COBOL構造的プログラミングシステムにおけるドキュメンテーションとその自動生成ツール

日本電気株式会社　東　基衛　中嶋　淳
情処営支本・応用プログラム部　三野村圭右　高橋雅昭

はじめに

優れた思想も人々が理解できなければ意味のない妄想と同じである。又，最新型の製品もその使用法がわからなければ置物ぐらいの意味しかない。一般的により高度な思想や製品になればなるほど，それを説明すること，すなわち文書化が重要になる。形のないソフトウェアにとってはまさにドキュメントが完備していなければ，使いものにならないといえる。

本文では，プログラムの開発／保守におけるドキュメンテーション方法について概観した後，本文で提唱するプログラムとドキュメントの一元化の要求を満たすのに適した，COBOL言語をベースにした構造的プログラミング用ツールCOBOL/Sの紹介と，そのドキュメンテーションを支援機能について述べる。

1．プログラム開発／保守におけるドキュメンテーション

1.1　プログラムドキュメントについて

本来ドキュメンテーションとはEDPシステム開発の全フェーズにまたがる作業であるが，本文ではプログラムの開発／保守におけるドキュメンテーションについて述べる。すなわちプログラム仕様書，プログラム説明書及び保守手引書といったプログラムそのものにまつわるドキュメントに限定して話をすすめよう。

プログラムドキュメントの必要性として次のような事が考えられる。

①開発ステップ間のコミュニケーションの手段。
②開発中における要員間のコミュニケーションの手段
③プログラム製作者と利用者のコミュニケーションの手段
④プログラム製作者と保守者のコミュニケーションの手段

これらの必要性に対して，以前から行われてきた方法は，文章によるプログラムの説明と，フローチャートによるプログラムの流れの表現であった。ところが，これら従来のドキュメンテーションの方法には次のような欠点が明らかになってきた。

①文章表現のドキュメントのあいまいさ及び冗長性
②本来プログラミングそのものの目的に必要でないドキュメンテーションに時間を割く不快さ
③フローチャートとプログラミング言語の間のギャップ
④ソースプログラム修正による，ドキュメントとソースプログラムの不整合性。

これらの解決策としてドキュメントの標準化という方法があり，例えばHIPOや，日本電気(株)のSTEPS[1]などがある。又，ソースプログラムからフローチャートをつくり出すフローチャーダや，フローチャートの代替となり，プログラムを構造的に設計するのに便利な技法として構造化チャート，SPD[2]などがある。

1.2 擬似言語によるソースプログラムとドキュメントの一元化

1.1で述べた従来のプログラムドキュメントの欠点をカバーする1つの方法として擬似言語によるドキュメンテーションがある．擬似言語とはプログラムの制御構造をいくつかのキーワードを用いてプログラムの流れに従って記述するものである．うまく選択されたキーワードを用いることにより，できあがるプログラムは構造的になることが期待される．キーワードの間の部分の記述は自由であり，内容の粗さによって概要フローにも，詳細フローにもなる．ところでその擬似言語に多少の言語仕様を持たせて機械による翻訳が可能なプログラミング言語にすることを考える．プログラム構造を明確にするという意味を持ったプログラミング言語はPL/1やPASCALをはじめ最近はたくさん開発されているが，新たな言語をどんどん開発することはプログラムのポータビリティ，過去開発されたプログラムとの関係，利用者の学習を考えると得策でない．そこで広く使用されている言語をベースとした擬似言語の仕様を開発し，コンパイラの前段に位置するプリプロセッサによって擬似言語をコンパイラ言語に変換するという方法が好ましい．

さらに擬似言語を用いてプログラムの処理を記述することがそのままオブジェクトプログラムにつながっているから，ドキュメントは常にプログラムの機能の最新の情報を提供することになる．又，擬似言語を変換するプリプロセッサが入力された情報を編集して見やすいドキュメントの形でソースプログラムリストを出力すれば，ソースプログラムリストのみで詳細部分を示すプログラムドキュメントとなり得るのである．

2. COBOL/Sとは

2.1 COBOL/Sシステム概要

COBOL/Sとは日本電気(株)が開発し，既に3年余り利用者に提供しているCOBOL言語をベースにした構造的プログラム作成システムである．

本システムは構造的プログラムの作成及び，ソースプログラムと一元化したドキュメントの作成などの目標に対する1つのアプローチである．COBOL/SではCOBOL/S言語と呼ばれる擬似言語を用いてプログラムを設計し，それをCOBOLコンパイラの前段に位置する処理系(COBOL/Sプリプロセッサ)によってCOBOL言語に変換する．そのときCOBOL/Sソース編集リストというドキュメントを得ることができる．さらにCOBOL/S言語で記述されたソースプログラムを解析し，保守に有用なドキュメントを自動生成する処理系(COBOL/Sド

