

引数単位の動的なメニュー切り替えによる コマンド入力支援システム

中村 和彦 守田 了 田中 稔

山口大学工学部

宇部市常盤台 2557

{kazuhiko, morita}@cs.csse.yamaguchi-u.ac.jp

UNIX等に用いられているコマンドユーザインタフェースの入力負担を軽減するために、コマンドラインの各引数を一般化して引数記号の列で表現し、コマンド予測に利用する事例が報告されている。しかし予測をコマンドライン単位で行ない、単一のコマンドライン候補を提示するため、予測誤りに対する訂正を行わなければならない。そのため本研究ではコマンドラインの引数単位かつ複数の候補を提示する支援システム Dynamic Menu shell (DMsh) を提案する。コマンドラインの入力状況に応じて、メニューに提示する引数候補のカテゴリを動的に切り替えることにより、その各引数をメニュー選択により提供する。実際に20人の被験者に本システムを使用してもらい、入力量や入力時間を計測しその有効性を評価する。

System Supporting Command Input an Operator by Dynamic Menu Shell

Kazuhiko NAKAMURA Satoru MORITA Minoru TANAKA

Faculty of Engineering, Yamaguchi University

2557 Tokiwadai, Ube, 755, Japan

To reduce the burden of the command user interface's input in UNIX etc, the computer supported interface, which generalizes each operators of command-line and predicts the command line, has been reported. But, it must correct errors of the prediction because it predicts a command-line and offers single command-line candidate. In our research, we suggest the supporting system "Dynamic Menu shell (DMsh)", which shows some candidates for a command-line's operator. It offers the operators by dynamically changing the category of the operator's candidates according to the input situation of command-line using selecting menu.

1 はじめに

UNIXでは、一般にコマンドユーザインターフェースが広く用いられている。文字列を入力するだけでコマンドを実行できるため、少ない領域で作業を行なうことができる。その反面キーボード入力に不慣れなユーザにとって、長いコマンドラインを入力するのに時間がかかってしまい作業が遅れ、ストレスの原因になる。そのためコマンド入力を簡易にするための様々なインターフェースが研究されている。[1]ではコマンドラインの各引数を一般化して引数記号の列で表現し、コマンド予測に利用する事例が報告されている。しかし、予測をコマンドライン単位で行ない、単一のコマンドライン候補を提示するため、予測誤りに対する訂正を行なわなければならない。

本研究では引数入力ごと細やかな支援を行なうために、予測をコマンドラインの入力状況に応じたコマンドラインの引数単位で行ない、複数の引数候補をメニューによる一覧で提示するシステムを提案する。

2. では設計方針と動的なメニュー切り替えシステムの概要について、3. では、動的にメニュー内に提示する項目を切り替えるメカニズムについて、4. では実際に本システムを用いて入力量や入力時間を計測し、その有効性を評価する。

2 動的なメニュー切り替えシステム (Dynamic Menu shell:DMsh)

2.1 設計方針

従来のコマンド予測システムでは、コマンドライン単位で予測を行ない、単一のコマンドライン候補を提示する。しかし、コマンドラインはコマンド名と引数単語の多様な並びから成り立っており、正しい予測を導き出せる確率は30%前後である[2]。コマンドラインの入力状況にリアルタイムに対応し、ある程度の選択許容範囲を持たせることが必要である。そこで、以下のような設計方針に従い、コマンド入力支援システムを設計する。

- ・ 予測はコマンドラインの入力状況に応じたコマンドラインの引数単位で行なう
- ・ 複数の引数候補を一覧で提示する

2.2 システムの概要

これから入力するコマンドラインのカテゴリを予測し、そのコマンドラインの入力状況に応じてメニューに

提示する引数候補のカテゴリを動的に切り替えることにより、コマンドラインの各引数をメニュー選択により提供するシステムを提案する。このシステムをDynamic Menu shell(DMsh)と名付ける。コマンドラインの各引数を属するカテゴリに置き換えることにより、コマンドライン間の関係をより明確にする。このシステムにおけるカテゴリをコマンドラインを細分化する最小限のものとして、以下の6種類に定めた。

