JAERI-M
89-146

ソース・プログラムの編集システム

1989年10月

清水　勝宏・平山　俊雄・白井　浩
谷　啓二・内藤新司朗*・安積　正史

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-M レポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　いばらき印刷㈱

ソース・プログラムの編集システム

日本原子力研究所那珂研究所臨界プラズマ研究部
清水　勝宏・平山　俊雄・白井　　浩
谷　　啓二・内藤新司朗*・安積　正史

（1989年9月14日受理）

大規模なソース・プログラムを編集，管理する事を目的とした，汎用性のあるシステムNEWORG（NEW Program organization system）を開発した。NEWORGは，ソース・プログラム・パッケージから，ユーザの計算目的に必要なモジュールを選択し，編集する機能をもつ。システムは，三つのステップから成る。ANALYZERステップでは，ソース・プログラムを解析し，モジュール相互の呼出し関係について調べる。DESIGNERステップで，ユーザは会話形式でもって，コードの仕様を決める。ENGINEERステップでは，不要なステートメントやモジュールを捨て，必要なモジュールのみを選択し，ユーザの目的とするコンパクトなロード・モジュールを組み上げる。このシステムを，トカマク輸送コードの様な多種多様な目的仕様をもって開発された大規模なコード群に適用する事により，コード群をLibrary化する事が可能である。また，組み上げたロード・モジュールについての情報（作成日付，使用ルーチン名等）を，コード・スペック・データとして保存しているので，ロード・モジュールの管理が容易に行なえる。

那珂研究所　：〒311－01 茨城県那珂郡那珂町大字向山801－1
＊（財）原子力データセンター

New program organization system (NEWORG)

Katsuhiro SHIMIZU, Toshio HIRAYAMA, Hiroshi SHIRAI
Keiji TANI, Shinjiro NAITO* and Masafumi AZUMI

Department of Large Tokamak Research
Naka Fusion Research Establishment
Japan Atomic Energy Research Institute
Naka-machi, Naka-gun, Ibaraki-ken

(Received September 14, 1989)

NEWORG is the system to control and manage large-scale FORTRAN source programs. This system edits modules, which are necessary for user's calculation purpose, from source program package. It consists of three steps, i.e., ANALYZER, DESIGNER and ENGINEER. On ANALYZER step, an objective source program is analyzed to obtain the calling relation between the modules with the condition. On DESIGNER step, the user selects the option flags which specify the calculation model by conversational procedure. On ENGINEER step, the unnecessary statements or modules are discarded and the necessary modules are linked. When ENGINEER step is completed, the user obtains the load module which is appropriated for his purpose. This system is of great use to produce a compact code from a large-scale source program package, for instance, a Tokamak Transport Code. The information on the code produced is preserved, so that NEWORG enables us to manage the load modules.

Keywords: Library System, FORTRAN Source, Tokamak Transport Code, ANALYZER, DESIGNER, ENGINEER

---

* Nuclear Energy Data Center

目　　次

Contents

1. 序

核融合研究において，大規模な計算機コード群を用いたシミュレーション解析が，プラズマ・パラメータを予測し，装置設計の指針を与える上で重要な役割を果している。さらに，実験データの解析，理論モデルの検証を行う上でも，有効な手段を与える。トカマク・プラズマ内で生じている輸送現象の解明は，核融合達成の鍵を握る要因の一つであり，トカマク輸送コードが，それに大きく貢献している。トカマク輸送コードは，プラズマ密度，温度の空間分布を求めるモジュールを核とし，図1－1に示す様な多くの物理モデルを含む。NBI加熱計算コード，ICRF，LHRFといった各種RF加熱計算コード，中性粒子輸送コード，不純物輸送コード，MHD現象が輸送に与える影響（鋸歯状振動，ティアリング・モードによる磁気島）を調べるコード等がある。各モジュールは2000～4000ステップ程度のソースから成り，ソース・プログラムは，全体で50000ステップにも達する。NBI加熱計算コードを一つとっても，Stixの定常解を用いるもの，速度空間での拡散現象を記述するFokker Planck方程式を解くもの，あるいはモンテ・カルロ法によって，高速イオンの軌道効果をとり込んだより詳細な計算モデルがあり，各物理コードには，いくつものバリエーションがある。また，プラズマ・パラメータの予測計算だけでなく，実験で測定されたプラズマ・パラメータをもとに，粒子及び熱拡散係数を評価する事を目的とする場合もある。こうした多くの物理計算コード及び多種多様な計算目的があるため，輸送コードは巨大なものとなっている。さらに核融合研究の進展あるいは計算機の性能の向上とともに，各々モデルが改良され，その内容を増やしつつある。この様な特徴をもつトカマク輸送コードのソース・プログラムを扱う上で，Libraryという考えがなぜ必要かを以下に述べる。本システムによって，巨大ソース・プログラムを標準化（Library化）する事が可能であり，これによって得られる利点についても述べる。

トカマク輸送コードは，プラズマの輸送現象の解析あるいは，実験データの整理のために多くのユーザに利用されている。固定された目的仕様に対しては，ユーザは入力データだけを用意し，ロード・モジュールを実行する事により目的とする結果を得る。しかし，実際には，目的毎にプログラムを修正あるいは新しいサブルーチンを追加しなければならない事が多い。コードを修正するにあたっては，変数の意味だけでなく，計算プロセスを理解する必要がある。従って，ユーザにとって必要なのは，ロード・モジュールではなくて，ソース・プログラムである。しかし，巨大なソース・プログラムを各ユーザが持つ事は，ディスクの記憶領域の無駄であり，コードのメインテナンスあるいは改良を行う事を困難にする。この問題を解決するには，Libraryの考え方が必要である。Libraryとして，標準のモジュール全てをシステム・ファイルに登録しておき，ユーザは必要に応じて，このソース・パッケージから，ロード・モジュールを作成する。コードのメインテナンス，改良／修正は，このシステム・ファイルに対して行う。これによって，ユーザは最新のモジュールを使用でき，ユーザはシステムのモジュールの中で修正したルーチン，追加したルーチンのみを自分のユーザ・ファイルの中に持つだけで良い。システムの一元管理によって，コードのメインテナンス，改良／修正が容易となり，複数のユーザによるコードの無秩序な修正が避けられる。又，同じ計算目的を持った物理コードの中には，共通するモジュールが少なくない。例えば，1次元と2次元中性粒子輸送コード，上下対称と非対称平衡コードには共通するルーチンは多い。これらのコードをシステム・ファイルに登録しておけば，

格納する記憶領域が少なくてすみ，同じ機能をもつモジュールを重複してメインテナンス／開発する必要がなくなる。その結果，モジュールの共有化が可能となり，開発コストが軽減される。

大規模ソース・プログラム・パッケージをLibrary化するには，ユーザの計算目的に応じて，必要なモジュールをシステム・ファイルから選択し，ロード・モジュールを組み上げる機能(ORGanization)をもつシステムとする必要がある。1次元輸送コードをLibrary化するために，大規模なソース・プログラムを管理するライブラリー・システム[1]が開発された。このシステムで書かれたソース・プログラムの例を図1－2に示す。図中，(b)の部分は，EOS言語[2]によるサイズ変数定義，コモン・ブロック定義，そしてその展開の方法を示す。モデルの選択は(c)に示す様な記法を用いて記述され，この行をオプション行と呼ぶ。各モジュールの最後には，サブ・プログラムの呼出し関係(展開指示)を書く((d)の部分)。ORGが起動されると，展開指示が解析され，ユーザの計算仕様に応じて，必要なモジュールが選択される。次にオプション行の論理値が計算され，偽の場合には，そのソース・ラインが削除される。最終的に，ユーザの必要とする最小限のロード・モジュールが得られる。このシステムは，トカマク輸送コードの様な，巨大ソース・プログラムの管理システムとして有効であった。しかし，システムとして以下の様な点で汎用性に欠けるため，いくつか不便な点があらわれている。

(1) ファイル形式がGEMであり，また言語がEOS言語であるため，システムが計算機の機種に依存する。

(2) オプション行が特殊な記法である点。

(3) 各モジュール毎に，展開指定を書く事のわずらわしさ。

これらの点を改善し，より汎用性のあるシステムにするため，NEWORGが新たに開発された。ファイル形式は，区分データ・セット(POファイル)，順編成データ・セット(PSファイル)のいずれでも良い。言語には，FORTRAN77を採用し，サイズ変数，コモン・ブロックの展開には，FORTRAN77のincludeの機能を用いる。モデルの選択のオプション・フラグには，parameter変数を用いる。そして，オプション行は，論理if文，ブロックif文で記述するものとする。各モジュール毎にtree構造を解析するため，NEWORGにおいては，展開指示を書く必要がない。

NEWORGは，モジュール相互の呼出し関係を解析し，ユーザの必要なモジュールのみを選択し，ロード・モジュールを組み上げるユーティリティである。FORTRAN77で書かれたソース・プログラムを対象にしており，唯一の規約は，「モデルの選択は，parameter変数を用いたif文で行う。」この点だけである。従って，その利用にあたって，ソース・プログラムの修正を行う必要は，ほとんどない。

NEWORGは，ANALYZER，DESIGNER，ENGINEERの三つのステップから成る。ANALYZERでは，ソース・プログラムを解析し，各モジュールの呼出し関係(ツリー情報)を得る。DESIGNERでは，ユーザは会話処理により，計算コードの仕様(コード・スペック・データ)を決める。ENGINEERでは，コード・スペック・データを基に，必要なモジュールを組み上げる。その際，オプション・フラグの値に応じて，ソース・ラインは修正される。このユーティリティは，大規模なソース・プログラム・パッケージから，コンパクトなロード・モジュールを作成する上で非常に有益である。同一名サブルーチンが複数

個ソース・プログラムにあった時，DESIGNERのステップで，どのサブルーチンを用いるかをユーザに問い合わせる。この機能によって，システムのモジュールとユーザのモジュールの置換が容易に行われる。トカマク輸送コードの様に多種多様な計算目的に対応して開発されたコードは，必然的にIF文の多いプログラムとなる。このIF文がくり返し回数の多い部分に現われた場合，IF文の判定に計算時間が費やされる事になる。しかし，NEWORGで組み上げたロード・モジュールには，実行前に確定した論理値をもつIF文は削除されるため，実行がはやくなるという利点がある。計算モデルを限定するオプション・フラグの値，ロード・モジュール作成時に用いたモジュールの名前はコード・スペック・データとして保存される。従って，ロード・モジュールの管理，その再生を簡単に行う事ができる。NEWORGはPFD(Program Facility for Display users)[3]を用いて，メニュー画面によるデータ入力の方法を採用している。その結果，前回用いたデータが表示され，入力するべきデータが少なくて済むので，初心者にも簡単に利用する事ができる。

Tokamak Transport Code

	(a)	(b)	(c)
● Main plasma transport code	○		
● Neutral transport code	○		○
● NBI heating deposition code	○		○
● LHRF heating deposition code		○	
● ICRF heating deposition code			
● Impurity transport code			
● Scrape-off transport code			
● MHD stability code	△		
● 2D Equilibrium code	○	○	○

Variation

- Prediction code
  - Diffusion coefficients ---→ Ne(r), Te(r), Ti(r), Jz(r)
- Experimental analysis code
  - Xi, Ti(o) ---→ Ti(r), TauE, Wkin
  - Ne(r), Te(r), Ti(r) ---→ De(r), Xe(r), Xi(r)

Calculation Purpose

(a) NBI heating plasma
(b) Current Drive Experiment
(c) Global Confinement (TauE vs. Ip/Pb)

図1－1　トカマク輸送コードに含まれる物理コード

トカマク・コードの特徴の一つは，コードが多くのモジュールから成るために巨大であり，計算目的によって使用するモジュールが異なる事である。例えば，閉じ込め特性を調べる事を目的とする(c)については，図の○印で示す様に中性粒子輸送コード，NBI加熱コード，平衡コードが必要である。

Example of 1D-LIBRARY

```
*SIZE
    N=5, M=7
*COMDECK COM1                                  (b)
    COMMON /COM1/ X(N,2),Y(N+M,9)
*DECK SUBA
    INCLUDE SUBA
    T=Y(1,1)+2.0
    CALL BBHY   :.LA(PR.EQ.1)                  — (c)
    X(1,1)=FUNC(T)
    RETURN
    END
/* INCLUDE */
    -INC BBHY, IN=SYSTEM :.LA                  (d)
    -INC FUNC, IN=SYSTEM
```

改　良　点

1D-LIBRARY		NEWORG
(a) file形式 GEM	⟶	PS, PO file
(b) EOS言語	⟶	Fortran77 Includeの機能
(c) Option line	⟶	Parameter変数を用いたIF文
(d) 展開指定	⟶	Tree構造の解析
(e) 実行はBatch	⟶	TSS処理可能

図1－2　1Dライブラリー・システムにおけるプログラムの記述例とNEWORGの改良点

ソース・プログラム記述の中で(b),(c),(d)はそれぞれ,EOS言語によるinclude展開,Option line,モジュールの展開指示の記述方法の例を示す。

2. システムの概要

モデル選択には,FORTRAN77のparameter変数を用いた論理if文,Block if文で記述する。パラメータ変数は計算の実行の途中で値が変わらない。その変数及び定数で構成されたif文の論理値は実行の前に確定する。従って,計算モデルの仕様を決めるオプション・フラグの変数として用いるのに,パラメータ変数が適している。インクルード・ファイルの中でもって定義されたパラメータ変数は,ソース全体に影響するという意味でGlobal変数と呼ぶ。サブルーチン内で定義されたパラメータ変数はそのサブルーチンの中でしか影響しないのでLocal変数と呼ぶ。

簡単なプログラムを例に,NEWORGの機能について説明する。

図2－1に示すプログラムは,典型的なトカマク輸送コードのメイン・ルーチンの例である。NEWORGは,ANALYZER,DESIGNER,ENGINEERの三つのステップから成る。

(1) ANALYZER

各モジュール毎に,呼び出しルーチン,パラメータ変数を用いたif文について調査する。メイン・プログラムにおいては,呼び出しプログラムとしてTRINPT,TRSTEP,TRNTL,TRNBH,TRNBST,…がある。

オプション・フラグ(Global変数)はLNTL,LNBI,LRFであり,それらは,それぞれサブルーチンTRNTL,TRNBH,TRRFを制御している。また,Local変数はLFIGであり,図形処理ルーチンGSTA,GPLS,GENDを制御している。これらの解析結果をツリー情報と呼ぶ。

