SCENARIO AND SCENE IN COMPUTER ASSISTED LANGUAGE INSTRUCTION: THE TEACHER AND PROGRAMMER 'INTERFACE'

Hüseyin ÖZ (*)

Key Words : CAI, CALI, courseware, curriculum, lessonware, programming, scenario

INTRODUCTION

Computer assisted instruction (CAI) came of age in the early eighties, but only in the late eighties was it allowed to register its marriage to the school curriculum. The major reasons for the delayed wedding ceremony were the early arguments for and against the employment of computers as teaching-learning aids in education, the scarcity of sound courseware applications appropriate to classroom (or lab) implementations, and administrators' hesitations to arrange the necessary 'felicity conditions' in the procedure. However late the 'community' decisions about the field were made, CAI is now strongly tied to the educational curriculum, and they live at most colleges and universities all over the world. As it would have been expected, they also have offsprings, given different names, in various academic fields such as mathematics, physics, education, foreign/second language teaching-learning, and so forth.

Computer assisted language instruction (CALI) is a relatively young offspring of CAI. Whereas it is still in its infancy in Turkey, the number of schools that implement CALI in the US and Europe is very great. Despite pervasive CALI applications abroad, there still remain some problems concerning the development and design of CALI materials, especially at schools where (language) teachers or educators try to develop their own courseware. In this paper, therefore, I would like to focus on 'scenario' and 'scene' in preparing courseware for language instruction with a reference to the roles of the teacher and the programmer.[1]

(*) Hacettepe Üniversitesi Eğitim Fakültesi Araştırma Görevlisi

SCENARIO

Sometimes also referred to as 'drama', the term CALI 'scenario' is, in my conception, analogous to that in an actual film or television program.[2] That is to say, just as film scenarists write the scenario of a film before it is produced and shot, so a courseware application should be developed first in the form of a scenario before it is designed and written on the computer. But my understanding of CALI scenarios are neither identical to those nor hypothetical. Koç (1990), who I sympathize with in this respect, describes the scenario in CALI as the written manifestations of actual CALI activities that will take place in the courseware. In other words, it implies the idea of writing down the courseware procedures, preparing the language materials to be used, and of determining teaching and learning strategies in courseware applications.

"Who is the 'scenarist' in CALI?" one might ask at this point. Naturally enough, s/he is not a film scenarist, or producer. In fact, it is the teacher, educator, or curriculum developer who will prepare language materials, make out procedures, and determine teaching/learning strategies for the CALI lesson. The scenarist language teacher has a great role in writing scenarios, rather than acting in the 'scene.' If a CALI lesson is planned beforehand through a scenario, not only the scenarist will have a clear understanding of what the courseware will turn out to be like in the end but also the computer programmer will definitely be aware of and know what s/he will be doing. Furthermore, any correction, revision, or modification of the lessonware will be easier and more appropriate in due time.

Now another question might be posed: "How is it that the language teacher will write the scenario of a CALI lesson?" First of all, the scenarist should make some preliminary specifications about the courseware application to be created. These include a number of items which can be easily written down on a form. See Figure 1.

After the scenarist has specified the language course, the number of the lesson, and the time which is allocated for the students to finish the lesson, s/he need list and state the specific objectives of the lesson in terms of terminal behaviors. Second, s/he will identify the dialogs or reading passages that will be used in the lessonware as well as the characters and the role relationships between and among them. After these specifications, the context in which dialogs or passages are used will be mentioned so that their degree of formality and informality can be stated.

The last five items of the scenario form include language forms, language functions in a dialog, kinds of exercises, and procedures for evaluation and

