

# ユーザフローを妨害しない ユーザインタフェースの検討

## An Investigation of a User Interface that Does Not Interrupt User Flow

溝上 武史(\*), 高橋 晃(\*\*)

(\*)静岡大学情報学部情報学研究科, (\*\*)静岡大学情報学部

Takeshi MIZOKAMI(\*), Akira TAKAHASHI(\*\*)

(\*)Graduate School of Informatics, Shizuoka University

(\*\*)Faculty of Informatics, Shizuoka University

<論文概要>：本稿では、使いやすいユーザインタフェース (UI) に対しての考察を“ユーザフロー (UF)”という視点から検討する。ユーザフローとはユーザが体験する主観的な作業の流れであり、これは設計者が想定するユーザ導線とは異なることが多い。ユーザフローを妨げるのが悪いUIであり、それは低いユーザビリティにつながると仮定する。この仮定の下に、UIの問題についての従来のような事例をとりあげなおし、UFの観点から再検討することでUF概念の重要性を示す。

キーワード：ユーザインタフェース、ユーザフロー、ユーザビリティ

**Abstract:**In this paper, we investigate a user interface (UI) in terms of "user flow (UF)". The UF is a user's subjective workflow, which may differ from what a designer presupposes. We suppose that a bad UI interfere with the UF, leading to low usability. Under this hypothesis, we reexamine various bad UIs and suggest the importance of the UF.

**Keywords:** user interface, user flow, usability

## 1. はじめに

これまで、ソフトウェアやハードウェアのユーザインタフェース (UI) は認知心理学をはじめとする様々な分野の観点から洗練され続けてきた。しかし、並行して進む多機能化や複雑化も影響し、使いにくいUIは依然として多い。本論文では“ユーザフロー”という概念から使いにくさの原因を捉え直し、事例とともに解決策を探る。

## 2. ユーザフロー (User Flow, UF) とは

ユーザがある道具<sup>1</sup>を利用する場合には、必ずその利用目的が存在する。ユーザはその目的に向けて道具の使い方を学習し、思考しながら道具を利用して、最終的に目的を効率よく達成しようとする。

しかし、コンピュータを代表とする複雑な道具類を利用する際には、道具の利用それ自体に

<sup>1</sup> 本稿では「人間がそれを用いて目的を達成する人工物」を道具と定義する。道具には大工道具のような単純な構造のものからコンピュータネットワークのような複雑なものまでが含まれる。

記憶や注意などの認知リソース(cognitive resource)<sup>1</sup>が消費されてしまい、最終的にユーザが行いたかったはずの目的にたどり着けなくなることも少なくない。

本論文における“ユーザーフロー (User Flow, 以下UF)”とは、道具を使うユーザが利用目的へと到達するための段階過程(プロセス)のことである。これはシステム設計者ないしUI設計者が一方的に想定・設計する“ワークフロー(導線)”とは異なり、現実のユーザが各作業において実際にたどる主観的な経路である。両者は一致する場合もあるが、使いにくいUIにおいては両者の乖離が想定される。乖離した場合には優先されるべきはUFである。

類似の概念として、手続き的知識が挙げられる。しかし、手続き的知識は行動のバリエーションを保証するものであり、手続き的知識があったからといってすぐに、UFが滞りなく流れるとは限らない。たとえば、タッチタイピングを習得しているユーザであっても、キーボードの形式が変わると、UFは滞ってしまう。ここから、手続き的知識を持っているだけでは目的を達成することはできず、道具の性質に即して具体的に実行する必要があることがわかる。そして、その際に、内在する知識と外界の道具とが相互作用をした結果がUFとして表面化するものと考えられる。

また、UFは手続き的知識より動的な側面が強調される。コンピュータ上の概念で例えるならば、手続き的知識は実行ファイルであり、UFはCPUで実行されているプロセス(process)そのものであるとも表現できる。実行できる機能がユーザ内部に保持されていたとしても、それが実際に働かなければ目的を達成することはできない。UFが自然に流れること自体が作業の進行につながる。

## 2.1 UFの性質

以下に、従来の様々な道具利用の観察から導かれたUFの性質を定義する。

### 2.1.1 UFは主観的である

設計者が想定するワークフローは目的に対する最短距離を想定していることが多い。一方、客観的にとらえた場合には、UFは必ずしも目的に対する最短経路になるとは限らず、迂回ないしは逆行することもある。場合によっては目的へたどり着けずに終わってしまう場合もある(図1)。しかし、ユーザの主観においてはUFは常に単一の経路である。

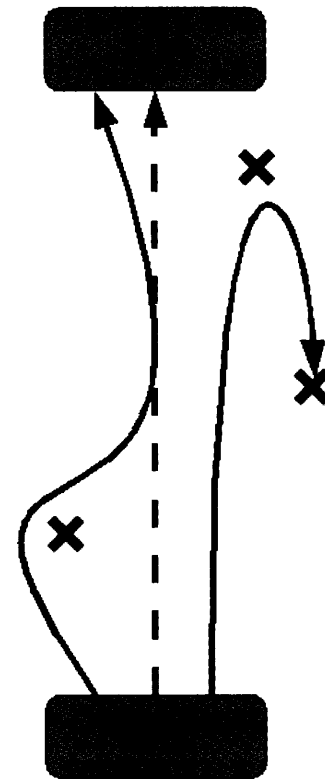


図1 UFの流れ

本来、UFは現在から目的へ一直線に向かうものである。しかし実際には、大きく迂回したり、誤った方向へ進んだりもする。妨害が大きいと目的へたどり着けずに終わってしまうこともある。

### 2.1.2 UFは注意の焦点と認知リソースの限界に制約を受ける

ユーザの意識・注意は常に一つの対象にしか向かないことが多い。ここから、UFは意識的な注意の焦点が遷移することによって表面化すると想定される。

また、ユーザが利用できる認知リソースは有限であり、総量は一定であると仮定している。すなわち、道具やUIそれ自体の理解に認知リソー

<sup>1</sup> 認知リソースとは、ここでは、注意や動機付け、記憶などを含んだユーザの心的なポテンシャルを意味する。認知リソースが不足すると、目的に向かうユーザのフローが止まってしまう(Norman & Bobrow, 1975)。また、この認知リソースを消費することを、本文中では「コストがかかる」と表現する。

スが割かれた場合には、目的とする作業に向けられる認知リソースは減少する。また、注意をそらすノイズが侵入してくれば、やはりその分作業に費やせる認知リソースは減少する。

### 2.1.3 UFの形成は下層から上層へ進む

UFを習得する場合には、訓練による下位の身体・技能レベルの階層から始まり、次に学習によるルールレベルの階層に進み、最終的に目的レベルの階層へ移行する（後述 第3節参照）。

UFは、学習初期には貧弱だが、作業を繰り返すなかで一定の形に形成され、行為が自動的に進行するようになる。また、繰り返し行われて慣れた動作は、意識的に行わなくとも済むように自動化される。自動化されたUFの遂行は認知リソースの消費が少なくなる（Anderson, 1980）。

### 2.1.4 UFは最小の労力で最大の効率を得るように進化する

ユーザは最小の手続きで最大の効率あげることを目指す。したがってユーザは無駄な認知リソースの消費は望まない。時間についても、可能な限り短い時間で行おうとする。

例外は、新しいUIの学習コストがそのUIを利用することによって節約できるコストを下回ると予想される場合のみである。

ここから、UFには一種の“慣性”が存在することもわかる。ユーザは新しいUFに出会った場合にも、すぐに新しいUFを採用することをせず、自分が保持しているUFを利用し続ける傾向がある。UFの変更は、新しいUFを採用する場合に節約できる認知リソースと、旧来のUFをキャンセルして新しいUFを採用する際に必要になる認知リソースとのバランスに依存する。もし新しいUFの採用に大量の認知リソースが必要であり、一方で採用した結果の節約量がその認知リソースの量を下回ると（主観的に）考えられる場合には、ユーザは旧来のUFを利用し続ける。

### 2.1.5 UFは入れ子状になる

設計者が想定するワークフローは、作業における障害がない状態で記述される。しかし、現実の作業では、ある目的を達成するために、別の目的を達成しておかなければならない事態はしばしば生じる。

