

# フォートラン流のCAI用言語をもった CAIシステムの開発

(昭和53年10月31日 原稿受付)

西日本工業大学	情報処理センター	深川幸紀
九州工業大学	情報工学教室 大学院	安部暁一
〃	情報処理教育センター	積山洋子
〃	〃	矢鳴虎夫
〃	〃	中山泰雄
八代工専	情報電子工学科	森本義広
九州大学	工学部電子工学科	吉田将行
九州工業大学	情報工学教室	磯泰行

## A Design and Development of the CAI-system with a FORTRAN-like CAI Language

by Yukinori FUKAGAWA  
Kyoichi ABE  
Yoko TSUMIYAMA  
Torao YANARU  
Yasuo NAKAYAMA  
Yoshihiro MORIMOTO  
Sho YOSHIDA  
Yasuyuki ISO

In the Educational Center for Information Processing, we have considered a CAI system and we have developed an interactive or batch system for the picture data of the character display, a core-resident FORTRAN compiler for the education and the CAI system which supports multi-terminals.

In this report, an outline of the FORTRAN-like language for the CAI-system, a software system design and an example of the CAI course program using this language.

The higher level language is effectively used in the CAI system because as almost all problems can not be skilfully processed without high intelligency and programs using the higher level language is easily written for setting a high intelligency.

### 1. まえがき

九州工業大学情報処理教育センター(以後センターと呼ぶ)では、計算機を基礎とした教育についての検討を行なってきたが、このなかで、CAIシステムをとりあげ、これに関するアプローチとして、構造を持ったデータをもとに動作するCAIシステム<sup>16)</sup>と、フォートラン流のCAI言語を持ったシステム<sup>17)</sup>を作成・計画した。

CAIシステムがより多く使用され、発展するには、以

下の2点が重要であると我々は考える。

(1) 問題(設問)とこれに付随する情報を端末に表示するソフトウェアデータの作成が容易であること。

(2) 表示された問題に対する学習者の応答入力情報を判断し、加工し、表示し、また、応答によって次の問題への分岐選択が容易にできるようなインテリジェンシーをシステムが持っていて、これを、CAIコース作成者が容易に書くことができること。

我々は、(1)に関しては、バッチ型並びに会話型画面デー



その3は、CAI 動作系 (解釈実行ルーチン) である。

1は問題を作成し、2はその問題に対する解答を判断するインテリジェンシーを付加し、3は問題を呈示し、応答入力判断し、問題フレームを進めて行くシステムである。本文は主として2、3について述べる。図-2にシステムの流れを示す。

本 CAI システムは、CAI の問題画面 (フレーム) (K) をあらかじめ作成して CAI 問題ファイルに登録しておいて、次にその問題に対する学習者の応答入力 ( $\Sigma$ ) を処理し、判断し、次の問題 (フレーム) に行かせるようなプログラム ( $\delta$ ) をフォートラン流の CAI 言語で書かせ、これを翻訳して生成された目的プログラム ( $\delta'$ ) を該当問題画面の付加プログラムとして記入することをオンライン会話形で行なう。そして次に、すでに目的プログラム ( $\delta'$ ) が記入された CAI コースに対しては、学習者からの応答入力 ( $\Sigma$ ) と付加プログラム ( $\delta'$ ) によってフレーム (K) を次々に進めて行くことを行なう。図-3にこの様子を示す。

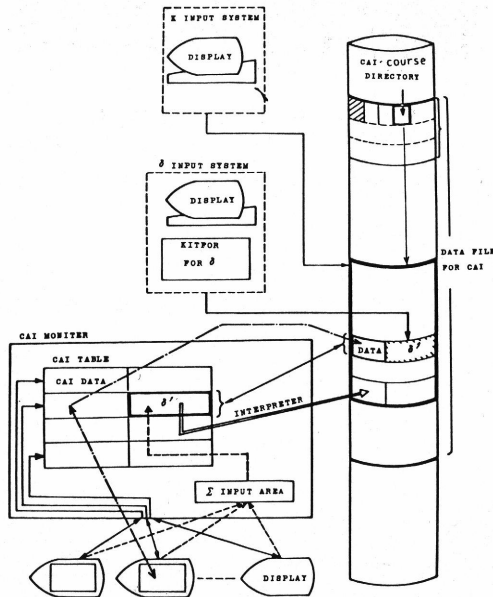


図-3 CAIシステムの定義  
Fig. 3 Outline of formal definition of our CAI system

3.1. CAI ソースプログラム作成ルーチン

本ルーチンは、すでに登録してある CAI 問題画面の1

つに対して、その問題に対する応答入力処理・判断情報 (CAI のインテリジェンシー) を 200 ステートメント以内のソースプログラム (CAI 用言語) で記述するとともに、ソースプログラムの翻訳・エラーチェックを行なうための前処理を行なう。このルーチンのコマンドとその意味は以下の通りである。