A SAMPLE TEACHER'S FORM OF SCENARIO FOR A LESSON

Course	Name of the language course for which courseware will be developed.
Lesson	Specifying the lesson number.
Time	Determining the time for the lesson in terms of teaching hours or minutes.
Objectives	Stating the specific objectives of the lesson in terms of terminal behaviors.
Dialog or Passage	Identifying the dialogs and reading passages that will be used in the lesson.
Characters in the Dialog	Stating the number of characters or people and their status involved in the dialog and defining the role relationships between characters in the dialog.
Context	Stating the degree of formality of the language in the context of dialogs and passages, such as *formal, informal, polite,* and *friendly* and where the conversation(if any) takes place.
Structures	Listing of language forms and vocabulary items which are used in the dialog or reading passages.
Functions	Listing language functions such as *asking for* and *giving information, giving advice* and *permission*, and *asking for* and *giving directions* which are used in the lesson.
Kinds of Exercises	Listing and determining the kinds of exercises to be used in the lesson including *yes-no answers, cloze exercises, multiple-choice exercises, matching words* such as *synonyms and antonmys, true-false questions*, and stating the number of trials that will be allowed on error and the types of feedback that will be given to the student upon correct and incorrect answers to the exercises.
Presentation Procedures	Determining phases of the lesson (e.g., pre-reading, on-reading, post-reading activities for a reading lesson) on the computer; suggesting scenes and screen designs for the presentation and processing of the lesson through animations and visual effects.
Evaluation Procedures	Preparing questions for evaluation; stating the points which will be given to the student in the evaluation stage of the lesson, determining the types of reports to be given to the student and written for the teacher after the student has completed the test.

Figure 1: A Sample Form of Scenario for a CALI Lesson
(Based on Koç, 1990 & Öz, 1991)

presentation. After stating the language structures, for example, *may* and *can* and language functions like *expressing possibility* and *giving advice*, the scenarist should determine the kinds of exercises that will be used in the lesson. The exercises may include questions with yes-no answers, multiple-choice exercises, cloze exercises, matching words such as synonyms and antonyms, and true-false questions. In addition to the kinds of exercises that will be used in the lesson, s/he may also state the number of trials that will be given to the students when they make a mistake or choose the incorrect answer in the exercises, as well as the types of the feedback to be given to the learners upon their correct and incorrect answers.

Then, the scenarist, no matter who s/he is (a language teacher or curriculum developer), must, of necessity, determine some presentation procedures that will be followed in the writing of the lessonware on the part of the computer programmer. The details of presentation thus include the phases of the lesson (e.g. pre-reading, on-reading, post-reading activities for a reading lesson on the computer) with language materials such as passages, exercises, questions for the evaluation, etc. In addition, I think that the CALI scenarist ought to determine and suggest appropriate screen designs for presenting the lesson through pictures, sounds, graphics, animations, and visual effects.

As the last case in point, the scenarist should state some evaluation procedures to be followed in the CALI courseware. For example, s/he will need to specify the number of questions which will be asked in the evaluation part of the lesson and the points to be given for correct answers. Furthermore, I believe that s/he should also determine the kinds of reports that will be given to both the student who has completed the lesson and the teacher who is teaching the language course under question. Finally, the scenarist will deliver the computer programmer all the questions to be asked in the test, with the correct answers.

A SAMPLE SCENARIO

The sample form of scenario given in Figure 1 serves just as a preliminary outline of the lessonware. In fact, the scenarist will state further procedures for the presentation of a CALI material so that the computer programmer can write it on the computer as desired. I will now endeavor to illustrate a scenario for a reading lesson, looking through the eyes of a language teacher.

Course: Reading

Lesson: Lesson 3 (Chapter 3) in *Developing Reading Skills: Intermediate* by L. Markstein and L. Hirasawa.

Title of the Passage: *Ordinary Aspirin is Truly a Wonder Drug.*

Time: 60-80 minutes.

Although the duration of the lesson is not strictly specified, I suppose it will take approximately 60 or 80 minutes for the learner to finish the lesson. Moreover, I assume that the same reading lesson would take more than that in a normal classroom context if one takes into consideration the fact that the teacher would sometimes spend more time helping some students individually.

Specific Objectives of the Lesson: At the end of the reading lesson, the learner will be able to:

- Define the meanings of the new words introduced in the passage.
- Answer the comprehension questions related to the passage.
- Tell the synonyms and antonyms of the words in the passage.
- Understand relations between pronouns and their references in the passage.
- Find specific information in the passage.
- Interpret given information in the passage by 'reading between the lines.'
- Write a summary of the passage.

Writing a summary of the passage will not be done on the computer; however, it is assumed that after finishing the lesson the student will be able to do so.

Passage: The passage which will be used in the CALI lesson is given below.

