

SIMPOSライブラリアンの カスタマイズ機能

見玉泰子¹ 藤本誠司²

^{1,2} 三菱電機東部コンピュータシステム(株)

1. 概要

SIMPOSは、逐次型推論マシンPSIのオペレーティング・システムである。ライブラリアンは、SIMPOSのプログラミング・システムの一つで、ESPのクラスやパッケージに対する操作機能をウィンドウやメニューを用いて提供する。

カスタマイズ機能は、ユーザがライブラリアンのメニューやコマンド等の標準インタフェースを作り替えるための機能であり、ライブラリアンの操作性の向上を目的としてSIMPOS5.0版より導入されたものである。

本発表では、背景となるライブラリアンの標準的なインタフェースを説明した上で、カスタマイズ機能のユーザ・インタフェースを紹介する。

2. ライブラリアンのプログラミング環境

SIMPOSのシステム記述言語ESP¹⁾では、プログラム・モジュールの単位としてクラスという概念を用いている。そしてライブラリと呼ばれるサブ・システムが、このクラスやクラスの集合であるパッケージの管理機能を持つ。SIMPOS上で動作する全てのクラスはライブラリに登録されている。ライブラリアン²⁾は、クラスやパッケージの登録/削除/検索、継承/参照関係の参照、実行コードの管理等のライブラリの機能をコマンドとして提供する。これらのコマンドは、SIMPOSのウィンドウ・システムを利用し、コマンド・メニュー形式で提供されている。(図1参照)

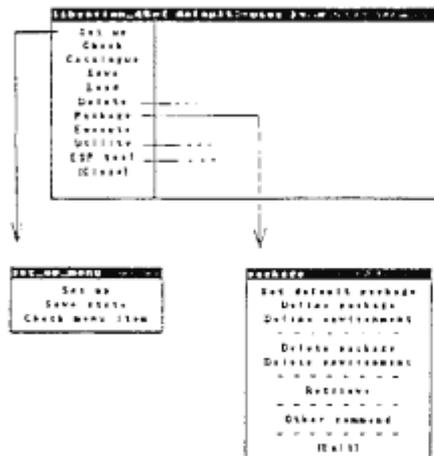


図1 標準のライブラリアン初期画面
(一部)

操作方法は、まずマウス・クリックでメニューからコマンドを選び、次に必要なパラメータ(クラスの登録ならばソース・ファイル名)をテキスト・ウィンドウに入力するというものである。コマンドはほとんどがメニューから選択できるようになっており、わずかなキー入用とマウス・クリックで全ての操作が行える。

3. カスタマイズ機能の導入

従来は、ライブラリアンのコマンド及びコマンド・メニューの構成はシステム提供のものに限られていたため、ユーザの細かな操作面での要求(頻繁に使用するコマンドをトップ・メニューに組み入れたい等)や、コマンドの追加/仕様変更の要求に、SIMPOS改版時以外対応することができなかった。そこで、ユーザ単位で標準のライブラリアンのコマンドやメニュー構成を使用形態に合わせて作り替えるためのカスタマイズ機能を導入した。これによって、システム側で対応しきれない細かな要求に応える事が出来るようになった。カスタマイズ機能は、具体的に以下の3機能から成る。

- ① ライブラリアンの部品クラスを利用したプログラムの作成により、コマンドを追加/変更する機能
- ② コマンド・メニューをその階層構造も含めて組み替える機能
- ③ 1, 2の応用として、ライブラリアンの入出力部分(ウィンドウやメニューの構成)を、そっくりユーザ定義のものに取り替える機能

次にこれらの機能の各々について例を用いて説明する。

4. カスタマイズ機能のユーザ・インタフェース

プログラム作成によるコマンドの追加/変更には、ESP言語のクラスの継承機能を利用する。まず、システムで用意されたライブラリアンの機能を継承するクラスを定義する。そこに新しいコマンドを定義したクラスを追加継承することにより、システムで用意されたライブラリアンの機能と、新しいコマンドの機能を合せ持つライブラリアンを作成することができる。

