

OA用統合型簡易言語Macro-Cにおける対話制御

6F-5

草鹿庸次郎 野寄雅人

常葉学園浜松大学

1. はじめに

Macro-Cはパソコンを使うエンドユーザにも良質のOA用アプリケーションが作成できるようなプログラム開発環境を提供することを目的とした簡易言語である。

可能な限り非手続き部を増やし、残る手続き部においても、高機能ながら汎用性を損なわず、しかも内部処理が完全に隠ぺいされた部品を提供してプログラミングのウエートを大幅に軽減している。

そのために、"インストラクションに従って行動するアクティビティ"という概念を考案し、その概念を用いてテンプレートプログラムにインストラクションの指示内容を書き加えることによってC言語プログラムを発生させるという手法を採った。さらにまた、インストラクションの中にアクティビティが実行すべきサブプロシージャを指示できる仕組みを組み入れて高度なマクロ化を可能とした。

図1は、Macro-Cによって作成した販売管理システムの伝票入力画面である。このような画面を使ったデータ入力処理などが容易に実現できる。



図1 売上傳票入力画面

2. アクティビティとインストラクション

Macro-Cにおけるアクティビティは、ある一連のまとまった仕事を遂行する、従来型高級言語においてサブルーチン、関数、マクロルーチンといわれているものを発展させたものである。

Macro-Cにおいては、従来型高級言語と同様な引数を受けて動作するアクティビティのほかに、与えられた指示書(インストラクション)に従って忠実に行動する能動性の高いアクティビティが利用できる。後者を第1種のアクティビティ、前者を第2種のアクティビティという。

インストラクションに従って行動する第1種のアクティビティの一覧を表に示す。

表1 第1種のアクティビティ

アクティビティ名	機能概要
[画面を使うアクティビティ]	
INPUT_C	画面上のカード内のフィールドにデータを入力する
INPUT_T	表形式で画面上のテーブル内にデータを入力する
INPUT_TXT	画面上のテキストエディット領域にデータを入力する
INPUT_SHT	画面上のスプレッドシートにデータを入力する
[ファイルを使うアクティビティ]	
READ_C	ファイルのデータをカード内のフィールドに読込む
READ_T	ファイルのデータを連続的にテーブルに読込む
READ_P	ファイルのデータを印刷等のために連続的に読込む
READ_RAM	ファイルのデータをソートしながらRAMに読込む
READ_TXT	ファイルのデータをテキストエディット領域に読込む
READ_SHT	ファイルのデータをスプレッドシートに読込む
WRITE_T	テーブル内のデータをファイルに書込む
[プリンタを使うアクティビティ]	
PRINT_P	ファイルから読込んだデータをプリンタに出力する

インストラクションには、図1のような画面(ウインドウ)を使うアクティビティに対するインストラクション(ウインドウインストラクション)のほかファイルインストラクション、プリンタインストラクションがある。

図2における項目は、アクティビティに対する指示項目を示している。このインストラクションはプログラムエディタで手続きを入力する際、ウインドウインストラクション設定用のウインドウを開けて必要事項を埋めるようになっている。

3. アクティビティに対する対話制御

(1) メニュー選択制御

OA用アプリケーションプログラムにおいては、ファンクションメニュー、ボードメニュー(ポップアップメニュー、プルダウンメニュー)、アイコンメニューなどによる選択処理は欠かせないものとなっている。Macro-Cにおいては今のところファンクションメニューとボードメ

```

window_no 0
sflld_instruction
  No pre_proc  post_proc  onKey_pro1 onKey_pro2 onKey_pro3 crsr msgNo
  0
  1
  2
  3          PROC_1      3      0          1 . 当月売上順 (24)
  4          PROC_A      4          2 . 当月残額順 (25)
  5          PROC_A      5          3 . 累計売上順 (26)
  6          PROC_A      6          4 . 終了 (27)
  7          PROC_A      7          K_1:READ_RAM(INST_A)
msgNo message
  0 0..資産 1..負債・資本 2..売上・収入 3..原価・経費(0,24,4) K_2:READ_RAM(INST_B)
direction= K_3:READ_RAM(INST_C)
procedure K_4:BREAK
onKey_pro1=          onKey_pro2=          onKey_pro3=

