

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

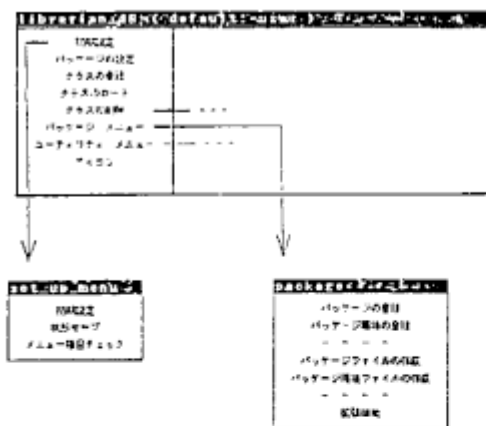


図3 メニューを日本語に読み替えた例(一部)

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。

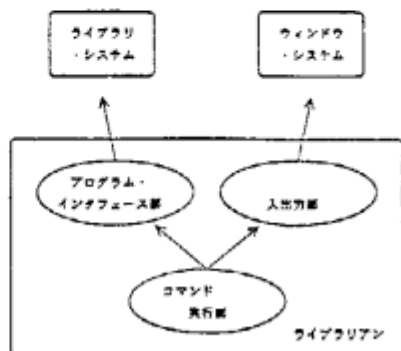


図4 実際は、各部の参照関係を示す

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・メニューの読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 (SIMPOS.0 版) ICOT, 1988