このコマンドを追加した新たなメニューを作成するには、ライブラリアンの初期化ファイルを用いる。初期化ファイルには図2に示すようなメニュー定義文が書かれている。このメニュー定義文を、エディタ上で書き替えることによってコマンド・メニューの組み替えをする。

Function of User-customizing on SIMPOS librarian

Yasuko kodama¹, Seiji Fujimoto²

^{1,2} Mitsubishi Electric Computer Systems(Iokyo) Co.,

メニュー定義文は、コマンド名とそれに対応するメニュー項目の対を要素とする項の形式で表わされており、項の中での階層が、トップ・メニュー/サブ・メニューといったメニューの階層に一致する。従って、コマンドを追加/削除するだけでなく、使用頻度に合わせてコマンド・メニューの階層構造を変更することも容易である。書き換えられたメニュー定義文をもう一度初期化ファイルにセーブしておけば、その次から新しいメニュー構成を持つライブラリアンが生成される。(図3参照)

```

menu_list :=
{
  { "Set up", set_up_menu,
    {
      { "Set up", set_up},
      { "Save state", save_state},
      { "Check menu item", check_menu_item}
    }
  },
  { "Check", check},
  { "Catalogue", catalogue},
  { "Save", save},
  { "Load", load},
  { "Delete", delete_menu,
    {
      { "Delete", delete},
      { "Abolish", abolish},
      { "Forget", forget},
      { "Delete resource", delete_resource},
      { "Exit", up}
    }
  }
}

```

図2 標準のライブラリアンのメニュー定義文

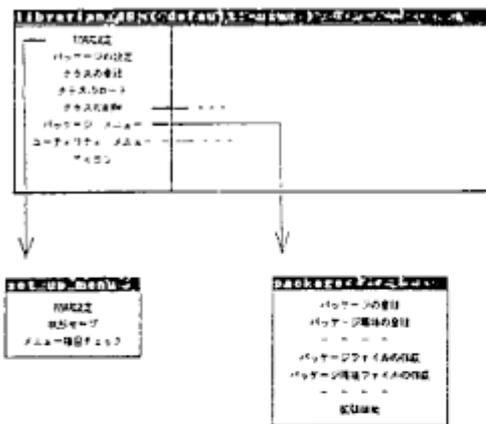


図3 メニューを日本語に組み替えた例(一部)

更に前節④で述べたように大抵なインタフェースの変更機能がある。これは、ライブラリアン自身の構成が、図4で示すようにコマンドの実行部、入出力処理部、更にコマンドの実処理部分を受け持つプログラム・インタフェース部に3分割されている点を利用したものである。この構成を利用して入出力部分をそっくり取り替えて全く違う種類のウィンドウやメニューを使ったり、プログラム・インタフェース部だけを使って他のサブ・システムでライブラリアンの機能をシミュレートすることが可能である。

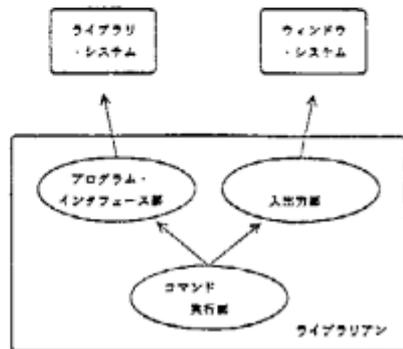


図4 実際は、各部の取扱順序を再す

図4 ライブラリアンの内部構成

5. 今後の課題

現在、SIMPOSは第6.0版の開発中で、カスタマイズ機能については、以上で述べてきた機能に加え、コマンド・メニューの組み替えに用いる初期化ファイルを複数ファイルから選択することを可能にする改良を行っている。

また将来的な展望としては、今回ライブラリアンに提供したカスタマイズ機能を一つのモデルとして、SIMPOSの他のサブ・システムへの応用を検討していきたい。

参考文献:

- 1) 近山, "ESP Reference Manual", ICOT IR-044, 1984
- 2) 小型化PSI/SIMPOS操作説明書, 基本編 (SIMPOS5.0版) ICOT, 1988