

## 利用時に変化する GUI 部品

1 Z A - 9

赤池英夫

電気通信大学 情報工学科

### 1 はじめに

GUI(Graphical User Interface) といった言葉が一般に使われるようになってから久しく、GUI を全く用いない対話型アプリケーションソフトウェアを思い出すのも難しいといった状況である。ウィンドウシステムをサブシステムとして含むオペレーティングシステムの場合は当然のことながら、GUI 構築用ツールキットやデスクトップ環境毎に、統一された見た目と振舞いが提供されている。このため初めて使用するアプリケーションであっても操作の類推が容易であり、初心者にも使いやすくなっている。

しかしアプリケーションの使用に慣れてくると機能を実行するためにボタンを押す操作が煩わしくなったり、高速な操作により隣接するボタンやメニュー項目を誤って選択してしまうことが多くなる。これは小さく多数の GUI 部品が狭い範囲にあればあるほど顕著となる。もちろん GUI 部品のレイアウトやサイズなどをカスタマイズする手段が提供される場合もあるが、変更は一意的でユーザのスキルの経時変化に追従するものは少ない。また設定ファイルの変更によるカスタマイズなど、初心者にとって難しい場合も多い。

そこで、本論文では見た目や振舞いがユーザの操作パターンから漸次変化したり、特定の条件を満足した時点で瞬時に変化する GUI 部品を用いてこれらの問題を解決する手法を提案する。

### 2 利用に即して変化する GUI 部品

たとえば様々な機能を提供する多数のボタンを持った対話システムがあるとすると、個々のユーザがどの機能を利用するのかあらかじめ判らないので、全てのボタンが同様なサイズで表示されていることは極めて自然である。しかしシステムを使い続けた結果、

個々の機能の使用頻度に差が生じた後にも初期サイズの領域を占める必要はなかろう。極端な言い方であるが、適当な期間全く使用されなかった機能に対応するボタンを削除しても問題にならない。さらに、空いた領域を使えば残ったボタンをゆったりと配置したり、システムが本来提示すべき内容(テキストやイメージ)に割り当てることができる。

一方、平均的に選択される項目を持つメニューやウェブブラウザの URL リストなど、システムによる自動的な削除が必ずしも許されない内容を表示する GUI 部品もある。これらは項目数が増加すると選択に手間がかかるが、たとえばマウスポインタの周辺の項目をより大きく表示し、離れるに従い小さく表示すれば、隣接する項目の誤選択も減らすことができる。マウスポインタを項目の端に移動させなければ下位のメニューが表示されないタイプの階層メニューでは、より有用となると考えられる。

つまり、GUI 部品を固定された位置にある、決まった大きさを占める受動的なオブジェクトとしてではなく、ユーザの操作に対し見た目や振舞いを能動的に変化させるオブジェクトと捉えることで、GUI の持ついくつかの問題が解決できる。

#### 2.1 トリガ操作

ユーザの挙動は入力デバイスを介して得られるが、その中で GUI 部品に変化を起こさせるものを“トリガ操作”と呼ぶ。アイコンやボタンの押下、メニュー選択、スライダの移動など、GUI 部品に対する操作はトリガ操作である。また GUI 部品に向かってマウス(一般にはポインティングデバイスの)ポインタを移動させる、あるいは移動させているように見える操作もトリガ操作である。あるいは短時間に様々な方向にマウスポインタを移動させるような恣意的な操作(ジェスチャ)もトリガ操作として扱う。

トリガ操作は操作の種類と値および時刻の組で表現されるが、値は種類毎に異なっている。たとえば

GUI 部品熟练操作であれば選択されたメニュー項目の識別子やバリエータの操作量などである。またマウスポインタの移動であればそのベクトルである。

アプリケーションにはトリガ操作を項として含む条件と、その条件を満たしたときに行われる GUI 部品の変化が規則としてあらかじめ与えられる。また選択されるであろう GUI 部品の予測なども条件として記述される。

## 2.2 GUI 部品の変化のバリエーションと応用例

現在、GUI 部品の可能な変化として次のようなものがある。ただし、それぞれの変化が単独に起こるだけでなく組み合わせが生じる場合もある。

### ● 生成と破棄

初期状態では存在しない GUI 部品が、繰り返し操作の結果として動的に作成される。あるいは使用頻度の極めて低いものが(少なくとも視覚的に)取り除かれる。たとえば特定のボタンを繰り返し押し下すと、そのボタンを複数回押すことに相当するボタンが作成される。また使用頻度の高い GUI 部品のゴーストがマウスポインタの回りに現れるようになる(図 1)。このゴーストを操作しても対応する機能が実行される。

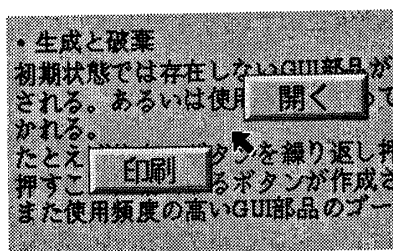


図 1: ゴーストを伴ったマウスポインタ

### ● パラメタの変更

GUI 部品の位置や大きさを変える。ボタンの押し下頻度を大きさに反映させる以外にも、たとえばアプリケーションの終了のような確認を必要とする機能に割りつけられたボタンにマウスポインタを近づけると(少しだけ)逃げるように移動させるとか、マウスポインタの近くのメニュー

項目を大きく表示しポイントしやすくさせるなどの応用が挙げられる(図 2)。

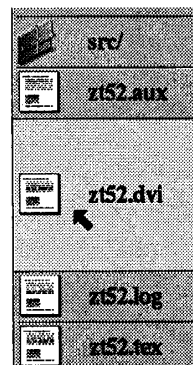


図 2: 表示領域の大きさに違いがあるメニュー

### ● 合成

操作間に適当な順序関係が検出された場合、いわゆるマクロのように複数の機能を順次実行するような GUI 部品が生成される。たとえばウェブブラウザを使用している際に"保存"⇒"戻る"といったシーケンスが多く現れると、"戻る" ボタンの上に "保存&戻る" ボタンが合成される(図 3)。

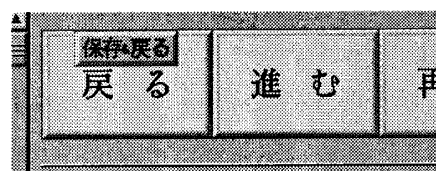


図 3: "保存" と "戻る" が合成されたボタン

## 3 おわりに

本提案はデスクトップあるいはノート計算機での使用のみならず、小さな画面をペンで操作するタイプの PDA や、逆に広い画面を持つ計算機上の対話技法として有効であると考えられ、その検証も計画中である。

また、現在はテストアプリケーションに埋め込む形で GUI 部品の変化規則((トリガ操作を含む条件、GUI 部品の変化)対にあたる)を記述しているが、ツールキットアプローチによる汎用化も検討している。