- CAISREAD .....ソースプログラムを作成する。
- CAISTART .....ソースプログラムを翻訳する。
- KITSDISP.....ソースプログラムを表示し、訂正する。
- CAIPOUT .....CAI 問題画面とそれに対するソースプログラムをラインプリンターに出力する。

CAISREAD コマンドに対して、CAI のコース名や問題画面番号 (4桁) の指定入力を行ない、問題画面をソースプログラム作成者 (CAI コース作成者) に確認させながらソースプログラムを作成させ、ワーク・ファイルに格納して行く。

この動作の様様を写真-1, 2, 3, 4, 5, 6 に示す。ラインプリンタ出力の例を図-4 に示す。

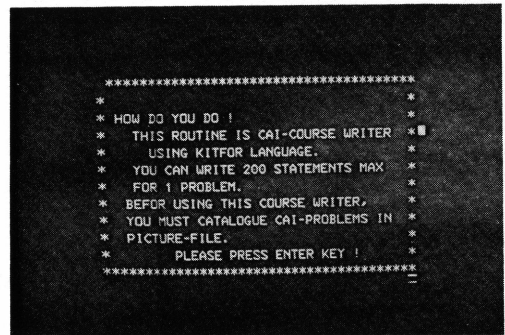


写真-1

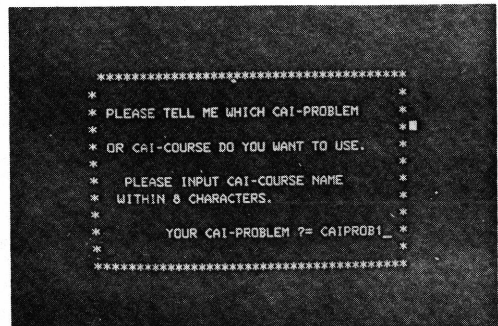
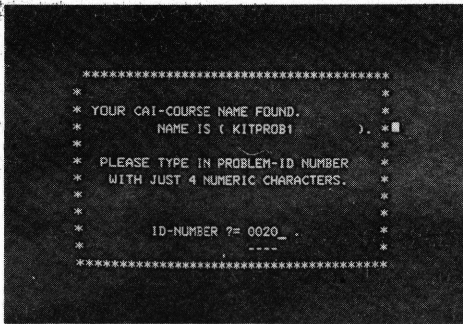
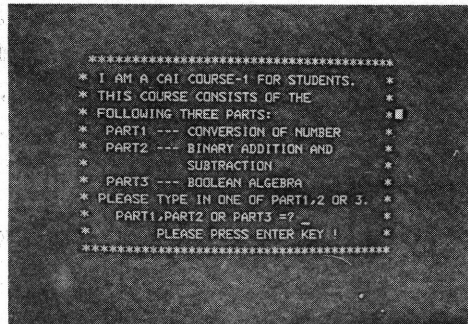


