

スクリプトタイプのプログラマレス CAI構築ツール

飯箸泰宏 西澤真一 相澤孝志
株式会社サイエンスハウス システム開発部

我々は、十数年にわたって、教育用のソフトウェアを開発してきた。1989年、DOSシステムの上で、イベントドリブンシステムの構築を容易にする「イベントドライバー (EVD)」を開発し教育用ソフトウェアの質的向上と生産性の向上の第一歩を築いた。1993年には、DOSシステムの上にウインドウライクな環境を実現する「パネルシステム」を開発し、質の高いGUIを均質かつ高い生産性で実現できるようにした。

今回、発表する「スクリプトタイプのプログラマレスCAI構築ツール:スクリプトウェア」は、これらの経験の上に作られたものである。「スクリプトウェア」は、教育用エージェントシステムのプロトタイプとなるもので、これを利用することによって、C言語などのプログラム言語を知らない教師や教材編集者が容易に高度な教育ソフトを独力で構築することができることに大きな特徴がある。実際、ある意味でこれは「プログラマキラー」となる。その威力は原理的にプログラム言語で実現できるすべての機能に及ぶことができるので、従来型の「オーサリングシステム」とはまったく次元を異にするものである。

Programmerless CAI Construction Tool with Script-style

Yasuhiro Ihashi, Shinichi Nishizawa, Takashi Aizawa
Dev. of System Development, Science House Co. Ltd.

We have been developing a educational software for ten years. In 1989, we developed the event-driver(EVD) that is easy to construct a event driven system on the MS-DOS system, building the first step for improving quality and productivity of educational software.

Furthermore, in 1993, we developed the panel system that realizes a window-like environment on the DOS system. As a result, it has become possible to product a high quality GUI with high productivity and homogeneity.

This paper describes the programmerless CAI construction tool with script-type (script-ware) which is based on our experiences. The script-ware is a prototype of educational agent system. The major feature of this system is that teachers or teaching materials editors can construct a high quality educational software easily by themselves even if they don't any program language.

Actually, this means "programmer killer". This system is different from the conventional authoring system in that it is possible to cover all function described with program language.

1 はじめに

長年教育ソフトウェアの開発に携わっている中で、教育現場や教材編者との交流も深まった。とくに市町村教育委員会との共同開発では、教育現場の深い悩みと喜びに心揺り動かされる事が多い。

