

研究発表 1

JET 98 - 1

日本教育工学会
研究報告集

RESEARCH REPORT

OF JET CONFERENCES

大学教育の改革と遠隔学習

メディア教育開発センター

1998年1月30日(金)

日本教育工学会

文書入力時における状況認知と処理過程

Cognition of Situation and Process of Operation in Using Computer for Word-processing

中野 靖夫

小岩 寿之

Yasuo NAKANO

Toshiyuki KOIWA

上越教育大学

武蔵野市立第五中学校

Joetsu University

Musashino 5th Junior

of Education

High School

あらまし コンピュータを利用した学習は個別化し、学習者毎に異なる。本研究はある学習者に課題文を与え入力させたときの、状況の認知と処理過程を明らかにした。状況の認知は入力直後に操作を再現しインタビューで行い、コンピュータ操作の再現システムで収集したデータで処理過程を解明した。状況の認知は、Norman, D.A. の7段階の認知的活動理論を援用した。

キーワード ワードプロセッサ, 操作過程, 個別化, 思考モデル

はじめに

コンピュータを利用した学習は、人間とシステムの対話によって展開される。学習は個別化し操作過程や学習過程は学習者毎に異なる。したがって、操作履歴あるいは学習履歴を分析し個人特性や集団の特性を明らかにしていくことが教育において新たな課題となろう。学習履歴のデータ収集法の一つに非反応測定法が用いられ、コンピュータ自体に情報を収集させることが可能である。中野ら(1997)は、この方法でキー入力データを収集し追跡・分析することによって、グラフ作成過程、文書作成過程、データベース検索過程、プログラム作成過程等を明らかにしてきた。この研究では、学習者の操作過程やエピソード(操作中の出来事)の抽出を行い個人特性や集団の特性を明らかにするとともに、学習者に内在する状況の認知や適用された方略を推測することができた。

しかし、操作中に学習者は各自の能力に応じて状況を認知し、対策を検討し、処理を行っていくと考えられる。Norman D.A.(1990)は機械・システム使用時の認知モデル過程について、①ゴールの形成②意図の形成、③行為の詳細化、④行為の実行、⑤外界の状況の知覚、⑥外界の状況の解釈、⑦結果の評価という7段階の行為理念を構築した。このモデルはインターフェイス、システムのデザインを支援している。一方、学習者(ユーザー)の行為をこのモデルで詳細に解明していくことは情報教育を進めるうえで意義がある。7段階のうち行為の行為の実行に関しては上述のようにキー操作履歴を収集して操作行為の実体を明らかにしてきているが、さらに研究を深化させるには、学習者の認知を明らかにしていく必要がある。

そこで、本研究はワードプロセッサを学習している初心者を対象に状況の認知と処理過程を解明

することを目的とする。具体的には、ワードプロセッサの学習の初期に行われる課題文の入力時に、入力段階と変換・段階でエピソードが発生したり、操作を変更する事態が発生する。そこに着目し状況の認知と処理過程を明らかにする。なお、本研究の状況の認知とは、Norman D.A.の7段階のうち行為の詳細化、行為の実行、状況の知覚、状況の解釈、結果の評価の段階を統合したものである。

2. 方法

ワードプロセッサの指導を行い、課題文を与え入力させる。この過程を再現システムで収集し、直後に操作を再現して学習者にインタビューを行い状況の認知を明らかにする。

2.1 被験者

中学3年生、家庭にワープロ所有、使用している。

2.2 事前指導

90分、また、授業でワープロを学習している。

2.3 課題文

沖縄本島を、本社機「千早」に乗って上空から見た。島をとりまく海の色は美しい。遠浅の岸辺は、それに緑色を混ぜて淡くした色だ。外の海との境目に、白波の立つサンゴ礁。白い線が、無造作な飾りのように、延々と島の外側を縁どっている。着陸し、その暖かさに驚いた。雪が降った北国もあるというのに、建物や自動車は冷房中である。半そで姿になる。蚊が刺した。快い南風に吹かれて歩く。道端に真っ赤な仏桑華、いわゆるハイビスカスが咲き乱れる。街路樹モクセンナの黄色い花がまぶしい。鳥の羽のような繊細な葉を茂らせたホウオウボクに、長さ三、四十センチもあるブーメランのような実が垂れ下がっている。密

生した分厚いフクギの葉の輝きが、とても十二月とは思えない。恵まれた自然である。(朝日新聞天声人語 1992年12月11日の一部を使用させていただいた)

2.4 データ収集法

(1) 状況認知

文書入力過程を再現し学習者にインタビューを行い、スキャンコンバータとVTRを使用して録画する。

(2) 処理過程

前田ら(1993)の開発したコンピュータの操作過程の再現システムを使用

2.5 分析

学習者の発言と処理過程から変換・確定段階の状況の認知を明らかにする。

3結果

ある生徒のキー操作履歴とインタビューの内容を示す。データにつけた数値は事例のマーキングである。Tは教師の発言、Sは生徒の発言である。

T「これから***君の入力していったのを再現します。いろいろ起こったことを説明してください」 S「はい」

2)okinawahonntouwo SP ↓ SP, SP

T「本島というのを選んだわけね」

3)honnshaki SP SP SP ↓ SP SP SP SP SP CR

T「ここは」 S「本社っていうのを、まず、でたから、次にした(↓)をおして、次に進めて、そこで、変換をおして、機械の機にかえて、リターン」

11) tihaya SP SP SP ↓ → → SP SP SP ← ←
SP SP SP SP CR

T「ここは」 S「ここは違う文字がでたっていうか、一文字しか変換されなかったから、えーと、カーソルキーの右をおして、カーソルをすすめて、それで1個ずつ選んでいった」

6) mita. CR BS BS BS mita. SP SP CR

T「ここは」 S「みたっているところ(見た)がひらがなになっていたからうちなおした」

8) siroi BS BS ro BS ranamino

T「いまのところは」 S「白までしかよんでなかったから、しらなみまでいれようとして」 T「改行ね」

SP SP → → SP SP SP SP SP SP SP SP ← ← SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP ↑ ↑ → → → ← ← ← ← ← → ESC
siro SP SP nami SP no CR

S「白波というのがでてこなかった、やっぱり最初やったみたいに白と波にわけてやりました」 T「全部けして」

9) sanngoshou SP SP SP SP ← ← f·7 → ←
f·7 ↓ SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP

SP SP SP SP SP SP n BS BS

T「ここは1回珊瑚礁と変換したけど」 S「カタカナだったからサンゴまでをまず、変換して」 S「それで」 T「今、一瞬かわったのに、また、変わっちゃったね」 S「今のは、ちょっとおしまちがいです」

T「なにをまちがえたの」 S「カーソルキーの右

おしちゃって、黄色いところが(し)まではいっちゃったんです」 T「範囲の指定をまちがえたわけね」 S「はい」 T「これは、しょう(礁)というじはなかなかでてこなかったわけね」 S「はい」 T「あら、これは」 S「途中でNをおしちゃったみたいです、なんか」 T「しょうという字がでてくる途中で」 S「はい」 T「おしちゃったの」

12) enen BS BS BS ennen to

T「今のところは」 S「えんえんとうとうとしたら、キーボード見て打ったから、えねとうっちゃったわけです」 T「NE」 S「Nを2回おさなかつただけです」

SP SP SP SP SP SP ← ← ← SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP ← → SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP ↓ SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP ← ← → SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP SP SP SP SP SP SP SP SP SP
SP SP ESC no ma SP

SP SP SP SP ESC f·7 f·8 f·9 f·9 f·9 f·9 →
← f·10 f·10 f·10 f·10 ↓ ↓ → → → → CR ESC
ESC ESC ESC ESC ESC ESC CR

S「今度は、これは延々というのがでてこなかったから、1回戻して、えんだけでやってみただけど、一個はでてきたんだけど、ここはあのノマという記号じゃないですか、1回これ全部消してやってみたんですけど」 T「まださがしてるね」 S「こんなにさがしてたかな」 T「今度は」 S「えんだけで変換やってみたんですけど、それでもなかったから」 S「今度は、記号をそのままノマとやってみたんですけど、これもでないから、最後の手段みたいなかんじで、JISコードのやつ

でですことにしました」 T 「これだ」 S 「だし
て、そこで、何かESCキーおしちゃって」 T 「メ
ニューがでてきたわけね」

18)reiboutyuudaruru.SP → → SP SP BS
BS BS BS desa BS aru. SP CR

T 「これはれいぼうちゅうだるになっているよ」
S 「いや、であるがEをぬかしてAを先に入力し
ちゃったからだるになっちゃったの」

19)hannsodesugataninaru.SP SP
SP SP ← SP ← ← SP ↓ SP SP SP SP ← ↓ ↓ .
SP BS BS

T 「いまのは」 S 「はんそですがたで、でるかとおも
ったんだけど、それだとでなかったから、はんそ
でで先ず変換して、それで変換したんだけど、そ
でっていう漢字がでてきちゃったから、先ず、半
だけやって、そではひらがなでそのままにして」
T 「お、まるが多いね」 S 「これはただ、まるを
うったのを忘れていて」

20)kagasaaita SP BS ← ← ← ← ← → → s
i DEL DEL → . SP SP SP SP SP SP SP SP SP SP
SP SP SP SP SP SP CR

T 「これは」 S 「蚊が刺したのところを、Aを2
回おしちゃったみたいで、それでそうなっちゃっ
たみたいです」 T 「Aがはいっちゃったわけね」
S 「AがはいってSがはいっていなかった」 T
「これ、挿入したわけね」 S 「はい」
T 「これは蚊という字をさがしてるの」 S 「はい」

21)S 「あ、まちがえた、指したという字がそのま
まになっている」 T 「本当は、ここはこの字じゃ
ないんだ」

25)gairojumokusennna SP ← SP ↓
SP SP SP SP ← → → → → SP SP f·7

T 「ここは」 S 「がいろじゅで、先ず変換して、
それであとからモクセンナを後からカタカナ変換
にしようとおもったんです」 「結局、変換しても
でないからエフの7番をおしたわけね」

27)nagasa3 BS san, yonnju SP SP SP
SP ↓ SP ↓ SP SP SP ← → → → → → uu → →
u ↑ SP SP SP u ← ← ← BS ESC ← ← ← ← yo
nnjuu SP CRT 「これは」 S 「長さ、3、40
センチってうとうとしたんですけど、よんじゅう
までいれないで、よんじゅでとまちちゃって、ま
ず、「さん」の方を変換して、長さ三になって、
それからさきでてこないから、一回全部消してそ
れで、よんじゅうとうちなおしたんです」

28)b-BSu-merannnoyouma BS na
T 「いまのは」 S 「ゆー (U)をおさないで、その
まものばす (一) をつけちゃったんです」
SP SP SP SP SP f·7 f·7 SP ← SP SP SP f·7 ↓
T 「これは変換が」 S 「[の]までカタカナ変換に
なっちゃったんで、[の]を1回はずしてそれでカ
タカナにして、それで、そのまま」

30)← ← ← ← ← ← ← ← ← ← ← ← ← ← ← ←
← you CR DEL → → → → → → → → → → →
→ → → → → → → → ← ←

T 「今のは」 S 「あれ、そうだ、ようなは漢字
じゃなくてひらがなでよかったんです」 S 「だから
それに気がついて、もどして、もどすために、も
どったんです」

32)haga BS nokagayakiga, SP SP ← ←
SP SP SP

T 「いま、最初、[が]がでてきたね、[はが]になっ
ていたけど」 S 「主語かなとおもってたんですけ
ど、葉のになっていたから[の]にもどしだけ」 T
「ここは、本当はどういうふうになればよかつ
た」 S 「はのかがやきがかいてあるから、どうし

たんだろう」S「[はの]までを変換させて、そうゆうふうにして」

この文章をいれて感じたことを簡単にいってください」S「内容についてですか」T「なんでもいい」S「家にワープロがあるんですけど」T「専用機」S「ワープロ専用のやりかたになれちゃっているから、けっこうつかいづらかったりして」T「変換が」S「はい」T「文章そのものについてはどうですか」T「入力しにくい」S「あまり入力しにくくはなかったけど、どっちかというとう入力しやすかったかな」T「それじゃ、どうも、ご苦労様でした」

4考察

本稿では、ある生徒が課題文を入力した直後に操作を再現し、インタビューにより入力中の作成状況を聞き取り、状況の認知と処理過程の解明を試みた。課題文を入力する場合、オペレータは課題文を読みとり、行為系列を詳細化し文章を切断して音を入力し、変換・確定していく。音の入力に対してシステムは自動的に音(かな)を出力し、音に対応しないキー操作に対しては入力したキーの文字を表示していく。変換・確定はシステムの持つ機能によって出力される文の状態は異なり、オペレータは目的を達成させるために、システムに指示を行い操作を行っていく。音の入力特性はオペレータに依存し、変換・確定はオペレータとシステムの両者に関連がある。変換・確定段階をNormanの7段階の行為に対応させ、抽出できた発言と操作過程から分析していく。なお、ここでは、Norman D.A.の認知モデル過程における③行為の詳細化、④行為の実行、⑤外界の状況の知覚、⑥外界の状況解釈、⑦結果の評価という行為は③詳細化、④実行、⑤知覚、⑥解釈、⑦評価と表現する。「」内は生徒の発言である。[]内に入力された音、表示された文字を示す。

a)一般的な作成過程

この生徒の入力、変換・確定の方法は、文頭の事例2)が示すように文節で入力し変換する。事例2)は逐次変換状態となり[沖縄]を確定し、[本島]を選択し確定した。[、]入力後、変換を行ったが、次の入力が行なわれ、確定している。

事例3)は事例2)と同様、逐次変換状態になった。

④実行:[ほんしゃき]と入力

④実行:変換

⑦評価:「本社っていうのを、まず、でたから」

④実行:「次にした(↓)をおして、次に進めて」[本社]を確定

④実行:「そこで、変換をおして、機械の機にかえて」変換

④実行:「リターン」[機]を確定

[本社]を確定し、[機]を選択し確定した。④実行(音の入力)後、システムの動作に応じた処理が目標達成まで継続した。入力した音によっては(例えば名詞と助詞の組み合わせ等で)変換区間を変更する場合がある。事例9)サンゴ礁、事例25)街路樹、事例32)葉の等の部分で確実に行われている。

事例4)は変換区間の変更が2度行われた。

④実行:[「ちはや」と入力

④実行:変換

⑤知覚:「ここは違う文字がでたっていうか」

⑥解釈:「一文字しか変換されなかったから」

④実行:「を確定

④実行:「カーソルをすすめて」変換区間を[ちはや]に変更

④実行:変換

④実行:「それで1個ずつ」変換区間を「ち」に変更

④実行:「選んでいった」変換

④実行:[千早]を確定

入力は[「ちはや」]である。変換し「を確定する。逐次変換状態であったので、変換区間を単語

の「ちはや」にして変換するが漢字は出力しない。そこで、カーソルを移動し、変換区間を「ち」に変更、変換する。このとき「はや」も「早」に変換された。「千」を選択し、「千早」を確定する。事例19)は文節で変換できるという見込みで操作したが、不成功。そこで、変換範囲を「はんそで」にして変換したところ、「半袖」になる。変換範囲を「はん」にして変換する。変換は、文節、単語、単漢字と細分化している。知覚、解釈、評価の発言は少ないが、行為を細分化して分析すれば④実行後に⑤知覚、⑥解釈、⑦評価が行われ、論理性の高い階層的な操作が行われている。

b)行為系列の変更

事例12)は行為系列が変更された。

- ④実行：[えんえんと]と入力
- ④実行：変換
- ⑥解釈：「延々というのがでてこなかったから」
- ④実行：「1回戻して」変換区間を[えん]に変更
- ④実行：変換、途中カーソル移動
- ⑥解釈：「えんだけでやってみただけど、一個はでてきたんだけど」
- ④実行：延を確定
- ⑥解釈：「ここは、あのノマという記号じゃないですか」
- ④実行：変換
- ④実行：変換区間をえんに変更
- ④実行：変換
- ⑦評価：「えんだけで変換やってみただけど、それでもなかったから」
- ④実行：「1回これ全部消して」消去(行為系列の中断)
- ④実行：ノマと入力(行為系列の変更)
- ④実行：変換
- ⑦評価：「記号をそのままノマとやってみたん

ですけど、これもでないから」

- ④実行：[のま]を消去
- ③詳細化：「今度は、最後の手段みたいなかんじで、JISコードのやつでだすことにしました」
- ④実行：ファンクションキーの操作
- ④実行：カーソルキーの移動
- ④実行：[々]を確定
- ④実行：ESCキーをたたく
- ⑥解釈：「そこで、何かESCキーおしちゃって」
- ④実行：改行キーをたたく