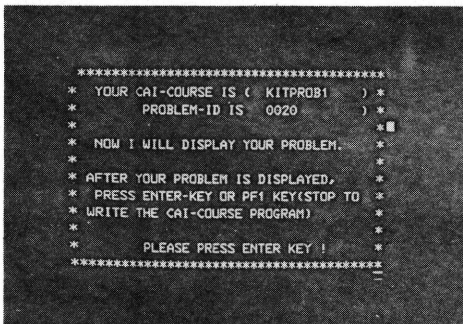
写真-2 コース名入力画面



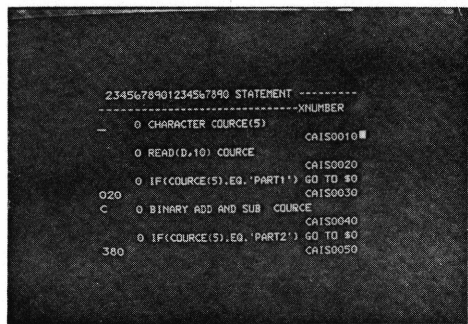
写真一 3 問題番号入力画面



写真一 5 問題画面表示例



写真一 4



写真一 6 ソースプログラム表示例

```

KITFR0C00 *****
KITFR0C01 *
KITFR0C02 * CAI COURSE NAME IS ( KITPROB1 ) *
KITFR0C03 *
KITFR0C04 *
KITFR0C05 * PROBLEM ID NUMBER ( 0080 ) *
KITFR0C06 *****
KITFR0C07
KITFR0C08 -----*****
KITFR0C09 * PROB A-1 ANS DECIMAL 41 = *
KITFR0C10 * BINARY 100101 *
KITFR0C11 * CONGRATULATIONS *
KITFR0C12 * PLEAASE ATTACK AGAIN THE FOLLOWING *
KITFR0C13 * PROBLEM. *
KITFR0C14 *
KITFR0C15 *
KITFR0C16 * PROBLEM A-2 *
KITFR0C17 * CONVERT DECIMAL 0.6875 TO BINARY. *
KITFR0C18 * ANS = A *
KITFR0C19 * PLEASE PRESS ENTER KEY. *
KITFR0C20 *****
KITFR0C21 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
KITFR0C22 C EXAMPLES-OF CAI STATEMENT C
KITFR0C23 C C
KITFR0C24 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
KITFR0C25 CHARACTER AAAAAA(16),BBBBBB(256),CCCCC(120)
KITFR0C26 DATA/AAAAAA(16)/'1234567890123456'/
KITFR0C27 WRITE(D,10)
KITFR0C28 10 FORMAT(1H,'***** EXAMPLE OF CAI STATEMENT *****')
KITFR0C29 READ(D,20) XX,YYY,ZZZZ
KITFR0C30 20 FORMAT(3F12.5)
KITFR0C31 CALL $NAME1$0010
KITFR0C32 CALL $0030
KITFR0C33 1 CHGFORMAT(P)
KITFR0C34 2 CHGFORMAT(U(10,10),U(10,05,05))
KITFR0C35 PASS XX,YYY,'MOJI DATA'
KITFR0C36 ACCEPT XX,YYY,ZZZZ
KITFR0C37 GO TO $$PROB1$0010
KITFR0C38 GO TO $0020
KITFR0C39 C EXAMPLE OF CAI STATEMENT
KITFR0C40 IF(AAAAAA(4).EQ.'ABCD'.OR.BBBBBB+INDEX(16).EQ.CCCCCC.AND.DDDDD+10CAI$0210
KITFR0C41 -(4).EQ.4HXXXX) GO TO $0050
KITFR0C42 RETURN
KITFR0C43 END
    
```

図一 4 出力例

### 3.2. CAI 字句解析, 構文解析, 目的コード生成・登録ルーチン

字句解析部は, 教育用メモリ常駐型フォートラン (KITFOR)<sup>18)</sup>と同様であるので文献を参照されたし。

構文解析部についても KITFOR と同様であるが, 目的コード (仮想マクロ命令) を短縮化し, ファイル中での記録密度を上げたので, この形式について述べる。

目的コードは,

〈長さ〉〈命令コード〉〈オペランド〉からなる。

〈長さ〉は 16 進 2 桁で目的コード全体の長さ, 〈命令コード〉は 16 進 2 桁で操作の種類, 〈オペランド〉は各操作に応じて必要となる情報となっている。すなわち, 短縮前の目的コードから実行時に直接必要でない情報 (デバッグ時に必要となる) を削除している。また, 〈オペランド〉中に表われるアドレスは制御領域等を含むプログラム実行領域 (後述) 先頭からの相対アドレス (長さは 2 バイトで先頭 1 ビットにより直接または間接アドレスであることを指定する) であり, こうすることによってプログラムの再配置を可能にしている。

次に, CAI コース記述のための用意されたステートメント (後述) に対して生成される命令コード (表意形式) とその機能を以下に記す。

〈命令コード〉	〈機能〉
INPT	端末からの入力。
FCHG	端末表示画面の入力域位置等の変更。
PGOT	問題 (フレーム) 移動。
PCAL	問題 (フレーム) 呼び出し。
PRTN	呼び出し問題 (フレーム) からの復起。
PASS	問題 (フレーム) 間通信域に値を移す。
ACPT	PASS で移された値を取り出す。

