

ソフト部品利用を基盤にしたプログラマズ・ワークベンチ(その1)

1F-7

-全体構成と概要-

・高橋弘利、辻田英雄、高原錦一、山岸 明

((株) 東芝 総合情報システム部)

1. はじめに

アプリケーション・システムの開発において、その業務を支援するPWB (Programmer's Work Bench) の整備が従来より行われており、生産性の向上に効果を上げている。特にプログラムの製造工程における生産性向上の手段として、ソフトウェアの再利用が注目されているが、今回我々はソフトウェアを部品化し、それらを再利用してソフトウェア開発を行なうという考え方を具体化、実現した。本発表では、ソフト部品利用を基盤にしたプログラマズ・ワークベンチPWB/SP (Programmer's Work Bench based on Software Parts) についてその全景を報告する。

2. PWB/SPの構成

PWB/SPの構成は図1に示すように、①ソフト部品の製作ツールとして整備した部品作成支援システム (SP-GEN)、②部品のオンライン検索やソースコードへの組込みをサポートする部品利用支援システム (SP-UTL)、及び、③部品ライブラリ (SP-LIB) からなる。SP-LIBは現在約600本のソフト部品が整備されており、PWB/SPは、当社32ビットミニコンのTSSの下で動作する。本稿では、SP-LIBについて、その特徴を以下に報告する。尚ソフト部品としては、

- ①プログラム部品 (サブルーチン、モジュール)
 - ②論理構造部品 (処理パターン、スケルトン)
 - ③データ定義部品 (ファイル、画面等)
 - ④パッケージ部品 (完結したプログラム)
- 等がある^[1]が、ここでは、①のプログラム部品でかつ、利用者が内部構造を意識しなくてよいブラックボックス部品に限定して考える。

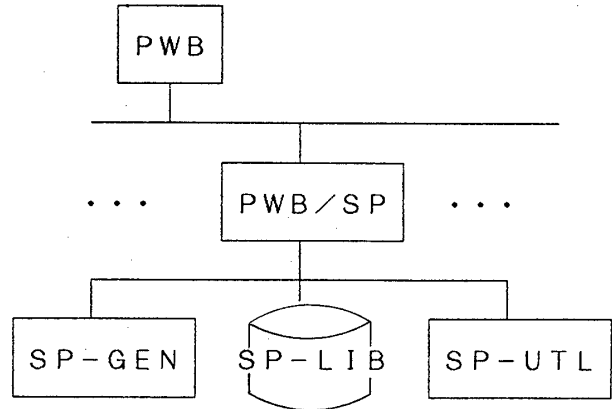


図1 PWB/SPの構成

3. 部品ライブラリ (SP-LIB) の特徴

SP-LIBに登録されている部品について、その特徴を以下に述べる。

(1) 体系的アプローチ

ソフト部品は、統一化された設計概念で作成されていることが望ましい。既存のソフトウェアを切出して、部品化した場合、設計者の個性が出てしまい、帰って使いづらい部品ライブラリになってしまう。SP-LIBに登録してある部品群は、全て、新規に作成した物であり、引き数の並び、リターンステータスの値、部品名等を統一化している。

また、サブルーチンの形態で部品を再利用する場合、引数の誤りによるトラブルを避ける必要がある。SP-LIBでは、引数の個数及びその値の範囲の妥当性について、判断し、エラー発見時にはコンソールロギングまたはタスクアポートを行なうしくみを部品側に持たせている。

次に、ソフト部品の分類・体系化については、図2に示すように、一般のソフトウェアをデータの空間的、時間的な移動に着目して、モデル化し、解析・変換・操作・編集・制御の5つの部品群に分類し、さらに詳細化を行い、図3に示すような体系となっている。