(2) DESIGNER

ユーザは会話処理により,コードの仕様を決める。

起点ルーチンより始めて,ツリー情報に基づいて下位ルーチンへと進み,そこで用いられているG／L変数の値を決めていく。また,同一名のサブルーチンがあった場合には,どれを用いるか選択する。このステップで決まったG／L変数の値,使用するモジュール名のテーブルをコード・スペック・データと呼ぶ。

```
Global or Local変数の問い合わせに答える。
  ? Global Spec LNBI = 2
  > Input new spec. (LNBI) ==> 1
重複名ルーチンの選択
  ? Same sub-program name found (TRINPT)
     1. J3051.LIB2D.FORT77(TRINP)
     2. J3859.TOKMK.FORT77(TRINP)
     > Input select dataset num ==> 2
```

(3) ENGINEER

コード・スペック・データをもとに，必要とするモジュールを組み上げる。パラメータ変数によって確定した論理値をもつソース・ラインは，その値に応じて修正される。DESIGNERのステップでG／L変数の値をLNTL＝1，LNBI＝1，LRF＝0，LFIG＝0と決めたとすると，オリジナルのソース・プログラムはENGINEERによって図2－2の様になる。

NEWORG全体の流れ図を図2－3に示す。システムとして標準的に用いられる輸送コード，図形処理ルーチン，行列演算や数値微積分といった科学計算用サブルーチン・ライブラリー等をあらかじめANALYZERによって解析しておき，そのツリー情報を保存しておく。この場合，ユーザはANALYZERからのステップではなくDESIGNERのステップから開始できる。DESIGNERでは使用するファイルのツリー情報のデータ・ファイル名を入力し，起点ルーチン（普通はメイン・ルーチン）を指定する。会話処理によって，ユーザは，G／L変数の値を決め，重複名ルーチンがある場合には，どれを用いるかを選択する。コード・スペック・データには組み上げたソース・ファイルについての情報（作成日付，使用したモジュール名，G／L変数の確定した値）が保存される。従って，ロード・モジュールの管理が容易に行える。また，ENGINEERで，コード・スペック・データのファイル名を指定すれば，ロード・モジュールを再生する事が可能である。

```
C-----  TEST PROGRAM  -----
C
*INCLUDE LOPT,FIXED,NOINSOURCE
      PARAMETER (LFIG=1)
C                    Local変数
C::INPUT DATA
      CALL TRINPT
C
C::STEP
  100 CONTINUE
      CALL TRSTEP(IEND)
      IF(LNTL.GT.0) CALL TRNTL
      IF(LNBI.GT.0) CALL TRNBH
      IF(LNBI.EQ.1) THEN
            CALL TRNBST
      ELSE
            CALL TRNBMC
      ENDIF
      IF(LRF.GT.0) CALL TRRF
      IF(IEND.EQ.0) GO TO 100
C
C::FIG OUTPUT
      IF(LFIG.GT.0) THEN
            CALL GSTA(0)
            CALL GPLS
            CALL GEND
      ENDIF
C
      STOP
      END
```

インクルード・ファイル(LOPT)の内容

```
PARAMETER ( LNTL=1, LNBI=2, LRF=1 )
                    Global変数
```

図2－1　トカマク輸送コードのメイン・ルーチンの例

```
C-----  TEST PROGRAM  -----
C
*INCLUDE LOPT,FIXED,NOINSOURCE
      PARAMETER ( LFIG    =     0 )
C
C::INPUT DATA
      CALL TRINPT
C
C::STEP
  100 CONTINUE
      CALL TRSTEP(IEND)
      CALL TRNTL
      CALL TRNBH
      CALL TRNBST
      IF(IEND.EQ.0) GO TO 100
C
C::FIG OUTPUT
C
      STOP
      END
```

インクルード・ファイル(LOPT)の内容

```
PARAMETER ( LNBI    =     1 )
PARAMETER ( LNTL    =     1 )
PARAMETER ( LRF     =     0 )
```

図2－2　NEWORGによって組み上げられたソース・プログラム

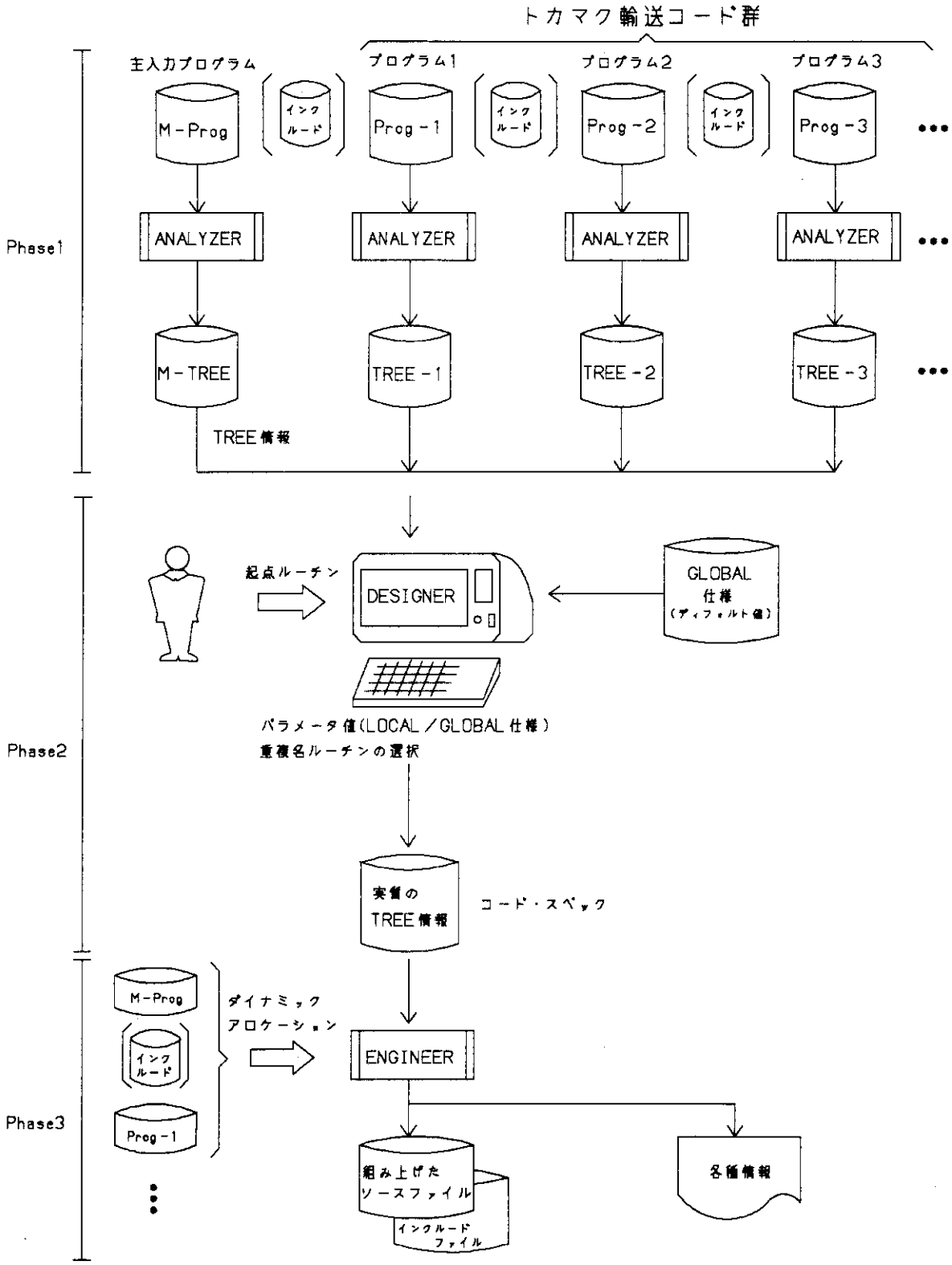

図2－3　NEWORGの処理形態

3. 機　　能

NEWORGは，FORTRAN77の言語でもって書かれたソースプログラムを対象にしている。

NEWORGで重要なステートメントは，G／L変数の値を定義するパラメータ文及びそれらの変数を用いた計算モデル選択のif文である。NEWORGでは，このif文を解析し，条件式の論理値が偽となったif文の範囲にあるステートメントは，プログラム組み上げの際に，削除される。その結果，不要なサブルーチンcall文や，計算式は除外され，無駄のないプログラムが組み上がる。この基本的機能のほかに，パラメータ変数，モジュールについての説明文をソースに書くstatementがある。DESIGNERのステップで，パラメータ変数の値あるいは重複名モジュールの選択を行う時に，これらのコメントが端末に出力される。モジュール選択がモデルの仕様に応じて自動的に選択される機能もある。いずれもこれらの文は，FORTRAN77言語ではコメントに分類される。

旧システムであるORGのユーザの為に，

- オプション行[1]
- プリント・ユーティリティ文[1]
- プロット・コマンド文[4]

は，書き換えなくても，NEWORGでもそのまま使用する事ができる。

インクルード文については，

　　INCLUDE（メンバー名）

　＊INCLUDE　メンバー名

のどちらも用いる事ができる。

3.1 計算モデルの仕様 : パラメータ文

モデル選択を行うオプション・フラグにはパラメータ変数を用いる。ここで定義された値は,DESIGNERのステップで,ユーザがモデル仕様を決める際に,デフォルト値として出力される。

(文法)

FORTRAN77のPARAMETER文の型式で記述する。NEWORGでは,その宣言場所により,パラメータ変数はGlobal変数とLocal変数に分かれる。

Global変数　　インクルード・ファイル内で宣言されているもの。

Local変数　　サブルーチン内で宣言されているもの。

タイプには整数型と論理型があり,他の型はオプション・フラグとして用いる事はできない。パラメータ変数の型に応じて,パラメータ変数同志で,算術演算,論理演算が可能である。

複数のインクルード・ファイルを使用する場合は,なるべくGlobal変数を1メンバに統一した方がよい。複数のメンバで同一名のG変数を重複して定義してはならない。Local変数名は重複してもよい。

PARAMETER文の例を図3－1に示す。

```
      SUBROUTINE TRSOL                    インクルード・ファイル(LOPTの内容)
      INCLUDE ( LOPT )
      PARAMETER ( LDBG=1 )                          ↓
C                Local変数                  PARAMETER (LPR=0,LDE=0,LTR=1)
C:: DENSITY EQU.                            PARAMETER (LHY=LPR+LDE+LTR)
      IF(LHY.EQ.1) CALL YDHY(DEN,1)         LOGICAL LALF
      IF(LHY.EQ.2) CALL YDHY2(DEN,2)        PARAMETER (LALF=LDE.EQ.1 .AND. LTR.
C
C:: ALPHA HEATING
      IF (LALF) CALL YALF
C
```

図3－1　PARAMETER文の例

3.2 計算モデルの選択 : IF文

パラメータ変数を使用しているIF文を解析して，その論理値が実行前に確定するIF文は，ENGINEERでソース組み上げの際に論理値（真／偽）に応じてソースが修正（出力／不出力）される。

（文法）

モデル選択のIF文は，FORTRANの論理型IF文もしくはブロックIF文である。そしてその論理式は，パラメータ変数と定数によってのみ構成され，実行前に確定するものでなければならない。従って，計算型GO TO文，算術IF文，あるいは条件式に普通の変数が含まれるIF文は，NEWORGのモデル選択IF文ではない。FORTRAN77の文として，ENGINEER後のソースにそのまま残る。条件式の中で，パラメータ変数と普通の変数が混在する場合，ANALYZERで警告メッセージが出力される。図3－2に示したAの場合，ブロックIF文を用いて修正する事によって，モデル選択IF文として認識される様になる。

```
      SUBROUTINE HEAT
      PARAMETER ( LNBI = 2, LMSG = 0 )
      PARAMETER ( NDSZ = 101 )
      COMMON /PLS/ TE(NDSZ),N
C
      IF (LNBI .EQ. 1) THEN
        CALL YNBI1
      ELSE IF (LNBI .EQ. 2) THEN
        CALL YNBI2
      END IF
C
      CALL SUBA ( X )                    Ⓐ
      IF (LNBI .GT. 0) THEN
        CALL YBCUR
      ELSE IF (X .LT. 0.0) THEN
        CALL YBCON
      END IF
C
      IF (LMSG .NE. 0) CALL MSGOUT
C
      IF (N .GT. NDSZ) THEN
        WRITE(6,*) 'ERROR ! N=',N
        STOP                             Ⓑ
      END IF
C
      RETURN
      END
```

修正

```
      CALL SUBA ( X )
      IF (LNBI .GT. 0) THEN
        CALL YBCUR
      ELSE
        IF (X .LT. 0.0) THEN
          CALL YBCON
        END IF
      END IF
```

図3－2 IF文の例

ブロックIF文でパラメータ変数と普通の変数を同時に使用するとANALYZERで警告メッセージが出力される。Aの部分は，図に示した様に修正すれば，NEWORGのモデル選択IFとなる。Bの部分は，ディメンジョンのサイズを決めるパラメータ変数と普通の変数を用いたIF文であるが警告メッセージは無視して良い。

3.3 モデル説明文 ： コメント文

システム・ファイルのオプション・フラグ（パラメータ変数）の意味をユーザが知っているとは限らない。このため，DESIGNERでオプション・フラグの意味あるいは重複名ルーチンについての説明文を補助出力する機能がある。

（文法）

FORTRAN77のコメントの型式（第1カラムが ′C′ ）で記述する。

COMM(パラメータ変数名)　　コメント

COMM(サブルーチン名)　　コメント

Global変数についての説明はINCLUDEメンバの中に記述する。

Local変数，及びサブルーチンについての説明は，サブルーチンの中に記述する。

図3－3に使用例を示す。

INCLUDEメンバー（LOPT）の内容

```
      PARAMETER (LNBI=2)
COMM(LNBI) LNBI=0 : OHMIC PLASMA
COMM(LNBI) LNBI=1 : STIX SOL / LNBI=2 : OFMC CODE
```

ソース・プログラムの内容

```
      FUNCTION SIGCX(EO,TI)
COMM(SIGCX) CROSS SECTION BY RIVIERE
C
      END
C
      FUNCTION SIGCX(EO,TI)
COMM(SIGCX) CROSS SECTION FROM AURORA CODE
C
      END
```

DESIGNER での出力

```
?GLOBAL SPEC LNBI=2
CMT.. LNBI=0 : OHMIC PLASMA
CMT.. LNBI=1 : STIX SOL / LNBI=2 : OFMC CODE
>INPUT NEW SPEC. (LNBI)-->1   enter

?SAME SUB-PROGRAM NAME FOUND <<SIGCX>>
  1. J3859.TOKMK.FORT77(SIGCX1)
CMT.. CROSS SECTION BY RIVIERE
  2. J3859.TOKMK.FORT77(SIGCX2)
CMT.. CROSS SECTION FROM AURORA CODE
>INPUT SELECT DATASET NUM--> 1   enter
```

