

## CLOSを利用したODETTEのウィンドウインタフェースの実現

3F-10

○久保昭一(日立超LSIエンジニアリング(株))

湯浦克彦(株)日立製作所 中央研究所)

高橋久(日立超LSIエンジニアリング(株))

## 1. はじめに

オブジェクト指向型設計支援環境ODETTE<sup>1)</sup>では、オブジェクト指向による設計モデルを図形で直接的に表示・編集・操作するインタフェースを実現する。このような設計支援の構築のため、オブジェクト指向言語CLOS<sup>2)</sup>上に適用型図形エディタ<sup>3)</sup>及びオブジェクト図形入出力言語<sup>4)</sup>という2層のインフラストラクチャを設けている。このうち、適用型図形エディタは、オブジェクト図形入出力言語上で図形からオブジェクトを操作するカスタマイズ可能な汎用インタフェースを提供する。図形エディタにカスタマイズ機構を設けたシステムとしては日本電気の鼎<sup>5)</sup>などがあるが、本稿で述べる適用型図形エディタのウィンドウインタフェースの実現方式は、CLOSの特徴機能である多重継承やメソッド結合を利用したカスタマイズ機構を主な特徴としている。

## 2. 適用型図形エディタ

適用型図形エディタは通常の図形エディタの機能のほかに、これらの機能に連動して、図形と設計モデルとの対応付けと、図形に対応する設計モデルのオブジェクトを操作する機能を含む。適用型図形エディタの主な機能を以下に示す。

## 2.1 部品図形の作成と表示クラスの定義

ユーザが適用型図形エディタ上に描いた部品図形を表示クラスとして定義できる。さらに、部品図形の構成図形の属性(色、線幅など)を、部品の表示クラス<sup>6)</sup>の表示スロット<sup>7)</sup>として定義できる。

## 2.2 部品定義

エディタ上の部品図形に対し設計モデルクラス<sup>8)</sup>を定義できる。これにより、2.1で作成した表示クラスと設計モデルの対応が設定される。

## 2.3 表示オブジェクトの生成と結合

定義した部品は部品メニューに並べておくことができる。部品メニューからコピー&ペーストにより、エディタ画面上に各部品を配置する。このとき、ペーストした部品に対応する表示オブジェクトと設計モデルオブジェクトが生成される。配置した部品間の結線操作も画面上で行なう。

## 3. 実現上の課題

ODETTEにおいては、適用型図形エディタを用いてユーザがアプリケーションを作成し、さらにそのアプリケーションでは、ユーザが自由に定義した様々な型のオブジェクトを扱うので、下記のような柔軟なインタフェースが必要であった。

## (1) アプリケーションに応じたインタフェース

アプリケーションに応じた操作環境を実現するには、ユーザが自分なりに環境をカスタマイズできる必要がある。

## (2) オブジェクトに応じたインタフェース

オブジェクト固有の操作をサポートする機能が要求される。

## 4. ウィンドウインタフェースの実現

## 4.1 ツールキットの構築

まず、基本のUI部品として、ボタン、図形操作ウィンドウ、ポップアップメニュー、ベースウィンドウ(ダイアログのトップレベルウィンドウ)等のクラスを用意した(図1)。

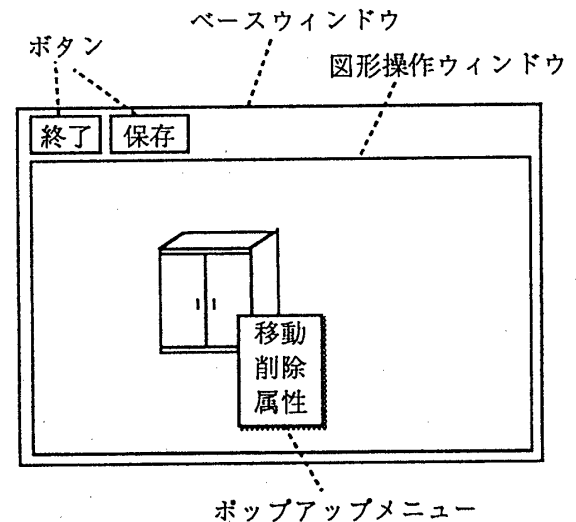


図1 基本UI部品による画面構成例

Window Interface for ODETTE

Shoichi KUBO<sup>1</sup>, Katsuhiko YUURA<sup>2</sup>, Hisashi TAKAHASHI<sup>1</sup><sup>1</sup>Hitachi VLSI Engineering Corp., <sup>2</sup>Central Research Laboratory, Hitachi Ltd.

