Unix のファイルシステムを中心とした コマンドインタプリタの視覚化

Visualization of Command Interpreter for The Unix File System

寺 茂夫　　田中 二郎
Shigeo TERA　　Jiro TANAKA

筑波大学
University of Tsukuba

概 要

Unix のコマンドインタプリタの視覚化を試みた。ここでは、Unix のコマンドに対応するアイコンを用意し、アイコンを選択することにより Unix コマンドを実行する。試作システムの実装を Tcl/Tk を用いて行ない、システムを使用した感想を収集した。また本システムの発展として「階層的なコマンド視覚化システム」及び「ソシアルインタフェースに基づくコマンド視覚化システム」について考察を行なった。

1 はじめに

計算機のオペレーティングシステムは、これまでは比較的計算機に詳しい個人の利用者を対象に機能を提供してきた。ところが、最近、計算機はその役割をオフィスから家庭へと拡大し、ユーザインタフェースやネットワーク機能の提供も計算機のオペレーティングシステムに必要な要件となった。

Unix はその誕生の経緯から開発志向の強い機能を、特に研究者やソフトウエア開発者に提供してきた。しかし、エンドユーザにとってコマンドラインベースの Unix は使いやすい環境とは言えない [1]。

パーソナルコンピュータに関しては、従来のコマンドラインをベースとするオペレーティングシステムは姿を消し、WIMP インタフェースに基づく、マック OS、Windows95 と言った GUI ベースのシステムが全盛である。しかし、Unix については、その後、X Windows System をベースとした GUI 環境によりユーザインタフェースの部分が改善されたと言っても、依然として、従来のコマンドラインの方式を取っている。

そこで、本論文では、従来の方式とは異なった、新しい Unix のコマンドインタプリタの視覚化方式を提案する。本論文の提案する視覚化方式は、アイコン [2][3] ベースのシステムであり、Unix のコマンドに対応するアイコンを用意し、アイコンを選択することにより Unix コマンドを実行するものである。

なお、本方式と、パーソナルコンピュータのマック OS、Windows95 などの視覚化方式との違いであるが、マック OS や Windows95 が、ファイルシステムの部分をアイコンによって視覚化しているのに対し、本方式では、Unix のコマンド自体をアイコン化しているのが相違点である。

2 Unix コマンドの視覚化システム

我々は、図 1 に示すような Unix コマンドの視覚化システムを Tcl/Tk[4] のウィジェットを用いて実装した。

図 1 の左側に、アイコンを用いて、縦に 5 列、横に 4 列で計 20 個配置されているのがコマンドであ

図 1　入力部のインタフェース

る。個々のコマンドはアイコン (絵) の部分、横の＊または - の書かれた部分、下の文字の書かれた部分の 3 つからなる。アイコンを用いた部分はボタンとなっている。＊または - の部分はメニューボタンであり、＊はメニューがあることを、 - はメニューのないことを示す。また下に書かれた文字は Unix のコマンド名を示す。

Unix のコマンドに対応するアイコンであるが、アイコンはそれぞれのコマンドに対し、コマンド実行により行なわれる処理の内容や、コマンドから連想されるイメージの日常生活への対応を考慮して選択した。アイコンは public　domain のものを用いた。

例えば、ls コマンドであれば、コマンドの処理内容は「指定されたファイルまたはディレクトリに関する情報を出力する」というものであるからカードファイルを表すアイコンを用いていた。すなわち「中に入っているものを整理して表示する」という点で対応づけたわけである。

