ED 255 170 IR 011 456

AUTHOR Hughett, Harvey L.
TITLE Ir' duction to Computer Aided Instruction in the La: age Laboratory.
INSTITUTION Idaho Univ., Moscow.
SPONS AGENCY Offic of Postsecondary Education (ED), Washington, DC.
PUB DATE [84]
GRANT G00-82-1497
NOTE 88p.; Eight articles from "Electronic Education" and "Aramco World Magazine" have not been included due to copyright restrictions.
PUB TYPE Collected Works - Serials (022) -- Guides - Non-Classroom Use (055) -- Reference Materials - Directories/Catalogs (132)

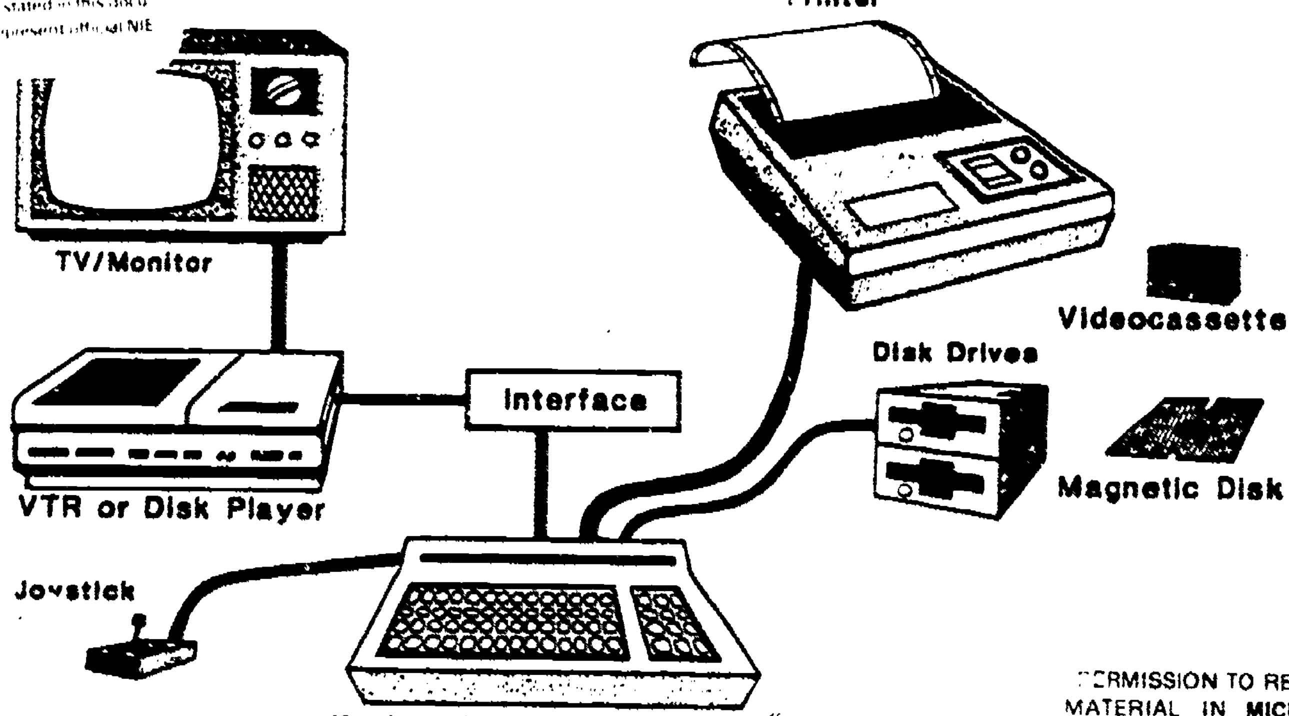
EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Autoinstructional Aids; *Computer Assisted Instruction; *Courseware; Higher Education; *Language Laboratories; Language Teachers; *Microcomputers; *Program Development; Purchasing; *Second Language Instruction; Second Language Programs; Videodisc Recordings
IDENTIFIERS Interactive Video

