INTRODUCTION

A course authoring language is a tool for developing computer-assisted instruction (CAI) lessons that normally takes care of managing the delivery of lesson frames, assessing student responses to questions and branching to appropriate segments of the lesson. Although a CAI course can always be written in any computer language, a course authoring language allows the author to concentrate on his lesson contents by taking over much of the drudgery of frame delivery, response assessment and branching.

Some examples of authoring languages commonly used today are PILOT, IIS SCHOLAR-TEACH, and GNOsis. Of these, GNOsis is not only easy to learn but also affords the author more power than PILOT or SCHOLAR-TEACH. However, GNOsis is written in ALGOL and produces a lesson in ALGOL - a language that is not used or supported by many computer installations.

CALIS is a course authoring language that is generally compatible with GNOsis. It is a SAS program that writes SAS programs. The programs that it writes are CAI courses that are directly executable by students. We designed and created CALIS for these reasons:

- to permit us to migrate many of our GNOsis lessons created in DEC installations running ALGOL to IBM installations running SAS
- to create an authoring language that is extendible and just as powerful as SAS
- to allow us to take advantage of the data analysis and graphics capability of SAS and SAS/GRAFPH to extend CALIS in the future.

CALIS is only the first in a series of course authoring languages that are compatible with one another in terms of the CAI lesson scripts that they translate. When all the compatible course authoring languages are in place, a teacher need only create a lesson once. This lesson can then be translated by any one of the authoring languages into a runnable CAI course in different high-level programming languages that will run in a microcomputer as well as in a mainframe. CAL/PAS creates courses in PASCAL, CAL/PL creates courses in PL/I and CAL/AD creates courses in ADA.

SOME BASIC FEATURES

With CALIS, an author can begin writing lessons using no more than 5 commands. As he becomes more proficient in both CAI and CALIS he can take advantage of other CALIS commands.

A course author can embed any number of SAS statements in his lessons. Thus, the SAS-literate author can extend CALIS to the limits of SAS itself.

Unless the author forbids it, CALIS gives the student the facility to alter the path that he takes through the course. Students also have the facility to force the lesson to pause at the beginning of each frame. This is useful for students working at CRTs.

CALIS randomly generates a stock message to a student's response to a question depending on whether the student got the right answer on his very first try, after several tries or whether he did not supply the correct answer at all. These stock messages are also personalized with the student's first name approximately 15 percent of the time. The stock responses and the frequency of message personalization may be changed by the author.

CALIS will not leave the student at a dead-end. If the student is no longer making any significant progress in a question frame, CALIS gives him the answer and puts him on the next frame.

A student taking a CALIS course always has the facility to leave comments to the course author. There is no limit on the length of the comments.

CALIS keeps track of student progress through the course. The author can use these performance statistics variables to control the student's path through the course. The statistics, along with

- the student's name
- the date and time he took the course
- the time he spent on the course
- all unanticipated responses
- any comments left by the student

may be kept in a student log for inspection by the author. The list of unanticipated responses is sorted by question frame. This, along with comments left by the student, provides the author with valuable feedback on his course.

PREPARING AND RUNNING A CALIS COURSE

Preparing a CALIS course is similar to writing a lesson script for, say, a programmed textbook. The lesson script consists of a series of question and text frames with appropriate feedback comments to the student and branching instructions depending on student responses to questions.

The author creates the course using a text
editor such as TSO EDIT (invoked with the ASIS parameter to preserve lowercase characters). In our case, we use a popular screen-oriented editor running in a Z80 microcomputer; the course is then uploaded to a mainframe computer when editing is completed.

The course consists of the lesson script and embedded CAL/S commands. CAL/S commands are easily distinguished from the lesson proper by the fact that they always begin with a command character (%, by default) which the author may change at any time using one of CAL/S's commands. For example, %QUESTION marks the beginning of a question frame, while ZNOEXTRA is a directive to consider as correct only perfect matches between an author-supplied answer pattern and a student response.

