Teaching Languages With Computers

The State Of The Art

EDITED BY

Martha C. Pennington

ATHELSTAN

2 A Direction For CALL: From Behavioristic to Humanistic Courseware

Vance Stevens

2.1 Introduction

Computer-assisted language learning, more commonly referred to as CALL, is a field whose recent development has reflected the exponential rate of change inherent in other aspects of computer technology. This change has been taking place against a backdrop of equally significant shifts in perceptions of how people learn languages, and in perceptions of who language learners are and what their individual needs and differences are. The shift of attention to individual needs of students, an important theme in this chapter, is taken up as a focus of CALL research in the next chapter.

This chapter characterizes the methodological stream of change as a factor affecting current approaches to CALL software development. In particular, I will examine how CALL software has, from its roots in behaviorism, evolved more humanistic and communicative applications to language learning.

2.2 Computers as Behaviorist-Based Teaching Devices

The first efforts of any significance in the teaching of languages using computers occurred during the late 1960's (see Ahmad, Corbett, Rogers and Sussex, 1985, for a thorough treatment) and for more than a decade, where computers were used in language learning, it

was usually in ways designed to structure learning along the lines of behaviorist models then in vogue. Accordingly, **programmed instruction** (PI) was considered the optimal model for courseware design. Lewis (1981, p. 48) observes that because CAI was used originally to speed up the process of PI, it was resisted by teachers and "has not been a huge success." Also, in Rivers' (1981) treatment of CAI in the context of the many ways then considered effective for teaching foreign language skills, PI is the only modality considered. Indeed, Rivers' definition (p. 45) of "effective courseware" (i.e., "a learning sequence which is carefully designed and executed") would not characterize all courseware considered effective in language learning today.

The salient flaw in instructional algorithms based on behaviorism was the assumption that learning could be reduced to its lowest common denominators, and that teaching could thus most effectively proceed as a series of pre-planned discrete steps. Because programming is also a discipline in which the steps in a given task are clearly defined, it was tempting to conceptualize early CALL efforts along the lines of the behaviorist models. Drill-and-practice became the preferred (and the most effective, according to Vinsonhaler and Bass, 1972) mode of delivery, and this mode managed to stay in vogue even after pattern practice had fallen out of favor in language classrooms. Even then, it was thought that drill-and-practice software could somehow satisfy the students' need for the sustained, tedious (i.e., boring) kind of teaching that many teachers preferred not to do in class. Unfortunately, this did not prove to be the case. Although computers are often able to amuse first-time users for long periods of time no matter what the software, the novelty wears off, and student users do not, of their own accord, spend any more time with redundant and repetitive courseware than they do with the books on which such courseware is based.

Besides its failure to stay current with methodological trends in language learning, CALL courseware has typically exploited few of those aspects of the medium holding greatest potential for use of computers as learning tools (e.g., inherent recursiveness of operations or the ability to randomize). True, many of the computer's special attributes have been utilized with drill-and-practice courseware, such as its repetitive capability, its capacity to evaluate responses and branch accordingly, and the privacy and immediacy of

its feedback (see Ch. 1, "Language," for discussion of these capabilities). These are valuable assets in an instructional medium, but they are not impossible to achieve in other media (a programmed text, for example). The situation was aptly characterized by Papert (1980a) as being analogous to the first efforts of film makers, who essentially staged plays and filmed them straight on, only gradually evolving the special characteristics and techniques which make experiencing a movie very different from watching a play.

Educators have likewise come to realize that computers, like Hollywood cameras, are best exploited in ways that take advantage of their particular characteristics rather than when they are used to try to "improve" deliveries in the media they seem to be replacing. In fact, cinema cameras and computers were only following on stage plays and books, not replacing them at all; plays and books have valid places in culture and education, and what they are good for need not be replaced. Thus, attention to computers in language learning is most wisely focused on those attributes which are most likely to uniquely facilitate that process.

2.3 Computers as Facilitators for Humanistic Learning

With the advent of the "personal" microcomputer in the late seventies, the realm of computing has come in a few short years to spread far beyond the walls of buildings containing mainframe computers. As a result, a wide spectrum of educators, many among them innovative language teachers, have been in a position to deal constructively with deficiencies in the software available to them. The majority of these teachers had long since revised many of the notions on which behaviorist-based courseware, such as drill-and-practice, were based, adopting instead more humanistic approaches to language learning. Before going further, it will be useful to examine what such approaches entail.