入力は[えんえんと]。変換するが[延々と]は出力しない。変換範囲を[えん]にして変換、[延]を選択、確定する。[えんと]を変換。次に[えん]を変換。[々]は出力しない。[えんと]をESCキーで消去。[のま]と入力し変換。これも不成功。最後にJISコードで出そうと計画。ファンクションキーを探りながら、[々]を作成した。システムに対し主体的に接近している。しかし、「延」を単漢字で確定したので、行為系列は[々]作成へ移行した。変換区間を単語にする操作は行われなかった。最後のESCキー操作は意図のない無意味な操作である。

c)変換不能時の処理過程

事例8)は変換不能時の処理過程である。

- ④実行：[しらなみ]と入力
- ④実行：変換(逐次変換の状態)
- ④実行：カーソルをすすめて、変換区間を[しらなみ]に変更
- ④実行：変換
- ④実行：カーソルをもどして、変換区間を[しら]に変更
- ④実行：変換
- ⑦評価：「白波というのがでてこなかった」
- ④実行：カーソル移動
- ④実行：消去

③詳細化：「やっぱり最初やってみたくて白と波にわけてやりました」

④実行：[しろ]と入力

④実行：変換

④実行：確定

④実行：[なみ]と入力

④実行：変換

④実行：確定

④実行：[の]と入力

④実行：確定

変換を行うと、逐次変換状態となる。変換区間を「しらなみ」にして変換するが、漢字は出力しない。変換区間を「しら」にして変換、辞書の確認2巡。カーソルを移動したが、入力した音を消去。操作系列全体を変更し、[しろ]と入力し、変換、確定。[なみ]と入力し変換・確定。[の]と入力、確定した。操作を細分化し単漢字と[かな]で作成した。論理的に可能な手続きで操作したが、目標を達成できなかつた。「白波というのがでてこなかつた」は目的を達成できなかつたという評価である。「やっぱり最初やってみたくて白と波にわけてやりました」は操作系列の変更であるが、ここでは、この方法しか残されていない。

d)同音異義語の確定

事例9)は「しょう」を変換したが、途中でNをたたき異なる文字が決定されてしまう。「途中でNをおしちやってみたくて、なんか」は状況の解釈であり、操作上の単純な誤りに対する自己の評価であろう。ここでは、[しょう]を入力して、変換・確定した。

事例20)は同音異義語を確定した。

④実行：[かがさした]と入力

④実行：変換

④実行：確定

[蚊が指した]で確定し次の処理に進んだ。評価の欠落である。S「あ、まちがえた、指したという字がそのままになっている」T「本当は、ここ

は、の字じゃないんだ」再生時に気づいた。

e)誤入力の変換

事例18)は異なる音の入力、キーの脱落であるが、変換操作を行い[冷房中]を変換・確定後に誤入力部を消去し、[である]を入力した。

事例27)は誤入力の変換である。

③詳細化：「長さ、3、40センチってうとうとしたんですけど」

④実行：[ながささんよんじゅ]と入力

④実行：変換

④実行：確定 長さ

④実行：「[さん]の方を変換して」変換

④実行：確定 三

⑦評価：「長さ三になって」

④実行：変換

④実行：カーソル移動

④実行：[う]と入力

④実行：カーソル移動

④実行：[う]と入力

④実行：カーソル移動

④実行：変換

④実行：[う]と入力

⑥解釈：「それからさきでてこないから」

④実行：カーソル移動

④実行：「一回全部消して」消去

④実行：「それで、[よんじゅう]とうちなおしたんです」[よんじゅう]と入力

音の入力の途中で変換。逐次変換を行い、最後に[よんじゅう]に修正しようとしたが、失敗し、消去。再入力して、変換確定。誤入力の変換操作は入力後の知覚、解釈、評価に問題があったといえよう。[う]の追加は遅い。また追加の操作は試行錯誤的で説明できない。

f)変換せずに確定

事例6)は変換せずに確定した。

④実行：[みた]と入力

- ④実行：確定
 - ⑦評価：「みたっていうところ（見た）がひらがなになっていたから」
 - ④実行：消去
 - ④実行：「うちなおした」[みた]と入力
 - ④実行：変換
 - ④実行：確定
- 確定段階の直後に状況を⑤知覚⑦評価し、再度「みた」と入力し変換・確定した。知覚は早い。

g)不要な変換

事例28)は[ブーメランのような]を[ブーメランの様な]で確定した。

- ④実行：[ぶーめらんのような]と入力
- ④実行：変換
- ④実行：変換 f・7（カナになる）
- ④実行：変換
- ④実行：変換区間を[ぶーめらん]に変更
- ④実行：変換
- ④実行：変換 f・7
- ④実行：確定

変換したが、カタカナにならない。ファンクションキーで変換。変換区間を「ぶーめらん」にして、再度変換、ファンクションキーでカタカナにする。[ブーメランの]を作成中に、[ような]は[様な]になる。ブーメランを確定後、次の音の入力が行われた。評価が欠落した。修正は次の文の入力中に行われた。

h)入力した文の修正

事例30)において誤りを生じた文を修正した。

「あれ、そうだ、ようなは漢字じゃなくてひらがなでよかったんです」「だからそれに気がついて、もどして、もどすために、もどったんです」知覚、解釈、評価は作業を行っている箇所だけでなく、画面上のモニタリングも行われていた。

以上、ある生徒の状況認知と処理過程を明らかにした。操作過程にインタビューの発言を加えることより処理過程がより明細化された。ワープロの操作になれており、文節で入力し、システムの状態を知覚、適切な解釈を行い詳細化で決めた手続きで達成できるように階層的、分析的操作を行っている。変換の誤りや、理解できない操作も存在するが、システムに接近するような処理過程である。

5おわりに

システムに対するユーザーの応答の分析はインターフェイスの研究やシステムの評価を行うためとユーザーサイドの行動に視点を置く研究がある。本研究は教育の視点からユーザーの認知と処理過程に着目した。今回は被験者1名について分析したが、さらに他の被験者の分析を進めている。生徒の操作過程の分析はシステムの理解や処理方略、分析、評価の視点や力を見極める一つの手だてとなろう。

なお、本研究は科学研究費補助金、基盤研究(C)課題番号09680209の研究成果の一部である。

参考文献

- Norman D.A,野島久雄訳(1990)：誰のためのデザイン，新曜社，pp.74-80
- 海保博之，原田悦子，黒須正明(1991)：認知的インターフェイス，新曜社，pp.72-75
- 中野靖夫(1997)：児童・生徒の情報処理過程に関する実証的研究，科学研究費補助金研究成果報告書
- 前田恵三，中野靖夫(1993)：コンピュータ操作の再現システム，日本教育工学雑誌Vol.16 No.4,pp.185-195

日本教育工学会
研究報告集

RESEARCH REPORT

OF JET CONFERENCES

社会が求める新しい能力開発

ECCコンピュータ専門学校

1998年3月21日(土)

日本教育工学会

プログラム作成過程の分析法

An Analysis of Programming Process

駿河 秀一* 前田 恵三** 井澤 英悦*** 南部 昌敏* 中野 靖夫*
 Syuuichi SURUGA Keizo MAEDA Eietsu IZAWA Masatosi NANBU Yasuo NAKANO

*上越教育大学 Joetu University of Education
 **東京工業高等専門学校 Tokyo National College of Technology
 ***山形県立北村山高等学校 Yamagata prefectural KITAMURAYAMA High School

コンピュータ操作過程の再現システムにより収集されたプログラム作成過程のキー入力データのカテゴリ別作成過程表示とキー入力データをカテゴリ分析データに変換するプログラムを作成した。再現システム、カテゴリ分析システム及び開発したプログラムの統合により、プログラムの作成過程及び計量的データの分析を可能にした。

キーワード：プログラミング、システム開発、学習分析、学習履歴

1.はじめに

学校教育の場でプログラミングの様相を観察してみると、学習者がさまざまな方法によって問題解決に取り組んでおり、プログラム作成過程には作成者個々の論理展開、思考、模索が表出してくる。

しかし、一般に指導者は作成終了時に学習者が提出したリストや動作状況をチェックしたりテストによって理解度やプログラム構築力を評価している。その時点では学習結果は把握、評価できるが、問題解決に取り組む際の作成過程は十分に把握することはできない。

作成過程は作成者とシステムの対話であり、モニタリングやコントロールの過程が存在し記述過程に大きな影響を及ぼす。したがって完成したプログラムを評価すると同時に、作成者個人の作成過程を明らかにし指導や評価に反映させていくことが重要であると考えられる。

プログラム作成過程に着目し動的データを分析した研究として、岡本ら(1992)が診断助言型のITSを用いて分析した事例がある。この研究はメンタルモデル解明のため、机上でコーディングを行いながら

作業内容を口頭で説明させ、これをテープレコーダーで録音しオンライン・プロトコルを採取している。しかし、言語プロセッサとの会話したときのありのままの作成過程の分析は行われていない。プログラム作成過程を分析するには、作成過程のデータを収集することが不可欠で、櫻井ら(1996)はプログラム作成中のキー入力データとエラーコードを収集し、タイピング速度の分析を行い学習状況を把握するための要因を明らかにしている。前田ら(1993)は、コンピュータ操作過程の再現システムを開発した。このシステムによって作成過程を再現し、観察できるようになり、キー操作履歴データを収集できるようになった。近藤ら(1994)は、前田らのシステムを活用し、プログラム作成中のキー入力データを収集しエピソードを抽出するとともにカテゴリ分析を行い集団の傾向を明らかにしているが個別的分析結果は示されていない。前田ら(1995)は高専生のC言語のプログラミング過程を個別に解説し、構築方法及び記述に関するカテゴリ分析を行っている。

プログラム作成に関し、分析の対象になるのはプログラムの作成過程やコマンドキーの操作及び計量的なデータである。

前田らが開発したコンピュータ操作過程の再現システムでは、観察によって作成過程の分析が可能である。このデータを処理し、テキスト形式で出力すれば、さらに分析が効率的になる。一方、計量的データの分析は、従来から使用されているカテゴリ分析を利用すれば、作成過程の時系列や状態遷移等を解明できる。したがって再現システムとカテゴリ分析システムを統合すれば、収集したデータを効率的・統合的に分析できるようになる。ただし、それには中継するプログラムが必要になる。そこで、キー入力データを10個のカテゴリに分類してテキスト形式で出力する機能及びカテゴリ分析システムの入力データに変換する機能を有するプログラムを作成した。この3種のプログラムの統合によってプログラムの作成過程と計量的データの分析を可能にした。

2. プログラムの構成

2.1 コンピュータ操作過程の再現システム

プログラム作成過程を分析するには、キー入力データの収集が不可欠である。キー入力データは、前田らが開発したコンピュータの操作過程再現システムを用いる。再現システムは収集したデータを自動的にアプリケーションシステムに再入力することによりキー動作の再現を行っている。

データ収集の概要は以下の通りである。キー入力後のキーデータをキーボードドライバが、キー入力リングバッファに格納するとともにヒストリ記録用テーブルにキーデータと操作時間を転送する。そしてアプリケーションシステムを終了する際に、ヒストリ記録用テーブルのデータを保存プログラムによって外部装置に記憶する。

作成過程の再現は、再現プログラムを起動し、外部記憶装置に記憶させたデータを再びヒストリ記録用テーブルに呼び出す。キーボードドライバは再現モードに切り替わり、データがBIOS、MS-DOSを介してアプリケーションソフトウェアに転送される。そしてCRT上で再現される。

キー入力データを外部記憶装置に記憶する形式は、バイナリー形式であるが出力プログラムでCRT、

プリンタあるいは外部記憶装置にテキスト形式で出力できる。出力されたデータ形式を図1に示す。

| | |
|----------------------------|----------|
| Date count : | 871 |
| Shift key group status --- | |
| SHIFT: | OFF |
| CAPS: | OFF |
| カ: | OFF |
| GRPH: | OFF |
| CTRL: | OFF |
| 2 | : 0.950 |
| 0 | : 2.525 |
| SP | : 14.375 |
| i | : 3.450 |
| n | : 8.650 |
| p | : 1.100 |
| u | : 1.075 |
| t | : 4.325 |
| " | : 2.175 |
| カ | : 8.300 |
| テ | : 3.100 |

図1 テキスト型に変換されたキー入力データ

2.2 カテゴリ分析システム

カテゴリ分析システムは、南部・井沢が開発したカテゴリ分析システムを用いる。このシステムはカテゴリ分析のデータを入力し、カテゴリ分析を行い、結果を一覧表とグラフに表示する機能を有する。データ入力にはキーボードからカテゴリ及び時系列データを入力する方法とエディタで作成されたデータファイルを読み込む方法の2つがある。

分析処理の機能を以下に示す。

- 1) カテゴリ・データの表示 : カテゴリとカテゴリ毎の操作時間を順次表示する。
- 2) 時系列グラフの表示 : カテゴリを縦軸に操作時間を横軸にとり時系列カテゴリを遷移を表示する。
- 3) カテゴリのドット表示 : 単数及び複数の時系列カテゴリの遷移をカラードットで表示する。
- 4) カテゴリの時間集計表示 : 総操作時間内のカテゴリ毎の操作時間割合を一覧表で表示する。
- 5) カテゴリの度数集計表示 : 総カテゴリ度数内のカテゴリの割合を一覧表で表示する。
- 6) カテゴリの前後出現マトリックス : 選択したカテゴリの前後関係をマトリックス表示する。
- 7) カテゴリパターン1構造表示 : 選択したカテゴリから選択したカテゴリへの系列の構造が表示される。例: 選択 RN->SV / 表示 RN(0:4:18)-ED

-IN-ED-SV(0:5:06)

- 8) カテゴリパターン2構造表示 : カテゴリと前後の範囲 $s \cdot t$ を指定し、指定したカテゴリを中心とする前 s 個・後 t 個のカテゴリの系列の構造が表示できる。
- 9) カテゴリ時間集計の割合のレーダーグラフ : 総時間内でのカテゴリ毎の操作時間割合をレーダーグラフで表示する。
- 10) カテゴリ度数集計の割合のレーダーグラフ : 総カテゴリ度数内のカテゴリの割合をレーダーグラフで表示する。
- 11) 時間変更 : データ処理する際の処理対象時間の変更。データの時間変更ではない。
- 12) カテゴリの統合 : データ処理する際のカテゴリを自動的に統合する。ただし、統合されたカテゴリは戻らない。
- 13) プリンタによる印刷 : 1)~10)までの処理で表示される一覧表やグラフをプリント出力する。ただし、レーダーグラフ等はハードコピーされる。
- 14) CHART用ファイル作成モード : 4)カテゴリの時間集計と5)カテゴリ度数集計のデータをCHART.COMのプログラムでグラフ化するようにモード変換する。

2種類のデータを読み込むことが可能であり、また2種類のデータが同じカテゴリであれば、2)の処理において時系列グラフを重ねて表示でき、3)の処理においては同画面上で表示でき、どちらの場合でも比較することが容易である。

2.3 中継プログラム

これまでコンピュータ操作過程の再現システムを使用し、観察法により作成過程の確認を行ってきた。しかし、データの内容が多量なときに、データ記述の負担が大きくなる。そこで、記憶されているキー入力や操作時間のデータを自動的にカテゴリ毎に分類し、そのカテゴリ毎に分類された内容を出力するとともに、キー入力データをカテゴリ分析システム用のデータに変換するプログラムを作成した。このプログラムは当面、言語プロセッサとしてBAS

ICを対象にしており作成過程のデータを以下のカテゴリで分析する。

- IN : 入力 (プログラムの入力、4文字以内の修正及び4個以内のカーソル移動を含む)
- TH : 思考 (30秒以上の入力がない)
- DL : 削除 (5文字以上の削除)
- ED : 編集 (5個以上のカーソル移動)
- LI : リスト (LISTキー[f・4]をたたく)
- RN : 実行 (RUNキー[f・5]をたたく)
- SV : 保存 (SAVEキー[f・6]をたたく)
- HO : HOME (HOMEキーをたたき画面をクリアする)
- MO : 不要操作 (RLUP, RLDNキーをたたく)
- EN : 終了

(1) カテゴリ別作成過程の出力

再現システムにより収集されたデータは、出力プログラムによりテキスト形式で出力できる。テキスト形式のデータファイルを呼び出し、順次キー入力データと操作時間をカテゴリ分類していく。はじめに操作時間が31秒をこえていれば操作時間を思考時間と考え、カテゴリは<TH>、カテゴリ内操作時間は(操作時間-1秒)で、1秒はキー操作時間として処理する。キー入力データがf・4、f・5、f・6、HOME、RLUP及びRLDNの場合、それぞれ、<RN>、<SV>、<HO>、<MO>のカテゴリに分類する。カテゴリ内操作時間はキー操作時間であり、キーデータ数は1である。キー入力データが削除キー(DEL・BS)の場合は、カテゴリ<DL>と<IN>を区別するために削除キー個数をカウントする。カウントを続け、次キーで削除キー以外のキー入力データのときに、カウントした値が5以上であればカテゴリは<DL>、そのカテゴリ内操作時間を計算する。また、カウントした値が4以下であればカテゴリは<IN>とし、そのカテゴリ内操作時間を計算する。キー入力データが編集キー(SP及び方向キー)である場合は、カテゴリ<ED>と<IN>を区別するために編集キー個数をカウントする。カウントを続け、次キーで編集キー以外のキー入力データのときに、カウントした値が5以上であればカテゴリは<ED>、そのカテゴリ内操作時間を計算する。

