



## COLUMN

ITルネサンスのために:ヒューマンインタフェースの復権を

## 専門用語とバグと納期の呪縛

天野 真家 (株) 東芝研究開発センター  
shinya.amano@toshiba.co.jp

## 非常

に具体的な例を挙げたい。

HIの根本は、ユーザにとってなじみのない概念をいかにして自然に認知させるかという問題に帰着する。これが、一筋縄ではいかないのが設計者は苦勞するのである。しかし、どう考えてみても、もっと基本的な部分でHIへの配慮がみられない場合も存在している。今回は、そんな例を挙げて、その原因を考えてみたい。

マルチメディアPCが普及し安価に手に入るようになったおかげで、筆者はいろいろな画像をスキャナで取り込んで保存している。過去5台程のスキャナと、同数のデジカメを併せれば10本以上の画像処理ソフトを使ってきた。その経験からいうと、画像処理ソフトにはHIの劣悪なものが多いというのが実感である。一般の家庭での使用者<sup>☆1</sup>なら、起動はしたが何をしたいかわからず、茫然とするにちがいないものが存在する。筆者も初めて買って来たときは投げ出してしまった。いわゆる押入れ行きである。

1つには画像というものを処理することの難しさがある。たとえばいきなり「キャリブレーション」という専門用語が平気で出てくるのである。このようなことは、同じ画像関連機器でも、VTRやテレビでは絶対にあり得ないし、IT機器の一角を担うデジカメにしても、経験したことはない。

しかし、キャリブレーションに関しては、その意味は分からないにしろ、マニュアルの最初近くに書いてあることが多いし、自動で行うものもあるので、それほど問題にはならないかもしれない。それでも、「キャリブレーションを実行中です」、というメッセージは家庭人に

にとってはほとんど無意味語である。「キャリブレーション」という家庭人にとってなじみのない概念を家庭人に認知してもらうための努力を怠っているといつてよい。さらに家庭人にとって未知なるものが、TWINというスキャナのインタフェース規格である。これが、またHI上の問題を産むインタフェースなのである。スキャナ用画像処理ソフトというものは、主として

- 処理1. 紙に印刷された画像を読み込む
- 処理2. 読み込んだ画像を表示する
- 処理3. 必要なら処理を加える
- 処理4. ディスクに保存する

の4つの操作を行う。家庭人は当然、これらすべては、今、起動して目の前に開いているスキャナ用画像処理ソフトが行うのだと考えるだろう。それが、一般の常識である。しかし、この一般の常識がひょっとするとITの世界では常識として通用しないのかもしれない。スキャナ用画像処理ソフトを起動すると、普通、スキャナのアイコンがあって(図-1の上部中程、円で囲ったアイコン)、これをクリックすればスキャンできそうである。だが、そういう作りにはなっていないソフトがある。最初に使うときは、スキャナのアイコンをクリックしても、「スキャナがない」と言われてしまう。スキャナは、USBでPCにPnP<sup>☆2</sup> (Plug and Play) 接続されていると思って

☆1 以後、本連載では、「一般の家庭での常識的使用者」を「家庭人」と呼ぶ。

☆2 これは、家庭人が考えるであろう概念としてのPnPである。厳密な意味でのPnPは、TWINとのかかわりまではサポートしない。ここにも家庭人とITとのギャップがある。



図-1 典型的画像処理ソフトの画面  
(ソフトを特定できないようモザイクをかけてある)

いる家庭人の期待は裏切られるのである。

「スキャナがない」と言われたユーザは、スキャナをつなげるために、懸命に「スキャナ」をメニューから探そうとする。しかし、筆者の手持ちのいくつかのソフトでは、メニューをくまなく探しても「スキャナ」というような項目は見つからなかった。まさか「TWAIN機器の選択」というメニュー項目がスキャナというハードウェアをこのスキャナ用画像処理ソフトから使えるようにする呪文だなどということをして、一体どれだけの家庭人が思いつくであろうか。

