

Prolog言語とオーサリング・ツールとの連携： AZ-Prolog と“Authorware 6”の連結用ActiveXコントロールの作成

清水 秀美 (愛知教育大学教育実践センター)

Using Prolog Language in Authoring Tools: Making ActiveX Control for Connection of AZ-Prolog with “Authorware 6”

Hidemi SHIMIZU

(Center for Research, Training and Guidance in Educational Practice, Aichi University of Education)

要約 オーサリング・ツールとして利用するのに必要とされる知識の前提条件が少なく、短期間の学習で利用でき、かつ動画や音声同期、インターネット配信に至るまで多くの機能を提供するMacromedia社の“Authorware 6”に、AZ-Prolog言語によるプログラムを接合することを試みた。ここでは接合のためのActiveXコントロールをVisual Basicで作成した。

Keywords：オーサリング・ツール, Prolog, ActiveXコントロール

Prolog言語にGUIを付与すること

これまでに教材作成をProlog言語で試みてきた(清水, 2000, 2001)。しかし、この言語は一般に論理演算を得意とするもので、GUIを備えたものは少ないか、あるとしてもかなり高価である。この言語の長所を生かし、かつWindows環境でGUIを十分に活用したい。このような要求は一般的であるのだろう。JavaとPrologとの連携を試みたシステムが市販されている。たとえば、IF/Computer社のMinervaなどがある。しかし、日本語対応のProlog言語でGUI対応のソフトとなると、それらを使うにはかなりの経済的負担と知識習得のための時間的負担が強いられる。教材作成という目的に照らしてProlog言語と連携するのであれば、Windows対応のオーサリング・ツールを求めるほうが、教員養成系のニーズにかなうと考えられる。その場合、オーサリング・ツールの操作習得には出来るだけ時間的な負担の少ない方が望ましい。今後、教員養成系でもオーサリング・ツールの習得が必須になると考えられるが、本題に入る前に、その教員養成系での必要理由について考えてみたい。

教員養成系でのオーサリング・ツールの必要性

教育現場の様々な状況で有効にコンピュータを活用するには、教師は単に他者の提供するソフトのみに依存しているわけにはいなくなる。もちろん本格的なソフトを開発する必要性は無いとの主張もあるが(村上, 2000)、特に小学生相手の授業では生徒の特性に合わせた手作りの教材が効果のある場合も多いのである。生徒の興味を取り込み、授業の流れに即して、かつ教師の教育方針・教授スタイルに合致した既存の教

材ソフトは、存在しない確率が高い。この状況はたとえ文部科学省が推進する教育メディアセンターでのソフト収集量が多くなったとしてもそうである。小学校での教科が7科目として、生徒の興味の種類を仮に5種類、授業形態でドリル形式、ゲーム形式、説明形式、生徒中心、教師中心の形態など5種類、小学校では6学年ある。領域に偏りがないと仮定して、小学校用教育ソフトが総数3000本であったとしても、当該の授業で選べる数は最大に見積もっても $7 \times 5 \times 5 \times 6 = 1150$ で除した約3個以内ということになる。したがって、教師は必然的に教材を自作せざるを得ない場合が生じる。

ソフトの進化によって、書籍が簡単に個人で作成することができる時代である。教材作成ソフトであるオーサリング・ツールも今日急速に進化してきている。授業での繋ぎの教材であれば、3時間の講習で作成できるようになってきている。本格的な商業ベースに乗る教材作成でもC++やJava言語を用いて始めから作成するのと比較して、かなりの時間節約が可能になってきている。言語が不要であるとか、言語学習が無用と言っているのではない。本格的なソフト開発には言語は必要不可欠であることを、誤解を避ける上で述べておく必要がある。ただ、ちょっとした繋ぎの教材作成ではその必要は低いと言っているだけである。

教師の専門性、特に学習指導能力の向上に向けて、オーサリング・ツールの活用は有効であろう。教員養成系学生にオーサリング・ツールを用いて教材作成をさせると、彼らは常に生徒の理解状況を意識しつつ、あらゆる場合を想定して教材を作成し、授業場面での適用方法を考慮することになる。教材を組み上げるこ

