



### CALL: THE STATE OF THE PROFESSION

Norman Johnson, Editor, *CALL Digest*

Think back with me. Ten years ago, in 1979, the microcomputer revolution was just breaking out. Up until that time, computer-assisted language learning had been the private domain of a few large research universities. Most of the CALL programs available were an outgrowth of the programmed instruction approach to learning, which was not exactly the accepted model of second language instruction, even at that time. With the sudden availability of inexpensive microcomputers many people, including teachers in a wide variety of settings, began playing around with these machines. Do you remember the first microcomputer you ever used? I was first introduced to microcomputers in 1980 and already by that time they were getting beyond the stage of being mere novelty toys. Practical applications such as simple word processors, spreadsheets and database programs were available, as well as some unsophisticated instructional software. Do you remember using the first version of Bank St. Writer or feeling excited about the educational philosophy of MasterType? Many of us didn't know what we were doing or why; we were merely experimenting with a new tool that we felt had classroom promise.

Today, in many ways that promise is being fulfilled. Expanded memory has increased phenomenally the microcomputer's power to manipulate language data. Local area networks and large screen projectors are overcoming many of the barriers to access to and efficient use of microcomputers with students. The fruit of grass roots experimentation has been mixed with careful research resulting in improved software for language learning. Schools are increasingly planning for the use of microcomputers through staff training, adequate budgeting and curriculum integration. (While it's true that not every school is experiencing this, yet more and more this is the case.) The examples of computer use at Harvard University and Union County College written up in this issue are not idealistic research projects but rather realistic models of what can be and is being done with CALL today. The basic understanding is that computers are powerful tools that can be used in a variety of ways to support instruction and that their use should be integrated wherever appropriate throughout the curriculum. There are many research questions yet to be answered about CALL, as Carol

Chapelle notes in her perceptive essay in this issue, but the state of our profession has certainly emerged from infancy, and, dare I say, come through the throes of early adolescence to the verge of maturity. We are certainly excited as we anticipate the developments of the 1990s in our new quarterly format as *CALL Journal*. In the meantime, enjoy this feast of thoughtful writing on the state of our profession in 1989.

### AN OVERVIEW OF COMPUTER APPLICATIONS IN THE HARVARD ESL PROGRAMS

Karen Price, Harvard University

Since 1981, Harvard ESL has been exploring computer applications and assisting instructors to incorporate technology in meaningful ways in the language classroom. Software and hardware are used for activities not otherwise possible. The starting point is always what the instructor already does. The technology supports the various ways instructors teach and serves pre-existing educational goals. Depending upon the objectives of the instructor, the technology can serve as discussion generator, lecture aid, intelligent electronic chalkboard, or arbiter, as it tests student hypotheses in simulation activities or in the manipulation of databases.

The diversity of computer applications at Harvard's Programs of English as a Second Language mirrors the diversity of the student population. The year-round programs welcome approximately 700 students per term from both Harvard and the greater Boston community. Students from more than 50 countries participate in ESL's various intensive and part-time English classes. The students have a range of educational and professional experiences, as well as a variety of incoming competency levels and motivations for studying English.

#### FACILITIES:

In 1984, ESL received an ACIS grant from IBM, which provided 16 IBM-PC's in an ESL classroom during a summer session. In subsequent years, students have had carefully prescribed access to sixteen workstations at Harvard's Science Center. A recent summer saw some 5,000 hours of computer use at the Science Center by ESL students and faculty. Experience has shown ESL that computer resources in the classroom as well

### Table of Contents

<i>Karen Price: Harvard overview</i>	1	<i>Carol Chapelle: CALL research in the 1980s</i>	7
<i>Marina Kolaitis: Putting computers in their place</i>	4	<i>Deborah Healey: AudioLab review</i>	10
<i>Vance Stevens: Give them what they want</i>	6		

use the various computer programs, a meaningful achievement in its own right.

Today, the IIE has a flexible networked system with a separate CALL lab and open language lab allowing the entire IIE student population at the Elizabeth Campus (700) to use CALL on a weekly basis. Approximately 80% of the classes are incorporating a segment of the curriculum. A computer equipped classroom and portable classroom computers are also used with classes in small group activities. Two large-screen overhead projectors are available to introduce students to the various CALL and writing process lessons in a whole class setting, especially in large classes of 25 students or more. Thus, with this uniquely designed curriculum and laboratory setup, students are better able to reinforce their language and cognitive skills as well as enhance their knowledge of computer concepts and operations on a regular basis.