Over the past decade, there have been shifts in emphasis in language teaching from form to function and from product to process, with a corresponding shift in perceptions of students from their being learners (through teaching) to acquirers (through discovery) of language. Krashen has been a major influence, hypothesizing (1982) that with enough concentration on communication, grammar will

take care of itself (i.e., will be acquired in natural order by an assimilation of linguistic data processed from comprehensible input). Consequently, language teachers have been looking more to sources of language in the environment and less to materials specially prepared for language learners.

Going on the assumption that learners might themselves learn language better than teachers can teach it, increasingly educators have put emphasis on the process of learning (as in the approach to writing described in Ch. 5), with suggestions that practice can best be generated by providing numerous activities involving real communication and a rich source of language data. The result has been a move towards learner-centered educational environments where the teacher becomes more a facilitator than a purveyor of knowledge disseminated from the head of the class. Thus language learning is thought to best be accomplished when language learning environments are non-threatening situations which make students feel at ease while giving them a responsible role in their own learning. Toward this end, teachers have been attempting to enlist students themselves as resources and informants, forming them into groups, getting them to speak to each other, write for each other, and to interact with each other in selecting options and responding to realistic challenges.

It would not be far-fetched to assume that a language teacher agreeing to many of the above statements might happen to observe a group of students who seemed to be enjoying a particular activity on the computer which was clearly not drill-and-practice. The question that should be forming in the teacher's mind is this: What is it about this activity that makes it so appealing, and how can I incorporate those factors into my own software development and/or utilization so that students will *want* to use the software to facilitate their own learning of a target language? I am about to suggest an answer hinging on three broad principles regarding the selection and production of humanistic CALL courseware: the principles of **intrinsic motivation**, **interactivity**, and **eclecticism**. Aspects of these three principles are represented schematically in Figure 1.

Intrinsic Motivation

- relevant and risk-free learning environment
- learning incidental to some other activity
- opportunities to use language in problem-solving
- multi-modal materials

Interactivity

- adjustment according to profile of individual student
- creation of environment facilitating interaction with computer
- creation of environment facilitating interaction with humans

Eclecticism

- creative adaptation of software designed for other audiences and purposes to classroom use in language learning

Figure 1. Principles for the Production and Selection of Humanistic Software

2.4 The Software Should Be Intrinsically Motivating

According to Moore and Anderson (1969), many cultures have evolved ways to impart learning through the use of games which are intrinsically motivating and relatively free of consequences, yet which are taken seriously by participants. Papert mentions two such activities in his book, *Mindstorms* (1980b, pp. 178–179): learning to hunt by "playful imitation," and learning Carnival dances at socially oriented Brazilian "samba schools." Papert aimed to create similarly motivating, yet risk-free, learning environments on computers. He called such environments, which students were encouraged to explore in order to discover how to function there, **microworlds**. Borrowing artificial intelligence techniques from Winograd's SHRDLU (1972), Higgins (1983) was among the first to apply this concept directly to language learning in his program GRAMMARLAND (see Ch. 1, "Lan-

guage"), which endowed the computer with a mini-language and an ability to operate within that language (though the microworld concept was already inherent in a variety of simulation and adventure games).

Stevick (1982) has noted the superiority of "the quality of learning that is incidental to something else we are trying to do" over that which "takes place when we focus our attention only on the items to be learned" (pp. 131–132). It is in thus subtly distracting the learner that computers are especially effective. In microworld mode, the computer is being used to provide game-like opportunities to use language and to act in conjunction with that language. How those opportunities are exploited, whether on the spot or in some consolidation activity later, depends, as with other media, largely on the imagination of the teacher. Before reaching the point where they must step back and take account of what they have assimilated, students are afforded the opportunity to enjoy language in a pleasant and non-threatening way, and that enjoyment may even carry over into the more thought-provoking consolidation later.

One singularly motivating aspect of computers exploited by microworlds is the challenge of figuring them out. Computers are impeccably logical, and the inherent logic in a problem or task can often be elucidated from available data. Thus computers can present puzzles which students, alone or in groups, can work to solve, taking into account the available information and filtering this data through the computer and/or one another to arrive at the rule governing the computer's behavior. Often, rules arrived at in this way can be tested, and hypotheses confirmed or rejected, according to whether the computer responds in ways predicted by the induced rules. Put to such use, computers become tools for discovery, and what is discovered can be something about the language being studied.