When the course creation and editing have been completed, the author invokes a Clist, CALS, (this could just as well be a CMS EXEC) that produces a complete and executable CAL course in SAS out of the CAL/S course. Students then take the course by invoking a second Clist, CALSS, which delivers the SAS-coded course.

Figure 1 shows the steps in running a CAL/S course.

**SUMMARY OF CAL/S COMMANDS**

CAL/S commands are prefixed with a command character, which by default is the percent symbol (%). The author may change this at any time using the XCC (command character) command. Commands may be in upper or lower case.

CAL/S commands that consist of four or more characters may be abbreviated to their first three characters (except the XSEND command).

The author may embed any number of comments in his lesson. The students do not see the comments. A comment line begins with either the exclamation point (!) or an asterisk (*) in column 1.

Some of the commands use a frame label. A label may be any name devised by the author that conforms to SAS label name conventions (eight characters or less, beginning with a character or underscore) or any one of these built-in special labels:

- next -> refers to the next frame
- repeat -> refers to the beginning of the present frame
- L_start -> refers to the beginning of the lesson
- L_end -> refers to the end of the lesson.

A label is the target of a ZGOTO command. The author can tag the beginning of a frame (TEXT or QUESTION) with a label, for example,

TEXT menu1:
QUESTION question:

At some point in the lesson branching can be accomplished with, say,

ZGOTO menu1;

When a label is defined, the label name must be followed by a colon; the label name is followed by a semicolon in the ZGOTO command (although, if the author forgets, CAL/S makes the proper adjustment).

This section presents a brief description of CAL/S commands. The following notations are used in this section:

1. < reqd > - parameters enclosed in angle brackets are REQUIRED
2. [option] - parameters enclosed in square brackets are optional
3. answer pattern - an author-supplied string against which student responses will be matched.

**A. Commands that appear once in a lesson, and at its beginning:**

**ZCOPYRIGHT [text]**
"text" will appear in the lesson as a copyright notice

**ZTEACHER [teacher name and/or address]**
The line "The author of this lesson is" [teacher name] appears in the lesson.

**ZNAME**
CAL/S will invite the student to supply his full name. The student's first name is used in personalizing stock responses. The full name is recorded on the student log file along with student responses to questions that were unanticipated by the lesson author (provided that the XDISK command also has been issued).

**ZDISK or ZDSK**
The student log file is opened and CAL/S records all unexpected responses from the student along with his name, date and time, and personal statistics.

**B. Command that appears once in a lesson, and at its end.**

**ZEND**
This must be the last statement in any CAL/S lesson. It is ALWAYS required.

**C. Commands that set switches (can go anywhere, any number of times)**

**ZEXTRA and ZNOEXTRA**
From this point on in the lesson, allows or disallows extra strings in a student response during matching with the answer pattern.

**ZPAUSE and ZNOPAUSE**
In Version 1.0 of CAL/S, the command
PAUSE causes the message "Please push RETURN to continue ..." to be printed (CALIS then waits for the student to do so) at the beginning of a TEXT or QUESTION frame. The %NOPAUSE command negates the %PAUSE command.

In Version 1.1, the student has the facility to override the author with his own %PAUSE command at run-time. The student's %PAUSE command works as a toggle switch. Issuing it when PAUSE is in effect produces a NOPAUSE.

LOCK and UNLOCK
LOCK disallows any branching to be dictated by the student.

ICC <command character>
Used to change CALIS's command character. All CALIS commands from this point on (until the next occurrence of ICC) are preceded by this command character. The default command character is the percent (%) symbol.

When the lesson begins, the default settings of switches are UNLOCK, PAUSE and EXTRA.

