

富豪的プログラミング

増井俊之

ちょっと貧しい GUI

ユーザインタフェースのプログラムといえばすっかりグラフィカルユーザインタフェース (GUI) が当たり前になりました。ところが最近の計算機は、メモリもディスクも大量に装備しているし、CPU パワーも従来とは比べものにならないのに、意外と貧乏臭いインタフェースが生き残っているようです。よく見られる以下のようなインタフェースはちょっと貧しいと思われれます。

• ウィンドウ枠のドラッグ

マウスでウィンドウを動かそうとするとき、ウィンドウの枠だけマウスに追従し、中身はドラッグ終了後に再描画されるというシステムがあります。画面描画が遅かった時代はこういう工夫も仕方なかったかもしれませんが、今でもこういうインタフェースを使っているのは貧しいでしょう。排他的論理和で枠を書いたりもとに戻したりしていれば、貧しさ度アップです。でも調べてみたら、同僚のほとんどが枠だけドラッグのシステムを使っていました。

• スクロールバーの移動と連動しないウィンドウ

カンガルー跳び現象というのがあります。

• 多段 undo できないエディタ

履歴情報ぐらいケチらずに十分覚えておいてほしいものです。

• ファイルのセーブ

いちいち指示しないと編集結果が消えてしまうかもしれないというの貧しいものがあります。黙っていても編集結果ぐらい記憶していてほしいものです。

• 検索実行

辞書などを検索するとき、検索キーワードを入力してから検索実行を指示するようなインタフェースもよく考えると貧しいものです。検索の実行と

表示に 1/10 秒もかからないのであれば、キー入力により検索条件が変わるたびに検索実行/表示すればよいでしょう。

ところで 10 年ぐらい前はワークステーションでも GUI があまり普及していなかったのも、ウィンドウシステムや各種のインタフェースツールキットの研究開発が盛んでした。しかし最近、新しいウィンドウシステムが作られたなどという話をとんと聞きませんし、GUI のプログラミング手法とかデバッグ手法というのほとんどお目に掛かれなくなっていました。実際ここ数年の **bit** 誌を見てもそういう記事はほとんどありませんし、計算機インタフェースの学会でもその手の研究はあまり流行っていないようです。

この理由はざっくり、ウィンドウシステムやツールキットが十分普及したので、苦労しなくても GUI が簡単に作れるようになってしまったという思い込みが浸透したからだろうと思います。特に素晴らしいプログラミング技術やインタフェース部品が発明されたわけではありませんから、実のところは、ユーザの我慢レベルとプログラマの我慢レベルが妥協点に達してきただけではないかと思えます。この過程で、けっこう貧乏臭いインタフェースが標準になってしまったようです。

Web の CGI がその典型的な例ではないでしょうか。片やインターネットの音声だとか動画だとかが話題になっている時代に、ユーザ側からは字の並んだだけのメニューからの選択や文字入力しかできないという仕様はかなり貧しいものです。Java や VRML の話題が沸騰しているのに対し、CGI の仕様が問題になっているとか、新しいバージョンが計画されているとかという話はあまり聞きません。CGI でも最低限の指示はできますし、ユーザが簡単にプログラマに変身できてしまうので、貧しい仕様で満足させられてしまっているようです。これでは昔の BASIC のようで

す。

富豪的プログラミング

プログラマというものは、つい昔の癖で効率的なプログラムを工夫したり資源を節約したりしようとしがちですが、ユーザインタフェースのプログラムを開発するときにはこれが裏目に出ることもあります。というのも、ユーザインタフェースのプログラムでは機械の効率よりも使い勝手が優先されるべきですし、プロトタイプ作成とその評価/改良のサイクルを数多く繰り返す必要があるのですが、計算機資源を節約しようとする、これらの条件が後回しになりがちだからです。

これを解決するのが富豪的プログラミングです。以下のような富豪的プログラミングを行えば、このような問題は起こりません。

- **メモリや実行効率を気にしないでお気楽にプログラムを作る**

効率を重視したプログラムは作るのが大変ですし、ちゃんと動かすにはデバッグも大変です。富豪的プログラミングでは一番単純で短いアルゴリズムを使います。

- **条件が変わるたびにすべての計算や表示を行なう**
再表示が必要な場所だけ書き直ししたり、出力のバッファリングをしたりする貧乏的な工夫はバグのもとになるので行わず、条件が変わるたびに計算を再実行したり全体を書き直ししたりします。

富豪的プログラミングでは現在の状態を記憶しておく必要が少なくなるのでプログラムが短く、バグを含みにくくなります。また、ユーザが何か操作を行なうたびにシステムがすぐに反応するので、直感的操作が可能になります。

アイコンのドラッグのようないわゆる直接操作インタフェースは富豪化への第一歩です。ユーザの入力により検索条件が変わるとすぐに検索を再実行して表示を更新する手法は動的検索とか Dynamic Query とか呼ばれており、将来の検索システムでは常識になると思われます。またユーザの操作に応じてリアルタイムに連続的にズームングを行なうことにより、巨大なデータを簡単に操作しようというズームングインタフェ

ースが、最近では注目を集めています。こういったインタフェース手法は、富豪的な発想がないと思いつかないでしょう。豊かな資源から豊かな発想が生まれるのです。

Web の成功はその富豪的な発想が鍵だったのではないのでしょうか。クリックするたびに海外から画像を転送するなどというのは、貧乏人には思いもよらない発想です。最近は何となく富豪的なインタフェースツールキットもあるようです。カーネギーメロン大学で開発された Amulet というツールキットを使うと“Hello, world”とウィンドウに表示するプログラムのサイズが5 MB になります。素敵な富豪感覚ですね。腹が減ってはこれからのシステム設計はできないのかもしれない。

もちろん発想がいくら富豪でも実際にプログラムが動かなければ仕方ありません。最近の PC は昔に比べると超大富豪的なハードウェアをもっていますから、何も気にしなくても富豪的プログラムが動いてしまうことも多いですし、プロトタイプするときだけでも高速な計算機を使って豊かな気持ちでインタフェースを設計するのが大事でしょう。まだまだ貧しさが潜んでいるかもしれません。

というわけで、これからの GUI プログラム/デバッグは富豪化がポイントでしょう。私は最近では富豪的プログラミングしかしないので、GUI のデバッグなどしたことはありません。はずなんですがね。

(ますい としゆき 富豪化理論研究者)