またカウントした値が4以下であればカテゴリは<IN>とし、そのカテゴリ内操作時間を計算する。以上の分類に当てはまらないキー入力データの場合は、カテゴリ<IN>と分類し、キーデータ数をカウントする。そのカテゴリ内操作時間の計算は、次キーがカテゴリ<IN>以外のキー入力データの場合に行う。以上の分類中にカテゴリが重なるときは、カテゴリ内操作時間とキーデータ数は常に合計される。すべてのキー入力データに対し以上の処理を行うことによりデータの入力順にカテゴリ、カテゴリ内操作時間、キーデータ数が計量できる。

出力は順次処理されたカテゴリとカテゴリ内操作時間をプリンタに表示する。次にカウントしたキーデータ数だけ再度キー入力データを読み出し順次表示していく。その際、キー操作時間が10秒・20秒以上の場合を示すために、キー入力データの出力の前にアンダーラインを10秒の時は1本、20秒の時は2本表示する。またキー入力データの出力は30字またはキャリッジリターンのとき改行する。図2に出力の形式を示す。

| | | | |
|------|------|----------------|--|
| <TH> | 32.5 | | |
| <IN> | 65.4 | 10 INPUT SP _ | |
| | | 20 PRNBS INT A | |
| ⋮ | ⋮ | ⋮ | |
| ⋮ | ⋮ | ⋮ | |
| ⋮ | ⋮ | ⋮ | |
| カテゴリ | 操作時間 | キー入力データ | |

図2 作成過程のプリンタ出力

(2) カテゴリ分析用データの出力

カテゴリ分析システムのデータ入力形式を図3に示す。まずカテゴリ分析システムに必要なデータはカテゴリファイル名、カテゴリの数、カテゴリ、通し番号、カテゴリ操作終了時間データである。そこで、カテゴリ別作成過程のカテゴリ内操作時間に対し、以下の処理を行う。カテゴリ分析は時系列で処理するために操作時間は累計し、カテゴリ終了時間を出力する。ただし、カテゴリ操作時間のデータ形式は分秒で読みとるので、操作時間データを秒から分秒に変更し、累計していく。処理した結果をテキ

スト形式でフロッピーディスクに出力する。以上の処理によってカテゴリ分析システムのためのデータファイルが作成される。

| | | | | |
|-------------|------------|-----------|----------------------------|-------------|
| プログラム | ⋯⋯ | カテゴリファイル名 | | |
| 10 | th | ⋯⋯ | カテゴリ数 10個 | |
| ln | ⋮ | ⋮ | | |
| ⋮ | ⋮ | ⋮ | ⋯⋯ | 10個のカテゴリコード |
| en | 10, th, 36 | ⋯⋯ | 通し番号, カテゴリ, カテゴリ内操作時間 (累計) | |
| 20, ln, 148 | | | | |

図3 カテゴリ分析システムのデータ入力形式

3 分析事例

3.1 課題

ある旅館の宿泊料金は1人9000円です。この旅館には団体旅行者のために基本料金を15名以上で2割引、25人以上で2割5分引き、30名以上で3割引と設定された割引がある。このときの団体旅行者N人の宿泊料金の合計を求めるプログラムを作りなさい。

3.2 プログラム作成過程

図4にカテゴリ別作成過程を示す。

作成者は初めにNEWコマンドで初期化し、100行目でプログラムに課題の見出しを付けている。110行目で宿泊人数Nを求める。120行目において変数入力の際、SYUと入力したが消去、次にSYUKUHAと入力したがHAを消去し、変数はSYUKUとした。130行目では15人以上のときの宿泊料金を求めているが、割引率を入力する際、0.02と入力したが、0.2と修正した。文番号170を入力する際、誤入力があるが、すぐに修正した。実行の際、データ入力を25と入力したが20と修正した。人数20は計算確認を容易にする。実行し、計算された値は一人分の宿泊料金であった。直ちに人数分の宿泊料金を求める計算値になるようにプログラムに追加を行っている。120行目の追加の際に*と挿入するはずが、SHIFTキーを押さず:Nと入力したため、:を修正した。また160行目を追加する際に(を挿入するときにINSキーを押さずに(を入力したため、数式中の9が消去した。直ちにINSキーを押し、9を入力をした。


```

<IN> 18.4  N E W CR
          1 0 0 SP ' T O I 2 CR
          1 1 0 SP

<TH> 36.5
<IN> 71.6  I N P U T SP " N I N Z U _ = " ; N CR
          1 2 0 SP I F SP N

<TH> 35.8
<IN> 273.3 < 1 5 SP T H E N SP _ S Y U _ BS BS BS _ S Y U K U H A BS BS = 9 0
          0 0 CR
          1 3 0 SP I F SP N _ > = 1 5 SP T H E N SP S Y U K U = _ 9 0 0 0
          - 9 0 0 0 * 0 . 0 2 BS BS 2 _ CR
          1 4 0 SP I F SP N > = 2 5 SP T H E N SP S Y U K U = 9 0 0 0 -
          9 0 0 0 * 0 . 2 5 CR
          1 5 0 SP I F SP N > = 3 0 SP T H E N SP S Y U K U = 9 0 0 0 -
          0 0 0 * 0 . 3 CR
          1 6 0 SP P R I N T SP " S Y U K U H A K U H I = " ; S Y U K U
          ] BS CR
          1 6 BS 8 7 BS BS 7 0 SP E N D CR

<RN> 1.2  f·5
<IN> 5.7  2 5 BS 0 CR
<ED> 12.4  ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ → → → → → → → → → → → → → → → → →
          → → → → → → → → → → ← ←

<IN> 4.6  : N ← ← * CR
<ED> 3.3  → → → → → → → → → → → → → → → → → → → → → →
<IN> 2.3  INS (
<ED> 2.9  → → → → → → → → → → → → → → → → → → → → → →
<IN> 5.7  ) * N CR
<ED> 2.0  → → → → → → → → → → → → → → → → → → → → → →
<IN> 7.0  INS (
<ED> 2.6  → → → → → → → → → → → → → → → → → → → → → →
<IN> 4.6  ) * N CR
<ED> 2.3  → → → → → → → → → → → → → → → → → → → → → → ←
<IN> 3.9  ( INS 9
<ED> 2.5  → → → → → → → → → → → → → → → → → → → → → →
<IN> 6.0  ) * N CR
          CR
          CR

<RN> 1.4  f·5
<IN> 3.6  2 0 CR
<SV> 22.0  _ _ f·6
<IN> 12.6  B : BS BS T O I 2 . B A S "

```

図4 作成過程の出力

追加後、再度実行を1回だけ行い、プログラムを保存している。全体でみるとプログラムは1人分の宿泊料金を計算し、次にN人分の宿泊料金を計算し完成した。プログラム作成中はキー入力の誤りに対し、直ちに気づき修正されている。常時入力後のモニタリングが行われていると考えられる。

3.3 計量的な分析

カテゴリ分析システムでプログラム作成過程を分析する。図5はカテゴリ分析による時系列表示である。図式表現により作成系列がより明確になる。

カテゴリ<TH>の遷移を分析すると<TH>は作成過程の前半に2回行っているが後半では一度も行われて

いない。つまり、この作成者はプログラム作成の前半部で思考時間をとり、完成させている。

カテゴリ<ED>の遷移を分析すると後半に集中している。それは実行後の追加作業であり、<ED>と<IN>の繰り返しが7回起こっている。

処理4)のカテゴリ時間集計表示を行った。操作時間の総計は9分4秒であった。カテゴリ<IN>は6分59秒(77%)である。カテゴリ<TH>は1分12秒(13.2%)、カテゴリ<ED>は29秒(5.3%)、カテゴリ<SV>は22秒(4.0%)、カテゴリ<RN>は2秒(0.4%)であった。思考と入力に関しては思考時間は短く、入力の時間が長い。

処理5)のカテゴリ度数集計表示を行った。カテ

参考文献

- 1) 岡本敏雄、安田恭一郎(1992) C言語プログラミングモデルの分析—診断助言型のITSを用いて、日本教育工学雑誌16-3, pp.119-130
- 2) 近藤智嗣、中野靖夫(1994) Logoプログラム作成過程の分析、電子情報通信学会技術研究報告、E T94-65, pp.31-36
- 3) 櫻井桂一、三輪和久、岡田 稔、熊谷 毅(1996) 時系列キー入力データの分析結果に対する一研究、日本教育工学雑誌20-3, pp.179-185
- 4) 前田恵三、中野靖夫(1993) コンピュータ操作過程の再現システム、日本教育工学雑誌16-4, pp.185-195
- 5) 前田恵三、中野靖夫(1995) プログラム作成過程の分析、日本教育工学雑誌19-3, pp.171-180

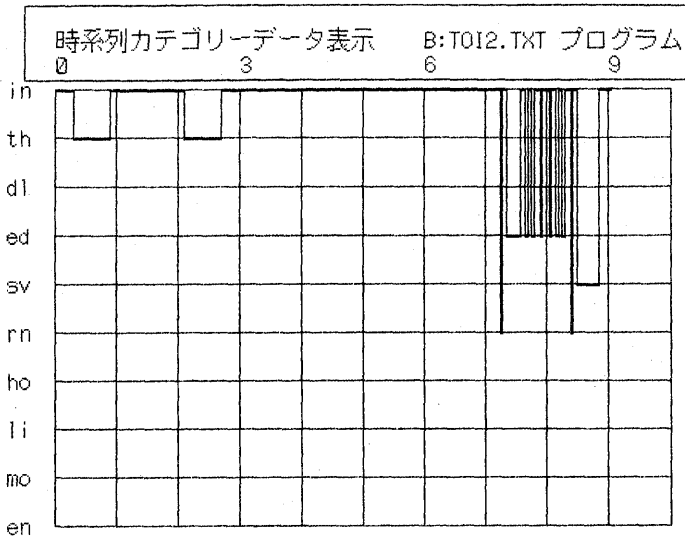


図5 時系列カテゴリーの表示

ゴリの総出現度数は25であった。その内訳は<IN>が13(52%)、<TH>は2(8%)、<ED>は7(28%)、<SV>は1(4%)、<RN>は2(8%)であった。

4. おわりに

この方法によりBASIC言語のプログラム作成過程の分析は効率的になり、多量の客観的データの出力が可能になった。この結果、作成過程と計画的データを照合が可能となりプロセス解明は明確になった。C、PASCAL等の履歴分析はBASICとは処理が異なる。したがって、他言語の分析処理が行えるよう、今後システムを拡張していく予定である。さらに、プログラミングの分析を質的に深化させるには質問や発活思考法等によって作成者の思考、解釈、意志決定等を抽出し、多次元の分析が必要となる。

日本教育工学会
研究報告集

RESEARCH REPORT

OF JET CONFERENCES

外国語・日本語教育への教育工学的アプローチ

仙台市戦災復興記念館

1998年11月14日(土)

日本教育工学会

文書修正における状況認知と処理過程

Cognition of Situation and Process of Operation in Using Computer for Word-processing

中野 靖夫

Yasuo NAKANO

上越教育大学

Joetsu University of Education

小岩 寿之

Toshiyuki KOIWA

武蔵野市立第五中学校

Musashino 5th Junior High School

あらまし 情報化社会では多くの情報が伝達される。ここではワードプロセッサで作成された誤まりの文章を生徒に与え、修正させた場合の修正時の状況の認知とその処理過程を明らかにすることを試みた。

キーワード 情報教育, ワードプロセッサ, 状況認知, 処理過程, 学習反応

1. はじめに

コンピュータは双方向性を有するシステムであり、ユーザーはシステムと対話しながら操作を進めていく。学校教育においてもコンピュータの設置が進み、新しい学習環境として活用されている。学習過程は個別化され、学習の方法、内容は学習者毎に異なる。そこで、コンピュータを使用した学習においては、学習過程のモニタリングが重要な課題となる。

一方、情報活用に関しては情報を収集し、引用加工していくことが多い。また協同学習においても文字情報の情報交換が行われている。文字による表現は記述した人の知識により異なり誤りを生じる機会がある。また、最近これらの文章はワードプロセッサを使用して書くことが多く、操作によって誤りを生じることがある。情報を活用する場合は、その内容を正しく理解すること、引用や活用を行う場合には、誤った箇所を修正する必要がある。

文章の修正に関しては、中野ら(1997)による研究で処理過程を解明した。しかし、この研究においては修正者の内的な活動を収集していないため修正箇所をどのように認知して、処理を行ったかが明らかでない。

そこで、本研究においては、ワードプロセッサで作成した誤文を生徒に与え、修正させ、その修正過程を再現してみせながら修正箇所においてインタビューを行い、状況の認知と処理過程を明らかにすることを試みた。

2. 方法

生徒にワードプロセッサで作成した文中に誤りがある文章を入力したフロッピーを与え、ワードプロセッサによって修正させる。その修正過程を

記録しておき、修正後にこの修正過程を再現して見せながら、インタビューを行い修正時の状況の認知と修正過程を明らかにする。

2.1 被験者

中学3年生 ワードプロセッサを学習した生徒(初学者8名)。以下、生徒を識別するために31から38までの番号を付しておく。

2.2 課題文

「沖縄本島を、本社記(1)「千早」に乗って上空から見た。衝撃を受けたのは、濁った川から赤土が海に流れ出ている眺めである。島をとりまくように(2)の色は美しい。紺屋のつばの中をのぞいたような、鮮やかな青が、果てしなく広がって(3)いる。遠浅の岸辺は、それに緑色を混ぜて淡くした色だ。外の海との境目に、白波の立つサンゴ礁。白い線が、無造作な飾りのように、炎々(4)と島の外側を縁(5)どっている。その光景がみごとなだけに、赤土の流出が無残に見える。川の水は、ミルクを多めに入れたコーヒーのようだ。それが青緑色の海に入って、扇子のように広がる。海汚れる(6)様子が一目両全(7)である。山や畑の上も、さぞ菓子(8)減ってゆくことだろう。上から見ると、細長い本島は実に幅が狭い。雨が降れば水は山から海岸に流れ落ちる。林の伐採、道路づくり(9)などの開発は、赤土流出の一因だろう。整備された排水溝に雨水と土が流れ込んで海に向かうのも、川がコーヒー色になる結果(10)だという。水流の途中に土を沈殿させる池を造ってもお(11)、ある程度たまれば流れ出す道理である。赤土が流れ込むと、海はオニヒトデが育ちやすい栄養過多の環境となる。オニヒトデ(12)によるサンゴの被害も進み、うみ(13)は大きな被害を受けているようだ。」

「天声人語」(朝日新聞1992年12月11日の一部

を使用させていただいた)

誤りの箇所を()付き番号で示した。なお、(1)~(13)は以下に示す誤りである。(1)変換・確定の誤り、(2)入力誤り、未変換、(3)入力誤り、(4)変換・確定の誤り、(5)類似の漢字、(6)入力ミス、(7)変換・確定の誤り、(8)不要な変換、(9)異なる音の入力、(10)反対語、(11)キータッチミス、(12)入力ミス、(13)未変換

2.3 事前の教示

「いま、ある人が文章を入力したフロッピーを渡します。この中の文章には誤りがあります、気がついたところをなおしてください」

2.4 修正

誤文を入力したフロッピーを与えワードプロセッサで修正させた。

2.5 修正過程のデータ収集法

前田ら(1993)が開発したコンピュータ操作過程の再現システムを利用

2.6 操作の再現と状況に関する認知のデータ収集

再現システムにより修正過程を再現し、修正個所でインタビューを行い状況を聞いた。修正時点で「ここは？」と問い合わせ、発言させた。この過程をスキャンコンバータを通しビデオテープに記録した。

2.7 分析法

修正過程と再現時の発言を統合し、状況の認知と処理過程を明らかにする。

3. 結果と考察

3.1 処理順序

修正箇所に順序性のある(文頭から文末に向かって修正した)生徒は、31, 32, 35, 36, 37, 38

である。発言によると、生徒31「最初から読んで」、生徒35「先ず、最初読んで、何かおかしいなと思ったところを直して」、生徒36「もう一度、読み直す」、生徒37「まず、頭の中で読んでいって、おかしいと思ったら、ここはちがうなと思って」である。先ず、文章を読み疑問点で修正を行っているので、文頭からの順序性のある修正になった。