「TWAIN」という専門用語が、これほど無造作に何のためらいもなく家庭用機器に使われていることを、関係技術者は家庭人の視点に立ち返って、考え直すべきでないか。さらに、「TWAIN機器の選択」というメニュー中の「選択」もまた、不可解な用語である。スキャナを買ってバンドルされているソフトなのであるから、そのスキャナが自動的に認識されて画像処理ソフトを起動したときに使えるような状態になっているというのが家庭人の常識である。それなのに、「選択」とは何を選択するものなのかという疑問が湧いてしまうのである<sup>☆3</sup>。家庭人にとっては、スキャナの「選択」とは、不可思議な概念なのである。

図-1 のスキャナ用画像処理ソフトでは、

「ファイル→入力用プラグ→ソースを選択」<sup>☆4</sup>

でスキャナを選択するというものであった。まったく想像を絶するメッセージの内容と、メッセージの置かれているメニュー上の位置である。

第1回の「man」のところでも、manはマニュアルではなく備忘録である。つまり、一度は理解していることが前提になっているのであると述べたが、明らかにスキャナ用画像処理ソフトの設計者は、使用者がこれだけを見て認知に至るであろうというように配慮した設計は行っていない。

スキャナのようなIT機器では、一般に通常の家電機器とは異なり、分厚いマニュアルを繰ってその操作方法を調べなくてはならないが、TWAINで頭がいっぱいの技術者の書いたマニュアルには、TWAINの観点からしか書かれていないことが多い。このような専門用語は家庭人には何の意味も持たない。これは、第1回で述べた使用者の認知能力に対する設計者の配慮不足なのである。このような配慮不足は、これを理解していても、なお犯してしまう性質がある。これは、利用者が理解できないだろうことに対する技術者側の認知的問題であるからである。

筆者は、20数年来苦い思いをしている経験がある。JW-10<sup>☆5</sup>のエディタ部の状態を示すメッセージに「文書を展開しています」というメッセージを出したことがある。当時は、何も疑問を感じなかった。JW-10では、HIを考えなかったというわけではない。むしろ、その逆である。HIについては、1つの操作キーの名称だけについても、1時間も議論したというような経験をした。たとえば、JW-10には、「全文対象」というキーがある。「削除」、「訂正」のような操作を、全文にわたって一挙に行うためのキーである。後続の他社のワードプロセッサでも、この用語は踏襲された。しかし、この用語はマニュアルなしでは理解できないだろう。

そのようなHIへの配慮の意識をもってしても、「文書を展開しています」というメッセージが使用者向けになっていないことには長らく気が付かずだった。ふと、気がついたのは、10年以上も経ってからである。このメッセージは、文書を開くとき、ハードディスクからの読み込み時に出していたものである。それほど使用者にとって意味があるものではなく、重要なものでもない。

しかし、HIを重要視している筆者にとっては、これは痛恨の失敗なのである。ここは、「文書を読み込んでいます」とすべきだったのである。では、なぜ、「展開」としてしまったのか。このエディタは、今でいうマークアップ形式で文書を記憶している。モニタに表示するためには、表示形式に「展開」しなければならない。そのためのサブルーチンにも「develop」という名を付けていた。これが、小さな主記憶(エディタ分は十数KB)と、限られた期間で開発しなければならないというのに、バ

☆3 TWAIN機器は、スキャナだけとは限らないという弁明はあるが、バンドルソフトでは、明らかに配慮不足である。

☆4 この記法は、WindowsなどのGUIで、メニューを辿るときによく使われる。「メニュー」の「ファイル」項目をクリックすると、「入力用プラグ」項目がある…という記法である。

☆5 第1回連載「新たな概念に対する人間の認知能力：操作とコード」、Vol.43, No.10, pp.1106-1107 (Oct. 2002) 参照。



グを出す難しいソフトであった。筆者の頭には、「展開」が染み付いていたのである。

技術者は、このように、自身にとっては何でもない空気のような存在になってしまっている専門用語を、何の疑問も持たずに使ってしまふことからいかにして自由になれるのであろうか。これはHIの永遠の課題かもしれない。技術者の思い込みを発見するためにもテクニカルライタが必要になるのであるが、テクニカルライタもすぐに専門用語に慣れてしまうので、あまり解決にならないような気がする。

