CHAPTER 3

THE SYSTEM DESIGNING

The tool is a cobol program written for generating the user outline flowchart and cobol program detail flowchart, it composes 4 consequent main sections:

3.1 The addition requirement handling section.

3.2 The preparing section.

3.3 The formatting section.

3.4 The reporting section.

3.1 The addition requirement handling section

The 4 procedures are established in this section for handling each of additional requirements, with no addition requirement included in flowchart generating, this section is ignored.

3.1.1 Extended verbs pool created

When the user submit verbs extended from a fixed set of cobol verbs defined in the program, these submitted verbs are read and then the extended verbs pool is created. By this procedure, the verbs set is expanded and it makes that not only cobol statements but also any statements in the source input text can be generated as the flow symbols. For example:

The user submitted verbs are:-

ADD-VERB LIST

REPLACE REMOVE SEND

ADD-VERB END

Then these statements below can be generated

REPLACE OLD RECORD WITH NEW RECORD

REMOVE THIS RECORD FROM THE MASTER FILE

SEND THIS THESIS TO THE BOARD FOR ACCEPTANCE

IF THE ACCOUNT NUMBER MATCHING THEM REPLACE OLD AMOUNT BY NEW AMOUNT

3.1.2 Unaffected flow verbs pool created

On several cobol programs, there are some specific verbs that not related with the program logic flow. Several cobol compilers provide them for compiler directing actions and program debugging aids. These verbs must not appear on the flowchart listing, then the user must tell the flowchart generator program what verbs to be deleted out. When the user submitted them, the unaffected flow verbs pool is created and used as referable verbs in the statement deletion in procedure 3.3.2) For example :

The submitted verbs are :

DLETE-VERB LIST

TRACE CODE

DLETE-VERB END

Then the TRACE and CODE statements do not appear on the flowchart report.

By default, the EJECT, SKIP1, SKIP2, SKIP3, DEBUG, READY and RESET statements are automatically deleted.

3.1.3 New more meaning dataname table created

When the user requires to change old dataname with new representative name, the pairs of old and new dataname are entered. These names are read into the computer memory in order to form the table, each entry in the table uses old dataname as referable key the new name is

retrieved by the dataname replacing process in procedure 3.3.2). For example:

The pairs of names input are:

```
MATCH-NAME LIST
I LOOP-CONTROL-COUNTER
NO STUDENT-NUMBER
KEY STUDENT-TRANSACTION-KEY
MATCH-NAME END
```

The name table is built up, the I, NO and KEY are keys of table. When the flowchart report is printed, all of datanames I, NO and KEY become to the names LOOP-CONTROL-COUNTER, STUDENT-NUMBER, STUDENT-TRANSACTION-KEY respectively.

3.1.4 Document title input stored

When the title heading of flowchart report is required, the title input must be submitted, like the example:

```
COMMENT LIST
SKIP3   THIS IS THE FLOWCHART GENERATOR TOOL
SKIP2   DEVELOPED BY  :  SATIRA
        ITS PURPOSE   :  PROGRAM DOCUMENTATION HELPING
FOOT    PROPERTY      :  CHULALONGKORN UNIVERSITY
COMMENT END
```

The title input is pick up and then saved in computer memory for future printing, the field that preceed the title text like SKIP3, is used for directing the line position which the title text to be printed.

3.2 THE PREPARING SECTION:

This section is provided for helping the logical structure of the formatting section easy and simple to be designed, 2 procedures below are advocated:

3.2.1 Pre-formatted input file created

In cobol program, we know that the necessary steps for solving the desire task, that is, the logical flow are defined only in the PROCEDURE DIVISION part, then other division and any control cards are undesirable, the text in the PROCEDURE DIVISION are selected to form a temporary pre-formatted input file. While the pre-formatted file is formed, the comment statements in PROCEDURE DIVISION are not written into the file and all of one more sentences per line in PROCEDURE part are separated into many single records before it is written. With this preparation, it makes the input text easy to be pick up for flowchart generating and the complication of formatting section is reduced.

```
DATA DIVISION.
      :
      .
PROCEDURE DIVISION.
      :
      .
A-PROCEDURE. OPEN INPUT TAPE-FILE.
    READ TAPE-FILE AT END STOP RUN.
    MOVE INCOME TO WORK-TEMP. ADD 1 TO RECORD-COUNTER.
      :
      .
end-of-program
any control cards
```

The pre-formatted file is created as:

```
PROCEDURE DIVISION.
      :
      .
A-PROCEDURE.
    OPEN INPUT TAPE-FILE.
    READ TAPE-FILE AT END STOP RUN.
    MOVE INCOME TO WORK-TEMP.
    ADD 1 TO RECORD-COUNTER.
      :
      .
END-OF-FLOWCHART.
```

3.2.2 Procedure description preparing

During the time that the pre-formatted file is created, when the input text in PROCEDURE DIVISION is paragraph or section, it is moved into computer memory to form a procedure description table, each procedure has its sequence number and type, the type code uses to indicate that it's paragraph or section. Each entry in the procedure description table is referred by the procedure name in GO TO, PERFORM and ALTER statements process in formatting section, when the procedure names matching occur, the procedure number and type is pick up and then assigned to it, if no procedures in the table match against it, the undefined number and type (in abbvr. UNDF) is given.

