

VOCにおけるマン-マシンインタフェース教示機構

3E-5

— MVCTモデルの提案 —

田代 秀一 岡田 義邦

電子技術総合研究所

1 はじめに

計算機の普及にともないソフトウェアの高品質化、とりわけ、視覚的オブジェクトを画面上で直接操作するといったことを含む、親しみ易いユーザインタフェースが強く求められている。

そこで、我々は視覚的ユーザインタフェースを備えたアプリケーションを効率的に開発することを目的の1つとしてビジュアル言語 voc の開発を進めている [1][2]。voc は、図とテキストを融合させてクラスを定義することのできるオブジェクト指向言語である。手続き的記述はテキストによって行ない、メッセージルーティングは図的に記述することができる。

本稿では、voc の持つ、視覚的ユーザインタフェースの作成に適した教示方式について報告する。

2 従来の問題点

アプリケーション中の視覚的ユーザインタフェース部分を開発する手順は、ボタン、スライダー、ポップアップメニューなどの基本的部品群（ビジュアルオブジェクト）の設計、それら部品の配置、部品とアプリケーションの本体部分との接続の3段階に分けることができる。

これらの作業を支援するために、これまで基本部品群を集めたツールキット、既成部品の実行時の大きさ、形状などを決定する初期化パラメータの設定およびウインドウ上への配置を、視覚的な操作によって教示的に行なえる NextStep, GUIDE 等のツールが作られてきた。

適切な部品とインタフェース作成ツールがあれば、座標を数字で指定するといった複雑な作業を省略して WYSIWYG スタイルで視覚的ユーザインタフェー

class definition window

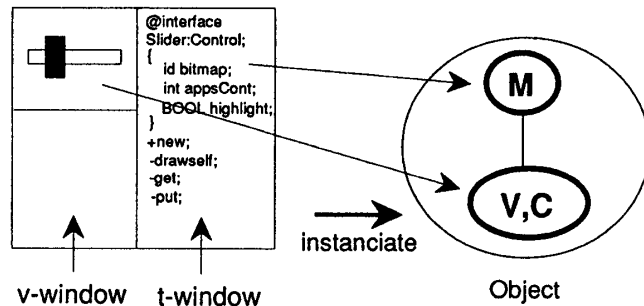


図 1: voc のクラス定義

ス部分を作成でき、その開発効率は極めて高くなり得る。

しかし、これまではツールの対象とする部品があらかじめ用意された既成部品に限定され、プログラマが新規に作成した部品を、既成部品と混在させ、それらと同様にインタフェース作成ツールの下で利用することのできない場合が多く、これがアプリケーション記述上、柔軟性の点で問題となっていた。

この原因は、教示の手段を部品側にではなく、ツール側に持たせていることに起因している。

3 MVCTモデル

Vocではあるクラスの定義を、そのクラスからオブジェクトが実体化された時に画面上に表出される図形を定義する部分 (v-window) と、手続きをテキストによって定義する部分 (t-window) の2つ(少なくとも1つ)によって定義できる。v-windowで定義した図形は実行時にはビジュアルオブジェクトとして利用者との間で直接操作による対話を行なうことができ、この対話に基づく処理の本体はテキスト部分に定義される。これを SmallTalk-80 の MVC モデル [3] に照らせば v-window が View および Controller、t-window が Model に対応する (図1)。

