

ソフトウェア移行におけるユーザモデルの再構成

2J-3

中村 有一 平嶋 宗 豊田 順一

大阪大学産業科学研究所

1. 序論

使用するソフトウェアをあるソフトウェアから別のソフトウェアに移行する場合、ユーザをとりまくタスクの実行環境が変化したといえる。タスクの実行環境が変化すれば、以前使用した機能と同じコマンドがどこにあるのかわからなかったり、新しく追加された便利な機能に気付かなかったりするという現象が発生する。したがってユーザは新しい環境への適応を要求されることになる。適応が容易であるかどうかは環境自体の善し悪しのほかに、以前に適応していた環境との差異によるものが考えられる。タスク実行環境の改良についてはこれまでも多くの努力が払われており大きな成果をあげている[1]。これに対して、以前に適応していた環境との差異が与える影響についてはほとんど研究されていない。今日のように非常に幅広い人々が多様なソフトウェアを利用し、またソフトウェアの改良が急速に進んでいる状況においては、使用ソフトウェアの移行はしばしば起こる現象であり、その影響を明らかにすることの意義は大きいと考えられる。

タスクの実行環境をタスクー行動マッピング知識の集合として表現した上で、差異の有無の影響を調べた研究はすでに行われている[2][3]。しかしながら、すべての差異がユーザのふるまいに同じ影響を与えているとは考えにくい。本研究の目的は、差異の種類と影響を調査、整理することである。

本稿では、タスクー行動マッピング知識をメニュー構造として捉え、メニュー構造の差異の種類とソフトウェア移行時のユーザのふるまいへの影響について検討した。また、メニュー構造の変化による効果の利用法として、ソフトウェアの移行がユーザに与える影響を予測するためのユーザモデルを実現する方法を提案する。

2. 移行時の知識の再構成

ソフトウェアの移行を新しいタスク実行環境への適応と捉えると、新しい環境に合わせた知識の再構成が必要となる。その知識としては2種類考えられる、1つはタスクの分割知識であり、与えられたタスクを分割することによって、行動と直接結びつけられるようなプリミティブなタスクに展開するためのものである。もう1つはタスクー行動マッピング知識であり、プリミティブなタスクとソフトウェア上での行動手順を結びつけるものである。タスクの分割知識は移行前後のソフトウェアに対してほとんど共通であり、移行前のソフトウェアで利用されたものがほとんどそのまま使えらる。つまり、移行前のソフトウェアにおいて実行したことの無いタスクを移行後のソフトウェアにおいて実行する場合を除くと、ソフトウェアの移行時の知識の再構成の大部分はタスクー行動マッピング知識に関して起こると考えられる。本研究では、タスクー

行動マッピング知識を対象とし、その知識としてメニュー構造を取り扱う。これはメニュー構造はタスクー行動マッピング知識を階層的に表現したものであると考えられるからである。そしてメニュー構造の差異をソフトウェア移行時の環境の差異と捉える。一般にメニュー構造で操作可能なソフトウェアが多く、また近年の発達したGUI環境におけるアイコンなどは選択肢の提示方法の違いと考えることができ、マッピング知識としてみた場合はメニュー項目の一種とみなせることからメニュー構造を扱うことは十分に妥当であると考えられる。しかし、画面上の配置など空間的な要因が含まれるので今後検討していく必要はある。

3. メニュー構造の抽出

3.1 メニュー構造

メニュー構造には、選択した項目によって次にあらわれる項目が変化する部分と何を選択しても次にあらわれる項目が同じままの部分が存在する。本稿では、前者の部分コマンド決定部、後者の部分をパラメータ入力部と呼ぶ。一般にこれらはコマンド決定部、パラメータ入力部の順であらわれる。例えば、文書を印刷したい場合、文書印刷という項目を選択するまではどの項目を選択するかによって次にあらわれる項目が変化するが、文書印刷を選択して以降に設定する行間やページ番号の有無などの選択は次にあらわれる選択項目に影響を与えない。また、パラメータ入力部はたいいてい画面上に同時に複数の項目が提示されている。このような違いはユーザの行動に大きな影響を与えていると考えられるので、本研究では差異の分析において、コマンド決定部とパラメータ入力部の違いを1つの分類基準としている。

3.2 メニュー構造の抽出

本研究では、移行前後のメニュー構造の差異の種類の違いによる、ソフトウェア移行時のユーザのふるまいに対する影響の違いを調べる。そこで、どのような差異が存在するのかを調査するために、実際に存在するソフトウェアについての2つのバージョンアップ関係を対象としてメニュー構造の差異部分を抽出し分類した。差異部分は大きく移動、変更、追加、削除の4つに分類できる。ソフトウェアを移行した場合、以前の知識が移行後の操作に影響を与えるのは移動と変更と削除であると考えられる。追加部分は移行前のソフトウェアには存在していないため、移行したユーザは追加部分の知識を持っていないと考えられる。また、削除部分は他のユーティリティソフトに移動していることが多くタスクの分割知識が変わるため今回は対象外とする。ここでは移動と変更について述べる。移動はコマンドの移動、パラメータの移動、コマンドのパラメータ化、パラメータのコマンド化、その他の5種類に分類できる。一階層上へ移動などを考慮に入れてさらに細かく分類すると移動は12種類に分類できる。変更はコマンド名の一部変更したものとコマンド名の全部変更したものの2種類に分類できる。