ABSTRACT

The first half of this book focuses on the rationale, ideas, and information for the use of technology, including microcomputers, to improve language teaching efficiency. Topics discussed include foreign language computer assisted instruction (CAI), hardware and software selection, computer literacy, educational computing organizations, ease of operation, keeping up with CAI developments, microcomputer reliability, interactive video, timesharing mainframe computer versus microcomputers, and research on CAI effectiveness. The five supporting articles reproduced here are: (1) "Drill Programming Consideration" (James P. Pusack); (2) "A CALI Glossary for Beginners" (Randall L. Jones); (3) "Computer-Based Foreign Language Instruction in Illinois Schools" (Robert L. Blomeyer, Jr.); (4) "Montevidisco: An Anecdotal History of an Interactive Videodisc" (Larrie E. Gale); (5) "Stringing Us Along: Programming for Foreign Language CAI" (Sue K. Otto and James P. Pusack); and (6) "Using Microcomputer Word Processors for Foreign Languages" (Kim L. Smith). A description of the Computer Assisted Language Instruction Consortium (CALICO) and a 38-item reference list are included. (LMM)

Title IV, HEA
G00-82-1497

INTRODUCTION TO COMPUTER AIDED INSTRUCTION IN THE LANGUAGE LABORATORY



BY

HARVEY L. HUGHETT

Funding Provided by the Department of Education

Edited and published by the: Department of Foreign Languages and Literatures
T/A Harvey L. Hughett
University of Idaho
Moscow, ID 83843

This material is based on research supported by the U. S. Department of Education, Office of Postsecondary Education, CFDA No. 84.017. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and in no way constitute endorsement nor necessarily reflect the views of the U.S. Department of Education.

Special thanks is given to Electronic Education, Aramco World Magazine, James P. Pusack, and the CALICO Journal for permission to reprint articles pertinent to the theme of this booklet. Original material in this publication may not be reprinted without the written permission of the appropriate publisher.

Printed in the United States of America

PREFACE

We live in exciting times! New technology presents language laboratories with challenges for change. Small computers are emerging in increasing numbers on university campuses, in a largely unplanned fashion, with largely inadequate technical and user support. Small computers come in a wide variety of architectures and sizes, and their users request an amazing variety of services. The purpose of this booklet is to supply information to the neophyte language instructor or lab director as to how technology, including the microcomputer, can be used to improve the efficiency of language teaching. Each day thousands of dollars are spent for new technology that will not produce the results desired by the purchasers. Thanks to a timely grant from the Department of Education, we have been able to gain valuable experience in using technology to help solve our teaching problems. The surprises, successes, and setbacks that we have experienced are reflected in the following pages.

Practically everyone who gets involved with the new technologies are surprised at the ancillary support associated with computing. While hardware may be important to the success of any program, appropriate and good quality software is critical. The writing of new computer programs is more labor intensive than most realize. It is disheartning to speak with those who are so anxious to get into computing that they purchase hardware before fully researching the availability of courseware, expandability, etc. The microcomputer that you buy today may not be in existence tomorrow, or may not be upgraded to accept improvements as they become available. Certain brands are strong in the areas of data processing and business management but may fail to meet requirements for serious computer assisted intruction(CAI), particularly for foreign languages. A good question to ask yourself as you consider ~~a computer~~ purchase might be: Could I afford to run this computer if it were given to me? The price of the hardware should not be the overiding factor considered.

It is difficult to predict just how any new technology will penetrate the educational establishment. The first school with a library was Aristotle's academy in 400 B.C., but although we have had inexpensive printed texts for hundreds of years, we still call what we do in the classroom a "lecture." Although the printed book has made a major penetration, it has not radically changed the basic educational process. The real payoff for computing in individual learning is still more potential than reality (Wegman, 1982).

Experience, so far, indicates that personal computing will make rapid penetration into the learning processes, and shows particular potential for revolutionizing language learning laboratories. A "revolution", to one degree or another, is inevitable and indeed will come. Whether the computer is universally accepted by language laboratories in the very near future is not sure. However, it can be predicted that those who conscientiously prepare now will have the advantage of being on the "cutting edge" of this exciting new field. Properly approached, the rewards can be great. Improperly approached, the problems and frustrations can overcome you.
Technology is our servant, ready to do our bidding. What shall we do with it?

TABLE OF CONTENTS

(CONTINUED)

SECTION II: SUPPORTING ARTICLES

*Deleted due to copyright restrictions.