D. Commands that extend the power of CAL/S

ISAS <one-line SAS statement>, or
ISAS any number of SAS statements ISASEND

These commands cause the SAS statement(s) to be embedded in the lesson at that point. The author should make sure that the statements make sense in the context of the lesson. All SAS statements that are valid in the DATA step, except the DATA statement are permitted. Note that ISASEND is the only command that cannot be abbreviated to three characters because doing so will cause a conflict with the ISAS command.

E. Command that presents a TEXT frame

TEXT
TEXT <label>; TEXT <label> <conditional statement>; [. text lines .]

All lines following the TEXT command until the next CAL/S command line will be presented to the student. The <conditional statement> may be of the form

IF xct < 70 THEN GO TO lessonl;

where "lessonl" is an author-defined label in the course. The variable "xct" is one of five run-time variables that CALIS tracks for the author. The five variables are

\[ \text{score} = \text{Number of first time right answer} \]
\[ \_iscore = \text{Number of finally right answers} \]
\[ \_qcnt = \text{Number of questions so far presented to the student} \]
\[ _pct = (\text{score/}_{qcnt})\times100 \]
\[ _lpct = (\text{iscore/}_{qcnt})\times100 \]

F. Commands that appear in a QUESTION frame

QUESTION
QUESTION <label>:
QUESTION <label> <conditional statement>; [. text lines .]

All lines following the QUESTION command until the next CAL/S command line will be presented to the student. CAL/S then waits for a student response for possible matching with author-supplied answer pattern. The <conditional statement> has the same form and use as described in the TEXT command.

XRIGHT <answer pattern>
[... text lines ...]

Matches the answer pattern with the student response. If a match is found a stock feedback message is randomly generated and the following text lines (if any) are presented to the student. The matching is affected by the EXTRA/NOREXTRA switch. The lesson then moves to the next frame or to the lesson segment dictated by a GOTO command.

XCA <answer pattern1,answer pattern2, ...>
[... text lines ...]

Similar to XRIGHT command but more powerful because it allows alternate correct answers with a single command.

NOTE: The command XRIGHT or XCA must precede all the commands that are listed below which are valid in a QUESTION frame.

ZWRONG
ZWRONG <answer pattern>
[. text lines .]

Matches the answer pattern (if any) with the student response. If a match is found, a stock message is randomly generated and any following text lines are presented to the student. The matching is affected by the EXTRA/NOREXTRA switch. The question is presented to the student again, unless the path is altered by a following GOTO command.

If both forms of the ZWRONG command are used in the same question frame, the form with the <answer pattern> must precede
those without an answer pattern. The \texttt{WRONG} command must come \textit{after} all \texttt{RIGHT} commands.

\texttt{\%WRONG <answer pattern1, answer pattern2, ...> [ .. text lines .. ]}

Similar to \texttt{WRONG} command but more powerful because it allows alternate wrong answer patterns with a single command.

\texttt{\%NEUTRAL <answer pattern> [ .. text lines .. ]}

Used when the author considers the student response to be neither right nor wrong. The following text lines are displayed and the lesson goes to the next frame \textit{unless} the author follows it with a \texttt{GOTO} command. This is most useful when presenting a menu to the student. Again, if both forms are used in the same question frame the form with the answer pattern must precede the one without.

\texttt{\%SAME}

Causes the text lines associated with the previous command to be presented to the student. This command is rarely used in CAL/S but was retained to maintain compatibility with GNOSIS.

\texttt{\%LACK <answer pattern> [ .. text lines .. ]}

If the student response is missing the answer pattern a stock message and the following text lines are given to the student. The question is presented to the student again unless the path is altered by a following \texttt{GOTO} command. This command should be used with extreme caution. If used, this command must come after pattern-matching commands such as \texttt{RIGHT}, \texttt{CA}, \texttt{WA} and \texttt{WRONG}.

\textbf{G. Command that alters the course path}

\texttt{\%GOTO <label>;

Causes the lesson to branch to the lesson segment with the designated label. The label may be defined by the user or may be any one of the four built-in labels described earlier. This command may go anywhere, and may be used repeatedly.