"ORDINARY ASPIRIN IS TRULY A WONDER DRUG

Aspirin is the most popular drug in the world today. Many people take aspirin when they have a headache. It is effective in relieving other pains too. This article discusses the history and uses of aspirin.

1. Americans this year will swallow 15,000 tons of aspirin, one of the safest and most effective drugs invented by man. The most popular medicine in the world today, it is an effective pain reliever. Its bad effects are relatively mild, and it is cheap.

2. For millions of people suffering from arthritis, it is the only thing that works. Aspirin, in short, is truly the 20th-century wonder drug. It is also the second largest suicide drug and is the leading cause of poisoning among children. It has side effects that, although relatively mild, are largely unrecognized among users.

3. Although aspirin was first sold by a German company in 1899, it has beer around much longer than that. Hippocrates, in ancient Greece, understood the medical value of the leaves and tree bark which today are known to contain salicylates, the chemical in aspirin. During the 19th century, there was a great deal of experimentation in Europe with this chemical, and it led to the introduction of aspirin. By 1915, aspirin tablets were available in the United States.

4. A small quantity of aspirin (two five-grain tablets) relieves pain and inflammation. It also reduces fever by interfering with some of the body's reactions. Specifically, aspirin seems to slow down the formation of the acids involved in pain and the complex chemical reactions that cause fever. The chemistry of these acids is not fully understood, but the slowing effect of aspirin is well known.

5. Aspirin is very irritating to the stomach lining, and many aspirin takers complain about upset stomach. There is a right way and a wrong way to take aspirin. The best way is to chew the tablets before swallowing them with water, but few people can stand the bitter taste. Some people suggest crushing the tablets in milk or orange juice and drinking that."

(Markstein and Hirasawa, 1981: 27-37)

Context: As an authentic article taken from a magazine or newspaper, the article or text is written in formal English style.

Structures and Functions: In the reading lesson, the students will meet grammatical structures like *superlative degree of adjectives, passive voice, adjective clauses,* and *reduced adjective clauses.* At the beginning of the lesson, the student will be reminded of these grammar points if s/he wishes. In addition, some parts of the text include functions such as *expressing degrees of comparison, describing things and people,* and so on.

Kinds of Exercises: In the reading lesson, there will be exercises in the form of yes/no questions, multiple-choice questions, true-false questions, cloze exercises for vocabulary study, matching words such as synonyms and antonyms.[3] In addition, the user will be given appropriate types of feedback when s/he answers a question correctly or incorrectly. For example, various expressions will be exploited, such as *very good/congratulations/bravo, userName[first or full]!, You are doing fine* for correct answers and *that is not correct, userName!, Would you like to try again?,* etc. for incorrect answers in the exercises.

Presentation Procedures:

There will be many procedures followed in the design and writing of the lesson. These include mainly screen designs, animations, programming or scripting the structure and flow of the lesson, determining different types of feedback mechanism which will be used for correct and incorrect answers, preparing tasks for the students in the pre-reading, on-reading, and post-reading stages of the lesson, as well as the organization and presentation of the reading passage to the learners. Here I will cite only some of them without going into details. As a matter of fact, the procedures below will be actualized on the computer by the programmer.

1. Pre-Reading Activities

In this stage of the lesson, there will be a short dialog between the user and the computer.[4] In the dialog, the computer will ask the learner how s/he is, if s/he ever has a headache, and what s/he usually takes if s/he does. The focus will be on *aspirin*, which is related to the title of the reading passage. In addition, the learner will also be given an opportunity to revise the related grammar points in the passage (passive voice, superlative degrees of adjectives, and adjective clauses).

Here I would not like to give details of the conversation that will take place between the user and the computer nor will I explain how each grammatical item should be introduced to the student. A very important case in this point, however, is that the way each linguistic structure is likely to be illustrated must necessarily written down in order that the computer programmer (who is probably not a language teacher) will figure out how this stage will be computerized.

Another issue is that the learner might be asked to make some predictions about the content of the reading passage—as most language teachers frequently do so in the classroom. Therefore, the student, shown the title of the reading passage, may be asked the following questions:

— *What do you think the passage is about?*

— *What do you think the writer of the passage feels about aspirin?*

These two questions are merely examples, so the scenarist can suggest and write other types of predictable questions so that the programmer will code them on the computer.

