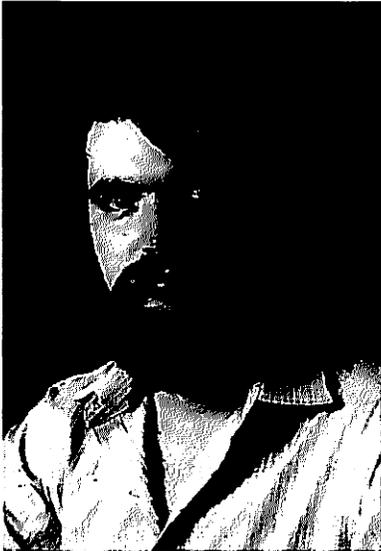


A HOME MADE PACKAGE FOR USE IN THE LANGUAGE LAB

Roger Kenner and Griff Richards



ABSTRACT

A computer-Assisted-Learning (CAL) system was developed in-house for Concordia University's Language Laboratories. Students of English and French as second languages and those requiring remedial English training have free access to a variety of instructional materials through any of the university's standard computer terminals. Authoring by faculty is encouraged via template lesson strategies and programming assistance. Automatic recording of student performance, time, comments, and programme access provides feedback for ongoing evaluation and modification of materials. The system was written in CCL and BASIC as available on Concordia's Cyber 174 with a NOS 1 operating system.

INTRODUCTION

The Language Laboratories of Concordia University's Audio-Visual Department have initiated a Computer-Assisted-Learning (CAL) service to provide students of English and French as a second language with opportunities for individualized remedial instruction and practice and to provide an outlet for graduate student research into CAL methodology and courseware development.

To launch the programme, use was made of the following existing resources: a small CRT (video display) terminal, a modem, and a telephone line assigned to the terminal. Free computer time for academic support was provided by Concordia University's Computer Centre (operating a Control Data Corporation Cyber 174 under a NOS 1 operating system). In addition, as a model, twelve lessons in remedial English grammar, written several years earlier by Arnold Keller were provided.

We set as our objective the development of a control system (the software) that would allow for automatic, error free operation by people totally unfamiliar with computers while, at the same time, keeping useful records for graduate student research. Teachers of English and French as a second language, students of applied linguistics and education, and language lab personnel were asked to provide lesson material (the courseware). However, most of the above were unfamiliar with computer programming, so we developed a "Do-it-Yourself Computer Programme Kit" which outlined lesson formats within which the intended authors were free to write any type of lesson they wanted. When an author provided us with the presentations and drills written out on paper, lab personnel then did the actual programming. The control system was a procedure file written in CCL, the job control language for Cyber 174. All the individual lessons and other programmes are written in Control Data BASIC. Within each lesson format were certain obligatory procedures oriented towards our control system such as a procedure to obtain the student's name from the control file, a procedure to write the student's results on a permanent record, and a procedure to allow the student to leave the lesson before it was finished. Full editorial control over the lessons as they appeared on our system was retained.

After a year of trial-and-error operation, we have arrived at a control system which achieved our objective. The terminal, modem, and telephone are installed in a modular learning area operated by the Language Labs. A poster gives the student instructions on how to contact the computer. In addition to this access, the lessons are available from any of the nearly 40 public terminals within the university.

HOW THE SYSTEM WORKS

1) Once contact has been made and the student identifies himself as a language lab user, our control system is summoned and all further operation is automatic. The system brings material in from a storage

account as needed. All material is kept on a separate account for security purposes (to keep users from tampering with the lessons).

A short programme opens the student's record on the permanent file by obtaining his name, the date, and the time.

2) The control file causes an index programme to be presented. The index is a list of all the lessons available. The student is given a choice of topics and, within each topic, a choice of individual lessons. The student chooses the lesson he wants by typing its code. For example, the student sees:

CHOOSE FROM AMONG THESE TOPICS:

1. ENGLISH (SECOND LANGUAGE) LESSONS
 2. LECONS DE FRANCAIS (LANGUE SECONDE)
 3. GAMES FOR ENGLISH LEARNERS
 4. JEUX POUR APPRENDRE LE FRANCAIS
- TYPE A NUMBER (1-4) ?

If he chooses "1", then he sees:

ENGLISH SECOND LANGUAGE LESSONS
ESL1 VERBS, THIRD PERSON PRESENT
ESL2 PRACTICE WITH CONDITIONALS
ESL3 ADVERBS OF FREQUENCY
A. TYPE THE NAME OF THE LESSON YOU WANT OR
B. TYPE 'B' TO CHOOSE ANOTHER TOPIC OR
C. TYPE 'Q' TO QUIT
?

3) If the student chooses "ESL3", the control system then records his choice on the permanent file, fetches "ESL3" from the storage area, and runs it for the student.

4) When the student leaves the lesson, information on his performance is written onto the permanent file. If the student finished the lesson, the elapsed time and some sort of criterion score (its form depending on the individual lesson) are stored. If he leaves before the end of the lesson, he is asked why he is leaving. His answer, along with the elapsed time and the place in the lesson where he quit is recorded. This information helps the author of the lesson evaluate its effectiveness.

All the programmes are written so that possible student execution errors are trapped and acted upon in such a way that the computer will not suspend execution with only a cryptic error message such as "ILLEGAL INPUT AT LINE 2450". In the above case, a special error procedure would be invoked which would respond with "THAT IS NOT WHAT WE ASKED YOU TO TYPE; PLEASE TRY AGAIN". The computer would then replay that particular problem.

If the student manages to HALT operation of our control system with the all-powerful command "STOP", he finds himself on a special account called a "Transaction Account", in which his only option is to start our control programme again or to exit.

A more normal exit is provided by the index programme. If the student types "Q", the computer says good-bye and logs him off. A final note is written onto the permanent file so that we can determine the student's total time on the system.

The permanent file is added to at each step so that all the information on a particular student session is not lost if he leaves abruptly.

Within the index the student has the option of leaving comments about the system or about a particular lesson. These comments give us feedback on problems students encounter. This, in turn, helps us provide better materials for the students.

Quantitative evaluation of the system as a whole and of individual lessons can be obtained at any time from the student records. A tabulation programme exists which will tabulate data for the system, for a particular lesson, or for a particular student. A record of incorrect student responses on items for some lessons is kept to further help author evaluate and modify their material.

CONCLUSION

In conclusion, the Concordia Language Lab CAL experiment has opened doors to a cost and time efficient method of providing language students with easy access to remedial and second language practice. It has encouraged us to further our efforts for the future. In our long term development plans we foresee the use of microcomputers at the student level, interfaced with the main-frame computer. The large computer will store the lessons (translated into microcomputer BASIC) on its disk memory, control the system, and keep records. The individual lessons will run on the microcomputers. This will provide us with multiple modern language fonts, colour graphics, control of peripheral tape recorders, slide projectors, etc., and sound synthesis capabilities not currently available from the Cyber 174.

REFERENCES

- * Keller, Arnold, **An Experimental Comparison of CAL and PI for Teaching the Mechanics of Writing**, (Concordia University: Thesis for M.A. in Educational Technology, 1979)