文名称	キーワード
見出し部定義文	ID:
環境部定義文	ED:
データ部定義文	DD:
手続き部定義文	PD:
代入文	:=
副プログラム呼び出し文	ECALL:
手続き定義文	PROC:
手続き開始文	ENTRY:
手続き終了文	EXIT:
副手続き呼び出し文	CALL:
手続き脱出文	RETURN:
2分岐選択文	IF: ELSE: END;
多分岐選択文	CASE: OF: OTHER: END:

図2-2 COBOL/Sキーワード一覧

キュメンテーションエイド）を持っている．COBOL/Sの位置付けを図2-1に，キーワードを図2-2に示す．

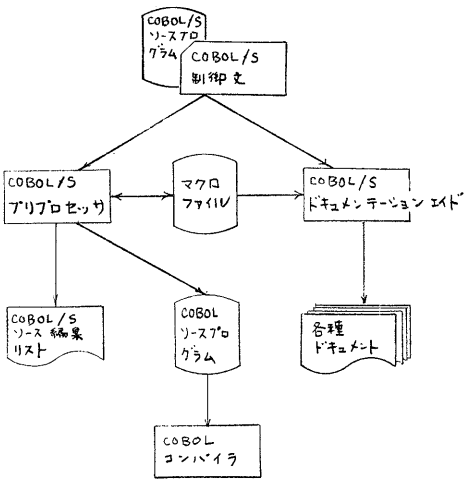

図 2-1　COBOL/Sの位置付け

2.2　COBOL/Sの主な機能

COBOL/S擬似言語及びその処理系には主な機能として次のようなものがある．

① COBOL/S言語はCOBOL/Sのキーワードと COBOL言語文を混在できる．
② コーディングの手間を省く，簡略化機能がある．
③ プログラム構造の明確化の為，手続きやブロックを簡潔に表現できる．
④ 錯綜した条件と動作の組み合わせを簡潔に表現する，デシジョンテーブルが記述できる．
⑤ COBOL/Sソース編集リスト上に注釈記述をしたり，改行，改ページを指示できる．
⑥ すべての文に，並列に出力される注釈（インラインコメント）を記述できる．
⑦ パラメータをもつマクロ機能がある．
⑧ ソースリストは自動段落付けされる．
⑨ COBOL/Sドキュメンテーションエイドにより，ドキュメントが自動生成される．

文名称	キーワード
繰り返し文（1）	LOOP: END:
繰り返し文（2）	WHILE: END:
繰り返し文（3）	ITERATE: UNTILE:
繰り返し文（4）	ITERATE; ～ UNTILE:
単純ブロック構成文	BEGIN: END:
ブロック脱出文	LEAVE:
ローカルデータ文	LD:
デシジョンテーブル用言語文	TABLE: COND: ACT; END:
注釈記述文	NOTE: NEND:
改行・改ページ文	SPACE:
マクロ呼び出し文	:マクロ名

図 2-2　COBOL/Sキーワード一覧（つづき）

```
PROC:  売り上げ表の作成
  ENTRY:
  BEGIN:  開始処理
    | 変数初期化，ファイルオープン，見出し
  END:
  BEGIN:  主処理
    LOOP:
      | 1レコード入力
      | EOFなら合計印字して終了処理へ
      | IF:  コード変わったか．
      |   | 一番最初でなければ合計印字
      | ELSE:
      |   | 計算
      |   | 1レコード出力
      | END:
    END:
  END:
  EXIT:
```

図 2-3　COBOL/Sキーワードを用いた詳細設計の例

2.3 COBOL/Sの利用の仕方

COBOL/Sはコンパイラの前段に位置するプリプロセッサなのでプログラム製造ツールとして利用する他，擬似言語を用いてのプログラム設計ツール，ドキュメンテーションツールとして利用できる．プログラムをトップダウン設計で設計する際，キーワードと文章表現で抽象化されたプログラムの機能を記述でき，段階的に詳細化できる．COBOL/Sはプログラムのコンパイルと切り離されているので，それらを各種のドキュメンテーション支援機能を用いて設計ドキュメントとして出力させることができる．図2-3参照．

3. COBOL/Sにおけるドキュメンテーション支援機能

3.1 ソース編集リスト上のドキュメント機能

①インラインコメント

COBOL言語文，COBOL/S言語文を問わず，どのような文に対してでも，その文に対する注釈をその文と同一行に記述できる．これをインラインコメントと呼ぶ．注釈は，文の終わりに空白を置いてセミコロン“;”を書けばうしろに記述できる．