2. On-Reading Activities

The first on-reading activity may be the reading of the passage on the part of the computer. In other words, the computer itself will first read the whole text paragraph by paragraph, using digitized human voice. The text may be spoken by a (native) speaker and recorded on a tape cassette, and the recorded speech must be given to the programmer who will then digitize the human voice to use in the courseware.

After the passage has been voiced, the student will be directed to study paragraphs separately or as a whole. Therefore, four distinct activities can be suggested at this stage of the lesson. First, the student will be provided with a dictionary for each paragraph, or as an alternative, a large lexicon containing all the unknown words in the text. Second, there will be clues or hints about grammar points in the paragraphs. The following (taken from the first paragraph of the text) can be an illustration:

"Americans this year will swallow 15,000 tons of **aspirin, one of the safest and most effective drugs invented by man."**

As it can be seen, the phrase in **bold type**, contains reductions of two adjective clauses. These are **"aspirin, which is one of..."** and **"... drugs which (that) have been invented..."** Therefore, it will be most useful to provide such grammatical hints for the user.

Third, there will be questions about the paragraphs which might run as "What does this paragraph describe?" Using the first paragraph of the reading passage again, I would like to illustrate the way such questions may be prepared.

(i) What does this paragraph describe?

A what aspirin is made of.

B the good things about aspirin.

C the bad things about aspirin.

(ii) In this paragraph, ***it is an effective pain reliever,*** **IT** refers to:

A drug.

B pain reliever.

C aspirin.

As is seen, the first is an *inference* question, and the second a *reference* one. They do not ask the learner to find any specific information in the paragraph. Thus such questions will enable the student to 'read between the lines' as well as to develop her/his understanding of pronoun references.

In addition, the scenarist should also tell the programmer not only the types of feedback but also the number of chances to be given to the student on correct and incorrect answers. For instance, s/he may ask the programmer to give the student only one chance to think over on error, or s/he might well want the programmer to give no chance at all but mention the correct answer immediately after the student has missed it.

The final on-reading activity will be about spelling. The programmer will be given a list of the problematic vocabulary items, and using the capabilities of the computer s/he will enable the courseware application to ask the learner to write the correct spelling of those vocabulary items, such that the program will first utter the digitized pronunciation of the related item and the student will type the answer. Furthermore, the user will be given immediate feedback after her/his typing, and even a hint if necessary.

3. Post-Reading Activities

There will be miscellaneous activities in the post-reading stage of the lesson. These are inclusively comprehension questions, true-false questions, matching words for synonyms and antonyms, and exercises on word forms. Excluding some of them, I would like to give an example for comprehension questions and a list of the words to be used in matching antonyms.

(i) What does this article discuss?

A only the good things about aspirin.

B only the bad things about aspirin.

C both the good and the bad things about aspirin.

(ii) Match the words on the left with their antonyms on the right.

1. relieve	a. safe
2. increase	b. suffer
3. speed up	c. a great deal
4. dangerous	d. reduce
5. few	e. slow down

In the first example the user will be asked to choose the correct answer, and in the second example s/he will be asked to match the words on the left column with their antonyms on the right column. In both, s/he will be given feedback as to her/his correct or incorrect answer. In the second example, however, correctly matched words will remain highlighted until all the words are tried.

Up to this point, I have tried to illustrate some reading activities that will take place on the computer. Obviously, these activities cannot go beyond being examples. But it should be stated that all of the exercises, charts, diagrams, pictures, etc. will be given to the computer programmer who will write the courseware. As a matter of fact, the organization of such activities will depend largely on the imagination and creativity of the scenarist, i.e., the language teacher.

Evaluation Procedures: In the evaluation section of the lesson, the user will be asked 15 questions including multiple-choice questions and true-false questions. In addition, the test score will be calculated out of 100, and the computer will give 6.6666666667 points for each correct answer. At the end of the test, the program will round off the number of total correct answers multiplied by 6.6666666667 and prepare reports for the student and the course teacher. The student report to be prepared will inform the learner of her/his total score, and the programmer will also enable the computer to give the language learner appropriate feedback for her/his success and make suggestions for a score below the average.