とにより、問題点の明瞭化と限界とが明確に意識されるようになる。この意味で近い将来、ワープロと同じく、オーサリング・ツールは教員養成系機関で教員資質向上の道具として必須のものになるだろう。

単に教室で教育機器を使用するのみでなく、教員の授業設計技能の向上にコンピュータを積極的に活用するという観点で、使い易いオーサリング・ツールの選択は欠かせない。

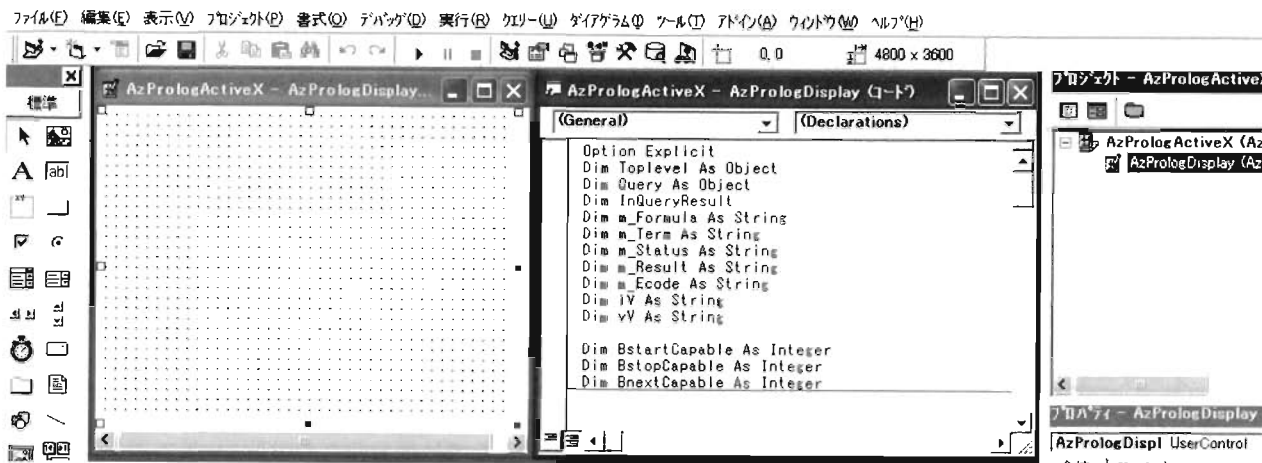


図 1. Visual Basic によるActiveXコントロールの作成

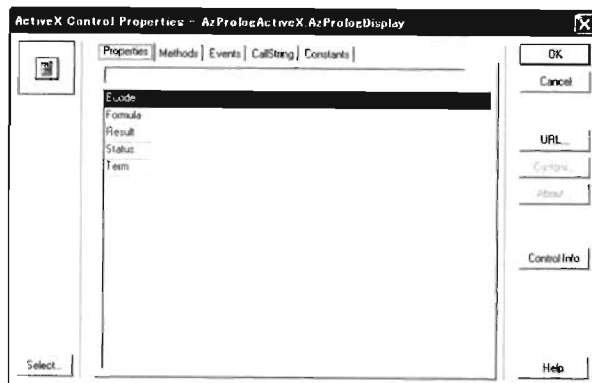


図 2. 設定されるプロパティ (Property)