これら MVC にプログラムを教示する手段 (Teaching-

Teaching mechanism for man-machine interface on VOC, based on MVCT model

Shuichi TASHIRO, Yoshikuni OKADA

Electrotechnical Laboratory

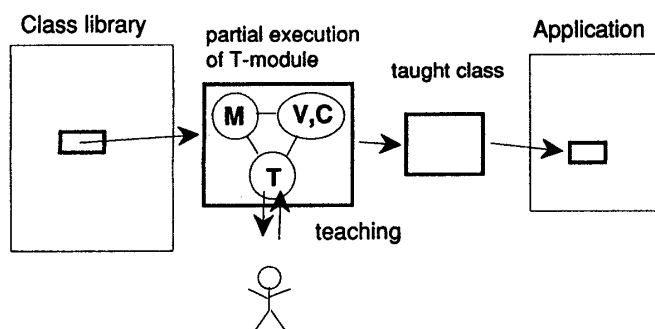


図 2: MVCT 型クラスの利用

module) を加えたモデルが MVCT モデルである。

クラスの定義時には一部のパラメータを未定にしておき、実体化時に初期値を与えてカスタマイズして利用することは、クラスライブラリに汎用性を持たせるために有効である。

Teaching-module はこのカスタマイズに関わる手続きを実行するモジュールである。これまではインタフェース作成ツール側に持たせていた、WYSIWYG スタイルの被教示手続きを含むカスタマイズ手続きを内蔵させる。

例えば、ボタンというクラスを汎用的なライブラリとして定義したい場合、ボタンクラスの定義に、ボタンの基本機能の定義にあわせてボタン形状の最終仕様を決定する手順、すなわち「画面にボタンを表示し、その一部をマウスポインタでつまんで引張るなどの直接操作によりボタンの形、大きさを決定する」といった被教示手続きを内蔵させる。もちろん典型的な被教示手続きは上位のクラスに定義し、そこから継承することになる。

4 教示と実体化

図 2 に、MVCT 型クラスを使ってアプリケーションを作成する手順を示す。MVCT 型クラスは 2 段階のステップを経てオブジェクトとして実体化される。

第 1 段階でクラスの中の教示モジュール (T module) を実行し、それと利用者が対話することで教示を行なう。教示作業の結果、それまで未定であったパラメータの値が決まり、教示済みクラスが作成される。教示済みクラスは、未教示クラスのサブクラスとして位置付けられる。

第 2 段階でこの教示済みクラスをアプリケーション中に組み込み、実行することによりオブジェクトが生成される。

5 教示モジュール

教示モジュールはクラス定義ウィンドウ中のテキストプログラム中に、@teachmodule というキーワードに続くセクションとして記述する (voc のテキストプログラムの文法は Objective-C を拡張したものである)。

教示の結果決定する値を記録するための記憶クラスとして、persistent 記憶クラスを用意している。persistent 記憶クラスとして宣言した変数に書き込まれた値は教示済みクラスのクラス定義テーブルに記録され、ファイルとして保存される。

教示済みクラスからオブジェクトが実体化される時、この変数の値は教示時に記録した値に初期化される。

6 まとめ

今回提案した MVCT モデルに基づいたプログラミングスタイルをとることにより、教示的に最終仕様を決定できる汎用的なユーザインタフェース用部品を自由に作ってライブラリに登録することができる。このことにより、プログラマは WYSIWYG スタイルの簡便さと、既成品にとらわれない大きな自由度とを同時に享受することができる。

又、教示を終えた部品がひとつの独立したクラスとして登録されるため、教示結果はひとつのアプリケーションに限定されることがなく、他のアプリケーション用に再利用することができる (教示結果の共有)。

教示済みクラスを再教示することも可能であり、過去の蓄積を基礎にしてアプリケーションないしはライブラリを少しずつ改良して行くことができる。

MVCT モデルをとりいれた voc は、現在 NeXT コンピュータ上で開発を進めている。

本研究の機会を与えていただいた棟上情報アーキテクチャ部長に謝意を表す。

参考文献

- [1] 田代, 岡田: “図的記述機能を持つオブジェクト指向言語 voc”, 情処研究会報告 89-MIC-55, '89 年 5 月
- [2] 田代, 岡田: “ビジュアル言語 voc プロトタイプの試作”, 第 40 回情処全大会, '90 年 3 月
- [3] W. Cunningham “Smalltalk-80 によるアプリケーション・プログラムの作り方”, bit, Vol.18, No.4, pp.379-396