

ドリル型 CALL ソフト活用のあり方と 備えるべき諸機能について

— 学習者の負担を軽減し、効率よく学習効果を上げる
ために英語教師はドリル型 CALL システムの開発者に
どんな機能を求めるべきか —

佐々木 勝 志

はじめに

コミュニケーション活動を重視する英語教育の流れの中で¹⁾、ドリル型の CALL (Computer Assisted Language Learning) ソフトについては、特に 1990 年代半ばあたりからコンピュータの活用が ICT (Information and Communication Technology) としてインターネットを軸とする新たな時代に入ることによって²⁾、最近ではほとんど話題になることがなくなった。それにもかかわらず、本稿では英語教育のある部分におけるドリル (練習) の必要性について論じ、その立場からドリルタイプの CALL 教材の有効な活用にとって意外に障碍となっている技術的諸点についてまとめることで、その意義を明らかにしたい。特に技術的諸点をめぐっては、今後自作用の教材作成システムを導入しようとする場合に専門の技術者に要望すべき諸点を整理することでその有効な活用の一助となればと思う。

コンピュータを活用した英語教育は 30 年ほど前から盛んになったが、

当初は Sunday programmer と称される英語教師自らが開発した手作りのソフトがマイコンの普及を背景として流れを加速させた。そこで手作りされ使用されたソフトの多くはドリル型と呼ばれるもので、手書きで空欄を埋たり選択する紙（プリント）による練習問題や小テストをコンピュータの画面に置き換えたようなものが多くを占めた。そのため、その後、コンピュータのハード・ソフト両面の急速な発展と音声・動画などマルチメディアによる高度な技術が活用されるとともに、さらにはインターネットの普及によって世界を結ぶコミュニケーション活動がより容易な環境が作り出されるなかで、その機械的な練習のマイナス・イメージもあって最近ではドリル型の学習ソフトはあまり評価の対象になっていないように思われる。しかし、全く無視されているかというところではなく、筆者の勤務校でも *Wordengine* というソフトが学生に推奨され、特に TOEIC や英検などの対策の一つ手立てとして活用されている。よく見ると、それは CALL 教材を使うか否かは別として、英語学習において何らかの形でドリル（練習）が必要であり、その理由と位置付けについては再度整理しておく必要を示している。

従って、まず、ドリル学習の必要性に関わって整理しておきたい。

1 ドリル学習の必要性と CALL 教材の活用

多くの学生が試験前になると出そうなところを丸暗記しようとする。このような現象をどう評価すべきだろうか。もしそれが試験に合格するための一時的な手段に過ぎないとしたら学習者は、必ずしも言語を習得し活用する上で有効なものとしているよりは、目先の「単位を落とさない」という次元でのみそうしていることになる。勿論、それだって、「結果として練習になるのだからやらないよりは力になっていいじゃないか」という見方もあるだろう。しかし、それは効果的・効率的な方法だと言えるだろうか。

EFL (English as a foreign language) の環境においては、学習した表現を ESL (English as a second language) の環境のように自然な日常生活の中で再びかつ頻繁に経験する可能性は極めて低い。したがって、外国語習得は技能習得であるため、日本での場合のような EFL の環境においては特に「練習活動が中心的な役割を担う。」³⁾

この場合の「練習活動」に冒頭の試験前の丸暗記を入れるべきか否かの判断は簡単ではないが、少なくとも短期記憶に一時的に留まるとしても長期記憶へ繋がる割合はかなり低いのではないか。丸暗記の対象となる様々な表現は、具体的な文脈で学習（教師からすると説明）されているはずだが、試験直前の丸暗記はそのようなことを念頭に置くことが可能な復習としては時間が経過しすぎて、和訳と対応する英語表現の一時的な記憶に終わってしまっていないか。

指導の仕方がうまい教師のおかげでそのとき習ったことはその場ではよく分かった。ところが復習がおろそかで、次の事項を学んだところ忘れてしまった。あるいは以前の学習事項を忘れていて、新たなところでそれを活用する必要があるが使いものにならなかった。これらのことは例えば文法など特に論理的な積み重ねが求められる学習分野ではよくあることで、既習事項の定着が強く求められることになる。

こう考えた時、理論的な説明を軸に学習者に講義するというだけではなく、その内容を学習者自身が自己に定着するプロセスを、学習者の自発的努力は当然前提としつつも、教師自身が指導過程にしっかりと組み込むことが重要であり、ドリル型 CALL 教材はその点で有効な手段だと言える。

1-1 ドリル型 CALL 教材の効果的活用の前提条件

かつて筆者は、CALL 教材の種類を大きく Drill Type、Tutorial、Simulation という 3 つの形態に分類し、その効果について論じたが⁴⁾、そ

こでの Drill Type の効果については次のように整理した。ドリルには2つの目的があり、一つはテストの場合と同様に、どれほど学習した基本原理が習得できたかを確認すること、もう一つは、習得した基本原理に基づく知識を増やしたり確実にする、またそのことで基本原理についての認識と実用的な技能をも確かなものに定着させる、ということがある。

ここで重要なのは、ドリルの目的は基本原理そのものの学習ではないということである。本来の目的である基本原理の習得にとって、その定着を練習によって確実にするための言わば手段である。従って、基本原理をある程度学習しないでドリルをしても意味がないが、他方、ドリルによってその定着を図るということからすると、必ずしも完璧にこの基本原理を習得していなくてもよいことになる。問題は、基本原理がある程度わかっていることを前提としないと、結果として単なる丸暗記を強制し、ドリル自体が無意味なものになってしまうということである。その意味で教材作成者は、学習者のこの点での習熟状況を常に念頭に置く必要がある。ドリル型の CALL 教材においてもこのことを常に意識しておくことが重要である。

他方、ドリル型の CALL 教材は、教師が直接指導をしなくても学習者がそれを用いて「練習」することによって教師が指導したことがらを練習し定着させる手段である。ちょうどプリント教材などによって学習者に課題（宿題）を与えるのと同様である。ただ、プリント課題の場合は教師がその達成状況を確認し指導する必要があるが、CALL 教材では直接指導しない自学自習にポイントがあり、しかもその達成度を採点業務なしで教師が把握できることが特徴である。プリント課題に比べて教師の負担を軽減するだけでなく自動採点による学習履歴に基づくフィードバックなどによって学習者の学習効果を高めることが期待できる。

なお、実際の活用においては既に述べたように一般のドリル課題同様、教師は CALL 教材によるドリルを指導過程にしっかり位置づけておく

必要がある。

1-2 コミュニケーション活動とドリル型 CALL 教材の活用法

学習事項の定着の上で効果が期待されるとは言え、ドリル型 CALL 教材はその活用の仕方を間違えると却って逆効果となりかねないのも事実で、前節で述べたように指導過程での位置づけが求められるが、コミュニケーション能力・運用能力の育成を課題とする今日の英語教育においては、それがあくまで補助的手段であることを留意したうえで、有効に活用すべきものと思う。その点をさらに明らかにするために、ドリル型 CALL 教材の直接の機能が類似していると筆者には思われるパターン・プラクティスとの関わりで考えてみたい。

