

マイクロコンピュータによる C A I コースウェア作成／実行システム

石 本 菅 生

1. 序

C A I (Computer Assisted Instruction) のコースウェアプログラムを作成する場合、指導形態、その方略等に関する意思決定をいかに行うかということが非常に大切であることは論をまたない。これらの基本的な事項を決定したら、次にそれをコンピュータのプログラムとして構成することになる。

コースウェアを作成する過程は、大まかに言えば、教材の内容をそれ自体がもつ論理構造あるいは教師が教材を展開する順序にしたがってコーディングし、そうしてできるプログラムをコンピュータにかけてテストを繰り返えし、プログラムが既定の流れにそって学習者を導くことができるかどうかチェックし、コースウェアとして完成させる作業であると言える。実際のプログラミングに取りかかるに先立つてプログラムの構造を決める必要があるが、利用可能なコンピュータ、使用できる端末装置やコンピュータ言語に限定されて、逆にプログラムの構造や C A I の形態が決められてしまう場合も実際には非常に多い。コースウェアの開発にたずさわる者にとっては、実際的なプランニングの段階で、教材を十分に生かすことのできるハードウェア、教材を展開しやすいプログラミング言語を選択できることが大切である。これらの選択がうまく行われるかどうかによって、質の良い、使い勝手のよい C A I が生まれるかどうかが決まると言っても決して言いすぎではない。

以上のような理由から、マイクロコンピュータを C A I に使うことは、

もし使いやすいC A I言語があるならば、機器の価格も安く、特別の施設やマンパワーを伴わずに導入ができ、また、利用の仕方についての意思決定も容易であるため、非常に得策と言える。

2. C A I 言語

現在、C A I コースウェアの作成には、TUTOR, COURSEWRITER, BASIC, FORTRAN, COBOL, APL, PL/I 等の言語が使われている。このうち BASIC, FORTRAN, APL, PL/I 等は般用プログラミング言語と呼ばれており、その名の示すようにコンピュータを色々な目的で使うために制御するプロセス・オリエンデッドな言語であるため、特別な目的（課題）でコンピュータを利用する者には必ずしも便利とは言えない。マイクロ・コンピュータの言語について言うならば BASIC, PASCAL とも般用言語であり、一般的なプログラミングを容易にするために開発拡張されてきた言語である。

これに対して特定の利用目的に合わせて開発されたプログラミング言語として課題向き言語 (Problem Oriented Language) がある。人口知能の研究のために開発された LISP やプロセス制御のための FORTH などはその例であるが、C A I 用にも専用言語がいくつか開発されている。大型、中型コンピュータ用のC A I言語としては、IBM のCOURSEWRITER, CDC の TUTOR がよく知られており、本学の IBM 4341 コンピュータシステムでも COURSEWRITER を使ってコースウェアを作成することができる。COURSEWRITER や TUTOR にはコースウェアの開発者やプログラマーにとって便利な命令語が豊富に備わっている。

C A I のプログラミング言語は一般に非常に多くのテキストを学習者に提示することと、答のマッチングを簡単にできるように作られている。テキスト文を提示したり、答えを定義してある正答と対照させるための論理を般用言語で記述するとプログラムステップが多くなるが、C A I 専用言語で記述するとわずかなステップでできるためプログラムが短かくて済み、

プログラミングのエラーも発見しやすく，コースウェアの作成作業が簡略かつ迅速にできるという利点がある。マイクロコンピュータが出現して以来，マイクロコンピュータによるC A Iプログラムの作成やC A Iの実施が報告されているが，そのほとんどが般用言語の BASIC によるものであった。BASIC が Beginner の為の言語という意味を持つとは言っても，これを駆使して1つのコースウェアを仕上げるにはそれ相当の知識と技能及び根気が必要である。従って，マイクロ・コンピュータによるコースウェアの作成にも，C A I 専用，あるいはある種の課題向き（C A I の幾つかの類型の中の1つによく適合する）言語が準備されることが望ましい。

3. コンパイラ言語とインタプリタ言語

C A I 言語は，利用者の用いる言語体系と，コンピュータの機械制御言語（マシン語）との関係にもとづいて，コンパイラ言語，インタプリタ言語，両者の機能を兼ね備える言語，の3種に大別される。

コンパイラは，プログラミング言語で書かれた命令群（ソース・プログラム・ステートメント）をコンピュータが即実行可能な機械語コードに一括して翻訳するプログラムであり，またインタプリタは，プログラミング言語で書かれた命令を，1命令単位で順次実行する通訳プログラムである。従ってコンパイラのC A I 言語を使うと，全てのプログラムステートメントを機械語に変換した後でなければプログラムの実行ができないので，コースウェア作成時にはコースウェアプログラムの部分的なトライアウトを繰り返えすことが必須であることから，手間と時間を費やし使い勝手が悪いため，コンパイラ言語は教材作成の作業をする者にはあまり好まれていない。