生徒33「最初から読んでいく。変だなと思ったら直して」という発言がある。修正は順序性があるが、最後の修正は文の中段で行われている。これは文章を見直していると考えられる。生徒34は前半、順序性がある。しかし、後半は順序性はない。この実験は「おもしろい」という発言があり、また「3行目に間違っていたのがあって、そこを直そうと思って」「こっちの方を見ようと思って」「とりあえず目についたから」など画面上を系統性を持たずに修正している。このような修正は個人の特性といえよう。

3.2 正答数

各生徒の正答数、誤りの箇所別正答数を図1にSP表で示す。SP表は段階的であり、問題に難易度がある。

正答数の多い修正箇所を以下に示す。()内の数字が正答数である。

さぞ菓子(8)、海汚れる(7)、ずくり(7)、広がって(7)

これらは、言語的には、かな文字に関する誤りである。

正答数の少ない修正箇所を以下に示す。()内の数字が正答数である。

結果(0)、一目両全(2)、うに(3)、オヒニトデ(3)

これらは言語的には意味的な誤り(反対語)、漢字の誤り、綴りの誤りである。

| dataname=skan | | | | | | | | | | | | | | | TOTAL | CARE | | |
|---------------|----|------|------|-------|-------|------|------|------|------|------|------|-------|------|------|-------|------|----|--------|
| SN | PN | 8 | 6 | 9 | 3 | 11 | 1 | 4 | 5 | 13 | 12 | 2 | 7 | 10 | | | | |
| 35 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | * | 0 | 0 | @ | 0 | 10 | 0.00 |
| 38 | | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | * | 0 | 1 | @ | 0 | 10 | 0.32 |
| 33 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | * | 0 | 0 | @ | 1 | 0 | 9 | 0.18 |
| 32 | | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | @ | 0 | 1 | 0 | 0 | 0 | 9 | 0.27 |
| 36 | | 1 | 1 | 1 | 0 | 0 | 1 | 1 | @ | 0 | * | 1 | 1 | 1 | 0 | 0 | 8 | ? 0.59 |
| 34 | | 1 | 1 | 0 | 1 | 1 | 1 | 1 | @ | 1 | * | 0 | 0 | 1 | 0 | 0 | 7 | 0.32 |
| 37 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | @ | 1 | * | 0 | 0 | 0 | 0 | 0 | 6 | 0.00 |
| 31 | | 1 | @ | 0 | 1 | * | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0.00 |
| TOTAL | | 8 | 7 | 7 | 7 | 6 | 6 | 5 | 4 | 4 | 3 | 3 | 2 | 0 | | | 62 | |
| CARE | | 0.00 | 0.00 | !0.84 | !1.05 | 0.31 | 0.46 | 0.41 | 0.00 | 0.14 | 0.17 | !0.87 | 0.22 | 0.00 | | | | |

different=0.115

図1. SP表

3.3 状況の認知と処理過程

課題文が与えられ、文章を読解していくとき誤りを発見する。そのとき生徒は状況を認知し修正を行う。インタビューは修正後、再現時点に行ったが、修正に至る過程、誤りの指摘、修正箇所、文字の推測、修正方法等が述べられた。以下に、正答数の多い誤り4箇所、正答数の少ない誤り4箇所について検討を加える。データは生徒の番号および再現時の初発の発言（「」に示す）である。発言は文章化した。発言の後の（ ）内は発言内容の要約である。

さぞ菓子(8)

生徒31「菓子が違う」（誤りのある文字を指摘）、生徒32「「かし」が漢字になっている」（「かな」が漢字になっている）、生徒33「さぞ菓子っていう字が文法的にっていうか、間違っていたので」（文法的な誤りを指摘）、生徒34「これは、変だから菓子をひらがなに、とりあえず、漢字に変換したんですよ」（修正方法）（変換操作を指摘）、生徒35「これは、さぞかしっていう字が、見た感じで違う字だと思ったので、ひらがなにしたんです」（誤りのある文字を指摘）（修正方法）、生徒36「さぞ菓子は、その、お菓子の菓子じゃないから直した」（意味的な誤りを指摘）、生徒37「お菓子の菓子は絶対におかしいから、ひらがなで」（誤りのある文字を指摘）（修正方法）、生徒38「え、かしが変換しているから、もとに戻そうと思って、ひらがなに」（変換操作を指摘）（修正方法）

さぞかしの、かしを変換した誤りである。この部分に漢字が存在しない異質な文字であることに全ての生徒が気づいた。生徒の内容は、誤りの内容を指摘している。生徒34、生徒38はシステム操作と関連のある指摘を行った。他の生徒は誤りのある文字を指摘、言語的におかしいと感じており（指摘しており）、全ての生徒が「かな」に修正した。

海汚れる(6)

生徒32「助詞がない」（文法的な誤りを指摘）、生徒33「「がが抜けていたので、一応入れました」（文字の脱落を指摘）（原文の文字を推測）（修正方法）、生徒34「海汚れるだと変だから「の」をいれた」（文字の脱落を指摘）（原文の文字を推測）（修正方法）、生徒35「これは、海汚れるで、なんか海と汚れるの間になんか言葉を入れた方がいいと思って、「の」か「が」がいると思った」（文字の脱落を指摘）（原文の文字を推測）、生徒36「海汚れるじゃ、なんか変だから「海が」って一文字加えた。（文字の

脱落を指摘）（原文の文字の推測）（修正方法）、生徒37「海汚れるになっていたから、日本語的におかしいから」（文法的な誤りを指摘）、生徒38「海汚れるで、がが抜けてたから」（文字の脱落を指摘）（原文の文字を推測）

助詞の脱落である。言語の解釈から誤りが発見されている。文を読んでいくと誤りは音韻的、意味的に容易に気づく。欠落は、「が」であるが「の」も正解にした。脱落した文字が的確に推測された。

づくり(9)

生徒31、32、33「すにてんてんを、つにてんてんにした」（かな文字の違いを指摘）（修正方法）、生徒34「ここを、直そうと思って、なんか違うような気がして」（疑問視）、生徒35「これは、道路造りの言葉は同じなんですけど、見た感じが違うと思って」（かな文字の違いを指摘）、生徒36「鈴のずじゃなくて、たちつてとのつづのほうで。だと思っただから直した」（かな文字の違いを指摘）（修正方法）、生徒37「つのでんてんと思った、なんていうか、道路づくりは、づくりだという記憶があったし、先ず、漢字に変換できるから、づくりにして漢字変換にしました」（かな文字の違いを指摘）（文字の推測）（記憶と不一致）（修正方法）、生徒38「ずが違うと思って、だぢづでどのづだとおもって」（かな文字の違いを指摘）

「ず」と「づ」は同音である。生徒34は的確な指摘は行っていない。「づくり」を入力し変換を行ったが変換を中断している。他の生徒はかな文字の違いを指摘し、修正した。生徒37は「づくり」と入力し、変換を行い、漢字に修正した。「記憶があったし」との発言があり、記憶されていた知識と比較照合が行われた状況認知である。

広がって(3)

生徒31「ちっちゃい、つだから」（促音の誤りを指摘）、生徒32「つが大きかったから、小さいつにした」（促音の誤りを指摘）（修正方法）、生徒33「つまるつだったのに大きいままだったので直したんです。」（促音の誤りを指摘）（修正方法）、生徒34「ここを直そうと思って」（誤りのある文字を指摘）、生徒35、37、38「これは、ひろがつてになっているんで」（促音の誤りを指摘）

全ての発言は促音の入力の誤りを指摘している。文を読んでいったときの音韻に違和感を生じる。また文法的な解釈からもここに大きな「つ」が存在する機会はないので、視覚的、意味的に気づくと考えられる。全て促音に修正された。

正答数の少ない誤りの修正についてインタビューを行った結果を以下に簡略に示す。

結果(10)

本文は「原因」。課題文では反対語を記述。文脈上も誤文と気づくのが難しい。修正した生徒はいない。

両全(7)

生徒33「これも、両全だけ直せばよかったんですけど、両全が違っていたので、一目瞭然を直しました」(漢字の誤りを指摘)(修正方法)、生徒34「ここがちがうとおもって、でも何か当てはまる字がないから終わりにしようとおもって終わりにした」(誤りの文字を指摘)(修正する文字が辞書にない)、(修正方法)、生徒35「一目両全の全が自然の然じゃなかったから、した」(誤った指摘)、生徒36「両全の全が間違っていたから」(誤った指摘)、生徒37「両全のりょうが、良いという字かなと思って、変えた」(誤字の推測)、生徒38「いってることがよく分かんないから、もう一回入れてみようと思った」(不確定)(確認操作)

誤った指摘、誤った推測、処理の中断により正答者は2名である。

うに(2)

生徒31「まず、うに見つけたから」(誤まりの文字を指摘)、生徒32「うに」(誤まりの文字を指摘)、生徒34「3行目に間違ってたのがあって、そこを直そうと思って」(誤りの文字を指摘)、生徒36「うにが海だと思ったから直しました」(原文の文字を推測)(修正方法)、生徒37「最後に「に」が入っているから、とりまく虹の色は美しいだと思った。」(原文と異なる文字を推測)、生徒38「鳥をとりにくうにの色ってよく分からない。で、うにを消しちゃって、海にしようとしたんだけど、消しちゃって、そのままにしちゃって」(意味的な誤まりを指摘)、(原文の文字を推測)、(修正の中断)

6名が誤りを指摘したが、原文と異なる文字の推測、修正の中断、かな文字の入力により正答者は3名である。

オヒニトデ(12)

生徒35「これは、読んで言葉が違ったので」(綴りの誤りを指摘)、生徒36「これは、オヒニトデっていうので間違っていたからオニヒトデにカタカナで書き直した」(綴りの誤りを指摘)(修正方法)、生徒38「オニヒトデが、オヒニトデになっている」

(綴りの誤りを指摘)

この誤りを指摘したものは3名である。他の生徒は同一文字として読みとられていると考える。

以上、修正箇所が発言、処理過程、各発言に対する要約を示した。文書修正は、3.1 処理順序の発言にもあるとおり、文章を見ながら、あるいは文章を読みながら、誤りを探していく。文の誤りに気づき、そこで状況を認知し、処理を行う。本研究では修正時の発言ではなく、再現時のインタビューから状況の認知を明らかにしようと試みた。これは中学生に発話思考法を適応することが困難なためである。厳密な考えによれば、この方法では状況の説明が修正後になり認知に変移を生じる場合がある。しかし、処理過程は修正時と同じに再現されており、生徒は再現の途中で発言を行っている。そして、発言の内容と処理過程には整合性がある。つまり短期記憶によって状況が復元され、発言を行っているので修正時と等価的な状況認知としてとらえていく。再現時の発言は教師に対する報告として行われるので、被験者は状況の認知ということばで発言を行うことはない。発言は、状況が認知され、それをもとにして修正時における状況の説明、誤りの指摘や修正の理由、疑問、推測した文字、言語、修正方法として述べられた。文書修正に関してはどのような誤りであり、どのような文字を入力したらよいかを思考する。すなわち、誤りの指摘と文字の推測の2要因をもとにして処理が行われる。この両者を統合し状況の認知とする。

誤りの箇所(1)から(13)の状況の認知と文字の推測を表1及び表2に示す

表1はすべての誤りの箇所の発話データの中から取り出した指摘と文字の推測を示した。指摘に関しては18カテゴリ、推測に関しては5カテゴリである。

表2は誤り箇所ごとの状況の認知である。(1)~(13)の誤り((10)は除く)は、ワードプロセッサの操作上のミスに起因している。しかし、文書修正は文章を読解していくので、誤りは文法的、意味的、言語的に指摘・認知された。

状況の認知は段階的である。すなわち誤りに対する指摘は誤りの状況を詳細に述べたものから、おかしいという疑問視した発言、誤った指摘等がある。また、文字の推測に関しては、原文に使用された正しい音や文字の推測から、誤字の生成等がある。

一般論として正答数の多い箇所は状況の認知が適切である。正答数の少ない箇所は状況の認知が

表1 状況の認知

| | |
|-------------|--------------|
| 指摘 | |
| C0: 誤りの文字 | C1: 文法的な誤り |
| C2: 意味的な誤り | C3: 漢字の誤り |
| C4: 促音の誤り | C5: かな文字の違い |
| C6: かなに漢字 | C7: 類似の文字 |
| C8: 文字の脱落 | C9: 不要な音の入力 |
| C10: 綴りの誤り | C11: 漢字がかな文字 |
| C12: 文字の不一致 | C13: 不確定 |
| C14: 疑問視 | C15: 誤った指摘 |
| C16: 記憶と不一致 | C17: 変換操作を指摘 |
| 文字の推測 | |
| G1: 原文の文字 | G2: 原文と異なる文字 |
| G3: 文字の追加 | G4: 誤字 |
| G5: 不確定 | G6: 確認操作 |

表2 誤り箇所ごとの状況認知

| 箇所 | 指摘 | 推測 |
|------|----------------------|--------|
| (1) | C2, C16 | G1 |
| (2) | C0, C2 | G1, G2 |
| (3) | C4 | |
| (4) | C2, C3 | G1, G5 |
| (5) | C2, C3, C7, C14, C15 | G2, G5 |
| (6) | C1, C8 | G1 |
| (7) | C0, C3, C13, C15 | G4, G6 |
| (8) | C1, C2, C6, C17 | |
| (9) | C5, C16 | |
| (10) | -- | -- |
| (11) | C2, C9, C14 | G3 |
| (12) | C10 | |
| (13) | C11, C12 | G1 |

曖昧で疑問視するだけ、誤った指摘になり、文字の推測が誤字を生成したり不確定になる。一例として、正答数の少ない「両全」について、状況の認知と処理過程を図2に示す。

漢字の誤りを指摘し四字熟語を入力したのは生徒33の1名である。誤りが不確定で確認操作で修正したのは生徒38である。誤った指摘で誤字を生成したのは生徒35, 生徒36の2名である。生徒34の発言は「両全」が誤りであり、「りょうぜん」と入力したが辞書に「瞭然」がなく、処理を中断した。状況の認知は被験者ごとに異なり認知の仕方や処理過程が異なった。生徒33と38は処理過程は同じだが状況の認知には差異がある。

ワードプロセッサの操作に関する発言は、さぞ菓子の生徒34の発言「これは、変だから菓子をひらがなに、とりあえず、漢字に変換したんですよ」

ね」、生徒38の発言「え、かしが変換しているから、もとに戻そうと思って、ひらがなに」及び両全における生徒38の発言「いってることがよく分かんないから、もう一回入れてみようと思った」等である。状況の認知に関して変換したというシステム操作を認知している。また、文字を確認するためにもう一度入力して確認を試みている。生徒34, 38は言語とシステム操作の複眼的な視点で修正を行っていたといえる。

4. おわりに

今回の実験においては、修正後の再現時にインタビューを行い、初発の発言によって分析を行った。したがって、生徒の発言にレベル差があり、発言の内容が認知を十分表現しているとはいえないが、文書修正時の学習者の状況認知及び処理過程

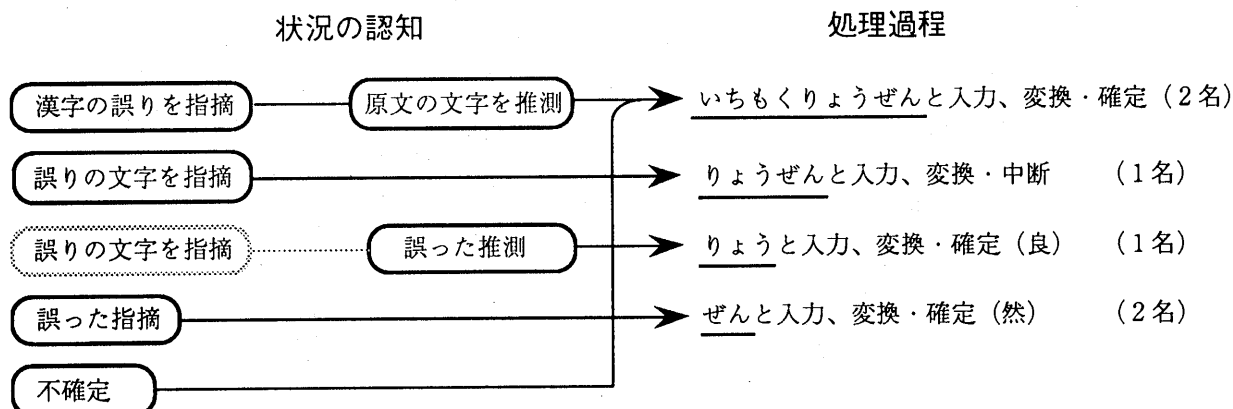


図2. 状況の認知と処理過程

の一部を明らかにすることができた。

文書修正において状況の認知は表1及び表2のようになった。他者の文章を修正するには誤りを抽出し、文脈に適合する語句を推測する必要がある。回答は選択肢で問われるような容易な環境でなく、言語に関する知識が問われ推論を行う問題解決の場となる。これが修正過程にあらわれる。今回の実験で正答数の多い修正箇所は誤りの内容が認知されており、正しい文字が推測され修正されている。しかし、正答数の少ない修正箇所は状況の指摘に誤りがあつたり、指摘が曖昧で誤った文字を入力したり、誤りに着目したが、誤った推測で修正したり、推測ができず想像して作成した事例もある。