現場によりよく接すれば、流通しているソフトウェアの質的限界にほとんど絶望に似た感慨を持たざるを得ない。

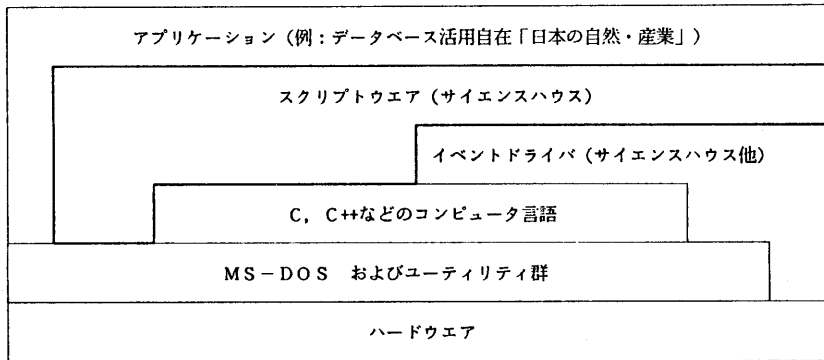
①「教育知らずが作る教育ソフト」

②「思いばかりの作りかけ教育ソフト」

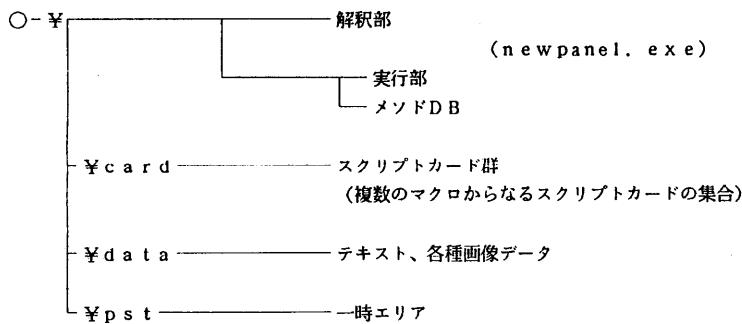
教師や教材編者でない者に、「教育」を教えるのは至難の業である。しかも、「教育知らずが作る教育ソフト」の被害は大きいのである。逆に、「教育」を知る教師や教材編者が「思いばかり」で終わらないソフトを作れば、教育界に対する貢献は極めて大きくなるに違いない。我々の研究開発の動機はここに始まっている。

2. 「スクリプトウェア」の構成

2.1 「スクリプトウェア」の位置



2.1.1 「スクリプトウェア」の構成



3 「スクリプトウェア」の特徴と機能

をひとまとめに受け取って解釈し実行する違いがある。

3.1 「スクリプトウェア」の特徴

3.1.2 スクリプトウェアがもつ知識ベース

知識ベースは固定のものではなく、随時追加変更できるものである。今回は、オブジェクトとその動作に関するものを中心としているが、今後はメソッドを随時追加し、強化される。

3.1.1 スクリプトウェアは、エージェントシステムのプロトタイプ

エージェントは、その名前と知識ベース(メソッドDB)と解釈機能を持っている。

今回のエージェントは、「スクリプトウェア」という名前をもち、マクロ(スクリプトのステートメント)ごとにメソッドが用意される知識ベースがあり、マクロ解釈機能を持っている。

エージェントは、オブジェクトの一種ではあるが、従来のオブジェクトが1つのメッセージずつ受け取って、その都度メソッドを実行するのに対して、複数のマクロ(スクリプトのステートメント)からなる文書(スクリプト)

3.1.3 ソフトウェア生産性の向上(1)

同一のスクリプトに同一のマクロが複数回記述されても、同一のメソッドが複数回呼ばれるに過ぎない。従来は、「構造化プログラミング」の精神論で類似コードを排除する努力を重ねてきたが、「スクリプトウェア」では開発者に精神論を強いることなしに重複コードを排除することができる。

3.1.4 ソフトウェア生産性の向上(2)

スクリプトの記述を変えるとスクリプトウェアは別のアプリケーションの動作をする。

メソッドの多くはそのまま使用できるので、アプリケーションごとに処理系をコーディングする必要はない。

従来から「ライブラリ管理」によって同様の生産性向上の努力がされてきたが、「ライブラリ化」するための工数や「ライブラリ管理」に必要な工数、さらに非人間的な注意力が必要であった。

「スクリプトウエア」では、応用アプリケーションが作成され続けられる限り、メソッドが散逸したりすることはない。また格別の「ライブラリ化工数」や「ライブラリ管理工数」は不要になる。

3.1.5 非日常的な「アルゴリズム」の排除

スクリプトの記述は、エンドユーザーの思考方法に合致しなければならない。

(非日常的な「アルゴリズム」の例)

”はじめにある数 I を 0 にする。次に、I に 1 を加えてある処理 a を行い、処理が終わったら、さらに I に 1 を加えて処理 a を繰り返す。I が 10 を越えたら、処理 a を実行せずに終了する”

上記のような、ソフト技術者にとって当たり前アルゴリズムも多く教師や教材編集者には大きな精神的負担を要求することになる。

仮に今後新しいメソッドを用意するとしても、「スクリプトウエア」では、上記のようなアルゴリズムをユーザーに強要することは回避できる。

ユーザーは、「スクリプトウエア」に対して次のような日常的な要求を記述すればよいようになるはずである。

(日常的な言葉)

”処理 a を 10 回繰り返さない”