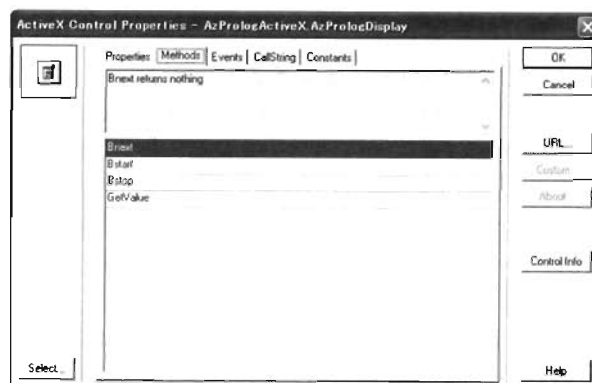


図 3. 設定されるメソッド (Method)

```
/* C:\pwork>azpc -p MyApp.pl /e MyApp /curses /lib user32.lib */
/* 上記の式にて実行ファイル(MyApp.exe)を作成し、これを実行する */
/* これにより OLE Objectとしてレジストリーに登録される。 */

top_level: -
  automation(
    '12B061D0 - DEAQ1CE A641 0000F4CA088B!',
    'MyApp.Toplevel',
    'c:\pwork\Myapp.exe p').

:- public sum/2.      /* public 宣言により任意の述語が呼び出し可能 */
sum(0,0): -!.
sum(N,S): - N1 is N-1,sum(N1,S1), S is S1+N.

:- public append/3.  /* public 宣言により任意の述語が呼び出し可能 */
append([],Y,Y).
append([A|X],Y,[A|Z]): - append(X,Y,Z).
```

図 4. OLE Objectとして登録されAZ-Prolog部分

オーサリング・ツールの選択の基準

教員はプログラマーを目指すものではない。従ってワープロと同じようにオーサリング・ツールを用いて自分の授業に必要な教材が作成できれば良い。従って、はじめからプログラム学習・訓練を想定するものは避けたい。以上より選択基準の第一は、WYSIWYG (What you see is what you get) という手軽さで。コンピュータと学習者のインタラクティビティーが可能であることである。

第二の選択基準としては、マルチメディア対応で、その素材の取り込みに制限が出来るだけ少ないものということになる。第三基準として、優れた互換性のある作図、動画作成ソフトの豊富さに加え、支援サード・パーティが多く存在し、他言語との接続が保証され、かつインターネット配信も容易に可能であるなどの条件を加味とする (Wisn & Thornton, 2002)。

以上の条件より、Macromedia社の“Authorware 6”を選択することにした。ここでの目的は、この Authorware の ActiveX コントロール機能を活用して、日本のPC上で一般的に用いられてきた“Prolog KABA”の系統を引く Sofnec 社の AZ-Prolog との接合を試みる。

AZ-Prolog用ActiveXコントロールの作成

ここでの目的は、ActiveX コントロールを、Microsoft 社の“Visual Basic 6”を用いて図1のようなプログラムを作成することにある。ここで作成したスクリプトは Sofnec 社よりサンプル・プログラムとして提供されたものをモディファイしたものである。

1. プロパティ (Property) として次の5つを設定した (図2)。

- ① MyAppToplevel → 接合する AZ-Prolog 言語によるプログラム名 (OLEObject 名)。
- ② Formula → プログラム中の述語名。
- ③ Term → 結果として求める変数名。
- ④ Result → 処理結果の取得。
- ⑤ Ecode → エラー番号表示。
- ⑥ Status → “Succ”, “Fail”, “Error” のいずれかの表示。

2. メソッド (Method) として次の3つを設定する (図3)。

- ① Bstart → プログラムの起動。
- ② Bnext → 二次以降の解の取得。
- ③ Bstop → プログラムの停止。

(注：メソッド GetValue は使用しない。)

以上により作成された ActiveX コントロールは AZ-Prolog 言語との接合において、大幅な変更を必要としない。



図5. 左上はAuthorwareプログラム、下がCalculationアイコン部のスクリプト、右上が実行結果。

ActiveXコントロールで利用可能なAZ-Prologプログラムの作成

Sofnec社の提供するマニュアルに沿って作成する(図4)。留意点を以下に列挙する。

- ① CLSIDをGUIDGEN.EXEで生成する。生成された値を述語automationの第一引数とする。
- ② Authorwareで述語としてFormulaに割り当てるには“:-public [述語名]/[引数の数]”を宣言する。
- ③ 実行形式ファイルに変換するには、作成したプログラム名をMyApp.plとすると、
C:\azpc -p MyApp.pl /e MyApp /curses
または、
C:\azpc -p MyApp.pl /e MyApp /i
で行なう。
- ④ この生成ファイルを実行してOLE Objectとして登録する。Regedit.exeで登録状況を確かめることが出来る。登録名は述語automationの第二引数で指定される。

