

ソフトウェア設計の基本(その5)

— 部品の考え方 —

3 J-1

今井 恒雄、萩田 直史、増田 秋俊(富士通) 平木 しげ子(東京家政学院筑波短大)
 荒木 雄豪(京産大) 古谷 千恵(東大生研)

1. はじめに

オープン化、ダウンサイジングが進む中で、これまで以上に現場の業務担当者が自らシステムを開発できる仕組み、すなわち『誰にでもできるOA化』の仕組みが必要になっている。我々はこれまで、大学や高校の事務システム(入試、学籍、成績、就職など)の開発や言語教育を通して、このためのシステム開発、ソフトウェア設計の考え方の研究と実践を進めているが、今回は、『部品の考え方』について紹介する。

2. 我々の考える部品

システムを開発するときには、新たに作る部分をできるだけ少なくすることが重要であり、このためにはこれまでの資産をできる限り使うことである。その考えに立てば、部品はサブルーチンに限られるわけではなく、システムを開発するときを使う色々なものを『部品扱い』にしても不自然ではないと考え、以下のものを部品として使っている。

◆ サブルーチン

- ・インタフェースの統一
- ・雛形プログラムへの組み込み

◆ 各種プログラムなどの自動作成システム(DENTRY)

- ・雛形プログラム
- ・データエントリプログラム
- ・情報検索プログラム
- ・レコード設計レイアウト印刷

◆ フローチャート

◆ 汎用ソフト

- ・エディタによるデータエントリ

3. サブルーチン

3.1 インタフェースの統一

サブルーチンを作る上で極めて重要なのは、インタフェースを統一することである。我々はサブルーチン利用のインターフェースを【図表1】のように統一している。

3.2 サブルーチンの種類

- ・汎用部分のサブルーチン化

機能が明らかで、どのようなシステムでも利用できるような汎用的なものはサブルーチンとして作成する。

- ・難しい部分のサブルーチン化

システム開発の作業分担をする時に、論理的、技術的に難しい部分(例えば卒業判定処理)は、技術力のあるメンバがサブルーチンとして作成する。

3.3 雛形プログラムへの組み込み

- ・ファイルの割り付け、デバッグ用サブルーチンなど、よく使用するものは、【図表2】のように雛形プログラムにあらかじめ組み込んでいる。

【図表1】 サブルーチン利用のインタフェース

```

        :
01360 01 LINK-AREA.
01370 02 LINK-KBN          PIC S9(9) COMP.
01380 02 LINK-LGT          PIC S9(9) COMP.
01390 02 LINK-ERR          PIC S9(9) COMP.
01400 02 LINK-DATA.
01410 03 LINK-D            PIC X(1) OCCURS 10000.
01420 02 LINK-DATAB REDEFINES LINK-DATA.
01430 03 LINK-DAT          PIC X(80) OCCURS 125.
        :
    
```

【図表2】 雛形プログラムに組み込まれたサブルーチン

```

        :
00520 PROCEDURE DIVISION.
00530     PERFORM SUB90 THRU SUB99.
        :
01910 *—— SUB90 - SUB99 : FILE ALLOCATION
01920 SUB90.
01930     DISPLAY SPACE.
01940     MOVE 2 TO LINK-KBN MOVE ZERO TO LINK-LGT LINK-ERR.
01950     MOVE "ALLOC REU F(111) DA(" TO LINK-DATA.
01960     MOVE " 入力ファイル名:" TO LINK-DAT (2).
01970     MOVE "SHR"          TO LINK-DAT (3).
01980     CALL "TRFACOM" USING LINK-AREA.
01990 SUB92.
02000     DISPLAY SPACE.
02010     MOVE 2 TO LINK-KBN MOVE ZERO TO LINK-LGT LINK-ERR.
02020     MOVE "ALLOC REU F(000) DA(" TO LINK-DATA.
02030     MOVE " 出力ファイル名:" TO LINK-DAT (2).
02040     CALL "TRFACOM" USING LINK-AREA.
02050 SUB99.
02060     EXIT.
        :
    
```

4. 各種プログラムなどの自動作成システム (DENTRY)

4.1 雛形プログラムの自動作成

ファイル処理に関する13種類の標準的なプログラムを自動作成する。作成されるプログラムは次のような特徴を持つ。

- ・ファイル毎に作成される。
- ・処理を追加する部分が明確であり、後述のフローチャートに従って手続き部分を追加してプログラムを完成させる。

4.2 データエントリプログラムの自動作成

このシステムを部品としてみたときに次のような特徴を持つ。

- ・自動生成されるプログラムの手続き部は常に同じである。
- ・データ入力の場合はすべて自動生成される COBOL プログラムの DATA DIVISION に作り出される。
- ・【図表3】のような処理を追加するための部分がある。