図3－3　COMMENT文の例

3.4 ルーチンの使用条件 ： コンディション文

単機能のモジュールであっても，シミュレーション解析を行う過程で，いくつものバリエイションができる。その結果生じた複数の同一名のルーチンをDESIGNERのステップで，ユーザがいちいち選択することは時に負担になる。このような時にコンディション文を用いて，ソース・プログラムの中に，使用条件を書き込んでおけば（無条件使用，無条件不使用），条件式が真となったサブルーチンが自動的に選択される機能がある。

（文法）

FORTRAN77のコメントの型式で記述する。

CORG　LIBCND　条件式

Lは7カラム目でなければならない。

条件式には論理値を記述することもできる（.TRUE.／.FALSE.）。

コンディション文の使用例を図3－4に示す。

```
C::CROSS SECTION
C              BY RIVIERE
      FUNCTION SIGCX(EO,TI)
      INCLUDE(LOPT)
CORG  LIBCND LCRCX.EQ.1
C
      END
```

```
C::CROSS SECTION
C              FROM AURORA CODE
      FUNCTION SIGCX(EO,TI)
      INCLUDE(LOPT)
CORG  LIBCND LCRCX.EQ.2
C
      END
```

```
C::OFMC CODE
      SUBROUTINE YNBI
      INCLUDE(LOPT)
C
C
      COEF=FUCT(TEO)
C
      END
```

```
C::OFMC CODE
      SUBROUTINE YNBI
      INCLUDE(LOPT)
CORG  LIBCND .FALSE.
C
      COEF=0.0
C
      END
```

図3－4　コンディション文の例

荷電交換の断面積を求めるルーチには，二つあるが，G変数　LCRCX＝1が選ばれた時Rivieresの計算式が選択される。

二番目の例は，COEF＝0.0としたバージョンのYNBIは無条件に使用されない。特別な計算目的をもって作成されたルーチン（しかしユーザが使うには，標準的ではない），古いバージョンのルーチン（新しいバージョンとの比較あるいは古い結果を再現する必要があるため）をシステム・ファイルに保存する時，このステートメントが便利である。標準ルーチンとして無条件使用したい場合には，コンディション文の論理値を.TRUE.とすれば良い。

3.5 宣言文の選択 : オプション行

モデルの選択によって影響を受けるのは，モジュールのみとは限らない。データ領域の場合もある。不必要なデータ領域を排除するため，INCLUDE文，DIMENSION文の取捨選択を行う機能がある。

（文法）

旧ORGのオプション行の記述方法と同じである。

＊INCLUDE　メンバ名　：．オプション・ラベル（G／L変数を用いた論理式）

＊INCLUDE　メンバ名　：．オプション・ラベル

一度G／L変数を用いてオプション・ラベルが定義されると，そのサブルーチンの中の任意の個所でオプション・ラベルのみを指定する事により，その文をENGINEERで取捨選択する事ができる。この文はFORTRAN77のステートメントでない事に注意を要する。

オプション行を使用した例を図3－5に示す。

```
      SUBROUTINE YNBI
      INCLUDE(LOPT)
      INCLUDE(CDIF)    :.LA(LDIF.EQ.1)
      DIMENSION CD(20000) :.LA      オプション行

C::DIFFUSION EFFECT
      IF(LDIF.EQ.1)THEN
      DO 100 J=1,20000
      CD(J)=TEO*0.01*FLOAT(J-1)
  100 CONTINUE
C
      ENDIF
```

図3－5　オプション行を使用した例

3.6 コモン変数名の変更 ： ローカル・コモン変数

インクルード・ファイルの指定メンバ内に宣言されているコモン変数名をリネームしてソース・ファイルに展開する。別々に開発されたプログラムにおいて，同じ名前のコモン変数をたまたま用いる事がよくある。これらのコモン変数が共に存在するインター・フェイスのルーチンで，致命的なエラーが生じる事になる。しかし，ローカル・コモン変数の機能を用いれば，このような状況を容易に解決する事ができる。

図の例では，コモン・ブロックCBEAMのコモン変数 R0,Aがそれぞれ，R0X,AXに変更されてソース・プログラムの中に展開されている。

（文法）

FORTRAN77のコメントの型式で記述する。

CORG　INCLUDE　メンバ名，変数名（新変数名）[，‥‥]

メンバ名 ： 展開したいインクルード・ファイルのメンバ名

変数名 ： リネームしたい変数名

新変数名 ： リネーム後の変数名

コーカル・コモン変数の例を図3－6に示す。

```
      SUBROUTINE NBMAIN
*INCLUDE CMPLAS                          インクルード・ファイル
      INCLUDE (CBIRTH)                          (CMPLAS)の内容
CORG  INCLUDE  CBEAM,RO(ROX),A(AX)       COMMON /COMPLS/ RMAJ,RMIN,NE(50),TE(5C
C
C::BIRTH  ( COM BLOCK : CBIRTH )                (CBIRTH)の内容
      RO=RMAJ*1.0E2                      COMMON /COMBIR/ RO,A,HDEP(50,3)
      A =RMIN*1.0E2
      CALL BIRTH
C                                               (CBEAM)の内容
C::HEAT DEPO. ( COM BLOCK : CBEAM )      COMMON /COMBEM/ RO,A,WE(50),WI(50)
      ROX=RMAJ*1.0E2
      AX =RMIN*1.0E2
      CALL BEAM
C
        ENGINEER後のソース

      SUBROUTINE NBMAIN
*INCLUDE CMPLAS
      INCLUDE (CBIRTH)
      COMMON /COMBEM/ ROX,AX,WE(50),WI(50)
C
C::BIRTH  ( COM BLOCK : CBIRTH )
      RO=RMAJ*1.0E2
      A =RMIN*1.0E2
      CALL BIRTH
C
C::HEAT DEPO. ( COM BLOCK : CBEAM )
      ROX=RMAJ*1.0E2
      AX =RMIN*1.0E2
      CALL BEAM
C
```

図3－6 ローカル・コモン変数の例

3.7 リスタート・ルーチン

モデルの選択によって，トカマク・コードは計算時間のかかる場合がある。この場合，リスタート・ルーチンが必要になる。VROUTのルーチンがソース・プログラムの中にあった場合，ENGINEERのステップでINCLUDEのコモン変数の内容を解析した後に，リスタート・ルーチンが作成される。

（文法）

VROUTの名前でサブルーチンを作成する。リスタートに必要なコモン変数を含むインクルードのメンバー名を指定する。

```
SUBROUTINE VROUT
INCLUDE(メンバ名)
INCLUDE(メンバ名)          } インクルード文
      :
      :
RETURN
END
```

図3－7にリスタート・ルーチンが作成された例を示す。

```
      SUBROUTINE VROUT
      INCLUDE(COMNBI)          インクルード・ファイル(COMNBI)の内容
      INCLUDE(COMPLS)          COMMON /CMNBI/ X(3),Y(3),A(3)
      RETURN                          (COMPLS)
      END                      COMMON /CMPLS/ DEN(51,3),TEM(51,3)

                ⇩ ENGINEER実行

      SUBROUTINE VROUT(NFT)
      INCLUDE(COMNBI)
      INCLUDE(COMPLS)
C
C::WRITE
      REWIND NFT
      WRITE(NFT) X,Y,A
      WRITE(NFT) DEN,TEM
      RETURN
C
C::READ
      ENTRY VRIN(NFT)
      READ(NFT) X,Y,A
      READ(NFT) DEN,TEM
      RETURN
      END
```

図3－7　リスタート・ルーチンの例

インクルード文はなくてもよい。その場合には使用したコモンのインクルード・メンバすべてが展開される。

4. プログラム記述

NEWORGは，三つのステップが独立に動作する。計算仕様を決めるDESIGNERのステップは会話処理である。ANALYZERとDESIGNERのステップは，時にCPU時間が，そしてIO回数が非常に大きいためにelapsed timeがかかる場合がある。たとえば，5万ステップのソース・ラインから成る1DライブラリーはANALYZERステップで30秒，DESIGNERステップはモデル仕様に依存するが40秒程度のCPU時間がかかる。そのため，これら二つのステップについては，TSSとバッチ処理が可能である。各ステップでの処理について，簡単に述べる。

4.1 ANALYZERの処理

各モジュール毎に，下位ルーチン名とパラメータ変数を用いたif文について解析する。この時，解析が複雑になる要因の一つは，スレーブ・ルーチンはサブルーチンと限らないでfunctionの場合がある事である。従って，代入文(assignment statement)の中で用いられる変数名であってもそれが配列名なのかfunctionかを認識する必要がある。ANALYZERではこれを次の様に行っている。[5] ソース・プログラムを一行ずつ読み込み，ステートメントのFORTRANの分類を行う。宣言文があらわれた時，配列宣言が行われる変数名はテーブルに格納する。実行文において左かっこの前に現われる変数名について配列宣言の変数かを調べる。テーブルに格納していなければfunctionであり，下位のルーチンとしてその名前を格納する。ANALYZERでは以下の様な項目について解析する。

(1) サブ・プログラム名とその所在位置
(2) スレーブ・ルーチン名とサブ・プログラム内での呼び出し位置
(3) モデル選択IF文の論理式とIF文の位置
(4) G変数の名前，その値とそれが定義されているインクルード・ファイルのメンバー名
(5) L変数の名前，その値とそれが定義されているサブルーチン名

これらのテーブルの作成を終えるとツリー情報ファイルにデータが出力される。

図4－1にトカマク輸送コードのメイン・ルーチン(図2－1)のツリー情報の一部を示す。

スレーブス・テーブル

	CALLEE	%	#INC	CD.P	POS
1 -	LOPT	I	1	0	3
2 -	TRINPT	S		0	7
3 -	TRSTEP	S		0	11
4 -	TRNTL	S		1	12
5 -	TRNBH	S		2	13
6 -	TRNBST	S		3	15
7 -	TRNBMC	S		4	17
8 -	TRRF	S		6	19
9 -	GSTA	S		7	24
10 -	GPLS	S		7	25
11 -	GEND	S		7	26

条件テーブル

+	NST	TYPE	POS	SL.P	NP	NC	(PARAM,%).....	CONDITIO
1 -	1	LOGIF	12	4	1	9	LNTL G	LNTL.GT.0
2 -	1	LOGIF	13	5	1	9	LNBI G	LNBI.GT.0
3 -	1	IF	14	6	1	9	LNBI G	LNBI.EQ.1
4 -	1	ELSE	16	7	0	0		
5 -	1	ENDIF	18	0	0	0		
6 -	1	LOGIF	19	8	1	8	LRF G	LRF.GT.0
7 -	1	IF	23	9	1	9	LFIG L	LFIG.GT.0
8 -	1	ENDIF	27	0	0	0		

図4－1　ツリー情報の例

INCLUDE文もG変数(L変数ではない)によって制御される事がある。(3.4 コンディション文参照) そこでNEWORGではINCLUDE文もスレーブス・ルーチンの一つとして扱う。スレーブス・テーブルにおいて,型を示すI,S,Fはそれぞれ,INCLUDE文,サブルーチン,ファンクションをそれぞれ示す。CD.P及びSL.Pはそれぞれコンディション・テーブルとスレーブス・テーブルの参照用インデックスを示す。

4.2 DESIGNERの処理

DESIGNERで，ユーザは用いるソースファイルのツリー情報（複数個可能）を指定する。指定されたツリー情報を全て読み込み，これらのデータをテーブルとしてメイン・メモリーに持つ。従って会話処理における応答はすこぶるはやい。

次に入力される起点ルーチン（普通はメイン・ルーチン）より始めて，最後の下位ルーチン迄，計算モデルの仕様を決めるオプション・フラグの値，重複名ルーチンの選択を会話処理により決める。

図4－2に，処理のフロー・チャートを示す。ここでNPRGは現在処理中のルーチンの名前，NODEはそのルーチンがツリー構造でどのレベルにあるかを示す。NSTPは，目下処理しているソース・ラインがルーチンの中で何番目にあるかを示す。まだ設定されていないG／L変数を含むIF文に出くわした時，その変数の値の問い合わせが一度だけ行なわれる。重複名ルーチンの選択も同様である。プログラムの中にブロック・データが含まれる時，最後にそれぞれのブロック・データに対して，採用／不採用の問い合わせが行われる。

以下に示す様なデータをツリー情報のテーブルに追加し，コード・スペックデータと呼ばれるファイルに書き出す。

(1) 起点ルーチンを1とし，そのルーチンの出現順序を示すインデックス
　（ゼロは未使用ルーチンである事を示す。）
(2) このテーブルは不要
(3) 確定した論理値
(4) ユーザによって限定されたG変数の値
(5) ユーザによって限定されたL変数の値
(6) 新しいテーブルとして，使用するインクルード・メンバーの名前