このとき、新しく発生したUFを“サブUF”と呼ぶ。このサブUFは、これ自体が一つの目的を持ち、このサブUFを解決しないと本来のUFに戻れない、という性質を持つ（図2）。図2の例では、もしもコンピュータをネットワークに接続できない場合には、プリンタドライバが入手できず、そこでUFは進めなくなってしまうため、接続のために多くの処理資源が割かれてしまい、そこでUFが途切れてしまう。

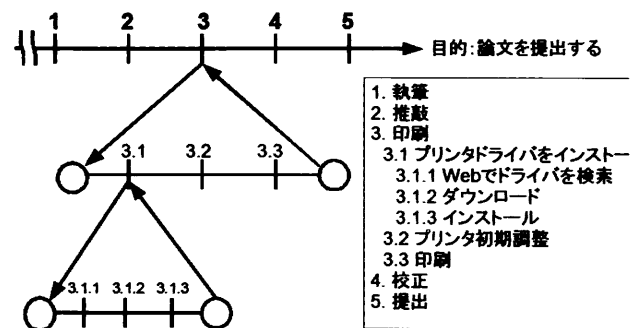


図2 サブUFの発生

多くの場合、UFは入れ子構造をなす。“論文を提出する”という最上位のUFの途中で“印刷”のサブUFが発生する。そのUFを進む過程で、さらに下位の“プリンタドライバをインストールする”というサブUFが発生する。

また、上記の例のように、サブUFは入れ子状に発生する可能性がある。そのため、深い階層のサブUFに入り込んでしまうと、その階層のUFを持続させることに資源が消費され、元の目的レベルのUFに戻れなくなる可能性が高くなる。

### 2.1.6 UFは各ユーザの中にある

UIは道具の側にあるが、UFは人間の側にある。本論文では、UIの目的はUFと道具の間にある乖離を少しでも小さくするためにあるもの

ととらえる。

道具や“ワークフロー”とUFとの間に乖離がある場合、優先すべきはUFである。ユーザの持つUFの性質を知らずに、よいUIを作り上げることはできない。これを反対側からとらえると、よいUIとは各ユーザのUFに各々最適化されたものとなり、すべてのユーザのすべての局面において優れたUIは存在しないといえる。すなわち、無条件に良いUI、ないし無条件に悪いUIというものとは原理的に存在せず、すべてユーザとの相互関係に依存すると考えられる。

一例としてCharacter User Interface (CUI) が挙げられる。CUIは、一部のUI研究においては悪いコンピュータ用UIの代表例のように扱われている場合があるが、むしろ最初からCUIに慣れてきたユーザにとってはGUIよりも作業効率が高いことがあることに注意する必要がある。

以上のように、UFは客観的に設計された“ワークフロー”とは異なる性質があり、これらの性質を踏まえた上でよいUIを考察しなければならない。

## 2.2 UFとユーザビリティ

UFが種々の理由で目的から遠ざけられたり、大きく迂回させられたりすることを“UFが妨害される”ととらえる。このUFの妨害が、ユーザに主観的な“使いにくさ”を感じさせ、また、作業効率の低下などの問題を発生させる要因になっていると仮定する。

一方、UFを妨害しない、あるいは妨害してもすぐに元のUFにユーザを戻せる場合には“UFが保護される”ととらえる。UFが保護されたUIは、ユーザにとって高いユーザビリティを持つと仮定する。

## 2.3 本論文の目的

本論文は、複雑な道具を使う際に発生する様々な問題点を“UFの妨害と保護”という観点からとらえなおし、現在の各情報機器の具体的

なUI製品について再検討を行うこと、ならびに様々な場面におけるUF概念の有用性を検討することを目的とする。

## 3. UFの3階層

日常的に利用しているソフトウェアにおいても、UFが妨害されていると思われるUIは数多く存在する。以下に、そうしたUFの妨害事例について、UFを3つの階層に分類することで詳細に検討する。

本論文では、人間の行為の3階層モデルを援用して、UFを3つの階層に分類する(Rasmussen, 1986)。この理由は、この3階層モデルによって、様々な人間の行動を“戦略・戦術・技能”という水準の異なる階層から切り分けることができるため、UFの改善に効果的な提言が可能になると考えられるためである。

1つ目はもっとも低次のもので、技能・身体運動のレベルである。これは、キーボードやマウスなどの単純な操作に該当する。1ボタンマウスか2ボタンマウスの違いによる操作の相違や、トラックボールやホイールマウスの操作の違い、キーボードのキーの数やキー配列による操作方法の違いといったことがこれに相当する。

2つ目は規則・ルールのレベルで画面UIの操作に該当する。これはOSやソフトウェアごとに異なる操作の流儀があり、同じ目的を達成するのにソフトウェアによって異なる操作ルールが要求されるような場合に問題が生じる。

最上位の3つ目は目的のレベルで、ユーザがそのソフトウェアで実現したい状態に向かう直接的な行為である。ユーザは本来、この階層においてのみUFを進行させることで目的に向かって進むことができるが、様々な理由でこの階層から下位の階層に注意の焦点を移す必要が生じる(図3)。そのたびに目的レベルのUFは止まり、それより下位のUFを進行させる必要が出てくる。

以下において、様々なUIの問題点をこれらの

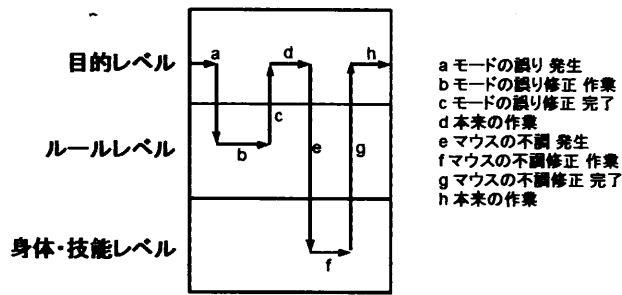


図3 3階層にわたる焦点の移動

目的レベルで進行してきたUFがaで妨害が発生し、ルールレベルに落とされる。bでその原因が判明・修正されてUFが目的レベルに戻る(c)。dで再び目的レベルでUFが進行するが、その後eにおいて、今度はUFが身体・技能レベルまで落とされる妨害が発生する。fで再び妨害の修復が行われ、UFは目的レベルに復帰し(g)、進行する(h)。

3階層の観点から解釈を行う。

なお、この3階層は厳密に区分されるものではなく、一つの問題に複数の階層が関与していることも多い。特に、上位の階層の問題はそれより下位の問題を内包して出現するものと考えられる(ルールレベルのエラーによって無駄なマウス操作を繰り返すなど)。以下では便宜的にある特定の要素が強いと考えられる例を特定の階層に位置づけている。

### 3.1 身体・技能レベル

技能レベルの階層のUFは、主としてUIの物理的な構造に影響を受ける。コンピュータとの関係では、人間が直接触れるUIの性質や、画面と視線の移動、過去経験によって得ている技能などが問題となる。身体・技能レベルの行為は訓練によって自動化されるため、問題が生じない限り、このレベルのUFは通常意識されることがない。

以下にこのレベルで生じうる問題の例を挙げる。

#### 3.1.1 キーボードとマウス

現在日本で利用されているキーボードには日本語106キーボードや英語101キーボードなど、

キー配列がわずかに異なった規格が多数存在している。複数のコンピュータを使い分ける場合に、これらのキーボード配列の相違により習得したタッチタイピングの技能は無効化ないしは弱体化させられてしまう。これは、従来習得していた身体技能が、似ているけれども異なる動作の干渉を受けることによる。

また、マウスの“すべり”が悪く、ユーザのイメージどおりにポインタが動かない場合などは、ユーザはマウスの清掃やマウスパッドなどの物理的な環境を整える。新しいPCを利用する初期などは、マウスユーティリティを利用して自分の操作感覚に見合った移動速度などを選ぶ初期設定を行う。これは、自分の保有する技能に適合するように、道具を調整することであるといえる。

#### 3.1.2 マニュアル・ヘルプを見ながらの操作する際のトラブル

アプリケーションを使う場合に、その印刷マニュアルやヘルプを見ながら操作を行うことがある。この場合、画面と印刷物の両方の間に頻繁に視線を大きく移動させる必要があり、操作のUFは断続的にならざるを得ない。