他のアイコンも、コマンドの処理内容と日常生活の関連、語源等対応づけて選択した。その対応づけを以下に示す。

- mv コマンドは、ファイルやディレクトリが動くことをロケットが動く動作に対応づける。
- rm コマンドは、ファイルやディレクトリがなくなることを爆弾によって物がなくなる動作に対応づける。
- cp コマンドは、ファイルやディレクトリが複製されることを同じ物が存在することに対応づける。
- cat コマンドは、ファイルの内容が表示されることを猫が内容を表示することに対応づける。
- head コマンドは、ファイルの先頭 (頭) の部分を表示することを人間が頭にかぶるヘルメットに対応づける。
- tail コマンドは、ファイルの末尾 (足) の部分を表示することを人間が足に履くブーツに対応づける。
- chmod コマンドは、ファイルの属性を変えることをシートをチェックすることに対応づける。
- split コマンドは、ファイルを分割 (切る) ことをはさみによって切ることに対応づける。
- man コマンドは、分からないコマンド等を調べることを分からないことを示すクエスチョンマークに対応づける。
- cd コマンドは、次に示す pwd コマンドを家で表したので、その周りを移動することに対応づける。
- pwd コマンドは、自分が今いる所を家に対応づける。
- mkdir コマンドは、ディレクトリを作ると言う動作をディレクトリのメタファとして箱が存在することに対応づける。
- rmdir コマンドは、ディレクトリを消すと言う動作を一部が消えた木構造に対応づける。
- date コマンドは、日時を表示することを時間を表示するディジタル時計に対応づける。
- cal コマンドは、コマンドを実施した月のカレンダを表示することをカレンダに対応づける。
- bc コマンドは任意精度の簡易計算を行うことを計算機に対応づける。
- mailx コマンドは、電子メールを扱う時に用いることをハガキに対応づける。
- talk コマンドは、他のユーザと対話することを電話およびその回線に対応づける。
- lp コマンドは、プリンタへの出力要求をすることをプリンタに対応づける。

このような対応づけからわかるように、日常生活にあるものを表すコマンドはアイコンが選択しやすいが、抽象概念や動作を表すコマンドはアイコンが選択しづらい傾向がある。

また、図 1 の右側には、コマンドの引数の入力部

が配置されている。******** と書かれたラベルとエントリ部を 1 組として、それが 3 組ならんでいるが、そこからコマンドの取る引数を入力する。

3　コマンド視覚化システムの操作

ユーザは、アイコンの貼られたボタンをクリックすることでそのコマンドの入力をシステムに指示することができる。ボタンの横に＊のメニューボタンがあるコマンドは、そこを押すことによってそのコマンドのオプションをあらかじめ、選ばれたものの中から選択することが可能になる。

また、引数の入力が必要な場合には、図 1 の右側の ******** と書かれたラベルの部分が変化し、file name?,directory? と行った形で引数をたずね、ユーザはそれに対してエントリ部から、それぞれ引数を入力すればよい。

4　Tcl/Tk での実装

個々の Unix コマンドは、frame ウィジェット上に button ウィジェット、menubutton ウィジェット、label ウィジェットを pack した形で実装し、button ウィジェット上にアイコンを貼りつけた。

また、引数の入力部は、frame ウィジェット上に label ウィジェット、entry ウィジェットを pack した形で実装した。通常 label ウィジェットには,******** と書いてある。

5　システムに関する議論

我々は、本システムを、研究室のメンバーの数人に実際に使用してもらい、システムを使用した感想を収集した。それらをまとめると以下のようになる。

- コマンドの処理内容のメタファとしてアイコンを用いたが、その中にコマンドの処理内容を的確に表していないもの、また表していても初心者にとって分かりにくいものが含まれている。
- アイコンがコマンドの処理内容のメタファとならない可能性があるならば、アイコンの代わりに日本語で説明したものを用いたらどうだろうか。
- Unix においてパイプやリダイレクションを用いたプロセス間におけるデータの移動は頻繁に行われるものであり、その実装を行うべきである。
- ls コマンドを実行するときなど、通常ユーザは複数のオプションを同時に用いる。よって、複数のオプションを同時に実行可能にすべきだ。
- ファイル名やディレクトリ名にさいしてよく使うと思われるものあるいは、最近使われたものなどは簡単に使えるようにすべきだ。
- 個々のユーザがもっとも能率よくシステムを使うためにコマンドの位置などカスタマイズ機能を導入したらどうか