先にも述べたように，インタプリタ言語はコースウェアを命令単位で順次実行させることが可能なため，プログラムのトライアウトに便利であり，コースウェアの変更や手直し，その結果のチェック等が容易にできるという利点がある。しかし，命令単位での順次通訳による実行であるため，

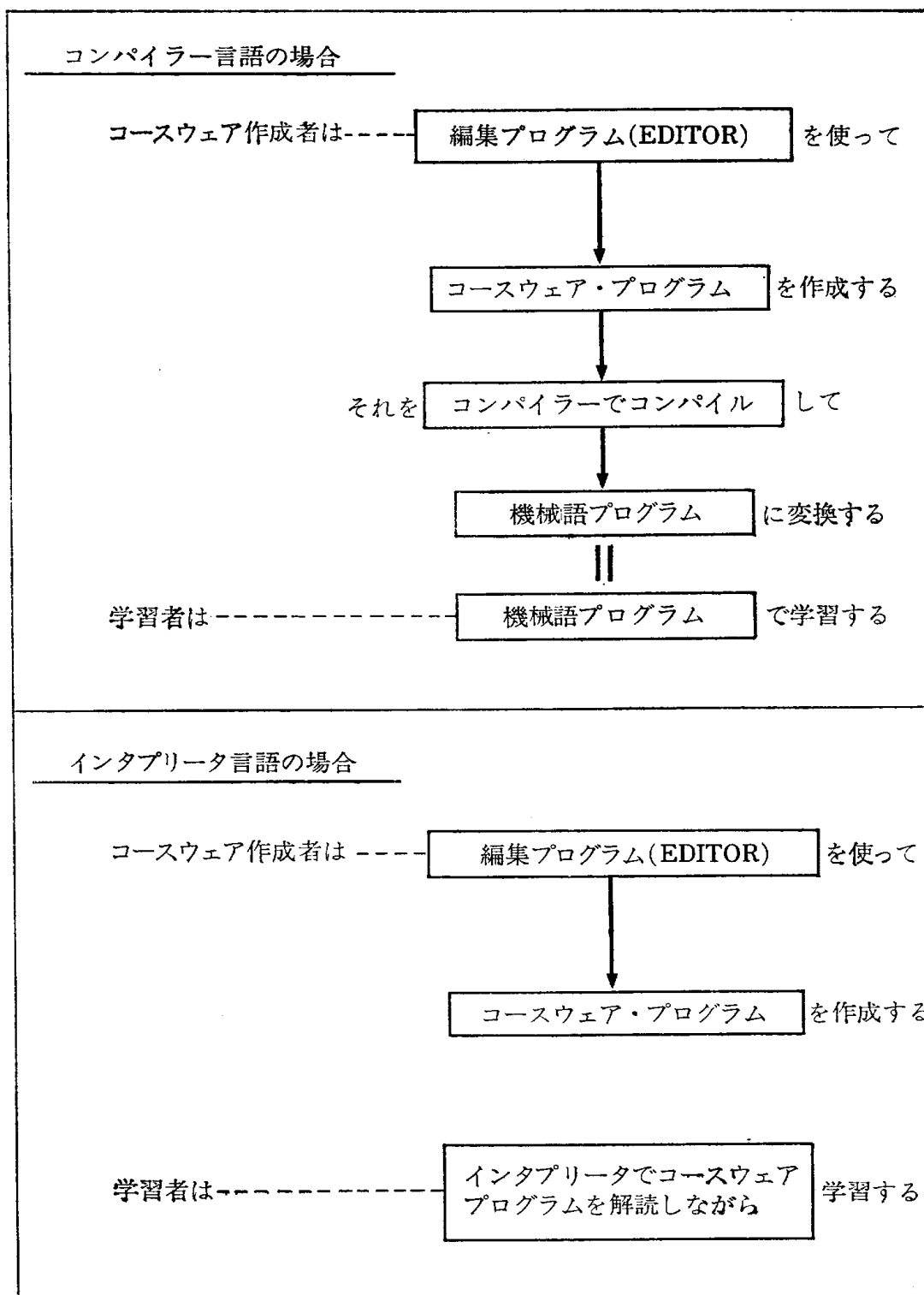


図1 コンパイラ／インタプリタによるコースウェアの実行手順

実行速度の点ではインタプリタ言語はコンパイラ言語に遠く及ばない。

大型コンピュータ用のC A I 言語は両者の優れた点を取り入れて、コースウェア作成時にはインタプリタし、学習に供用するときにはコンパイルして実行することのできるシステムが使われるようになり、C A I のシステムプログラム自体が複雑なものになっている。