平嶋(2007)では、日本でのフランス語学習の環境が英語における EFL と同様である趣旨のことを示してパターンプラクティスの有効性について次のように言う。

このような学習環境で効率よくコミュニケーション能力を養成しようとする場合、外国語学習成功者の例が示しているように、暗記やパターンプラクティスを活用して技能学習を促進するのは有効な教育方略だといえる。既に示した通り、コミュニケーション能力とは社会・文化的側面をも含めた総合的言語運用能力であり、コミュニケーションを成立させるためには言語内外の様々な要素に配慮する必要がある。意識せずに文法的に正しい文を作り出すことができれば、実際のコミュニケーション活動に必要な談話・社会・心理的要素により多くの注意を払うことが出来る。⁵⁾

パターンプラクティスは、文型パターンの練習の繰り返しで習慣形成することによる文法の習得自体が言語習得であるということを行動主義心理学と構造主義文法によって基礎づけたものだったが、それ自体が問

題であると言うより、この方法を自立的に一番優先すべきものとした点に問題があったと言える。ここで述べられているように、「総合的な言語運用能力」育成にとって、これを支えるものとしての文法力⁶⁾の育成方法としてパターン・プラクティスを位置づけるなら、有効な方法だと言える。なお、筆者自身は文法と言うよりもより機能的な視点⁷⁾で語彙・文法的 lexicogrammatical な要素の習得手段として考えたい。

いずれにせよ、パターン・プラクティスの欠陥はパターンの自己目的化であり、この練習方法を（実際の）コミュニケーション活動と有機的に結びつけて活用する⁸⁾なら極めて有効な方法だと言えるが、その点ではドリル型 CALL 教材も全く同様の効果が期待できる。既習の学習事項の中の定着した表現が次回の Unit の指導の中で生かされ、生かされるために定着が求められる、という関係をうまく継続していくなら、学習者にとっても負担を軽減するものとなり、限られた時間で効率的に活用可能な長期記憶へとつなげることが期待できる。

特に、英語ネイティブ教師による Speaking 指導では、聞いて理解するだけでなく、発信が強く求められるわけで、表現がしっかり身についてないと文脈上でおおよそわかっただけでは対応できない。日本人教師による Reading や Listening 指導で学習したことを定着させ Speaking クラスでその活用を促すような連携が求められるが、そのため、一定の文脈のあるテキストの学習を前提にその復習やまとめのためにドリル型 CALL 教材による表現練習が単元毎に授業から一定の時間をおいて行われることが望ましい。そして、そのための教材の作成を考えた場合、これに関わる技術的な事柄をある程度知る必要あるため次節でその点を述べる。

1-3 ドリル型 CALL 教材そのものとその作成システムの技術的観点からの検討の必要性

CALL 教材の作成と学習システムを考えたとき、それはプログラミング言語の知識がなくても教材作成が可能な①オーサリング・システム authoring system、このシステムによって作成された②教材そのもの、この教材をコンピュータの画面に表示して学習者が実際に学習出来るようにする③エグゼキューター-executor の大きく 3 つの要素から成り立っている。最近では開発したシステムについて独自の呼称を与えているため、これらは必ずしもそのように呼ばれているわけではないが、基本概念としてはこれらが分かりやすい。

教材作成者は、①オーサリング・システムを用いて教材を作成することになるが、筆者の経験ではドリル型の CALL 教材を作成する場合は、高度で様々な機能を有するものよりもできるだけシンプルで操作が容易なシステムを選択すべきだと思う。LL などでもそうであるが、色々機能がありすぎてそのために操作法が複雑になり、実際には大して必要ない諸機能のために非常に使いにくいものになっている場合がある。また、その使用法を作成者が習得するのも相当時間を要することにもなる。そのため空白補充問題程度のものを作るのに大変な時間と手間がかかるとしたら、紙によるプリント教材の宿題で良いと言うことになる。

オーサリング・システムでは、②教材そのものを作るのだが、それ自体は言わばデータである。コンピュータの画面にこのデータのどの部分を表示し、どこに解答を入力する部分を表示するかの情報をデータとして作るわけである。このデータ(教材そのもののデータと制御用データ)を用いて学習が出来るようにするのが実行するシステムである③エグゼキューターだが、これも様々な機能があるために学習者が複雑な操作を強いられないものであることが重要である。また、一見マウスでの選択はその点で分かりやすいように思われているが、教材の選択などの場面

は別として、学習そのものにおける解答入力場面では、マウスに持ち替えたりしないでキーボードだけの入力で一貫させた方が学習の流れを止めることなく学習者はスムーズに取り組めるようである。

なお教材を作成していく中で、あとから欲しい機能が出てくる場合があり得るが、かつてはオーサリング・システムやエグゼキューターに自作のプログラムを組み込むことができるものがあった。筆者自身はシステムの考え方として、その方が効率的ではないかと思う。

さて、ドリル型の CALL 教材で学習する場合、意外にちょっとした技術的な欠点がせつかくの教材を無意味にすることがある。とりわけ、「紙(プリント)による練習問題を単純にコンピュータソフトに置き換えることができる」というように考えると、実際には学習者自身や教師が無意識に当然のようにやっていることが、CALL 教材にすると実はそう簡単ではないことを見落とすことになる。そして、この点での技術的保証がないと CALL 教材でこんなに素晴らしいことが出来るということを強調しても単なる絵に描いた餅でしかなくなる。次章ではその点に関わる技術的諸点について具体的に例示しまとめることにする。

2 ドリル型 CALL 教材が備えるべき機能的諸点

筆者は、現在 Listening I、II と Pronunciation I、II という科目を教えているが、その両方で自ら開発した *DrillMaster* と呼んでいるドリル用エグゼキューターによって、学習事項の確認のための復習を促している。ただし今のところ基本的に強制はしていない⁹⁾。ここでの目的がドリル型 CALL 教材を効果的にするための技術的機能説明が趣旨であるため、以下 *DrillMaster* を例に諸機能を明らかにする上で必要な範囲で説明することにする。なおここでの機能とはエグゼキューターとしての機能である。

2-1 スペース処理

練習問題をコンピュータ画面で行う場合に、記号で選ぶ場合や単語 1 語だけを入力させる場合はあまり問題とならないが、より長いフレーズや文をタイプ（入力）させそれを評価（採点）する場合、一見単純だがやっかいな問題が生じる。

例えば次のように、

コンピュータの用意した正答	They are notable scholars.
学習者の入力解答	They are notable scholars.