4. 実験

ここでは、移行前の知識を持つユーザが、ソフトウェア移行時にメニュー構造の変化に遭遇した場合にどのようなふるまいがみられるかを調査した実験について述べる。

表1 差異とその効果

差異の種類		効果	
移動	コマンドの移動	一階層下へ	×
		一階層上へ	○
		同じ階層へ	△
	パラメータの移動		○
	コマンドのパラメータ化		×
	パラメータのコマンド化	同じ位置へ	○
		一階層上へ	×
	その他	コマンドの分解	○
操作系列の変化		×	
変更	一部名称変更	○	
	全部名称変更	△	

影響有：× 影響無：○ 不明：△

4・1 実験方法

実験のために、抽出した差異を反映させることのできる2つのメニューシステムを作った。これらのメニューシステムはメニュー選択機能のみをもち、その表示形態は全く同じである。2つのメニュー構造から抽出した差異部分のうち移動と変更がみられた機能に対して、その機能を選択させるような課題を用意した。

実験は各被験者に対して以下の手順で行った。ここでシステム1は移行前のソフトウェア、システム2は移行後のソフトウェアに対応している。

(1) 課題の中から順に1問ずつ与え、システム1を操作してもらう。十分に操作できるようになるまで課題系列を繰り返し与える。

(2) (1)で与えた課題と同じ課題をシステム2で実行してもらう。

以上の手順において課題が達成されるまでにメニュー項目を選択した回数とメニュー項目を選択するのに要した時間を測定した。それにより、メニュー構造の各差異について移行前のユーザの知識が移行後のメニュー項目の選択に影響を与えるかを評価した。

4・2 実験結果

課題を達成するためのメニュー選択系列が変化していないものは、移行前のシステムの選択系列を憶えたときとほぼ同じ選択時間と選択回数で選択できた。このことから移行前のソフトウェアの知識が役に立っていることが分かる。移動に関して影響が見られたものには、コマンドが一階層下へ移動したものの、コマンドがパラメータ化したものの、パラメータが異なる位置でコマンド化したもの操作系列の変化したものであった(表1)。これらのメニュー項目選択回数を課題を達成するために必要な選択回数と比較すると約2~3倍ほどであったが、コマンドが下へ移動したもののうち、新しいコマンドの下へ移動したものは3倍以上であった。影響が見られなかったものには、コマンドが一階層上へ移動したものの、パラメータの移動、コマンドの分解、パラメータが同じ位置でコマンド化したものであった。これらは課題を達成するために必要な選択回数とほぼ同じであった。コマンドが同じ階層へ移動したものは課題によって結果が分かれた。変更に関してはコマンド名が一部変更したものは影響がなく、コマンド名が全部変更になったものは課題によって分かれた。メニュー項目を選

択するのに要した時間は、選択回数が特に多かったものは非常に時間を費やしているが、それ以外は明確に区別するには至らなかった。これは課題の内容を認識する時間やコマンド選択時にコマンドの意味による影響を受けている可能性を示唆している。

4・3 考察

実験により、差異の種類によってユーザに与える影響が異なるとの知見が得られ、ユーザが以前のソフトウェアの知識を持っていることによって困惑する差異と困惑しない差異があることが分かった。このことから差異の種類とその影響の整理が可能ではないかとの知見が得られた。さらに困惑する差異のうち、課題を達成するための選択回数に対して2~3倍のものは、近傍を探索した結果見つけられた場合で、3倍以上のものは全探索を行う傾向が見られた。このことはユーザの用いる探索戦略との関係を考慮することの必要性を示唆している。また、今回の実験では、メニュー項目の意味の影響については考えていなかったが、被験者がコマンド名の意味を考慮に入れて選択していることが考えられるので、今後メニュー項目を意味を含まないように記号化して構造の差異のみを反映した実験を行う必要がある。また、一般性を持たせるために被験者の数を増やして調査する予定である。

5. 移行時ユーザモデルへの利用

実験で得られた知見は、ソフトウェア移行時のユーザのふるまいを予測するのに利用することができる。ソフトウェアにおけるユーザのふるまいをメニュー構造上の操作に限定してユーザモデルを考えると、このユーザモデルはメニュー構造上のオーバーレイモデルとなっており、ユーザの使用した選択系列と選択回数などが記述できる。実験の結果のようにメニュー構造の差異部分とその効果が分かれば、それを利用して移行前のソフトウェアのユーザモデルを再構成することができる。再構成したユーザモデルは、ユーザがソフトウェアを移行した場合にどのような行動をとりうるかを表した移行時ユーザモデルでありユーザが遭遇する困難さが予測できるとともに、以前の知識を利用して問題なくふるまえる場合もあきらかになるので、移行前のソフトウェアのユーザモデルの再利用という側面もある。また、差異の種類とその影響が明らかになれば、メニュー変更時の指針としても利用できる。

6. まとめ

本稿では、ソフトウェアの移行をタスクの実行環境の変化として捉え、新しい環境への適応の困難さを以前に適応していた環境との差異という観点から検討した。そして、メニュー構造の差異の種類と移行時のユーザのふるまいについて調査した実験について述べた。今後はメニュー項目の意味を排除した場合におけるメニュー構造の差異の種類による影響の調査と画面配置による影響を調査し、システム上でユーザモデルの実現方法を検討する予定である。

参考文献

- [1] Ben. Shneiderman 著、東他訳：“ユーザインタフェースの設計”，日経BP社。
- [2] D.E.Kieras & P.G.Polson: "An approach to the formal analysis of user complexity.", Int. J. Man-Machine Studies, 22, 365-394(1985).
- [3] A. Y. Lee, P. W. Foltz, P. G. Polson: "Memory for task-action mapping: mnemonics, regularity and consistency", Int. J. Human-Computer Studies, 40, 771-794(1994).