3.2 「スクリプトウエア」の機能

現在用意されているメソッドの機能が「スクリプトウエア」の機能である。以下には、現在「スクリプトウエア」に用意されているメソッドの主な機能を述べる。

- 3.2.1 任意の大きさ、種類、色でパネル、ボタン、エディタを構成できる。
- 3.2.2 ボタンに BMP 画像を貼り込むことができ、アイコンボタンが簡単に作れる。
- 3.2.3 階層パネル構造となっており、パネル内に任意のボタンが設置できる。
ボタンの押下によって、階層化したパネル系列の上下を次々と開いたり、別系列のパネルを開いたりすることができる。
- 3.2.4 文字出力機能の自由度が高い。
学習者にインパクトのあるようなテキストアトリビュートを自由に設定できる。

3.2.4 空白付きの印刷ができる。これで、穴あき問題の出力が可能になる。

3.2.5 紐付き文字列機能がある。

学習者が興味をもつ言葉や重要語にパネルを紐付けできるようにしてある。これらの言葉をマウスカーソルでクリックすると、紐付けられたパネルが開き、その場で辞書を引くような感覚が味わえる。ハイパーカードライクな機能である。

3.2.6 用語事典機能がある。

ユーザーが作成した用語、読み、回折を書いたデータファイルをパネルから自由に呼び出すことができ、50音見出しの辞書として活用できる。

3.2.7 豊富なファイル形式に対応するグラフィック機能がある。

現状で、BMP, PCK, MAGに対応している。

機能としては、line, circle, pixel, rectangle, ... などの基本的な機能を揃えている。

3.2.8 快適な文字列入力を可能とするエディタ機能がある。

例えば、パネル上で学習者が問題に対する答えを入力できる。

入力した解答の正解、不正解を判断するドリル機能も併せてもっている。

4 「スクリプトウエア」の応用事例

ここでは、本研究開発で作成された「スクリプトウエア」を用いて、作成されたアプリケーションソフトウエア「データベース活用自在：日本の自然・産業」（株式会社学習研究社発売）を応用事例として取り上げる。

「スクリプトウエア」を作成したのは標記株式会社サイエンスハウスの研究開発グループであるが、これを利用して応用事例の「データベース活用自在：日本の自然・産業」を作成したのは、株式会社学習研究社の優れた教材編集者である。C言語などを知らなくとも「教育」を知る教材編集者が「思いきり」ソフトを作れば、かくなるものが作成できるというよい例である。

4.1 「データベース活用自在：日本の自然・産業」の狙い

探索型である。学習者の「能力に応じた」「自発的」な学習活動を促し、「発見」を導く。

学習者は、まず目次をひもとき、大項目やその小項目を選択する。その項目に対応した大

筋の図解パネルが表示される。説明文の中を探索すると、ハイパーカードのようにさらに文中用語の説明を見ることができる。表示されている図解パネルと関連するパネルもそれぞれにまとめられているので、気が向けば探索することが可能である。関連パネルから関連パネルへと興味を赴くままに探索しつつさらに知識を獲得することができる。思いついた用語について辞典を引くように説明を見ることも可能である。

「データベース活用自在：
日本の自然・産業」
における探索の手段

- ① 目次探索
- ② 検索・保存
- ③ 紐付け文字列探索
- ④ 関連パネル探索
- ⑤ 用語辞典探索

各パネルがそのままに印刷できるほか、重要語が穴開きになって印刷される「穴開き印刷」もできる。簡便に穴開き問題が作成できる機能である。

「データベース活用自在：
日本の自然・産業」
における印刷の手段

- ① 開いているパネル1枚を印刷
- ② パネル全部を印刷
- ③ 開いているパネル1枚を穴開きで印刷
- ④ パネル全部を穴開きで印刷

4.2 操作の流れ(例)

4.2.1 オープニング画面

データベース活用自在
日本の自然・産業

学習
検索
編集
終了

4.2.2 学習範囲の選択

[オープニング画面] で [学習] ボタンを押下すると開かれる。

学習範囲の選択

検索 項目

1 日本の自然と生活、身近な地域	1 日本の自然と生活
2 九州地方	2 身近な地域
3 中国・四国地方	
4 近畿地方	
5 中国地方	
6 関東地方	
7 東北地方	
8 北海道地方	
9 テーマ別学習	