特に、マニュアルのページを左手で支えながら右手でマウスを操作するなどといった状態の場合には、視線の移動に加えて抑える動作に対する注意も必要されるため、アプリケーションに対するUFの持続性は大きく落ちる。

#### 3.1.3 ゲームのUIのユーザビリティ

ゲームなどでは、内容に応じて現実に近い形状のUIが利用されることもあり、一般的にはそのほうがパフォーマンスは良いとされる。

しかし、実際には必ずしも現実に近いUIが望ましいとはいえず、ユーザがすでに持っている身体技能に合ったUIの方が良いパフォーマンスを示すこともある。たとえば、自動車レースゲームの専用UIは一般的にハンドルの形状をしているが、他のゲームで一般的なパッドUIに慣れて

いるユーザの場合には、むしろ慣れたUIの方が楽であるという<sup>1</sup>。また、対戦格闘ゲームのような、ユーザがそもそも技能を持っておらず、類似動作を行えない場合には、むしろ“ゲーム固有のUI(ジョイスティックなど)”の方がユーザビリティは良い(cf. Wii スポーツのボクシングは実際にユーザがボクシングの動作を行うものだが、格闘ゲームのUIとしてはむしろ違和感がある)。

ここから、物理的なUIのユーザビリティの良し悪しは、UIの物理的な属性それ自体にあるのではなく、それをを用いたユーザのUFが流れるかどうかによって依存することがわかる。

上記のように、身体・技能のレベルにおけるUFは物理的なUIの形状、大きさや反応速度など、UIの物としての側面に影響される。

## 3.2 異なるルールベースによってユーザが混乱する例

ルールレベルの階層は、様々なUIの操作ルールが異なることによって影響を受ける。この場合、身体・技能レベルの行為は正しく行われているが、それを適用するルールを誤ってしまうために、結果的にUFが妨害される。このレベルのUFは学習によって自動化され、やはり問題が生じない限りほとんど意識されない。以下に、こうしたルールベースでの問題が発生するUIの例を挙げる。

### 3.2.1 モード

一部のソフトウェアには“モード”が存在する。モードとはUIの一種で、あるモードに入ると特定の操作命令を受け付けるが、モードから抜けると受け付けなくなるような、ソフトウェアの状態の変化である。

UNIX上の代表的なエディタであるviにはこのモードが存在する。しかし、初期のviにはこのモード状態を示すフィードバックがなく、ユーザがコマンドを入力するモードにいたつ

りで、実際には文字入力モードに入っていたために、文面上にコマンドそれ自体が打ち込まれてしまう状態が頻繁に見られた。

これは、モードに応じて操作ルールが変化するが、UFはその変更に沿って変化せず、旧モードのルールを適用して操作をしてしまった状態を示唆している。こうした事態が多発すると、ユーザは常にモードを意識してルールを学習する必要がある。しかし、外見が同じ刺激に対して、状況に応じて別々の反応セットを学習することは困難である。このように、外見の手がかりが同じUIに対して異なるルールを適用する場合には、干渉が発生するためユーザのルールの学習コストは大きい。

### 3.2.2 Meadow

EmacsというUnix上のテキストエディタのWindows移植版として、Meadowという多機能エディタがある(図4)。このソフトウェアは標準的なWindowsユーザからは使いにくいという感想を持たれることが多い。特に、操作体系に対する違和感は、ある程度コンピュータ利用に慣

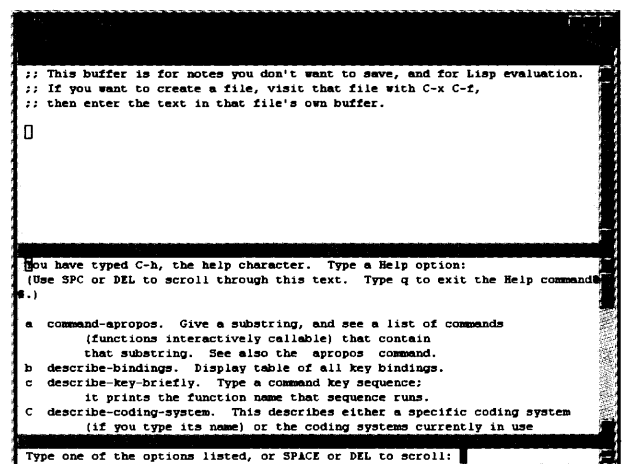


図4 テキストエディタ“Meadow”  
“Ctrl+X, Ctrl+C”などの2ストロークと呼ばれる連続したキー操作は、Windows上のソフトではまず見かけない。逆にWindowsで一般的なキー操作は利用できない。また、ウィンドウの中が分割されるバッファや、設定をすべてテキストファイルで行う点も、Windowsではあまり見られない特徴である。

<sup>1</sup> 静岡大学ゲーム研究会 私信

れたユーザでも強く感じられる。

この理由としては、Meadowそれ自体の機能の複雑さに加え、マウスを利用せずにキーボードを多用する利用ルールの相違による、ルールレベルでの妨害が発生しており、二重の認知的負荷がかかっていることが原因であると考えらることで解釈がしやすい。

機能としては、同じテキストエディタであるメモ帳のような外観を呈してはいるものの、その操作ルールはまったく異なっている。ファイルを開くあるいは終了するという基本操作をとっても、OSであるMicrosoft Windows標準の操作体系とは異なったルールを適用せねばならない。

Meadowの利用訓練後でも、習慣的にWindowsのUFに沿って“手が動く”ことも多く、Meadow固有のルール（特殊なキーバインド）と干渉する。逆に、一度Meadowの利用ルールが強く学習され、UFが自動化されると、今度は他のソフトウェアでもその操作ルールが意図せずに出現してしまうこともある。こうした状況は、コンピュータ初心者が両者のルールの使い分けを学習する際に大きなコストがかかる原因となるだけでなく、ある程度コンピュータ利用に慣れた上級者においても恒常的な負荷となる。

画像系ソフトのGIMP for Windowsも、元の出自がWindows系OSではないことから、ファイルダイアログの表示がOS標準の形状と大きく異なり（図5）、そのために利用の際にその点だけルールが異なる。そのため、GIMPを使うユーザはファイル操作の際に新しいパターンのUFを作ることになり、コストがかかる。

Microsoft Windowsの“コマンドプロンプト”も、Windows上のプログラムでありながら、ウィンドウの中で右クリックをした場合の反応や、キー操作に対する反応が、ほかのものと大きく異なっている。このことは、コマンドを入力するという操作体系の違いと同じくらい、UFを妨害するものである。

こうした“ルールの異なるUI”をもったソフ

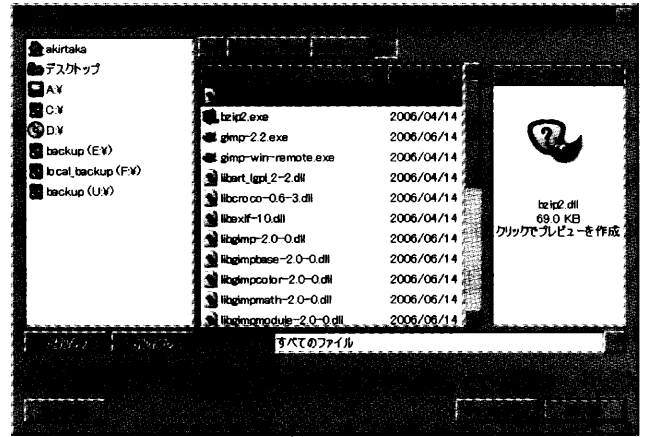


図5 画像編集ソフト“GIMP”のファイルを開くダイアログ  
通常のWindowsのファイルを開くダイアログとは異なったUIであり、ユーザは混乱する。

トウェアは、利用方法を学習する際に、その“機能の学習”に加えて“利用ルールの学習”を重ねて行う必要があるため、ユーザにとっての負荷は高い。また学習後にもルール同士の干渉が発生するため、慣れた後でも使いにくさを感じることが多い。

### 3.2.3 似た機能の別なソフトウェア

オフィススイートと呼ばれるソフトウェアの中には、表面的には似た機能を持ちながら、操作体系が異なるものも多い（例：Microsoft OfficeとOpen Office、Star Suiteなど）。これらを並列的に利用する場合には、達成できる結果は類似しているが、そこに至るまでの操作ルールが異なるために、ユーザは大変混乱をきたす（表1）。

