

**CALL:
DEVELOPING AN EFL GRAMMAR LESSON
FOR SELF-STUDY ON MACINTOSH USING
HYPERCARD AND COURSE BUILDER**

Hüseyin ÖZ (*)

INTRODUCTION

In an award-winning book of John Underwood in 1984, Earl W. Stevick, the eminent ELT methodologist and linguist, wrote the following short, but rather striking facts about language teaching in general and computer assisted language instruction in particular:

In recent years, two trends have caught the imagination of language teachers. One is the growing emphasis on communication -on using language for purposes beyond merely getting the right answer and pleasing the teacher. Communicative teaching concentrates on genuine exchange of information, on [language] games and other self-rewarding activities, and on lively simulations of real and interesting encounters. The second trend is *computer-assisted instruction*, which takes a piece of equipment whose IQ is exactly zero and puts it at the service of the human mind... (Underwood, 1984: xi):

As Stevick tells us with crystal clarity, computer assisted instruction (CAI) makes use of such a machine that it has no inborn wisdom and no inherent ability to act, learn or teach something by itself. Moreover, its intelligence quotient is not worth mentioning at all, for it cannot interact with users who do so sensibly with other people.

In spite of the fact that computers cannot communicate with their users with the sensitivity of a human teacher, computer users can interact with computers with fruitful results in the right environment. What is more important is that language teachers can be freed by computers from some of the routine classroom tasks of teaching such as explanations, illustrations, and exercises which are traditionally dependent upon books or tape recorders. Consequently, the language teacher, freed from

(*) Hacettepe Üniversitesi Eğitim Fakültesi Araştırma Görevlisi

such a routine work, will have more time so that he can cope with the functions that the computer cannot achieve (Ingraham, 1990: 27-28).

THE PLACE OF COMPUTERS IN LANGUAGE TEACHING/ LEARNING PROCESS

Should we language teachers also use computers in our profession, as mathematicians and engineers do so in their fields? If so; why, when, and how should we use these machines in the process of language teaching-and learning? These are only a few questions some people frequently ask pertaining the practical applications of computers in second/foreign language teaching.

For one thing, we strongly believe that computer applications are not-and should never be-confined to the disciplines of mathematics, engineering, physics, and so forth. In other words, it would be foolhardy not to think that social sciences, education and language teaching too, can benefit from advances in technology. Just as we have been using tape recorders and videos in laboratories, so is there no reason why we should not make use of this technological product in the humanities. Furthermore, it is an undeniable fact that computers not only "increase our students' proficiency in the language but are also effective for remedial and tutorial work for students who have special problems and need extra help" (Underwood, 1984: 94).

In considering the use of computers in foreign language teaching and learning, the place where the activities are carried out on the computer will probably be either the language classroom or the language laboratory. In either, case, there need to be computers situated in appropriate positions. Supplying the requirements for using the computers in the classroom, the language teacher can provide a variety of activities for the students working with the machines. To mention some of these, the following can be considered as computer assisted/aided language learning (CALL) activities in the context of class using computers:

- (i) Individual and group work including grammar practice, spelling and vocabulary;
- (ii) Games to motivate students, such as crossword puzzles and guessing games developing students' vocabulary knowledge;
- (iii) Problem-solving activities, like using simulation programs asking the student to play a role and to act as if he is in a real-life situation.
- (iv) Word processing activities for writing and composing on the computer.

At present, these classroom activities are among the most common CALL activities on the computer. In addition, according to the present literature (See Higgins and Johns, 1984; Underwood, 1984; Davies, 1985; and Robinson, 1985), most of the CALL programs used in these classroom activities are in the form of language

exploration activities. Although such programs "allow students to participate in the process of discovering certain aspects of a language" (Rubin, 1986: 13), there also need to be curriculum-based CALL programs suited to the needs of language students who need extra help or remedial teaching on certain aspects of language. Furthermore, it is necessary to take into consideration the content of the active curriculum while preparing CALL courseware for individual work or self-study on the computer.

DEVELOPING COURSEWARE FOR LANGUAGE INSTRUCTION

In developing CAI courseware for language instruction, or any other educational purposes, there are, at present, three alternatives to choose concerning the preparation of CAI materials:

1. using programming languages, such as BASIC and PASCAL,
2. using authoring languages, like Course Builder, HyperCard, and PILOT,
3. using authoring systems (or authoring packages,) like Word Attack, Story Board, and Versa Text.

Since programming languages are rather flexible and authoring systems quite inflexible, using authoring languages seems to be the best way to follow in creating courseware, especially for those who are new to computers and computer programming. Developed for non-specialist programmers as purpose-built programs, authoring languages have built-in commands and ready-made routines for designing courseware. Consequently, using an authoring language seems to be more appropriate for preparing CAI materials.

