

4M-4

ユーザインタフェース構築支援システム  
- 自由図形によるユーザインタフェース -

谷越浩一郎, 荒井俊史, 谷正之  
(日立製作所 日立研究所)

0. はじめに

実際のプログラムの本体であるアプリケーションプログラム(以下AP)とユーザインタフェース(以下UI)とを分けて管理するUIMSの考えに基づき, オブジェクト指向を取り入れたユーザインタフェース管理システムMUを開発した[1]。このMUでは画面を構成する"部品"を予め用意しておき, それらの組み合わせと部品同士やAPからのメッセージ通信によりUIを構築する。

しかし, この方法では構成する部品の種類により, 適用できる範囲が限られて定形的なUIしか作れない。

そこで本文では, 定形的なUIを作るのではなく, 適用範囲が固定されてない非定形的なUIをMUの枠組みで構成する手段について検討する。

1. 非定型UIを構成する際の問題点

例えば図1のようなa)プロセス制御パネルのUIとb)銀行のCDの表示画面のUIを両方とも作ることを考えてみる。このようなUIをMUの枠組みにより"部品"で構成するには, a)の場合はメータやスクロールバーを部品で用意してそれらを組み合わせる形で画面を構成することになる。しかし同じ部品ではb)の方をそのまま作ることはできず, 作業進行を表わす矢印のような新しい部品を追加しなければならない。

さらに図形による入力の方法の問題がある。図形による入力では, マウスでドラッグ(ボタンを押したまま走行)した時にそれに伴い表示を変化させ, 最後にマウスボタンが離された時の位置情報をもとに, なにかしらの値を取り込む。例えばスクロールバーは, マウスでドラッグされた時にある決められた範囲内を動くようになっているが, そのようなドラッグ中の表示は部品の管理によって行なわれ, 同じような動きをするスライドボリュームを作る時でも新たに組み込んでいかねばならない。

このように, 今までのような部品を使って構成する方法は非定型UIに対して,

- 1) 部品を常に新しくしなければ新しい種類のUI構築ができない。
- 2) 新しい部品に対して個別に入力手段を組み込まねばならない。

という限界がある。

2. 非定型UIの構成法の概念

このような限界を無くすために, まずUIの画面の絵を自由に図形を用いて描いておいてから, それに対して動作をつけてやるという方法を取ることにした。このようにすれば現存の部品に縛られることなくUIを作成することができる。

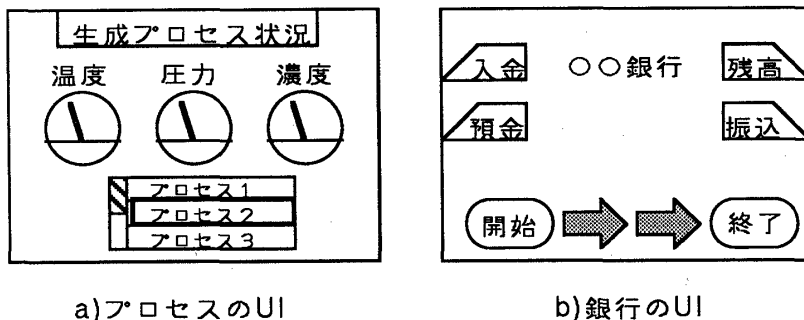


図1 さまざまの形態のUI

具体的には円、直線といった図形のプリミティブあるいはそれらをグルーピングしたものを部品として扱い、それらに動作定義を設定することで実現する。そして、設定するものとして、マウスのボタンがその部品上で押された時にどうするかといったようなイベント駆動的な動作定義と、他の対象物からのメッセージに対する動作定義の2つを設けた。これは今までのMUの部品に対するものと同じ考えである。(図2)

さらに入力に関する動作を「移動」、「回転」、「変形」といった入力形式を指定することで図形に定義できるようにした。定義はどの入力形式をとるのかということと、各々の入力動作固有のパラメータを決定することにより行われる。例えば「移動」のパラメータは移動範囲と値域へのマッピング関数(線形、対数など)、移動の方式(なめらかに動くのか、あるとびとびの点をとるように動くのか)などによって構成されていて、具体的に移動範囲は画面上ここからここと決定することにより、「移動」という入力動作は実行可能となる。そして移動完了後にAPに対してメッセージにより値を渡し入力ができる。

3. 構成

以上のようなシステムを実現する非定型UI定義環境の構成は図3ようになる。まず図形エディタでU

Iとなる画面を描く。このデータは単なる図形データではなく、そのままMUのオブジェクトとなるように作成される。その後で図形エディタについているUI構築用の機能を用いて、図形に対して動作定義や入力動作定義を行う。実際は、図形エディタ自身がMUで構成されているので、UI作成の環境である図形エディタとUI実行環境は即座に切り替えが可能であり、すぐにUIの使い勝手を試してみても不都合な所を修正するといったことができる。

入力動作を行うUIを作成する場合、例えばスクロールバーを作成する場合については、矩形を描いてからそれに対して「移動」という入力動作定義を行う。そして入力動作時には、スクロールを行う対象に対してその移動量をメッセージ送信するように定義しておくことでスクロールバーとしての働きをすることができる。

4. 終わりに

以上述べてきた方法により、面倒であった非定型UIを効率よく構成できるようになった。

[参考文献]

[1]荒井ほか, ユーザインタフェース構築支援システム -メタユーザインタフェース構築支援- 情処第38回全国大会, 1988

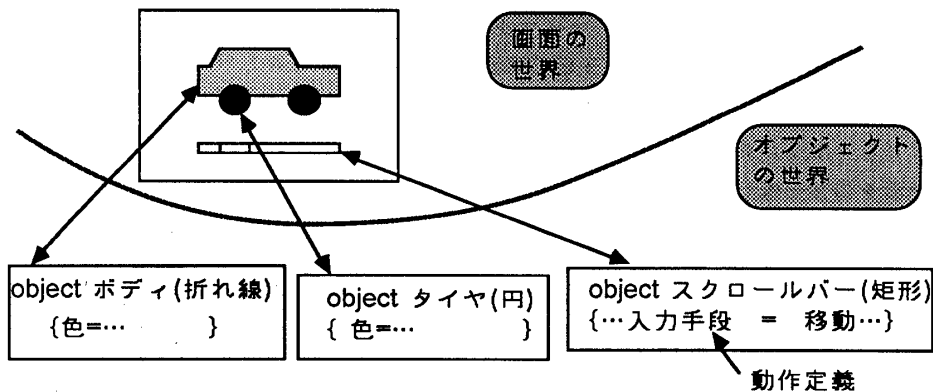


図2 自由図形によるUI定義

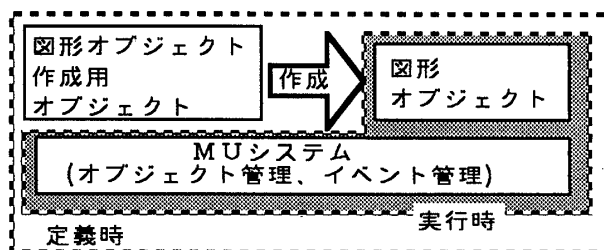


図3 MUシステム