たとえば、プレゼンテーションソフトにおいて、新しくスライドを作成するための手続きが、Microsoft PowerPointの場合にはControl-M（コントロールキーを押しながらMキーを押す）であるのに対して、OpenOfficeのPresentationでは標準状態ではキーバインドが存在せず、マウスを利用してしか命令ができない。これは自動化された操作部分では身体技能レベルと解釈できるが、意識的な操作においても同様のエラーを発生させることから、ルールレベルの妨害であると考えられる。

逆の事例に、KINGSOFT Office というソフト

表1 Microsoft Office と Open Office

ともにオフィスソフトであるが、操作におけるルールは互いに異なる。ここに挙げた相違は一例である。外見が似ているのに対して操作ルールが異なると、ユーザーは強いストレスを感じる。

	Microsoft Office	OpenOffice
新規スライド挿入のキー操作	Ctrl+M	なし
箇条書き記号の出現ルール	文字が入る行にのみ、箇条書き記号が出現する。	文字の有無にかかわらず、箇条書き記号が出現する。
オブジェクトの選択ルール	選択可能な場所に遊びがあり、かつ視覚手がかりが多い。	選択可能な場所が極めて限られており、かつ視覚手がかりに乏しい。
フォント変更のルール	オブジェクト自体を選択すれば、オブジェクト内部のフォントの属性を変更できる。	オブジェクト内部のフォント自体を選択しないと、属性は変更できない。

ウェアがある<sup>1</sup>。このソフトは、一見すると Microsoft Office と見間違えるほどにその外観がよく似ている。さらに外見ばかりではなく、メニューの配置やアイコンのデザインはもちろん、詳細なキーバインドやファイル形式に至るまで Microsoft Office のそれに酷似している。そして、実際の操作ルールも Microsoft Office とほぼ同じであったために、ユーザは両者の区別がつかないほどであった。すなわち、Microsoft Office のユーザが KINGSOFT Office を使う際には、ユーザは新しいルールの学習を必要としないため、自然に乗り換えることが可能である。これは同じオフィススイートである Open Office にはない特徴である。

ここから、ユーザは、UFが同じように流れて作業ができ、目的が達成できるのであれば、ソフトウェアの内部構造に関しては意識しないといえる。ユーザはソフトウェアを利用する際に、その名称や内部構造よりもUFを重視するが、一方でプログラムやシステムの設計者がそれまでのUFと適合しない各ソフトウェア固有のルールを強制してしまうこともある。そのような場合には、ユーザにとってはそのソフトウェアのユーザビリティは低いものととらえられる。

### 3.2.4 バージョンアップによる混乱の例

あるソフトウェアが新しいバージョンに上が

る際には、何らかのインタフェースの変更が行われることが多い。たとえば、Microsoft Office は“2007”バージョンにおいて、それまでの画面インタフェースを大きく変更し、リボンと称する新しいインタフェースを取り入れた(図6)。リボンは多くの機能を整理して表現していたが、しかしユーザがこれまで利用してきた機能がどこに配置されるのかという配置ルールが大きく変更されたため、既存ユーザには混乱をもたらした。

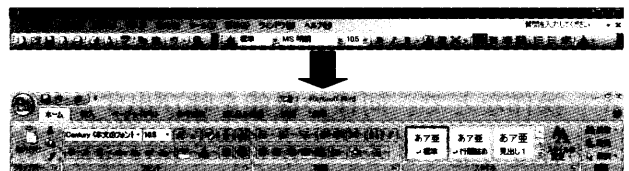


図6 2003までのUIと、2007での新UI“リボン” Office2003からOffice2007へのUIの変化は“リボン”の導入であった。Microsoftはユーザビリティの向上をうたっていたが、ユーザはUIの利用ルールが変化したために混乱した。

これは、既存のユーザの内部にある“利用ルール”から発生するUFを無効化したことによるものと解釈できる。新しいUIの導入によって旧来のUFとの整合性がなくなる、ないしは弱くなる場合には、導入の条件として、新しい学習プロセスの負荷や旧ルールとの干渉による損失以上に、新しいUIの習得後に作業効率上がるということがユーザに理解できる形で提示されなければな

<sup>1</sup> KINGSOFT <http://www.kingsoft.jp/>



らない。

また、単に見た目が変わっただけでユーザが利用する機能が従来と同じ場合には、新しいUIの導入は既存のUFを破壊するだけの意味になるだろうと考えられる。新しいUIの利用ルールを学習し、旧UIと同様の水準にまでUFを自動化させるためには、ユーザが認知リソースを投入して学習を行うことを強制する。これは、上述の“似た機能の別なソフトウェア”と同じ状況であり、ユーザは既存のUFを有効に生かした作業を行うことができないため、あるソフトウェアに慣れたはずのユーザを、初心者と同様の水準にまで強制的に引きおろすことになる。

ここから、新しいUIを導入する場合には、可能な限り既存のUIに戻せる仕組みを用意しておくことが望ましいこともわかる。あるサードパーティはOffice2007のUIを旧バージョンに戻すためのソフトウェアを販売している<sup>1</sup>。

### 3.2.5 操作制限のある Web ページ

Web ページの中には、ページの中で右クリックをすることができないものがある。右クリックをすると、それを禁止する旨のポップアップウィンドウが表示されるものもある。

右クリックができなくても、メニューバーやキー操作などで、同様の機能呼び出すことはある程度可能である。しかし、それは通常の操作ではないためスムーズに行えるとは言い難い。遠い上部メニューまでのマウス操作や、慣れないキー操作が必要である。また、右クリックメニューからしか呼び出せないような機能も存在している。

Web ページの作成側としては、そうした機能を使うことを禁止する目的があったものと考えられるが、問題は、それ以外の機能まで禁止されてしまう点にある。一般的な操作ルールが適用できないという点で、こうしたページはルールレベルの問題を内包していると考えられる。

厳密にはWebページの振る舞いとは言えないが、操作が制限されるという意味では、ブラウ

ザウィンドウの中でPDFファイルを開く場合もこれに該当する。PDFファイルへのリンクをクリックしたとき、Adobe ReaderなどのPDF閲覧ソフトが新たに起動するのではなく、ブラウザの中に、あたかも通常のHTMLリンクをクリックしたかのようにPDFファイルの内容が表示される。しかしこの状態になると、一部の操作、たとえばウィンドウを閉じる場合のControl+Wというキーバインドなどは使えなくなってしまう。押したとしても「外部ウィンドウではこの操作は実行できません」というエラーメッセージの書かれたダイアログ・ボックスが表示される。PDFに限らず、何かブラウザ内部に開かれた場合には、ユーザはブラウザ利用のルールを適用して操作しようとする。そのため、内部に開かれたものがブラウザの操作ルールに干渉すると、ルールレベルのエラーが起こる<sup>2</sup>。

### 3.2.6 特殊なウィンドウ

アプリケーションの設定ファイルの中には、一般的なGUIのウィンドウの操作ルールが適用できないものがある。たとえば、通常のGUIウィンドウはその大きさを自由に変更することができ、またフォーカスを外して背後に回すことができることがルールとなっている。しかし、アプリケーションの設定ウィンドウの多くは、一度設定ウィンドウを表示するとその大きさを変更することができず、また「OK」「キャンセル」を押して終了するまでフォーカスを完全に奪うことが多い。これは、ウィンドウ操作という基本ルールを破った特殊なウィンドウであり、ユーザがこれを扱う際にはややストレスがたまる。

ソフトウェアの設計者は、その設定画面の途中から抜け出すことをさせないために意図的に制限をかけている可能性もあるが、しかしその期待した導線を通じてUFが流れるとは限らない。

上記のように、ルールレベルのUFは認知レベ

<sup>1</sup> 株式会社マグノリア “Back to 2003” <http://www.magnolia.co.jp/back2003/>

<sup>2</sup> 以前のバージョンではデフォルトのドラッグの挙動がテキスト選択ではなく、手のひらツールになっていた。

ルの情報処理に基盤を置いており、記憶、注意や判断などの人間の情報処理特性の限界に制約を受ける。

### 3.3 目的レベルでユーザが混乱する例

目的レベルの階層は、ユーザが道具を利用したい目的そのものへの道筋である。ユーザは本来このレベルでのみ意識的にUFを進行させることで、目的に向かって効率よく進むことができる。