With this established program, Union County College's Institute for Intensive English now serves as a model for other ESL programs in New Jersey.

Project ESL Author lesson plan units are available to ESL programs at large. Dissemination will commence at the beginning of 1990. For further information contact:

Marinna M. Kolaitis, Project ESL Author Director, Institute for Intensive English - Union County College, 10 Butler St., Elizabeth, NJ 07202. Tel: (201)965-6031.

PROJECT ESL AUTHOR  
LESSON PLAN FORMAT

1. Level: 061
2. Syllabus Prerequisite: Verbs of Urgency: Indirect Speech
3. Unit Title: AGING
4. Program: SEQUITUR  
RHUBARB  
READING STRATEGY      Text Title: SOCIAL SECURITY-DIALOGUE  
SOCIAL SECURITY  
ROOMMATES FOR THE ELDERLY
5. Related Materials:  
VIDEO - "COCCOON"  
COMPOSITION
6. Suggested Lesson Plan:
  1. Give out composition topics; the students should be prepared to write on one of the topics in class the next meeting
  2. Watch the movie "Cocoon" up to the end of the disco scene (approx. 1 hr.) Discuss the main characters, retirement homes and nursing homes.
  3. The next day, write in class.
  4. Finish the movie.
  5. Go to the computer lab. Students do SEQUITUR (90% or higher in order to move on).  
The SAME DAY (if possible), do the RHUBARB. The RHUBARB program is a retelling of the SEQUITUR, so it's important to do it the same day, if possible.
  6. The following week, do the READING STRATEGY.  
Before doing the program, discuss the title.  
Have the students do the program.  
For homework, give them the READING STRATEGY cloze to fill in at home.
6. Approx. wks/cycle: 4th or 5th week of 061  
(lesson plan by Dorothy Burak)

REFERENCES

Hubbard, Philip. 1988. An integrated framework for CALL courseware evaluation. CALICO JOURNAL 6 (2).

**MAYBE WE SHOULD GIVE THEM WHAT  
THEY WANT ...**

Vance Stevens, Sultan Qaboos University

When the cinema camera was first invented, early directors staged plays and filmed them straight on; it was only through gradual experience that the cinematographer's art produced effects that took advantage of the unique and powerful properties of the medium. In *Mindstorms*, Seymour Papert used this example to illustrate how courseware developers with minds deeply rooted in linear, textual modes of lesson delivery had transported book-like exercises to computers and so missed potentials inherent in that medium.

Software developers now have a much better understanding of how computers should be programmed to exploit those aspects of CALL environments that lend themselves to exploratory language learning, and we rarely encounter these days the bookish, wrong-try-again sort of software that many of us cut our teeth on but have since abandoned. That is to say that CALL software developers have over time discovered much about what computers can do to promote learning and are exploring novel ways to appropriately use them. But what of the users of our products? Has their sophistication with computers kept pace with ours, or do they fail to grasp the significance of what we are doing for them? Although answers will differ from one situation to another, courseware developers should at least be asking these questions.

Four years into running a self-access CALL lab at Sultan Qaboos University in Oman, I am struck by the indifference shown to our advanced CALL facilities by a large proportion of entering freshmen. For the most part, our students encounter computers hands-on for the first time in their lives at our university, having never had opportunities in earlier schooling for exposure to computer-based discovery learning. Although we have many inquisitive and clever students at our university, for those whose minds are rooted in bookish, wrong-try-again modes of study, our computer lab with its impressive databases and tools for exploring these might as well contain windows on runes from another planet.