誤りはシステム操作に依存するものを多く設定した。ワードプロセッサを使用した場合には操作上の誤りから独特の文字列が生成される。このような状況の中で修正を行うには言語に対する分析とワープロ操作における分析の複眼的な視点で分析・解釈・操作していく力が必要になる。情報システムにおける文書修正は、紙に書かれた文書の修正と異なり、実験・検証ができる環境である。状況を認知し、何故そのようになったかをシステム操作によって検証・解釈すれば正しい文字を確定できることがある。「うに」において「うに」と入力し「に」を消去、「み」と入力した事例があるが、これはキー入力時に隣接キーをたたいたことを検証しているとも考えられる。両全において「いつてることがよく分かんないから、もう一回入れてみようと思った」の発言はシステムへ接近した取り組みといえる。このような態度を助長していくような指導をすすめ、学習者がシステムに積極的に接近していく力を育成していかなければならない。

システムへの接近について述べたが、処理過程はシステムに依存する。両全において「これも、両全だけ直せばよかったんですけど、両全が違っていたので、一目瞭然を直しました」と発言した生徒は正しい四字熟語を作成した。しかし、両全が誤りであると認知し、「りょうぜん」と入力したが、瞭然に変換されず処理を中断した事例がある。また、「排水溝」に疑問を持った生徒と教師の会話を次に示す。「『こう』の字がわかんないんだよね、だから、コンピュータにさがせて、だから『はいすいこう』ってってやれば」先生「ああ、でも3文字でちゃんと変換しているとは限らないんだよ、『こう』は

たまたま最初の字がでるかも」「そういうことは考えなかった」この事例のようにシステムの機能に依存した場合、異なる文字を作成することがある。システムの動作を評価し適切な方法を考えさせていくことも大切な課題である。

情報教育においては、情報を取り扱う機会が日常的になる。特にネットワークによって社会と接続された場合、児童や生徒の発達段階に整合しない内容や十分に吟味されない情報が流れてくる。このような情報を扱う機会は今後ますます増加してこよう。今回の修正は正答率59.6%である。修正は他者の作成した文章が静的データで与えられる。これは個人がプランニングして文章を作成していくことと異なり、状況の認知は低下すると考えられる。文章の内容は文脈からおおよそ把握できると考えられるが、動機付けなどによって文章読解に関する積極的な取り組み方を育成していかなければならない。

今回は文書の修正に視点をおいた。状況の認知と処理過程は、課題文の状況によって異なるので、継続的な研究が必要である。また、状況の認知と処理過程は、読みにおける心理的要因、言語に関する力量及びシステムの操作に関する技能等の要因で左右される。量的研究と質的研究を検討していく必要がある。今後、さらにコンピュータを活用した授業実践が増加する考えられる。情報システムや学習場面の様々な状況に対する学習者の認知と処理過程を明らかにし教育に反映していく必要がある。

なお、本研究は科学研究費補助金（基盤研究C）課題番号09680209の研究成果の一部である。

参考文献

- 1) 中野靖夫, 和田道明, 小岩寿之(1997): 中学生の文書修正過程, 日本教育工学会研究報告集, JET97-2, pp.105-110
- 2) 前田恵三, 中野靖夫(1993): コンピュータ操作過程の再現システム, 日本教育工学雑誌, VOL 16, NO4, pp.185-195
- 3) 御領謙(1994): 読むということ. 東京大学出版会
- 4) 海保博之, 原田悦子, 黒須正明(1995): 認知的インタフェース, 新曜社

研究発表 4

電子情報通信学会技術研究報告

ET 98-72~92

〔教育工学〕

1998年11月20日

プログラミングにおける状況認知と処理過程

| | | |
|--|---|--|
| 中野 靖夫 上越教育大学 〒 943-0834 新潟県上越市西城町 1-7-2 Phone: 0255-25-9147 E-mail: nakano@juen.ac.jp | 前田 恵三 東京工業高等専門学校 〒 193-8610 東京都八王子市櫛田町 1220-2 Phone: 0426-68-5064 | 和泉 嘉則 山形大学附属中学校 〒 990 山形県山形市松波 2-7-3 Phone: 0236-41-4440 |
|--|---|--|

あらまし これまで、キーボードの操作履歴の分析により、プログラム作成過程を解明してきたが処理過程における学習者の状況認知や意思決定が不明であった。そこで、非反応測定法によりBASICプログラム作成中のキー入力データを収集するとともに、発話思考法によって発話データを収集し、両者の情報を統合してプログラミングにおける状況認知と処理過程を明らかにすることを試みた。本研究は中学生2名の状況認知と処理過程について報告する。

キーワード プログラミング、BASIC、状況認知 学習履歴

Recognition of Situations and Information Processing Procedure in Programming

| | | |
|--|---|--|
| Yasuo NAKANO Joetsu University of Education 1-7-2 Nisisiro, Joetsu, Niigata 943-0834, Japan Phone: 081-255-25-9147 E-mail: nakano@juen.ac.jp | Keizo MAEDA Tokyo National College of Technology 1220-2 Kunugida, Hachioji, Tokyo 193-8610, Japan Phone: 081-426-68-5064 | Yoshinori IZUMI Yamagata University Junior High School Attached to Faculty of Education 2-7-3 Matsunami, Yamagata, Yamagata 990, Japan Phone: 081-236-41-4440 |
|--|---|--|

Abstract

Because various situations occur in the process of programming, this study tried to collect the data by a protocol method while subjects make programs, as well as the input data of the subjects given through the keyboard of a personal computer.

The purpose of the study is to clarify how students recognize various situations and process information in programming, based on the data collected as stated above. The subjects were two second-grade junior high school students, called **A** and **O**. The results revealed that the student **A** first wrote statements, then submitted the program repeatedly, which enabled **A** to logically recognize the situations and correct the program for completion. On the other hand, the results showed that the student **O** submitted and corrected the program without recognizing the situations. However, though the student **O** inappropriately recognized the situations, he wrote the program and finally completed it.

key words programming, BASIC, recognition of situations, learning process

1. はじめに

コンピュータを動作させるにはプログラムが不可欠である。プログラミングは論理的な思考力及び創造力の育成、知的生産等情報科学を支える能力開発に大きく寄与する教育内容である。これまで、学校教育においてプログラミングは情報処理教育の専門科目として履修されてきた。また、情報教育の教材として高等学校においても取り上げられ、さらに、中学校技術・家庭科では新領域として情報基礎が加えられ、内容の一つとしてコンピュータの基本操作と簡単なプログラムの作成がもりこまれた。プログラミングの様相を観察してみると、学習者がさまざまな方法によって問題解決に取り組んでいる。

Gerald M. Weinberg[1]はプログラミングの分析についてつぎのように述べている。「実験心理学者なら、まず、すべての被験者が同じ作業をしたのだろうか、と問うであろう。注意深く行われた対照実験においてさえ、とかく被験者がやっていること、またはやっていると思っていることの中に、多くの差異が忍び込んでくるものだ。その上もし2つの異なる実験を比較したいと思えば、まず、作業内容に何らかの相違がなかったどうかを調べてみないことには有意義な結論を引き出すことはできない。プログラミングにおいては、どの2つの作業をとってみてもそこに無数の差異があるものだからわれわれとしてはまず、作業同士の違いという問題について考えてみる必要がある。」

プログラム作成過程に関する研究は岡本ら[2]が診断助言型のITSを用いて分析した事例がある。この研究ではメンタルモデル解明のため、机上でコーディングを行わせながら作業内容について口頭で説明させ、これをテープレコーダーで録音しオンライン・プロトコルを採取している。しかし、言語プロセッサと対話したときのありのままの作成過程の分析は行われていない。

前田ら[3]は高専生のC言語のプログラミング過程を個別に解説し、構築方法及び記述に関するカテゴリ分析を行っている。この研究は操作過程を明らかにしたのが作成者の思考については検討されていない。

プログラム作成中にはさまざまな状況が発生する。作成者はその状況を認知し解釈し記述していく。そこで、本研究はプログラム作成中に学習者にプログラム作成に関する考え方を発話させるとともに、プログラム作成中のキー入力データを収集し、作成者の状況認知と処理過程を明らかにすることを試みた。

2. 方法

課題を与え、発話思考法によりプログラムを作成させ、前田ら[4]の開発したコンピュータ操作過程の記録・再現システムによってキー入力データを収集す

る。発話データと処理過程を統合し状況認知と処理過程を明らかにする。

2.1 被験者 山形県内の中学2年生 部活動でプログラミングを自学自習で学習した者に事前指導

2.2 言語プロセッサ BASIC

2.3 実施時期 1997年7月

2.4 事前指導

実験前に接続、判断、反復に関する指導を行った。

2.5 課題

青、黄、赤の半径が50の3つの円を作成します。青い円の中心は(65,320)、黄色い円の中心は(130,210)、赤い円の中心は(130, 420)です。

次に5人のテストの点数を入力し、その点数によって、青、黄、赤色の半径を10ずつ大きくしていくプログラムを作りなさい。ただし、80点以上のときは、青の円の半径、60点から79点のときは黄色の円の半径、59点以下のときは赤の円の半径を大きくします。

2.6 データ収集

被験者に発話思考法でプログラムを作成させ、スキャンコンバータを使用してディスプレイの画像と発話をVTRで録画。作成過程は前田らの開発した記録・再現システムでキー操作履歴を記録する。

2.7 分析法

発話データとキー入力データを統合し、状況の発生とその認知及び処理過程を明らかにする。

3. 結果と考察

被験者2名のキー入力データと状況認知に関する発言を以下に示し分析を行う。キー入力データは、命令として記述した。「」内は生徒の発話である。T「」は教師の発言である。命令ごとの記述の誤りは解釈、評価が行われて直ちに修正されている。この状況はデータから除いた。状況として取り上げた事象には、分析の単位として状況Nとラベルを付した。状況はプログラムの記述、修正過程、実行等における、あるまとまりとした。*は状況認知と処理過程に関する分析者の解釈である。なお、各状況に分析はD.A.Norman[5]の7段階のモデルを援用していく。

3.1 生徒Aの状況認知と処理過程

プログラム作成前に教師が作成方法について聞いている。プログラム作成のデザインはできている。

10 CLS 3

20 CIRCLE(65,320),50,1

30 CIRCLE(130,210),50,6

状況1 エディタの誤操作

「うーん、めんどくさいんで AUTO します」

AUTO

リターンキー 3回

40 CIRCLE(130,420),50,2

* AUTO で行番号を自動発生。行番号 10～30 にリターンキーをたたく。ここで文が消去される。しかし、記述した命令はディスプレイ画面に表示されているので、状況の変化を、知覚できない。状況認知に関する発話はなく 40 行目から命令の入力。10 行目から 30 行目は記述されたと解釈しているのではないか。

状況 2 FOR 文の指定回数の誤り

50 FOR A=0 TO 5

* 課題のデータ入力数は 5。FOR 文の指定回数は 6 回、発話はなく、処理は行われない。

状況 3 INPUT 文の誤り

60 INPUT A"ニンメノテンスウハ";B

「1 人の場合は、いちにんが、まあいいや」

* 入力した INPUT 文に誤りがある。この文に誤りがあるという状況を認知する発話はない。実行時の機能を確認している。FOR 文の指定回数は、0 から 5 であるが 1 人目についての発言。「まあいいや」は生徒の命令文に対する評価。次の行の入力が行われる。

状況 4 判断・分岐の記述

70 IF B>79 THEN

「IF B 大なり 79 THEN あれ、あれどうなるのかな。THEN の後はあとで入れるということで」

* 80 点以上の条件で分岐する。判定の条件式は記述したが、条件を満足したときの詳細化した記述は行われない。

80 IF B<80 修正

「IF B 小なり 80、あれ、79 っていうのは。うーん、どうしようかな。はー。あっ、79 より多いのはすでに 70 (行)の時点で他のところにいるはずなので」

* 60 点以上の判定に対し、B を 80 以下と記述した。判定条件の 60 から 79 の範囲を決定する際、79 点について思考する。79 以上の数値は 70 行目で判定したので、次式に変更した。既に記述された文 70 行目を参照しこの 80 行目の判断の要素にしている。

80 IF B>59 THEN

「ここん所は、まあ、えー大なりの 60。うーん、え、59 ということで、うん THEN」

* 条件判断は 60 以上であるが、発話は「大なりの 60」と発話したが「うーん」と発話、70 行目と同様に、59 をしきい値とし演算子に > を使用した。

90 IF 消去

「IF、あれ、何これは、黄色の、赤の円のとき、59 点。こんときには、既に B は、まあ、59 点以下なので、これはいらぬということで」

* 59 点以下の条件を判定するために IF と記述したが 59 点以下の判定は不用ということに気づき消去。

* 70,80,90 行は点数により処理を判断・分岐する。IF 文を構成すること、IF 文の判別式の構成について思考している。しかし、条件を満足したときの詳細化は行われていない。

既に記述した文を参照し、処理の流れによって文を修正している。記述内容、動作状況を確認しながら、プログラムを構成している。

90 GOTO 120

* 飛び先行番号を 120 に設定。70 行、80 行の処理が 100,110 行目に入力することを計画している。

70 IF B>79 THEN 100

80 IF B>59 THEN 110

* 70、80 行に、飛び先番地を入力。

100 CIRCLE(130,420)

* 100 行目に円の中心を設定。円の描き方に関する質問をする。行為を詳細化するために必要な手続き。質問により 70,80,90 行目の処理を詳細化し、行為を実行(記述)する。70,80,90 行目にカウンタ及び GO TO を追加。

70 IF B>79 THEN C=C+1:GOTO 100

80 IF B>59 THEN D=D+1:GOTO 110

90 E=E+1:GO TO 120

「ここんところは、BCDE だから E=E+1,GO TO 120」

* 変数名を追跡、次の変数名を決定。

100 CIRCLE(65,320),C*10+50,1

110 CIRCLE(130,210),D*10+50,6

120 CIRCLE(130,420),E*10+50,2

状況 4 の判断分岐に関しては、まず、3 つの条件式により処理の分岐を行っている。条件式の記述の際、条件の関連、すなわち状況を検討している。条件式が成立する場合の処理は途中で課題の要求事項を教師に確認し、意図を形成し行為を実行している。詳細化も飛び先番地を仮に記述しておくなど合理的な処理過程である。

130 NEXT A

140 END

「では、最後に END で、あれ、END。はい、これから直してみます」

* プログラム全体の確認に関する発話はない。また、LIST の表示は行われていない。実行によって修正する方法がとられた。

RUN STOP LIST

状況 5 1 回目の実行 (Break in 60)

「ありゃ、円が描かれていないので、まあ、調べてみます。あら、今の 10 番から 30 番まで何かなかったので、もう、1 度入れます」

* 実行した結果、プログラムの停止。Break in 60 の表示。原因を調べるために LIST を表示。行番号 10-30

が入力されていないことを、認知（知覚）評価。状況1が原因だが、具体的に命令が欠落している状況になる。エラーメッセージとは異なる状況を認知。再入力を行う。60行目の誤りの状況は認知されず修正は行われない。

```
10 CLS 3
20 CIRCLE(65,320),50,1
30 CIRCLE(130,210),50,6
「じゃあ、もう1度やっています」
```

RUN

状況6 2回目の実行

```
Break in 60、黄色い円の一部を表示
「あれ、これってもしかして」
T「円が下にいつてる」
「これって、SCREEN やりました？」
*画面表示から通常の処理結果でないことを認知。
SCREEN 命令の使用を推測。
T「場所が違うの」「はい」
T「直してください、じゃあ、円の場所」
「じゃあ、場所直します、あれ、65の320だから、でも130、青い円、まん中にくるはずなのに、うーん」
*教師の指導によるプログラム修正。具体的な指示がないので、座標と円の位置関係を検討。
T「座標が逆かな、320の65」
「あ、では、座標が逆だったようなので、元に戻します。320の65,210の130、次は420の130、(40行目430 130と入力、正しい入力は430,130)下の方も直します」
「もう1度やってみます」
```

RUN

状況7 3回目の実行 (Overflow 40)

```
「何か、40番が間違っていたみたいです、あ、何か変になっていました。あれ、直しました」
*青と黄色の円が描かれる。エラーメッセージにより誤りを認知、再入力。
(40行目の入力時には入力ミスを検知していない)
```

