

マウスカーソルの移動方向を考慮した階層メニュー展開法

小林 正朋[†] 五十嵐 健夫[‡]

[†] 東京大学理学部情報科学科 [‡] 東京大学大学院情報理工学系研究科

^{† ‡} {kobayash, takeo}@ui.is.s.u-tokyo.ac.jp

1. はじめに

GUI システムにおいて広く用いられているコンポーネントの一つに、階層メニューがある (Windows^(R)の“スタートメニュー”、Mac^(R) OSの“メニューバー”等)。しかし、現在普及している階層メニューに関して「意図していないアイテムにフォーカスが移ってしまう」「サブメニューの展開が遅い」といった難点を挙げるユーザも多い。

本稿では、上記の問題点を考慮した階層メニューの新しい動作手法を提案する。

2. 通常の階層メニューの問題点

様々なプラットフォーム上で実装されている階層メニューについて、サブメニュー展開時の挙動にはそれぞれ細かな相違があるものの、概ね以下の二点にまとめられる。

メニュー項目上でのマウスカーソルの停留またはマウスクリックによって、対応するサブメニューが、親メニューの右端に密着して出現する。

前者の特徴により、サブメニューを開こうとするユーザは無意味な待機または意図的な行動 (マウスクリック) を強いられることになる。

また、後者の特徴のためにユーザは自分が開いたサブメニューまでマウスポインタを水平方向に長距離移動させなければならないのだが、このとき誤って垂直方向にもマウスを動かしてしまうと、意図しないフォーカス移動を生じてしまうのである (図1)。

これらの問題点は相反する関係にあり、たとえばサブメニューの出現に要する待機時間を短くすると、過敏なフォーカス移動を誘発してしまう。

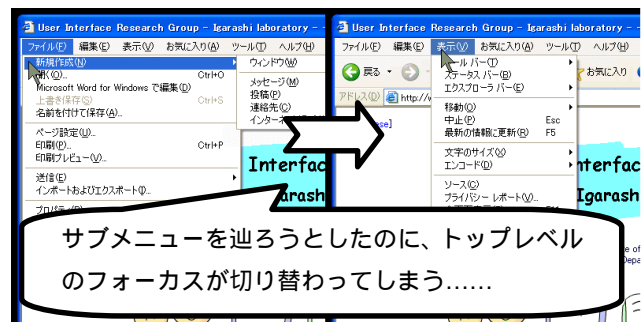


図1：意図しないフォーカス移動の例

3. 提案する手法

新しい手法では、サブメニューを展開するためのトリガーとしてマウスポインタの停留やマウスクリックではなく、マウスの右方向の移動を用いる。マウスが水平寄りの方向へ動かされた場合には移動の垂直成分を無視するため、僅かな軌道の誤差によって意図していない項目を選択してしまう事故を防げる。なお、左方向に移動させた場合にはそのメニューがクローズされる。

逆に、マウスの垂直方向の移動には従来通りメニュー内の各項目に対するフォーカスの移動を割り当てる。水平方向の場合と同様に、このとき移動の水平成分は無視される。

また、マウスの移動量を軽減するため、サブメニューを親メニューの右端ではなくマウスポインタの直下へ生成するようにする。サブメニューが現れた後、いったんマウスを垂直方向に動かさない限りサブメニュー内の項目がフォーカスされることはないので、サブメニューの展開に要するマウス移動量をごく小さく設定したとしても、移動させすぎて過剰な展開が生じてしまうという問題は起きない (図2)。

この手法は、ユーザが既存の階層メニューを利用する際に見出されるいくつかの傾向に基づいている。

第一に、ほとんどのユーザはメニュー内の深い階層を辿る際、マウスポインタを最短経路で移動

させず、縦移動と横移動を繰り返しつつ階段状に動かしていく（図3）。また、ユーザがサブメニュー展開の遅延を待つ場所はサブメニューの出現場所に近いメニュー枠の右端ではなく、メニュー枠の中ほどであることが多い。