\textbf{SAMPLE CAL/S COURSE}

A sample course created with TSO Edit is shown in Figure 2. In this example, all commands are shown in uppercase and are spelled out completely. CAL/S will accept commands in any mix of upper- or lower-case. It also allows one to abbreviate to the first three letters of any command \textit{except}, of course, the SASEND command.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{sample_course.png}
\caption{A sample CAL/S course.}
\end{figure}

\textbf{Figure 3 shows a segment of the student session. The following should be noted:}

\begin{itemize}
\item CAL/S signs on with instruction to the student on how he can alter the lesson flow (provided the lesson author allows it). The author disallows it with the \texttt{LOCK} command.
\item The \texttt{NAME} command causes CAL/S to ask the student for his full name. The name is recorded in the \texttt{STUDENT LOG FILE}. The log file is activated by the \texttt{DISK} command and contains along with the student name, the time the lesson started, the time it ended, elapsed time, all unanticipated responses by the student, student performance data and any messages he left for the lesson author.
\item All \texttt{TEXT} and \texttt{QUESTION} frames show the message 'Please push RETURN to continue ...' unless the author disables it with the \texttt{NGPAUSE} command and the student did not override it. The student toggles \texttt{PAUSE} on and off by responding with \texttt{PAUSE} to a question. CAL/S reminds him of the action taken, and the question is re-asked.
\item If the student provides the wrong answer to a question, the question is normally displayed again so as not to put those who are working on CRT terminals at a disadvantage.
\item When a student supplies an answer that contains more characters than are provided in the teacher answer pattern, CAL/S echoes the entire student response and marks the portion of the response that matched the answer pattern with '"'. The author (teacher) can force an exact match with an answer pattern using the \texttt{NOEXTRA} command.
\item CAL/S randomly generates a stock message that depends on whether the student got the answer on the first try, after several tries or did not get the right answer at all. These stock messages may be changed by the author. CAL/S also personalizes the stock message by using the student's first name (provided that the \texttt{NAME} command was issued by the author previously) approximately 15 percent of the time.
\item CAL/S produces more compact code if the author uses \texttt{CA} or \texttt{WA} commands in place of \texttt{RIGHT-SAME} or \texttt{WRONG-SAME} combinations. In this lesson, the commands:
\begin{verbatim}
  "RIGHT text ...
  "RIGHT command
  "SAME
\end{verbatim}
are more efficiently replaced with the single command:
\begin{verbatim}
  "CA text,command ...
\end{verbatim}
CALIS — PRESENT AND FUTURE

At present, CALIS is being used at a major Federal agency to (1) teach DP instructors how to write and run a CAL/S course, (2) deliver security briefing to all users of the central computer both at headquarters and at field offices and (3) supplement an entire DP training program that includes such courses as Introduction to Data Processing, SAS, TSO, JCL.

CAL/S-created courses can be taken by students with perhaps only a portable printing terminal at remote locations. CAL/S can be used also to deliver a survey questionnaire.

CALIS was written in SAS 79.5 and is upward compatible with present and future releases of SAS. A more powerful version, written in SAS 82, is under study. We expect this version to take advantage of smart CRTs (for those who have them) with full video attributes such as blink and reverse video. Surely, the complement packages such as CAL/PAS which we are initially targeting for micros will have an install program for different CRTs and will have the capability of supporting split-screen displays.

REFERENCES

1. GNOSIS was written by Walter Maner and Jacob Palme and is available from DECUS (DEC Users Group).

2. CAL/S, CAL/PAS, CAL/PL and CAL/AD are trademarks of Darwin Systems, Inc.

3. Address all inquiries to:
   Darwin Systems, Inc.
   17 Thorburn Road
   Gaithersburg, MD 20878

<table>
<thead>
<tr>
<th>Procedure Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Assumptions:
LESSIN.DATA = file containing the lesson created by the author
COURSE.DATA = file to contain runnable CAL/S lesson produced by CAL/S. This file must be preallocated.
CHEM.DATA = file to contain student performance data

2. The author executes CAL/S:
   CALS lessin.data course.data

3. The student takes the course by invoking the CALSS CList:
   CALSS course.data chem.data

Comment
Step 2, running CAL/S, is done only once provided that CAL/S did not find any errors in the author’s lesson. Step 3 is carried out by each student who wants to take the course.