インラインコメントがソース編集リスト上に出力された例と，マクロ登録リスト上に出力された例をそれぞれ図3-1及び図3-2に示す．

この機能により行単位にコメントを手軽に記述することができ，132字のリストを有効に用いることができる．

②注釈記述文

プログラム全体に対する注釈または手続きに対する注釈を記述するには，NOTE:文とNEND:文の間に自由に注釈を記述すればよい．そうすればソース編集リスト上に注釈が星印“*”で囲まれて出力される．このとき星印のboxの大きさや，印字位置を指定できる．図3-1及び図3-3を参照．

③ブロック構造の視覚化

手続き部内に記述された命令は，利用者の記述開始欄に関係なく自動的に段落付けされる．従って，利用者は命令をB領域以降に記述すれば，その開始欄を意識しなくても，ブロックごとに段落付けされたソース編集リストを得ることができる．

又，ソース編集リスト上ではブロックの開始を示すキーワードと，ブロックの終了を示すキーワードの間には段落記号として“|”が印字される．図3-1参照．

④編集リストの改行・改ページ指示

改行・改ページ文（SPACE:）により必要な行数の改行や，改ページを指示できる．一般にソース編集リストをプログラムドキュメントとする場合，手続きはリスト1ページ以内におさめた方が，プログラムのモジュール化にも役立つし，ドキュメントとしても見やすくなる．そのような場合，適当に改行や改ページをすることが必要になってくる．図3-1は手続きの1ページコーディングの例である．

図 3-1
COBOL/S
プログラム
編集リスト
の例

```
ACOS-4   COBOL/SDT   V/R 03.72   ACCOUNTS  PAYABLE                    AP503                    DATE: 79-06-21  PAGE:    5

STNO  ....*....1....*....2....*....3....*....4....*....5....*....6....*....7..      ----.----1----.----2----.----3----.----4 ....*.

      ************************************************NOTE:*************************************************
      *                                                                                                      *
      *                          2.3. MTC=CHK          :   MATCH=CHECKS                                      *
      *                                                                                                      *
      ************************************************NEND:*************************************************

  5         PROC:   MTC=CHK                                          ;                FUNCTION
  6             ENTRY:                                               ;                --------
  7             ITERATE:                                             ;     OBTAIN A MASTER RECORD.    COMPARE
  8              | M=MST=RCD := A=CHK                                ; CHECK NUMBER WITH CURRENT INPUT.
  9              | !GET ( A=OLD=DSK,"Y",W=MST=EOF )                 ;
 10              | CASE:                                            ;     IF EQUAL DETAIL  CHECK,  GET A DE-
 11              | OF:  M=CHK=NMB = L=CHK=NMB                       ; TAILED MATCH.  IF DETAIL FAILS,  UPDATE
 12              |  | W=MTC=FLG := "Y"                               ; NEW MASTER INTERVENING RECORD AND EXIT.
 13              |  | CALL:   DTL=MTC                                ;
 14              |  | IF:  W=MTC=FLG = "N"                           ;     IF DETAIL CHECK SUCCEEDS  UPDATE,
 15              |  |  | CALL:   EDT=ECT                             ; INDICATE VOID UPDATE WITH FLAG SINCE NO
 16              |  |  | CALL:   UDT=PRC                             ; DUPLICATE INPUT CHECK  NUMBER WILL THIS
 17              |  | ELSE:                                          ; PROCEDURE  THE MASTER  RECORDS  INVOICE
 18              |  |  | W=UDT=FLG := "NN"                           ; NUMBER WILL BE PASSED.
 19              |  |  | CALL:   UDT=PRC                             ;
 20              |  |  | W=UDT=FLG := "NN"                           ;     IF MASTER  LESS THAN INPUT,  UPDATE
 21              |  | END:                                           ; NEW MASTER WITH INTERVENING  RECORD AND
 22              |  * LEAVE: 2                                       ; RECYCLE IF INPUT FILE NOT EOF.
 24              | OF:  M=CHK=NMB < L=CHK=NMB                        ;
 25              |  | CALL:  UDT=PRC                                 ;     IF MASTER  GREATER  THAN INPUT RE-
 27              | OF:  M=CHK=NMB > L=CHK=NMB                        ; ERROR AND ABORT RUN.
 28              |  | CALL:  BAD=FIL                                 ;
 29              |  * LEAVE: 2                                       ;
 31              | END:                                             ;
 33             UNTIL:  W=MST=EOF = "Y"                             ;
                EXIT:                                                ;
```