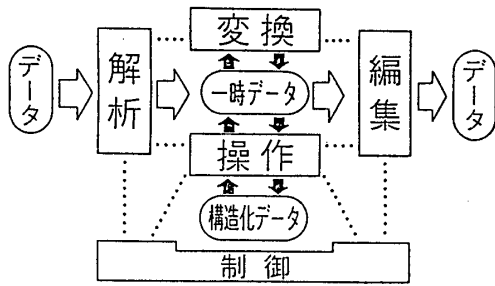


図2 データから見たソフトウェアの一般モデル

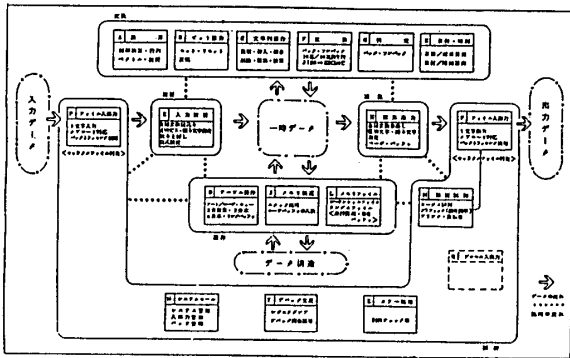


図3 SP-LIBの体系

(2) 移植性の考慮 - C言語での開発 -

従来、部品としてはアセンブリ言語で記述することが多く、OS（オペレーティング・システム）が変る度に作り直すとか、FORTRANやCOBOLというような高級言語では、細かい記述ができない、あるいはリンケージ規約の制約から相互呼出しができないOSもあるという問題があった。我々は、言語としては移植性の高いと言われているC言語を採用した。具体的には、約600本の部品のうち、SVC（スーパーバイザコール）マクロなど一部を、アセンブリ言語としたが、約80%はC言語で記述している。

(3) ソースコードとドキュメントの一体管理

ソフト部品の取扱説明書は、特にブラックボックス部品では重要である。SP-LIBでは、図4に示すようにドキュメントを定型化し、すべての部品に対して、用意している。特徴的なのは、ソースコードと一体化している点で、ドキュメントとソースコードを一元管理することにより、内容の不一致によるトラブルやバージョンアップの際の更新ミスを生じないようにしている。

4. おわりに

PWB/SPでは、SP-LIBに対して、①部品の登録をしたり、②部品を検索活用するシステムが用意されている。これらについては、後続の発表で述べる。

```

/* S */
/*
A0 [1] 識別情報
SP
A1 識別記号      : FAF3S-PSAF-031
A2 ソフトウェア名称 : 16進文字列の整数への変換
A4 サブルーチン名  : FAF031
A5 キー・ワード   :
A7 著作権         : (株)東芝
SP
A8 REV01.00 ..... 昭和59年11月22日
SP
B0 [2] 適用仕様
SP
B1 機種          : FAPSCON
B2 O S           : SR8F
B3 記述言語      : C
B4 呼出言語      : COBOL, FORTRAN
B5 関係編集属性  : PURE, MUSE
B6 オブジェクトサイズ : 0.5KB
B7 スタックサイズ : 0KB
B8 バッファサイズ  : 0KB (B. POOL= )
SP
C0 [3] プログラム仕様
SP
C1 機能説明      :
C2 16進文字列を整数に変換する。
SP
C3 呼出形式      : CALL 'FAF031' USING ARG1 ARG2 ARG3 ARG4.
SP
C5 引数説明      : 引数; データ型; 繰返数; 入/出; 説明
C6 ARG1; X; n; 入力; 16進文字列
C6 ARG2; I; 001; 入力; 16進文字列のサイズ
C6 (0 ≤ n ≤ 8)
C6 ARG3; I; 001; 出力; 整数
C6 ARG4; I; 001; 出力; リターンステータス
C6 = -1; パラメータエラー。
C6 = 0; 正常。
C6 = 2; 16進文字列に誤りがある。
SP
/*

/* M */
/*
SP
C7 処理概要      :
C8 1) ARG2が0より小さい場合又は8より大きい場合パラ
C8   メータエラーとし、処理を終了する。
C8 2) ARG1が16進文字列以外ならばエラーとし、処理を
C8   終了する。
C8 3) ブランクを除く16進文字列を整数に変換し、ARG3
C8   に報告する。
SP
C9 注意事項      :
CA 1) ARG1で使用可能な文字は、“0”～“9”、“A”
CA   ～“F”のみとし、それ以外は誤りとする。
CA 2) サイズはバイト単位としARG2が0である場合、AR
CA   G3は不定、ARG4は0となる。
SP
D0 [4] 外部参照情報
SP
D3 外部サブルーチン : FAZ003, FAZ004
D9 インクルード文   : FAF3S.H
/*
/* E */
#include <FA3S.H>
#define FNC_NAME "FAF031"
#define A_ARGMIN 3
#define A_ARGMAX 4
FAF031(ARG1, ARG2, ARG3, ARG4)
TEXT ARG11;
INT32 *ARG2, *ARG3, *ARG4;
{
    STATIC TEXT S_FONAME[1]=FNC_NAME, S_ERCODE[1]="PE01";
    STATIC INT32 A_ARGLST[1]=(A_ARGMIN, A_ARGMAX);
}
    
```

図4 ドキュメントと一体化されたソースコード

参考文献

[1] 山本欣子: "ソフトウェア再利用の実践 - パネルの背景とねらい-", 第5回ソフトウェア生産における品質管理シンポジウム, PP.241-242, (1985)