詳細 決定

4.2.3 学習範囲の詳細設定

[学習範囲の選択] 画面で [詳細] ボタンを押下すると開かれる。

学習範囲の詳細設定

検索 1 日本の自然と生活、身近な地域

項目 1 日本の自然と生活

詳細

1 日本の位置と範囲
2 日本の自然
3 日本の人工と人々の生活

戻る 決定

4.2.4 検索画面

[オープニング] 画面で [検索] ボタンを押下すると開かれる。

検索

キーワードで該当するパネルを選びます。

キーワード検索

以前の検索記録を呼び出します。

検索記録呼び出し

4.2.5 パネルが開かれる

[学習範囲の選択] 画面または [学習範囲の詳細設定] 画面で [決定] ボタンを押下すると開かれる。

日本の自然・産業

起伏地形

起伏

起伏…川が山地から平地に出るところにできる。
起伏…起伏のゆるやかな傾斜地が広がる。
等高線…谷口を中心に閉鎖的に広がる。
主地の活用
起伏…森林、森林など。
起伏…果樹園、農地など。
起伏…水田、道路、集落など。

地形図を見る

▲起伏地のような山梨県甲府盆地地形

関連パネル 用語辞典

[<] で前のパネルにもどる。[>] で次のパネルに進む。[関連パネル] で表示中のパネルに関連するパネルの一覧を表示する。[用語辞典] で、思いついた用語の説明が見られる。

[メニュー] で [オープニング] 画面に戻るか否かを選択できるようになる。[印刷] で、このパネルまたは全パネルのままの印刷または穴あき形式の印刷ができる。

[Help] で、ヘルプメッセージが得られる。

4.3 スクリプトの記述

以下にはスクリプト (マクロの系列的記述) の例を挙げる。

```
-----
panel( @HOME ){          // ホームパネル
    area( 0, 0, 640, 400, 0 )
    area( 15, 39, 609, 321, 1 )
    frame_color( 8 )
    type( rect_edge )
    word( "『日本の自然・産業』", 46 )

    message(open){
        open_panel( @CHILD )
        set_button( menu )
        set_button( print )
        set_button( help )
        set_button( @prev )
        set_button( @next )
        set_button( kanren )
        set_button( vocab )
    }
}

button( menu ){
    message(choose){
        close_panel( help )
        close_panel( print )
        open_panel( menu )
    }
    area( 360, 6, 80, 28, 0 )
    word( "menu.bmp", 0 )
    type( graph_push )
}

(後略)
```

5 マクロの仕様

5.1 スクリプト (マクロファイル) について

スクリプトは、拡張子 .mcr を持つテキスト形

式のファイルである。基本的にオブジェクト記述の羅列である。オブジェクトには、パネル、ボタン、テキスト、グラフィックの4種類がある。

インデントはタブ及びスペースを用いて自由に出來、行中のセミコロン(;)以降は注釈文となる。また、各オブジェクトの名前は自由に付けられ、前方参照可能である。

5.2 パネルについて

パネルは、以下のような形で記述する。

```
panel( パネル名 ){
    パネル設定
    :
    :
}
```

5.3 ボタンについて

ボタンは、以下のような形で記述する。

```
button( ボタン名 ){
    ボタン設定
    :
    :
}
```

5.4 テキストについて

テキストは、以下のような形で記述する。

```
text( テキスト名 ){
    テキスト設定
    :
    :
}
```

5.5 グラフィックについて

グラフィックは、以下のような形で記述する。

```
graphic( グラフィック名 ){
    グラフィック設定
    :
    :
}
```

5.6 message() について

パネル、ボタンの動作を設定する命令であり、次のような形式で記述する。

```
message( action ){
    動作の設定
    :
    :
}
```