AuthorwareによるAZ-Prologプログラムの操作

図5(左上)に示すデモ用のAuthorwareプログラムを例に説明しよう。Displayアイコン“Result Display”は結果を表示するものである。“AZ-Prolog”はActiveXコントロールアイコン名である。この“AZ-Prolog”を操作するのがCalculationアイコン“Set Variables for AzProlog”である。Calculationアイコン内にスクリプトを記入するが、それについて見てみよう(図5の下)。

ActiveXコントロールのプロパティ(Property)に値を設定するには、SetSpriteProperty関数で、値を取得するには、GetSpriteProperty関数で、メソッド(Method)を使用するには、CallSprite関数をそれぞれ用いる。

先ず最初にActiveXコントロールで使用するOLE Object名を次のように指定する。

SetSpriteProperty(@" Activeアイコン名",
#Property名, OLE Object名)

これに続いて、プロパティ、Formula、Termを定義する。この後、次のようにしてAZ-Prologプログラムを実行する。

CallSprite(@" Activeアイコン名", #Bstart)

この後、結果を次により収集する。

GetSpriteProperty(@" Activeアイコン名",
#Result)

以上を実行した結果が図5の右上である。

評価プログラム

AZ-Prolog言語で簡単なappend(Y,X,[a,b,c])をFormulaの値として、TermをXとして、その値をメソ

ッドBnextにより、次々に求めることにする。ただし、Statusが“Fail”になったとき、ボタンStopを押したときにInteractionアイコンの繰り返しから抜けるものとする。作成したプログラムを図6に示す。

参考文献

- 村上温夫 2000 ITでめざせ、教育改革： 発見・探求の喜びをインフォメーションテクノロジーで新曜社
- 清水秀美 2000 筆算シミュレータの作成 愛知教育大学教育実践センター紀要 No.3 73-76.
- 清水秀美 2001 算術式のGUI作成 愛知教育大学教育実践センター紀要 No.4. 75-78
- Sofnec 2000 *User's Manual: AZ-Prolog for Win32*. Sofnec INC.
- Wilson,S.J.& Thornton, J. 2002 *Authorware 6: Inside Macromedia*. Onward Press

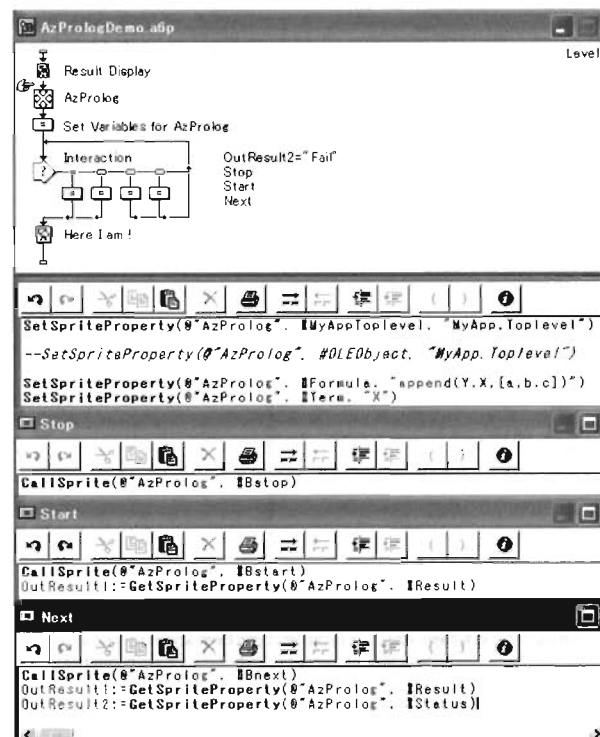


図6. 評価プログラム。4種類のプロパティ変数と、3種類のメソッド変数の動作確認を行なった。