なお“正しい目的”はユーザが決定すべきものであり、その目的に向かってうまく導くUIないしは妨害しないUIがこのレベルでのよいUIとなる。もともとユーザが決めた目的が誤っている場合には、UIでの修正はできない。その誤った目的に対して進む手助けをするだけである。

#### 3.3.1 元のUFに戻れない

UFは、ある程度正しく進んだところで、誤った方向に進むこともある。その場合でも、誤りに気がついた段階ですぐに直前のUFに戻ることができれば、事実上問題は発生しない。しかし、UIの性質によっては直前のUFに戻ることができないこともあり、ユーザにストレスを与える。以下にその具体例を示す。

##### 3.3.1.1 待ち時間の長さ

UFを進行させるためには注意を連続的に保持する必要があり、そのために認知リソースを消費する。ここから、長い待ち時間もまたUFを失わせると考えられる。ユーザがWebページを閲覧する際に、待ってられる時間は8秒であると言われている<sup>1</sup>。これは、それ以上の待ち時間の場合には、目的へのUFが保持しにくいためではないかと考えられる。ユーザがある持続時間以上待たされて、まだ作業に戻れない場合にはそのUFは失われ、本来向かうべき目的とは別の目的に向かうUFが発生する可能性がある。

一般にコンピュータやシステムは挙動がすばやいほどユーザにとって望ましい。この根本的

な理由は、すばやい挙動はUFを継続して保たせるために有効であることによると考えられる。反対に、遅いコンピュータでの作業においては、長い待ち時間を経ているうちにUFが失われる可能性は高い。

#### 3.3.1.2 専用インストーラ

Microsoft Windows (Windows) で、ソフトウェアをインストールするとき、インストーラと呼ばれるプログラムが実行される場合がある。その際に、タスクバー領域を含むデスクトップすべてが青い背景に切り替わり、一番手前側にウィザードと呼ばれる形式のウィンドウが表示されることがある (図7)。

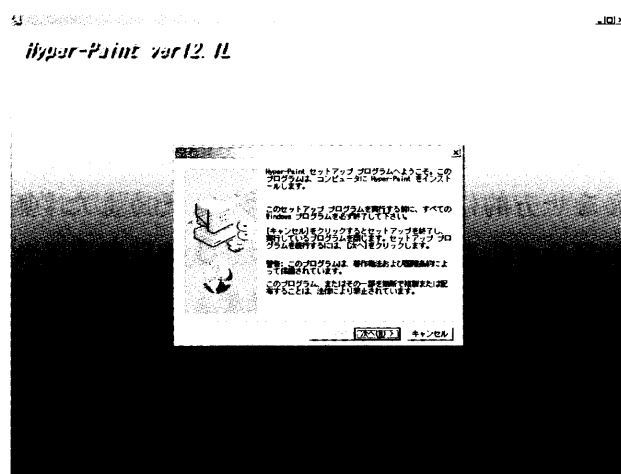


図7 全画面インストーラ

このインストーラでは、インストールを中断すると再びインストールする際には、最初からやり直さなくてはならない。

インストール作業の間、ユーザはそのウィザードしか操作することができない。ウィザードに入力するために必要な情報の書かれたテキストファイルがディスク上のどこかにあるような場合は、いったんウィザードを終了しなければならず、インストールを中断せざるを得ない。一度中断したインストール作業は途中から復帰させることができない場合が多いため、あらためて最初からはじめる必要がある。目的へのUFが途中で途切れて、復帰できなくなってしまうことから、目的が達成できなくなる。

ここで、インストーラの起動中にもファイル参照が可能である場合にはユーザは一時的にイ

<sup>1</sup> Zona Research, The Need for Speed II, April 2001. [http://www.keynote.com/downloads/Zona\\_Need\\_For\\_Speed.pdf](http://www.keynote.com/downloads/Zona_Need_For_Speed.pdf)

ンストーラ画面から退避し、必要な情報を参照した上で“インストール作業”に戻ることができると、UFの流れは確保される。

### 3.3.1.3 Web ページの書き込み

記入が必要な Web ページの中には、書き込んだ後の確認のページを設けているものもあるが、その際に、内容の修正のために元のページに戻ると書き込んだ内容がすべて消去されてしまうものがある。その場合には書き込みを最初からすべてやり直さなければならない。一部を修正するだけでも直前の状態には戻れず、すべて最初から書き直さなければならない場合、ユーザは非常に大きなストレスを感じる。

## 3.3.2 目的に至る道筋をさえぎってしまう

目的レベルのUFが進行している最中に、それらを一方的にさえぎるUIも多い。そうした“無神経なUI”はユーザに嫌われ、次第に利用されなくなる。以下にその例を示す。

### 3.3.2.1 ポップアップウィンドウ

WebブラウザでWebサイトを閲覧しているとき、突然新しいウィンドウが開き、画面の全体ないしは一部が覆い隠され、そちらのウィンドウに強制的にフォーカスが奪われることがある(図8)。

これらはポップアップウィンドウと呼ばれ、しばしば広告表示や、システムによるユーザへの注意喚起の目的で使われる。

しかし、ポップアップウィンドウは、表示する前に適切な教示をしないと、突然現れてユーザの作業を中断させてしまうことになる。また、ウィンドウなので背景を隠してしまうという問題もある。さらに広告目的の場合は、動いたり、消しても再度表示されたり、うるさくつきまとうようなものも少なくない。

ユーザが行いたい作業を乗っ取ってしまい、その割り込んだ作業を優先的に行わざるを得なくする、という点で、これは目的レベルのエラーを誘発させるUIであると考えられる。

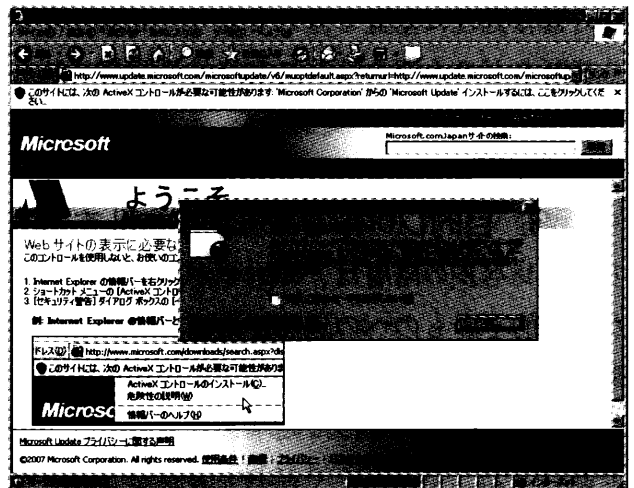


図8 ポップアップウィンドウ  
ポップアップウィンドウは、ユーザの意図と関係なく画面の最前面に現れ、強制的にフォーカスを奪う。ユーザ文脈はここで中断されてしまう。

ポップアップウィンドウは意図的にユーザの注意を引くようになっているため、表示されると現在のUFをいったん中断して、意識をそちらへと向けざるを得ない。OKボタンをクリックするなど、ウィンドウの指示に何らかの応答をしたり、ウィンドウを閉じたりといった、本来の目的とは無関係な作業が必要になる。こうした作業によって目的への道筋がねじ曲がってしまったり、道筋を見失ってしまったりするという事態が起こることもある。

そうしたユーザの声を反映してか、ポップアップウィンドウを表示しないようにする機能が登場した。Microsoft Internet Explorer (IE)をはじめ、現在における主要なブラウザは、その多くがポップアップ・ブロッカーを標準機能として搭載している。これは、やはりポップアップウィンドウがユーザのためになっておらず、使いにくさを持ったUIであったことの現れであると考えられる。

### 3.3.2.2 過剰な表現の Web ページ

Webページの中には、開いた瞬間にFlashなどの動画が自動的に再生されるように設計されているものがある。これを停止して目的に向かうためには、あえてキャンセルボタンを押す作業が必要になる。こうした作業はユーザのたどり

着きたい目的に向かう UF に対して妨害的に働く。そのため、このようなページをトップに位置づけている Web サイトはユーザに嫌がられ、次第にアクセス数が減少する。ないしは、ユーザは動画ページを迂回するアクセスをとるようになるため、そうしたページの実質的な意味はなくなるものと考えられる。