4.3 情報検索プログラムの自動作成

順編成ファイルに対して一般的な検索 (富士通製の FAIRS-I の SEARCH, AND, OR, NOT 相当の検索) が行える。また、この検索は当然データエントリ中にもできるようになっている。

4.4 レコード設計レイアウト自動印刷

レコード設計書を自動的に印刷する。

5. フローチャート

現場担当者に対する教育の段階で標準的なパターン処理に対するフローチャート作成の訓練を徹底して行う。

雛形プログラムを部品とし、フローチャート (例: 【図表4】) に従って処理部分を追加すればよい。特に帳票出力プログラムの多くがこのような流れとなるので、初心者でも容易にプログラムを作成できる。

6. 汎用ソフトの利用

6.1 エディタによるデータエントリ

東京家政学院筑波短期大学では、開学当時より新人事務職員と学生とでOA化を行っているが、現在では、データエントリに関しては入力画面の作成及びデータ入力をエディタ利用に統一している。これについては第43回全国大会^(*)2)で紹介したがこの方法はプログラムの作成が殆ど不要なので現場の担当者には大変好評である。

7. おわりに

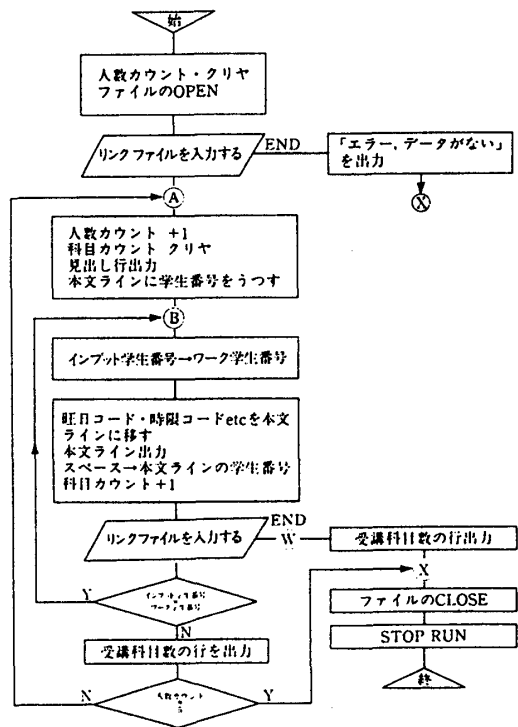
部品を用意し、それを使いこなすには訓練が必須である。これは今回説明している部品の話に限らず、システムを開発するときのソフトウェア設計全体に言えることである。既存の考え方にとらわれることなく、これらの方法論、部品、ツールを使いこなす訓練を受けることが前提になる。この後の発表、「ソフトウェアの設計 (その6)」はこうした訓練を受け、部品を最大限に使ってOA化を行った事例の報告である。

【図表3】利用者プログラムの追加

```

K99.
PERFORM HAZIME-10 THRU HAZIME-19.
DISPLAY SPACE.
DISPLAY
"*****".
DISPLAY " = DATA-ENTRY-END DATE("
TR-HIZUKE ") TIME(" TR-ZIKAN ") =".
DISPLAY
"*****".
GO TO HAZIME-10.
*****
USER1-RTN : USER-SYORI (1) O USER1-11 <=> USER1-90 NI KIZYU =
: USER-INPUT-FILE (O-FILE) NO YOMITOBASI O SURU =
: INPUT-RECORD WA O-REC NI SET SARETE IRU =
: YOMITOBASI-RECORD WA TR-SW NI 1 O SET SURU =
: (EXP) IF O-GAKUBU = 4 MOVE 1 TO TR-SW. =
*****
USER1-RTN.
IF DEBUG-SW NOT = ZERO DISPLAY
" =DEBUG.MAIN.USER1-RTN(START)".
USER1-11.
GO TO USER1-90.
USER1-90.
IF DEBUG-SW NOT = ZERO DISPLAY
" =DEBUG.MAIN.USER1-90(END)".
USER1-END.
EXIT.
    
```

【図表4】フローチャートの例



参考文献

*1 今井他: データエントリプログラムの自動作成について
 情報処理学会 第34回全国大会 2T-8 (1987)
 *2 今井他: ソフトウェア設計の基本 (その1)
 - データエントリ -
 情報処理学会 第43回全国大会 3N-1 (1991)
 *3 今井他: ソフトウェア設計の基本 (その2)
 - 現場担当者主導のシステム開発 -
 情報処理学会 第44回全国大会 4J-1 (1992)
 *4 平木他: ソフトウェア設計の基本 (その3)
 - 学生が行った図書館業務のOA化 -
 情報処理学会 第44回全国大会 4J-2 (1992)
 *5 今井他: ソフトウェア設計の基本 (その4)
 - ファイル設計 -
 情報処理学会 第45回全国大会 6S-4 (1992)