図 3-2
マクロ登録
リストの例

```
          /MACR   MAST-FILE
          FD      #1-#2-DSK
  1                       BLOCK CONTAINS 5 RECORDS
  2                       RECORD CONTAINS 80 CHARACTERS
  3                       LABEL RECORD IS STANDARD
  4                       DATA RECORD IS A-CHK.
  5
  6             SPACE: 1
  7          01      #1-CHK.
  8
  9             SPACE: 1
 10           05     #1-VDR-IDT   PIC X(06) .
 11             SPACE: 1
 12           05     #1-VDR-NAM   PIC X(30) .
 13
 14             SPACE: 1
 15           05     #1-CHK-NMB   PIC X(08) .
 16
 17
 18             SPACE: 1
 19           05     #1-ISU-DTE   PIC X(06) .
 20             SPACE: 1
 21           05     #1-IVC-NMB   PIC X(10) .
 22
 23             SPACE: 1
 24           05     #1-AMT-PAD   PIC X(06) .
 25             SPACE: 1
 26           05     #1-VID-FLG   PIC X(01) .
 27
 28
 29
 30           05     FILLER   PIC X(13) .
```

```
|  FILE DEF. OF  MAST FILE.
|    #1 <- ID.
|    #2 <- OLD/NEW
|
|
| INPUT RECORD OLD MASTER FILE.   FILE IS
|    PRE-EDITED.
|
| VENDOR NUMERIC CODE.
|
| VENDOR NAME FROM VENDOR
|    REGISTER.
|
| SEQUENTIAL  CHECK  NUMBER.
|    FILE HAS NO GAPS, MAJOR
|    SORT FIELD.
|
| DATE CHECK WRITTEN : MMDDYY.
|
| IN SEQUENCE WITHIN CHECK
|    NUMBER.  MINOR SORT FIELD.
|
| DOLLAR VALUE OF CHECK.
|
| UPDATE FIELD FOR THIS PROGRAM,
|    CONTAINS CODE FOR STOP
|    PAYMENT OR CODE FOR HAND
|    PAYMENT.
|
```

図 3-3
NOTE: 文
による プ
ログラム
概要の記
述例

```
ACOS-4   COBOL/SDT   V/R 03.72   ACCOUNTS  PAYABLE                    AP503                    DATE: 79-06-21  PAGE:    1
                                                                  INPUT FILE : SYSIN           REV.: 000

      ************************************************************************
      *                                                                      *
      *  AP503                          : ACCOUNTS PAYABLE                   *
      *                                                                      *
      ************************************************************************

STNO  ....*....1....*....2....*....3....*....4....*....5....*....6....*....7..      ----.----1----.----2----.----3----.----4 ....*.

  1M1       ID: AP503

  3M1       ***********
  4M1       * FUNCTION *
  5M1       ***********
```

```
      ************************************NOTE:************************************
      *                                                                          *
      *                                                                          *
      * INPUT   : A=OLD=DSK   DISK SEQUENTIAL.                                   *
      *                       MAJOR SORT CHECK NUMBER.                           *
      *                       MINOR SORT INVOICE NUMBER.                         *
      *                                                                          *
      *           B=CRD=IN    SORTED CARD FILE.                                  *
      *                       ON SEQUENTIAL DISK.                                *
      *                       MAJOR SORT CHECK NUMBER.                           *
      *                                                                          *
      *                                                                          *
      * OUTPUT  : C=NEW=DSK   DISK SEQUENTIAL.                                   *
      *                                                                          *
      *           D=EDT=LST   PRINTER.                                           *
      *                       LISTING OF UPDATED DISK RECORDS.                   *
      *                       ERROR LIST, PRINTER (CONDITIONAL).                 *
      *                       CONDITION:  ERROR  IMAGES AND THEIR  FAILURE       *
      *                            CODES  ARE  ENTERED IN  ERROR-TABLE.          *
      *                            IF ERROR-TABLE FILLS IT IS FLUSHED ONTO       *
      *                            A FILE  CALLED  PRINT  IMAGE FOR  LATER       *
      *                            PRINT.                                        *
      *                                                                          *
      *                                                                          *
      *                                                                          *
      * PROCESS : ANY  CHECKS  ISSUED IN  ERROR  MUST BE  FLAGGED  BEFORE       *
      *           CHECKS ISSUED FILE IS PRINTED.   FLAGS ARE "S" FOR STOP       *
      *           PAYMENT  AND "H" FOR HAND  VOID.   INPUT  CARD-IMAGE IS       *
      *           EDITED TO INSURE CORRECT/COMPLETE DATA IS PRESENT.             *
      *                                                                          *
      *                                                                          *
      ************************************NEND:************************************
```