- (1) Com コマンド名
- (2) Option オプション名
- (3) File ファイル名
- (4) Directory ディレクトリ名
- (5) Transmit 入出力ダイレクション、パイプ等
- (6) Express 入力文字列、未登録のオプション名、新ファイル名、新ディレクトリ名

2.3 システムの使用法

まず作成したシステムの概観を図1に示す。システム上のコマンド入力フィールドにメニュー内の引数を選択することでコマンドラインを入力することができる。

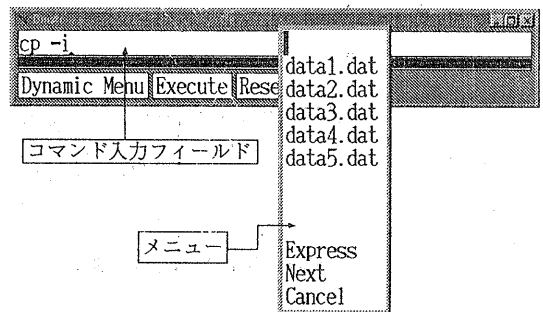


図 1: DMsh

システムの使用法について図2を用いて説明する。システム内のコマンド入力フィールド上に“cp”と入力されている状態の時、カテゴリを引数にもつコマンドライン“cp File Directory”が予測されたとする。この時((1)の状態)、メニューをポップアップすると、カテゴリがFileである引数候補がメニュー上に表示され、必要とする引数を選択することにより、コマンド入力フィールドに入力することができる。また、(2)の状態でメニューをポップアップすると、カテゴリがDirectoryである引

候補がメニュー上に表示され、必要とする引数を選択することにより、コマンド入力フィールドに入力することができる。また始めからまたは途中でカテゴリを変更することも可能であり、必要な時だけ支援を要求することができる。なおこのシステムはキーボードまたはマウスのどちらでも操作が行なえる。

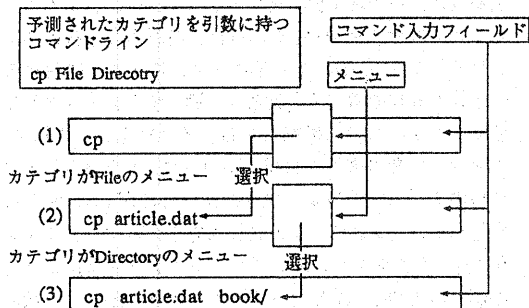


図 2: DMsh の使用法

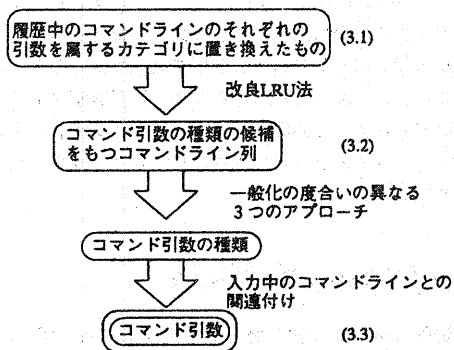


図 3: 引数候補予測のための流れ図

3 メニュー切り替えのメカニズム

本システムの最も主要なシステムである動的なメニュー切り替えによりコマンド引数を予測するメカニズムについて説明する。コマンド引数の入力状況が変化することにより、メニューに使用するコマンド引数の種類の候補をもつコマンドライン列を求める手法について説明し、それからコマンド引数の種類をもつコマンドライン列に変換し、入力状況に該当するメニュー内コマンド引数の候補予測する手法について説明する。その流れを図3に示す。

3.1 コマンド引数に基づく改良 LRU 法

コマンド予測に用いられている LRU 法は、直前のコマンドラインをもとに、次のコマンドラインを決定する。本システムでは、LRU 法を改良して、ユーザの履歴に含まれるコマンドラインの各コマンド引数を 2 章で定義した Comi を除く 5 種類のカテゴリに置き換え、直前に実行されたコマンドラインと入力中のものの入力状況までが一致する 2 行分のコマンドラインを履歴の最新のものからすべて取得する。この直前のコマンドラインをもとに次のコマンドラインに含まれる引数を決定する改良 LRU 法を提案する。図 4 にその例を示す。