ある問題画面に対して, 目的コードの生成並びにエラーチェックが終了すると, 生成された目的プログラムはその問題画面と対して CAI 問題ファイル中に記入される。図-5 に記入形式を示す。記入されるのは, 目的プログラム (目的コード, 初期化された変数と配列, 定数, その他付随する情報) とカードデータである。データ記述域中には, それらの記入位置等に関する情報が, 書き込まれる。なお, 翻訳 (字句解析, 構文解析) 中にエラーが発見されると記入は行なわれずにエラーハンドラー<sup>18)</sup>によってエラー情報が表示される。

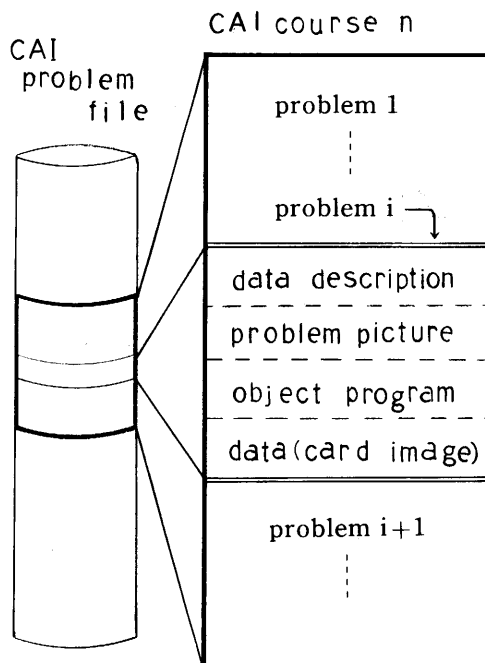
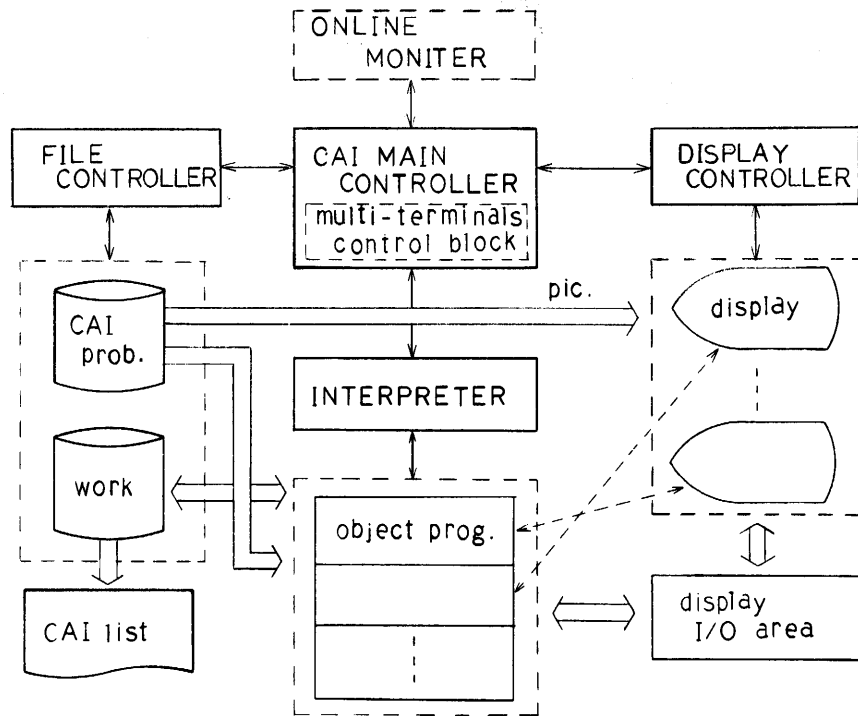


図-5 問題 (フレーム) の記入形式

### 3.3. CAI 動作系 (CAI-course Interpreter)

CAI 動作系は, 問題画面作成システムや, CAI ソースプログラム作成, 字句解析, 構文解析, 目的コード生成, 登録の各ルーチンとは独立して動作し, それ自体でオンライン・サブシステムを構成する。本システムは, 端末から, '\$\$KITCAI' のコマンドを入力することにより起動され, CAI コース名の入力を経て, その CAI コースを開始する。CAI コースの実行制御は次のようになる。