```

図3 記述例

図2 ウインドウストラクシヨ

ニューが利用可能で、FUNC_MENUアクティビティとBOARD_MENUアクティビティ（いずれも第2種のアクティビティ）を用いて行う。BOARD_MENUアクティビティの記述例を図3に示す。

図3のようにすると画面上の(60, 14)の位置にボードを表示し、さらにMENU以下に書かれたメニュー項目を表示し、キーボードからの選択入力待。

カーソルキー ↓, ↑ を押すと棒状カーソルが上下し、リターンキーを押すと棒状カーソルの位置にある項目が選択され、このアクティビティの処理を終了する。

1から9のキーあるいはZのキーを押しても項目が選択され処理を終了する。終了したあとの処理は押されたキーに従って、K_1:K_2:などのあとに記述されたステートメントに従う。K_1は1のキー、K_2は2のキー、K_ZはZのキーを表す。上記のキー以外は受けつけない。

FUNC_MENUアクティビティも同様な対話制御のしくみをもつ。

(2) アクティビティの処理終了制御

第1種のアクティビティは、画面上のフィールドに繰り返しデータを入力したり、ファイルからデータを繰り返し読み込んだりの繰り返し処理を実行する。その繰り返し処理をMacro-Cにおいては任意のキーの押下で中断・終了させることが可能である。

READ_Tアクティビティの記述例を以下に示す。

```

READ_T(INST_A)
MENU
      06印刷(5) 10終了(6)
F06:
F10:

```

記述例のようにするとまず画面下のf・6とf・10のファンクションボックスにそれぞれ”印刷”, ”中止”と表示する。そのあとインストラクションに記述された検索条件(filter)に合うものを、指定したファイルから連続的に読み込み画面のテーブル上に一覧表形式で表示する。カーソルキー ↑, ↓ を押すとテーブルの表示領域がスクロールするのですべてのデータを画面上に見ることができる。

ファンクションキー「f・6」あるいは「f・10」を

押すとこのアクティビティの処理を終了する。終了したあとの処理は、押されたキーにしたがってF06:, F10:のあとに記述されたステートメントに従う。

(3) インストラクションを用いる対話制御

第1種のアクティビティのうち、画面を使うアクティビティはウインドウストラクシヨに従う。

図2に示したウインドウストラクシヨの中の指示項目のうち3行目にあるonKey_pro1~3, およびprocedureのところにあるonKey_pro1~3は画面を使う第1種のアクティビティに対して対話的なやりとりを可能とする機構を与えるものである。

onKey_proc欄に、F03:PROC_AとかG_Z:PROC_Bとかのように記述すれば画面を使ってデータを入力している段階で、「f・3」のファンクションキーを押すとPROC_Aというサブプロシージャが呼び出され、GRPHキーを押しながらZのキーを押すとPROC_Bというサブプロシージャが呼び出される。

4. マルチウインドウを用いての対話制御

OA用アプリケーションプログラムにおいては、複数のウインドウを適宜表示させて対話的に処理を進めていくマルチウインドウ機能が欠かせないものとなっている。

Macro-Cにおいてもマルチウインドウ機能がサポートされている。カードへのデータ入力時あるキーを押すとサブウインドウが開きファイルのデータが読み込まれ、サブウインドウ上のテーブルに表示されるといったことが可能である。表示されたデータの中の任意のデータを別のウインドウ内に転記(カット&ペースト)することもできる。

5. おわりに

本稿で示したようにMacro-Cにおける対話制御は、上位の手続き(プロシージャ)とアクティビティに対するインストラクションの記述の中で実現可能であり、プログラム作成者は各アクティビティの内部処理に関わる必要がない。しかもプロシージャ内の記述は簡明であるので対話制御は容易に実現できる。