### 3.3.2.3 ソフトウェアによる介入

ソフトウェアによっては、ユーザの振る舞いを予測し、条件が満たされると自動的にそのモードに入っていくものがある。たとえば、Microsoft Word では、箇条書きのマークを一つ入れると、次の改行後にも自動的に箇条書きのマークが入るような仕組みがあった。

しかし、その箇条書きモードからの脱出方法は明示されず、またその箇条書きのマークが通常の文字とは性質が異なり、削除することができない。このため、本文執筆をしていたユーザはその強制的な箇条書きモードから抜け出すために様々なメニューを調べなければならない。その結果、UF は本来の目的から大きく逸れる。

同様の内容は Apple Mail のメール引用部分にも存在する。引用部分を示すマークが通常の文字ではないために、キーボード上の削除操作やマークしての領域削除では消すことができない。このため、ユーザは強制的に引用部分を編集する方法を模索する、あるいは不自然な手順でそうした引用領域を発生させない方法を見出さざるを得なくなっている。これは、本来の目的である“メールを書く”という目的への UF とは異なった方向にユーザを強制的に方向付けており、不要なサブ UF (後述 3.3.3.1 参照) を出現させているともいえる。

これらのように、目的にいたる道筋をさえぎる UI は UF の流暢性を損なうため、ユーザにとって悪い UI であるといえる。

### 3.3.3 目的を乗っ取ってしまう

もっとも問題のある UI は、ユーザに目的を見失わせる UI である。これは、それまで費やして

きた認知リソースを空費させ、ユーザに無力感を味あわせることで、その後の道具の使用から遠ざける原因になる。以下のそれらの例を挙げる。

#### 3.3.3.1 UF の迷路

複雑な道具であるコンピュータを使っていると、様々な“サブ UF”に入り込んでしまうことも多い。

たとえば、印刷を行いたいのが、印刷するためにはプリンタ設定を行わねばならず、そのためにはプリンタドライバをダウンロードしなければならない、そのためにはコンピュータをネットワークに接続しなければならない、そのためには使っていなかった無線 LAN の設定を行う必要がある、そのためには接続のための WEP キーを入力する必要がある……などの事例である。

こうした“サブ UF”は、階層化が深く進むにつれて認知リソースを多く消費し、ユーザに元の目的を見失わせたり、あきらめさせたりする傾向がある。ある程度以上の認知リソースを消費すると、ユーザはもとの UF に戻ることができなくなる。また、サブ UF の解決が最後までできないこともあり、その場合には、本来達成すべき目的の UF もそこで途切れる。こうしたシステムはユーザを迷路の途中に置き去りにしてしまう。

#### 3.3.3.2 ヘルプの過剰

ソフトウェアのヘルプの中には、ヘルプ内部の別な項目への参照 (リンク) を持っているものがある。ユーザがヘルプを利用する際に不足する情報がある場合にそのリンクをたどって細く情報を得ることができるものである。

しかし、元のヘルプ情報が不完全で、ユーザの目的達成をサポートしない場合にそのリンクによって、ヘルプからヘルプを渡り歩く現象が発生する。その結果、多くの場合に、ユーザはヘルプを読むこと自体に認知リソースの多くを割かれてしまいいつの間にか本来の目的を達成しないままに作業を終えてしまうことがある。これは目的を乗っ取ってしまうタイプの UF の

妨害である。

これと同様のことが、Microsoft Office 2000よりマイクロソフトが導入していたエージェント型ヘルプ(通称“イルカのカイル”)にも当てはまった(図9)。これは、ユーザがソフトウェアで新しい機能を利用すると突然画面にキャラクターが出現し、機能について説明を一方向的に始めるというものであった。

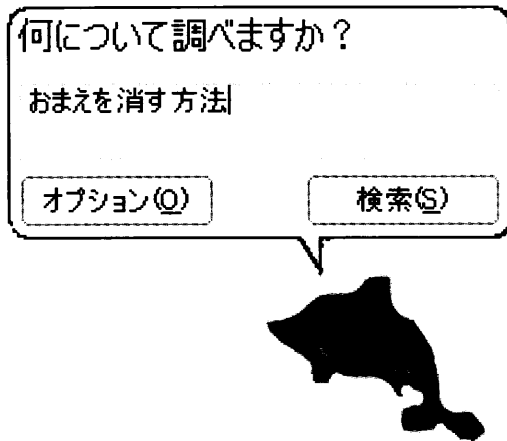


図9 エージェント型ヘルプ  
悪評の高かったイルカ“カイル”。Office 2000から採用されたが、Office XP以降は見かけなくなった。

このキャラクターは表面的には親しみやすいものだったが、実際の使用に際しては、強制的にマウスのフォーカスを奪うことや、このキャラクターにどのように指示を与えるのかについて(特にこのキャラクターをどのように消すのか)という点についての情報がなく、ユーザはこのエージェントの操作法を調べることに注意を移さざるをえなかった。その結果、ユーザが本来行いたかったはずの作業は中断され、またエージェントの操作法を調べることで自体に資源が振り向けられることが多かった。さらに、初心者ユーザには最終的にこのイルカを消す方法がわからないことが多かったため、ユーザの一部はソフトウェア自体を終了してしまい、目的としていた作業が最後まで遂行できなくなる現象も観察された。

なお、このエージェント型ヘルプは次版のOffice XPよりデフォルトでは表面には登場しな

くなった。これも、この種のヘルプがユーザに対してストレスを与えていたことの証拠と考えられる。

これは、本来の目的達成をサポートするための道具の機能が、むしろ多くの認知リソースを消費してしまい、その結果として本来のUFを乗っ取ってしまったという悪いUIの事例である。

### 3.4 まとめ

上記のように、一見するとさまざまな要因があるようにとらえられる“使いにくい道具”にも、ユーザの視点からは“UFの妨害”という共通した問題点があることがわかる。すなわち、技能レベル・ルールレベル・目的レベルのいずれにおいてもUFの妨害があり、各レベルにおいてそうした妨害を取り除くことでよいUIが実現できると考えられる。

## 4. UFを保たせる・妨害しない工夫のあるインタフェースの例

一方で、従来のUI研究の成果から、UFを補助する、あるいは妨害しないように改善されたUIもある。これらは、元来は個別の理由に基づいて行われてきた改善であるが、UFという観点から統一的にとらえなおすことで、より明確な意義づけが可能であると考えられる。以下にその例を挙げる。

### 4.1 やり直しができる

誤った操作をした際に、すぐに元の状態に戻ることができる機能が用意されていることが多くなった。それ以前は、誤った操作をした直後でも戻ることはできず、ユーザは誤った操作を修正するのに大きなコストがかかっていた。そのため、修正のサブUFが発生し、それが終了するまでは元のUFには戻れなかった。現在では、簡単にやり直しができる機能が一般的になり、すぐに元のUFに復帰できるようになった。

これは正しいUFを保護するという観点から、優れたUIであるとみなすことができる。

### 4.2 フィードバックがある

自分の行った行為に対して、その結果のフィードバックがあるUIは使いやすいUIである (Norman, 1988)。この理由は、フィードバックがあることで、ユーザは自分の行った行為の結果を即座に知ることができ、UFを継続的に構成することが可能なためであると解釈できる。フィードバックが欠如したUIは、その段階でのUFの正しさがわからないため、UFが失われる危険性を含んでいる。

しかし、遅いフィードバックはUFの進行を足止めする可能性があるため、フィードバックの遅延時間はUFを失わせない程度の時間が望ましい。人間の現在は“約2秒”であるとする見解もある (海保、2000) ことから、このフィードバックの遅延も2秒以内であればUFが保たれる可能性があると考えられる。

### 4.3 目に見える

同様に、操作対象とその変化が目に見えることもUIの構築に大切である。この例にジニーエフェクト (図10) がある。ウィンドウを開いたり閉じたりする際に、速度は遅くなるが、あえてアニメーションの効果を見せる場合がある。技術的にはより速い実行が可能であるが、あえ



図10 目に見えるUFのつながり  
MacOSXから採用された“ジニーエフェクト”。慣れないユーザに対して、UFをつなぎ止める効果があった。しかし、ユーザが慣れた場合には妨害になりかねないため、この効果を切ることもできた。