3.3 THE FORMATTING SECTION

We know, the format of cobol source program and outline input is variable, a space or more than one space is used as a word separator, an statement is ended when new verb found and a sentence is terminated by a period, a sentence or a statement can occupy more than one line and the hyphen in continuous area is used as continuation word or literal.

Because of its variable format, then it must be compressed and rearranged before the flowchart printing. The more than one space word separator is compressed into a single space and each statement is rearranged into a continous string. After this process, the flowchart symbol code type is given to the statement and when some conditional or unconditional branching verbs are detected, the branch flow direction code is inserted for branch line printing control. The formatted statements are written into formatted file, the file is used as input for printing in the reporting section. We can classify this section into 3 main functions:

3.3.1 The sentence string format

The source texts from pre-formatted file are pick up consequently and then are moved into a string work area, when hyphen found in any continous area, the word or literal must be concatenated according with concatenation rule. After completion of sentence string formed, the string will be decomposed into one or more statements.

3.3.2 The sentence string decomposed

To decompose the sentence in string into one or more statements, each statement has its difference decomposition process, it depends on each statement syntax rule. While the string is decomposed, if new more meaning name option is set, old datanames are replaced with new names in table created by procedure 3.1.3).

On the beginning, the texts in sentence string are forward scanned word by word, to justify that the word is verb or not, the word must be compared with the verbs set defined in the flowchart generator program and when new verbs added option is set, it must be compared with verb added pool too. When the word is verb, the decomposition process should take place.

Some decomposition processes are complex such as the "IF" decomposition, the reader can see the description of this process in appendix I some decomposition process like "GO TO", "ALTER" and "PERFORM" statements decomposition must pick up the procedure number and type from the procedure table created by procedure 3.2.1) and used it as the procedure reference. After the decomposition, the decomposed statement is compressed and rearrange into a continous text string with only space as word separator.

3.3.3 The formatted statement file created

The flowchart box code is assigned to each decomposed statement, the code is used for telling the printer to plot the shape of each flow symbol. Especially for conditional statements like "IF" and "AT END", the branch flow direction code is given for line direction printing. The temporary file that composed with these records is created on an auxiliary storage device and after every decomposed statements are already written, the file is opened as input for flowchart printing by the next reporting section.

The example below may be shown to clarify the function of this section:

```
the texts in pre-formatted file;
PROCEDURE DIVISION.
...
    ooooo
    ••••••
    DISPLAY IDENTCARD " THIS ID-NUMBER IS
-   "NOT IN STUDENT MASTER FILE " GO TO
    NEW-GET.
...
    ......
    .......
    READ STUDENT-INPUT-CARD AT END,
    GO TO FINISH.
    ......
    .......

END-OF-FLOWCHART
```

By procedure 3.3.1), the continuous string of sentence is created in memory;

DISPLAY IDENTCARD "THIS ID-NUMBER IS NOT IN STUDENT MASTER FILE" GO TO NEW-GET.

By procedure 3.3.2), the string is decomposed into 2 statements, the DISPLAY and GO TO statements;

DISPLAY IDENTCARD " THIS ID-NUMBER IS NOT IN THE STUDENT MASTER FILE "

and GO TO NEW-GET (P004)

the box codes are given and the records are written into the formatted file;

ßBDISPLAY IDENTCARD " THIS ID-NUMBER IS NOT IN THE STUDENT MASTER FILE"

and ßH(P004)

box code ↑

procedure no. ↑

After every decomposed statement are written, the formatted file contains;

.....

.....

ßBDISPLAY IDENTCARD "THIS ID-NUMBER IS NOT IN THE STUDENT MASTER FILE"

ßH(P004)

.....

box code .....

ßBREAD STUDENT-INPUT-CARD

F

branch line

code ßI

ßH(P010)

.....

.....

.....

3.4 THE REPORTING SECTION

3.4.1 The title printing

The title submitted by the user are printed at the top of the flow-charting report, each of title lines position are directed by the line control field that precede the title texts.

3.4.2 The diagrammatic flowchart printing

To print out the diagrammatic flowchart and its appropriate texts in the flowchart boxes, each of formatting records are read and the flowchart symbol code in the record is used to determine what shape of the flowchart boxes to be drawn and the texts in the record are printed within the box. The appropriate texts in every boxes are central justified, the size of each boxes is specified by the length of its appropriate texts. The arrow symbol is used to indicate the logical flowing movement of these texts.

3.4.3 The replaced and replacing dataname lists

The old and new meaning datanames are listed after the flowchart printing, its purpose is for the name reference with the old names in cobol source program.

3.4.4 The procedure name lists

The procedure name, number and the page number that the procedure appeared are listed. With these procedure lists, it makes the required procedure can be found quickly.