FIGURE 1. HOW TO RUN CAL/S
This is a sample CAL/S lesson;

* The command character is changed to " intercourse ";

—1—

(c) John Doe Enterprises

TEACHER Mary I. Instructor, (304) 353-3333, Mail Stop AI-5020

FAX

0070 "TEXT

Welcome to the Introductory Course in CAL/S. Before attempting

this lesson you should read the CAL/S primer.

0110 "QUESTION menu:

The following choices are available to you:

0120 0 - Exit from this lesson

0130 1 - Introduction to CAL/S

0150 2 - Creating a CAL/S lesson and running it

0160 3 - Summary of CAL/S commands

0170

0180 Please choose a number (0,1,2,3).

0190 "NORMAL 0

0200 "QUIT L end;

0210 "NORMAL 1

0220 "QUIT intro;

0230 "NORMAL 2

0240 "QUIT create;

0250 "NORMAL 3

0250 "QUIT command;

0270 "NORMAL

0280 I did not quite get that. I am expecting a number between 0 and 3.

0290 Please try again.

0300 "QUIT repeat;

0310 "QUIT intro;

0320 At the end of this module you should be able to

0330 1. Explain what CAL/S is and what it does

0340 2. Describe the types of lines contained in a CAL/S course

0350 3. List the advantages of CAL/S

0360 4. Distinguish between a TEXT frame and a QUESTION frame

0370 5. Describe the CAL/S commands found in a QUESTION frame

0380 6. List the types of information contained in a student log file

0390 "TEXT

0400 CAL/S is a course authoring language similar to QNOSIS. But while

0410 QNOSIS is written in AUDL, and produces a CAL lesson in AUDL, CAL/S

0420 is written in SAS and produces a lesson in SAS.

0430

0440 You create a course, really a lesson, in CAL/S using a text editor like

0450 PSDT. The course may be a numbered or unnumbered text data set. It

0460 consists of (1) CAL/S commands, and (2) text lines. The text lines are

0470 what the students see.

0480

0490 A CAL/S command always begins with a command character in column 1.

0500 By default, this character is the % symbol. You may change it anytime.

0510 "QUESTION

0520 A CAL/S lesson consists of two kinds of lines. Name one.

0530 "RIGHT text

0540 A CAL/S lesson consists of:

0550 (1) CAL/S commands (those that begin with the command character)

0560 (2) Text lines (that the students see).

0570 "RIGHT command

0580 "SAME

0590 "WRONG

0600 A CAL/S lesson consists either of (1) Command or (2) Text lines.

0610 Please enter either one.

0620 "QUESTION

0630 A CAL/S lesson should be an unnumbered text data set.

0640 Is this true or false?

0650 "TA true

0660 "WA true

0670 You're all right. It may be numbered or unnumbered.

0680 "WA true

0690 You goofed here. The answer is false. It may be numbered also.

0700 "QUIT next;

0710 "WRONG

0720 I did not get that. Please answer TRUE or FALSE.

............ additional frames ............

0810 "QUIT menu;

0820 "COURSEx create:

............ module on Creating and Running a CAL/S lesson .........

0900 "QUIT menu;

0910 "QUIT command;

............ module on CAL/S Commands goes here ............

0980 "QUIT menu;

0990 "END

* END must always be the last line in a Lesson;

FIGURE 2. SAMPLE CAL/S LESSON
SAMPLE INTERACTION BETWEEN THE STUDENT AND THE LESSON

In this sample interaction, all student responses are marked with an arrow (→) on the left-hand side.