DESIGNERの処理

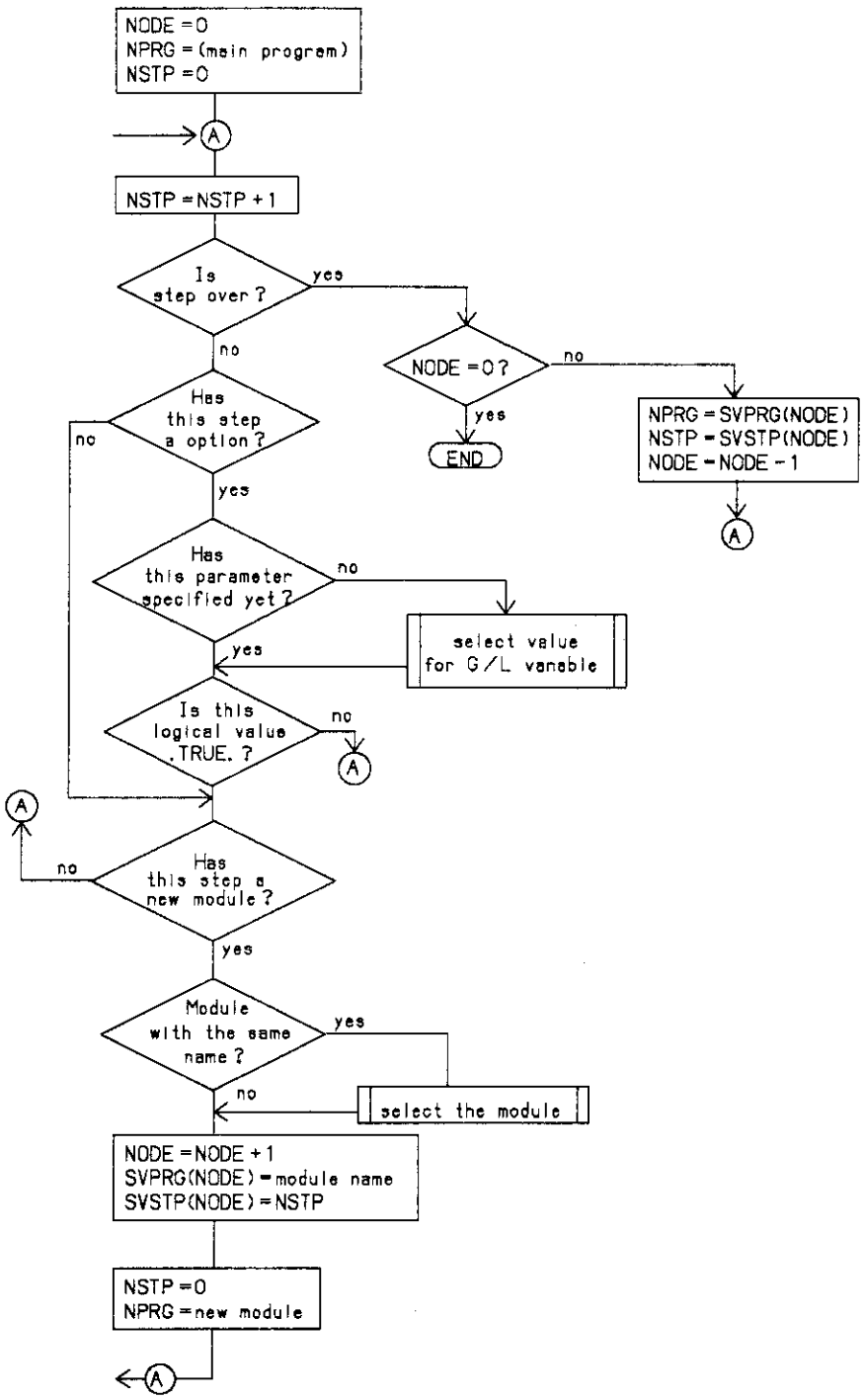

図4－2　DESIGNERのフロー・チャート

4.3 ENGINEERの処理

コード・スペック・データ(1)の出現順序番号の順に，ソース・プログラムのメンバーをダイナミックにアロケーションを行う。そしてソース・プログラムを1行ずつ読み込み，テンポラリー・ファイルに書き出す。その際条件テーブルの論理値が偽のオプション・ステートメントは書き出されない。論理IF文の論理値が真の場合には，IFの部分(例えばIF(LNBI.EQ.1))がとり除かれ，実行文の部分のみを書き出す。Block If, ELSE,ELSE IF文の論理値が真の場合には，これらの文は捨てられ，その範囲内にある実行文のみが書き出される。L変数の値を定義するパラメータ文は，その値をコード・スペック・データのL変数の値に変更する。ソース・プログラムの組み上げを終わると，次に使用されたインクルードの全てのメバーがテンポラリー・ファイルにコピーされる。G変数の値は，コード・スペック・データの値に変更される。リスタート・ルーチンの必要があれば，これを作成する。最後に組み上がったソースに対して，静的解析を行い，未定義ルーチン名，ツリー構造図の出力を行う。

5. メニュー画面

NEWORGの使用にあたって次の様な事に注意してほしい。

(1) 初めてNEWORGを用いる時は，新しくコマンド・プロセジャーNORGを作成する。

```
PROC 0
EX ' J3859.TSSMAC.CLIST(NORG) '
EXIT
```

(2) NEWORGの使用時にはregion sizeとして3072KBが必要なので起動は次の様に行う。

```
LOGON TSS JXXXX／password S(3072)
READY
NORG
```

(3) データセット名は，ユーザIDを含めて，4ケ以内の名前のつながりとする。

データセット名 : Jxxxx.aaaa.bbbb.cccc

入力データセット名の記述は

完全修飾形 : 'Jxxxx.aaaa.bbbb' or
'Jxxxx.aaaa.bbbb(mmmm)'

省略形 : aaaa.bbbb or
aaaa.bbbb(mmmm)

の形式である。

(4) NEWORGの画面については4.2以降詳しく説明しているが初心者の方は，以下に示す画面の使用についてのみ知るだけで，ソース・プログラムの編集は可能である。

- 画面A : プライマリー・メニュー画面
- 画面B : ツリー情報，アロケート及び組み上げがソース／ロード，モジュールかの指定
- 画面D : DESIGNER画面 使用するソース・ファイルとインクルード・ファイルの指定
- 画面E : スタート・ルーチン，オプション・フラグの修正，コード・スペック名
- 画面F : 新しく作成するソース・プログラム名

5.1 NEWORGの画面構成

NEWORGの画面構成を図5－1に示す。

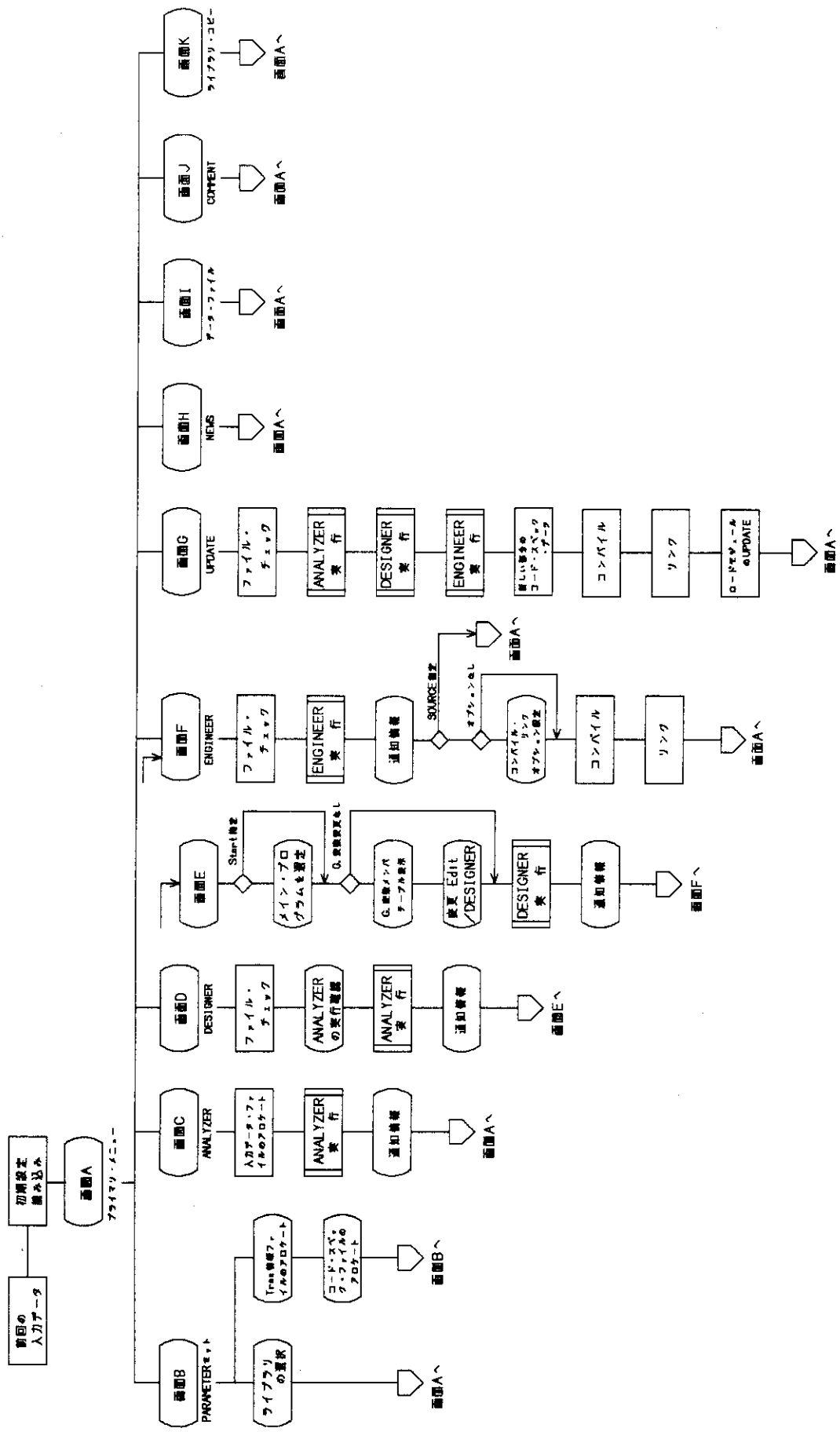

図5－1　NEWORGの画面構成

5.2 画面A : プライマリー・メニュー画面

```
 READY
NORG
  +-----------------------------------------------+
  +  DID YOU LOGON WITH SIZE(3072) KB MEMORY ?   +
  +-----------------------------------------------+
***  enter

---------------------<  NEWORG PRIMARY OPTION MENU  >--------------------------
OPTION --->
                                                          USERID    - J7867
<<< NEWORG >>> ---------------------------------------------TIME      - 16:11
                                                          TERMINAL  - F9526
  0    ATTRIBUTES    - DEFINE NEWORG PARAMETERS           PF KEYS   - 24
  1    ANALYZER      - ANALYSIS OF SOURCE PROGRAM
  2    DESIGNER      - SPECIFICATION OF CALCULATION MODEL
  3    ENGINEER      - ORGANIZATION OF PROGRAM CODE
  4    EDIT          - EDIT DATA SET
  5    UPDATE        - UPDATE LOAD MODULE
  6    NEWS          - MESSAGE FOR LIBRARY USER
  7    LIST          - LIST OF NEWORG DATA
  8    COMMENT       - COMMENT OF TREE INFORMATION FILE AND CODE SPECDATA FILE
  C    COPY LIBRARY  - COPY LIBRARY MEMBERS

  P    PFD           - EXECUTE PFD MENUS UNDER NEWORG(EDIT.UTILITY.TSS...)
  X    EXIT          - TERMINATE NEWORG
```

図5－2 画面A PRIMARY画面

(1) NORGと入力すると，まずPRIMARY画面が表示される。

(2) 初めてNEWORGを使用する場合にはATTRIBUTES画面でパラメータの設定を行う。

OPTION ===> 0 enter

一度パラメータをセットすれば，変更の必要がなければ二回目以降は行う必要がない。

5.3 画面B : NEWORGパラメータのセット

```
--------------------------< NEWORG PARAMETER >-------------------------------
COMMAND --->

<<< ATTRIBUTES >>>                          PRESS ENTER KEY TO SET PARAMETERS.
                                            PRESS END KEY TO RETURN PRIMARY MENU.
   PASSWORD              ---> XXXX      ....(1)
   LIBRARY  (NO/YES)     ---> YES       ....(2)
   TREE INF. DATASET     --->
   CODE SPEC DATASET     ---> *SPEC.DATA   ....(3)
   PRINT TYPE   (A)      ---> 0 (0/1/2)    ===> TSS  (TSS/FILE)
                (D)      ===> 0 (0/1/2)    ---> TSS  (TSS/FILE)  ....(4)
                (E)      ===> 0 (0/1/2)    ---> TSS  (TSS/FILE)
   LOAD/SORC             ---> S  (L/S)                           ....(5)
```

図5－3 画面B ATTRIBUTES画面

(1) パスワードを入力する。このパスワードはBATCH用JCLを作成するのに使われる。

(2) ライブラリ(トカマク輸送コード)使用の有無を入力する。'YES'を入力して enter キーを押すと,図5－4に示す画面が表示されるので,使用するライブラリを選択する。

```
------------------------< LIBRARY DATASET TABLE >---- LINE 000001 COL 001 080
COMMAND --->                                              SCROOL ---> PAGE

<<< SELECT LIBRARY >>>
S - SELECT
SELECT DATASET NAME                          OPT COMMENT
*      ****************************          *** ***************************************
→⑤     'J3859.LIBJT300.FORT77'                1  1D-TRANS  05.10.89
       'J3859.LIBJT100.FORT77'                1  1D-TRANS  ADD SCOOP MODULE
       'J3859.LIBJT80.FORT77'                 1  1D-TRANS  OLD-VERSION
       'J3859.LIB2D.FORT77'                   0  1.5D-TRANS
*** BOTTOM OF DATA ***
```

図5－4 ライブラリ選択画面

(3) ツリー情報ファイル名とコード・スペック・ファイル名を入力する。ツリー情報ファイルは,テンポラリでしか使用しない場合は入力する必要はない。コード・スペック・ファイルはアロケートする必要がある。enter キーが押された時点で入力した名前のファイルが存在していなければ,図5－5に示す様な画面が表示され,ファイルのアロケートを行う。

```
----------------------------< DATASET CREATE >----------------------------
COMMAND --->
<<< CREATE DATASET >>>                    PRESS ENTER KEY TO SET PARAMETERS.

   NEW DATASET NAME   ---> *SPEC.DATA
       SPACE PRIMARY  ---> 50
           INCREMENT  ---> 10
    DIRECTORY BLOCKS  ---> 10
            UNIT      ---> D0010A   (NOT TSSWK)
```

図5－5 データセット・アロケート画面

(4) ANALYZER, DESIGNER, ENGINEERの実行時のメッセージ・クラス及び，それらをファイルに出力するかどうかを指定する。

(5) 組み上げたソースのロード・モジュールを作成するか，ソース・ファイルを作成するかを指定する。

(6) 最後に enter キーを押すと入力データのチェックが行われる。問題がなければパラメータがセットされ，画面右上に´OK´の文字が表示される。

(7) PF3 キーは終了を意味し，PRIMARY画面に戻る。

(8) NORGを多数回使用すると，コード・スペック・ファイルの領域がオーバ・フローする場合があるので，時々コンデンスする必要がある。

5.4 画面C : ANALYZER(ツリー情報解析)

```
-----------------------------< ANALYZER DATA >--------------------------------
COMMAND ===> GO

<<< ANALYZER >>>                          COMMAND GO IS TO EXECUTE ANALYZER.

      SOURCE FILE    ===> TOKMK.FORT77

      INCLUDE FILE   ===> TOKMKI.FORT77
                     ===>

  TREE INFORMATION LIBRALY
       MEMBER NAME  ===> TOKMK     (XXXX/TEMP) XXXX : IF EXIST,THEN REPLACE.

        COMMENT     ===> TRANSPORT CODE

      EXECUTION     ===> BATCH (TSS/BATCH)
```

図5－6 画面C ANALYZER画面

この画面はANALYZERの実行に非常に時間がかかり,BATCHで処理する必要がある時,あるいはツリー情報ファイルに解析結果を保存する時に用いる。

(1) ソース・ファイル,インクルード・ファイルを入力する。

(2) ツリー情報ファイルにデータを出力する際のメンバ名とコメントを入力する。
'TEMP'と入力すると,ツリー情報はテンポラリ・ファイルに出力される。

(3) ANALYZERの実行をTSSで行うかBATCHで行うかを指定する。

(4) GOコマンドでANALYZERを実行する。

COMMAND ===> GO (enter)