They are notable scholars. が正答である英文を学習者に入力させるという場合、学習者が解答を入力中、スペースキーを 2 回押したために文の 1 部の単語と単語の間に文字 2 つ分のスペースができた時（上の例では、学習者の入力解答の are と notable の間のスペース）、人間が判断するなら正答とするだろう。しかし、コンピュータでは学習者が入力したデータと予め用意された正答を示すデータを照合 matching することになるわけで、当然のことながらこのままだと誤答とされる。

この場合に、〈1〉学習者に「単語と単語の間に 2 つ以上のスペースを空けてはいけない」という注意書きを画面に出しこれを処理することにしたらどうだろう。筆者もかつてそのようなやり方をしたことがあるが、学習者に大きなストレスをかけることになり、言わば「不注意」でスペースを 2 つ空けたものが誤答になって「わかっているのにどうしてこれが？」と反発され学習意欲を低下させるものとなった。

あるいはまた別のやり方として、〈2〉スペースが一つでないと正答にならないように解答欄の全文字数を正答の場合の文字数に固定するという対処法もある。そういった場合は、偶然同じ字数で誤答を入力する場合を除いて学習者が正答にたどり着けない場合に、何度も試行錯誤を強

制することになり、同様に大きなストレスになるばかりではなく、本来のものと異なる要因で多くの時間を無駄にすることになる。

そのほかの対処法としては、〈3〉正答も入力したのもも全てスペースを取ってしまうようにフィルターをかけるやり方もある。その場合、例えば They are notable scholars. と They are not able scholars. とはスペースを取り除くと同じ文字列になってしまう。もちろん文脈上そんな取り違えはない、というふうには言えなくもないが、様々な問題の中には間違った理解の元に学習者が解答したとしても結果として絶対に正解扱いにならないとは言えない可能性があり、よい方法とは言えない。

以上〈1〉～〈3〉で見たようにこの問題は意外に厄介だと見えて、筆者が過去に目にしたドリルタイプのCALL教材では、全文を入力させるタイプのものは多くはないように思える。しかし、ドリルの目的が単に知識の確認だけではなく、練習による定着と言うことになると、記号や単語一つの入力ではその必要性に十分に応えることはできない。文全体を入力させないまでも、フレーズ単位で複数の単語の入力に対する対応は最低限欠かせないと言える。

ではどのような対処法が他にあるか。*DrillMaster* では、スペースが2つあるいはそれ以上あった場合でも正答とする、というように対処している。より具体的には、文字データにフィルターをかけ空白が2つ以上ある場合は空白一つに変更する。また、この場合、教材作成者による正答データ自体もその入力の際にスペースの数を間違わないとは言えないので同様にこのフィルターを通す。さらに細かいことでは、文末のピリオド(.)、クエスチョンマーク(?)、感嘆符(!)、などは逆にその前の最後の単語との間が空いては誤答扱いになるため、間のスペースを取り除くようにフィルターをかける。もっとも文の最後にこれらの文字の入力を求めるような全文入力問題を避けるならこの問題は生じない。ところで、一文全てを入力させるように作問した場合に、入力ミス

を避ける目的で文末にこれらの文字を表示するように教材を作成すると、意外と学習者自身が見逃しピリオドを入力してしまうこともあり得るが、これについてはここで述べているフィルターでの処理がよいかどうか疑問があるため、その対処法をさらに次の節「2-2 複数正答対応」で論ずる。

なお、ここで見た学習者の入力の正答との照合に関わることは、多くの英語教師の間では一見英語教育とは関係のない単なる技術的なことのように思われているようだ。しかし、既に見てきたような、人間なら簡単に見分けて対応できるのに機械であるが故にできない、そのために学習者に本来的ではないことで気を遣わせることによるストレスとその対処の必要性については、コンピュータ技術者も基本的に気がつかないか、自分の仕事と考えてはいないようである。ドリルタイプに限らず CALL の教材作りにとって、このことは実はかなり決定的であると筆者は考えているが、その点については、最後に「まとめ」の中でさらに論ずることとする。

2-2 複数正答対応

学習者の解答入力についての対応でもう一つ重要なことに、複数の正答について対応するということがある。

図表 1 は *DrillMaster* 用の教材データの例であるが、私の担当科目 Pronunciation II で取り上げている音の同化 assimilation による変化を聞き取るディクテーションの練習教材である。

この教材では、ファンクション・キーを押して音声を聞き、聞こえた表現をタイプすることを学習者に求めるのだが、アットマーク @ の間の英語が想定している正答である。しかし、当然のことながら going to の同化による変化した音である gonna が聞こえてきた場合はそのように入力する可能性も高く、本来話し言葉ではあるが、これをも正答として

```

$$Sound_g_01.wav
1. 彼を訪ねるつもりですか?
  (@ Are you going to visit him @)

$$Sound_g_02.wav
2. でもうまくゆかないわ。
  (@ But it's not going to work @)

$$Sound_g_03.wav
3. 頼み込ませような真似はしないわ。
  (@ I'm not going to beg @)

$$Sound_g_04.wav
4. (私たちが) 二人きりで話をさせてあげるわね。
  (@ We're going to let you two talk @)

$$Sound_g_05.wav
5. 君はそれを気に入らないだろう。
  (@ You're not going to like it @)

**01
Are you gonna visit him
**02
But it's not gonna work
But it is not going to work
But it is not gonna work
**03
I am not going to beg
**04
We are going to let you two talk
We're gonna let you two talk
We are gonna let you two talk
**05

```

図表 1

扱う必要があるために、それぞれの問題の出力順にアスタリスク (*) 2つに番号を付して他に考えられる別解を設定している。さらに、会話文でのディクテーションの場合、この他に、短縮形 (I am と I'm など) の取り扱いによる複数の正答に対応することは不可欠である。正答を1つしか想定しない CALL システムではこのような問題自体を事実上避けるしかないが、この例のような場合以外でも選択式ではない練習問題で正答が1つしかないように作問することは実はかなり大変で、複数正答対応のできないシステムでは教材を作成する可能性自体が大きく制限されることになり、教材作成の生産性が著しく低下する。

ところで前節末尾で触れたことだが、ここでの例のように、一文全て

を入力させるように作問した場合に入力ミス避ける目的で文末にピリオドなどの文字を表示するように教材を作成すると、意外と学習者自身がそれを見逃してしまうとしたらどうだろう。この教材例ではこれを不注意として誤りとするようになっているが、理想的には正答として処理できることが望ましいように筆者自身も思う。そして、複数正答をサポートしている場合はピリオドのついたものとつかないものを正答設定すればよいのだからそれ自体は可能である。しかも、コピペしたものにピリオドをつければ良いのだから作業量も大したことはない。しかし、想定される正答が2つの場合はそれぞれにピリオドのついたものを正答として加えることになるが、データの量は倍になる。ここで示した教材例では複数正答の数が4つのものがあるがその場合はそれぞれにピリオドをつける場合とそうでない場合を加えて複数正当数は全部で8つとなる。ここに大きなジレンマが生じることになる。

ここでのジレンマとは設問1つについてのデータ量がかなり多くなることに伴う、正誤訂正がその1つである。どんな問題も特に問題数が多くなればなるほど入力ミスなどによる訂正の必要が後から生じてくる。今見たように、ピリオドに対応する作業量自体はその時は大したことはないのだが、設問数が増える中でその中で一カ所でもスペルミスなどによる訂正の必要が生じると、複数正答の数（別解の数）だけの訂正の必要が生じ、ワープロの編集機能の語句置換だけで対応でききないことも多く、その場合は個々の別解で確認しなくてはならないため作業量はかなりのものになる。

では、この部分を2-1で示したようなフィルターによって処理できないか、コンピュータの得意とするプログラムによって対応することができないか、ということになる。例えば正答にピリオドがついてなくても、学習者の入力にピリオドがついていれば、このピリオド（感嘆符、クエスチョンマーク）を削除して正答と照合するようにプログラムすればよ