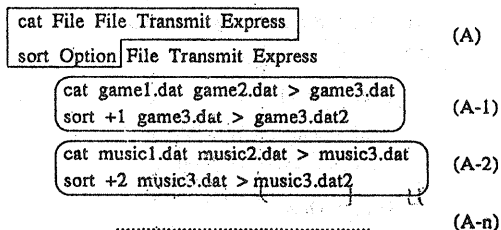


図 4: コマンド引数に基づく改良 LRU 法 (A) 入力中の 2 行分のコマンドライン (A-1) 改良 LRU 法による第 1 候補 (A-2) 改良 LRU 法による第 2 候補

直前に実行したコマンドラインの各引数をカテゴリに置き換えたものが“cat File File Transmit Express”とする。この時、入力中のコマンドラインの各引数をカテゴリに置き換えたものが“sort Option”の時、履歴から改良 LRU 法により (A) のカテゴリ順序をもつコマンドライン列 (A-1)(A-2)...(A-n) が取得される。最近の履歴から取得した順に第 1 候補 (A-1)、第 2 候補 (A-2)、第 n 候補 (A-n) とする。

3.2 コマンド引数の種類

コマンドライン間の依存関係を反映させるために、取得した 2 行分のコマンドラインを用いてメニューに使用するコマンド引数の種類を一般化の度合いが異なる 3 つのアプローチを用いて決定する。

1 つめのアプローチでは取得した 2 行分のコマンドラインにおけるコマンド引数がファイルであるものをファイル名から拡張子を取り除いた部分と拡張子に分けて、2 行分のコマンドライン中のファイル名が同じものを共通の識別子で置き換える。また、ディレクトリ名も同じ名前のもを共通の識別子で置き換える。2 つめのアプ

ローチでは、入力中のコマンドラインに対応する履歴から得られた1行分のコマンドラインに対して、拡張子が同じファイルと同じ識別子に置き換える。3つめのアプローチでは、コマンドラインの各引数を属するカテゴリに置き換えたものを用いる。

アプローチは番号が大きい方が一般化されているため、1つ目のアプローチで該当するコマンドラインが存在しない場合は、2つめのアプローチ、3つめのアプローチを使用してメニューに使用するコマンド引数の種類を決定する。その例を図5に示す。

```
(A)
cat game1.dat game2.dat > game3.dat
sort +1 game3.dat > game3.dat2

(B)
(Approach 1)
cat $1.dat $2.dat > $3.dat
sort +1 $3.dat > $3.dat2

(Approach 2)
sort +1 *.dat > *.dat2

(Approach 3)
sort Option File Transmit Express
```

図5: コマンド引数の種類の決定 (A) 入力する2行分のコマンドライン (B) (Approach1) (Approach2) (Approach3) によるコマンド引数の置き換え

コマンド引数の種類の候補を持つコマンドライン列が(B)である時、1つのアプローチ、2つのアプローチ、3つのアプローチにより変換されたコマンド引数の種類を持つコマンドライン列は図のようになる。

3.3 コマンド引数

取得したコマンド引数の種類を持つコマンドライン列を用いて、メニュー内に表示するコマンド引数の候補を予測する手法について説明する。メニューに使用するコマンド引数の種類を決定したアプローチに従って置き換えられた2行分のコマンドラインから、入力中のコマンドラインの直後に、メニューに使用するコマンド引数の種類を置いたコマンドラインと直前に実行されたコマンドラインを置き換えた2行分のコマンドラインを比較する。

1つめのアプローチでは、比較の結果が同じ履歴中の2行のコマンドラインのうち、メニューに表示するコマ

ンド引数が Option または Transmit の場合、入力途中の直後にくるコマンド引数をメニューの第1候補とする。メニューに表示する第1候補とは、メニューの一番上に表示するアイテムである。また、メニューに表示するコマンド引数が File の場合は、履歴中の2行のコマンドラインと入力中の2行のコマンドラインにおけるファイルの識別子と拡張子を比較し、適合するファイル名をメニューの第1候補とする。メニューに表示するコマンド引数が Directory の場合は、履歴中の2行のコマンドラインと入力中の2行のコマンドラインにおけるディレクトリの識別子を比較し、適合するディレクトリ名をメニューの第1候補とする。その例を図6に示す。コマンド引数の種類の候補を持つコマンドライン列を1つめのアプローチにより置き換えたものが(a)、直前に実行したコマンドラインおよび入力中のコマンドラインが(b)、それを1つめのアプローチにより置き換えたものが(b')の時、(a)と(b')よりターゲットとなる引数の種類は“\$3.dat”であり、変数の値“\$3 = music”であるため、予測された引数の候補は“music.dat”となりメニューの1番上に表示する。