て速度を遅くし、変化の様子を可視化することでユーザにどのような変化がおきたのかを示し、そのUIに慣れないユーザのUFを保つことができるようにしたものである。

しかし、操作に慣れたユーザはすでにUFが出来上がっているため、こうした視覚効果が不要になる。その場合には、その効果を切ることができるになっている。これもUFの変化に合わせたよいUIである。

### 4.4 展開型UI

MicrosoftのWindows Updateサイトなどで使われている、“+”記号をクリックすることで、見出しの内容を展開して表示するUIがある (図11)。

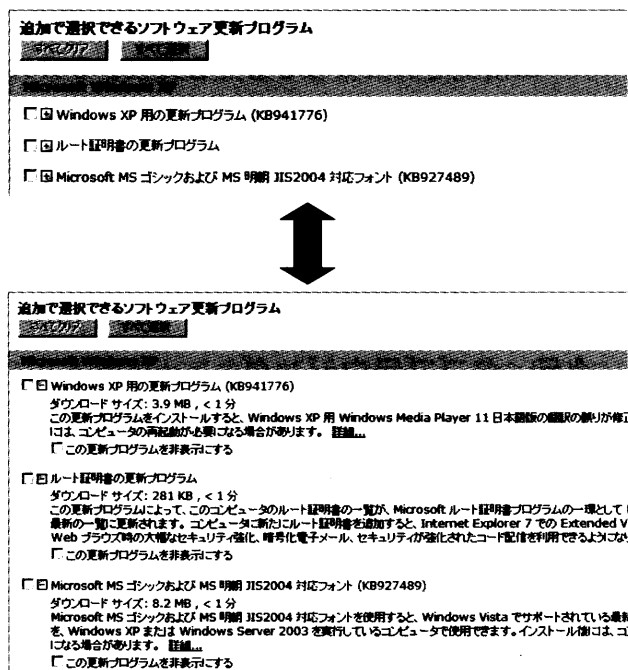


図11 展開型UI  
上図において見出しの“+”記号をクリックすると、下図のように詳細な情報が展開して表示される。その際、同じ場所の“-”記号をクリックすることで上図の状態に戻すことができる。そのため、UFが妨害されにくい。

こうしたページの場合、以前は見出しを書いたページと内容を書いたページを分け、それらをハイパーリンクでつなぐことによってコンテンツを実現していた。あるいは、ページの先頭部分に質問を集め、ページ内リンクで実現しているものもみられた。

そうした方法をとった場合、ユーザは頻繁に“進む”や“戻る”などの操作を行わなければならないが、こうした操作の繰り返しはユーザの認知リソースを消耗させ、目的の達成に対して負荷をかけていた。具体的には、画面左上の“戻る”ボタンまでマウスカーソルを毎回移動させる必要があり、身体・技能レベルで無駄な動きを強制していたと考えられた。また、別ページに詳細を表示させる方式では、本筋のUFのページを覆い隠してしまうこともあるため、やはりUFが妨害されていた。

それに対して展開型のUIは、ページが切り替わることはないため、ページをまたがって進んだり戻ったりする必要がない。新しい情報も同じページ内に表示されるため、ユーザの作業への介入は低く抑えられるようになった。また、ユーザは展開された情報を参照した後、マウスカーソルを移動させることなくその場のクリックで折りたためる。そのため、すぐに元の作業に戻ることが可能になり、UFが失われる可能性が減ったことから、作業効率は向上している。

#### 4.5 タブブラウジング

現在のブラウザはタブブラウジング機能を備えたものも少なくない。タブブラウジングとは、新しいウィンドウを開くのではなく、タブを利用して1つのウィンドウの中で複数のWebページを切り替えられるようにした仕組みである(図12)。

これは、あるページを開く際に背後のタブで開くことを可能にした。これにより、仮に待ち



図12 タブブラウザ

ひとつのウィンドウの中に複数のページを表示できるUI。複数のウィンドウを重ねて開くよりも、一度に多くのページを閲覧することが容易になった。

時間が長いページがあっても、背後でそのページを待っている間に前面ページで本来のUFを保たせることができる。このため、検索エンジンなどで多数のページを同時に開いても、表示に時間がかかるページは背後に回すことができるようになったため、大量の検索途中でUFを見失ってしまうことが少なくなった。

また、ポップアップページが開いても、背後のタブで開かれる場合には、ポップアップページが前面に出てくることがないため、UFは妨害されにくい。

#### 4.6 検索ウィンドウと検索フォーム

Webブラウザのページ内検索のUIにも、UFに配慮した形での改善がみられる。

従来のIEのページ内検索のUIはポップアップウィンドウ形式になっていた(図13)。ページ内検索の機能呼び出すと、検索後を入力するフォームを備えたウィンドウがページの中央に表示される。だが、ページ内検索という作業を考えたときに、背景を隠すポップアップウィンドウでは都合の悪いことが多い。検索にヒットした箇所が検索ウィンドウの下になっている可能性もあるため、ユーザは意識的に検索ウィンドウを移動させたり、閉じたりしなければならず、検索というUFが妨害されていた。2006年10

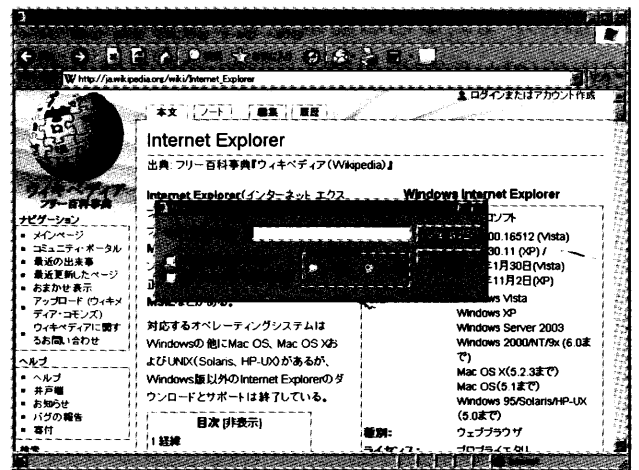


図13 ページを隠してしまう検索ウィンドウ

ポップアップウィンドウと同様にフォーカスを奪い、背後の情報を隠してしまう。そのためユーザは、検索ウィンドウを移動させて検索結果が隠れていないことを確認したくなる。

月に登場したIE7では、ページの中央ではなく、右上隅の方に表示されるようになった。少しでも画面を隠さないような配慮が感じられるが、やはり何らかの情報は背後に隠されてしまうため、本質的な問題は変化していない。

一方、Mozilla Firefox (Firefox) は、同じページ内検索のフォームとして、ページ下端からせり上がってくる“検索バー”の形をとっている(図14)。この方式では、隠されてしまうページの内容は多くても下端の1~2行である。また、ウィンドウではないため動かす必要もない。UFの妨害という観点で見た場合、ポップアップウィンドウ形式に比べて優れたUIだと考えられる。

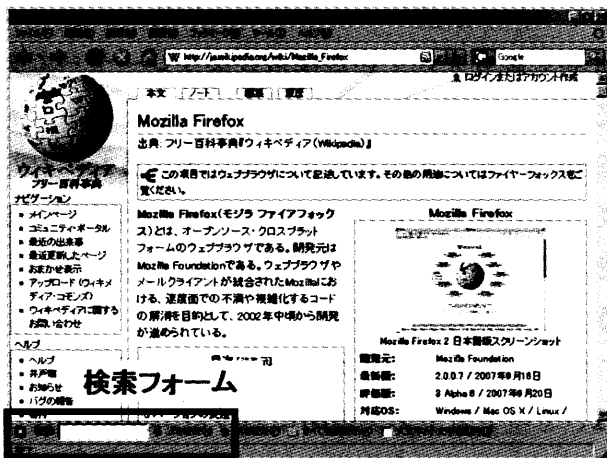


図14 ページを隠さない検索フォーム  
ポップアップウィンドウと異なり、フォーカスを奪わず、背後の情報に与える影響が少なく、また、隠された部分もスクロールで容易に閲覧可能である。

また、Apple iTunesのように検索フォームを常に表示しているものもある。メニューをたどって機能を探したり、ショートカットキーを打ったりする必要もない。常にウィンドウの右上に見えているフォームをクリックして文字を入れるだけで目的が達成できる。さらに、検索結果はメインウィンドウに直接示され、別ウィンドウが出現しないため、検索結果を知ることが簡単であり、また検索ワードを削除することで表示が元に戻るため、すぐに元のUFに戻ることもできる。これは、検索するというUFに対する妨