適用型図形エディタは、この基本部品のみで構成されている。ユーザがさらに自分なりのインタフェースを構築する場合は、これらのクラスを継承するクラスを定義する。

生成・表示・破棄など、基本的な操作手続きは、すべて総称関数で定義されているので、必要に応じて簡単に拡張可能である。

図2に本インタフェースのイベント処理ループを示す。イベント処理の基本方式はXt(Xtoolkit<sup>®</sup>)等と同様のコールバック方式である。個々のオブジェクトのイベント処理はコールバックとして追加定義可能である(図2の(a))。Xt等と異なるのは、コールバック起動関数が総称関数になっていることである(図2の(b))。実際の起動部は、primaryメソッドに定義されているので、ユーザが:before, :after, :aroundメソッドを定義することで、各UI部品の動作をカスタマイズしたり、新たに定義した部品の動作を容易に記述することができる。

例として、感応型ボタンの作成方法について述べる。感応型ボタンとは、マウスポインタの出入りを感じ取るボタンで、マウスポインタがボタン領域内に侵入したときに強調表示になり、マウスポインタが領域外に出たときに通常表示に戻るようなボタンである。これを作成するには、まずボタンクラスを継承した感応型ボタンクラスを定義する。次に、強調表示処理と通常表示処理を行なうイベント処理の:beforeメソッドを定義する。これだけの定義を追加するだけで、新たなUI部品が作成できる。

4.2 汎用図形操作インタフェース

図形操作のための特別な機能を持ったウィンドウとして、図形操作ウィンドウを用意した。このウィンドウには、複数の図形操作メニュー(ポップアップメニュー)を設定することができる。操作メニューは、ユーザが指定した対象や対象の状態に応じて適切なメニューが選択

される。

メニュー選択を実現するために、ウィンドウにメニュー管理用スロットを設ける(図2の(c))。このスロットには、起動条件とポップアップメニューオブジェクトのa-listが入る。ウィンドウ上で特定のマウスボタンが押されたとき、メニュー呼び出し処理部が条件述語を順に評価し、結果が真のものに対応するメニューがウィンドウに表示される。メニューを追加するには、このa-listに起動条件とメニューをペアにして追加すればよい。これにより、起動条件に総称関数を用いて図形のクラスごとにメニューを出すなど、さまざまなカスタマイズが可能になる。

5. おわりに

CLOSの継承およびメソッド結合の機能を用いることで、オブジェクトに応じたインタフェースやアプリケーションに応じたインタフェースを容易に構築できるウィンドウインタフェースを実現できた。現在、これを用いてODETTEの適用型図形エディタはもとより、いくつかのODETTEアプリケーションにおいて専用の設計環境の構築に利用している。

[参考文献]

[1]湯浦、外：オブジェクト指向型設計支援環境ODETTE、情報処理学会第44回全国大会、1992。  
 [2]D.Bobrow,et.al., Common Lisp Object System Specification,X3j13 Document,88-002R(1988)。  
 [3]高橋、外：ODETTEにおけるオブジェクトの図形入出力言語、情報処理学会第44回全国大会、1992。  
 [4]暦本、外：「エディタを部品としたユーザインタフェース構築基盤：鼎」、情報処理vol.31No.5、pp602-611、1990。  
 [5]Adrien Nye and Tim O'Reilly, X Toolkit Intrinsic Programming Manual, O'Reilly & Associates,Inc.

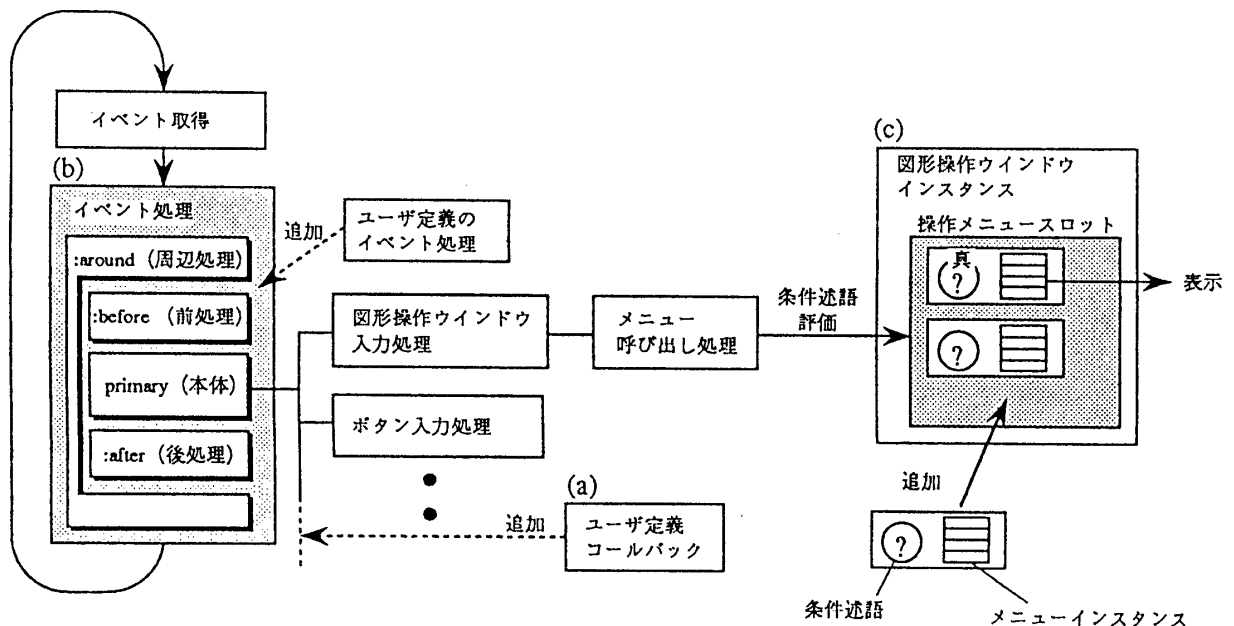


図2 イベント処理とメニュー選択処理の方式