3.2 マクロ機能

COBOL/Sのマクロ機能はCOBOLの登録集機能に似たものであるが，次のような特長をもっている．

① マクロ定義文（/MACR）以降につづくデータをCOBOL/Sがマクロファイルに登録する．このときマクロ登録リストを得ることができる．

② ソースプログラム中にマクロ呼び出し文（：マクロ名）を記述することにより，COBOL/Sはマクロファイルからマクロデータを読み出して展開する．

③ マクロ参照時にマクロデータ中の仮引数に実引数を対応させるパラメータ機能をもち，必要に応じてマクロデータ中のある行を展開したり，しなかったりすることができる．

④ マクロ内で他のマクロを呼び出せる．このようなネストは3レベルまで許される．

このようなマクロ機能を用いてソース編集リストのドキュメンテーションを助ける使い方に次のようなものがある．

① 部や節の宣言の部分や手続きのソース編集リストの標準的なレイアウトの仕方を決めて，ドキュメントを手軽に標準化できる．図3-4参照．

② 一連の命令をマクロ化して擬似命令をつくることにより，標準的な処理を記述でき，リスト量を減らすこともできるし，処理の理解を早めるのに役立つ．図3-5参照

```
1       /MACR   A1      N
            ADD 1 TO #1.

        /MACR   GET     N
1           READ #1 AT END MOVE #2 TO #3.

        /MACR   PUT     N
1           WRITE #1 AT END MOVE #2 TO #3 .
```

図3-5 マクロによる擬似命令の作成例

```
        /MACR   ID  Y
1       ID: #1
2           SPACE: 2
3       ************
4       * FUNCTION *
5       ************
6           SPACE: 2

        /MACR   DD  Y
1       DD: .
2           SPACE: 2
3       ******************
4       * DATA DEFINITION *
5       ******************
6           SPACE: 2

        /MACR   PD  Y
1       @1PD:#1
2       %1PD:
3           SPACE: 2
4       ************************
5       * PROCEDURE DEFINITION *
6       ************************
7           SPACE: 2

        /MACR   FS
1       FILE SECTION.
2           SPACE: 2

        /MACR   WS
1       WORKING-STORAGE SECTION.
2           SPACE: 2
```

図3-4 レイアウト標準化のためのマクロ例

3.3 デシジョンテーブル記述機能

デシジョンテーブルは，複雑に絡み合う複数の条件と動作があるときに，どういう状態のときにどのような処理を行うべきかを簡潔な表形式で表現したものであり，問題点を整理したり，手続きを表現したりする手段として，これまでも利用されていた．しかしそれらはあくまでも処理の表現であって，実際にプログラミングするときは手作業で変換するのが一般的であった．

COBOL/Sのデシジョンテーブル記述機能は，利用者がデシジョンテーブル用言語文を用いてテーブルを表現すれば，自動的にCOBOL言語文に展開する機能である．

COBOL/Sでは制限型，拡張型およびその混在した混合型のデシジョンテーブルを記述でき，それをいろいろな形に編集して出力できる．

この機能を用いることによりプログラムのドキュメンテーションについて次のような効果を期待できる．

①誤りのないプログラム仕様を記述しやすい．
②プログラム仕様書の標準化が促進される．
③プログラムの知識の余りない人々も仕様を理解しやすい．
④プログラムの修正による他の部分への影響が明確に把握できる．

図3-6に実際にプログラム中で用いられているデシジョンテーブルの例を示す．

```
J := 0
LIST-R := SPACE
WHILE: J < 20
 | J := J + 1
 | TABLE: CONV-TBL
 |  | COND:
 |  | W (1, J) = "    a"  :Y,N,N,N,N,N,N,N,N,N:
 |  | W (1, J) = "****a"  :N,Y,Y,Y,Y,Y,Y,Y,Y,Y:
 |  | W (1, J) = "*   a"  :N,Y,N,N,N,N,N,N,N,N:
 |  | W (2, J) = "    "   :Y,Y,Y,Y,N,N,N,N,N,N:
 |  | W (2, J) = "*   a"  :N,N,N,N,Y,Y,Y,Y,Y,Y:
 |  | W (2, J) = "*   "   :N,N,N,N,Y,Y,Y,Y,N,N:
 |  | W (4, J) = "    a"  :Y,N,N,N,N,N,N,N,N,N:
 |  | W (4, J) = "****a"  :N,Y,Y,Y,Y,Y,Y,Y,Y,Y:
 |  | W (4, J) = "*   a"  :N,Y,N,N,N,N,N,N,N,N:
 |  | W (5, J) = "    a"  :Y,N,Y,Y,Y,Y,N,N,Y,N:
 |  | W (5, J) = "*   a"  :N,Y,N,N,N,N,N,N,N,N:
 |  | W (5, J) = "*   "   :N,N,N,N,N,N,Y,Y,N,Y:
 |  | W (7, J) = "    a"  :Y,N,N,N,Y,Y,N,Y,N,N:
 |  | W (7, J) = "****a"  :N,Y,Y,Y,N,Y,Y,N,Y,Y:
 |  | ACT:
 |  | CHAR (J) :=         :1,2,3,4,5,6,7,8,9,0:
 | END:
END:
WRITE LIST-R AFTER 2
```