The teacher report, on the other hand, will be more specific than the student report in that it will include the name of the student who has studied the lesson and taken the test, give the long time, the long date, the score out of 100, the number of questions which are asked in the test, the number of correct and incorrect answers, the time which has been spent answering the questions in the test, and moreover detailed reports for each question concerning the choice that the user has made while taking the test.

In order to write the evaluation part of the lesson according to the above description, the computer programmer will naturally enough require all the questions with choices. Above all, s/he needs the correct answers to the questions. Therefore, the scenarist will also hand the programmer all the questions with choices and correct answers. The following may serve as a simple example:

(i) A good title for this passage would be:

A Aspirin: The Most Popular Medicine in the World

B Why I Take Aspirin

C The Side Effects of Aspirin

(The correct answer is A.)

To sum up what has been said so far, scenario is an important component of developing CALI courseware; it is a documentation of the procedures to be followed in the lesson and includes language materials to be used in the presentation of a dialog, a reading passage, etc. Undoubtedly, it is the scenarist language teacher, educator, or curriculum developer that will document the activities for the lesson, regardless of whether s/he originates them herself/himself or chooses them from a course book for adaptation.

The CALI scenarist who has prepared the language materials is now ready to give the documentation to the computer programmer. The programmer has as much hard work as the scenarist does, for s/he will not only design screens with graphics and pictures but also write programming codes or scripts to develop the desired lesson as the scenarist wants it. The result will be a scene or stage throughout which the lesson will be running and the language learner will be interacting with the computer.

SCENE

After I have acted as a scenarist language teacher and have documented some materials for the lesson, I will now look through the eyes of a computer programmer and consider the scenic aspect of a CALI application. In doing so, I will confine myself to stating the procedures or steps to be pursued in programming, giving some sample programming scripts, and two output scenes.[5]

Predominantly, the following procedures will be traced in designing and programming the reading lesson on the computer:

1. Getting the student's name, necessary program variables, and other data to be used later in the flow of the program and preparing screen designs.

2. Writing specific scripts that will take the learner to a certain stage of the lesson according to the data asked for.

3. Designing screens, such as card and background layers, for each stage of the lesson.

4. Preparing and writing program lines or scripts that will be used to make ready the student for the passage in the pre-reading activities by asking questions, creating dialogs between the user and the computer, and introducing grammatical structures that are utilized in the reading passage.

5. Making the computer first read the digitized human voice and the user listen to and follow the computer. (Designing appropriate cards, screens and writing the required subprogram scripts.)

6. Preparing procedures and subprograms to make the student herself/himself read the passage.

7. Preparing activities and tasks for the student in the on-reading phase. (And their designs and scripts.)

8. Preparing help utilities for grammar and vocabulary for each paragraph of the passage.

9. Asking for specific information and questions related to each paragraph of the passage. (Also preparing scripts and programming lines to achieve these tasks.)

10. Preparing spelling questions for vocabulary items in the text.

11. Designing links between the on-reading activities and other stages of the lesson.

12. Preparing post-reading activities for the student: asking further comprehension questions about the passage; preparing True-False questions and cloze exercises for word forms, designing tasks for vocabulary matching for synonyms and antonyms, etc, all of which will need to be designed and scripted.

13. Designing links between the on-reading activities and other stages of the lesson.

14. Preparing questions for the testing on the computer.

15. Designing the questions in the form of multiple-choice, True-False questions, inferential questions, etc.

15. Preparing instructions for the learner to take the test.

16. Preparing reports of the test results for both the student and the teacher, separately.

17. Reporting and giving the student her/his success concerning the passage and the score that s/he has got in the test.

18. Giving feedback, making suggestions and comments for the student about her/his success.

19. Designing subprograms for the student so as to direct her/him according to the feedback which will be obtained from the test.

Generally speaking, the computer programmer will comply with the above steps so as to establish appropriate screen designs, or more specifically 'scenes' through which the students will be looking and studying.