RUN

状況8 4回目の実行 Syntax error in 60

```
「RUN, になりました。ん、では、まず、最初の人の点数は何点にしましょうか。じゃあ、うーん、2だから20点で、まず、20点」
*「RUN, になりました」はプログラムが実行された。3色の円が描かれ、この時点ではエラーがないことを認知（評価）。
```

Syntax error in 60

```
「60番が間違っていたみたいなので、又、もう1度みます。あれ変だな、うーん、あってる、あってる」
「はー、ではここ、ただの点数はにします。あれ、ここは1つになるのかな」
```

```
*20点を入力したが Syntax error in 60となる。60行目に誤りがあるとエラーメッセージの表示。あってる
```

と解釈・評価。しかし、以下のように“テンスウハ”に修正。

```
60 INPUT "テンスウハ";B
「では、もう1度やってみます」
```

RUN

状況9 5回目の実行 データ入力が6回

```
「20点、次、60点、80点で100点。何か6人分まで続いちゃったんで、もう1度調べて見ます」
```

```
*データを5回入力。さらにデータが要求された。入力が5個以上になることを認知（解釈）。再検討を行う。
```

```
RUN「1点、2点、3点、4点、5点、6点、何か6点になっちゃてるんで、Aを0から始めて5までだから6つだったと思うので、ここを4までに直します」
```

```
*回数を確認しやすいように1から6までの数値を入力。プログラムの誤りを検出するための合理的な手段。評価を行うための詳細化。その結果、入力回数は6回になることを認知（知覚）。繰り返しを0～5までを4までに変更。
```

```
50 FOR A=0 TO 4
```

```
*50行目、繰り返しを4までに修正。
```

```
「もう1度やってみます」
```

RUN

```
「20点、40点、60点、80点の100点、はあ、今度はいいみたいなので。SAVEで」
```

```
*正しく実行されたと認知（評価）
```

生徒Aの状況認知と処理過程を図1に示す。状況の発生とそれに関連する認知あるいは処理を番号で示す。プログラムの作成は、プログラムを記述する段階と実行・修正する段階に分かれている。プログラムを記述段階において記述した各命令の解釈、評価に関する発話は少なく処理は行われていない。つまり、モニタリングやメタ認知は活性化していない。プログラムの構築法は事前に検討してあり、いかに構成するかに力点が置かれているといえよう。様々な状況の発生はその時点で解決しなくてもプログラムの実行により具体化した事象を対象にして気づき、解釈、評価し処理していく方法がとられている。コード化した命令の解釈、評価ではなくシステムの機能を活用していく方略、思考である。従って、いくつかの状況が積層されていくが、実行結果によりその状況を認知し、気づき、解釈、評価しプログラムの記述をコントロールしている。つまりメタ認知が機能しているといえよう。D.A.Normanの7段階の行為の理論を適用すれば目標設定から行為の実行の段階と知覚、解釈、評価の評価段階が明確に2分化して表出したといえる。

3.2 生徒Oの状況認知と処理過程

プログラム作成以前に教師が作成方法について聞いている。課題は解釈されデザインはできている。

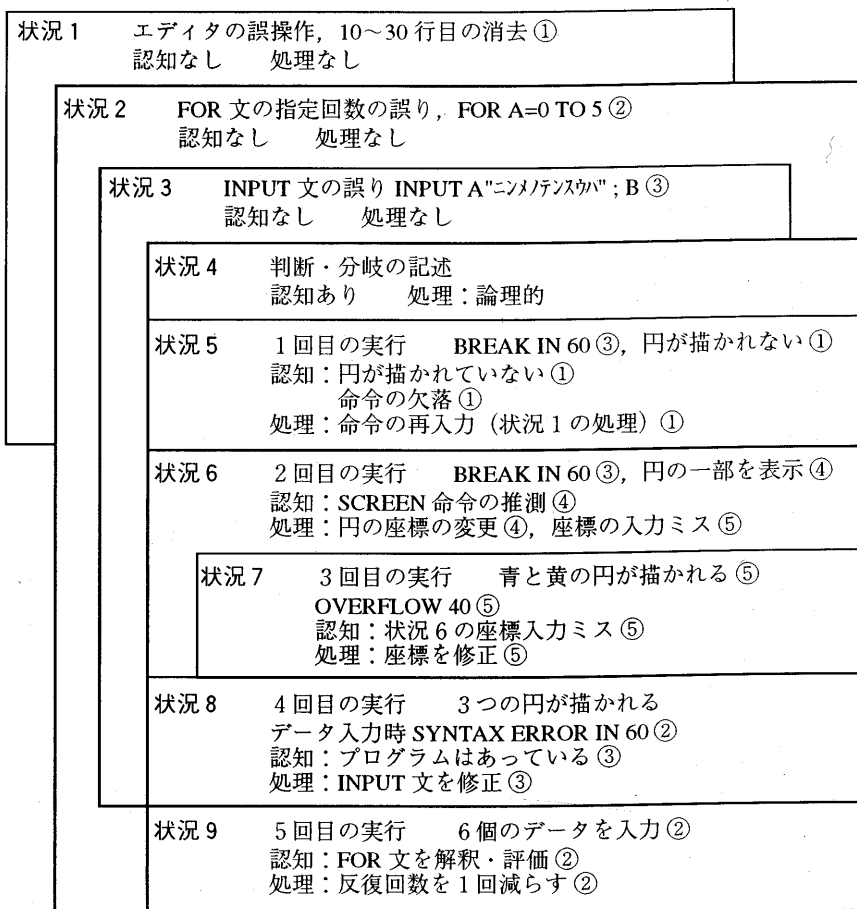


図 1. 生徒 A の状況認知と処理過程

状況 1 SCREEN 3 の記述

10 SCREEN 3:CLS 3

* 画面構成は 640 × 400 になる。課題の座標の入力はこの設定にあわせる必要がある。この状況は最後まで認知されていない。

状況 2 同一の色の指定

20 X=50:Y=50:Z=50

30 FOR A=1 TO 5

40 CIRCLE(320,65),X,1

50 CIRCLE(210,130),Y,1

60 CIRCLE(420,130),Z,1

「アララのラって、何かずうっと 1 をおしてしまったので、ということで 50 を直しましょう。LIST おしちゃいまして、この 1 を黄色で、黄色と赤ですから、黄色は 6 ですね、6。赤は 2 でしたっけね。2 でやって」

* 60 行目の入力を終了し、色の指定が全て 1 にしていたと認知 (知覚、評価)。50 行目 60 行目を修正を行う。

50 CIRCLE(210,130),Y,6

60 CIRCLE(420,130),Z,2

AUTO 70

状況 3 FOR 文の指定回数の変数名と INPUT 文の変数が同一

70 INPUT;A

* 発話はなく修正は行われぬ。以下、判断・分岐の部分において変数名は使用されたが発話、修正は行われぬ。

80 IF A>80 BS BS 79 THEN X=X+10

「A 大なり 80、違う。そうしたら 80 は除けちゃうので 79 にしちゃいます」

* A>80 は課題の条件を満たさないと認知 (解釈) 条件を変更する。コーディングの記述ミスの修正ではなく、条件を満足するかに視点をおいた修正。内容に関する変更。

90 IF A>59 THEN Y=Y+10

100 IF A BS BS BS BS Z=Z+10

「あ、これって言うのは、このあとこれしかないはずですね。だから、めんどくさいから IF A っていうのは消しちゃって、Z は Z たす 10 にします」。

* 3 つの条件判定の最後の式、この行以下の判定はないと認知。IF 文を変更する。

80 IF A>79 THEN X=X+10:GOTO 110

90 IF A>59 THEN Y=Y+10:GOTO 110

110 NEXT A

「これでできなかつたら、死のう。っていうことで、一応自分としてはへたくそですけど、できたような気がしますので、ちょっとやってみます」

*全体を確認したような発言、処理は行われない。

RUN

状況4 1回目の実行 3つの円は上部に表示 データ90を入力、OKの表示。

「確かに大きさも50なので、何か割合としては変なんですけど、ここのとおりには描きましたので、悪いとは思っていませんけど」

*課題のデータによる図形表示。3つの円は上部に表示。割合としては変だけど処理は正しいと評価している。状況は認知されず修正は行われない。

T「ん、確認してみてください」

「もしも僕が、例えばA子さんが90点とりました。OKなっちゃいました。えーとなぜか知りませんが、ちょっと見てみましょう。サークル、XYZですね。つぎINPUT A あそっか」

*データ90を入力。OKの表示。エラーがあることを認知。このプログラムは変数名と繰り返し変数名が同じである。入力が行われた時点で誤動作する。理由は分からないのでLISTの表示を行う。

文番号 110を120に変更。

「じゃあ、ここを。もう、しょうがないからIF文作るのが間違っていたので訂正します。えいっ、しょうがないですね。と小なり59,60ですね。60 THEN。これだったら、わざわざGO TO 110だっていらないのでGO TO 110消します。何で違うんだろう」

*IF文に誤りがあると状況と不一致な解釈、3条件で判別するよう以下のように修正。しかし誤りの原因は確定していない。判断分岐に誤りがある。

```
100 IF A<60 THEN Z=Z+10
```

```
80 IF A>79 THEN X=X+10
```

```
90 IF A>59 THEN Y=Y+10
```

「新しいのをを出してみます。確認しましょう。SCREEN 3 CLS 3,50である。これに大きさ設定しただけですね。もし、79以上だったら、Xに10足されて60になる。110」

120行目を消去

RUN

状況5 2回目の実行 データ90を入力、OKの表示「さあ、どうしましょうね、RUNしちまうか。また、90でOKになっちゃいましたよ。よし、決めます。FOR NEXT やめましょう」

*ためらいがあるが実行する。90を入力するとOKの表示。FOR NEXT やめようという意味決定。何故かは不明、原因の分析を行わず、プログラムの構造を変更した。

```
30 FOR A=1 TO 5 を消去
```

```
110 NEXT A を消去
```

```
70 B=B+1 カウンタを追加
```

```
110 IF B=5 THEN END
```

```
120 GOTO 40
```

ここまでの修正で以下のような後判定反復に修正。

```
10 SCREEN 3:CLS 3
```

```
20 X=50:Y=50:Z=50
```

```
40 CIRCLE(320,65),X,1
```

```
50 CIRCLE(210,130),Y,6
```

```
60 CIRCLE(420,130),Z,2
```

```
70 INPUT;A:B=B+1
```

```
80 IF A>79 THEN X=X+10
```

```
90 IF A>59 THEN Y=Y+10
```

```
100 IF A<60 THEN Z=Z+10
```

```
110 IF B=5 THEN END
```

```
120 GOTO 40
```

「やっぱし難しいんだよな、THEN,そっか、6だったらENDにしちゃえば、サークルを描いていることになりすね」

*データ5個で終了する。分岐先の記述はない。しかしカウンタBが6になれば終了と解釈、次のように変更。後判定なので1回余分の入力になる。

```
110 IF B=6 THEN END
```

RUN

状況6 3回目の実行 青、黄色の円が2つになる。

「エー、何で90入れたら、あっそうか、これだったらだめなんでGOTO 110。GOTO 110を付け加えます」

```
80 IF A>79 THEN X=X+10:GOTO 110
```

```
90 IF A>59 THEN Y=Y+10:GOTO 110
```

*80行目の処理と90行目の処理が行われている。この状況を認知、だめなんでと発話し判定後にGO TO文を付け加える。文構成に誤りがあると解釈、評価。修正を行う。

RUN

状況7 4回目の実行 データ入力6回

「ちょっと確かにこれがおかしいですね。やっぱりここにしましょう」

*データ入力が6回、回数に誤りがあるので、110行目IF B=6 THEN ENDを消去。

```
65 IF B=5 THEN END ROLLUP(CRの入力がない)
```

「これを65でやってみましょう。IF Bは5 THEN END。これでやってみましょう」

*状況認知は適切で、繰り返しの回数を1回減少させる。しかし、入力操作に誤りがあったので修正は行われない。

RENUM LIST

RUN

状況8 5回目の実行 データ入力6回

「また、やっちゃいました。なぜでしょう。分かりました。110だからですね。え、なんで」

*入力回数は6回、どのように解釈したかは不明。

LIST

```
100を65 IF B=6 THEN ENDに修正
```

*処理は前判定反復に変更したが再度6と指定。状況と一致しない処理過程である。

```
100 リターン
```

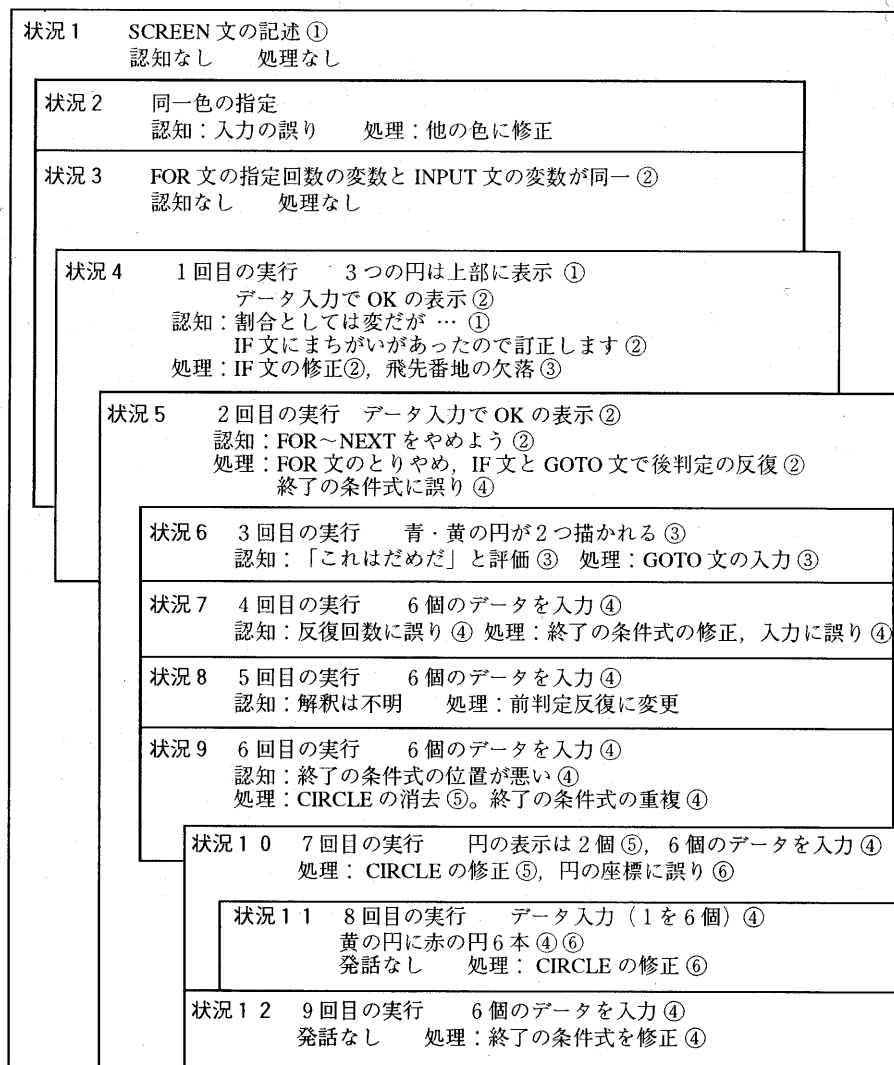


図2 生徒Oの状況認知と処理過程

RENUM

LIST

80,90行目 GOTO 110 にする。

*前判定反復に修正。この時点でプログラムは以下
のようになる。

10 SCREEN 3:CLS 3

20 X=50:Y=50:Z=50

30 CIRCLE(320,65),X,1

40 CIRCLE(210,130),Y,6

50 CIRCLE(420,130),Z,2

60 INPUT;A:B=B+1

70 IF B=6 THEN END

80 IF A>79 THEN X=X+10:GOTO 110

90 IF A>59 THEN Y=Y+10:GOTO 110

100 IF A<60 THEN Z=Z+10

110 GOTO 30

*70行目にB=6のときEND。入力は6回でENDに
なる。繰り返し回数の解釈、評価が行われていない。

RUN

状況9 6回目の実行 データ入力6回

「さあ、90またですね」

LIST

「ここ55にして、55にして。これでいいとおもいま
す」

70 を50に変更(50 CIRCLE(420,130),Z,2は消去され
る)

50 を55に変更

70 リターン

*状況に不一致の解釈、修正。命令の消去が行われ
る。

RUN

状況10 7回目の実行 入力6回 円の表示は2個

*LISTで確認して、実行するのではなく直ちに実行
している。入力6回、状況9においてプログラムの変
更に誤りがあり以下のようなプログラムになった。

LIST

10 SCREEN 3:CLS 3

20 X=50:Y=50:Z=50

30 CIRCLE(320,65),X,1

```

40 CIRCLE(210,130),Y,6
50 IF B=6 THEN END
55 IF B=6 THEN END
60 INPUT ;A:B=B+1
80 IF A>79 THEN X=X+10:GOTO 110
90 IF A>59 THEN Y=Y+10:GOTO 110
100 IF A<60 THEN Z=Z+10
110 GOTO 30