2つめのアプローチでは、入力中のコマンドラインのみのファイルの拡張子と同じ拡張子を持つものをカレントから探し出しメニューに表示する。3つめのアプローチでは、メニューに表示するコマンド引数がファイルまたはディレクトリの場合、カレントディレクトリ中に存在するすべてのファイルまたはディレクトリをメニューに表示する。

4 DMsh の評価実験

動的なメニュー切り替えシステム DMsh を用いたコマンドラインの入力と最も一般的に利用されているコマンド入力支援である tcsh を用いたコマンドライン入力を比較し、ユーザの作業負担に関して検証する。

履歴にあらかじめコマンドライン間に依存関係のある4種類の作業(ファイルの整理、プログラムのコンパイル、TEXのコンパイル、データの検索)を登録し、用いられているファイル名、ディレクトリ名を変更したコマンドラインを以下に示す3つの実験方法で被験者に入力してもらった。被験者は日頃 UNIX を用いて作業している研究室のメンバー20人を対象とした。

実験1: tcsh 等日頃用いているコマンド入力支援

実験2: マウス操作主体の DMsh

実験3: キーボード操作主体の DMsh

それぞれの実験に要した時間を表1に示す。単位は秒である。

図7は実験1のtcshによるコマンドライン入力に対する実験2、3のコマンドライン入力の入力時間短縮率を表している。括弧内の数字は左から平均値、標準偏差を表している。実験2、3ともに入力時間を短縮できていることを表している。

図8は実験で用いた履歴におけるDMshを用いた時の予測正答率を表している。一回参照した時のメニュー内に必要とするアイテムが含まれていれば正答とする。横軸がコマンド引数の数、縦軸が正答率を表している。ほぼ、100%正しい予測をしていることが分かる。

また、図9は実験で用いた履歴によりシミュレーションで作成したキーボード入力、LRU法による入力、DMshによる入力のコマンドライン数と入力量の関係を表したものである。横軸はコマンドライン数、縦軸は入力量を表しており、DMshによる入力量がかつとも少なくなっていることが分かる。

また、図10は図9を用いて作成したキーボード入力に対するLRU法、DMshによる入力のコマンドライン数と補完率の関係を表したものである。横軸がコマンドライン数、縦軸が補完率を表している。DMshによる入力が高い補完率を示していることが分かる。

また、各被験者に実験後にアンケートに答えてもらった。その結果を表1、表2に示す。単位は人数である。

表2 アンケート結果1

	あり	なし
tcshの使用経験	20	0

(単位 人数)

表3 アンケート結果2

	◎	○	▲	△	×
tcshの理解度	1	9	2	3	3
DMshのマウス操作	10	9	0	1	0
DMshのキーボード操作	8	9	1	0	0
メニューによる引数候補の選択	11	8	1	0	0
DMshによるコマンド入力	11	9	0	0	0

(単位 人数)

実験結果およびアンケート結果より、被験者全員がtcshを使用した経験があり、その大半がその機能について理解しているにもかかわらず、本システムを用いた方が被験者全員、速く入力することができた。また、被験者全員が本システムを用いたメニューによる引数候補を選択することが便利であると感じ、本システムを用いたコマンドライン入力が楽であると感じている。また、本システムをマウスにより操作するよりもキーボードにより操作する方がマウスを動かすというオーバーヘッド

時間がないため、速く入力できたと考えられる。また、DMshのキーボード操作に関してコントロールキーを用いているため慣れるまで操作しづらく、困惑した被験者もいたが、キー配置を変更することで解決することができる。

5 おわりに

コマンドライン入力を簡易に行なうために、コマンドラインを引数単位かつ複数の候補をメニューにより提示するDMshを提案した。動的に切り替わるメニューを用いることで、コマンドラインの引数単位で予測を行なうことができた。実験によりコマンドラインの入力時間および、ユーザの負担を軽減できることが明らかになった。