い、ということになる。しかし、実際にピリオド（感嘆符、クエスチョンマーク）が全く必要のないというように想定してしまってよいだろうか。やはりそうはいかないだろう。状況によっては必要になる練習問題の可能性はゼロとは言えない。だとするとやはり手間はかかるが複数正答の中にピリオド（感嘆符、クエスチョンマーク）を含んだものを必要に応じて加えるしかないのだろうか。実は、これに対処するもう一つの方法がある。それは、設問のどこかにマーカーを付して、マーカーがあるときは学習者の入力にピリオド（感嘆符、クエスチョンマーク）があっても無視するようにフィルターをプログラムするというものである。おそらくこれがこの場合一番合理的な解決法であるが、ここにもジレンマがないわけではない。それは、次章で述べるように、ドリル型のCALL教材は、CALL教材を作り慣れていない英語教師がぱっと見て、わかりやすいということが極めて重要であるが、ここでのマーカーによる処理は、その点で複雑化する要素となる。この例のように、あることに配慮するとその分だけデータ、及び教材自体の構造が複雑になるということに常に留意しないと良い機能を用いたために却って使いにくくなるということになりかねない。

現在のところ筆者は、自分以外の教員にとっても分かりやすいシステムということで、少しでも単純化するという観点で考え、このようなマーカーによる方法も複数正答による方法も用いてはいない。また、これまでのところ学生からのその点についての指摘はない。とは言え、学習者の反応データを収集して対応が必要か判断したいと考えている。

いずれにせよ、ここでも、本質的には人間であれば無意識に処理可能なものをコンピュータで処理するが故に、生じる問題として整理しておくべきものと思う。

2-3 学習履歴情報処理

ほぼどんなシステムでも学習者が CALL 教材を用いて学習した場合には、どの問題にどれほどの時間をかけて取組み、その達成率はどのくらいかなどのデータが記録として残るようになっている。そのフォーマットは様々だが、例えば *DrillMaster* では次の図表 2 のようになる。



図表 2

これは、学習者自らが自分の学習状況を確認するためのものだが、「学習モード学習状況」では、選択した教材（この例では「DM_Listening_句動詞熟語_03」）の総学習時間、正答率などが表示され、おおよその程度学習したかがわかるようになっている。また、「テストモード」が選択できそこでは、一通り問題に取り組んで全問題に解答し終えない限り、途中で「学習モード」へ行って練習できないようになっており、ある一時点での達成度をしっかり計測することを意図している。この「テストモード学習状況」は、同じ問題に取り組んでいる場合は最大過去 10 回分までのデータ（実施日、実施時、得点、総学習時間＝テストにかかった時間）が残るようになっている。つまり、1 週間前と現在の「テストモー

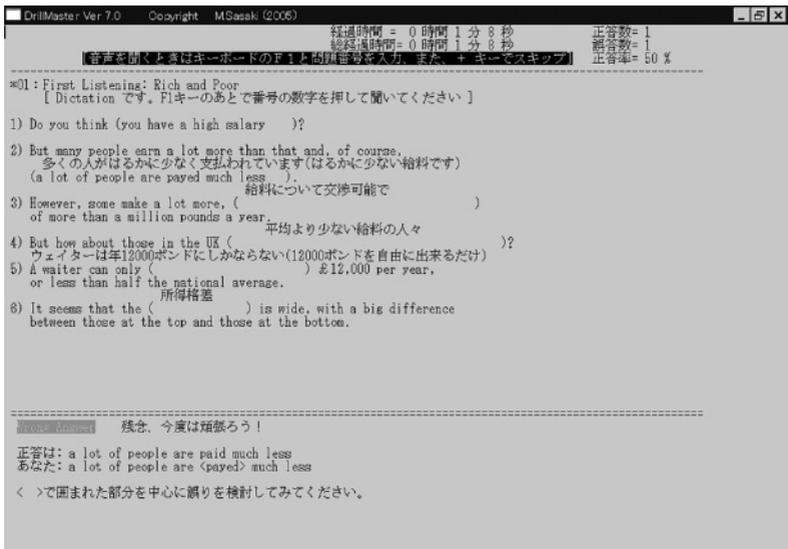
ド」の成績が残せるため、自分で定着の変化を確認できる。

他方、教師の方で指導上あるいは評価をする上で、個々の学習者がどの教材をどれほど学習しているかを把握する学習履歴も当然必要である。*DrillMaster* の場合は、学籍順に学習者が取り組んだ全ての教材の「テストモード学習状況」の一番最後の履歴をエクセルのデータ（正確には csv ファイル形式による）に出力し、例えば、それぞれの教材で 8 割以上達成しているか否かがすぐ分かるようになっている。それによって筆者自身は「8 割以上達成している場合に平常点に 10 点分加算」というようにそのデータを使用している。

さて以上見たように、学習者用の履歴と教師用の履歴の両方の例を示したが、ここでも留意すべき点がある。1 つは、どの情報をどこまで表示するか、あるいはデータ化するかということを決めることは、意外に厄介な問題である。「学習者の自主性に任せ選択してもらう」として、あれもこれもということになると学習者は履歴情報のなかから自分に必要な情報を見つけだすのに手間がかかりすぎて、却って利用しなくなる。また、設計上どの情報をどの位置に表示するかと言うこと自体がここで「見つけ出しやすさ」に関わってくるわけで、安易にどんな情報でも、という訳にはいかない。目的をしっかりと絞って表示情報も絞る必要がある。筆者の場合は、あまり特殊な目的を設定しないで最小限にとどめることにしている。もう 1 つは、教材作成者である教師の負担を増やさないようにすべきである。より客観的な情報になるようにということで「テストモード」で各設問に配点をつけるようになっているシステムをかつて見たことがあるが、必ずこれを入力しなくてはならないようだとして 1 つ 1 つの問題に軽重の判断をしなくてはならず、大きな負担となる。せめてこの場合、そのようなデータは特に必要だと判断した時だけ入力すればよいように設計されるべきである。

2-4 簡易誤答分析表示

学習者の入力した解答の正誤は、用意された正答及び別解と照合されて判断されるが、誤答の場合どこが間違っているのかをある程度指摘できるなら学習者にとっては、当該問題のどこができなかったのかを把握することが出来る。*DrillMaster* では、誤答に対して図表3のように表示されるようになっている。



図表 3

この例では、正答に対して、学習者の解答では paid のスペルミスがあるためその点を指摘している。ここでは、正答と解答の照合の結果についてただ正答か誤答かを表示するだけでなく、正答と比べてどの部分が違うのかを機械的な処理を行っているだけである。誤答分析ということと典型的な誤答例をデータ・ベース化してそれぞれの誤答にふさわしい指摘のメッセージが表示することを想像するかも知れないが、ドリルタイ

プの CALL 教材では必ずしもその必要はない。何故なら、このタイプの教材ではあくまでも基本的な考え方がわかったうえでの定着のための練習であるので、文法上の細かな事柄についてのミスなどの指摘を前提にしているわけではないからである。極めて機械的だが、比較的長いフレーズや文の入力においてどの部分に間違いがあるか、あるいは想定の手答と比べて何語不足しているか、あるいは何語多いか、程度の情報によって学習者が素早く自分の誤りを確認して次の問題に取り組むことに意義がある。また、教師にとっても、1問1問に詳細な誤答分析データを付するという事は事実上不可能でありこれまた教材の生産性を著しく落とすことになる。