```

40を50に、YをZに、6を2に修正、変更。

```
50 CIRCLE(210,130),Z,2
```

*状況は認知されCIRCLE文の追加。円の座標に誤り。

RUN

状況 11 8回目の実行 データ入力6回

*1を6回入力、黄色の円のまわりに赤線6本、発話はない。しかし、状況は認知され50行を以下のように修正。

```
50 CIRCLE(420,130),Z,2
```

RUN

状況 12 9回目の実行 データ入力6回

LIST 55行目を以下のように修正。

```
55 IF B=5 THEN END
```

*発話はないがデータ入力をカウントを5に変更。

RUN

データ5個入力

「いけ、これでどうだ。やりましたねー、今回の感想としては、エーとめちゃくちゃ、思ったほど作動しなかったことですね。やはり、ここはもう少しよくプログラムを見てからやった方がいいですね」

*プログラムは完成。データの入力回数は5回で、課題の要求どおりに動作。完成するまでの手続きを反省、自己評価している。しかしSCREEN文については認知されず修正は行われていない。

生徒Oの状況認知と処理過程を図2に示す。課題の内容を読みとりプログラムの作成は速やかに(741.9秒)行われた。状況2及び判断・分岐の記述に関しては命令の記述は論理的に行われた。モニタリングも適切でコントロールが適切に行われた。しかし、状況3においてFOR文の繰り返し変数と、点数の変数を同一にしまい、繰り返し回数は1回しか実行されなかった。状況4においてプログラム修正はIF文の誤りという認知(解釈)。状況と整合性のない解釈と処理過程となる。状況5、状況8、状況9においても状況と不整合の命令が記述された。プログラミングの記述に関する知識はあり、プログラムの構成法をFOR文からIFとGO TO文によって前判定反復、後判定反復に変更した。目標設定と発生した状況の解釈は不明確で次々に目標を設定し行為を実行している。操作履歴から見ると試行錯誤型になるがD.A.Normanのモデルの知覚、

解釈、評価がプログラム全体に行われていないことになる。最後の自己評価で自分もそのことに気づいた。

4. おわりに

本報告では2名の生徒のプログラム作成過程を明らかにした。Gerald M.Weinbergが述べるように2者のプログラミングは異なっている。状況の発生は生徒ごとに異なり、この認知や解釈により処理過程は左右される。生徒Aは状況が発生しても解釈、評価は行っていない。しかし、実行したときの状況の認知が論理的に行われ適合する処理を行い完成させている。生徒Oは状況と不整合の解釈、評価によりプログラムを変更した。変更はFOR文、IF文とGO TO文を使用した前判定反復、後判定反復と構造的な変更を行い、状況にを解決する処理に至るまでの経緯は長い。2者の差はD.A.Normanの7段階モデルの評価段階すなわち知覚、解釈、評価の差ととらえてよからう。

プログラミングの指導の問題点は、教師の経験が少なく指導方法が確立されていないことと、生徒たちが実際にどのように取り組んでくかが未解明な点である。しかし、プログラム作成は生徒各自の思考力、判断力、表現力、新たな発想や論理の方略が要求される教科内容である。作成段階では知識の統合が要求される。それゆえ、個々の生徒の学習を教師がモニタしていくことが新たな課題となる。

本報告は教育実践の場でデータを収集し事実を明らかにした。さらに研究を深化させるには教育学、認知科学などと学際的に取り組んでいく必要がある。

なお、本研究は科学研究費補助金 基盤研究(C)課題番号(09680209)の研究成果の一部である。

参考文献

- [1] Gerald M.Weinberg(1994)/木村泉, 角田博保, 久野靖, 白濱律雄訳: プログラミングの心理学, 技術評論社
- [2] 岡本敏雄, 安田恭一郎(1992): C言語プログラミングのメンタルモデルの分析—診断助言型のITSを用いて日本教育工学雑誌16-3, pp.119-130
- [3] 前田恵三, 中野靖夫(1995): プログラム作成過程の分析, 日本教育工学雑誌19-3, pp.171-180
- [4] 前田恵三, 中野靖夫(1993): コンピュータ操作過程の再現システム, 日本教育工学雑誌16-4, pp.185-195
- [5] 海保博之, 原田悦子, 黒須正明(1995): 認知的インタフェース, 新曜社, pp.72-74
- [6] 海保博之, 原田悦子(1994): プロトコル分析入門, 新曜社

研究発表 5

電子情報通信学会技術研究報告

ET 98-72~92

〔教育工学〕

1998年11月20日

プログラミング過程の認知面からの考察

前田 恵三

東京工業高等専門学校
情報工学科

中野 靖夫

上越教育大学
学校教育研究センター

〒 193-8610 東京都八王子市櫛田町 1220-2

Phone: 0426-68-5064

E-mail: keizo@sp.cs.tokyo-ct.ac.jp

〒 943-0834 新潟県上越市西城町 1-7-2

Phone: 0255-25-9147

E-mail: nakano@juen.ac.jp

あらまし

C言語を用いたプログラミング時のコンピュータ操作過程(操作プロトコル)と発話プロトコルデータを収集し、プログラミング過程を認知面から考察を行なった。なお、プログラミング終了後、被験者に自分の操作の再現を観察しながら再び説明を行なってもらうことで発話プロトコルの補完を行なっている。プログラミングに慣れている熟練者の、プログラム作成手順、使用したスキーマや方略が、一部ながら解明できたと考えている。

キーワード プログラミング, コンピュータ操作, 問題解決, 認知過程

A Case Study of Students' Programming Process from Cognitive Side

Keizo MAEDA

Tokyo National College of Technology
1220-2 Kunugida, Hachioji, Tokyo 193-8610, Japan
Phone: 081-426-68-5064
E-mail: keizo@sp.cs.tokyo-ct.ac.jp

Yasuo NAKANO

Joetsu University of Education
1-7-2 Nisisiro, Joetsu, Niigata 943-0834, Japan
Phone: 081-255-25-9147
E-mail: nakano@juen.ac.jp

Abstract We analyzed and considered students' programming process in C language from cognitive view point. Used protocol data consists of speech protocol and computer operation protocol. This paper reports students' schema structure and strategy which were used to build a model and write chips of program code.

Key Words programming, computer operation, problem solving, cognitive process

1. はじめに

これまでプログラミング教育に長年携わってきたが、初歩的な段階における文法エラーからの回避は、演習を繰り返すことで、多くの学生は次の段階へ移行していく。しかし、手本となる例題を変更するのではなく、最初から自分で作成していくプログラミングにおいては、多くの演習問題を解いても、プログラミング能力が向上しない学生と、簡単な例題だけで、一息に向上してしまう学生がいることを見てきた。

プログラムの作成は、計算問題のように一つの解を求めるのとは異なり、各自が解き方を見出し表現する作業になる。

既に身に付けているスキーマを使って解けることもあるが、新規に出会う課題内容では、身に付けていたスキーマを部分的に取り入れたり組み合わせたりしながら、新たにスキーマを構築していくことが必要になる。

以前に、学生がどのようにプログラム作成に取り組んでいるかを見るために、プログラムを作成していく過程のキー入力を逐次記録しておくことで分析を行ない、学生のプログラミングスタイルが4種類に分類できることを示し、それぞれのスタイルに合わせた指導方法の提案を行なった[1][2]。

しかし、プログラムの構築過程は把握できても、そのときに何に反応し、何を考えていたかまでは把握することはできなかった。

そこで、本研究は被験者にそのとき考えていることを口に出しながらプログラミングを行なってもらい、発話プロトコルを収集し、同時にコンピュータの操作過程を逐次記録し、操作プロトコルとした。プログラム作成後、プログラム作成過程をコンピュータ上で再現し、それを被験者と分析者が一緒に観察しながら、そのとき、何を考えていたかを被験者に話してもらい、発話プロトコルの補完を行なった。そして、収集した発話プロトコルと操作プロトコルを元に、考えを整理しながらプログラムを作り上げていく過程を認知面から解明することを目的とした。

被験者はプログラミング経験が豊富であることと課題が小規模であるため、課題に取り組むまでの準備の必要がなく、コンピュータを操作していたときのプロトコルがプログラム作成時の認知過程を反映していると言える。

プログラム作成は、(1)与えられた課題文を理解し、(2)モデル化して解法のアルゴリズムを構築し、(3)そのアルゴリズムをプログラムにコード化し、

(4)動作させながら完成させていく段階に分けられるが、本研究では(2)～(4)について検討を行なった。

熟練者でも、すぐに全部のプログラムコードを頭に思い浮かべる訳ではなく、大まかなスキーマによってプログラムを入力しては詳細化していく過程が把握できる等、プログラミング過程を認知面から一部ながらも明らかにすることができたと考えている。

また、熟練者の用いたスキーマやスキーマを構築していく過程を、HowTo的なものとして、初心者教育に役立てられると考え、今後はこのことについても検討を続けていきたいと考えている。

2. 実験方法

2.1 データ収集方法

MS-DOS上のTURBO Cを用いてプログラムを作成した操作過程を、操作の記録・再現システムKBHIS [3]を用い、キーコードとキーが押された時刻によって逐次記録した。また、ディスプレイ表示と被験者の発話をビデオテープに記録した。

引き続き、操作終了後、記録したキー入力データをKBHISによってコンピュータに再入力することで、コンピュータ上でプログラム作成過程の再現を行なった。それを被験者に観察させながら、何をそのときに考えていたのかを話してもらい、また、分析者が気付いたことを質問し、その時の会話とディスプレイ表示をビデオテープに記録した。

そして、これらから発話プロトコルと操作プロトコルを文書化した。

2.2 被験者

C言語のプログラミングに熟練している高等専門学校の卒業研究生2名。データ収集時の頃には、被験者Aは、移動ロボットの製作とその制御用プログラムの作成をC言語を用いてワークステーション上で行っていた。被験者Bは、インターネットのWWWのテキストデータを集め、自然言語処理でキーワードを抽出するプログラムをワークステーション上でPerlとklicを用いて作成していた。なお、どちらの学生もエディタには普段emacsを用いていた。

二人は1年生から3年生までTURBO Cを使ってプログラミング教育を受けている。

2.3 課題内容

「与えられた正の整数値が素数であるかどうか調べるプログラムを作成しなさい」

という簡単なプログラムであるが、プログラミングに慣れている被験者であることから、効率がよく、安全なプログラムを作成することに努力すると期待した。

この課題は、被験者が3年前に授業で解いたことのあるプログラムであるが、本人達の記憶には、あまり残っていないようであった。

2. 4 分析の視点

プロトコルを分析することで学生がどのような方略で臨んだかなど、以下のことを明らかにしたいと考えた。

- (1) どのような方針、どんな手法、アルゴリズムを用いようとしたか。
- (2) 考え違い、考え不足によって、どのようにしたか、どのような影響を受けたか。
- (3) いつものやり方、使い慣れた手法、癖（フラグを使った考え方が好きとか等）。
- (4) for と while の使い分け、表現のしやすさ。
- (5) 画面表示内容と認知との関連。
- (6) プログラムを作成していく各段階におけるスキーマの変遷。

これらのことを明らかにすることで、演習時に教師が適切なコメントやフィードバックを与えることができ、学生が自分の体験から多くを学ぶことが可能になると考えた。

特に、スキーマの使い方、作り方の練習を積むことが大切と考えている。中野が試みた手法 [4]、「学生がプログラムのプリントアウトに書き込みをしていくことで、自分独自の学習記録ノートを作成し、それを何度も見返すことで、自分の学習・思考過程の見直し・理解が行なえ、そしてプログラミング能力が身に付く」にスキーマのノウハウ的な教示を加えていくことによって、プログラミング能力を向上させることが可能となると考える。

3. 収集したプロトコル

プログラミング過程を発話が豊富であった被験者Aを中心にして、作成段階に沿って追っていく。

二人の被験者とも、課題の文を読むと、まず、`#include`などの決まり文句的な部分をすぐに入力している。このときの発話は、図1のように説明的なもので、声も解説口調であった。あまり思考を必要としない部分と考えられる。

次に、必要な変数と処理の骨格の入力になる。結果として素数であるか否かを表示するプログラムになるはずであるのに、二人ともキーボードから

値を読み込んだ後に結果として値を表示するプログラムを入力している。被験者Aは入力時に気付いたが、被験者Bは気付かずにそのまま入力が続けている。

プログラミング過程を再現したときの説明では、最初の時点で、入力された値が素数であるか調べるには、その値より小さい数値で割ってみて、すべて割り切れなければ、素数であると判断する方針があったことが分かっている。この大きな方針に沿って具体的にプログラム化していく訳であるが、処理を入力しながら、より具体化していった様子が分かる。

図2画面1の発話では、「2から割り始める」ということで変数宣言の段階で初期値に2を設定し、ループの記述になって、while か for か迷い、for を採用したことで、変数宣言の初期値が不要になり削除している。

また、図2画面3では、for の繰り返しの条件で平方根の関数 `sqrt` を入力して悩み、平方根をループの外側で計算するように変更している。このように、入力しては迷い考えという最中では、発話は独り言に近くなり、発話内容も被験者が注目しているキーワード的なものになっている。そして、方針が決まった後で、平方根の計算をループの外に記述するときには、自信のある口調で、「エヌは入力された数の平方根です」という解説的な発話になっている。

図2画面4では、関数で求めた結果について、一度変数に保存しようとして、変数名の命名を考えている。結果 (`result`) の `r` にしようとして入力したが、それを消し、フラグ (`flag`) の `f` に決め、入力したが、その後で、この変数が必要でないことに気付いている。注意が変数名の命名に注がれていて、他のことを考えられず、入力を終えたことで、客観的にその変数を見られるようになったと考えられる。

同様に、プログラムが動作しだした図2画面5での、「2は素数なんだね」という発話も、2を除外しなくては、ということに注意が向いたために、その実現方法に考えがいき、入力したことで、他にも注意が広がって、for 文の動作原理から不要であることに気付いた、と考えられる。

また、画面3で関数の `return` を2度入力しているが、それは、以前に入力した内容が、被験者の短期記憶から薄れ、その内容が画面に見えていなかったために、再び同じ処理を記述してしまったと言えよう。画面2の閉じ括弧も同様である。言い換えると、注目していること以外のことは、たとえ

CRT 画面のイメージ

```
#include <stdio.h>

void main( void ) {

    int i ;

    printf( "input > " ) ;

    scanf( "%d", &i ) ;

    printf( "anser : %d \n"
           すぐに、消す
    )
}
```

```
-----

#include <stdio.h>
int sosuu( int i ); /* */

void main( void ) {
```

```
#include <stdio.h>

void main (void) {
    int input_n ;
    int result_n ;
    printf(" input number > ");
    scanf( "%d", input_n );

    printf(" ", %d") ; /* result */
}
```

プロトコル

(括弧内はキー入力, ▼はリターン, □は空白)

え、まずはスタンダードアイオーのインクルードして、(#include □<stdio.h> ▼)
 え～、まずはメインから、(void □main(□void □) □{ ▼)
 え～と、まずは入力され (int □) た数を入れる変数 (i □; ▼) を
 え～、これを (sc) スキャンエフ (anf) で、(scanf を消して) (printf("input □>□") □; ▼) プロンプトを出して、
 スキャンエフ (scanf(□"%d") で読み込んで (□&i □) □; ▼)、う～んと、
 え～と、結果の (printf(□) 表示を ("¥n □anser BS BS BS BS BS BS anse: BS r □ : %d □ ¥n")、
 え～と (anser を消す)、どうしよう。う～ん、素数であるかどうか、え～、
 じゃ、ちょっと、(▼) ↑↑↑↑↑↑↑↑↑↑←←←←
 ▼最上行へ)

まず、じゃ、素数の判定をする関数を作ります。素数(プロトタイプ宣言 int □sosuu(□int □i □);)。あ、え～、(□/* コメント入力 ^XFER^XFER) あ、f e p 走んない (* / ▼)。素数だったら 1 を返すような (カーソルを最下行へ移動)、関数です。

(a). 被験者 A のプロトコル

(#include) インクルード (<← □)、あ、そうか、カーソルを使わなきゃいけないんですね、これ。(stdio.h) ▼
 う～ん (void □main □(void) ▼{ } ←← BS □ → ▼▼
 ↑ ▼TAB printf(" □", □ %d") □; ←←←←←←←←←←
 ←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←
 →→→→→→→→→→ □ /* result □ */ ↑ ↑ ←← ↑ ←←
 ←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←←
 ▼TAB int □ ▼) やっぱ、これ、scanf 必要じゃないですか。はあ。