まず, CAI 問題ファイルより, 問題画面並びにそれに付加された目的プログラムの読み込みを行なう。次に, その問題画面を該当端末に表示するとともに, 目的プログラムを実行領域に移し, その解釈実行を開始する。目的プログラムから学習者の応答入力 (解答) を要求する命令が出されたときには, 学習者が解答を入力し, 該当端末の割り込み発生キーを押し, それが受け付けられるまでは, その問題表示端末は入力待ちの状態になる。その間は他の端末に対する処理がなされる。その後, 受け付けられた応答入力に目的コードを解釈実行することによって判断, 加工, 表示等がなされ, 問題移行を要求する命令が出されたときに現ステップの問題に対する処理



図一六 CAI 動作系の構成と制御並びにデータの流れ

が終り、続いて、該当端末からの割り込み発生キーを待つ、次の問題の表示を行なう。この問題に対しても以上に述べたのと同様な処理がなされ、これがくり返されて CAI コースのステップが進められて行く。

CAI 動作系の構成は、

- (1) 主制御部
- (2) 端末（ディスプレイ）制御部
- (3) ファイル制御部
- (4) 解釈実行部本体

である。図-6に全体構成と制御、データの流れを示す。

以下に各部の機能を述べる。

主制御部は各端末ごとの CAI コース進行状況（端末の状態）が記入された端末制御表を参照しながら、CAI 動作系全体の制御を行なう。

その主な機能を以下に記す。

- a) コマンド、割り込み発生キーの処理

問題（フレーム）の後戻り、端末表示中の画面のハードコピー、CAI コースの打ち切り、その他特定画面

の表示に関する処理を行なう。

- b) 目的プログラム実行時間の割り当て  
オンラインシステム全体としての応答時間を平均化するためである。
- c) 目的プログラム実行領域の割り当て
- d) 解釈実行部本体の起動並びにその復帰情報の解析とその処理  
復帰情報に応じて端末制御部を起動し出力データを表示したり、次の問題（フレーム）に移るための処理を行なう。
- e) CAI コース内履歴の記録

新しい問題（フレーム）へ移るたびごとにその問題番号（画面通し番号）を端末（学習者）ごとに記録しておき、問題の後戻り（後戻キーが押されたとき）や、CALL 文（後述）で呼び出された問題から元の問題に戻るときには、その記録が参照される。

端末（ディスプレイ）制御部は CAI 動作系内の端末に対する入出力作業をすべて行なう。

その機能を以下に記す。

- a) 問題画面の表示
  - b) 出力データ (目的プログラム実行中の表示)
  - c) システム・メッセージの表示
  - d) 端末表示画面の読み取り (ハードコピーのため)
- ファイル制御部は CAI 問題ファイルからのデータの読み込みや、ワークファイルの制御を行なう。その機能を以下に記す。

- a) CAI 問題ファイルからの問題画面並びにそれに付加された目的プログラムの取り出し
- b) ワークファイルへの実行プログラムの一時退避とその復元 (多重処理時に必要となる)
- c) ワークファイルへのラインプリンタへ出力するデータの格納

解釈実行部本体は、プログラム実行領域中に格納された図-7の形式のプログラムの実行制御を行なう。

その構成を以下に記す。

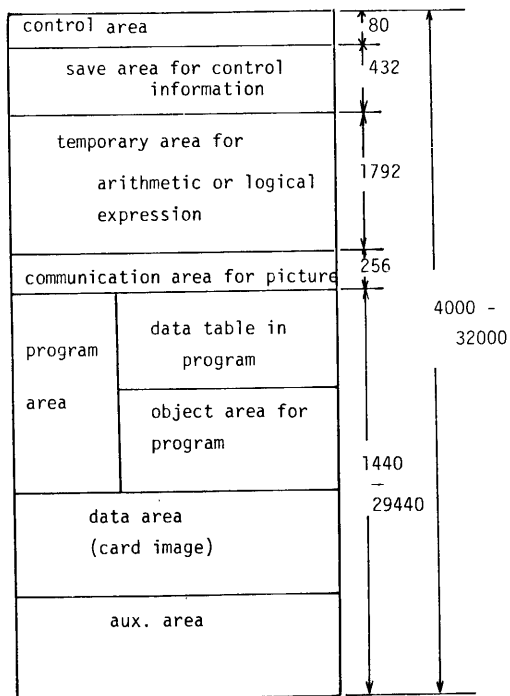


図-7 CAI インタープリターが制御するプログラムエリア  
Fig. 7 The program area for CAI interpreter

- a) 目的コード (仮想マクロ命令) の取り出し制御を行なう部分。
- b) 取り出されたものを実行する部分 (サブルーチン群からなる)
- c) 実行後のエラーや入出力要求等をチェックする部分