この点に関しては、既にワープロなどでは入力した語のスペルチェックやパンクチュエーションだけではなく、最新の AI を駆使して文法上の分析も行っており、疑わしい表現に波線のアンダーラインを表示する機能があるが、将来的にこのような機能の活用も考えられなくはないが、ここでの目的からするならば、却ってこのような機能は不要ではないかと思う。

2-5 その他の諸機能

CALL 教材の機能と言うことでは、そのほか色々あり得るが、ここでは以下の4つの機能について簡単にまとめておきたい。

1) カラー文字表示

学習上特に押さえておいてほしいことを異なった色で表示するという事はよくある。筆者の場合は、学習者に記憶してほしい句動詞などのフレーズ表現の練習問題などで日本語で全文の意味を表示する場合の当該フレーズ部分の意味を濃い青を用いて表示している。あまり多くの色を用いないことが重要だ。かつて CALL 教材の初期の頃には様々な色を

用いすぎて、却って何を言おうとしているのか分からなくなるようなものもあったが、最近ではさすがにそういう例を見ることはない。

2) スキップ機能

「学習モード」では、同じ問題を繰り返し練習するために、同じ画面にある既に学習者にとっては分かっている問題にも一々解答入力しなければならないということが起こりうる。学習者からするとそれは明らかに時間の無駄で、そう言った問題は、スキップによって回避することが望ましい。従って、画面の選択した問題だけ解答出来るようになっていればよいわけだが、そのためマウスで選択するようになっているものもあるようだ。しかし、キーボードでセンテンスやフレーズを入力している流れの中で、マウスに持ち替えて選択するというのは意外に学習のリズムが崩れるのではないかと思う。*DrillMaster* では、「+キー」を押すことで画面に正答が表示されて次の問題に進むことが出来るようにしてある。また、語彙など全く思い出せないこともあり得ないことなく、その場合何かいい加減な答えを入力するよりもこのスキップ機能で答えを確認して次に進む方が、時間的にも合理的だと言える。ただし、安易にこの機能を使うと却って逆効果になる場合もあるので、その点での指導は必要である。

3) ヒント出力

これは前節の誤答分析にも述べたことだが、この機能も教材作成段階で教師がヒント・データの作成を強制されないことが重要である。既に授業で取り上げた事項の復習と定着を主要な目的とするわけで、ヒントが絶対に必要とは限らないのだから。

特にドリルタイプの CALL 教材では、この部分が詳しすぎると却って学習者の効率を下げることになる。それよりも、「この問題のポイントは

○○○なのだが、それがはっきりしない場合は、先生に質問するか、テキストの△△ページの説明を参考にすること」などのメッセージを表示し、これをメモさせて次の問題に進むようにさせる方が効率的である。

4) 励ましメッセージ

DrillMaster では、正答が続く場合にはそれを評価して励ますメッセージを表示するようにしている。具体的には、最初の正答には、「You made it!」、続く問題にも正答すると「その調子!」、更に正答が継続する毎に「快調!」、「なかなかやるな!」、「絶好調!」と表示されていく。ここでは機械的に正答が継続した場合に表示されるようにしているのであって、同じ問題を2回、3回と繰り返して取り組んだ場合に前回の結果をデータとして保存しておいて、それに対応して表示するというものではない。そのようなことは可能なのだが、ここでの「励まし」という目的からするとあまり意味がないと思われる。

従って、この機能は必ずしも必要だということではないが、学習者のモチベーションとしては、正答が継続した場合にそれぞれ異なった評価メッセージを表示されることで、単純な正誤表示よりは幾分か前向きな姿勢が期待できる。

3 ドリル型 CALL 教材及びその作成システムのあり方についての若干の考察

1-3 において教材作成システムとしてのオーサリング・システムについて述べたが、*DrillMaster* では、それにあたるものはワープロ・ソフトを含めたいわゆるエディターである。ドリル型の CALL 教材においてオーサリング・システムはできるだけシンプルなものが望ましいと述べたが、それによって教材(教材データと制御用データ)を作ることが出来れば良いわけであるからそれが可能であるなら何を用いても良いわけ

ドリル型 CALL ソフト活用のあり方と備えるべき諸機能について

である。そこでワープロで作成したものが次の図表 4 であるが、これによって教材そのものを作るという観点からそのあり方を考えてみたい。

```

*01 : 歯草・摩擦音
^1=====^0
次の単語の^1 占色^0 の音を英語の専門用語で定義しなさい。
      有声・無声の区別      調音点・調音法
      ↓                      ↓
voi^1ce^0d : (@ voiceless @) で alveolar fricative
ri^1se^0   : (@ voiced @)   で alveolar fricative

*02 : /s/ の影響を受ける you
^1=====^0
      ^1 [ Dictation です。F1 キーのあとで番号の数字を押して聞いてください ] ^0
$$$Sound_syu_01.wav
1. 忙しなくては私を手伝ってくれ。
   (@ Give me a hand unless you're busy @).

$$$Sound_syu_02.wav
2. あなたは彼を忘れてしまったのだと思う。
   (@ I guess you forgot him @).

$$$Sound_syu_03.wav
3. 列車に乗り遅れるぞ。
   (@ You'll miss your train @).

**01
give me a hand unless you're busy
Give me a hand unless you are busy
give me a hand unless you are busy
**02

**03
you'll miss your train
You will miss your train
you will miss your train
**04

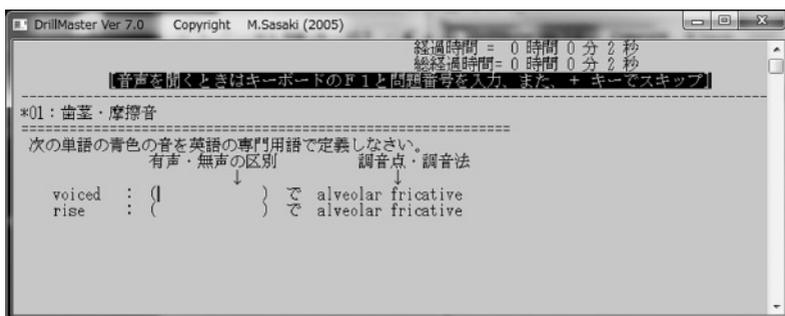
#01
/s/の音とj/(ニ)の音がつながって互いに影響し合い「シ」のように聞こえることがあります。
#02
/s/の音とj/(ニ)の音がつながって互いに影響し合い「シ」のように聞こえることがあります。
#03
/s/の音とj/(ニ)の音がつながって互いに影響し合い「シ」のように聞こえることがあります。
#04

*03
    
```