(b). 被験者 B のプロトコル

図 1. プログラミング開始時のプロトコル

CRT 画面のイメージ

```
scanf( "%d", &i );          画面 1
printf(

int sosuu( int i ) {

    int m = 2;

    for(m = 2 ; m < sqrt( i ) ; m++) {
        if( i % m == 0) return 0;
    }
    return 1; ← 画面に見えていない
} ← main の閉じ括弧だったもの

-----
} ← 余分                                画面 2
return 1;                                画面の外で、
}                                         見えていない

-----
int sosuu( int i ) {                    画面 3
    int m, n;
    n = (int) sqrt((double)i) ;
    for(m = 2 ; m <= n; m++) {
        if( i % m == 0) return 0;
    }
    return 1;
}

-----
void main( void ) {                    画面 4
    int i ;
    printf( "input > " ) ;
    scanf( "%d", &i ) ;
    if ( sosuu( i )) {
        printf("yes\n");
    }
    else
        printf("no\n");

-----
int sosuu( int i ) {                    画面 5
    int m, n;
    n = (int) sqrt((double)i) ;
    if ( i == 2) 入力してから消す
    そして
    if ( i >= 2) これもやめて
    if ( i < 2) return 0;

    for(m = 2 ; m < n; m++) {
```

プロトコル

(括弧内はキー入力、▼はリターン、□は空白)

えーと、(int | | sosuu(□ int □ i □) □ { ▼)
うーん、(int) まず、えーと、(i) えー、割って、割
る数を、決めます。(BS で i を消し、m □ = □ 2; ▼)
(while) で、うーん、フォー (while を消す) でい
いかな (for()。えーと、じゃあ、これ、やめて (変
数宣言の =2 を削除)、エム 2 (m □ = □ 2 □;) か
ら、えーと、えーと、エムが (m □ < □) えーと、
(sqrt(□ i □) □;) 平方根まで (□ m++) { ▼) たし
て。(if(□ i □ % □ m □ == □) えー、どうしよ
うかな。ふーん。エムが。
(0) □ return □ 0 □; ▼ } ▼ return □ 0 □; ← ← BS 1 }
↑ ↑ ↑) えーと、(最上行へ) あれ、(カーソルを
下へ。最後の return 1 が見えずに、その前に余分
な } を入力)。
(最上行へ。↓ ↓ ↑ #include □ <math.h> ▼ m の上
で、ヘルプ f.1 で sqrt をヘルプ階層の先頭から探す)
ええーと、ダブルで、(for の条件を m < (int) sqrt(i) と
してから) あ、やっぱり 「n」 エヌ、
エヌは入力された数 (n □ = □) の平方根です。
((int) □ sqrt((double)i) □; ▼) (for で m < n にした後
m <= n)
(return □) で、こうじゃなかったら (1:) 1 を返
します。あれ? (既にあった return 1 を削除)
うーん、で、
(カーソルは int の上) 返された数をアール (r) に
して、あ、フラグ エフ (f) にしましょう。フラ
グ。(scanf の下で) ああ、やっぱ (int にカーソル)、
いらなかな、いらなかな (言いながら削除)。
(if) もし、えー、素数 (sosuu(□ i □))、が、(f)
ツールーだったら、(printf() イエス ("yes\n"); ▼)。
(printf("no\n"); ▼) ノー、これで、いいかなあ。(関
数 sosuu へ移動、そして 1 行目へ)
(f.2^f.9 保存・実行) あ、じゃ、3 (3 ▼)。
あ、2 は素数なんだね。じゃ、困ったな。2 はど
うしよ。うーん。うーん、えー。(for (m=2 の上で)
つう、ま、あ、2 の場合は、(for の前の行へ) そ
のまんま、(if (i □ == □ 2)) 2、2 だったら、あ、
いいの (条件を消す)。え、1 だったら、1 は素
数じゃないものな (>=2)。i が (<2) 2 より小さかっ
たら、(return 0) 素数ではないと。(保存) (実行。
2 入力) できているかなあ。(メニューをいじる)

図 2. 被験者 A のプログラミング過程 (部分)

それが操作してから時間があまり経過していなくても、画面に表示されていないと、記憶から薄れてしまうと考えられる。

その他、プログラミング作業を観察して言えることは、プログラムを作るときは紙にコーディングをして考えを整理するのではなく、キーボードでプログラムを入力して画面に表示された内容を見ながら考えを整理していつている。そして、入力したプログラムを見直すときは、見る位置にカーソルを移動していることが殆どであり、全体を見る時だけ、カーソルの上下移動になっている。一度に見ることのできる行数は xx 行と少なく、見えていない行についての注意力は弱くなる。プログラミング過程の分析では、画面に表示されている内容と、カーソルの位置、動きにも注意を払う必要がある。

メインフレーム、パンチカードの時代のプログラミングでは、紙にプログラムを書いて再検討し、ラインプリンタに打ち出されたプログラムリストをカラーペンを持ちながら追跡していたが、そのときに一度に見ることができた範囲に比べると、現在のパーソナルコンピュータの画面は狭すぎると言え、プログラミングに与える影響が大きい。

4. 結果の考察

課題の内容が小規模なため、被験者はシステム分析などを行わずに、課題文を読むと、すぐにコンピュータに向かいプログラムの入力を始め、プログラムの完成までコンピュータに向かっていったことから、発話プロトコルと操作プロトコルはプログラム作成の認知過程を反映していると言える。

また、コンピュータに向かってプログラムを作成していく作業は CRT 画面の上に思考を外化しているといえ、操作プロトコルには認知過程の一部が現われている。被験者は発話思考法に慣れているとはいえないため、発話プロトコルと操作プロトコルの両方を使って分析することで、プログラム作成時の認知過程や方略を明らかにしていくことができる考える。

まず、問題解決に関する既存の理論や説がプログラミングにも適用できることを述べ、次に、被験者のスキーマの形成過程からプログラミング過程の分析を試みる。

4.1 操作のカテゴリ分け

図1に示すように、課題文を読んだら、`#include` から `printf` まで定型的な場面では、あまり考えずに

入力している。発話はその解説を行なっているだけである。値の出力は必要でないのに、二人とも値を表示する `printf` を入力している。何度も演習を繰り返してきて、習慣化した作業になっていると言える。Norman が述べている体験的（反応的）認知（*experiential cognition*）に該当する。単純な繰り返しや条件判定などの記述では、表現しようとしたことが無意識にコード化され入力されているのが普通である。

スキーマを形成していく過程では、内省的認知（*reflective cognition*）になり、発言も独り言に近くなり、発言だけでは他人には意味がつかみにくいものになっているが、そのときの画面の内容と操作プロトコルから何を考えているのかを想像することは可能である。

プログラミング時の操作を認知面から以下のようなカテゴリに分けることができる。

- (i) これまでの経験から反応的に入力 … 解説的な発話か無発話
- (ii) 見通しが立った処理の入力、簡単な誤りの訂正 … 解説的な発話
- (iii) 思考を整理しながら入力・編集 … 独り言的な、あるいは気付きの発話
- (iv) 誤りの推理・発見、論理の見直し … 独り言的な、あるいは気付きの発話

以上は、発話内容と操作プロトコルから判断が可能である。

これらは、以下の Rasmussen の 3 階層モデルにも当てはめることができ、目標達成のための行動としては一般的なものであることが分かる。

- (1) スキルベースの行動 … (i) が該当
- (2) ルールベースの行動 … (ii) が該当
- (3) 知識ベースの行動 … (iii) と (iv) が該当

4.2 問題解決の過程

目標達成のための認知モデルに Norman の 7 段階（目標、意図の形成、行為の選択、実行、状態の知覚、状態の解釈、評価）があるが、プログラミングのような解法を求める場合は、意図の形成、行為の選択がもっと細かく分かれている。この一般的なモデルをプログラミングに当てはめて考察するのは適当でないといえる。

伊藤ら [5] は問題解決の過程を以下の 5 段階に分け、この一連の解決過程を問題解決スクリプトと呼んでいる。

- (1) 問題理解の段階 … 与えられた問題の意味を理解する段階
- (2) 解法探索の段階 … 解決者の知識ベースの中

- から問題に当てはまる解法を探索する段階
- (3) 解法適用の段階 … 解法を利用できる形にして適用する段階
 - (4) 解法吟味の段階 … 解法を適用して得られた答えが題意に合うかを吟味する段階
 - (5) 答案作成の段階 … 解決者内部の解決を外に表現する段階

プログラミングの作業も(5)を除くと同じ様に考えられるが、それぞれの段階が時間的、内容的に重なりあっている傾向が見られる。問題を十分に理解する前にプログラムを書き始め、気になった部分の解法、アルゴリズムを確かめようとする。また、プログラムを作成しながら(解法を適用しながら)、適当な解法を探し、試してみる(解法吟味の傾向もある)。

実際のプログラミング過程を具体的な言葉で表現すると以下のようなになる。前述の問題解決スクリプトに近い。この過程の表現だけでは、トップダウン的な作業に見えるが、実際の作業では、各処理は時間的に重なっており、考えを整理する段階や方針を決める段階でボトムアップで確認を行ない見通しを立てている。

- (1) 問題の理解
- (2) こうやろう、という方針(背骨)を立てる
- (3) それを実現するための枝葉的目標の作成
- (4) 実現するための良いアルゴリズムの導出、発見
- (5) プログラムへのコード化。処理単位への切り分け、ループ構造、関数、再帰
- (6) 動作の確認とプログラムコードの見直し

プログラミングが苦手な学生は、初期の段階ではプログラミング言語の機能を使いこなせず(5)の段階でつまづく傾向が見られるが、課題が高度になり、内容が現実的な問題になってくると、コンピュータ向きに抽象化が行なえず、(2)(3)(4)などでつまづくようである。

今回の課題では、内容は単純であり、被験者が熟練者であることから、つまづく過程は見ることができなくても、初心者の参考になる方略が得られると考えた。

被験者Aのスキーマの形成過程を追うと、図3のように表わせる。

課題に着手した段階から全てを見通していた訳でなく、プログラムを構築しながら具体化されていったことが分かる。これまでのプログラミング経験から、不足していること、考えなくてはならないことにプログラムを入力しながら気付いているのではないかと考えられる。初期の段階で

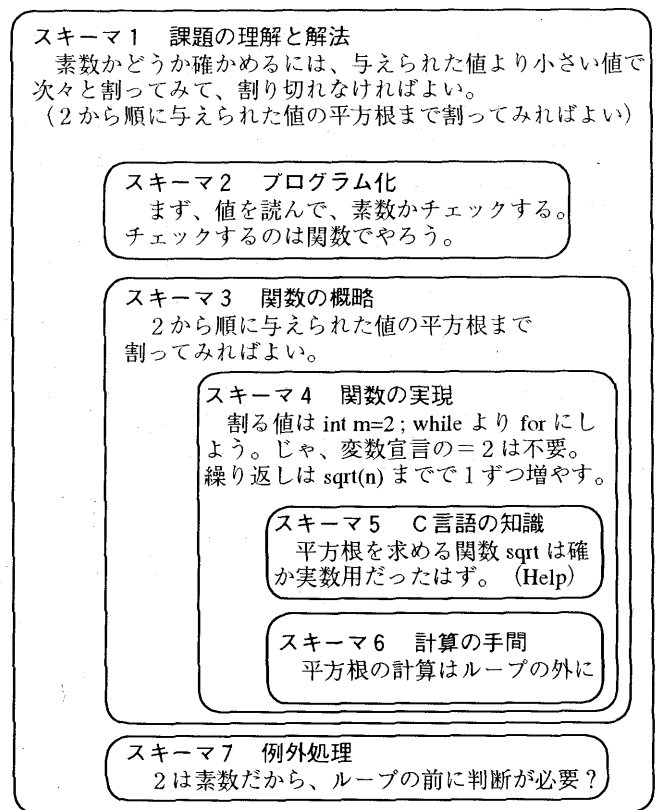


図3 スキーマの形成(被験者A)

何をおさえ、どのように発展させていけばよいか、という過程は初心者への教示に有効であると考えられる。

プログラムのコード化では、入力する段階で判断が行なわれており、アルゴリズムを組み立てる思考がプログラミング言語によって行なわれていないことが想像できる。この作業は Rasmussen のルールベースの行動に該当し、イメージしている手法を規則的にプログラミング言語に変換している作業といえる。プログラムにコード化される前の表象について、初心者と熟練者でどのような違いがあるのか調べてみる必要があると考える。

なお、被験者Bも同様のスキーマを形成しているが、素数かどうか調べるのに使う割っていく数を、2からではなくて、大きい値から順に2まで割っていく方針を立てている。スキーマを形成するときに約数の性質に関するスキーマが取り込まれなかったためと考えられる。

その他、平方根の計算に要する手間、2で割った後は偶数で調べる必要がないなどがスキーマとして取り込まれていない。操作再現時の聞き取りで、分析者から言われると気付くことから、知識としては持っていたも、スキーマとして取り出すことができなかつたようである。

実験の後、メタ認知についての解説を行なったが、被験者Aはメタ認知をしている自覚があると答えたが、被験者Bは考えてもみなかったと答えた。プログラミングに要した時間は被験者Aが19分、被験者Bが28分である。被験者Bがforの条件記述誤りや割る数に自分自身まで含めてしまったなどの些細なミスからの回避に手間取ったことが何度かあったことと合わせて考えると、メタ認知との関係もあると考えられる。

5. 関連研究

プログラミング過程について分析を行なった研究には、大谷らの報告[6]と、竹市らの報告[7]があるが、プログラムを入力しているときの認知的研究ではなく、問題理解・解決に関する研究の傾向が強い。

大谷らの研究では、Prologのプログラミングについて、プログラムの動作や概念を具体例と結び付けることで学習効果が上がることを確かめている。スキーマを形成する手助けを行なう方法が参考になると考える。

竹市らの研究では、プログラムを作成するときに行なうモデル化とアルゴリズムの構成の過程で「行為者スキーマ」と中間レベルの表現「部分的抽象記述」を導入し、スキーマからモデル化・アルゴリズム記述が正しく行なえたかを見ている。そして、課題内容や課題文の記述がモデル形成にどのような影響が出るのかを見ている。思考過程のプロトコルを採ってはいないが、アルゴリズムを考えたときの表象と、課題文から導かれるスキーマについて参考になる。

この2編の報告では、我々の研究で着目しているプログラミング過程、実際のプログラムを入力・編集していく作業においてスキーマが具体化されていく過程、については検討されていない。

また、学習の対象が数学ではあるが、伊藤らは[5]、思考を作図によって外化したときの問題解決過程について論じている。プログラムをCRT画面に記述することで思考の外化を行なうのと類似しており、分析のアプローチが参考になる。

6. おわりに

プログラミングを目標達成、問題解決の過程として捉えると、既存の理論・説で把握・説明することが可能である。しかし、創造的問題解決(Creative Problem Solving)であるプログラミングでは、コンピュータに入力しプログラムコードとして思考を

外化して自分の考えを整理しスキーマを形成していく過程が、自分なりの解法を構築していく創造過程であり、プログラミング能力に大きな違いを生じる場所である。

本研究の分析によって、熟練者のスキーマの形成過程の一部が明らかにできたことは、今後、プログラミング教育を指導していく上で意味のあることであったと考える。

また、発話プロトコルと操作プロトコルを併用する手法は、プログラミングのようなコンピュータに向かいながらの創造活動の分析に有効であることが分かった。今後、オーサリングツール等を用いた創造活動の分析等にも応用していく予定である。

なお、本研究は科学研究費基盤研究(C)課題番号(09680209)の研究成果の一部である。

参考文献

- [1] 前田恵三, 中野靖夫: プログラム作成過程の分析, 日本教育工学雑誌, Vol.19, No.3, pp.171-180, 1995
- [2] 中野靖夫, 和泉嘉則: 中学生のプログラム作成過程, 日本教育工学会研究報告, JET97-1, pp.75-82, 1997
- [3] 前田恵三, 中野靖夫: コンピュータ操作の再現システム, 日本教育工学雑誌, Vol.16, No.4, pp.185-195, 1993
- [4] 中野靖夫: 学習者個人に学習過程を記述させる, 赤堀侃司編ケースブック大学授業の技法, 有斐閣, pp.232-235, 1997
- [5] 伊藤毅志, 大西昇, 杉江昇: 人間の作図過程を説明する問題解決スクリプトと作図の分類, 電子情報通信学会論文誌, J77-D-II, 4, pp.811-822, 1994
- [6] 大谷紀子, 河辺信吾, 沼尾正行: 複数の情報媒体を用いた学習の過程, プログラミング言語Prologを題材にして, 認知科学, Vol.2, No.1, pp.72-85, 1995
- [7] 竹市博臣, 市川伸一: シミュレーション・プログラム作成の思考過程, 行為者スキーマと部分的抽象記述の役割, 認知科学の発展5, 講談社, pp.187-200, 1992