Another motivating quality of computers is their ability to incorporate and accommodate other media. For example, graphics and animation, used creatively but in moderation, can enhance explication and retention. Furthermore, computers interfaced with videodiscs, voice digitizers, audio or video cassettes, or other audiovisual media, can compound the educational potential of these media (see Ch. 1, "Inputting and Outputting Sounds and Video Images," for more detail). As Rubin (1984) puts it, "multi-modal materials tend to attract and hold student's attention to a greater degree [with]

enormous implications for increasing learning" (p. 33).

Used in a "subtly distracting" way, as learning tools and facilitators (as what Marty, 1981, called "allies," and Higgins, 1983, called "pedagogue"), rather than as excuses for exercise books (Higgins' "magister"), and appropriately interfaced and programmed, computers can be compelling. Computers are indeed compelling when they satisfy human needs or desires such as the desire for novelty and challenge. Other needs of language learners which computers can help to satisfy are the needs for responsibility, options, and opportunities for communicative interaction. All of these features are likely to make courseware more intrinsically motivating to language learners.

2.5 The Software Should Be Truly Interactive

Interaction with a computer can be either unobtrusive or overt, the former taking place without the student's knowledge, whereas the latter simulates communication. Although truly viable (some would say plausible) communication is not presently available with computers themselves, computers do facilitate interaction with other communicative humans (see Chs. 1 and 6 for further discussion), and this latter attribute is taken by many to constitute their greatest potential in language learning.

Interaction with a Computer Can Be Unobtrusive

Interactivity has commonly been a feature of CALL programming. Typically this has meant that the program branches according to its author's anticipation of a certain student response. However, the fact that branching features can achieve greater sophistication than this type of anticipatory function has been supported by recent research on individual differences in students (as treated more fully in Ch. 3), a decidedly humanistic development.

In one such study, Chapelle and Jamieson (1986) report that, although students who had been tested to have a certain learning style disliked CALL on PLATO, one factor may have been the approach to CALL taken in the lessons themselves, which, like most CALL, was "notoriously 'insensitive' to individual learner differences" (p. 41). It

follows that we may be doing some students an injustice by making them all work through the same lessons. Accordingly, computer programs could determine a student's learning style and then deliver a lesson appropriate for that student. Coupled with other media, options for individualization multiply. Rubin (1984, pp. 31–32) calls the videodisc a "superb vehicle" for developing learner profiles and then tracking students, "depending upon the student's learning style, language level or modality preference."

In another project considering individual differences in courseware development, Dalgish (1985) conducted error analysis studies on students of various languages and then produced generative CALL lessons which individualize tasks according to the student's first language. Still another consideration of individual differences in courseware design is pointed out by Johnson (1985), who cites evidence that girls may prefer cooperation in learning math while boys prefer competition. Clearly, the field would benefit from other studies isolating further areas where CALL can be individualized.

Providing Overt Interaction with Computers

Providing options within a lesson is crucial to viable CALL. Moore and Anderson (1969), in their discussion of clarifying educational environments, emphasize the importance of learners' being able to shift at will between several perspectives on learning, an idea that was to an extent validated for CALL by Stevens (1984). Given the existence of numerous options, computers can provide the ultimate in open-access, individualized instruction. Options can include instantaneous access to HELP panels, hints, and perhaps even solutions.

Because computers can frustrate by appearing obtuse, by failing to respond to what seems logical to the learner but was unanticipated by the programmer, there should be uncomplicated ways of moving around in the program. Whether to review a past section, advance to another, skip a frustrating problem, or simply to preview the material, where to go in the program and how to use it once there should be up to the user. Furthermore, there should always be a convenient total escape from a program. The best means of escape would provide an option to save the current state of the program, so that the learner could return to that point if desired. This is not always practical on personal computers, but at the very least, no program should make a learner feel that it is necessary to resort to

switching off the system.

Feedback and other transactions with students can be presented randomly from a data base. This can be made to appear communicative, as in Johns' (1981) suggestion to store response components in chunks, presenting fewer chunks on subsequent passes through the program. In this way, the computer appears to become more familiar with the student, as would happen in normal conversation. For example, it might ask at first "Would you like to try again?" or "Do you care to have another go?", reducing (and interlacing) these responses on subsequent passes through the program to "Like to have another go?", "Care to try again?", "Another go?", "Again?", and so on, but varying the language even more with the addition of other sentence parts stored in the program.