図3-6 デシジョンテーブルの例

3.4 ドキュメンテーションエイド

COBOL/S言語で記述され，COBOL/Sプリプロセッサによって出力されたソース編集リストは，そのドキュメント支援機能によってプログラムの論理を走査する場合に有効である．しかしながら，このソース編集リストだけでは各手続き間の相関情報，手続きとマクロ間の相関情報などについての収集，整理は手作業で行わなければならない．

COBOL/Sドキュメンテーションエイドは，COBOL/S言語で記述されたソースプログラムからプログラムの理解，保守に有用な情報を，各プログラム単位に，又は関連するプログラム全体に渡って収集し，ドキュメントとして編集し出力するためのツールである．

次にCOBOL/Sドキュメンテーションエイドが指定によって出力する各種ドキュメントの説明とその例を示す．

①手続き概要一覧表

ソースプログラム上の各手続き内に，注釈記述文で記述されているプログラムあるいは手続きの概要を一覧表として編集し，出力する．いわゆるプログラム一覧とか，手続き一覧に相当する．図3-7参照．

図3-7 手続き概要一覧表

```
ACOS-4          COBOL/SDA          V/R 01.5    **********  COBOL/SDA-4  **********    DATE: 79-06-22  PAGE:  6
*****************************************************************************
*  CDMAIN                    : SDA COMMAND ANALYZER                           *
*****************************************************************************

****************
* FUNCTION NOTE *
****************
```

PROG./PROC. NAME	FUNCTION
CDMAIN	ﾚﾍﾞﾙ: 1. ﾒｲﾝ : COBOL/SDA ｾｲｷﾞｮｶﾝ ﾉ ｶｲｾﾂ ﾓｼｺﾐ
CDMAIN	ﾚﾍﾞﾙ: 1-0. ﾒｲﾝ : CDMAIN ﾉ ｺﾝﾄﾛｰﾙ
PAR-SCT	ﾚﾍﾞﾙ: 1-1. ﾒｲﾝ : SDA ｾｲｷﾞｮｶﾝ ﾉ ﾊﾟﾗﾒｰﾀ ｲﾝｼｭﾂ
MAC-SCT	ﾚﾍﾞﾙ: 1-2. ﾒｲﾝ : ﾏｸﾛ ﾉ ﾄｳﾛｸ

② 参照手続き一覧表，手続き相関図

プログラムや手続きの定義／参照およびその相関関係を表形式（図3-8）あるいは図形式（図3-9）で編集し出力する．あるプログラムや手続きの修正が他のどの部分に影響するであるとか，プログラムのインタフェースを変更するときなどに有効である．

図3-8
参照手続き一覧表

```
ACOS-4          COBOL/SDA          V/R 01.5          ********** COBOL/SDA=4 **********          DATE: 79-06-22 PAGE: 16

*****************************************
*  TABLE OF REFERED PROC. & PROG.       *
*****************************************
```

PROG. NAME	PROC./PROG. NAME	
1 CDEXEC	CDEXEC	
	CDMAIN	CDPROG
	CDPSUB	
2 CDMAIN	CDMAIN	MAC-SCT
	PAR-SCT	
	CDPUTO	CSIGET
	XMACIO	
3 CSIGET	CSIGET	INP-INP
	CHK-SIF	GET-GET
	OPN-CLS-SCT	
	GETQF	CLSSQF
	OPENQF	OPNSQF
	CLSQF	

図3-9
手続き相関図

```
ACOS-4          COBOL/SDA          V/R 01.5          ********** COBOL/SDA=4 **********          DATE: 79-06-22 PAGE: 17

*****************************************
*  FIGURE OF PROC. & PROG. RELATION     *
*****************************************
```

OVERALL PROGRAMS

PROG. NAME	1 2 3 4 5	6 7 8 9 0	1 1 1 1 1 1 1 2 3 4 5	1 1 1 1 2 6 7 8 9 0	2 2 2 2 2 1 2 3 4 5	2 2 2 2 3 6 7 8 9 0	3 3 3 3 3 1 2 3 4 5	3 3 3 3 4 6 7 8 9 0
1 CDEXEC	* * *							
2 CDMAIN	*	* *						
3 CSIGET		* * *	* *					
4 CDPROG								
5 CDPSUB								
6 CDPUTO								
7 XMACIO								
8 GETQF								
9 CLSSQF								
10 OPENQF								
11 OPNSQF								
12 CLSQF								

CDMAIN

PROC./PROG. NAME	1 2 3 4 5	6 7 8 9 0	1 1 1 1 1 1 1 2 3 4 5	1 1 1 1 2 6 7 8 9 0	2 2 2 2 2 1 2 3 4 5	2 2 2 2 3 6 7 8 9 0	3 3 3 3 3 1 2 3 4 5	3 3 3 3 4 6 7 8 9 0
1 CDMAIN	* * * *							
2 PAR-SCT	*							
3 MAC-SCT		*						
4 CDPUTO								
5 CSIGET								
6 XMACIO								