通常このようなC A I システムはTSS (Time Sharing System) のもとで作動しており、1つのコンピュータを時分割して多くの利用者に共用させるため、コンパイルされたコースウェアを実行させても非常に高いパフォーマンスを期待することは大変難かしい。

C A I にマイクロコンピュータを使うということになると、学習者1人ひとりが stand alone のコンピュータを100%占有することになるので、インタプリタ言語でも十分に実用にたえるパフォーマンス（入力に対するレスポンスタイムが3秒程度であればよいとされている）を得ることができる。

以上述べたことから、マイクロ・コンピュータのためのC A I 言語はインタプリタタイプのもの方が使い勝手もよいし、パフォーマンスの点でも実用上十分であると言えよう。

4. PILOT 言語

PILOT は、1975年に John Starkweather によってコンピュータを医学部の学生達に薬学の知識を教える道具として使う目的で開発された言語である。Programmed Inquiry Learning or Teaching の頭文字を連ねて PILOT と名付けられた。その後色々なコンピュータシステムに適合するよう幾つかの PILOT 言語が作られ他のC A I 言語の開発にも示唆するものがあったが、すでに COURSEWRITER や TUTOR など大型コンピュータ用のC A I 言語の開発が進んでおり、そのため多くのC A I が大型機使用に走ったこともあって次第に影がうすれ、現在では Western Washington University (Washington 州, U.S.A.) のミニコンによる

C A I で使われているだけであると報告されている。

大型、中型、小型のコンピュータシステムでは、オペレーティングシステム自体に豊富な機能が備わっているのので、その上でオペレートされる C A I システムにも当然色々な機能が要求される。しかし、マイクロコンピュータの場合は豊富な機能をもつことよりも使い勝手の良さ、限られた記憶容量の効率のよい利用という面が尊重される。

PILOT 言語はインタプリタ言語であり、その格納に要する RAM (ランダムアクセス・メモリー) のサイズがわずかで済み、また、問答形式の C A I コースウェアは、簡単な、わずかな数のコマンドで手軽に記述できる、という特長をもっている。そのため、最近になってマイクロコンピュータに適した C A I 言語として見直されはじめ、すでに試作がなされ、また、米国では或る種のマイクロコンピュータのソフトウェアとして発売が開始されたとも伝えられている。

5. MINI-PILOT / TRS レベル II の開発

著者は、最近 PILOT 言語インタプリタの試作に関する資料を入手する機会があったので、本学における C A I 教材作成実習用ソフトウェア及び自主学习教材作成の媒体として実際に利用できるよう、ハードウェア・システムに合わせて必要な改訂と改善を行った。

この Authoring / Operating System は、マイクロコンピュータに備わっている優れた編集機能とリンクさせることで非常に使いやすいコースウェア作成システムとして権能させることができ、また、コマンド1つで実行モードと切りかえて作成中のプログラムのトライアウトをすることも可能である。

システム開発の全体的な枠組みとその内容を図 2 に示した。Phase 1 では、Z80 を CPU に用いた他機種用に開発された PILOT インタプリタの機械語コードを、BASIC 言語で記述したプログラムの DATA として、使用するマイクロコンピュータのメモリーに書き込むためのプログラムを