We have tried various means of informing our students about the scope and purpose of our lab setup; for example, we have given orientation sessions, conducted semester-long obligatory and guided "self-access" sessions, and provided mini-courses on exploratory software within other English courses. The computers are mobbed at orientation sessions, and the students look forward to using them if given the opportunity in class. They even come with some regularity to true self-access evening sessions. But they seem mainly interested in games, nevermind the content. They try out the exploratory tools, but often without grasp-

ing the philosophy behind them. Despite our efforts to tell them, they often fail to see the (often direct) relationship between the databases and the courses they are studying, and rather than devote the sustained and concentrated effort to these tools that might yield benefits, they prefer to jump purposelessly from one item of software to another, as one might idly change channels when watching TV with nothing better to do. Even though I might personally sit with students and patiently walk them through our most powerful programs, explaining in terms they can understand why we designed them the way we did and how they can use them to improve their English, the reaction is often a polite five or ten minutes with the software, and then the inevitable question, "Do you have anything that teaches grammar?"

It's a question that rankles, but one that is repeated often enough to require attention, and possibly even a carefully considered answer. At the most recent TESOL Conference, Claire Bradin raised a few eyebrows with her talk on a possible role for drill-and-practice in computer labs. I myself have suggested in print and more than once that one should at least try out some drill-and-kill software that teaches a language one is trying to learn before tossing the concept out entirely. Furthermore, recent indications are that some students may be predisposed to deductively-based courseware just as others may acquire a bit of a target language through simulations or adventure games. So why don't we have any software that "teaches grammar?"

Of course, we do! What do you think these exploratory software packages are supposed to do — I mean, if you just sat down and gave it a chance ... !!? At least that's what I would like to say in response to this question. But have you ever tried to distill for a group of computer-naive ESL students in the amount of time you have before they start looking furtively at their watches the philosophy behind the approach you have chosen to impart to them some competence in the language they are supposed to be serious about learning? To do justice to the attempt risks inducing glazed faces and blank stares.

The most important consideration is why they ask the question in the first place. I think it is simply that they resist modes of study different from the one they were brought up on, and although one goal of university training is to promote inquiry learning, it will take time before the old ways are extinguished in favor of more productive ones. Therefore, the proper answer to the question is probably to try and inculcate the philosophy behind discovery learning until the eyes start darting toward the wristwatch, and then go and get them a grammar lesson. I think that most of us have, when we were students, experienced wasting time in one course of study or another, blithely ignoring what someone was trying hard to teach us and even failing to seek ways of learning it on our own, only to at some other time in life gravitate toward the same subject because it has taken on a new relevance to us. Accordingly, it is unrealistic to expect every student who walks into a computer lab to suddenly absorb a predilection for methods of learning that we think works best there. The best we can expect is that such a predilection may evolve over time. A self-access CALL lab needs to have something for every student, and for some of these students, that

something is a straightforward grammar lesson.

I'm not suggesting that we revert to modes of delivery that replicate lessons found in books. I am saying that some students want the computer to teach them something, and we software designers must on occasion remember to bring our heads in out of the clouds and meet learners on their own turf. We who have been growing the most comfortable working with computers over the years are perhaps in the greatest danger of racing ahead of our audience, who may need a little time to catch up. Meanwhile, we should be giving them a little of what at least appears to be what they want, and designing it cleverly so that it teaches content in a tangible sense, yet in a way that will expose learners to the unique benefits of computer-based media, whet their appetites for more, and lead them gradually into the modes of inquiry learning that we enlightened developers think they should be pursuing. Easier said than done!

Meanwhile, I think I may try relabeling some of my discovery programs Grammar Lesson I, Grammar Lesson II, and so forth. Who knows, it may work!



## CALL RESEARCH IN THE 1980s: SETTING THE STAGE FOR THE 1990s

Carol Chapelle, Department of English, Iowa State University

CALL research, like other second language acquisition (SLA) research, attempts to assess factors related to students' second language learning. The specific methods used to make those assessments have been modified over the past decade as both CALL developers and second language researchers refine their understanding of their fields. These refinements become apparent when we examine the assumptions underlying the CALL research question of the 1970s: "Is CALL effective in improving students' second language competence?"

The question, although stating precisely what teachers want to know, is based on four assumptions which have been challenged throughout the 1980s and must be addressed by the research of the 1990s.

First, use of the term "CALL" in the question assumes that all CALL activities are the same. Perhaps all CALL activities are unique from other language learning activities, but are they all the same as one another? This monolithic view of CALL, although inaccurate even in the 1970s, has been invalidated repeatedly by both theorists and practitioners throughout the 1980s. In theory, Papert's (1980) conceptualization of the computer as an environment to explore rather than as a teacher to instruct marked the