CSIGET

PROC./PROG. NAME	1 2 3 4 5	6 7 8 9 0	1 1 1 1 1 1 1 2 3 4 5	1 1 1 1 2 6 7 8 9 0	2 2 2 2 2 1 2 3 4 5	2 2 2 2 3 6 7 8 9 0	3 3 3 3 3 1 2 3 4 5	3 3 3 3 4 6 7 8 9 0
1 CSIGET	* * * *							
2 INP-INP	*							
3 GET-GET		* *						
4 CHK-SIF		*	*					
5 OPN-CLS-SCT								
6 GETQF								
7 CLSSQF								
8 OPENQF								
9 OPNSQF								
10 CLSQF								

③ 参照マクロ一覧表，手続き／参照マクロ相関図

プログラムや手続きがどんなマクロを参照しているかや，その相関関係を表形式（図3-10）あるいは図形式（図3-11）で編集し出力する．マクロの修正がどのプログラムに影響があるかを知るのに有効である．

図3-10 参照マクロ一覧表

```
ACOS-4          COBOL/SDA          V/R  01.5      **********    COBOL/SDA-4    **********      DATE: 79-06-22  PAGE:  19

**********************************
*   TABLE   OF   REFERED   MACRO  *
**********************************
```

PROG./PROC. NAME	MACRO NAME	
CDEXEC	ED	SPE-NAM
	WS	A-INP-BUF
	DD-SPR-BUF	ID-IND-FLG
	PD-SDA-PAR	WD-IMP-FLG
	TD-IDN-TBL	TD-NAM-TBL
	TD-APR-TBL	D-CONST
	GD-IMP-LST	
CDEXEC	CHK-INDENT	

PROG./PROC. NAME	MACRO NAME	
CDMAIN	EDX	WS
	A-INP-TYP	DD-SPR-PAR
	W-TMP-WRK	Q-PTR-XXX
	Y-MC-FIJK	Z-DSP-X40
	LS	A-INP-BUF
	DD-SPR-BUF	ID-IND-FLG
	PD-SDA-PAR	WD-IMP-FLG
	X-CHR	
CDMAIN	SCT	OPN-SPR
	OPN-INP	GET-INP
	INSPECT	UNSTRINGX
	NAM-SPR	
PAR-SCT	SCT	FETCH-PAR
	INSPECT	+1
	UNSTRINGX	
MAC-SCT	SCT	OPN-MACLX
	INSPECT	UNSTRINGX
	OPN-MACO	PUT-MAC
	GET-INP	CLS-MAC

図3-11 手続き／参照マクロ相関図

```
ACOS-4          COBOL/SDA          V/R  01.5      **********    COBOL/SDA-4    **********      DATE: 79-06-22  PAGE:  21

**********************************
*  FIGURE  OF  PROG. & MACRO RELATION  *
**********************************
```

PROG. NAME	1 1 2 3 4 5	1 6 7 8 9 0	1 1 1 1 1 1 1 2 3 4 5	1 1 1 1 2 6 7 8 9 0	2 2 2 2 2 1 2 3 4 5	2 2 2 2 3 6 7 8 9 0	3 3 3 3 3 1 2 3 4 5	3 3 3 3 4 6 7 8 9 0
CDEXEC	* * * * *	* * * * *	* * * *					
CDMAIN	* * *	* * *		*	* * * * *	* * * * *	* * * * *	*
CSIGET	* *			*	*	* *	*	* * * *

1 ED	2 SPE-NAM	3 WS
4 A-INP-BUF	5 DD-SPR-BUF	6 ID-IND-FLG
7 PD-SDA-PAR	8 WD-IMP-FLG	9 TD-IDN-TBL
10 TD-NAM-TBL	11 TD-APR-TBL	12 D-CONST
13 GD-IMP-LST	14 CHK-INDENT	15 EDX
16 A-INP-TYP	17 DD-SPR-PAR	18 W-TMP-WRK
19 Q-PTR-XXX	20 Y-MC-FIJK	21 Z-DSP-X40
22 LS	23 X-CHR	24 SCT
25 OPN-SPR	26 OPN-INP	27 GET-INP
28 INSPECT	29 UNSTRINGX	30 NAM-SPR
31 FETCH-PAR	32 +1	33 OPN-MACLX
34 OPN-MACO	35 PUT-MAC	36 CLS-MAC
37 IOS	38 SELECT-SIF	39 FS
40 S-BUF		

PROG. NAME	4 4 4 4 4 1 2 3 4 5	4 4 4 4 5 6 7 8 9 0	5 5 5 5 5 1 2 3 4 5	5 5 5 5 6 6 7 8 9 0	6 6 6 6 6 1 2 3 4 5	6 6 6 6 7 6 7 8 9 0	7 7 7 7 7 1 2 3 4 5	7 7 7 7 8 6 7 8 9 0
CDEXEC								
CDMAIN								
CSIGET	* * * * *	* *						