作成することが主な仕事であった。本稿では、主として Phase 2 のコースウェアの作成段階でのシステムの使用法を解説する。

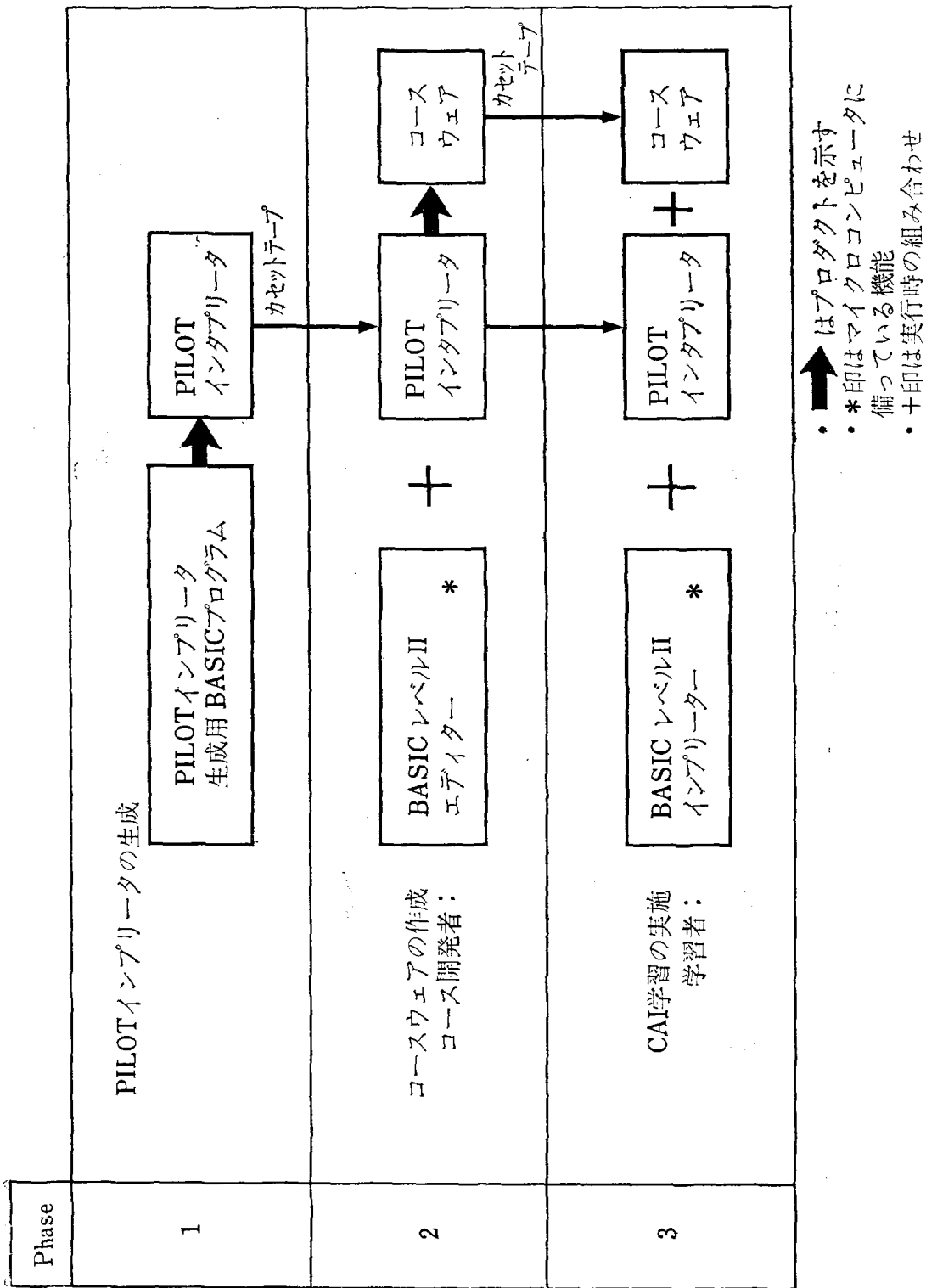


図2 MINI-PILOT/TRSLレベルIIシステム

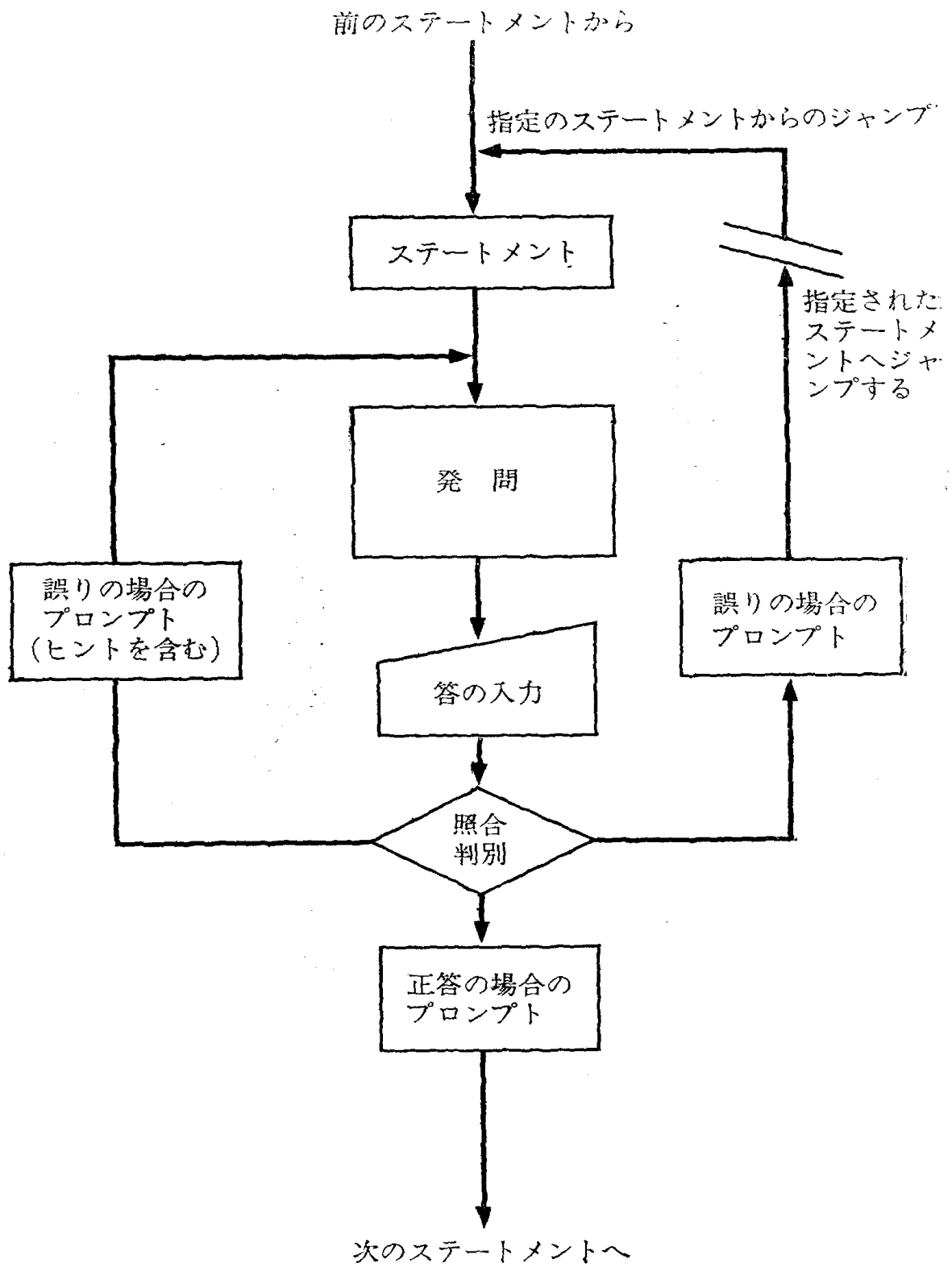


図3 Dialogue Mode の学習指導の流れ

6. MINI-PILOT 言語の機能

先にも述べたように、この言語は問答形式 (Dialogue Mode) の C A I に適している。従って、学習指導の一般的な流れは図 3 のように表わされる。

これをコンピュータの機能として説明すると次のようになる。

1. ステートメントを提示する。
(テキストの提示)
2. 「ディスプレイのスクリーンをクリヤして」発問を提示し、学習者に答えを求める。
(テキスト提示, 入力待ち)
3. 学習者の答の正誤を、予め決めてある正答と比較する。正答の場合は正答を示すプロンプトを表示して次のステートメントの提示へ進む。
(照合判別, プロンプト提示, ジャンプ)
4. 学習者の反応で誤っているものを、予め決めてある誤答例 (複数可) と比較して、一致した場合には誤りの理由 (説明) を提示する。前の発問をもう一度くり返えす場合は 2 にもどる。再学習の必要な場合は指定のステートメントにジャンプする。
(照合判別, プロンプト提示, ジャンプ又はステートメント)

7. ハードウェアシステム

使用したマイクロコンピュータは TRS 80 Model 1 基本システムである。これは CPU (Central Processing Unit) に Z 80 マイクロプロセサーを用い、ROM (Read-Only Memory) 13KB, RAM (Random-Access Memory) 16KB からなるシステムで、CPU 本体はキーボードのケースに内臓されている。拡張インターフェイスをつけると RAM を 16KB 単位で増設でき、システム全体の RAM 容量を最大 48KB までふやすことが可

能であり、また同時に、Diskett Drive（フロッピー・ディスク外部記憶）装置、プリンター等も使えるようになり、大幅に機能がふえるが、本稿で述べるCAIインタプリタを使ったCAIシステムには基本システムのみがあればよい。教材の作成と提示には12インチの白黒あるいはグリーンのビデオディスプレイを用いるが、1行当り64文字表示で16行の標準文字表示の場合は1画面に最大1024文字、また1行32文字表示の大型文字の場合は512文字を表示することができる。