図表 4

この例で、アスタリスク* を1つ用いた *01、*02、*03 は言えば、プリント教材の設問 1、設問 2、などに当たる。例えば、*01 と *02 の間の教

材データをエグゼキューターで作動させた実際の画面は次図表5のようになっている。

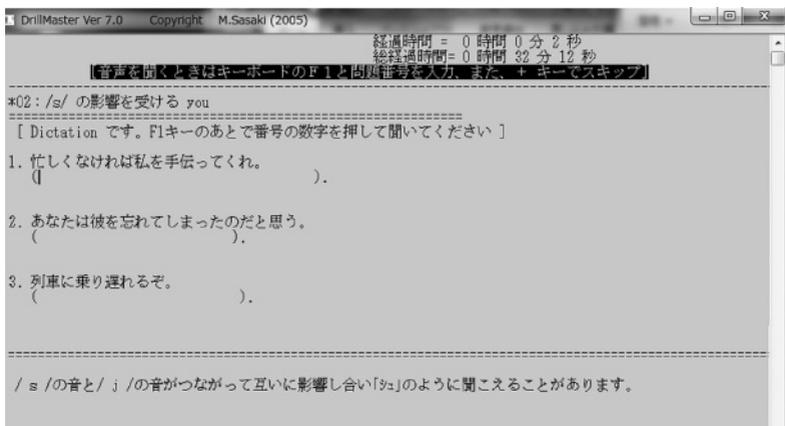


図表 5

既に2-2で述べたように、アットマーク@で囲まれた部分が空白問題となっている。またそこでは述べなかったが、^1や^0が色指定でこの画面では青色に文字色が変わるようになっている。非常に単純化して言えば、異なった色を用いないなら、設問の区切りをアスタリスク*1つの数字で行うことと、問題にして解答させる部分をアットマーク@で囲むようにすれば、この手の空白問題はすぐ作れることになる。

設問2にあたる*02と*03の間の部分は、2-2で図表1で示したものと同じである。アスタリスク*が2つの数字の場合は複数正答(別解)であることは既に述べた。そこでは述べなかったが、\$のあとの「Sound_syu_01.wav」は音声のファイル名を示すもので次の図表6のように学習画面には表示されなくて、F1キーと問題の番号を押すと当該の音声流れるようになっている。また、#のあとの数字と数字の間の文はヒントで、F5キーを押すと画面の下に表示されるようになっている。

ドリル型CALL教材の練習問題そのものについて考えた場合、1-1で述べたドリルの2つの目的である、習得状況の確認と習得内容の定着量的拡充という点をはっきり分けて考える必要がある。特に、後者の習



図表 6

得内容の定着と言う目的では、記号選択式の問題は避けるべきものと思う。初めて取り組む時は、問題の趣旨を理解しないと解答しようがないが、2回目、3回目となると記号だけが記憶に残っていて機械的に反応することも多く、定着のための練習にならない。この例では、キーボードからスペル入力をするようにしているが、選択式の場合もスペルを入力するにすることで記号入力に比べて練習になると言える。

ドリル型 CALL 教材は「紙（プリント）による練習問題を単純にコンピュータソフトに置き換えることができる」というように考えるわけにいかないことを1-3で述べたが、それはエギューゼキューターの設計上のことで、その作動において2章で述べた技術的な諸機能の問題が解決していれば、実は紙（プリント）によるものとほとんど同じ発想で処理できるようにしたのがこの例である。教材の作成ソフトであるオーサリング・システムや実行ソフトであるエギューゼキューターの制作者に設計を依頼する場合1-3で述べたことが極めて重要になるが、その結果としてプリント教材感覚で使用できるということは逆に有益だと言える。

ここで示した例では、複数正答（別解）の問題を除いてドリル型 CALL

教材を作成する上で、プリント教材をワープロで作成したものをほぼそのまま転用出来るようにしてある。そしてここで重要なのが、カラー文字や音声、ヒントなどの活用を強制されないばかりか（かつてはそういうものがあつた）、それらを使用したいと思ったとき簡単に利用できるシステムであることである。1-3でも述べたように、その使用にシステム上覚えなくてはならないことが色々あるようだと思ふ気にならない。筆者自身、厚手のマニュアルを見せられて、これを読んで覚える時間よりも不十分な面はまだ色々あるが自分の開発した *DrillMaster* で1つでも多く教材を作った方が効率的だと思つたことがある。

ドリル型 CALL 教材ということでもう一つ重要なのは、全体が見渡せることである。実際場面では、作成した教材の修正が容易に出来るかどうかはかなり決定的なのだが、その時、例えば複数正答、カラー文字、音声、ヒントなどの情報がばらばらでアクセスに手間がかかると、その訂正も困難になる。できるだけ一覧で全体が見通せて、しかも修正機能へすぐアクセスできることが教材を作成する生産性に大きく影響する。

最後にメンテナンスに関わることだが、他のシステムで同じ教材データが利用できる可能性が保障されているか、あるいは、その可能性が高いことも極めて重要である。最近では Windows XP のサポートが終わってタッチパネル操作を前提とした新たな流れとなっているが、OS の変更やバージョンアップによって既存のソフトが動かなくなって、場合によってはせっかく作成したソフトが無意味に近い状態になることがある。同じ内容の教材を全く新たなシステムで作作り直さなければならぬとすると大変な損失を意味する。筆者が知る限りではかつての MS-DOS 時代の教材の多くがそうなつてしまつてゐるのではないだろうか。その意味で教材データのデータ構造が単純で基本的には契約ユーザーに公開されている必要がある。そうでなければ、A社のシステムでせっかく作成した教材が仮にその会社がそのサポートをやめた場合、OS が変

わずらに既存のものが使えているうちはよいが、古くなって現在使用しているエグゼキューターが使えなくなった場合、使い物にならなくなる。そんなとき、教材データの構造がわかっていたら別のソフトで動くように新たなエグゼキューターを別のB社が作ることも出来る。それだけではなく、インターネットのブラウザなどの例で見られるようにコンピュータのOSや機種が異なっても作成した教材データで使用可能なシステムの開発が可能となる。

いずれにせよ、学習者にも教材作成者にも優しいシステムがなければ継続して使われないことになる。

4 まとめ

1-2において、ドリル自体は英語教育に不可欠であるとしてもそれはあくまでも手段であって、それを英語教育そのものとして自己目的化することは本末転倒で、かつてのパターン・プラクティスの失敗はその点にあったと述べた。そして2と3において、そのことを前提として、ドリルを効果的にする更に1つの有力な手段として、ドリル型のCALL教材について、それが効果を発揮するために技術的諸機能がどうあるべきかについて整理してきた。そこで重視したことは、学習者に負担をかけない、ストレスを与えないということと、教材（データ）作成者である教師にとっても負担のないシステムであるべきだ、ということだった。特に、後者については教師がCALL教材を作るという前提に立っているが、その理由は既に1-2で述べたように、コミュニケーション能力を育成する上でその時々で使用する教材に対応したドリル教材を丁度プリント教材でそうするようにその都度作って学習者に定着のための練習を促すためだった。勿論、将来的に教科書会社が発行するテキストに対応する形でこのようなドリル型CALL教材が提供されることを期待したいし、事実、筆者の目からはかなり不完全だがそれを意図したと思われる

テキストも一部で販売されている。とは言え、当面は現場の教師が教材を作成しその効果を示すという段階なのだと思う。

しかし、そうあるためにはドリル型 CALL 教材システムとして備えておくべき機能について技術的視点から整理し、どんな機能がこのシステムにとって必要かを技術開発者に要望することが不可欠である。それは、学習者の入力解答の問題を人間である教師であれば簡単に判断できるのに対して機械である CALL 教材ではそうはいかない、ということに基づく。このことをより踏み込んで言うと、複数正答（別解）への対応に明らかのように、人間なら無意識に処理していることを、別解それぞれを意識的に全てデータ化しそれぞれに対応するか、あるいは 2-1 で示したフィルターによる方法のようにデータ化の一部をフィルターとなるプログラムを書くことで処理することになるためである。それは人間の無意識の判断も含めた質的なものをデータという量的なものに換算することで機械が人間の代わりに処理することが可能になり、これを逆に見れば、このようなコンピュータという「量」によるシステムが人間による「質」を代替しているものと言える。つまり、人間教師の一部の機能（あくまで一部の機能）を機械で代替しようとするものだが、このことに関わって英語教師とコンピュータ技術の専門家では必ずしも認識が一致していないことがあると思う。