DEVELOPING A SAMPLE ENGLISH GRAMMAR LESSON FOR SELF-STUDY

This sample CALL courseware focuses on only one aspect of English grammar, that is modals, and it implements two hypermedia authoring languages: **HyperCard** and **Course Builder**. The former is used for presenting the student information on the screen, and the latter is used for providing the student with exercises to show him if and how he has acquired the information given him in the course of study. This exercise mode is for the student himself, and it does not necessarily involve testing. In addition, since the courseware is primarily for self-study, the student can work on the computer both in and out of class.

Objectives

Since we language teachers "cannot escape from the necessity of evaluating the progress of our students, the appropriacy of the materials used and the adequacy and appropriacy of the methods of teaching" (Koç, 1984: 65), we need to specify the ob-

jectives of our courseware beforehand, so that we will be able to enhance the teaching-learning materials presented in the program for the future development.

Taken as a whole, this program aims at providing the student with a summary knowledge of English modals with relevant language functions. Furthermore, it is expected that, at the end of the program, the student will be able to use each modal in various sentences correctly, stating the uses and language functions that can be realized with each modal presented on the computer.

Design of the Courseware

As it has been mentioned earlier, the authoring language HyperCard is used to present the student the information stored in its file called stack. At the beginning, the student is offered two options: *Go to Exercises* and *Go to Modals*. See figure 1. These two are up to the student, for he may first wish to take exercises first and come back to information about modals later, or the other way round. In either case, the student will have access to the information area.

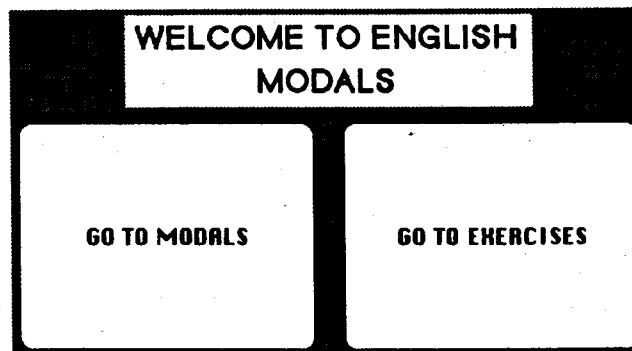


Figure 1

All the information in the program is given with sounds, texts, and graphics in screen windows called backgrounds and cards that include various fields and buttons. The student can have access to any modal or auxiliary at will throughout the program. As is seen figure 2, the student is given a list of the modals in the form of card buttons, and clicking the mouse at a button in this card which serves as the contents or main menu card will automatically bring and present him the information about that modal auxiliary stored on cards. For example, the clicking of the card button named may in figure 2 will take him to the other card on which the modal may be presented.

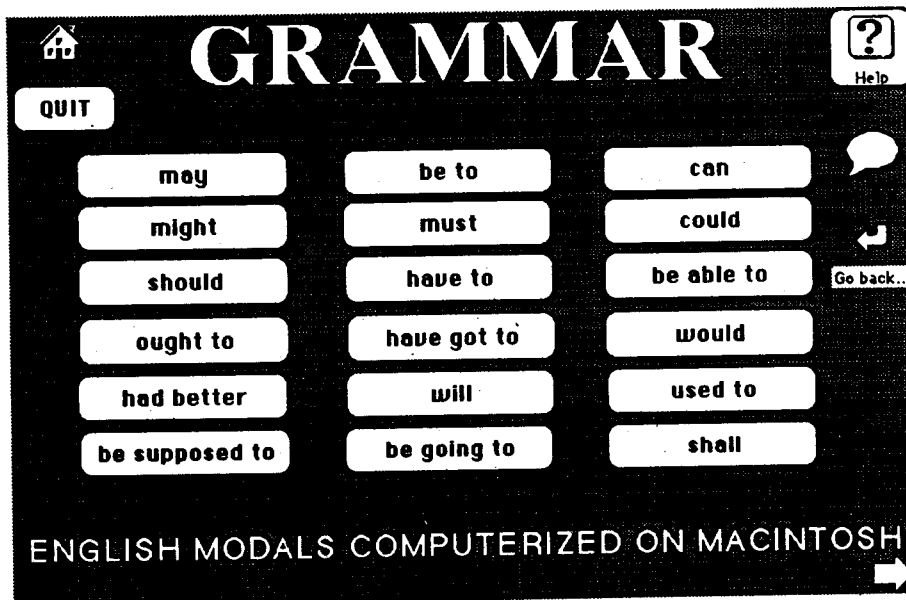


Figure 2

Furthermore, the card where *may* is introduced includes *explanations*, *functions*, *examples*, *past*, and *present/future* buttons as well as some utility buttons, such as directory, help, main menu, previous-next card, and find buttons. See figure 3. On this card clicking at the *explanations* button, for instance, opens up a field containing *explanations* about the uses of *may*; the *functions* button gives the language functions performed with the modal; and the *examples* button lists sample sentences in the other fields under the ones showing the language functions. In addition, the

pressing of the mouse on a field makes the computer read the text in that field through the use of a speech synthesizer called MacInTalk in the computer system. Alternatively, a digitized human voice can also be included in the stackware so that the texts in each field can be read by a human voice. This is really a very good feature of HyperCard as an authoring language that makes CALL courseware more enjoyable; however, it has been neglected in this courseware because of the cost and unavailability of hardware necessary for doing it.