図4にシステムの構成を示した。

なお、TRS 80 モデル1システムは、本学ILC自主学习室にも4台設置されている。

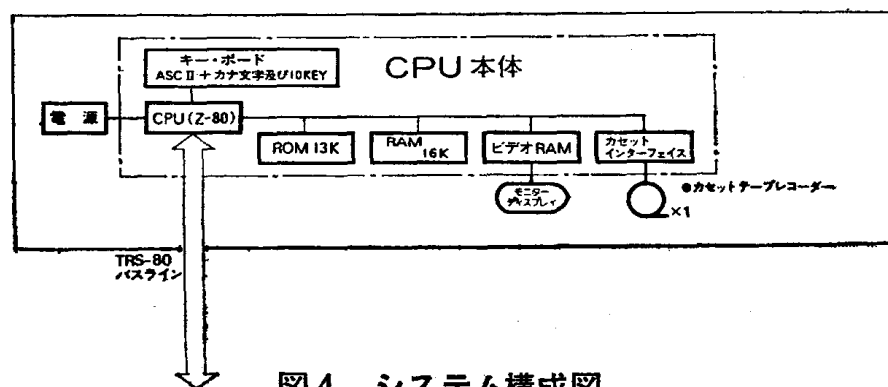


図4 システム構成図

8. コースウェア作成のための機能

コースウェア作成は、コマンドとテキスト、コマンドとプロンプト、あるいはコマンドと正(誤)答の定義を重ねることによって行なう。コマンドは第1表に掲げた7種のアルファベット記号とその組み合わせによる拡張コマンドからなっている。

第1表 コマンドとその機能

| | |
|---|---|
| T | ビデオ・スクリーンにステートメントを提示する。（最大235文字 235字を超えるテキストの提示にはTコマンドを繰り返して使う） |
| A | ビデオ・スクリーンに発問の文あるいは選択肢等を表示し、システムは待ち状態となり、学習者が答を入力するのを待つ。？が自動的に表示され、力入待状態を示す。 |

- M 学習者が入力した答えを、予め決めてある正答のリストと比較し、マッチするかどうか判定する。
MコマンドはAコマンドを組み合わせて使うが、1つのAに対してMがいくつあってもよい。AのみでMが無い場合は、入力されたものは全て誤答とみなされる。
- J 決めてある番号のステップへ分岐する。
- C ビデオ・スクリーンをクリアする。
WのあとでCを使うと、大型文字による表示の終了を意味する。以後の表示は標準サイズの文字による。
- W ビデオスクリーンをクリアして、ステートメントを大型文字で表示する。次のCコマンドに出会うまでの表示はすべて大型文字である。
- YT YフラグがONの場合に表示するステートメント。
- NT NフラグがONの場合に表示するステートメント。
- YJ YフラグがONの場合の分岐番号を示す。
- NJ NフラグがONの場合の分岐番号を示す。
- YC YフラグがONの場合にスクリーンをクリアした上で行う表示を示す。
- NC NフラグがONの場合にスクリーンをクリアした上で行う表示を示す。

9. コマンドの記述

コマンドはステップ番号を最初にもち、次にコマンド記号、3番目に二重引用符（"）をもつ。二重引用符はコマンドをステートメント、定義済みの答（正答、誤答）のリスト、あるいは分岐先番号等と分ける役目をしているが、ステートメントの文章の中で記号本来の目的で使うことも自由である。

第2表にコマンドステートメントの例を示す。

コマンドステートメントの先頭にくるステップ番号には継続した数は使わない方がよく、10, 20, 30, ……あるいは10, 50, 100, ……と任意の間隔をもつ一連の数（但し最大 65529）とする方がよい。例えば10, 20, 30, ……というように10間隔にすると、コースウェアの作成中に教材の論理を変更して20番と30番のステップの間にコマンド・ステートメントを追加す

る必要が生じた場合、最大9個(11, 12, 13, …… , 18, 19)のコマンド・ステートメントを加えることが可能である。

第2表 コマンドステートメントの例

| ス テ ッ プ 番 号 | コ マ ン ド | ステートメント |
|----------------------------|-----------------------------|-------------------|
| 10 | T"THIS IS A SAMPLE PROGRAM. | ……ステートメントの表示 |
| 20 | A"ARE YOU AN ICU STUDENT | ……発問の表示 |
| 30 | M"YES/YES, I AM/YES, /Y | ……定義してある正答リストとの照合 |

10. 編集機能

コースウェアを作成する過程で、タイポグラフィカル・エラーの修正、センテンスの変更、削除、追加などを度々行わなければならない。これらの作業はマイクロコンピュータに備っている BASIC 言語編集用の Editor を用いて簡単に行うことができる。第3表に編集コマンドの主なものを一覧して示したが、TRS-80レベルII BASIC マニュアル第10章に詳細な記述がある。

