Recognition of Student Input in Computer-Assisted Language Learning

Carol Chapelle, University of Illinois at Urbana-Champaign
Joan Jamieson, Northern Arizona University

Available at: https://works.bepress.com/carol_chapelle/10/
RECOGNITION OF STUDENT INPUT IN COMPUTER-ASSISTED LANGUAGE LEARNING
Carol Chapelle and Joan Jamieson

Abstract
Computerized instruction has captured the interest of many educators as a means of individualizing language study for their students. The quality of this individualization is maximally dependent on the degree to which a computer can understand what the student communicates to it usually by typing a message on the keyboard. This article provides an overview of this student communication, or input: its types, its recognition, and some uses of its recognition. A general understanding of the potential of student-computer interaction will enlighten those who are examining Computer Assisted Language Learning (CALL) lessons for use in their curriculum. Some of this potential has been realized in the numerous foreign language lessons currently in use on the PLATO IV system at the University of Illinois at Urbana-Champaign.

One of the unique features of computer based education is interaction between student and computer. In order for this interaction to take place, it is not only essential that the student understand the messages the computer gives him but also that the computer understand what the student communicates to it. It is computer recognition of this student input that is the central topic of this article. First, different kinds of input that a student might enter into a computer lesson will be outlined. Then considerations involved in computer recognition of student input will be discussed. Finally, possible actions the computer might take, based on its recognition of the student input will be described. Through awareness of kinds of student input, strategies for recognizing it and an understanding of its uses, CALL lesson developers can work toward implementing better recognition of student input, thereby making computer interaction a more valuable learning experience for students.

Student Input
The Foreign Language lessons on the PLATO IV System use a variety of formats for instruction. However, one feature that they typically share is that they ask a student to respond to a question or a situation by typing something from the keyboard or touching the screen. Generally speaking, two kinds of input are common in CALL. Both types are student responses to computer requests: the student chooses one of several options presented or he produces an original language form within a context that is given to him. The length and complexity of this original language form differs from one lesson to another.

The FL lessons on the PLATO IV system in which the student is asked to choose from options presented on the screen are numerous and varied. Many traditional recognition exercises such as True/False, Multiple Choice, Matching and Vocabulary Identification are of this type. For example, a low level ESL vocabulary lesson uses this method (Figure 1) as it presents a picture of a dormitory room...
and asks the student to tap a touch sensitive screen where, for instance, the bed is (Pech et al., 1979).

In a higher level lesson, a student is asked to read a paragraph and then to choose its main idea from four possibilities presented to him (Dollein et al., 1976). Some simulations and games use this same format of input from choices. One simulation in Spanish which makes use of both a microcomputer and a videodisc requires this format. "The first environment of this presentation simulates a visit to a Mexican town. Through the videodisc, natives address the student in Spanish and wait for his answer" (Schneider and Bennion, 1983). The program then gives the student several options from which he can choose an appropriate answer. His answer will determine whether he will be sent to a market, a bullfight or a hospital, for instance. PLATO provides a final example of this choosing format — a game of "Concentration" in Russian in which the student must choose boxes by touching them on the screen in an attempt to match Russian words to their English equivalents (Hart and Provenzano, 1973). Although these lessons illustrate a variety of methods of presentation, they share the feature of student input as a choice from options presented on the screen.

Other Foreign Language lessons call for the student to produce a language form within a context. This type of input may be a single work, a phrase or a whole sentence. On PLATO IV the most common type of FL lesson is that in which the student is asked to complete a sentence by filling in a word or phrase. For example, in an ESL lesson on modal verbs, the student reads "It's a good idea." and also "Tony _______ wash his clothes. They're dirty." He then types "should" or "ought to" to fill in the blank (Pech et al., 1976). There are also numerous lessons in which the student is asked to write a complete sentence. Both transformation and translation exercises often request the student to type an entire sentence. Another common sentence writing exercise uses an audio device to have the student listen to an utterance and then type it. Typically the entrance defines the upper limit on the length of student input because of the complexity of having the computer recognize and process long expressions.

**Computer Recognition of Student Input**

In order for some communication to take place between the student and the computer, it is necessary for the computer to recognize the input it has received from the terminal. The ease with which this recognition takes place depends largely on the nature of the input. When the student simply makes a choice from several possibilities presented on the screen, the program has the relatively simple task of recognizing which one of the options was chosen. For example, in a multiple choice exercise in which there are four choices: a, b, c or d, the programs can easily recognize whether the student has typed none of the four, or if he typed one, which one it was. On the other hand, a program that requests a student to type in a word, phrase or sentence has a larger task in recognizing exactly what the input is. This difficulty arises from the fact that in such a context there is a variety of acceptable and unacceptable answers that a student may type in; that is, the number of possible answers is much greater than in the "choose an option" type of presentation. In this situation, it is necessary for the program to recognize each of these possible student responses. While at the word level the computer might have to recognize synonyms and misspellings, a longer response string would require the computer to take into account syntax in order for recognition to take place.