Having worked on the modal *may* presented in the related card (figure 3), the student can go to either contents card by pressing the main menu button, previous or next card by pressing previous and next card arrows, or just click at find button to find any modal by typing it into a dialog box which the computer will offer. In brief, all the modals are handled in this fashion, and what makes the student access to any modal, information, and utilities is simply HyperCard's built-in language called HyperTalk, which is used in programming the courseware.

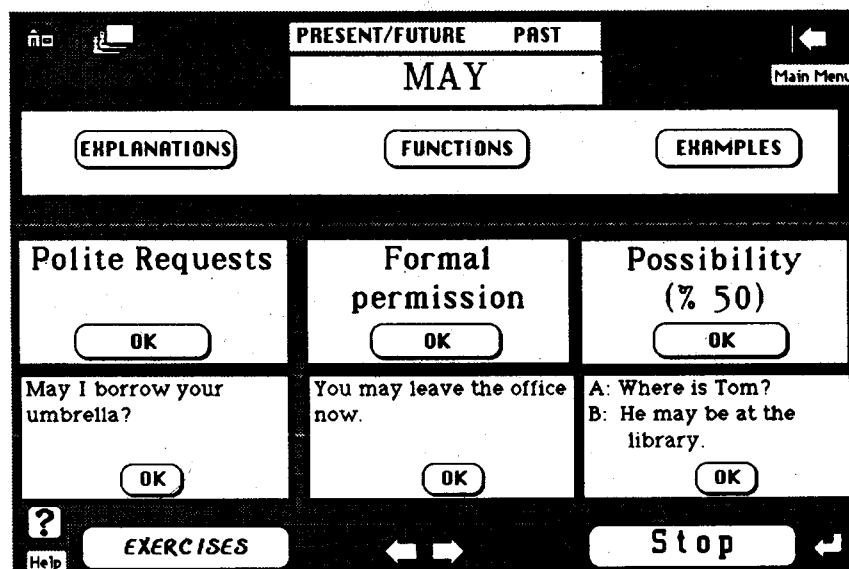


Figure 3

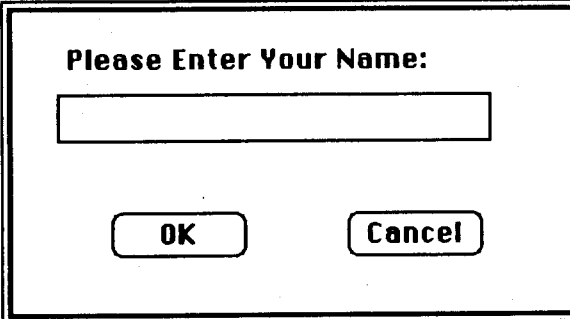
Exercises

When the student finishes studying the modals, he can go on to take the exercises at any time he wishes since there already *exists* an *exercises* button on each card. The exercises for the English modals presented in the HyperCard program are pre-

pared in Course Builder, another hypermedia authoring language. Actually, it is also possible to develop different exercises of true-false, multiple-choice and others requiring typing or matching. Because of its availability and time-saving feature, however, Course Builder is more (maybe the most of all present for Macintosh) suitable to program such exercises in.

Therefore, the exercises in this program are created in Course Builder since it is also possible to make bridges between a HyperCard stack and Course Builder standalone using menus and some linking tools available in the two authoring languages. The presentation of the exercises prepared in Course Builder is shortly explained below in the way the student encounters the questions and feedback given him through dialog boxes. For the structuring and organization of practice exercises and drills in detail, see Course Builder Manual by TeleRobotics International.

When the student wants to take exercises clicks at the *exercises* button in the HyperCard, he is asked to enter his name before studying exercises. See Figure 4. Though not essential, this information is later used to make a report file for the teacher about the success and progress made by the student who has taken the exercises.



The image shows a rectangular dialog box with a double-line border. At the top, the text "Please Enter Your Name:" is centered. Below this text is a single-line text input field. At the bottom of the dialog box, there are two buttons: "OK" on the left and "Cancel" on the right, both with rounded rectangular shapes.