確かにコンピュータの技術的なことは、そもそも通常は英語教師が関与しない領域であり、コンピュータの専門家にそのニーズを伝えれば、その通りできあがるものとも考えるかも知れない。他方、コンピュータの専門家はニーズがわかればその通りシステムを組んで対応出来ると考えて開発するが、意外に期待通りの結果を生まないで、お互いに、一方は痒いところがどこかわかってもらえない、他方は痒いところがどこなのかの説明がよくわからないので対応できない、ということになる場合が多いように思う。そうすると、それぞれが相手が責任を持つべき領域だ

と考えるとデッドロックは固定化されてしまう。筆者自身は Sunday programmer として *DrillMaster* を開発し、ある意味教師と技術者の両方の視点から見てきたことになるのだろうが、今日のように OS のバージョンアップが次々と起こりソフトもハードも短時間に発展し少し前のものが陳腐化するなかでは、予算が許すならソフト開発の専門家と協働する方がはるかに合理的だと考えている。その意味で、おそらくここで述べてきた技術的なことのほとんどは、学習システムを開発するシステム・エンジニアやプログラミングの専門家に直接読んでもらったほうが、とりあえずは話が早いのだと思うが、実際には英語教師がある程度それを理解して細部について要望していかないと期待通りの結果とはならないと思われる。

今後も様々な CALL 用システムが開発されるかも知れないが、本稿で示したようなドリル型 CALL 教材を作成して活用するうえでの技術的諸機能を備えることで、より英語教育の現場の要請に真に応えるものとなっていくことを期待したい。

付録 2章で示した諸機能を実行するプログラムのソース例

一部だが、オーサリング又はエグゼキューターのシステム開発について、専門の技術者に説明する上での参考になるかも知れないので、筆者が作成したソース・プログラムのいくつかを挙げる。なお、使用言語はBASIC (Beginners' All-purpose Symbolic Instruction Code) であるが、アルゴリズムを考える上での参考にはなると思う。

```
[1] 2-1 スペース処理関連
sub SPACECTR (MOJ$, SD$)
  MOJ$ 入力文字列
  SD$  スペース処理後の文字列

  J% = 1
  ST2$ = mid$(MOJ$, J%, 1) : セリズ / J パン / モ'
  while (ST2$ = "")
    J% = J% + 1
    ST2$ = mid$(MOJ$, J%, 1)
  wend: 'セ' ン / セリズ / パン / セリズ / パン / セリズ
  SD$ = "" : ST$ = ""
  MOJ% = len(MOJ$)
  for I% = J% to MOJ%
    ST$ = mid$(MOJ$, I%, 1)
    if ST$ <> "" then SD$ = SD$ + ST$ : FLG = 0
    if ST$ = "" then if FLG = 0 then FLG = 1 : SD$ = SD$ + " "
  next I%

  '文字列終端処理
  LS = len(SD$) : if LS <= 0 then LS = 1
  if right$(SD$, 1) = "" then SD$ = mid$(SD$, 1, LS - 1) : LS = len(SD$)
  if right$(SD$, 2) = "?" then SD$ = mid$(SD$, 1, LS - 2) + "?"
  if right$(SD$, 2) = "." then SD$ = mid$(SD$, 1, LS - 2) + "."
  if right$(SD$, 2) = "!" then SD$ = mid$(SD$, 1, LS - 2) + "!"
  if right$(SD$, 2) = " " then SD$ = mid$(SD$, 1, LS - 2) + " "
  if right$(SD$, 2) = " " then SD$ = mid$(SD$, 1, LS - 2) + " "
  if right$(SD$, 2) = " " then SD$ = mid$(SD$, 1, LS - 2) + " "
end sub
```

[2] 2-2 複数正答(別解)対応、2-4 簡易誤答分析表示関連 ②正答との照合

```
'人力解答が正答と同じ場合、点数表示
if S1$=QUESANS$(Q) then
  '点数とメッセージ
  POINTDISPLAY "!", EXAMCORRECT%, EXAMINERROR%, EXAMOUTCORRECT%, EXAMOUTERROR%
  EXAMCORRECT% = EXAMOUTCORRECT%
  POINTMESSAGE "!", EXAMQFB%, EXAMQFN%
  EXAMQFB% = EXAMQFN%
end if
```

ドリル型 CALL ソフト活用のあり方と備えるべき諸機能について

```

'入力解答が正答と異なる場合、別解チェックへ
if S1$<<QUESANS$(Q) then
  ALTNUMBER = Q : ALTFLAG = 0
  for K = K1 to K2
    '該当の有無確認(有無とそれに続く2桁の数字を検索しALTNUMBERと比較)
    if left$(EXAMMATERIAL(K), 2) = "*" and val(mid$(EXAMMATERIAL(K),3,2)) = ALTNUMBER then
      ALTSTART = K + 1 '別解開始行
      ALTFLAG = 1

      '別解の「終了行」を得るための指標
      ALTNEXTNO$ = str$(ALTNUMBER + 1)
      ALTNEXTNO$ = right$("0" + right$(ALTNEXTNO$, len(ALTNEXTNO$) - 1), 2)
      for J = K1 to K2
        if EXAMMATERIAL(J) = "*" + ALTNEXTNO$ then
          ALTEND = J - 1 '別解終了行
          end if
        next J
      end if
    next K

  if ALTFLAG = 1 then
    for K = ALTSTART to ALTEND
      SPACECTR EXAMMATERIAL(K), EES '別解のスペース処理
      EXAMMATERIAL(K) = EES
      if S1$ = EXAMMATERIAL(K) then
        '点数とメッセージ
        POINTDISPLAY "I, EXAMCORRECT%, EXAMINERROR%, EXAMOUTCORRECT%, EXAMOUTERROR%
          EXAMCORRECT% = EXAMOUTCORRECT%
          POINTMESSAGE "I, EXAMQFB%, EXAMQFN%
          EXAMQFB% = EXAMQFN%
          S1$ = QUESANS$(Q) '次の入力解答が正答と異なる場合)のための補正
        end if
      next K
    end if
  end if

  '入力解答が正答と異なる場合、正解表示と点数表示
  if S1$<<QUESANS$(Q) then
    '点数とメッセージ
    POINTDISPLAY "O, EXAMCORRECT%, EXAMINERROR%, EXAMOUTCORRECT%, EXAMOUTERROR%
      EXAMINERROR% = EXAMOUTERROR%
      POINTMESSAGE "O, EXAMQFB%, EXAMQFN%
      EXAMQFB% = EXAMQFN%

    'locate 2, 32: color 5: print "正答は:"; QUESANS$(Q); color 7
    '背景色が黒の時
    locate 2, 32: color 1: print "正答は:"; QUESANS$(Q); color 0
    '背景色が灰色の時

    '----- テストモードでの解答状況記録へ誤答マーカー "*" をつける
    INFO(TESTALLFRAMEQUESNUM) = "*" + INFO(TESTALLFRAMEQUESNUM) + "*" + QUESANS$(Q)
    '-----

    '入力解答が正答と異なる場合、正答のオリジナルと比較したヒントメッセージ
    TYPED0$ = S1$
    MODEL0$ = ORIGINALQANS$

    ANSWERCHECK TYPED0$, MODEL0$, TYPED00, MODEL00, CORRECT0$, MSG01, MSG02

    locate 2, 33: print "あなた:"; CORRECT0$
    if instr(CORRECT0$, "<") > 0 then
      locate 2, 35: color 1: print "<" >で閉まれた部分を中心に誤りを検討してみてください。"
    endif

    locate 2, 36: print MSG01
    'locate 2, 37: print MSG02: color 7
    '背景色が黒の時
    locate 2, 37: print MSG02: color 0
    '背景色が灰色の時

    CORRECT0$ = "" : ORIGINALQANS$ = "" : MSG01 = "" : MSG02 = ""

  end if