参考文献

- [1] 上田康裕、西村俊和、美濃彦彦、池田克夫、"一般化コマンドラインを用いたコマンドラインの予測および誤り検出・訂正"、信学技報、HC94-88(1995)
- [2] Saul Greenberg, Ian H. Witten, "Supporting command reuse: mechanisms for reuse", Int. J. Man-Machine Studies, pp.391-pp.425 (1993)

表1 各実験における入力時間

被験者	実験1	実験2	実験3
A	406	341	*222
B	585	358	*269
C	571	362	*274
D	482	405	*238
E	734	365	*293
F	670	421	*366
G	402	227	*204
H	829	434	*403
I	738	403	*344
J	831	605	*511
K	421	410	*254
L	550	386	*312
M	678	389	*363
N	451	343	*227
O	474	402	*230
P	945	422	*358
Q	918	360	*353
R	593	459	*366
S	638	540	*324
T	534	425	*254

(単位 秒)

```

(a)
cat $1.dat $2.dat > $3.dat
sort +1 $3.dat > $3.dat2

(b)
cat music.dat music2.dat > music3.dat
sort +1

(b')
cat $1.dat $2.dat > $3.dat
sort +1

(c)
$3 = music && $3.dat --> music.dat

```

図 6: コマンド引数候補の導出 (a) 履歴中の 2 行を 1 つ目のアプローチにより置き換え, (b) 入力中の 2 行 (b') 入力中の 2 行を 1 つ目のアプローチにより置き換え, (c) \$3 は music と予測されている。

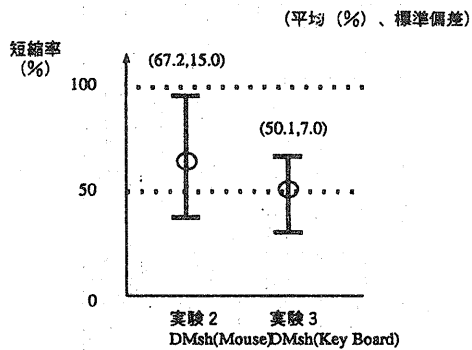


図 7: 実験 1(tcsh) の入力に対する実験 2、3 の入力時間短縮率

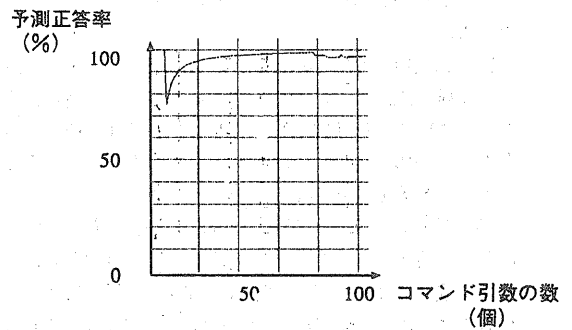


図 8: DMsh の予測正答率

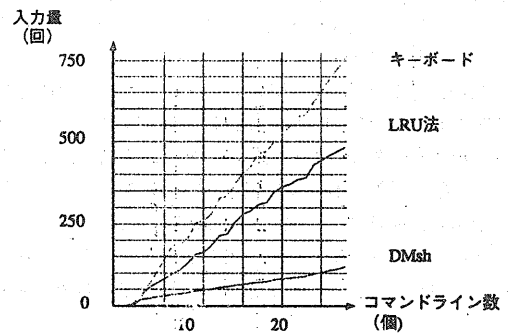


図 9: 各手法によるコマンド入力量

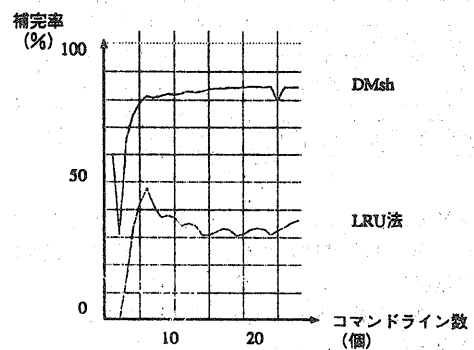


図 10: LRU 法と DMsh の補完率