さて、スキャナにはまだ問題がある。上記、TWAINに関する問題である。スキャナ用画像処理ソフトを起動し、スキャナアイコンをクリックすると2つのウィンドウが開く。開いたウィンドウの1つは、TWAINドライバのもので、処理1を行うものである。もう1つのウィンドウは、画像処理ソフトのもので処理2～4処理を行う<sup>☆6</sup>。この2つのウィンドウの関係をどのように設定するかには、HIのセンスがかかっている。

2つの事例をみてみよう。

#### 【事例1】

図-2、図-3では、2つのウィンドウは独立な関係にある。

図-2で、手前にある、右半分が黒い小さなウィンドウがTWAINドライバのものであり、後方の大きいウィンドウが画像処理ソフトである。スキャナを認識したあと、後ろ側にある画像処理ソフトのスキャナのアイコン(図-2の上部の円内)をクリックすると、このTWAINドライバのウィンドウが出てくる。スキャンする紙の大きさ、解像度などを設定して、下部右方の楕円の「スキャン」ボタンをクリックすれば、画像などをスキャンできる。この画像処理ソフトと、TWAINドライバはよくできていて、最初の、「ファイル→入力用プラグ→ソースを選択」問題さえなければ非常によくHIに配慮されていると筆者は感じた(図-2は英語版であるが、日本語版では、日本語化されている)。このTWAINドライバは、スキャンした後、図-3のように後方に退く。Windowsの通常の操作法に従った作りになっている。これは、17インチ、解像度1024×768程度の小さな画面で操作しているときには非常に重要な性質である。なぜなら、家庭でさえ、画像を1枚だけスキャンするよりは、何枚かを連続して取り込む場合は少なくないと思われるからで、その場合、このウィンドウの関係は操作が非常に楽なのである。

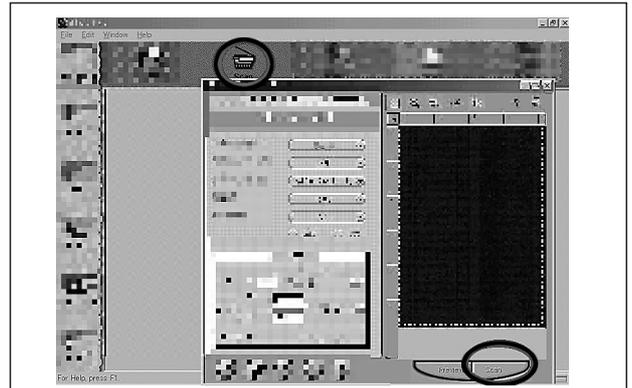


図-2 TWAINドライバウィンドウは前方にあるところ



図-3 TWAINドライバウィンドウは後方にまわっているところ

#### 【事例2】

一方、対極にあるのが図-4の場合である。

注目したい場所は、前面のTWAINドライバが出しているウィンドウである。このウィンドウには、最上部にウィンドウを操作するためのタイトルバー(図-5)がない。

したがって、移動することも、不活性にして画像処理ソフトの裏に持っていくこともできない。画像処理ソフトウィンドウを見てみると、タイトルバーが紺色ではなく灰色になって不活性状態であり、これも操作できない。この状態では、TWAINドライバが出しているウィンドウを右下の「閉じる」ボタン<sup>☆7</sup>で閉じなければ、取り込んだ画像を見ることさえできない。閉じてしまえば、次の画像を取り込むには、再度、最初と同じ操作でTWAINドライバを呼ばなければならず、余分な手数と時間がかかる。この場合、TWAINドライバのウィンドウは移動することもできないので、たとえ21インチという大きなモニターを使っていようと、画像処理ソフトウィンドウの脇に置いて使うということはできない。