```

[3] 2-5 3) ヒント出力関連

```

'2-006] ヒント出力 -----
*HINT

'ヒント・データの初期位置確定
'var HINTNUMBER as integer
'var HINTNEXTNO$
'var HINTSTART 'MATERIAL(K)におけるヒント画面の開始行
'var HINTEND 'MATERIAL(K)におけるヒント画面の開始行

'HINTNUMBER がその画面の問題の通し番号 Q によって与えられる
HINTNUMBER=Q

HINTFLAG=0
for K=K1 to K2

    '該当のヒントの有無確認#の有無とそれに続く2桁の数字を検索し HINTNUMBER と比較
    if left$(MATERIAL(K), 1) = "#" and val(mid$(MATERIAL(K), 2, 2))=HINTNUMBER then
        HINTSTART = K + 1 'ヒント開始行
        HINTFLAG = 1

        'ヒントの終了行を得るための指標
        HINTNEXTNO$ = str$(HINTNUMBER + 1)
        HINTNEXTNO$ = right$("0" + right$(HINTNEXTNO$, len(HINTNEXTNO$) - 1), 2)
        for J=K1 to K2
            if MATERIAL(J) = "#" + HINTNEXTNO$ then
                HINTEND = J - 1 'ヒント終了行
            end if
        next J
    end if
next K

'ヒント出力
if HINTFLAG=1 then
    '30行～38行までの画面データをクリアする。
    for A=1 to 9
        locate 0, A+29 : print spc(100)
    next A

    X%=2 : Y%=31 '左の座標からヒントを表示開始
    for K=HINTSTART to HINTEND
        HINT$ = MATERIAL(K)
        DISPLAY HINT$, X%, Y%
        Y% = Y% + 1
    next K
end if

'ヒントの有無の表示
if HINTFLAG=0 then
    '30行～38行までの画面データをクリアする。
    for A=1 to 9
        locate 0, A+29 : print spc(100)
    next A

    locate 5, 28 : print "No hint.": beep
end if

return: 'HINT END

```

```
[4] 2-5 4) 勉強しメッセージ関連
'3-002] 正誤メッセージ表示 -----
sub POINTMESSAGE (POINTFLAG$, QFBFORE%, QFNOW%)

    '30行~38行までの画面データをクリアする。
    for A=1 to 9
        locate 0, A+29: print spc(100)
    next A

    locate 2, 30

    if POINTFLAG$ = "1" then
        color 14: print "Right Answer"  "'正解 !"
        QFNOW% = QFBFORE% + 1
        if QFNOW% > 5 then QFNOW% = 5
    end if

    if POINTFLAG$ = "0" then
        beep
        color 10: print "Wrong Answer"  "'誤り !"
        QFNOW% = 0
    end if

    color 3: locate 18, 30
    if QFNOW% = 1 then print "Y o u   m a d e   i t ! "  'QP=1-5: 連続して正答している場合
    if QFNOW% = 2 then print "そ の 調 子 !"
    if QFNOW% = 3 then print "快  楽 !"
    if QFNOW% = 4 then print "な か な か や る な ! "
    if QFNOW% = 5 then print "絶  好 調 ! "
    if QFNOW% = 0 then print "残念、今度は頑張ろう ! "

    'color 7
    '背景色が黒の時
    color 0
    '背景色が黒の時

end sub
```

注

- 1) 現在では「コミュニケーションのための英語」というのが当たり前になっていて、そういう発想が英語教育にないことは、元々間違っていたかの認識が一般的ではないかと思うが、このような発想自体が「時代の子」なのだと思われる。「平泉・渡部論争」として知られる英語の実用性をめぐる論争が示すように、今日のような理解が必ずしも一般的ではなかった時代もある。ICT によって英語の実用性が我々の日常においても現実化しているという技術的な条件だけではなく、1985年の「プラザ合意」以降の円高の条件のもと日本の輸出企業が生産拠点を海外に移したことによるグローバル化の加速が更に大きな背景にあると言える。
- 2) ここでは「コンピュータの活用」という表現をとっているが、現実には、スマホを始めタブレットなど様々なハードが ICT を構成するとともに、記憶媒体としてのハードディスクなどもネット環境を前提にクラウド化

しており、単純に「コンピュータ」というだけでは活用可能な ICT 技術を把握し切れていない印象を持たれるかも知れない。しかし、これら新たなハード及び記憶媒体にかかわる変化も、英語教育においてドリル型の CALL を論ずる限りでは、「コンピュータの活用」という表現で統一してもその本質、あるいは本質的機能を変えるものではないため、以下そのような概念として「コンピュータ」を用いる。

- 3) 中森誉之 (2013) p.116
- 4) Sasaki, M (1992) pp.235-236
- 5) 平嶋里珂 (2007) p.84
- 6) Canale と Swain がコミュニケーション能力の要素の 1 つとして「文法的能力 communicative competence」を挙げていることは、よく知られている。
- 7) 筆者としては言語の社会的側面に注目して M. A. Halliday の機能文法に示された方法の活用を考えている。
- 8) 「パターンプラクティスに使用される例文は現実の文脈から離れて形式の習得に終始することが多く、身についた文法能力を現実のコミュニケーションに転化させるのは容易ではなかった。」平嶋里珂 (2007) p.81
- 9) Listening II のほんの一部の範囲で平常点の中に、テストモードの成績を 10 点分加えた。

参考文献

- Canale, M. and M. Swain (1980). "Theoretical Bases of Communicative Approaches to Second Language Teaching and Testing" *Applied Linguistics 1*: 1-47.
- Halliday, M. A. K. (1994) *An Introduction to Functional Grammar*.: Arnold
- 平泉渉・渡部昇一 (1975) 『英語教育大論争』文藝春秋
- 平嶋里珂(2007) 「コミュニケーション能力を養成するためのパターンプラクティス (齋藤榮二教授退職記念号)」『関西大学外国語教育研究 13』 pp.

79-95

中森誉之 (2013) 『外国語はどこに記憶されるのか』 開拓社

Sasaki, M (1992) “The Significance of CALL Materials — Examining
Three Main Types of Courseware” 『筑波英語教育学会』 第 13 号