	41 R-S-NMB	42 GET-QF
43 CLS-SQF	44 OPN-QF	45 UNSTRING
46 OPN-SQF	47 CLS-QF	

④ 構成図

プログラムの構成（図3-12）や，1つのプログラムの中の手続きの構成（図3-13）をトリー形式で出力する．

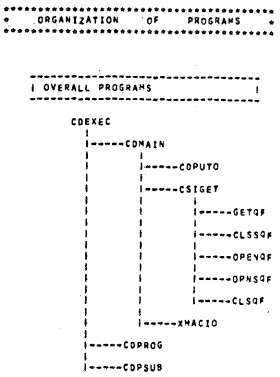

図3-12 プログラム構成図

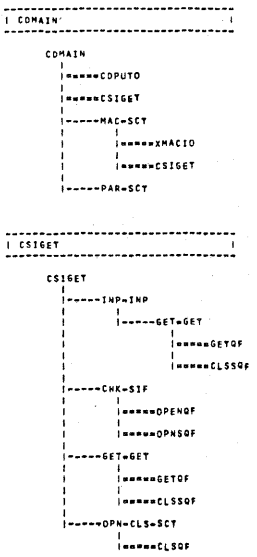

図3-13 手続き構成図

4．COBOL/Sの利用状況，今後の方向など

4.1 利用状況

COBOL/S言語及びCOBOL/Sプリプロセッサは，当初日本電気社内におけるプログラム開発／保守の効率化の為にそのプロトタイプが開発され，十分効果があったので昭和51年正式にリリースされたものである．デシジョンテーブル記述機能およびCOBOL/Sドキュメンテーションエイドは追加機能として昭和53年にリリースされた．

昭和54年3月末現在の利用状況は表4-1のようになっている．

機種	利用者数
小型	63
中型	50
大型	24

表4-1 COBOL/S利用状況

COBOL/Sを利用した場合のCOBOL利用に対する効果は，10Kステップ程度のプログラムの場合，設計に多少工数がかかるようになったが，製造，デバッグ，保守にかかる工数が減り，全体として40％工数削減という例がある．これはプログラムの開発ステップ数が増大するとますます効果があることが経験的に知られている．

さらに同様の思想のもとに，FORTRAN言語をベースにしたFORTRAN/Sも開発され，利用されている．

4.2 今後の方向

CODASYLによる改訂COBOL（1978年）にはプログラムの構造化機能が一部加わっているので，これからのCOBOL/Sはさらに設計用ツールとしてまた，ドキュメンテーション用ツールとしての方向づけがなされてくる．そこで考えられるのが第1にデータ部の支援機能である．現在のCOBOL/Sは主に手続き部に対して構造化やドキュメンテーションの支援機能があるが，データ部に対する支援に

より，手続きとデータを明確に分離する方向がある．第2にEDPシステム開発のライフサイクルを通して，要求定義・検証ツールやテストツールといかにうまく組み合わせてトータルなシステム開発サポートツールとしてゆくべきかという問題である．

おわりに．

プログラムドキュメンテーションの一方法として，ソースプログラムとドキュメントの一元化を実現するCOBOL言語をベースにした擬似言語とその処理系について紹介した．筆者も含めて，体力がものをいうコーディング／デバッグの作業から開放されるべく，有効な各種ツールの開発に参考となれば幸いである．

参考文献

[1] STEPS-4 概要説明書，日本電気（株）

[2] 管理者のためのソフトウェアの標準化(1)～(6)，水野幸男・東基衛，Computer Report 1978年7月～1979年1日

[3] ドキュメンテーションの標準化，マックス グレイ・キース R.ロンドン（竹下亨訳）日本経営出版会

[4] ソフトウェア・エンジニアリングへの道(1)～(11)，宮本勲・東谷秀夫，bit Vol.9 No.4～No.14，Vol.10 No.1～No.5

[5] DECISION TABLE Program の管理，標準化への応用について，関東NEACユーザ会資料，1970年7月

[6] ソフトウェアの標準化，水野幸男・東基衛編，日本経済新聞社

[7] Data Documentation and Decision Tables，D.L.Fisher，COMMUNICATION OF THE A.C.M.，1966年1月

[8] Pseudo-language and their pre-processors，S.Shinozawa 他，IFIP77 Congress

[9] COBOL/S言語説明書＜基本機能編＞，日本電気（株）

[10] COBOL/S言語説明書＜デシジョンテーブル機能編＞，日本電気（株）

[11] COBOL/Sドキュメンテーションエイド説明書，日本電気（株）