図-8に制御領域の構造を、図-9に制御情報退避域の構造を示す。

解釈実行部本体は主制御部によって起動されるが、次の場合に制御は復帰情報とともに主制御部へ返される。

- a) 該当端末 (学習者) からの入力要求 (INPT)
- b) 該当端末への表示要求 (WRIT 等)
- c) 実行一時停止の要求 (PAUS)
- d) 問題 (フレーム) 移行の要求 (PGOT 等)
- e) CAI コース終了の要求 (STOP)
- f) 割り当て時間の経過
- g) エラーの発見

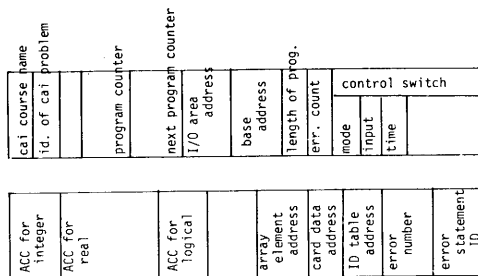


図-8 制御領域  
Fig. 8 The control area

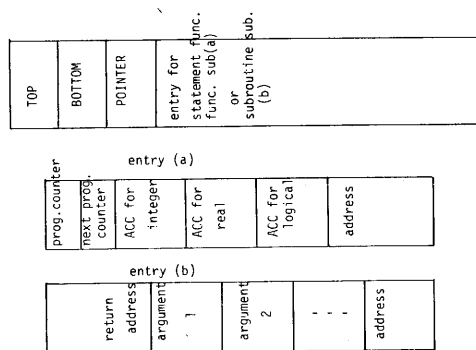


図-9 制御情報退避域  
Fig. 9 The save area for control information

#### 4. CAI 用言語の文法と意味と使用例

##### 4.1. 文法

KITFOR<sup>18)</sup>で用いられるステートメントはすべて CAI 用言語で使用可能であり、ここでは簡単に述べる。

- (1) DIMENSION 文
- (2) COMMON 文
- (3) TYPE 文
- (4) FORMAT 文
- (5) DATA 文
- (6) 文関数, サブルーチン, 関数副プログラム
- (7) 算術代入文
- (8) 算術 IF 文
- (9) DO 文, CONTINUE 文
- (10) READ 文, WRITE 文
- (11) CALL 文, RETURN 文
- (12) GOTO 文, 計算型 GOTO 文, 割り当て型 GOTO 文
- (13) COMMENT 行, END 行

CAI 用ステートメント

- (1) 文字型宣言文  
CHARACTER  $v_1 (l_1), v_2 (l_2), \dots$   
ここで  $v_1, v_2$  は変数名,  $l_1, l_2$  は文字長である。
- (2) 文字型 IF 文 (文字型比較)  
IF ( $b$ ) S  
ここで, S は実行文 (DO 文, IF 文を除く),  $b$  は文字型式である。  
文字型式  $b$  とは次に示すようなものである。  
i  $v_1 + iv_1 (l) r\text{-op} v_2 + iv_2$   
ii  $b_1 l\text{-op} b_2$   
iii .NOT.  $b_1$   
iv ( $b$ )  
 $v_1, v_2$ ; 文字型変数名または, 文字型定数  
 $iv_1, iv_2$ ; 整変数名または, 整数 (インデックス修飾時に書く, 省略可)  
 $l$ ; 長さ (256 以下の整数)  
 $b_1, b_2$ ; 文字型式  
 $r\text{-op}$ ; 関係演算子 (.GT., .GE., .EQ., .NE., .LE., .LT.)  
 $l\text{-op}$ ; 論理演算子 (.AND., .OR., .XOR.)