INTRODUCTION

The role of many small university and college foreign languages and literature programs will continue to deteriorate unless means are developed to increase their "teaching reach" through continued development of technology and innovative instructional programs. At the University of Idaho, teaching loads are heavy and enrollments are light. Since the elimination of the language requirement in 1970, foreign language enrollments have shown a steady decline. Although students generally admit that foreign language study is of value, many have found our courses neither attractive nor relevant to their educational objectives. Other limiting factors include a variety of scheduling conflicts. Stated in general terms, we are faced with adapting our curriculum to coincide with the trends toward a career education <u>without</u> abandoning our necessary role in the humanities, and be able to do so with fewer teaching faculty.

In 1979 the State of Idaho passed the 1% property tax initiative and the effects on education are far from being over. Capital outlay and general operation funds practically have been frozen. The Department of Foreign Languages and Literaturs lost one instructional position last year, two this year, and is in constant fear of the possible reduction of another.

With these problems hanging over our heads, we have been sufficiently motivated to incorporate a variety of activities into our teaching program in order to improve teaching efficiency and increase overall enrollments. The challenge was to increase our "teaching reach" without compromising the quality of our teaching.

Since similar situations of financial exigency are experienced by many schools throughout the country, there is a genuine need to develop technology and innovative instructional materials that can aid instructors in multiplying their efforts, especially in areas where workloads or programs are being threatened.

Through the careful design and production of special auto-tutorial language laboratory courses, for credit, the Department of Foreign Languages and Literatures at the University of Idaho has been able to reverse the trend of dropping enrollments and in just five years the Laboratory accounts for 47% of all departmental enrollments. Since 1976 there has been an 11% increase in student enrollments for the university as a whole. During that same period of time there has been an unprecedented 172% increase in enrollments in the Department of Foreign Languages and Literatures. Without the enrollments generated by laboratory courses, enrollment increases would have amounted to only 25% for this same period.

We have redesigned the learning lab so as to offer the student much more flexibility than available in the conventional language lab. We have taken advantage of affordable products of advanced technology and find that, with proper research and development, they can make a significant difference in the efficiency with which learning objectives can be achieved. We are convinced that learning modules designed for slide/sound, video cassette, video disc, and particularly, the small desk-top microcomputer formats, promise to revolutionize language learning laboratories in both qualitative and quantitative terms without having to spend inordinate sums of money. Probably the greatest advantage of this use of modern technology, and perhaps a lifesaver for small foreign languages departments, is that such courseware can offer unlimited advantages in course tracking and scheduling. The number of course offerings can be increased, various languages, specialized low enrollment courses, personalized tracks of study, and "off-step" courses can be offered without a corresponding increase in faculty.

For the most part, our lab is quite conventional and rather modest. The lab is organized into three areas, a level III section allowing students to record their voices and compare them to native speakers, an audio-visual room equipped with synchronized slide/sound decks, video-cassette tape positions, and microcomputer-controlled learning stations. Both of these areas are connected to a third area, the lab control room where learning programs are copied and distributed to students through checkout windows.

Most lab courses are entirely auto-tutorial in nature and make use of special texts, reprints, mimeographed handouts, carefully structured syllabuses which rely on audio cassettes, video cassettes, synchronized slide-sound sets, and microcomputer floppy diskettes.

The material to be learned is subdivided into small units or study modules. Tests for each study unit are available in multiple versions. Each version consists of a random sample of questions drawn from a pool of questions.

The rules for a given course are spelled out precisely at the beginning of the semester. Students know exactly what they must do to end up with an acceptable grade.

Tests are repeatable. Students not satisfied with a test score can take the test over. Students are given another version of the test, and the new score replaces the old.

Students can check their mastery of material. A student wishing to know whether he or she has mastered a unit can take a trial test and receive immediate feedback about his or her performance. If the score is low, the student can restudy the material.

For the first time in more than twenty years a major new technology exists that promises to radically improve, if not revolutionize, language laboratory learning. There is a growing awareness among language department administrators that language laboratories and learning somehow are going to be affected by the latest microprocessor technology, although there is much confusion as to how this might take place. As prices for this new technology decrease and capabilities increase, more and more educators are considering using small desktop microcomputers to augment instruction.

If computer growth had happened to the automobile, it would now get 16,000 miles per gallon, cost 10¢ and be the size of the textbook.