☆6 画像処理ソフトとTWAINドライバは独立であり、任意のものを組み合わせて用いることができる。

☆7 この図では、モザイクのため、文字は読めない。

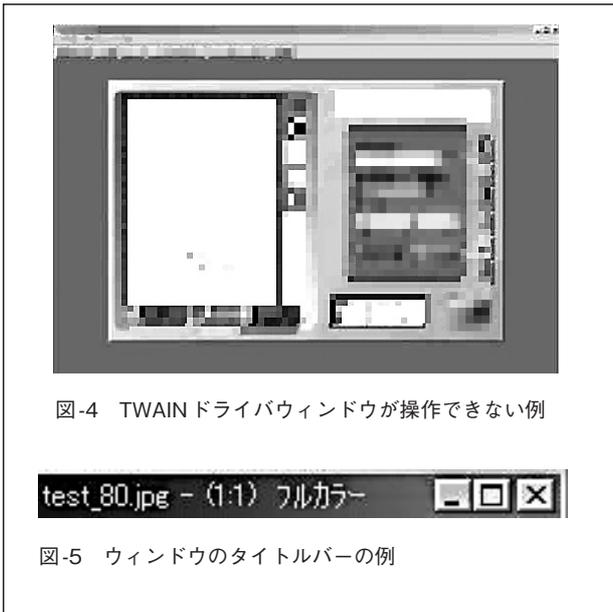


図-4 TWAINドライバウィンドウが操作できない例

test\_80.jpg - (1:1) フルカラー

図-5 ウィンドウのタイトルバーの例

この画像処理ソフトのHI思想は一貫して、スキャナで画像を取り込むごとにすべてのパラメータの設定をやり直さなければならない設計になっている。設定ファイルを使わず、パラメータをハードコーディングしてあるようなのである。こうして取り込んだ画像をJPEGで保存しても、異常に大きいサイズになった。JPEGにするということは、使用者は、容量が1/10程度になることを期待している。それが、1/3程度にしかになっていないのである。

たとえば、このような具合である：

2.20MB → 695KB (品質：100%)

(参考：152KB (品質：80%))

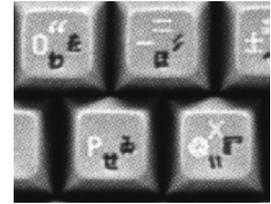
その一部を切り出し、品質とともに見てみよう(図-6)。見て分かるように、通常の家庭の使用法では、80%の品質で十分であろう。

調べてみると、デフォルトでは100%の画像品質の設定になっている。しかも、この設定は、変更して保存するという通常のソフトがやっている機能がないので、画像を1枚保存するごとに、「オプションボタン→スライドバーで品質の%を設定→OK」というような多大な手間をかけて指定しなければならない。

また、この画像処理ソフトは、「編集メニュー→元に戻す」の「元に戻す」が1回しかできない。普通は、10レベル程度は用意されていたり、自分でレベル数を設定して保存したりできるような設計になっている。画像処理では、言うまでもなく、試行錯誤の連続といていいような操作をする。ああしてみよう、こうしてみようとした挙句、何レベルも前に戻すのである。それができないのである。



原画像 (BMP 123KB)



(JPEG 100% 43.3KB)



(JPEG 80% 10.5KB)

図-6 BMP形式の画像とJPEG形式の画像の品質

以上の事例は、技術の難易度の問題ではなく、すべてHIの設計思想の問題である。なぜなら、ウィンドウを事例1のように扱う方法と、事例2のように扱う方法とではプログラミングの難易度、あるいはステップ数はほとんど変わらないからである。むしろ、事例1の方がデフォルトのウィンドウの設定がしてあり、事例2のような状態にするためにはウィンドウのプロパティをプログラミング時にわざわざ変更する必要があるはずである。プログラミングには、プログラマのさまざまな意思が反映される。ウィンドウを使用者が操作できるようにすると、別の面でプログラマにとって、プログラムの状態が危険になったり、あるいは、バグが現れやすくなったり、その後のプログラミングが煩雑になったりする可能性がある。これを避ける最も安易な道は、使用法を制限することである。

そして、もう1つ、このようなプログラミングをしてしまう可能性は納期である。迫りつつある納期のために最初に犠牲になるのがHIである。機能、性能はカタログに記入され、そのソフトを使うまでもなく、客観的評価ができる。しかし、HIには客観的基準——カタログに書き込まれる基準——が存在しない。この意味で、HIは工学足り得るのか、という疑問も湧く。HIはどのようにしたら、家庭人に訴え得る、かつ、粗悪なHIを排斥でき得る工学的基準を持つことができるのだろうか。これは、この連載を通しての課題ともいえる。

(平成14年10月17日受付)