Indeed, the question of communication with a computer is one of the most intriguing in CALL. A classic test for artificial intelligence, known as the Turing Test, puts a person in conversation with two devices, one of which is driven by a computer and the other by another human. If it cannot be distinguished which device is operated by the computer, then the computer is said to have passed the Turing Test. Whether this is possible at all with CALL software is not certain, nor is it certain whether it is necessary (compare the perspective of Ch.1, "Natural Language and Artificial Intelligence"). There is a need to determine whether or not a rich matrix of comprehensible input is possible with computers, and to learn more about the effects of "computerese" on language acquisition.

Communication in CALL is more often talked about than implemented. However, some interactive video projects, notably Montevidisco (Gale, 1983), achieve a high degree of simulated communication, and Kramsch, Morgenstern and Murray (1985) report on a project involving advanced parsing techniques to negotiate with students in a number of appealing ways. Underwood's (1984) methodological base and thirteen premises for communicative CALL are well conceived and often cited, but many of his examples of communicative software (ELIZA, for example), while perhaps of value in language learning, would fare poorly on the Turing Test. Articles like Barrutia's (1985), in which expert systems and their potential for communicative CALL are discussed, provide ideas for the future, but little of substance for the present.

Given developments in parsing and artificial intelligence (e.g.,

Addams, 1985), the quality of communicative interaction with computers will continue to improve, and it is possible that the communicative software available now meets communicative needs adequately (and certainly better than these needs are met with other media). However, it may be in putting students in touch with each other and with native speakers that the potential of CALL is best realized today.

Interaction with Other Humans

One need that people have in common is a need to communicate. This need is essentially what language teaching is all about, and computers happen to be very good at facilitating communication (a function explored at greater length in Ch. 6). Johnson (1985), on surveying a number of people active in CALL throughout the United States, found that "computer activities can serve as a catalyst that brings students together to interact, negotiate meaning, and negotiate strategies related to the task at hand," and that "peer and small group work centered around a computer-based activity can be a powerful force in a second language development program" (p. V-5). In addition, she notes "positive social effects of instructional work centered around computers." That group work has a beneficial effect on second language learners has been substantiated by Long and Porter (1985).

Discovery learning necessitates use of the language to communicate the discovery, thus creating situations ideal for socialization. Small groups often form spontaneously around a computer, and they can be convened more formally for deeper discussion of a problem or simulation. In such cases, students will have more than an artificial need to communicate; they will have real information to share, and may even be spurred to improve their reading and communicative skills in English specifically to be able to cope with and impart information related to such activities (as evidenced in Taylor, 1986a).

The most universally used communications software is word processing software (see Ch. 5, this volume; also Daiute, 1983, for an excellent characterization of how word processing facilitates writing). Such communication is facilitated by the fact that for many users, writing on a computer is itself intrinsically motivating. Marcus (1983) has coined the term *videotext* to describe text that, in flitting about the screen, takes on many of the appealing qualities of

video. Videotext may be a factor in motivating writing within the context of a composition assignment, and when used communicatively, videotext facilitates collaborative efforts. Marcus suggests, for example, that students exchange computer screens so that one can comment on the other's writing while it is in progress. Daiute (1985) suggests other forms of collaborative writing, particularly of plays, and in use of the computer as a dynamic blackboard for group revision. Collaborative uses are also described by Friel (1985), who stresses that, when exploiting courseware in language learning, one must consider that "what at first seem autonomous exercises may, if successful, be combined to form parts of a more complex classroom exercise" (p. 37).

Other configurations may place students in direct communication through satellite or telephone links; PLATO, for example, allows conferencing across oceans, and Crookal (cited in Dunkel, 1986) has had students engage in internationally played simulations. Other forms of computer-based communication, such as electronic mail and bulletin boards, tend to elicit spontaneous communication even from non-native speakers. Alternatively, the text and graphics capabilities of computers can be combined in appealing ways (see Ch. 4 for discussion in relation to reading software), encouraging students to produce greeting cards, newspapers, yearbooks, and other more or less ambitious documents of which they can be proud.

2.6 Eclecticism: Going Beyond CALL Software

Much has been written to the effect that there is little available in the way of CALL software, or that what is available is disappointing. This may be true, strictly speaking, in that software made specifically for an audience of language learners is frequently found lacking. But if one is looking for authentic text meted out in manageable quantities and used in situations which appear natural, then the criterion for "language learning software" broadens.

