

5G-7

オブジェクト指向による
画面エディタの部品化

天海 良治

(NTT電気通信研究所)

1. はじめに

LispマシンELISに構築中の知能的プログラミング環境NUE(New Unified Environment)では、現在エディタとしてEmacsライクな画面エディタZEN(Zeroth Editor for NUE)が稼働している。ZENはELISの核言語TAOで記述したが、主にTAOのオブジェクト指向の機構を利用して入出力に係る部分を部品化し、その部品の集合から画面エディタを構成した。これらの部品は他のプログラムからの利用も可能である。

ここでは、ZENの構成、部品の利用法などについて述べる。

2. ZENの構成

ZENは強力なエディタの利用者カスタマイズ機能と、一般プログラムからのエディタ機能の利用を目的として、オブジェクト指向の機構を用いて、キー入力、バッファ操作、再表示機構など利用者インターフェースに関わる部分を部品化した。これらとエディタとしてのコマンドテーブル、ファイル管理の機能などとあわせて画面エディタとなる。

3. オブジェクト指向と部品化

TAOのオブジェクト指向の機構は、多重継承が可能なクラスをもち、そのメソッドの結合法も豊富に備えている。また、TAOのLispの機構と自然なかたちで融合できる。

今回の部品化に当たっては、次のような点でオブジェクト指向が生かされた

- ・このエディタでは、多くのバッファが独立に存在し、各々、名前、現在の注目点などコンテキストをもつ。よって、バッファをオブジェクトとすると自然なかたちで扱える。

- ・TAOオブジェクト指向の多重継承、メソッド結合の機構が、既存の部品を組み合わせるとき、部品の前後に処理を付け加える場合にうまく利用できること。例えば、上位クラスのメソッド内で使用しているメソッドを下位クラスで用意すれば、上位メソッドの変更なしに上位メソッドの動きを制御できる。プログラム例を示す。

```
(defclass A () ()) ; クラスAの宣言
```

```
(defclass Sub () ()(A))
```

; Subのスーパークラスは A

```
(defmethod (A X) () ; クラスAのメソッドX
... [self error-report] ...)
```

```
(defmethod (A error-report) () (bell))
```

```
(defmethod (Sub error-report) ()
```

```
(screen-flash))
```

クラスSubにメソッドXはないとする。

クラスSubのインスタンスをOBJとする。

ここで、メッセージSEND[OBJ X]は、部品として上位クラスAのメソッドXを利用することをあらわす。だが、error-reportはSubのものが使われるので、エラー時にベルはならない。

ZENのクラスの一部をあげる。

zen-link

バッファ名、文字列のダブルリンクリスト等、
バッファの実体

zen-buffer

バッファの注目点、ウィンドサイズ等、
バッファのコンテキストにあたるもの、ひとつのzen-linkに対して複数存在しうる。

shi-zen (zen-buffer のサブクラス)

zen-buffer + ファイル管理

エディタとしてのバッファになる。

search-buffer (zen-buffer のサブクラス)

zen-buffer + インクリメンタルサーチ
のためのコンテキスト

zen-class

他のプログラムとエディタとしてのZENとの
インターフェース用

これらのクラスのもとに、400あまりのメソッドが定義されている。

4. 部品とその利用

これらのメソッドのうち部品として利用できるのは、次のようなものである。

- ・画面表示にかかわるもの

画面上に残っている文字を再利用して文字の転送量を減らす表示方式が備わっている。また文字

Software Components of Screen Editor in Object Oriented Style
Yoshiji AMAGAI
N.T.T. ECL

の反転, 下線, 点滅にも対応している。画面の分割表示の機構も利用できる。

・入力のエディットの機構

注目点の移動, 検索, 文字の挿入, 削除, ローマ字かな変換, かな漢字変換などのエディタの機能。ファイル名などの入力において文字列を補う機構等が利用できる。これらは, 画面表示と結びついている。

・その他

文字列上でのLispの要素の切り分け, プリントアウトなど。

部品の利用法

次のようにいくつかの方法がある。

・エディタとして利用するもの

エディタ開始時に引数としてファイル名, 文字列, コマンド列等をわたすインターフェースが用意されている。mailなどで使用している。

・ZENのバッファを仮想デバイスとすること

open, read, write などの入出力関数でバッファを扱う。バッファに書き出したものはエディタで編集可能なかたちになる。

・新たなクラスを作成すること

コマンドに対応するメソッドとコマンドテーブルを作成する。コマンドはTAOで記述することになる。このときはバッファの構造などある程度ZENの内部の約束を知っている必要がある。例えば, クラスsearch-bufferは, 独自のコマンドテーブルをもっていて, 部品利用のかたちでサーチを実現している。

5. おわりに

ZENの部品化は, オブジェクト指向という枠

組みを利用してLispマシン上のLisp環境という均質な場に溶け込むような自然な実現法をとっている。これは操作の共通化, 利用者対応の共通化に役だっているばかりでなく, TAO/ELISのマルチユーザ, マルチプロセスの世界でコードのシェアなど実効的利益もある。だが, 知的プログラミング環境として全体をみたとき, トップレベル, エディタ内, ユーティリティのコマンド待ちなど入力状態一つをみても種々のものがあるし, カレントディレクトリ, ファイルのありか, 関数のソーステキストのありか, デバイスの直接アクセスなどLispの均質さと相いれない部分が多く残っている。これらは広い意味でモードととらえることができる。現在のプログラミング環境は利用者が現在のモードを常に意識していないといけない。知的なプログラミング環境においては, このようなモードはないのが望ましい。ZENの部品化はこのモードレス環境構築の足掛りと考えている。

参考文献

- [1] Richard Stallman: "GNU Emacs Manual", Third Ed. Emacs Ver. 17 (Dec. 1985)
- [2] 鈴木則久編: "オブジェクト指向 解説とWOC'85からの論文", 共立出版, 1985
- [3] H.G.Okuno, I.Takeuchi, N.Osato, Y.Hibino and K.Watanabe: "TAO: A Fast Interpreter-Centered System on Lisp Machine ELIS" in Conf. Record of the 1984 ACM Symposium on Lisp and Functional Programming (ACM, Austin, Tex) (Aug. 1984)
- [4] I.Takeuchi, H.G.Okuno, N.Osato: "TAO-A harmonic mean of Lisp, Prolog and Small-talk", ACM SIGPLAN Notices, Vol. 18, No. 7, 1983.