したがって、階層メニューを辿るユーザの行動を総合すると、およそ次のようになる。

メニュー上で垂直方向にマウスポインタを動かし、目的の項目に辿り着く。

さらに深い階層へ進む場合には、その場でサブメニューが現れるのを待つ。

水平方向にマウスポインタを移動させ、サブメニューに向かう。

に戻る。

新しい手法は、このような傾向を持ったユーザが操作に違和感を覚えることなく、水平方向の移動量を抑えながら的確に階層メニューを辿れるよう留意して設計されている（図4）。

なお、本手法を用いた GUI コンポーネントの外観は通常の階層メニューとほぼ同一であるため既存のシステムとの親和性が高く、併せて実装が容易であると考えられる。

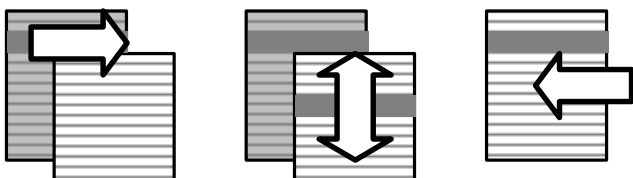


図2：提案する手法によるサブメニューの展開（左）、フォーカス移動（中）、サブメニューのクローズ（右）

4. おわりに

本稿では、マウスポインタの移動方向に関する情報を用いてメニュー内の項目に対するフォーカシング操作とサブメニュー展開操作を区別し、効率的に階層メニューへアクセスするための手法を紹介した。

ポインティングカーソルの移動方向を利用したメニュー選択機構としては“パイメニュー[1]”等があるが、本手法はメニュー項目を選択または決定するためではなく選択操作と決定操作を区別するために方向情報を活用しているという点で異なっている。また、本手法はマウス入力およびペン入力のいずれかに不得手ということはない。

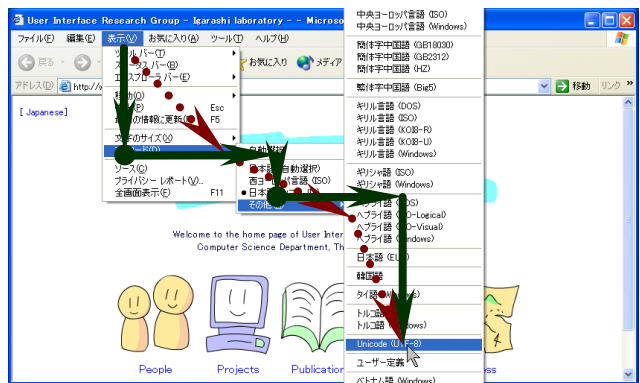


図3：既存のメニューバーにアクセスする場合のマウスポインタの最短移動経路（点線）と実際のユーザが辿った軌跡（実線）

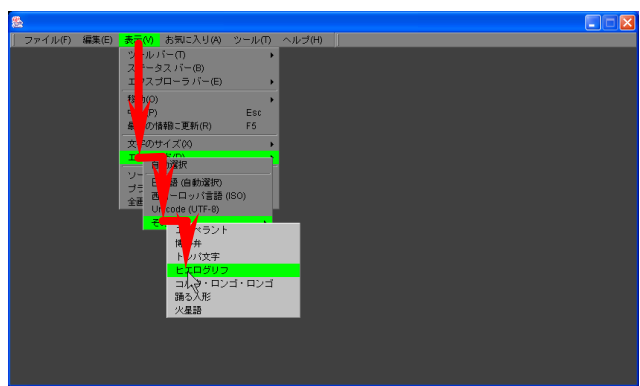


図4：作成したプロトタイプでのメニュー選択操作

しかし、幅の狭いメニュー項目にマウスポインタを重ねる作業の困難さといった Fitts' Law[2]に関わる欠点は依然として残されており、将来の改善が期待される。

今後、メニューアクセス速度、エラー率、およびメニューアクセス時のマウスポインタの軌道等に関して本手法と既存の階層メニューとを比較し、定量的な分析を行う予定である。

参考文献

[1] Don Hopkins. The Design and Implementation of Pie Menus. *Dr. Dobb's Journal*, 1991.

[2] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, vol. 47, 381-391, 1954.

[3] I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. Accuracy Measures for Evaluating Computer Pointing Devices. In *Proceedings of CHI 2001*, 9-16, 2001.