BATCH処理を選択していれば,図5－7に示す様なJCLが自動的に作成される。

```
JZL0021 STOP 1
<<<  START  ANALYZER DATE:05/31/89 TIME:16:40:44  >>>
CREATE ##WORK.CNTL.
PLEASE SUBMIT THIS JCL !
***
```

↓ (enter)

```
EDIT --- J7867.##WORK.CNTL ---------------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CUR
****** ***************************** TOP OF DATA ***************V10L30********
000010 //JCLG JOB
000020 //JCLG EXEC JCLG
000030 //SYSIN DD DATA,DLM='++'
000040 // JUSER 11977867,SI.NAITOU,0019,06
000050   T.2 W.0 I.4 C.2
000060  OPTP PASSWORD=XXXXX
000070 // EXEC LMGO,LM=3859.Z2.NEWORG.PNM=ANALYZER
000080 //FT66F001 DD DUMMY
000100 //CMDIN    DD DUMMY
000110 //CMDOUT   DD SYSOUT=*
010000 //FT01F001 DD DISP=SHR,DSN=7867.TOKMK.TREE(TOKMK)
020000 //SYSIN DD *
020010  ORG(1),LIST(0)
030000  SOC(J7867.TOKMK.FORT77)
030010  INC(J7867.TOKMKI.FORT77)
040000  CMT(TRANSPORT CODE)
045000  ELM(-STREES)
050000 ++
060000 //
****** ***************************** BOTTOM OF DATA ****************************
```

図5－7 ANALYZER実行用のJCL例

5.5 画面D：DESIGNER(1)(編集するソース・ファイルの入力)

```
-----------------------------< DESIGNER DATA >--------------------------------
COMMAND ===> GO
<<<ANALYZER>>>                          COMMAND GO.COPY.CLEAR.BROWSE IS AVAILABLE.

 (0) LIBRARY ===> NO
                  (NO/YES)
 (1) SORC ===> TOKMK.FORT77
     INCL ===> TOKMKI.FORT77                    ===>

 (2) SORC ===>
     INCL ===>                                  ===>
 (3) SORC ===>
     INCL ===>                                  ===>
 (4) SORC ===>
     INCL ===>                                  ===>
 (5) SORC ===>
     INCL ===>                                  ===>
 (6) SORC ===>
     INCL ===>                                  ===>
 (7) SORC ===>
     INCL ===>                                  ===>
 (8) SORC ===>
     INCL ===>                                  ===>
```

図5－8　画面D　DESIGNER画面(1)

(1)　ライブラリを使用するかしないかを指定する。

(2)　使用するユーザ・ファイル名を入力する。ソース・ファイルは8個まで入力可能で，各々のソース対して2個までのインクルード・ファイルが指定できる。

(3)　コマンド欄で′GO′と入力し enter キーを押すと，

(a)　ツリー情報ファイルに当該情報がなければANALYZERを実行する。

(b)　当該情報があれば図5－9に示す画面に入って，このツリー情報を使うかどうかを選択する。選択後，コマンド欄で′GO′と入力し enter キーを押せばANALYZERが実行される。

ANALYZERの実行例を図5－10に示す。

```
----------------------< CONFIRM DESIGNER DATA 2.0  >-- LINE 000001 COL 001 080
COMMAND ===>                                          SCROLL ===> PAGE
<<< CONFIRM ANALYZER >>>                    COMMAND GO TO EXECUTE ANALYZER.
                                                   END KEY TO CANCELE ANALYZER.
SELECT  A (ANALYZER) / N (NO ANALYZE)
A/N     MEMBER   DATE      TIME  MODIFIED   SOURCE DATASET NAME
*       ******** ********  ***** ********** ************************************
→(N)    TOKMK    06/02/89  13:11 NO         J7867.TOKMK.FORT77
*** BOTTOM OF DATA ***
```

ツリー情報作成後ソースの変更がないことを意味する

図5－9　ツリー情報確認画面
あらかじめ作成してあるツリー情報を使う場合は，
新たにツリー解析は行わないということだから
′NO ANALYZE (N)′を指定する。

(4)　ツリー情報作成日付時間とソース・プログラムのディレクトリーに書かれている最終更新日付時間との比較によりMODIFIEDの欄のデータを出力している。
NO ANALYZEを指定する時には，ツリー情報がソース・ファイルに対応している事

を再度確認した方が良い。完成したソース・プログラムで変更する必要のないファイルに対してのみツリー情報を保存するべきである。

```
JZL002I STOP 1
     <<< INPUT DATA >>>
 ORG(1),LIST(0)
 SOC(J7867.TOKMK.FORT77)
 INC(J7867.TOKMKI.FORT77)
 CMT(DANALYZER SOC1)
 ELM(-●TREE●)
<< CRSDAT>>       - SUBROUTINE -
<< INTEQU>>       - SUBROUTINE -
<< MAIN   >>      - MAIN -
***** INFORMATION ***** PROGRAM NAME "MAIN" EXISTS ALREADY.
<< MAIN   >>      = MAIN =
***** INFORMATION ***** PROGRAM NAME "MAIN" EXISTS ALREADY.
<< MAIN   >>      - MAIN -
<< NBMAIN>>       - SUBROUTINE -
***** INFORMATION ***** PROGRAM NAME "NBMAIN" EXISTS ALREADY.
<< NBMAIN>>       - SUBROUTINE -
<< TRBAL  >>      - SUBROUTINE -
<< TRCOE  >>      - SUBROUTINE -
<< TRFUS  >>      - SUBROUTINE -
<< TRINP  >>      - SUBROUTINE -
***** INFORMATION ***** PROGRAM NAME "TRINP" EXISTS ALREADY.
<< TRINP  >>      - SUBROUTINE -
***
***** INFORMATION ***** SUBPROGRAM AVTEM    IS NOT DEFINED.
***** INFORMATION ***** SUBPROGRAM CKAI     IS NOT DEFINED.
***** INFORMATION ***** SUBPROGRAM DIF      IS NOT DEFINED.
***** INFORMATION ***** SUBPROGRAM PRNTV    IS NOT DEFINED.
***** INFORMATION ***** SUBPROGRAM PRNTS    IS NOT DEFINED.
     CREATE TREE INFORMATION : 'J7867.TOKMK.FORT77.●TREE●'
<<< FINISH ANALYZER DATE:05/31/89 TIME:16:18:30   >>>
***
```

図5－10　ANALYZERの実行例

5.6 画面E : DESIGNER(2)(起点ルーチン,コード・スペック名の入力)

```
--------------------------< DESIGNER DATA 2.1 >-----------------------------
COMMAND ===>
<<< DESIGNER >>>                          COMMAND GO IS TO EXECUTE DESIGNER.
    CODE SPEC NAME ===> TOKMK     (MEMBER NAME IS CODE SPEC DATASET)
           COMMENT ===> TRANSPORT CODE
    START  DATASET ===> 'J7867.TOKMK.FORT77(MAIN)'
    START PROGRAM  ===> MAIN
    CHANGE GLOBAL-PARAMETER ===> YES(YES/NO)
    CHANGE LOCAL-PARAMETER  ===> YES(YES/NO)
```

図5－11 画面E DESIGNER画面(2)

(1) コード・スペック名とコメントを入力する。コード・スペック名はコード・スペック・データをファイルに出力する際のメンバ名となる。

(2) スタート・ルーチンとそのデータセット名を入力する。この欄を空白のまま enter キーを押せば,図5－12に示す様なメイン・ルーチンのテーブルが表示されるので,これを見てスタート・ルーチンを決めることもできる。

```
--------------------------< START PROGRAM TABLE >----- LINE 000001 COL 001 080
COMMAND ===>                                            SCROLL ===> PAGE
S SELECT  B BROWSE
SELECT PROGRAM  MEMBER     DATASET NAME
*      ******   ********   ***************************************************
→Ⓢ     MAIN     MAIN       J7867.TOKMK.FORT77
       MAIN     MAINTEST   J7867.TOKMK.FORT77
       MAIN     MAIN2      J7867.TOKMK.FORT77
*** BOTTOM OF DATA ***
```

図5－12 スタート・ルーチン選択画面
SELECT欄で'B'を入力すればメンバの内容を見ることができる。

(3) Global変数の値を変更するかどうかを入力する。'YES'と入力して enter キーを押すと図5－13の画面に移り,変更する方法を選択する。(E：Edit画面で変更／D：DESIGNER実行時の変更) 選択後 PF3 キーで元の画面に戻ると図5－14で二重下線を引いたメッセージが表示され,Global変数の変更が行われることを示す。

```
--------------------------< EDIT GLOBAL PARAMETER >--- LINE 000001 COL 001 080
COMMAND ===>                                            SCROLL ===> PAGE
B BROWSE   E EDIT   D DESIGNER
SELECT STATUS   MEMBER   DATASET NAME
*      ******** ******** ***************************************************
→Ⓓ             COPTIN   J7867.TOKMKI.FORT77
*** BOTTOM OF DATA ***
```

図5－13 Global変数変更画面
'B' 見るだけ
'E' EDIT画面で変更
'D' DESIGNER実行時に変更

```
--------------------------< DESIGNER DATA 2.1 >------------------------------
COMMAND ===> GO

<<< DESIGNER >>>                          COMMAND GO IS TO EXECUTE DESIGNER.

    CODE SPEC NAME ===> TOKMK    (MEMBER NAME IN CODE SPEC DATASET)
           COMMENT ===> TRANSPORT CODE

    START  DATASET ===> 'J7867.TOKMK.FORT77(MAIN)'

    START PROGRAM  ===> MAIN

    CHANGE GLOBAL-PARAMETER ===> NO  (YES/NO)            SPECIFYED

    CHANGE LOCAL-PARAMETER  ===> YES (YES/NO)              指定済
```

図5－14　Global変数が変更されることを示すメッセージ

(4)　GOコマンドでDESIGNERが実行される。

COMMAND　===>　GO　(enter)

DESIGNERの実行例を図5－15に示す。

```
<<<  START  DESIGNER DATE:06/02/89 TIME:19:17:38 >>>

        <<< START NEW RUN DESIGNER >>>

           --- START DATA INPUT ---

--- LISTING OF TREE INFORMATION DATASET(S) ---
  1. J7867.TOKMK.FORT77.*TREE*

--- LISTING OF NEW GLOBAL SPEC  DATASET(S) ---
  1. J7867.TOKMKI.FORT77(COPTIN)


        ----  END OF DATA INPUT  ----

   1  MAIN     J7867.TOKMK.FORT77(MAIN)
         ? SAME SUB-PROGRAM NAME FOUND ((TRINP    ))
           1. J7867.TOKMK.FORT77(TRINP)
           2. J7867.TOKMK.FORT77(TRINPN)
          INPUT SELECT DATASET NUM ==>  2    [同一名ルーチンの選択]
***
   2  TRINP    J7867.TOKMK.FORT77(TRINPN)
         ? GLOBAL SPEC LNBI    =   2
         CMT..    LNBI = 0 : OHMIC PLASMA
         CMT..    LNBI = 1 : STIX SOL.  /  LNBI = 2 : OHMC CODE
          >INPUT NEW SPEC. ( LNBI    ) ==>  2
   3  NBMAIN  J7867.TOKMK.FORT77(NBMAIN2)
         ? GLOBAL SPEC LRF     =   0          [Global変数の変更]
          >INPUT NEW SPEC. ( LRF     ) ==>  0
         ? LOCAL  SPEC LDBG    =   1
         CMT..    LDBG : DEBUG INFORMATION ( OFF = 0, ON = 1 )
          >INPUT NEW SPEC. ( LDBG    ) ==>  0
                                              [Local変数の変更]
         ---  FINISH DESIGNER PROCESSING  ---

        -----  BLOCK DATA SUB PROGRAM SELECTION -----

  1.  BLKNBI.... J7867.TOKMK.FORT77(BLKNBI)
***
>INPUT DO YOU REQUIRE IT. ( YES/NOT ) ==>  Y    [ブロック・データの使用・不使用の選択]
     CREATE CODE SPEC FILE    : 'J7867.NORG.SPEC(TOKMK)'
<<<  FINISH DESIGNER DATE:06/02/89 TIME:19:18:58  >>>
***
```

図5－15　DESIGNERの実行例

5.7 画面F : ENGINEER(ソース・組み上げの場合)

```
----------------------------< ENGINEER DATA 3 >-------------------------------
COMMAND ===> GO
<<< ENGINEER >>>                          COMMAND GO IS TO EXECUTE ENGINEER.
                                                  BROWSE IS TO DISPLAY SPEC DATA.
    CODE SPEC NAME ---> TOKMK    (MEMBER NAME IN CODE SPEC DATASET)
      SOURCE FILE  ---> #TOKMK.FORT77
           NEW/OLD ---> NEW                       UNIT ---> TSSWK
      INCLUDE FILE ---> #TOKMKI.FORT77
           NEW/OLD ---> NEW                       UNIT ---> TSSWK
    RESTART OPTION ---> YES    (YES/NO)
         EXECUTION ---> TSS    (TSS/BATCH)
```

図5－16 画面F ENGINEER画面(ソース指定)
この画面に限り,前のデータは表示されない。
ただし,DESIGNERよりこの画面に入った
場合はコード・スペック名は自動的に入力される。

(1) コード・スペック名を入力する。
(2) ソース・ファイル名及びインクルード・ファイル名を入力する。
(3) リスタート・オプションの有無を入力する。
(4) ENGINEERの実行をTSSで行うかBATCHで行うかを指定する。
(5) BROWSEコマンドで,ツリー情報ファイルあるいはコード・スペック・ファイルの内容を見ることができる。