Often, educational software designed for native speakers has an editor with which lexis can be changed, and there is much courseware which, though not designed specifically for education, is rich in language that foreign or second language learners will be motivated to learn in order to participate in an activity or in the discussion

that might follow a session with the software. Baltra (1984), for example, has documented successful communication and learning in a language class based on a commercial adventure game that engages students and teachers as equal collaborators in trying to find the solution to a mystery. Similarly, Taylor (1986b) relates experiences with a "sophisticated" commercial simulation game in which "much of the vocabulary is new, but students have little difficulty learning it" (p. 12). Similarly, public domain software, while lacking in some production features of its commercial counterpart, often comprises unique and entertaining programs useful to language learning. Since there is usually free access to the source code, public domain programs are infinitely more adaptable to one's own language learning or teaching situation (for suggestions, see Stevens, 1985 and 1986).

Software that offers training in problem-solving and higher-order thinking skills is of particular interest from a humanistic point of view because it lends itself to collaborative work by the students, who must manipulate the program—e.g., by creating some kind of simple machine or figure—to achieve a certain goal. Pogrow (cited in Johnson, 1985) found that limited English-proficient students who used such software had 50% more friends than those in a control group, possibly because of increased opportunities to interact with peers and native speakers, and increased confidence in their cognitive competence.

Johnson (1985) concluded from her survey that "the use of a computer as a tool to accomplish functional tasks has far greater potential for second language learning than traditional or even communicative CALL" (p. III-5). She suggests that the study of language *per se* on computers be "a by-product when focusing on tasks related to both social and academic success in school" (p. III-6). Thus one might employ science lab or economics simulations as CALL courseware. Another excellent but often overlooked exploitation of functional software in language learning is programming languages, and there is at least one textbook (Abdulaziz, Smalzer, and Abdulaziz, 1985) for teaching language through the medium of programming. Similarly, skills such as typing, word processing, use of writing aids, and spreadsheet and database manipulation, can be taught as part of what is really a language course (see Barlow, 1987).

2.7 Conclusion

There are numerous features of computers that are uniquely exploitable in language learning, many of which were not utilized in behaviorist-based courseware but which have begun to appear as CALL development has adapted to current trends in language teaching methodology. Current uses have placed the computer in such roles as linguistic informant, game partner, a means of getting a message out to a variety of people, a tool, or even a drill master if that is what students want. And some do. Before entirely abandoning drill-and-practice software, teachers should give it a try with a foreign language they themselves want to learn. They can then form their own assessment of the efficacy of drill-and-practice, based on insights gleaned from a learner's perspective.

Language teachers who have applied the principles of intrinsic motivation, interactivity, and eclecticism in their selection and development of CALL courseware have begun to see that computers, used in a variety of ways, can attend to individual differences among learners and take on roles supportive of humanistic language learning. Adherence to these principles, and discovery of new ones through research and classroom practice, will constantly improve the ability of computers to facilitate the language learning process.

SERIES IN COMPUTER-ASSISTED LANGUAGE LEARNING

Teaching Languages With Computers
The State Of The Art

The first integrated volume on computer-assisted language learning (CALL), *Teaching Languages With Computers: The State Of The Art* , is a highly readable overview of the theoretical and practical aspects of the use of computers in language instruction. The book is designed to be accessible to those with minimal background in computers, while also providing valuable information for the experienced user. It includes seven comprehensive chapters written by specialists in the field of language education and computer-assisted instruction. The contributors include:

- **Carol Chapelle**
- **Susan Young Dever**
- **Joan Jamieson**
- **Carolyn J. Keith**
- **Peter Lafford**
- **Martha C. Pennington**
- **Marianne Phinney**
- **Vance Stevens**
- **David Wyatt**

The Editor

Martha C. Pennington, who is on the faculty of the University of Hawaii at Manoa in the Department of English as a Second Language, is a recognized specialist in language education and research. She has given workshops and presentations on CALL for several years at regional, national and international conferences.

Teaching Languages With Computers provides an excellent overview of recent developments, current usage and the future potential of computers in language education.

ATHELSTAN

$23.95 Hardcover ISBN: 0-940753-11-1
$12.50 Softcover ISBN: 0-940753-12-X