例 1. IF (ANS+5(3). EQ. 'ANS') GO TO 10

例 2. IF (ANS (1). NE. CHAR1. AND.  
ANS (1). NE. CHAR2) STOP

- (3) 文字型代入文  
 $v_1 + iv_1 (l) = v_2 + iv_2$   
 $v_1 + iv_1 (l) =$  文字型定数  
例 1. ANS (15) = 'TOKYO AND OSAKA'  
例 2. ANS+10 (5) = ANS
- (4) 入出力文 (character display に関する I/O 文)  
i .READ (D) 変数リスト  
ii READ (D,L) 変数リスト  
iii WRITE (D,L) 変数, 定数リスト  
L; FORMAT 文番号
- (5) GO TO 文 (問題移行)  
GO TO \$  $nnnn$   
GO TO \$\$  $name$  \$  $nnnn$   
 $nnnn$ ; 4 桁の整数 (分岐する問題番号)  
 $name$ ; コース名
- (6) CALL 文 (問題呼び出し)  
CALL \$  $nnnn$   
CALL \$\$  $name$  \$  $nnnn$   
問題  $nnnn$  を呼び出し, 次に述べる PRETURN 文で戻る。
- (7) PRETURN 文 (呼び出し問題への復帰)  
PRETURN
- (8) PASS 文  
PASS 変数, 定数リスト  
変数または定数の値を問題間通信域に移す。
- (9) ACCEPT 文  
ACCEPT 変数リスト  
問題間通信域から値を取り出す。  
PASS 文と ACCEPT 文で問題間のデータの引き渡しを行なう。
- (10) CHGFORMAT 文 (display 画面属性の変更)  
CHGFORMAT (P)  
CHGFORMAT (U ( $col, row, l, \dots$ ))  
P; 保護化指定  
U; 非保護化指定 (入力エリアを作る)  
 $col$  = 行,  $row$  = 列,  $l$  = 長さ
- (11) 文字型定数  
'aaaa'  
 $nHaaaa$   
 $nCaAAA$



aaaa; 任意文字列

n; 文字列の長さ

#### 4.2. 使用例

以下に簡単な問題とこれに対する判断プログラムの例を示す。

PLEASE ANSWER THE VALUE PI? =

この問題に対する判断プログラムは次のように書ける。

```
READ (D) ANS
IF (ABS(ANS-3.14).LT. 0.01) GOTO $ 0030
GOTO $ 0040
```

我々はこの言語を用いて、学習者の解答(応答入力データ)中に含まれる順不同のキーワードの識別や、順序付けされたキーワードの識別を、DO 文とインデックス修飾を用いた文字型 IF 文で記述することが出来る。これらを以下に示す。ただし、キーワードを A と BB とし、応答入力データの文字数は 256 文字とし、ANS という文字型変数中に入力データは読み込まれているとする。

##### (1) 順不同のキーワードの検出プログラム

```
10 DO 1 I = 1,256
INDEX = I-1
IF (ANS+INDEX (1). EQ. 'A') GO TO 20
1 CONTINUE
GOTO $ 0030
20 DO 2 I = 1,255
INDEX = I-1
IF (ANS+INDEX(2). EQ. 'BB') GO TO 30
2 CONTINUE
GO TO $0030
C TWO KEY-WORDS A, BB FOUND
30 any statement when found
C IF KEY-WORD NOT FOUND, GO TO PROB 30
```

##### (2) 順序付けされたキーワードの検出プログラム

```
10 DO 1 I = 1,256
INDEX = I-1
IF (ANS+INDEX (1). EQ. 'A') GO TO 20
1 CONTINUE
GO TO $0030
20 DO 2 J = 1,256
INDEX = J
```

```
IF (ANS+INDEX (2). EQ. 'BB') GO TO 30
```

```
2 CONTINUE
```

```
GO TO $0030
```

```
30 any statement if found
```

```
C IF KEY-WORD NOT FOUND OR
```

```
NOT SEQUENCE, GO TO PROB 30
```

#### 4.3. CAI 問題間のリンク情報について

我々の高級言語を用いた CAI システムにおいて、1つの CAI コースの全問題に対するリンク情報(問題間のつながり)を知ることが出来る。例えば、図-10 においては、P<sub>0</sub> の問題に対するソースプログラム中に GO TO \$nnnn 文を (P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>9</sub>) の問題へ分岐するためには少なくとも 4 つ以上書かななくてはならない。これを利用して、問題間のデータ構造やリンク構造をソフトウェア的に把握することが可能である。そして把握された情報をもとに、問題間のループの検出や、CAI コースの作成者に問題間のリンク構造図等を出力してやることも可能である。

