

3 F-11 ODETTEにおけるオブジェクトの図形入出力言語

○ 高橋 久 (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)

久保昭一 (日立超LSIエンジニアリング(株))

1. はじめに

Common Lisp上のいくつかのウィンドウツールキットにおいて、オブジェクトの入出力に関するプレゼンテーションという概念が示されている[1]。プレゼンテーションではオブジェクトの表示もオブジェクトとして実現し、その対応をシステムが管理する。これにより表示の操作および表示操作によるオブジェクト操作も含めて、一貫したオブジェクト指向プログラミングが可能となる。

本図形入出力言語はこの概念をベースに機能拡張を行い、ODETTE[2]における図形によるモデル構築やアニメーションをより容易に記述することを可能とした。本稿では本言語の概要とCLOS[3]を用いた表示クラスの実現方式について述べる。

2. 図形入出力言語の概要

2.1 表示クラス

表示クラスは図形の形式や図形処理メソッドを持つもので、そのインスタンスを表示オブジェクトと呼ぶ。これに対し設計モデルの機能に関するデータやメソッドを持つものを設計モデルクラス、設計モデルオブジェクトと呼ぶ。表示クラスは設計モデルクラスとは独立に、適用型図形エディタ[4]により定義され、部品メニューに登録する。そして表示クラスを図1(1)の様に、部品メニューから画面へコピー&ペーストすることにより、表示オブジェクトと設計モデルオブジェクトを自動生成する。

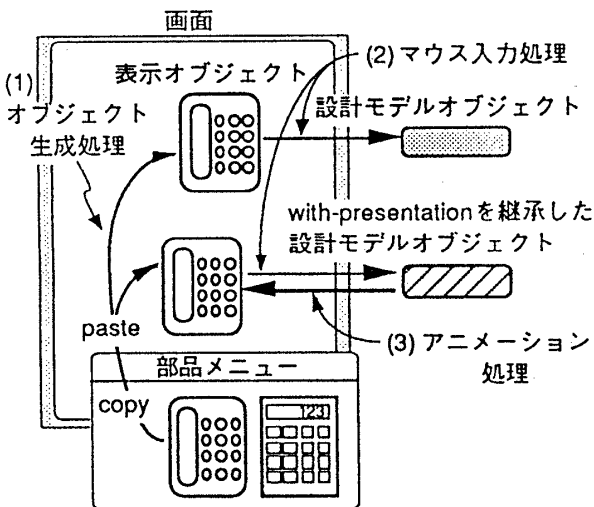


図1 図形入出力の概要

「マウス入力」メソッドでは図1(2)の様に、画面上の表示オブジェクトから設計モデルオブジェクトを獲得する。このメソッドを利用することで、ユーザインタフェースプログラムの作成を可能としている。

シミュレーション時には図1(3)の様に、設計モデルオブジェクトの状態を表示オブジェクトに反映したい。with-presentationクラスは設計モデルオブジェクトから表示オブジェクトを獲得できる様にする概念クラスで、このクラスと表示スロットを利用することで、アニメーションプログラムの作成を可能としている。

表示クラスは一般のクラスと同様に、メソッド結合によるメソッドの利用や、再定義による動的な対応も行うことができる。

2.2 表示スロット

ユーザ図形の一部の属性を表示オブジェクトのスロットとして定義したもので、その値は常に表示に反映される。表示スロットには専用のアクセスメソッド「表示スロット」が用意され、これでのみスロットへの参照・更新を行う。図形属性の更新は図2(1)の形式で行い、一般のオブジェクトの属性更新(図2(2))と同様のスロット更新で、画面上の図形の変形・変色等のアニメーションプログラムを作成できる。

(setf (表示スロット オブジェクト 表示スロット名) 値)

(1) 図形属性更新

(setf (slot-value オブジェクト 'スロット名) 値)

(2) 一般のオブジェクト属性更新

図2 属性更新形式

2.3 表示クラスの継承

表示クラスの継承ではCLOSのクラス継承の他に、他の表示クラスを多重継承して、さらに複合化した(重ね合わせた)図形のクラスを定義することができる(図3(1))。この時表示スロットもCLOSの多重継承のような優先順序で引き継がれる。また、設計モデルクラスと表示クラスを継承した、両者一体化したクラスを定義することも可能である(図3(2))。