Computers are dumb--they don't think. All a computer does--and it does it with amazing speed and efficiency--is manipulate pieces of information. The computer works only with information on hand, and it can only do with that information tasks it has been programmed to do.

In all likelihood, as the microcomputer "boom" gains momentum, there will be those who may feel that this technology is the long awaited remedy for all ills or difficulties associated with foreign language (FL) teaching. One does not have to go back very far to remember that when audio language labs were first introduced, it was believed that if students could listen to what they had said and compare it with what they were supposed to have said, that all problems in pronunciation would vanish overnight. Unfortunately, too much was expected of the lab and after companies had installed their expensive lab console systems, many educators patiently awaited the "linguistic miracles" that were to be performed. When the fantastic claims were never substantiated, the laboratory seems to have fallen into disfavor with many foreign language teachers.

As the number of learners and what they need to learn increases, learning strategies have to be modified. Most of our current educational methodology at all levels is shaped by large numbers. Thus, the lecture, the textbook, electronic recording, and broadcasting media are all mechanisms to disseminate knowledge to large numbers of people. All are essentially one-way media, with communication flowing from the instruction authority to the student; they provide little if any capability for the authority to respond to individual learners.

Something has been lost in this process. Teachers and students yearn for the situation where they can work togther closely. Foreign language educators almost all agree that the optimum setting for education is when one teacher works with one student or a small number of people.

The computer is the first technological innovation in education that enables us to "personalize" education, even with a large number of learners, to a situation similar to the Socratic dialog wherein the instructor queries students, asking questions that depend on the answers given to earlier questions, always framing new questions for the learner to react to.

We do not claim that a computer dialog, even when prepared by a group of excellent teachers, can emulate fully the Socratic situation but it is the only educational tool with which we can approach that situation with a large number of learners.

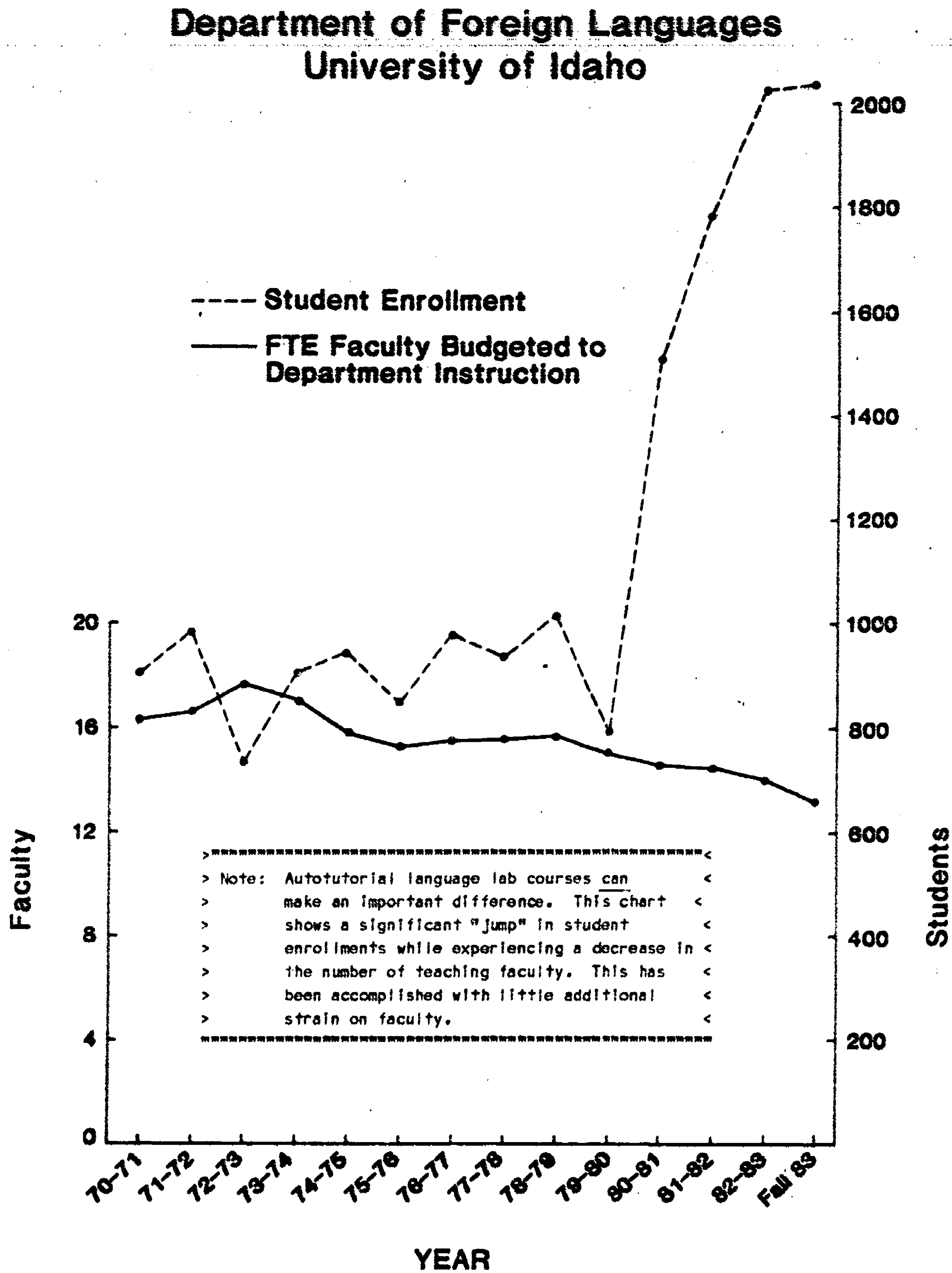
Although central time-sharing computers in recent years have proven the value of CAI; high costs, limited availability, lack of standardization in programs and languages, poor graphics, slow turn-around on heavily used systems, etc., have all tended to exclude the small schools from their use.