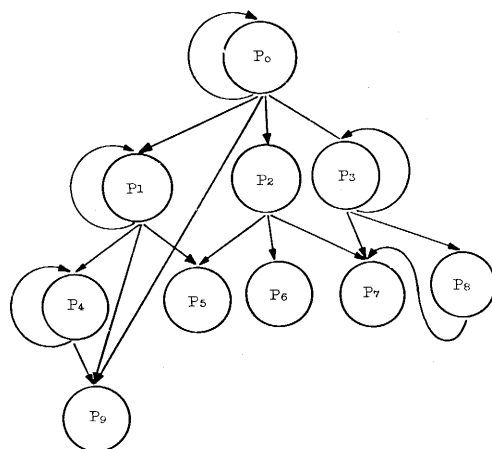


図-10 CAI コースの構造例

## 5. 検 討

システム開発にあたり、CAI システムに高級言語を採用することの有効性ははっきりした。特に、強力な文字処理文や、問題間の通信機能を果す文が必要であることもわかった。また、学習者の解答(応答入力)をある規則に従って標準形に変換するルーチン等があると便利であることもわかった。

本システムは稼動したばかりであり、また、CAI システムの必要条件への1つのアプローチであり、今後検討すべきことも多々あるが、今後本システムを使用して効果があるだろうと思われる分野のCAI コースを選び、適当な問題を作成し、本システムに関する評価等を行ないたいと考えている。

## 謝 辞

本システムの開発にあたり、コーディング、パンチやデバッグ作業にたずさわっていただいたセンター夜間操作員の学生諸君、卒業研究の一環として本システムの一部を作成された本学卒論生、多数の図面作成を行なっていただいた野田事務官、並びにその他多数の関係者の方々に深く感謝いたします。

## 参 考 文 献

- 1) 情報処理教育センター内部資料; “KITFOR コンパイラの概要”, 1977.
- 2) 情報処理教育センター内部資料; “KITFOR コンパイラ I/O 制御部仕様書”, 1976.
- 3) 情報処理教育センター内部資料; “KITFOR コンパイラ字句解析部サブルーチン仕様書”, 1977.
- 4) 情報処理教育センター内部資料; “KITFOR コンパイラ構文解析部仕様書”, 1977.
- 5) 情報処理教育センター内部資料; “KITFOR コンパイラ解釈実行部仕様書”, 1977.
- 6) 情報処理教育センター内部資料; “KITFOR コンパイラエラーハンドラ仕様書”, 1977.
- 7) 情報処理教育センター内部資料; “会話型ディスプレイ用画面データ作成・ファイル管理システム概説書”, 1976.
- 8) 情報処理教育センター内部資料; “会話型ディスプレイ用画面データ作成・画面データファイル管理システムについて”, 1976.
- 9) 情報処理教育センター内部資料; “会話型ディスプレイ用画面データ作成・画面データファイル管理システム操作仕様書”, 1976.
- 10) 情報処理教育センター内部資料; “階層構造画面群表示システム”, 1975.
- 11) 情報処理教育センター内部資料; “キャラクター・ディスプレイを用いたパッチ型 CAI 教材作成システム手引書”, 1975.
- 12) 情報処理教育センター内部資料; “キャラクター・ディスプレイを用いた CAI システムへの1つのアプローチ”, 1976.
- 13) 情報処理教育センター内部資料; “CAI コース記述用言語構文解析部仕様書”, 1978.
- 14) 情報処理教育センター内部資料; “多端末 CAI 動作系仕様書”, 1978.
- 15) 深川, 矢鳴, 中山, 吉田; “キャラクター・ディスプレイ用画面の会話形成・データ構造付加システム”, 情報処理学会-MMS-2-7-資料, 1977.
- 16) 矢鳴, 深川, 中山, 吉田; “キャラクター・ディスプレイを用いた会話型 CAI 作成システム”, 電子通信学会-技研-ET 76-9, 1977.
- 17) 深川, 安部, 積山, 矢鳴; “常駐型コンパイラ-KITFORの開発”, 電子通信学会-技研-ET 77-7, 1977.
- 18) 深川, 安部, 積山, 矢鳴; “メモリ常駐型コンパイラ-KITFORの開発”, 九工大研究報告, No. 36, 1978.
- 19) W. H. Holtzmar; “Computer-Assisted Instruction, Testing and Guidance”, Harper & Row, 1970.
- 20) IBM; “Coursewriter III, Version 3, Student Text”, GC 20-1-744-1.
- 21) J. M. Lafuente and D. Grios; “Language Facilities for Programming User-Computer Dialogues”, IBM J. RES. Develop, Vol. 22, No. 2, 1978.
- 22) Y. FUKAGAWA et al; “A CAI-System with FORTRAN-like CAI Language”, International Computer Symposium, Academia Sinica, Republic of China, 1978.