BROWSE TR enter ツリー情報ファイル
BROWSE SP enter コード・スペック・ファイル

これについては,5.14節参照の事。

(6) GOコマンドでENGINEERが実行される。

GOMMAND ===> GO enter

ENGINEERの実行例を図5－17に示す。

ATTRIBUTES画面でロード・モジュールの作成を指定した場合は画面F′が表示される。(図5－18参照)

```
<<<  START  ENGINEER DATE:05/31/89 TIME:16:25:28   >>>
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  E N G I N E E R   - +
I          TREE STRUCTURE FILE IS J7867.TOKMK.SPEC(TOKMK)                        I
I                             CREATED DATE 89/05/31 & TIME 16:22:28              I
I                                         VERSION : 01       LEVEL : 00          I
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  E N G I N E E R   - +
------ CREATE NEW SOURCE FILE ------
     # PROGRAM    REFERED MEMBER & DATASET NAME
     1  MAIN         MAIN       J7867.TOKMK.FORT77
     2  CRSDAT       CRSDAT     J7867.TOKMK.FORT77
     3  TRINP        TRINPN     J7867.TOKMK.FORT77
     4  NBMAIN       NBMAIN2    J7867.TOKMK.FORT77
     5  TRCOE        TRCOE      J7867.TOKMK.FORT77
     6  INTEQU       INTFAC     J7867.TOKMK.FORT77
     7  TRJUL        TRJUL      J7867.TOKMK.FORT77
     8  TRBAL        TRBAL      J7867.TOKMK.FORT77
     9  TROUT        TROUT      J7867.TOKMK.FORT77

    -- TOTAL -- 9 PROGRAM(S) & 739 LINES WERE WRITTEN.
------ CREATE NEW INCLUDE FILE ------
     #  GLOBAL MEMBER & REFERED DATASET NAME
     1         COPTIN         J7867.TOKMKI.FORT77
     #  OTHER INCLUDE MEMBER & REFERED DSN
***
     1         COMCRS         J7867.TOKMKI.FORT77
     2         COMAAA         J7867.TOKMKI.FORT77
     3         COMGEO         J7867.TOKMKI.FORT77
     4         COMEQU         J7867.TOKMKI.FORT77
     5         COMEQV         J7867.TOKMKI.FORT77
     6         COMBAL         J7867.TOKMKI.FORT77
     7         COMEQT         J7867.TOKMKI.FORT77
     8         COMTRN         J7867.TOKMKI.FORT77
     9         COMFUS         J7867.TOKMKI.FORT77
    10         COMNBI         J7867.TOKMKI.FORT77
    11         COMR2D         J7867.TOKMKI.FORT77
    12         COMCOM         J7867.TOKMKI.FORT77
    13         COMCNT         J7867.TOKMKI.FORT77
    14         COMSOR         J7867.TOKMKI.FORT77
    15         COMRFH         J7867.TOKMKI.FORT77
    16         COMNTR         J7867.TOKMKI.FORT77

    -- TOTAL -- 17 INCLUDE(S) & 202 LINES WERE WRITTEN.
------ CREATE DUMMY ROUTINE ------
                                                         23 LINES WERE WRITTEN.
------ NO CREATE RESTART ROUTINE ------
     CREATE NEW SOURCE  FILE(PS) : ●●NEWSOC.ENGNTSS
     CREATE NEW INCLUDE FILE(PO) : #TOKMKI.FORT77
***
<<<  FINISH  ENGINEER DATE:05/31/89 TIME:16:25:43   >>>
START VIVAPO(PSTOPO) ●●NEWSOC.ENGNTSS TO #TOKMK.FORT77
001.MEMBER *MAIN     * (MAIN PROGRAM   )     57 CARDS.
002.MEMBER *CRSDAT   * (SUBROUTINE     )     32 CARDS.
003.MEMBER *TRINP    * (SUBROUTINE     )     51 CARDS.
004.MEMBER *NBMAIN   * (SUBROUTINE     )     22 CARDS.
005.MEMBER *TRCOE    * (SUBROUTINE     )    118 CARDS.
006.MEMBER *INTEQU   * (SUBROUTINE     )    202 CARDS.
007.MEMBER *TRJUL    * (SUBROUTINE     )     36 CARDS.
008.MEMBER *TRBAL    * (SUBROUTINE     )     23 CARDS.
009.MEMBER *TROUT    * (SUBROUTINE     )    198 CARDS.
010.MEMBER *DUMMY    * (SUBROUTINE     )     23 CARDS.


       TOTAL CARD DECK =          762
       ------------------------------ THANK YOU !
<<<  UPDATE CODE SPEC TOKMK.SPEC (TOKMK) >>>
***
```

図5－17　ENGINEERの実行例

5.8 画面F′ : ENGINEER(ロード・モジュール作成)

```
----------------------------< ENGINEER DATA 3 >-------------------------------
COMMAND ===> GO
<<< ENGINEER >>>                          COMMAND GO IS TO EXECUTE ENGINEER.
                                                BROWSE IS TO DISPLAY SPEC DATA.
    CODE SPEC NAME ===> TOKMK    (MEMBER NAME IN CODE SPEC DATASET)
      LOAD MODULE  ===> #TOKMK.LOAD
           MEMBER  ===> TEMPNAME
           NEW/OLD ===> NEW                        UNIT ===> TSSWK
    RESTART OPTION ===> YES    (YES/NO)
         EXECUTION ===> TSS    (TSS/BATCH)
    FORT77 & LINK  OPTIONS ===> YES (YES/NO)
```

図5－18 画面F′ ENGINEER画面(ロード・モジュール指定)

(1) コード・スペック名を入力する。

(2) ロード・モジュール名を入力する。

(3) リスタート・オプションの有無を入力する。

(4) ENGINEERの実行をTSSで行うかBATCHで行うかを指定する。

(5) コンパイル及びリンク時のオプションの有無を入力する。

(6) GOコマンドでENGINEERが実行される。

GOMMAND ===> GO (enter)

コンパイル及びリンク時のオプションが有る場合には,ENGINEERの実行時に図5－19に示す様な画面が表示されるので,ここで,必要なオプションを指定する。

```
-------------------< FORT77/LINK OPTION PARAMETER >---------------------------
COMMAND ===> GO
<<<FORT77/LINK>>>                         COMMAND  GO TO EXECUTE COMPILE & LINK.
                                                   END KEY TO CANCEL.
   FORT77 PARAMETERS :
     BYNAME/NOBYNAME   ===> BYNAME
     FLAG(I/W/E/S)     ===> W
     OTHER OPTIONS     ===>

CAUTION:  OPTIONS OBJ, ELM, AND INCLUDE ARE GENERATED AUTOMATICALLY BY NEWORG.

   LINK PARAMETERS (CALL LIBRARYS)
              PRVLIB1 ===> 'J9952.DACCESS.LOAD'
              PRVLIB2 ===>
              PRVLIB3 ===>
              GRLIB   ===>       (PNL/PLT/PTS/COM/...OTHER GRAPHICKS)
              SSLA    ===> JSSL  (JSSL/SSL/SSL2)
              SSLB    ===> SSL   (JSSL/SSL/SSL2)
              SSLC    ===> SSL2  (JSSL/SSL/SSL2)
     OTHER OPTIONS    ===> LET

CAUTION:  OPTIONS LOAD, LIBDD, AND  FORTLIB GENERATED AUTOMATICALLY BY NEWORG.
```

図5－19 コンパイル&リンクオプション

コンパイル時のオプションOPTは2を標準としている。

5.9 画面G : ロード・モジュールUPDATE

```
-------------------------------< UP DATE  5  >--------------------------------
COMMAND ===> GO

<<< UPDATE LOAD MODULE >>>              COMMAND GO IS TO UPDATE LOAD MODULE.

    CODE SPEC NAME ===> EQDD    (MEMBER NAME IN CODE SPEC DATASET)

      LOAD MODULE  ===> EQDD.LOAD
           MEMBER  ===> TEMPNAME

      SOURCE FILE  ===> '2855.¥EQDD.FORT77'
            (IF PO FILE;SPECIFY MEMBER NAMES BELLOW OR
             IF PS FILE;SPECIFY SUB-PROGRAM NAMES BELLOW:)
              NAME ===> BADATE    ===>            ===>            ===>
              NAME ===>           ===>            ===>            ===>

           EXPAND  ===> NO  (YES/NO)

      INCLUDE FILE ===>
         (LIB)     ===>
                   ===>

      FORT77 & LINK OPTION ===> NO  (YES/NO)
```

図5－20 画面G UPDATE画面

この画面はロード・モジュールのUPDATEを行う。ソースにエラーが有った場合，PFDで修正を行った後，UPDATEするメンバを指定する。

NEWORGで比較的時間がかかるのはENGINEERとコンパイルである。したがってソースのエラーを修正した後は，このUPDATE画面を利用してロード・モジュールのUPDATEを行えば短時間に処理でき便利である。ただし，修正したロード・モジュールのコード・スペック・データは保存されないので注意を要する。ロード・モジュールが完成したら，もう一度DESIGNERからやり直し，ロード・モジュールとコード・スペック・データの対応関係をはっきりさせておくのが良いと思われる。

(1) ロード・モジュールの作成に用いたコード・スペック名を入力する。
　Global変数及びLocal変数の情報はこのコード・スペック・データを用いる。
(2) UPDATEするロード・モジュール名を入力する。
(3) ソース・ファイル名と，その中でコンパイルするメンバ名(PSファイルの場合はサブルーチン名)を入力する。
(4) 修正モジュール以降のルーチンを展開するか，しないかを指定する。
(5) インクルード・ファイル名を入力する。
(6) コンパイル及びリンク時のオプションの有無を入力する。
(7) GOコマンドで実行を開始する。
　COMMAND ===> GO (enter)

図5－21に実行例を示す。

```
<<<  START  ANALYZER DATE:06/02/89 TIME:10:48:21   >>>
     <<< INPUT DATA >>>
 ORG(1),LIST(0)
 SOC(J2855.¥EQDD.FORT77)
 CMT(UPDATE ANALYZER)
 ELM(-*TREE*,BADATE,.....)
<< BADATE>>      - SUBROUTINE -
     CREATE TREE INFORMATION : 'J2855.¥EQDD.FORT77.*TREE*'
<<<  FINISH ANALYZER DATE:06/02/89 TIME:10:48:30   >>>
<<<  START  RE-EDIT  DATE:06/02/89 TIME:10:48:31   >>>

         <<< START RE-EDIT DESIGNER >>>

            --- START DATA INPUT ---

--- LISTING OF TREE INFORMATION DATASET(S) ---
  1. J2855.¥EQDD.FORT77.*TREE*

         ----  END OF DATA INPUT  ----
***

   1  BADATE  J2855.¥EQDD.FORT77(BADATE) - 指定メンバ

         ---  FINISH DESIGNER PROCESSING  ---

     CREATE CODE SPEC FILE    : 'J7867.**REEDIT.EQDD'
<<<  FINISH DESIGNERDATE:06/02/89 TIME:10:48:54   >>>
<<<  START  ENGINEER DATE:06/02/89 TIME:10:48:58   >>>
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - -   E N G I N E E R   - +
I          TREE STRUCTURE FILE IS J7867.**REEDIT.EQDD                          I
I                          CREATED DATE 89/06/02 & TIME 10:28:21               I
I                                VERSION : 01        LEVEL : 00                I
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - -   E N G I N E E R   - +
------ CREATE NEW SOURCE FILE ------
     # PROGRAM   REFERED MEMBER & DATASET NAME
     1  BADATE     BADATE   J2855.¥EQDD.FORT77

    -- TOTAL -- 1 PROGRAM(S) & 80 LINES WERE WRITTEN.
          NO GLOBAL MEMBER
     #  OTHER INCLUDE MEMBER & REFERED DSN
          NO INCLUDE MEMBER

    -- TOTAL -- 0 INCLUDE(S) & 0 LINES WERE WRITTEN.
***
------ CREATE DUMMY ROUTINE ------
                                                   5 LINES WERE WRITTEN.
------ NO CREATE RESTART ROUTINE ------
     CREATE NEW SOURCE  FILE(PS) : **UPD.FORT77
     CREATE NEW INCLUDE FILE(PO) : **UPDI.FORT77
<<<  FINISH  ENGINEER DATE:06/02/89 TIME:10:49:39   >>>
FORT77   'J7867.**UPD.FORT77' INC('J7867.**UPDI.FORT77') OBJ('J7867.**UPD.FORT7
7.OBJ') BYNAME FLAG(W)
FORTRAN 77 COMPILER ENTERED                                   新ソースの作成
END OF COMPILATION
ALLOC DD(LKEDLIB) DSN(    'SYS9.JSSL.LOAD' 'SYS9.SSL.LOAD' 'SYS9.SSL2.LOAD') SH
R REU
LINK ('J7867.**UPD.FORT77.OBJ' *) LO('J7867.EQDD.LOAD') LIBDD(LKEDLIB) FORTLIB
LET
 INCLUDE OLDLM(TEMPNAME)                                      ロード・モジュールの
 ENTRY   MAIN                                                     UPDATE
 NAME    TEMPNAME(R)
END OF CONTROL STATEMENTS
** MEMBER NAME **  TEMPNAME NOW REPLACED IN LIBRARY.
***
```

ANALYZER

DESIGNER

ENGINEER

図5－21　UPDATEの実行例

5.10 画面H ： NEWS

```
----------------------< NEWORG NEWS    89/05/12    >--------------------------
     NEWS FOR NEWORG USER :
         ALL OF NEWORG FUNCTION IS AVAILABLE TODAY.
               FUNCTION : ANALYZER
                          DESIGNER
                          ENGINEER
                             &
                          UPDATE OPTION
         OTHER PFD FUNCTIONS : ATTRIBUTE
                               BROWSE
                               EDIT
                               UTILITY
                               COMMAND
         SPECIAL APPLICATIONS : VIVAPO
                                ANALYSIS 77
       IF YOU HAVE ANY PROBREM ,PLEASE CALL UP ME.    NAKA  (3342)    SHIMIZU
                                                      TOKYO (230)4481 TAKIGAWA
```

図5－22 画面H NEWS画面

NEWORGユーザのために最新の情報(バージョン・アップ，使用時の注意事項等)を伝える。

5.11 画面Ｉ ： ツリー情報，コード・スペック・データ管理

```
----------------------------<  LIST & DATA FILE >------ LINE 000001 COL 001 080
COMMAND ===>                                                  SCROLL ===> PAGE
LX LIST DATASET SYSPRINT('X')  B BROWSE DATASET  K KEEP DATASET.
SELECT STATUS TYPE      DATASET NAME
**     ****** *******   ***************************************************
→K             TREE      'J2855.¥EQDD.FORT77.●TREE●'
               SPEC      'J7867.●●EDIT.●SPEC●'
               SPEC      'J7867.●●REEDIT.EQDD'
*** BOTTOM OF DATA ***
```

図5－23 画面Ｉ データ・ファイル管理画面

NEWORG使用時に作成されたデータ・ファイルのリスト出力，保存を行う。

SELECT欄で

LC	enter	Ｃクラスにリスト出力する
B	enter	ファイルの内容を見る
K	enter	ファイルを保存する
：		

と入力する。

例えば ' J2855.¥EQDD.FORT77 ' のツリー情報はテンポラリ・ファイルであってNEWORG終了時に消去されるが，SELECT欄で ' K ' を入力しておけば保存され，次回のNEWORGにおいてそれを用いることができる。ただしTSSWK上に作成されているので，その日一日だけである。

5.12 画面J ： ツリー情報／コード・スペックのフォーマットデータ

```
--------------------< NEWORG : FILE FORMAT COMMENT >------------------------
OPTION --->

<<< SELECT FILE >>>

    1   TREE        - TREE INFORMATION FILE
    2   CODE SPEC   - CODE SPEC DATA

    X   EXIT        - RETURN TO THE NEWORG PRIMARY MENU
```

