 1) Arthur Naiman: The Macintosh Bible, second edition, Japanese-language version, Gijyutsu Hyouron Inc., pp.91-111, 1990.
- 2) 岡野, 横関, 並木, 高橋: “OS/omicon 第3版の設計と実現”, 情処OS研資, 45-11, 1989.
- 3) 横関, 岡野, 並木, 高橋: “OS/omicon 第3版ファイルシステムの設計と実現”, 第40回情処全国大会, 1990.3.

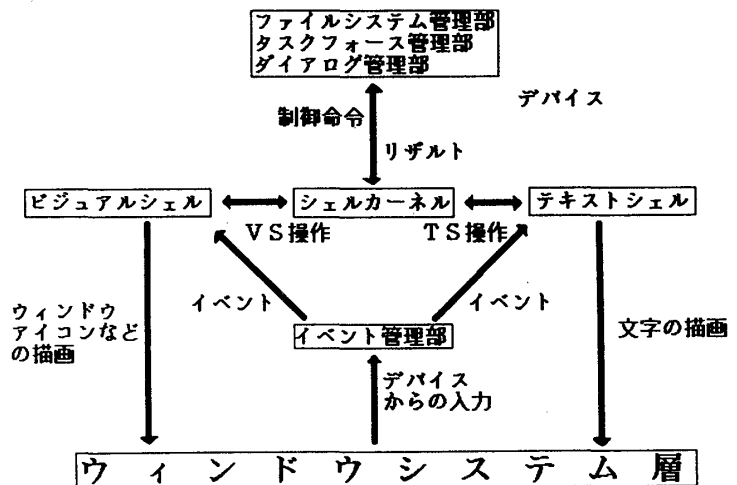


図2 シェルの全体構成