Welcome to CAL/S.

INSTRUCTION NOTE:

When I am waiting for your response you may enter
HELP → to start the lesson
SKIP → to skip to the next frame
END → to go back to the prior frame
EXAM → to pause between questions (if I am not pausing now)
ON EXAM the pause (if I am now pausing)
END → to list this module.

CAPTIONS (c) John D. Breuer, 1986.
The author of this lesson is Mary L. Instructor, (304) 555-3333, Mail Stop A1-0580

Please enter your full name:
→ Jack.
I need your last name also.
Please enter your full name:
→ Jack ripper.
Please press RETURN to continue...

Welcome to the introductory course in CAL/S. Before attempting this lesson, you should read the CAL/S primer.
Please press RETURN to continue...

The following choices are available to you:
0 → Exit from this lesson
1 → Introduction to CAL/S
2 → Creating a CAL/S lesson and running it
3 → Summary of CAL/S commands

Please choose a number (0,1,2,3).

→ help
I did not quite get that. I am expecting a number between 0 and 3. Please try again.

The following choices are available to you:
0 → Exit from this lesson
1 → Introduction to CAL/S
2 → Creating a CAL/S lesson and running it
3 → Summary of CAL/S commands

Please choose a number (0,1,2,3).

→ 1
At the end of this module, you should be able to:
1. Describe what CAL/S is and what it does
2. Describe the types of lines contained in a CAL/S course
3. List the advantages of CAL/S
4. Distinguish between an EXAM frame and a QUESTION frame
5. Describe the CAL/S commands used in a QUESTION frame
6. List the types of information contained in a student log file

Please press RETURN to continue...

CAL/S is a course authoring language similar to OKE/DRE. But while OKE/DRE is written in ADL, and produces a CAL lesson in ALB, CAL/S is written in SAS and produces a lesson in SAS.

You create a course, really a lesson, in CAL/S using a text editor like ED or Edit. The course may be a numbered or unnumbered text data set. It consists of (1) CAL/S commands, and (2) text lines. The text lines are what the students see.

A CAL/S command always begins with a control character in column 1. By default, this character is the % symbol. You may change it anytime.

Please press RETURN to continue...

A CAL/S lesson consists of two kinds of lines. None
→ I forget.
incorrect. Jack.
A CAL/S lesson consists of (1) Command or (2) Text line.
Please enter either one.
A CAL/S lesson consists of two kinds of lines. None
→ ok, a command line to one

OK. A COMMAND LINE IS ONE

I know you could do it.
A CAL/S lesson consists of:
(1) CAL/S commands (those that begin with the command character)
(2) Text lines (that the students see)

Please press RETURN to continue...

A CAL/S lesson should be an unnumbered data set.
Is this true or false?
→ OK.
false.

From this point on the lesson will NOT pause between questions.
Issue this command later to request data.
I will ask the question again if it is OK with you...

A CAL/S lesson should be an unnumbered data set.
Is this true or false?
→ I need help
I did not get that. Please answer TRUE or FALSE.

A CAL/S lesson should be an unnumbered data set.
Is this true or false?
→ ok.
I know you could do it.
You're all right. It may be numbered or unnumbered.

The lesson continues until the student...

either (1) selects Option 0 the next time
the main menu is presented to him, or
(2) he responds with RETURN to a question
(and the course author permits it).

A sample end-of-lesson interaction is shown below:

Out of 20 questions you answered
15 correctly (75%).
You answered 15 (75%) on your first try.
If you have any questions for the author, please type them now. End the message with an extra press on the RETURN key.
→ I think the lesson can stand some improvement
especially in the section on creating a CAL/S lesson
→ (TO STUDENT HITS CANCEL RETURN KEY)
OK/S signing off... Have a nice day.

FIGURE 3. SAMPLE CAL/S STUDENT SESSION