On the PLATO IV system recognition of such input is accomplished through 1) the built in features of TUTOR, the programming language, and 2) the input recognition algorithms provided by the lesson designers. TUTOR allows the programmers to specify correct as well as anticipated wrong answers, words that can be ignored, synonymous forms, and whether capitalization is required in the student's response (Sherwood, 1976). Also, the system can identify possible misspellings of a word in terms of missing, extra or inverted letters, as well as some word order problems such as missing or extra words (Tenzcar and Golden, 1972). These built in features of TUTOR allow the author to write a program that will recognize a number of close student answers from each one he specifies in the program.
Although the language can in many cases help to recognize a variety of forms that student input might take, it cannot always identify syntactically malformed sentences, nor can it determine which student response requires a particular message. This type of recognition goes beyond the built-in features of the TUTOR language; consequently, it requires the lesson designer to plan a way for the program to recognize the anticipated answers of the students so that they may be responded to appropriately. At the word level this often takes the form of "error recognition,” (Pusak, 1983) or looking for a particular word that is known to be possible input at a certain part of the program. For longer expressions, this method is less appropriate. Longer expressions can be better dealt with through the use of parsers (Markosian and Ager, 1983) which look at the syntax of a sentence as well as individual words.

**Action Based on Recognition of Student Input**

After the student's input has been recognized by the program, an action can be performed based on this recognition. The action can be an immediate response to the student, or it can be the storing of some information in the computer or both. In many drill-type lessons, the immediate action taken is often in the form of information given to the student indicating whether his answer was correct. If it was incorrect, and the program was able to recognize it, the computer can also give the student an appropriate message. This message can be a single sentence on the bottom of the screen or it may consist of a help sequence in which the student has access to remedial information. The program may also respond to student input by providing the answer to his question or request for information. This request for information might be as an aid to understanding something in a lesson or as the goal of the lesson itself. As an aid to understanding, sometimes a single keypress will signal that the student desires assistance; such is the function of the HELP key on PLATO. In a series of French translation lessons on PLATO (Marty, 1981), six keys like the HELP key can be chosen (see Figure 2). In lessons in which student-computer communication is the goal, principles of artificial intelligence are used to create a small world in which the student can ask the computer a question or tell the computer to do something, and in essence a dialogue can take place (Higgins, 1983).

Another action that the program can take based on a student's response is branching to enable a student to go to another part of the lesson or to another lesson. Thus, by allowing each student's input to determine his path through a lesson "the program accommodates students who need just a brief review as well as those who are learning the material for the first time" (Smith and Sherwood, 1976). The PLATO IV system, a mainframe computer, stores all lessons in one central location making an online index and this kind of branching based on student input possible.

As the computer responds to the student in some way, it can also be collecting information about the nature of the input that the student produces. On the PLATO IV system, commands can be programmed in lessons to "permit automatic collection of exact responses, certain standard statistics or, more generally, author specified information" (Hart, 1981). This information can then be used to inform teachers and students of progress. This use of the computer as a homework grader and reporter of performance can be added to by utilizing the computer to process this information. By processing information concerning students' problems and progress, the program can construct a performance profile on each student. The French lessons on PLATO IV take advantage of this capability by counting and classifying the errors made by each student (see Figure 3). This data is then shown to a student whenever he requests it (Marty, 1981).

The performance profile produced by the computer could be used not only by the student and instructor but by the computer as well. That is, information about the student's past performance can be used to determine which parts of a lesson or which lessons he should be assigned in the future. In this way the computer uses its recognition of student responses over a long period of time to learn something about the student and can therefore make better judgments about the lessons that will be beneficial to him in future. These types of CAI individualization strategies as they currently exist are unsophisticated in view of their potential (Kearsly, et al., 1983).

**Summary**

Clearly then there are many things that the computer can do in response to the student after it has "understood" what he has typed in. The sophistication of the action taken by the computer depends largely on the degree to which it can identify what the student has entered.

In order for the computer to respond in an intelligent and helpful way, it is necessary for the program to determine whether the student's input matches what is expected and if it does not, the program must then be
able to recognize how the answer deviates. Adequate recognition of student input is a simpler task in a case where a student simply chooses from options. However, it should also be noted that not only are there a limited number of lesson types using this format but also that these, for the most part, develop recognition skills. Input that consists of language produced by the student is more complex to recognize, but through the use of special features of a programming language and sophisticated programming strategies more complete recognition can take place. Once the computer can recognize the student's input, meaningful messages can be give, paths for remediation can be presented, records can be updated, responses can be classified, and ultimately computer-student interaction will be better suited to the various needs of individual students.

REFERENCES


Pusack, J. 1983. Answer processing and error correction in CALL. System 11, 1:53-64