5.7 カラーについて

カラーは基本的に16色使用できる。パレットはMS-Windows標準パレットである。指定方法は番号かコードで、以下の通りである。

色	カラー番号	カラー名
黒	0	BLACK
暗い赤	1	A_RED

暗い緑	2	A_GREEN
暗い黄	3	A_YELLOW
暗い青	4	A_BLUE
暗いマゼンタ	5	A_MAGENTA
暗いシアン	6	A_CYAN
灰	7	A_BLACK
暗い白	8	A_WHITE
赤	9	RED
緑	A	GREEN
黄	B	YELLOW
青	C	BLUE
マゼンタ	D	MAGENTA
シアン	E	CYAN
白	F	WHITE

この他に、色を塗らない(透明)SEETHROUGHを指定することが出来る。

5.8 テキストの出力文字列について

テキストの出力文字列は「ダブルクォーテーション()」で括って記述する。

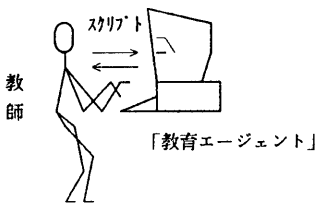
文字列には「%」で始まるアトリビュートを指定することが出来る。

ただし、%をそのまま出力したいときは、%%と記述する。

テキストの桁数が奇数の場合、自動的に禁則処理を行う。`.,..., . . .` のどれかが行頭に来てしまうような場合に、前の行の行末に出力するようにする。

6 将来への発展

6.1 教育エージェントシステムへの発展



教師や教材編集者でない者に、「教育」を教えるのは至難の業である。しかも、「教育知らずが作る教育ソフト」の被害は大きい。逆に、「教育」を知る教師や教材編集者が「思いばかり」で終わらないソフトを作れば、質の高い「教育ソフト」が多量に供給されるようになる。*

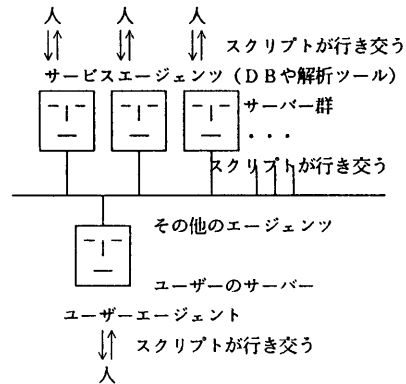
教師の意向を汲んで動作する「教育エージェントシステム」が望まれるところであるが、本システムはそのための第一歩である。スクリプトの知的生成機能の作成、「解釈部」

「メソッドDB部」「メソッド実行部」が相互の独立性を強化されなければならないなどの課題が残っている。

文部省や財団法人コンピュータ教育開発センターなどの公共機関のご指導と調整を図りながら、今後の課題に取り組みたい。

* これらの課題が実現することは、我々のようなソフト開発業の存立を危うくするものとして、同業者や社内にも反対意見は強い。しかし、自分たちの職域を本質的に小さくする努力をすることが職業倫理にかなうものであると我々は考えている。逆に職域の拡大のみを望むことがどれだけ社会的発展に逆行するかを考えれば、その理は明らかである。

5.2 ネットワークエージェントへの発展



我々は、幾つかの業界団体で「科学技術資源の有効利用」のための研究開発のプロジェクトに取り掛かり始めている。

これは、主としてインターネットを介して全国の科学技術用サーバーを結合し、ファクトデータベースや解析ツールを相互に結合して利用できるようにするためのプロジェクトである。これらのプロジェクトは、公共性の強いものなので、文部省はもとより、通産省

(IPA)や郵政省などの協調も必要であり、業界団体としての模索が続けられている。この仕組みに関して議論するもはや余地はないが、アプリケーションレベルのユーザーエージェントと科学技術用サーバーのサービスを代行するエージェント等が相互に協力するヘテロジニアスなエージェントフィールドが構築される。

人とエージェント、エージェントとエージェントが会話するためには、ある種の言語が必要であり、今回発表した「スクリプト」がその役割を担うもののプロトタイプとして期待されている。

我々は、そのための研究開発も引き続き行っていくつもりである。