第3表 編集モードにおけるサブコマンドと記号キーの機能

| サブコマンドまたは記号キー | 機 能 |
|---------------|------------------------------------|
| ENTER | 編集を終了しコマンドモードへ戻る。 |
| SHIFT ↑ | サブコマンドから抜け出す。編集モードのまま。 |
| n 回のスペースパー | カーソルを右に n だけ動かす。 |
| n ← | カーソルを左に n だけ動かす。 |
| L | 行をリストし行の始めに戻る。 |
| X | 行をリストし行の最後に追加文を入れることができる。 |
| I | カーソル位置から文字を挿入できるようにする。もとに戻すには↑を使う。 |
| A | 変更を取り消して行の始めに戻る。 |

| | |
|--------------------------|---|
| E | 編集終了,すべての変更を記録しコマンドモードに戻る。 |
| Q | 編集終了,すべての変更を取り消してコマンドモードに戻る。 |
| H | 現カーソル位置以降を削除して文字を入れることができるようにする。もとに戻すためにはシフト↑を使う。 |
| <i>n</i>D | 指定された <i>n</i> 個の文字を消す。 |
| <i>n</i>C | 指定された <i>n</i> 個の文字を変える。 |
| <i>n</i>S<i>c</i> | <i>n</i> 番目の <i>c</i> 文字の位置までカーソルを動かす。 |
| <i>n</i>K<i>c</i> | <i>n</i> 番目の <i>c</i> 文字の位置までを削除する。 |

11. 教材作成者によるシステムの使用

このシステムを使ってコースウェア作成作業を行う場合, 次のような手順をとる。

- 1 マイクロコンピュータの電源を入れる。図5の①の表示がスクリーンに表われる。PILOT インタプリータのカセットテープをテープレコーダにセットし, PLAY モードにする。
- 2 SYSTEM とタイプし<ENTER>キーを押し, 次のプロンプトに対し PILOT とタイプするとインタプリータの読み込みが始まる。
- 3 テープの読み込みが済むと②の表示が表われる。ステップ番号を自動生成する場合は編集コマンドの AUTO をタイプする。最初のステップ番号が表示され, コマンドステートメントの入力待ちとなり, コース作成を始められる状態となる。ステップ番号を自動生成せず作成者がタイプする場合は, テープの読み込み終了とともに作業にとりかかればよい。
- 4 コースウェアの作成の途中で行うテストラン及びCAI学習時(Phase 3)には, プログラムは NAME コマンドを入力(NAME とタイプ)すれば直ちに実行を開始する。
- 5 作成したコースウェアプログラムあるいは作成中のコースウェアプログラムはレベルⅡBASICのCSAVE, CLOAD コマンドを使うこ

とによって、カセットテープに格納することができる。

```

①
MEMORY SIZE? (hit ENTER key)
RADIO SHACK LEVEL II BASIC
READY
)

②
RADIO SHACK LEVEL II BASIC/PILOT INTERPLETER
READY
)

```

図5 PILOTのスタート表示

12. コースウェアのフレームの作成例

初等統計と英語のコースウェアの例を1フレーム分、PILOT言語でコーディングした場合とBASIC言語でコーディングした場合を対比させて、リスト1, 2及びリスト3, 4に例示した。

```

10 C"
      << SAMPLE PILOT PROGRAM FRAME >>
      HIT (ENTER)

12 A"
14 C"
20 T"      COMPUTE "THE MEDIAN" FOR THE FOLLOWING DATA:
           1, 2, 3, 3, 4, 4, 4, 6, 7, 7, 8, 8, 9, 10, 11

30 A"      (ANSWER)==>
40 M"6
50 YJ"200
50 M"4
70 YT"      THE MOST FREQUENTLY OCCURRING VALUE IS CALLED "THE MODE".
75 YT"      TRY AGAIN.
80 YJ"500
90 M"5.8
100 YT"      THIS FIGURE IS CALLED "THE ARITHMETIC MEAN".
105 YT"      TRY AGAIN.
110 YJ"500
120 NT"      THIS FIGURE IS NOT "THE MEDIAN".
           WHAT IS THE MIDDLE FIGURE OF ALL?

130 NJ"500
200 T"      YES. "THE MEDIAN" IS A VALUE SUCH THAT HALF THE OBSERVATION
205 T"      FALL ABOVE IT AND HALF BELOW IT.
210 A"      HIT (ENTER) TO NEXT
220 J"1000
500 A"      HIT (ENTER)
510 J"14
1000 T"
           ::::::::::: NEXT FRAME :::::::::::
1100 E"