図 2　トップ画面

6　新しいコマンドインタプリタ

本視覚化システムの発展としては、いくつかの形態が考えられるが、ここでは、「階層的なコマンド視覚化システム」「ソシアルインタフェースに基づくコマンド視覚化システム」について述べる。

6.1　階層的なコマンド視覚化システム

Unix コマンドをフラットに並べるのではなく、階層的に整理した「階層的なコマンド視覚化システム」が考えられる。

本視覚化システムの基本的なコンセプトは以下の 2 点である。

- コマンドを関連づける。
- 段階的にユーザの要求に答える。

例えば、今、あるディレクトリにあるファイル名 conferense を conference に変更したいという要求がある場合を想定する。

まずユーザは、図 2 に示すトップ画面の中からアイコンをマウスでクリックすることによって、

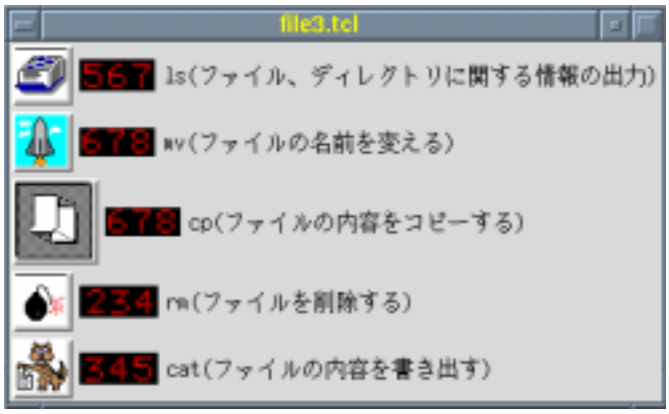

図 3　ファイル関連コマンド一覧

- ファイルに関連するもの

を選択する。

ユーザが、「ファイルに関連するもの」を選択すると図 3 に示すようなこのシステムが持つファイルに関連したコマンドの一覧が表れる。

ユーザは要求に合わせて

- mv コマンドのアイコンをクリックし、そのコマンドを選択する。

そうすると、変更したいファイルの現在の名前と変更後の名前を聞いてくるので、それを入力すればよい。

ここでは、コマンドおよびオプション等の参照回数を示すために、コマンドを表すアイコンの横にカウンタを設けてある。

6.2　ソシアルインタフェースに基づくコマンド視覚化システム

Unix コマンドが処理される空間内を街にみたて表現し、コマンドインタプリタを、街とコマンドの処理内容を対応づけて、一種のソシアルインタフェースとして実現することが考えられる。具体的には、入力部からイベントを受け取ると街の中でエージェントがそのイベントに対応した行動を行うようにする。

図 4 に示すように、

- cd コマンドを入力として与えれば、エージェントは家に向かい、その後でコマンドに与えられた引数にしたがって家の中を上下する。
- pwd コマンドを入力として与えれば、家に向かい、その中でエージェントが、今いる部屋の表札を大きく表示する。

図 4　ソシアルインタフェースに基づく視覚化

- mailx コマンドを入力として与えれば、エージェントは街の中を歩き郵便局についた後で、アドレスの示す国の国旗に向かって手紙を投げる。
- man コマンドを入力として与えれば、エージェントは街の中を歩き図書館へ着いた後で、コマンドのアルファベットを索引として建物の中を上下する。

こうしたエージェントの行動を見ることによってユーザはそのコマンドの処理内容を生活感覚に対応づけ理解することができると思われる。

参考文献

[1] Francesmary Modugno
"Interface Issues in Visual Shell Programming," in Visual Object-Oriented Programming, pp.95-111, Manning Publications

[2] 市川 忠男、平川正人共著.
かわりゆくプログラミング、情報処理学会、 1994

[3] Nan C. Shu.
Visual Programming Van Nostrand Reinhold, 1988 (西川 博昭訳. ビジュアルプログラミング、日経 BP 社、 1991)

[4] John K.Ousterhout.
Tcl&Tk Toolkit Addison-Wesley Publishing Co., 1994 (西中 芳幸、石曾根 信訳. Tcl&Tk ツールキット、SOFT BANK、 1995)