Figure 4

After the student arrives at the dialog boxes presenting questions, he is asked to choose the best alternative and click at the OK button by the program. See Figure 5. At this point, the student's attempts to choose the correct answer are limited. Actually, the student is given only two chances to find the correct answer. If he chooses the correct answer, which is *may* in this example, he is given immediate feedback to encourage him for the coming questions. See figure 6.

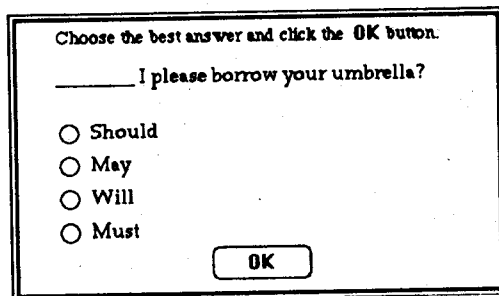


Figure 5

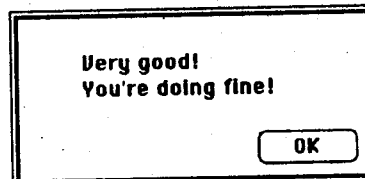


Figure 6

When he could not identify the correct answer, however, another dialog box, see figure 7, appears, politely saying his answer is not correct and telling him to try again. This try-wrong technique is again confined to two tries. If he chooses the correct answer on the second attempt, he will again be shown the feedback in figure 6.

If he still cannot find the correct answer, on the other hand, the program will apologize for not giving him another chance, wish him better luck for the next question, and present him another questions. All the exercises are arranged in a manner similar to the above. Furthermore, all the texts and questions in the exercises are speech-synthesized by means of MacInTalk system software.

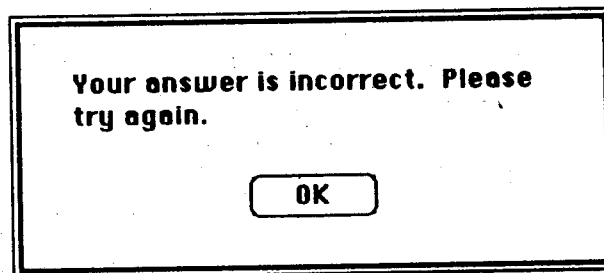


Figure 7

At the end of the exercises, the program tells the student that he has completed the exercises and asks him to either quit the program and shut down the computer, or review the English modals in the HyperCard stack. In short, this is how the exercises are organized in this sample program.

CONCLUSION

In recent years, the use of computers as learning-teaching aids has been alarmingly increased not only in ESL/EFL situations but also in other areas of education. As Chapelle and Jamieson (1986: 28) point out, there are three important, though inter-related, educational goals underlying this apparent fact: *individualization, record keeping, and answer judging*. Through the use of CALL materials for individualized instruction and self-study, students can work both alone and at their own pace. Moreover, poor students can be provided with additional practice and help. Although record keeping and data collecting features of CAI are for instructors, rather than for students, the answer judging capability of computers is highly advantageous to students since students are given immediate feedback as to their correct and incorrect answers and are presented explanatory information about any subject whenever needed.

Finally, the courseware explained in this article exemplifies the creation of an EFL grammar lesson for individualized instruction and self-study that can be used by intermediate (or higher) level students learning English. It actually illustrates a piece of CALL courseware that uses a hypertext system that allows students to choose any portion of text on the screen and open it up so as to get information, or any other related text without losing track of the original context (Underwood, 1989: 77-79).

BIBLIOGRAPHY

- Chapelle, C., & Jamieson, J. (1986). Computer assisted language learning as a predictor of success in acquiring English as a second language. *TESOL Quarterly*, 20 (1), 21-46.
- Davies, G. (1985). *Using computers in language learning: A teacher's guide*. Regent's Park, London: Center for Information on Language Teaching and Research.
- Higgins, J., & Johs, T. (1985). *Computers in language learning*. London and Glasgow: Collins ELT.
- Ingraham, B.D. (1990). Multimedia language learning. *ReCALL* 2, 27-28. Hull: CTI Centre For Modern Languages.
- Koç, S. (1984). Designing a reading course: A modular model. *Journal of Human Sciences*, III (2), 65-80.
- Robinson, B. (1985). *Microcomputers and the language arts*. Philadelphia: Open University Press.
- Rubin, B. (1986). Two approaches to CALL: Curriculum-based courseware and language exploration activities. *TSN*, 20, 13-14.
- Underwood, J. (1984). *Linguistics, computers, and the language teacher*. Rowley, Mass.: Newbury House.
- Underwood, J. (1989). On the edge: Intelligent CALL in the 1990s. *Computers and the Humanities*, 23, 71-84.