```

リスト1 初等統計のプログラムフレーム例 (PILOT版)

```

10 CLS
20 PRINT"COMPUTE < THE MEDIAN > FOR THE FOLLOWING DATA:
           1,2,3,3,4,4,4,6,7,7,8,8,9,10,11
25 PRINT TAB(50);" ";;INPUT R
30 IF R=6 THEN 200
32 IF R=4 THEN 70
34 IF R=5.8 THEN 100
36 IF (R<>6)AND(R<>4)AND(R<>5.8) THEN 120
70 PRINT"THE MOST FREQUENTLY OCCURRING VALUE IS CALLED <THE MODE>.
           TRY AGAIN."
75 PRINT"
80 GOTO 25
100 PRINT"THIS FIGURE IS CALLED <THE ARITHMETIC MEAN>.
105 PRINT"
           TRY AGAIN."
110 GOTO 25
120 PRINT"THIS FIGURE IS NOT <THE MEDIAN>.
           WHAT IS THE MIDDLE FIGURE OF ALL?"
130 GOTO 25
200 PRINT"YES. <THE MEDIAN> IS A VALUE SUCH THAT HALF THE OBSERVATION
205 PRINT"FALL ABOVE IT AND HALF BELOW IT.
210 PRINT"END"

```

リスト2 初等統計のプログラムフレーム例 (BASIC版)

```

10 C"CHOOSE THE ONE WORD OR PHRASE WHICH CORRECTLY COMPLETES
THE CONVERSATION.
   (FIRST SPEAKER)  "DID YOU REMEMBER MY PHONE NUMBER?"
   (SECOND SPEAKER) "YES I ...."
20 T"
           1. CALL YOU UP YESTERDAY
           2. DID CALL YOU UP YESTERDAY
           3. CALLED UP YOU YESTERDAY
           4. CALLED YOU UP YESTERDAY
30 A"
40 M"4
50 YT"YOU ARE CORRECT..
60 YJ"200
70 M"1
80 YT"YESTERDAY!!           TRY AGAIN.
90 YJ"30
100 M"2
110 YT"NO.   DID + CALL  = ?           TRY AGAIN
120 YJ"30
130 M"3
140 YT"YOUR ANSWER IS NOT RIGHT.
150 YJ"30
160 M"1/2/3/4
170 NT"PLEASE CHOOSE 1,2,3,OR 4
180 NJ"30
200 E"

```

リスト3 英語のプログラムフレーム例 (PILOT版)

```

5 CLS
10 PRINT"CHOOSE THE ONE WORD OR PHRASE WHICH CORRECTLLY COMPLETES
THE CONVERSATION.
    (FIRST SPEAKER)    DID YOU REMEMBER MY PHONE NUMBER?
    (SECOND SPEAKER)  YES I ...."
20 PRINT"
                1.  CALL YOU UP YESTERDAY
                2.  DID CALL YOU UP YESTERDAY
                3.  CALLED UP YOU YESTERDAY
                4.  CALLED YOU UP YESTERDAY
30 PRINT TAB(50);" ":INPUT R$
40 IF R$="1" THEN 80
42 IF R$="2" THEN 110
44 IF R$="3" THEN 140
46 IF R$="4" THEN 50
48 IF (R$="5")OR(R$="6")OR(R$="7")OR(R$="9")OR(R$="0") THEN 170
50 PRINT"YOU ARE CORRECT..
60 GOTO 200
80 PRINT"YESTERDAY!!          TRY AGAIN.
90 GOTO 30
110 PRINT"NO.    DID + CALL  = ?          TRY AGAIN
120 GOTO 30
140 PRINT"YOUR ANSWER IS NOT RIGHT.
150 GOTO 30
170 PRINT"PLEASE CHOOSE 1,2,3,OR 4
180 GOTO 30
200 PRINT"END"

```

リスト4 英語のプログラムフレーム例 (BASIC版)

なお、1982年3月現在、この MINI—PILOT インタプリータ TRS—80 48KB DISK 版の開発を継続して行っている。

参考文献

1. Hawkins, R. PILOT—The Language of Computer Aided Instruction
80 Microcomputing, July 1981
2. Schnyler, J. A. Programing Languages for Microprocesser Courseware
Educational Technology, Oct. 1979
3. TRS-80 レベルII BASIC マニュアル 工学図書出版 昭54
4. マイクロコンピュータ・システム ——TRS-80のハードウェア——
工学図書出版 昭54

The Implementaion of CAI Courseware Authoring/Operating System for Microcomputer

Sugao Ishimoto

The author discussed some basic requirements of CAI language for microcomputer system, and PILOT, a CAI language designed for a minicomputer system, is recommended for use with microcomputer. Adoption of a PILOT interpreter to microcomputer system is briefly touched upon. The procedure of authoring course material is detailed in order to provide information to prospective users of the system.

The CAI authoring/Operating system, which is the product of the present development, is now operational with TRS Model 1 microcomputers at Independent Study Room in ILC and Audiovisual Laboratory in H—155. at ICU