図5－24 画面J COMMENT画面

ツリー情報ファイルあるいはコード・スペック・ファイルのフォーマットを表示する。

例えばツリー情報ファイルのフォーマットが知りたい時は

OPTION ===> 1 enter

とする。このようにして表示されたフォーマットの一部を図5－25に示す。

```
 BROWSE - J3859.Z2.NEWORG.MSGS(COMNTTRI) - 01.00 ------- LINE 00000 COLS 001 080
 COMMAND --->                                              SCROLL ---> CUR
******************************* TOP OF DATA ********************-CAPS ON-***
*-----------------------------------------------------------------------00000100
*   O R G   /   A N A L Y Z E R                                         00000300
*                                   LEVEL  1.1 (1986/10/17)             00000500
*   <<  TREE INFORMATION FILE  >>                                       00000700
*-----------------------------------------------------------------------00000900
*                                                                       00001000
*SYSTEM                                                                 00001100
*       + FILE-ID        ..DATE..       ..TIME..        VRS      LVL...COM00001200
*                                                                       00001300
*SOURCE                                                                 00001400
*          DATASET NAME                                                 00001500
*                                                                       00001600
*INCLUDE                                                                00001700
*          DATASET NAME                                                 00001800
*                                                                       00001900
*MAIN                                                                   00002000
*SUBPROG                                                                00002100
*BLOCKD                                                                 00002200
*          PGNAME     : SUB.PROGRAM NAME                               00002300
*          MEMBER     : MEMBER NAME                                    00002400
*          #SOC       : SOURCE DATASET NUMBER                          00002500
```

図5－25 ツリー情報ファイルのフォーマットの一部

5.13 画面K ： ライブラリ・コピー画面

```
--------------------------<  COPY LIBRARY MEMBER  >-----------------------------
COMMAND ===> GO

<<< COPY >>>                                COMMAND GO IS TO EXECUTE COPY.

      LIBRARY        ===> 'J3859.LIBJT300.FORT77'
        SOU/INC      ===> INC

      USER FILE      ===> TOKMK2.FORT77
        NEW/OLD      ===> NEW UNIT ===> TDS

         MEMBER      ===>            ===>            ===>            ===>
                     ===>            ===>            ===>            ===>

      ( IF INC SPECIFY, OPTION SIZE DOUBLE ARE COPYED AUTOMATICALLY.)
```

図5－26 画面K ライブラリ・コピー画面

この画面は，ライブラリのモジュールをユーザ・ファイルにコピーして，それを修正して使うといった時に用いる。特に#SYSGEN，#SIZE及びDOUBLEの3モジュールは，1Dライブラリのモデルを選択する上で必ず必要となるもので，モジュール名を入力しなくても自動的にコピーできるようになっている。

(1) 使用するライブラリ名を入力する。
(2) ソース又はインクルードを指定する。インクルードを指定すると，#SYSGEN，#SIZE，DOUBLEの3モジュールが自動的にコピーされる。
(3) ユーザ・ファイル名を入力する。
(4) コピーするモジュール名を入力する。

5.14 補足説明 (BROWSEコマンドについて)

NEWORGのCOMMAND欄ではTSSコマンドの他にBROWSEコマンドが利用できる。これはツリー情報ファイル，あるいはコード・スペック・ファイルの内容を表示するもので

BROWSE TR (enter) ツリー情報ファイル

BROWSE SP (enter) コード・スペック・ファイル

のように入力する。

コード・スペック・ファイルをBROWSEした時の例を図5－27に示す。

```
-------------------------<  CODE SPEC LIBRARY  >-------- LINE 000001 COL 001 080
COMMAND ===>                                                  SCROLL ===> PAGE
<<< BROWSE SPEC >>>
     CODE SPEC : NORG.SPEC
B MEMBER    MODE  DATE      TIME  DATASET NAME
* ******** ***** ******** ***** ********************************************
  APORO     BATCH 11/18/88 19:52 #APORD.LOAD
  EQDD      TSS   01/23/89 14:37 #EQDD.LOAD
→ⓑ TOKMK    TSS   06/02/89 19:20 #TOKMK.FORT77
  TOKMKT    TSS   06/01/89 11:26 #TOKMKT.FORT77
*** BOTTOM OF DATA ***
```

図5－27 コード・スペック・ファイルのBROWSE
メンバ・リストの中から自分が見たいメンバを
選択すれば，その内容が表示される。

NEWORGを用いて組み上げたソースあるいはロード・モジュールは，作成日付や条件(Global変数やLocal変数の値，使用ルーチン名等)が，すべてコード・スペック・ファイルに記述されている。従ってこの画面を利用すれば，ソースやロード・モジュールの管理が容易に行なえる。

5.15 補足説明 (作業用ファイル)

以下に示す名前のデータセットを作業用ファイルとして,自動的に作成,消去される。

@@WORK.DATA
@@WORK.CNTL
@@ENGN.WORK
ソース・ファイル名.@ANAL@
ソース・ファイル名.@TREE@
コード・スペック・ファイル名.スペック名
コード・スペック・ファイル名.スペック名.@DESN@
@@BLNK1.DATA
@@BLNK2.DATA
@@DEBUG.DATA
コード・スペック・ファイル名.@ENGN@
@@NEWSOC.FORT77
@@NEWSOC.FORTY77.OBJ
@@NEWINC.FORT77
@@REEDIT.スペック名
@@UPD.FORT77
@@UPD.FORT77.OBJ
@@UPDI.FORT77

6. 使　用　例

6.1 ユーザ・プログラム

輸送解析を行うユーザプログラム

'J7867.TOKMK.FORT77'　　‥‥　ソースファイル

'J7867.TOKMKI.FORT77'　　‥‥　インクルードファイル

を用いて，NBI加熱のシミュレーションコードを作成する。

NEWORG起動

```
 READY
NORG  enter
+-----------------------------------------------+
+  DID YOU LOGON WITH SIZE(3072) KB MEMORY ?   +
+-----------------------------------------------+
***
```

⇩

PRIMARY　MENU画面

```
----------------------< NEWORG PRIMARY OPTION MENU >---------------------------
OPTION ---> 2  enter
                                                          USERID    - J7867
<<< NEWOGR >>> ---------------------------------------------TIME      - 14:08
                                                          TERMINAL  - F9526
  0   ATTRIBUTES    - DEFINE NEWORG PARAMETERS            PF KEYS   - 24
  1   ANALYZER      - ANALYSIS OF SOURCE PROGRAM
  2   DESIGNER      - SPECIFICATION OF CALCULATION MODEL
  3   ENGINEER      - ORGANIZATION OF PROGRAM CODE
```

⇩

DESIGNER画面(1)

```
----------------------------< DESIGNER DATA >-------------------------------
COMMAND ---> GO  enter
<<<ANALYZER>>>                        COMMAND GO.COPY.CLEAR.BROWSE IS AVAILABLE.

 (0) LIBRARY ---> NO
                  (NO/YES)
 (1) SORC ---> TOKMK.FORT77
     INCL ---> TOKMKI.FORT77                    --->

 (2) SORC --->
```

⇩　ANALYZER実行

DESIGNER画面(2)

```
--------------------------< DESIGNER DATA 2.1 >-----------------------------
COMMAND ---> GO  enter

<<< DESIGNER >>>                        COMMAND GO IS TO EXECUTE DESIGNER.

    CODE SPEC NAME ---> TOKMK     (MEMBER NAME IN CODE SPEC DATASET)
           COMMENT ---> TRANSPORT CODE

    START  DATASET ---> 'J7867.TOKMK.FORT77(MAIN)'

    START PROGRAM  ---> MAIN
```

⇩　DESIGNER実行

同一名サブルーチンの選択

```
? SAME SUB-PROGRAM NAME FOUND ((TRINP    ))
  1. J7867.TOKMK.FORT77(TRINP)
  2. J7867.TOKMK.FORT77(TRINPN)

 INPUT SELECT DATASET NUM ---> 2 enter
```

⇩

Global変数の値を決定

```
? GLOBAL SPEC LNBI    =   2
CMT..    LNBI = 0 : OHMIC PLASMA
CMT..    LNBI = 1 : STIX SOL.  / LNBI = 2 : OFMC CODE
 >INPUT NEW SPEC. ( LNBI    ) ---> 2 enter
```

⇩

Local変数の値を決定

```
? LOCAL  SPEC LDBG    =   1
CMT..    LDBG : DEBUG INFORMATION  ( OFF = 0, ON = 1 )
 >INPUT NEW SPEC. ( LDBG    ) ---> 0 enter
```

⇩

ENGINEER画面

```
----------------------------< ENGINEER DATA 3  >------------------------------
COMMAND ===> GO enter
<<< ENGINEER >>>                          COMMAND GO IS TO EXECUTE ENGINEER.
                                                  BROWSE IS TO DISPLAY SPEC DATA.

    CODE SPEC NAME ---> TOKMK    (MEMBER NAME IN CODE SPEC DATASET)

      SOURCE FILE  ---> #TOKMK.FORT77

           NEW/OLD ---> NEW                       UNIT ---> TSSWK

      INCLUDE FILE ---> #TOKMKI.FORT77
```

⇩ ENGINEER実行

ソースプログラム完成

これにより，ソースライン(図6－1)は不要なIF文等が削除されて図6－2に示したように修正される。又，Global変数の値も変更されて図6－3のようになる。

```
C               TRAAMAIN                LEVEL=27        DATE=81.12.10
C-----------------------------------------------------------------------
C       MAIN PROGRAM OF TRANSPORT                                     JAERI
C-----------------------------------------------------------------------
*INCLUDE COPTIN,FIXED,NOINSOURCE
*INCLUDE COMAAA,FIXED,NOINSOURCE
*INCLUDE COMEQT,FIXED,NOINSOURCE
CORG  INCLUDE COMTRN,DEN(DEN1)
C-----------------------------------------------------------------------
COMM(LDBG)  LDBG : DEBUG INFORMATION ( OFF = 0, ON = 1 )
      PARAMETER ( LDBG = 1 )
C-----------------------------------------------------------------------
      DIMENSION DENMAX(0:INDM)
C-----------------------------------------------------------------------
C-----INPUT DATA
C-----------------------------------------------------------------------
      CALL CRSDAT
      CALL TRINP
      IF(LNBI.GT.0) CALL NBMAIN
      IF(LRF .GT.0) CALL RFMAIN
C-----------------------------------------------------------------------
C-----STORE
C-----------------------------------------------------------------------
      DO 105 NI=0,NION
      DENMAX(NI)=0.
      DO 100 N=1,NT
      IF(DENMAX(NI).LT.DEN1(N,NI))DENMAX(NI)=DEN1(N,NI)
 100  CONTINUE
 105  CONTINUE
C
C::DEBUG WRITE
      IF (LDBG .NE. 0)  THEN
        WRITE (6,*) 'DEN1(CENTER) = ',DEN1(1,1),DEN1(1,2)
      END IF
C-----------------------------------------------------------------------
C-----SOLVE TRANSPORT EQS.
C-----------------------------------------------------------------------
      CALL TRCOE
      CALL INT2DP
      CALL TRJUL
      IF(LFUS.GT.0) CALL TRFUS
      IF(LBAL.GT.0) CALL TRBAL
C-----
      IF(LNBI.EQ.0 .AND. LRF.EQ.0) THEN
        CALL TRSVN
      ELSE IF(LNBI.GT.0 .AND. LRF.GT.0) THEN
        CALL TRSVD
      ELSE
        CALL TRSVS
      END IF
C-----------------------------------------------------------------------
C-----OUTPUT DATA
C-----------------------------------------------------------------------
      CALL TROUT
      IF(LPLOT.EQ.1) THEN
        CALL NLPLOT
      END IF
C=======================================================================
      STOP
      END
```

図6－1　オリジナルのソース・プログラム(メイン・ルーチン)

```
C                TRAAMAIN              LEVEL=27        DATE=81.12.10
C-----------------------------------------------------------------------
C      MAIN PROGRAM OF TRANSPORT                                      JAERI
C-----------------------------------------------------------------------
*INCLUDE COPTIN,FIXED,NOINSOURCE
*INCLUDE COMAAA,FIXED,NOINSOURCE
*INCLUDE COMEQT,FIXED,NOINSOURCE
CORG   INCLUDE COMTRN,DEN(DEN1)
       COMMON/TRNO/NRO,NROM,RO(ITDM),ROH(ITDM)
      >            ,NION,MION,FMASS(0:INDM),FCHAG(0:INDM)
      >            ,DEN1(ITDM,0:INDM),PRE(ITDM,0:INDM),TEM(ITDM,0:INDM)
C-----------------------------------------------------------------------
COMM(LDBG)  LDBG : DEBUG INFORMATION ( OFF = 0, ON = 1 )
       PARAMETER ( LDBG    =              0 )
C-----------------------------------------------------------------------
       DIMENSION DENMAX(0:INDM)
C-----------------------------------------------------------------------
C-----INPUT DATA
C-----------------------------------------------------------------------
       CALL CRSDAT
       CALL TRINP
                        CALL NBMAIN
C-----------------------------------------------------------------------
C-----STORE
C-----------------------------------------------------------------------
       DO 105 NI=0,NION
       DENMAX(NI)=0.
       DO 100 N=1,NT
       IF(DENMAX(NI).LT.DEN1(N,NI))DENMAX(NI)=DEN1(N,NI)
 100   CONTINUE
 105   CONTINUE
C
C::DEBUG WRITE
C-----------------------------------------------------------------------
C-----SOLVE TRANSPORT EQS.
C-----------------------------------------------------------------------
       CALL TRCOE
       CALL INT2DP
       CALL TRJUL
C-----
         CALL TRSVS
C-----------------------------------------------------------------------
C-----OUTPUT DATA
C-----------------------------------------------------------------------
       CALL TROUT
         CALL NLPLOT
C-----------------------------------------------------------------------
       STOP
       END
```

図6－2　組み上げたソースプログラム（メインルーチン）

図5－1と比較すると，Global変数及びLocal変数を用いたIF文はなくなり，コモン・ブロックTRN0の中のコモン変数DENがDEN1にリネイムされた後に展開されている事に注意。

```
      PARAMETER (LBAL=1, LNBI=2, LRF=0, LFUS=0, LPLOT=1)
C
C-----------------------------------------------------------------
C::OPTION FLAG
COMM(LNBI)  LNBI = 0 : OHMIC PLASMA
COMM(LNBI)  LNBI = 1 : STIX SOL.  / LNBI = 2 : OFMC CODE
C-----------------------------------------------------------------
```