The microcomputer situation is changing rapidly. This new technology will be the major educational delivery system of the foreseeable future.

For the uninitiated, this clamor of public interest and sophisticated marketing campaigns can cause a mild case of hysteria. The problem for the educator is to figure out which computer is best, how much money to spend and how to go about training students and staff. Suffice to say, buying computers can be expensive. Surprisingly enough, buying the initial hardware may be the least expensive part. Purchasing software, upgrading systems, hiring programmers, providing training workshops and equipment maintenance are the real "killers." So, just as schools hire architects to plan new facilities, it is important to have planning expertise available for computing decisions. The field is moving quickly with new developments occurring almost monthly. Therefore, the school administrator cannot afford to remain at the mercy of the salesman.

Let me emphasize that the number of course offerings and enrollments can be increased through the use of technology. Various languages, specialized low enrollment courses, or personalized tracks of study can be offered effectively without a corresponsing increase in faculty. This does not infer that the technology would replace the teacher but simply assist and aid instructors to multiply their efforts, improve their effectiveness, and in a sense, increase their production with little or no more effort than with the traditional approach to teaching.

The time is right to give serious consideration to the development of [illegible]ized technology related to teaching processes and the production of su[illegible]e courseware to meet projected needs. It is now feasible for even small schools to provide AT teaching programs and even CAI.



BEST COPY AVAILABLE

PROBLEMS WITH MICROCOMPUTER FOREIGN LANGUAGE INSTRUCTION

Unfortunately, the microcomputer is very new as an educational aid and demands more knowledge and "tinkering" for effective use than most realize. For the untrained, a computer is like a foreign language and must be systematically learned. Even though microcomputers may be relatively inexpensive, they also represent a potential waste, if the user spends an inordinate amount of time trying to modify the equipment or software to his or her particular needs.

Potentially, the greatest problem associated with microcomputer assisted instruction is that expectations might be unrealistic. It is commonly accepted among informed educators that language laboratory learning hardware (regardless of whether it is a tape recorder or a computer) never comes close to its potential without properly constructed and supervised software materials that are judiciously integrated, by the instructors, into their classroom teaching (Molnar, 1977).

Other problems associated with microcomputer assisted FL instruction include:

General lack of instructional computing literacy among lab directors and teachers of foreign languages. Before instructors can decide whether the microcomputer can aid them in achieving instructional goals, they must understand first the capabilities and limitations of both the hardware and the software. Instructional computing literacy is a blend of knowing how and when to use computer programs, of being able to impart some degree of programming skill, and of feeling comfortable with computer hardware. Individual teachers must be able to determine the blend needed. At the present time, there is no organized effort to improve FL CAI literacy (Ricketts, 1979).