Sample Scripts

1. To get the name of the student at the beginning of the lesson and to direct the flow of the lesson:

```
On openStack
   global userName
   ask "Please type your full name:"
   put it into userName
   go to card "userPreferences" of stack "Home"
   put userName into cd fld "userName"
   go to card "contents" of stack "Lesson ?"
   pass openStack
end openStack
```

2. To prepare the spelling activity (each vocabulary item is stored in the lines of an invisible field called "words", and there is a field called "userPar" where the related text is stored):

```
on mouseUp
   global userName
   hide bg fld "words"
   hide bg fld "userPar"
   get any line of fld "words"
   repeat until it is empty
      get any line of fld "words"
      if the number of items of it is 1 then
         put it into spellingWord
         put it into playWord
      else
         put item 1 of it into spellingWord
         put item 2 of it into playWord
      end if
      put false into SpelledRight
      repeat until SpelledRight
         play word one of userName
         wait 1 sec
         play canyou
         wait 4 secs
         play playWord
         ask ""&word 1 of userName&", please spell the
         word."
         if it is spellingWord then play excellent
            wait 2 secs
            put true into SpelledRight
         else
            if it is not empty then play hint
                               :
```

```
        wait 4 secs
        put spellingWord
        wait 1 second
        hide msg
        put false into SpelledRight
      else
        put true into SpelledRight
      end if
    end if
  end repeat
end repeat
show bg fld "userPar"
end mouseUp
```

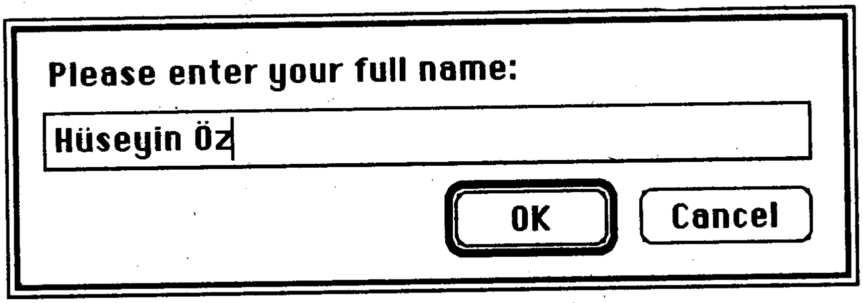
3. To enable the computer to voice the recorded speech of any paragraph (1, 2, 3, 4, or 5) during the on-reading activities, the following lines scripted in a button will yield the playing of the related sound resource.

```
on mouseUp
  global userName
  answer "This will take some time,"&&word 1 of
  userName&"?" with "Cancel" or "OK"
  if it ≠ "OK" then exit mouseUp
  get the short name of this card
  put it into isim
  if isim = "User1" then play paragraph1
  if isim = "User2" then play paragraph2
  if isim = "User3" then play paragraph3
  if isim = "User4" then play paragraph4
  if isim = "User5" then play paragraph5
  pass mouseUp
end mouseUp
```

Sample Scenes

The three scripts that are given above are just examples. The programmer will write hundreds of lines of scripts to produce the scenic aspects of the CALI lesson, and these program lines will in turn bring about the 'dialog' boxes and screen cards with which the students will be faced. I would now like to give two sample scenes that are likely to appear before the language learners.

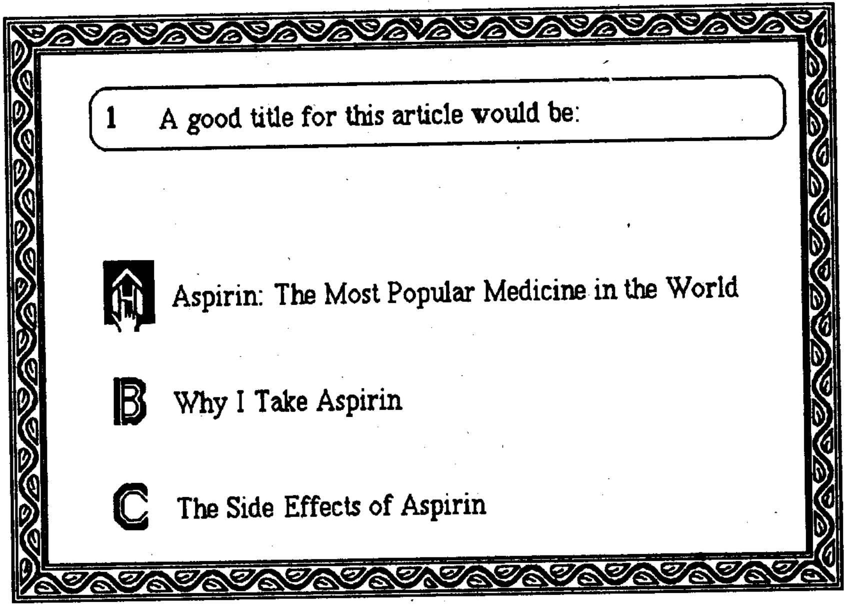
The first one illustrates some output scene of the first sample script given in **sample scripts** above, and it simply asks the user to type her/his name. The second example is related to the design of a screen card in the testing part of the lesson, and it shows how multiple-choice tests can be written on the computer. See Figures 2 and 3 respectively.