（オリジナル）

```
      PARAMETER ( LBAL   =          0 )
      PARAMETER ( LFUS   =          0 )
      PARAMETER ( LNBI   =          2 )
      PARAMETER ( LPLOT  =          0 )
      PARAMETER ( LRF    =          0 )
C
C-----------------------------------------------------------------
C::OPTION FLAG
COMM(LNBI)  LNBI = 0 : OHMIC PLASMA
COMM(LNBI)  LNBI = 1 : STIX SOL.  /  LNBI = 2 : OFMC CODE
```

（組み上げ後）

図6－3　インクルード・ファイル(COPTIN)
Global変数の値はDESIGNERで入力した値に置き換えられる。

サブルーチンNBMAINは図6－4に示すように，同名のものが2つあるが，プログラムの中に記述してあるLIBCND文の条件式に従って，この例ではメンバ名NBMAIN2のルーチンが採用されることになる。

```
      SUBROUTINE NBMAIN
*INCLUDE COPTIN,FIXED,NOINSOURCE
*INCLUDE COMSTX,FIXED,NOINSOURCE
C-----------------------------------------------
C::    NBI HEATING ( STIX SOLUTION )
CORG  LIBCND LNBI.EQ.1
C-----------------------------------------------
      IF(STEP.EQ.0) CALL NBINI
      CALL NBINJ
      CALL NBSTIX
      RETURN
      END
```

メンバNBMAINの内容（ LNBI＝1のとき採用される ）

```
      SUBROUTINE NBMAIN
*INCLUDE COPTIN,FIXED,NOINSOURCE
*INCLUDE COMNBI,FIXED,NOINSOURCE
C-----------------------------------------------
C::     NBI HEATING   ( OFMC CODE )
CORG  LIBCND LNBI.EQ.2
C-----------------------------------------------
      IF(STEP.EQ.0) CALL NBINT
      DO 100 J=1,JMAX
      TE(J)=FINT(NRO,RO,TEM(I,0),RB(J))
  100 NE(J)=FINT(NRO,RO,DEN(I,0),RB(J))
      CALL NBINJC
      CALL NBOFMC
      DO 200 I=1,NRO
      WE(I)=FINT(JMAX,RB,PBE,RO(I))
  200 WI(I)=FINT(JMAX,RB,PBI,RO(I))
      RETURN
      END
```

メンバNBMAIN2の内容（ LNBI＝2のとき採用される ）

図6－4　サブルーチンNBMAIN

6.2 ライブラリ・システム

1Dトカマク輸送コードの最新版ライブラリ

'J3859.LIBJT300.FORT77'

の中からNBI加熱のシミュレーション・コードを以下の条件で作成する。

H, NBI(2), 倍精度, history, restart

手順は,6.1のユーザ・プログラムの場合と大体同じなので,違う部分だけを記述する。

まず,ATTRIBUTES画面(図5－3参照)でライブラリを指定する。

```
--------------------------< NEWORG PARAMETER >------------------------------
COMMAND ---> enter

<<< ATTRIBUTES >>>                    PRESS ENTER KEY TO SET PARAMETERS.
                                      PRESS END KEY TO RETURN PRIMARY MENU.
   PASSWORD            --->
   LIBRARY  (NO/YES) ---> YES
```

⇩

'J3859.LIBJT300.FORT77'をセット

```
------------------------< LIBRARY DATASET TABLE >---- LINE 000001 COL 001 080
COMMAND ---> enter                                          SCROLL ---> PAGE

<<< SELECT LIBRARY >>>
S - SELECT
SELECT DATASET NAME                          OPT COMMENT
*      ****************************          *** ************************************
→5     'J3859.LIBJT300.FORT77'                1   1D-TRANS  05.10.89
       'J3859.LIBJT100.FORT77'                1   1D-TRANS  ADD SCOOP MODULE
       'J3859.LIBJT80.FORT77'                 1   1D-TRANS  OLD-VERSION
       'J3859.LIB2D.FORT77'                   0   1.5D-TRANS
*** BOTTOM OF DATA ***
```

⇩

COPY画面(図5－27参照)で#SYSGEN, #SIZE及びDOUBLEの3メンバーをコピーする。

```
--------------------------< COPY LIBRARY MEMBER >---------------------------
COMMAND ---> GO enter

<<< COPY >>>                          COMMAND GO IS TO EXECUTE COPY.

      LIBRARY        ===> 'J3859.LIBJT300.FORT77'
        SOU/INC      ===> INC

      USER FILE      ---> #TYPE1.FORT77
        NEW/OLD      ---> NEW UNIT ---> TDS
```

⇩

DESIGNER画面(1)

```
----------------------------< DESIGNER DATA >--------------------------------
COMMAND ===> GO
<<<ANALYZER>>>
                                        COMMAND GO,COPY,CLEAR,BROWSE IS AVAILABLE.
 (0) LIBRARY ===> 'J3859.LIBJT300.FORT77'
                  (NO/YES)
 (1) SORC ===>
     INCL ===> #TYPE1.FORT77                    ===>
                 先に作成したユーザーファイル
 (2) SORC ===>
     INCL ===>                                  ===>
```

⇩

DESIGNER画面(図5－9参照)でスタート・ルーチンを指定する。

```
--------------------------< DESIGNER DATA 2.1 >-----------------------------
COMMAND ===>  enter

<<< DESIGNER >>>                        COMMAND GO IS TO EXECUTE DESIGNER.

    CODE SPEC NAME ===> TYPE1    (MEMBER NAME IS CODE SPEC DATASET)
           COMMENT ===> TYPE1 ( H, NBI(2), DOUBLE, HISTORY,RESTART )

    START  DATASET ===>

    START PROGRAM  ===>
```

⇩

スタート・ルーチン決定

```
--------------------------< START PROGRAM TABLE >-----LINE 000001 COL 001 080
COMMAND ===>  enter                                      SCROLL ===> PAGE
S SELECT  B BROWSE
SELECT PROGRAM  MEMBER      DATASET NAME
*      ******   ********   ***************************************************
→Ⓢ     MAIN     YMAIN      J3859.Z2.LIB300.FORT77 (ｼﾐｭﾚｰｼｮﾝｺｰﾄﾞ)
       MAIN     YMAIN2     J3859.Z2.LIB300.FORT77 (SCOOPｺｰﾄﾞ)
*** BOTTOM OF DATA ***
```

⇩

#SYSGEN, #SIZE及びDOUBLEの修正をEDITで行う。

```
--------------------------< DESIGNER DATA 2.1 >-----------------------------
COMMAND ===>  enter

<<< DESIGNER >>>                        COMMAND GO IS TO EXECUTE DESIGNER.

    CODE SPEC NAME ===> TYPE1    (MEMBER NAME IS CODE SPEC DATASET)
           COMMENT ===> TYPE1 ( H, NBI(2), DOUBLE, HISTORY,RESTART)

    START  DATASET ===> 'J3859.Z2.LIB300.FORT77(YMAIN)'

    START PROGRAM  ===> MAIN

    CHANGE GLOBAL-PARAMETER ===> YES (YES/NO)

    CHANGE LOCAL-PARAMETER  ===>     (YES/NO)
```

1Dライブラリーは DESIGNER でG変数の変更はできないので EDIT で修正する。

```
-------------------------< EDIT GLOBAL PARAMETER >--- LINE 000001 COL 001 080
COMMAND ===>  enter                                        SCROLL ===> PAGE
B BROWSE   E EDIT   D DESIGNER
SELECT STATUS   MEMBER    DATASET NAME
*      ******** ******** ************************************************
→E              #SYSGEN   J7867.#TYPE1.FORT77
                #SIZE     J7867.#TYPE1.FORT77
                DOUBLE    J7867.#TYPE1.FORT77
*** BOTTOM OF DATA ***
```

EDITで修正 ⇩

#SYSGEN : モデル仕様

```
EDIT --- J3859.Z2.LIB300I.FORT77(#SYSGEN)------------------- COLUMNS 001 072
COMMAND ===>                                                SCROLL ===> CUR
****** ***************************** TOP OF DATA ***************V10L30********
000001 *SYSGEN #SYSGEN.LIBJT300  LIBRARY SYSTEM REFERENCE SYSGEN DATA
000002 C: TYPE=0    ;          0=SINGLE, 1=DOUBLE
000003         INTEGER      TYPE
000004         PARAMETER ( TYPE=0 )
000005 C: WSC=1     ;          WIDE-USED SIMULATION CODE
000006         INTEGER      WSC
000007         PARAMETER ( WSC=1 )
000008 C: EAC=0     ;          EXP. ANALYSIS CODE
```

DOUBLE : 倍精度／単精度

```
EDIT --- J3859.Z2.LIB300I.FORT77(DOUBLE) - 01.00 ------------ COLUMNS 001 072
COMMAND ===>                                                SCROLL ===> CUR
****** ***************************** TOP OF DATA ***************V10L30********
000001 *COMDECK                    DOUBLE.SYSTEM
000002 CX          TYPE4 (TYPE=0)
000003 CX      IMPLICIT REAL*4 (A-H,O-Z)
000004 CX          TYPE8 (TYPE=1)
000104         IMPLICIT REAL*8 (A-H,O-Z)
****** ***************************** BOTTOM OF DATA ****************************
```

#SYSGEN記述の精度を決めるTYPEパラメータとのDOUBLEの内容は必ず一致させておく事。

⇩ DESIGNER実行

```
---------------------------  ENGINEER DATA 3  -------------------------------
COMMAND ===> GO
<<< ENGINEER >>>                        COMMAND GO IS TO EXECUTE ENGINEER.
                                                BROWSE IS TO DISPLAY SPEC DATA.
    CODE SPEC NAME ===> TYPE1    (MEMBER NAME IN CODE SPEC DATASET)
      LOAD MODULE  ===> #LIB1D.LOAD
           MEMBER  ===> TEMPNAME
           NEW/OLD ===> NEW                        UNIT ===> TSSWK
    RESTART OPTION ===> YES    (YES/NO)
         EXECUTION ===> TSS    (TSS/BATCH)
    FORT77 & LINK  OPTIONS ===> YES (YES/NO)
```

⇩ ENGINEER実行

ユーザの目的とするロード・モジュールが#LIB1D.LOADに作成される。

7. 結　　語

汎用性のあるソース・プログラム編集ツール(NEWORG)を開発した。NEWORGは，ソース・プログラム・パッケージから，モジュール相互の呼出し関係を解析し，必要なモジュールを選択し，ロード・モジュールを組み上げ，これを管理するユーティリティである。FORTRAN77言語で書かれたソース・プログラムであれば，このユーティリティを利用する事ができる。これを用いる事によって，トカマク輸送コードの様な多様な目的仕様をもって開発された大規模ソース・プログラム・コードの標準化(Library化)が可能である。

1Dライブラリーは，既にこのシステムに移行している。非円形断面をもつ高ベータ・プラズマの輸送解析を目的とした1.5D輸送コードの開発を，このシステムを用いて，現在行なっている。完成されたソース・プログラム・パッケージから，ユーザの目的とするロード・モジュールを組み上げるのに，NEWORGが有効であるばかりでなく，これによって，発展性，互換性のあるプログラム開発が可能となった。

謝　　辞

本システムの設計，開発に際して，(財)原子力データ・センター(ナサック(株))滝川好夫，熊倉利昌両氏の努力に負う所が大きく，ここに感謝します。ツリー解析(ANALYZER)を開発するにあたりまして，「ANALYSIS77」の使用を許可して頂いた，計算センター外来研究員(富士通(株))の奈良岡賢逸氏に感謝します。メニュー画面の設計，システムの改良におきまして，(財)原子力データ・センター(NSK(株))木原和久氏に有益な助言を頂きました事に感謝します。

なお，本報告書はCATSアプリケーション　PANDA，DOGS(臨界プラズマ第二実験室開発)を用いて作成されました。その入力にあたりまして，(株)原子力資料サービスの島崎敦子さんにお世話になりました。ここに感謝の意を表します。

JT－60実験解析コード開発・整備計画に関し，吉川允二理事，那珂研究所　田中正俊所長，臨界プラズマ研究部　田村早苗部長の御激励に感謝致します。

7. 結　　語

汎用性のあるソース・プログラム編集ツール(NEWORG)を開発した。NEWORGは，ソース・プログラム・パッケージから，モジュール相互の呼出し関係を解析し，必要なモジュールを選択し，ロード・モジュールを組み上げ，これを管理するユーティリティである。FORTRAN77言語で書かれたソース・プログラムであれば，このユーティリティを利用する事ができる。これを用いる事によって，トカマク輸送コードの様な多様な目的仕様をもって開発された大規模ソース・プログラム・コードの標準化(Library化)が可能である。

1Dライブラリーは，既にこのシステムに移行している。非円形断面をもつ高ベータ・プラズマの輸送解析を目的とした1.5D輸送コードの開発を，このシステムを用いて，現在行なっている。完成されたソース・プログラム・パッケージから，ユーザの目的とするロード・モジュールを組み上げるのに，NEWORGが有効であるばかりでなく，これによって，発展性，互換性のあるプログラム開発が可能となった。

謝　　辞

本システムの設計，開発に際して，(財)原子力データ・センター(ナサック(株))滝川好夫，熊倉利昌両氏の努力に負う所が大きく，ここに感謝します。ツリー解析(ANALYZER)を開発するにあたりまして，「ANALYSIS77」の使用を許可して頂いた，計算センター外来研究員(富士通(株))の奈良岡賢逸氏に感謝します。メニュー画面の設計，システムの改良におきまして，(財)原子力データ・センター(NSK(株))木原和久氏に有益な助言を頂きました事に感謝します。

なお，本報告書はCATSアプリケーション　PANDA,DOGS(臨界プラズマ第二実験室開発)を用いて作成されました。その入力にあたりまして，(株)原子力資料サービスの島崎敦子さんにお世話になりました。ここに感謝の意を表します。

JT－60実験解析コード開発・整備計画に関し，吉川允二理事，那珂研究所　田中正俊所長，臨界プラズマ研究部　田村早苗部長の御激励に感謝致します。

文　　献

[1] JT－60実験解析コード作業グループ(編)平山俊雄，一次元トカマク輸送コード・ライブラリー・システム(LIBJT－60)，JAERI－M 82－204(1982)

[2] 竹田辰興，常松俊秀，栗田源一，可変配列サイズ・プログラムのためのプリプロセッサー・システム'EOS'，JAERI－M 82－097(1982)

[3] 富士通FACOM OS IV／F4 MSP PFD使用手引書 プログラム開発機能編(1983)

[4] 平山俊雄，安積正史，汎用図形処理ユーティリティ"GPREP"と"LIBGR"，JAERI－M 86－023

[5] 奈良岡賢逸，私信(1987)
奈良岡賢逸，FORTRAN静的解析プログラム ANALYSIS77，Computer情報No.40.