害が入り込む余地が極力減らされており、かつ元に戻ることも容易なUIであるといえる。

## 5. UFの可能性

UFの概念は単にUIの設計にとどまるものではない。UFの概念を用いると、従来の様々な現象を説明できる。以下にいくつかの例を示すことで、その可能性を展望する。

### 5.1 テキストファイルの利点

情報は、その寿命や可用性を保つため、できるだけ多くのUFに取り込んで運用できることが望ましい。

ある特定のソフトウェアでしか利用できないファイルは、そのアプリケーションソフトが失われると再利用できないことが多い。しかし、テキストファイルは一般的な情報機器であれば利用可能であり、あらゆるUF中で利用することができる。

ユーザ側では情報はできるだけ多くの場面において利用したいものであり、また時間が経過しても利用が可能であってほしい。その意味で、再利用可能性の高いテキストファイルは、多くのユーザが持っているUFに取り込みやすいファイル形式であるといえる。一方、特定のアプリケーションに関連づけられた情報はそれを持っているユーザのUFにしか取り込めない。そのため、利用が限定的になり、状況の変化とともに利用ができなくなっていく危険性が高い。

### 5.2 マルチタスク機能不要の可能性

コンピュータはマルチタスクが可能になっているが、一方ユーザの注意は一度に一つの対象にしか焦点化しないため、UFも一本の流れにしかかなりえない。複数のソフトウェアを起動させていても、ユーザが一度に向き合えるのは一つだけである。ここから、コンピュータも実は一度に少数のソフトウェアを表示するだけで十分なのではないかとも考えられる。



近年のパーソナルコンピュータの利用者は、画面上にメールソフトやRSSリーダー、Messengerソフト等を常駐させ、画面に表示させていることが多く見受けられるが、そうしたユーザはこれらのソフトによって細かい割り込みを数多く受ける。そのため、UFの切り替えが頻繁に発生し、作業効率という観点からはパフォーマンスは低下しているのではないかと予測される。

ただし、新しい作業を行うためのソフトウェアの起動に時間がかかる場合には、その間にUFが失われてしまう可能性がある。逆に、個々のソフトウェアの起動と機能実行がUFの保持にとって妨害にならない程度に十分に高速であれば、コンピュータ側のマルチタスク機能は実は必要がない可能性がある。

### 5.3 ユビキタスコンピューティングの目的

情報機器を生活の中に溶け込ませる、という形をとる“ユビキタスコンピューティング”の概念においても、このUFの妨害の観点を用いることで問題点を導き出すことができる。

ユビキタスコンピューティングの目的は、どこでも誰でもが情報機器を利用したサービスを受けられることである。そこにおいて、情報機器自体はすでに透明になっていることが望ましい。

しかし、現在の携帯電話機器のように複雑な情報処理機器を使う場合には、たとえそれが“どこでもネットワークが利用可能”であったとしても理想的な状態とは言えない。ユーザはその機器の利用自体に対して認知的な処理資源を消費してしまうため、本当の目的(電話機能やメール機能を用いたコミュニケーションや、GPS機能を利用した現在位置特定など)を達成するまでの間にUFを横取りされてしまう、あるいは元のUFを失ってしまう可能性が高いためである。

また、別の問題として、ユビキタスネットワークがユーザに過剰に介入してしまう場合にも問題が発生する。たとえば、マーケティングなど

において、ユーザが望まない商業情報が機器を通じて提示される状態などは、たとえユビキタスコンピューティング社会が到来したとしても、UFに干渉する情報もまた増加するため、利便性は損なわれることが予想される。

ユビキタスコンピューティングとは、単に場所や時間の自由度の問題だけではなく、ユーザの目的に対して、できるだけコストをかけずに到達できるためのUFをサポートする仕組みであるととらえた場合に、はじめて有効な方向性を得ることができるだろう。

### 5.4 マーケティングにおける応用

AppleのiTunesは、2007年現在最も多く利用されている音楽プレーヤソフトであり、iPodは世界で最も売れている携帯音楽プレーヤーである。またiTunes Music Storeは現在世界のネット音楽市場を大きくリードしている。これに対して、他の音楽プレーヤーやネット配信はそれほどの成功を収めていない。この理由は何だろうか。

このiTunesが現在の位置を占めた理由としては、マーケティング戦略における、UFの観点が重要となっていたと考えられる。

iTunesは最初に使い勝手のよいPC用の音楽プレーヤーとして無料で配布された。これによって、ユーザはiTunesを使って音楽を管理するというUFを自然に身につけた。そして、そのUFが整ったところでAppleはiPodというiTunes専用のハードウェアを提供した。これは従来のiTunesを利用するUFをそのまま利用・拡張する形で行われたため、ユーザは違和感なくそのハードウェアを利用することができた。また、iPodによって、音楽を持ち歩くというUFが新しいタイプのUFとして受け入れられた。さらにAppleは、iTunesでの音楽管理とiPodでの音楽持ち歩きというUFの延長線上に、ネット上での音楽販売を位置づけた。ここにおいて、音楽を購入し、管理し、好きな場所で自由に聴取するというUFが、iTunesを中心とする大きなUFに、

ユーザ側の余計なコストをかけずに統合できるようになった。その結果、iTunesとその周辺のシステムは、多くのユーザが自然に利用するようになり、他社を引き離すことになった。

これは、他のメーカーの単純なDRMによる囲い込み戦略とは一線を画した戦略であり、音楽を聞く際のユーザのUF制御を段階的に計算した戦略的マーケティングであったと考えられる。この例は、結果的に見ると、UFを企業側が有効に生成し、活用した例であったといえよう。

このiTunesの例のように、製品開発の際にUFの概念を戦略的に取り込むことで、ユーザにとって自然に使える道具を提供することができるようになり、その結果、ユーザから支持を受ける道具を作ることができる可能性が高くなるだろう。

## 6. まとめ

これらの様々な領域の事例から、よい道具とは“UFがよどみなく流れる状態で利用できるもの”であるといえそうである。そのように作られた道具はユーザにとって使いやすく感じられる。また、使いにくい道具の共通のパターンとして“UFが滞る”ないしは“UFが途切れてしまう”という点も認められる。

したがって、よいUIとは“スムーズなUFをうまく生じさせることで、ユーザの目的を達成しやすくするUI”と定義することができる。より具体的には、“ユーザの既存のUFを再利用させるもの”、それが困難な場合には“ユーザに対してコストをかけさせずに自然なUFを作り出すもの”とみなすことができる。これは、身体技能・ルール・目的のすべてのレベルにおいて当てはまる。反対に、ある道具について、これらのいずれのレベルでもUFが滞る場合には、その道具のユーザビリティが各々の側面で低下すると考えられる。

すべての道具は人間が利用する目的で存在する。情報機器を含めた各種の複雑な“道具”類

を開発する際に、ユーザの最終的な目的をいかに容易に果たさせるのか、という視点に基づいて品質を考慮する必要がある。その際に、この“UF”という概念を戦略的な上位概念として位置づけ、これを満たすことを目的として開発を行うことで“使えない道具”を産み出してしまいう危険性を減らし、またある製品をより多くのユーザに利用してもらうことができるようになるだろう。

## 引用文献

- Anderson, J.R. 1980 *Cognitive Psychology and its Implications*, San Francisco, Freeman. (富田達彦・増井透・川崎恵里子・岸学訳 1982 認知心理学概論 誠心書房)
- 海保博之編著 2000 瞬間情報処理の心理学 人が二秒間でできること 福村出版.
- Norman, D.A. 1988 *The Psychology of Everyday Things*. Basic Books, New York. (野島久雄訳 1988 誰のためのデザイン 新曜社)
- Norman, D.A., & Bobrow, D. G. 1975 On Data-limited and Resource-limited Processes, *Cognitive Psychology*, 7 (1) , 44-64.
- Rasmussen, J. 1986 *Information processing and human-machine interaction: An approach to cognitive engineering*. Elsevier Science Publishers, Amsterdam, The Netherlands. (海保博之・原田悦子・黒須正明 1991 認知的インタフェース 新曜社より)