Please enter your full name:

Hüseyin Öz

Figure 2: The Dialog Box Used to Get the Student's Name at the Beginning of the Lesson



1 A good title for this article would be:

Aspirin: The Most Popular Medicine in the World

B Why I Take Aspirin

C The Side Effects of Aspirin

Figure 3: A Screen Card Illustrating the Design of Multiple-Choice Questions in the Evaluation Section of the Lesson

CONCLUSION

In developing CALI courseware applications, all in all, there should be a dynamic, running relation between the scenarist who is a language teacher or a curriculum developer and the computer programmer who may or may not have ties in education. The educator is the scenarist who is responsible for preparing teaching materials inclusive of dialogs, passages, a variety of exercises, and questions for the lessonware. Moreover, s/he must organize these materials in such a way that the programmer who is given a well-documented account of the course materials will make them into a courseware application. Here lies the teacher and programmer 'interface.'

NOTES

[1] Some parts of this paper are directly taken from my MA study, though there are considerable modifications. For more details and a scenario application on the computer, see my study (Öz, 1991: 63-112).

[2] Some people have described 'scenario' differently. Speaking of classroom scenarios, Phillips states that "scenarios are hypothetical but entirely possible, and are intended to demonstrate something of effect and value that may be achieved in typical teaching situations..." (1985: 37) Philips goes further and gives seven hypothetical scenarios. One of these reads as follows: "Two learners are seated at the computer. No teacher is present. Pairs of short phrases flash up on the screen, on the left, the other on the right. The learners each have a key to press; if they think the phrases are identical, they press their key. Whoever presses first and correctly scores a point..." It sounds a good scenario for a game; in reality, however, such tales can hardly serve for curricular purposes. Thus, CALI needs more realistic activities and procedures to achieve the language teaching goals.

In his *A Suggested Approach to Computer-assisted English Language Teaching at Elementary Level*, Faruk Türker (1990: 25-37) also describes and exemplifies scenarios. However, his CALI scenarios (seven in number) and his examples are contradictorily hypothetical in description and similar to those of Phillips' in exemplification.

[3] The exercises are also adapted from Markstein and Hirasawa, 1981: 27-37.

[4] The interaction between the student and computer is based on the notion of 'intelligence' of computers. Such an interaction is tried to be applied in Intelligent Computer Assisted Language Instruction (ICALI), a sister of CALI . Underwood (1989) mentions some features of ICALI in one of his articles in the field.

[5] The steps, sample scripts, and sample output scenes are in accordance with the Apple Macintosh and HyperTalk, the built-in authoring language of HyperCard.

REFERENCES

Koç, Sabri. (1990). A Model Approach to Creating Scenarios for CALL on Macintosh. Paper presented at **EUROCALL Conference 1990**, Kalgenfurt University, Austria, 7-9 September 1990.

Markstein, L. & Hirasawa, L. (1981). **Developing Reading Skills.** Cambridge, Mass.: Newbury House.

Öz, Hüseyin. (1991). **A Model Approach to Developing CALI Courseware for Teaching Reading in English on Macintosh.** Unpublished MA Thesis. Ankara: Middle East Technical University.

Phillips, Martin. (1985). Logical Possibilities and Classroom Scenarios for the Development of CALL. In C. Brumfit, M. Phillips & P. Shekan, **Computers in English Language Teaching,** Oxford: Pregamon Press, 25-46.

Türker, Faruk. (1990). **A Suggested Approach to Computer-assisted English Language Teaching at Elementary Level.** Unpublished MA Thesis. Ankara: Middle East Technical University.

Underwood, John. (1989). On the Edge: Intelligent CALL in the 1990s. **Computers and the Humanities, 23,** 71-84.