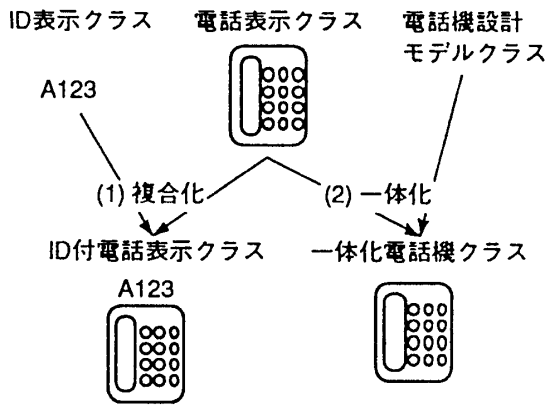


図3 表示クラスの継承

3. 表示クラスの実現

3.1 データ構造

表示クラスはCLOSのクラス定義機能で図4に示す構成で定義した。クラスに共通な図形形式およびダイレクト図形形式は共有スロットとして、図形の表示、移動、消去等の図面操作はメソッドとして持たせた。ここで図形形式とは直線、折れ線、多角形等の基本図形を組合せた式、ダイレクト図形形式とは各表示クラスで直接定義した図形形式を示す。また各表示オブジェクトには設計モデルオブジェクトと、幾つかの表示スロットを局所スロットとして持たせた。

共有スロット	図形形式 ダイレクト図形形式
局所スロット定義	設計モデルオブジェクト 表示スロット1 : :
メソッド	図形表示、移動、消去 マウス入力 表示スロット (self 表示スロット)

図4 表示クラス定義

3.2 表示クラスの継承方法

局所スロットとして定義した表示スロットやメソッドの継承はCLOSのクラス定義機能により実現できる(図5(1))。ところが図形形式の継承は、ダイレクト図形形式が格納されているスロットの値の和集合を取る処理で、CLOSのクラス定義機能だけでは実現できない。そこで以下に示す方式により、図形形式の継承機能を実現した。

(1) 図形形式の作成とその準備

図形形式を作成する方法は、CLOSのAPPEND型のメソッド結合を利用した。このメソッド結合により図形形式を得るためには、各表示クラス定義時に次の準備を行っておく。

- (a) ダイレクト図形形式をスロットに設定する。
- (b) 自クラスのダイレクト図形形式を得る、APPEND型メソッド結合のメソッドを定義する。

すると継承するクラスの優先順に各メソッドを実行し、その各結果をAPPEND(結合)して図形形式を得ることができる(図5(2))。

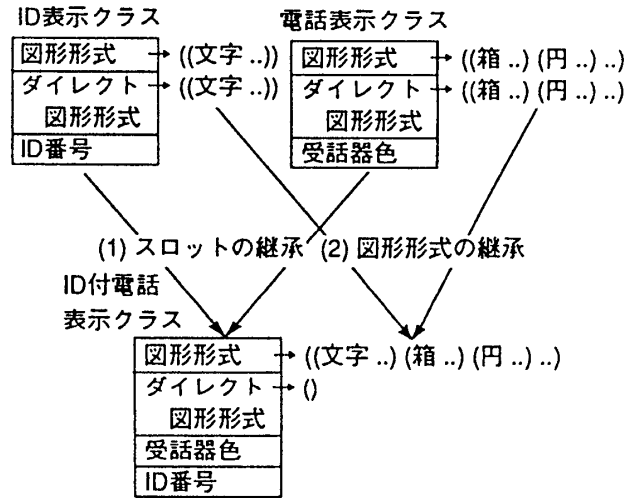


図5 表示クラスの継承機構

(2) 図形形式の作成時期

図形形式の作成時期は、継承するクラスが決定する表示オブジェクト生成時に行う。この時に図形形式スロットに値が設定されていない場合、図形形式を作成し設定する。

またCLOSのクラスの再定義では旧定義クラスのインスタンスを新定義クラスのインスタンスに自動的に変更する機能がある[5]。表示クラスの再定義においては、更に図形形式スロットの値を更新する必要がある。そのため表示クラスの再定義時にそのクラスを継承するすべてのサブクラスに対し、そのスロット図形形式の値を未束縛にする。そしてインスタンス修正時にスロットを更新するようにしている。

4. おわりに

表示クラスの継承機能と表示スロット等を持つ図形入出力言語を用意することにより、図形操作を含めた、オブジェクト指向プログラミングを可能とした。

参考文献

- [1] M. Ida, et. al., A Requirement Analysis for Portable Window System on Top of Common Lisp, 情報処理学会第40回全国大会論文集, pp.919~920(1990).
- [2] 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.
- [3] D. Bobrow, et. al., Common Lisp Object System Specification, X3j13 Document, 88-002R(1988).
- [4] 久保、外：CLOSを利用したODETTEのウィンドウインタフェースの実現、情報処理学会第44回全国大会、1992.
- [5] 高橋、外：HiOBJ-2におけるクラス変更方式、情報処理学会第39回全国大会論文集, pp.1350~1351(1989).