There is a pressing need for documented courseware. After a telephone survey of 28 universities already using some form of CAI and participation in educational microcomputer workshops, we have confirmed that micro software for serious FL learning is practically nonexistent. Unfortunately, programs designed for the larger "maxi" or "mini" computers are not compatible with the microcomputer, nor are they easily transferred to the micro format. This situation, however, is rapidly improving.

Existing instruction authoring programs often require specialized adaptations before being capable of displaying authentic foreign language characters. Even the widely distributed and flexible PILOT program would require alterations before the neophyte FL instructor could effectively use it. The PILOT author language is relatively simple and the basics can be taught to most groups of teachers within twenty hours. Even though PILOT is not non-threatening to English dialogue-oriented or mathematically-oriented teachers, it can be terribly frightening to the FL teacher who is required to incorporate interactive modes.

Information on how FL instructors and lab directors can use this new technology to increase their "teaching reach" without compromising the quality of their teaching is not readily available. Although studies indicate that CAI is at least as effective as traditional instruction for most students, in terms of achievement (Roecks, 1979), there will always be a percentage of students who really need the human interaction with the teacher. Not only do potential users of this format need courseware models and programs, but also a tutored awareness of capabilities and limitations of drills, instantaneous feedback, self-pace courses, testing sequences, and so on.

Conventional language laboratory teaching approaches tend to be monotonous and soon lose their ability to hold ones attention. In spite of this weakness, audio taped programs remain the mainstay of most FL laboratory programs although some attempts have been made to incorporate visuals through video and film presentations. The present state of the art FL laboratory fails to provide for an active and multi-sensory process of learning that stimulates the student to be involved (Taylor, 1979).

Although teaching loads are heavy (in excess of twelve hours), enrollments are light. The problem confronting the University of Idaho's foreign language programs is typical for departments throughout the country. With the curtailment, in 1969, of the foreign language requirement in the College of Letters and Science, the number of student credit hours taught by the department descended from 3650 to 2559 in 1972. Enrollments continued to drop until spring of 1979 when we registered an all time low of 2360 credits generated. In that same year, Idaho passed the 1% property tax initiative which drastically cut capital outlay and general operation funds and one instructional position.

Since similar situations of financial exigency are experienced by many schools throughout the country, there is a genuine need to develop technology and innovative instructional materials that can aid instructors in multiplying their efforts, especially in areas where workloads or programs are being threatened.

WHERE DOES ONE START?

Getting started is always a problem. Where should I go first? What should I look out for when considering purchases? How can I learn what I need to know? I hope that the enclosed materials will help answer these questions.

Having offered various computer workshops to foreign language teachers who showed an interest in the microcomputer, I am sorry to say that there are many who are overwhelmed by the sheer volume of information and hardware available and soon lose interest. The reasons for this are complex. I am convinced, however, that intellect is not an important factor, but that patience and making adequate time available for a systematic study are absolutely vital to any serious inquiry.

There are dozens of magazines and journals that can be helpful to you. For foreign language professionals, I highly recommend the CALICO Journal. Other publications that may prove helpful might be: Byte, Personal Computing, Popular Computing, Electronic Education, Technological Horizons in Education Journal, Association for Educational Data Systems Journal, The Computing Teacher, Apple Orchard, PC(for IBM PC), and so on. Also, it might be helpful to visit local computer stores to get an idea of what is available, although you would be much better off visiting a college or university that is actively using microcomputers in their educational programs. Too often, sales personnel have little knowledge about using this technology in teaching, particularly in the humanities. Do not get into too big a hurry. I wish that I had a floppy diskette for every computer that was purchased prematurely and later found to be unacceptable.

Read this booklet carefully. Identify your needs (You may find that computer aided instruction is not feasible for your particular situation at this time!). Contact others who are using micros to do what you wish to do. Listen carefully to their suggestions. A few phone calls can save you thousands of dollars and hundreds of hours. Weigh the pros and cons of different brands. Identify software sources, service contracts, mail order, compatibility, flexibility, etc. Don't be dazzled by frills, big names, or low prices. Be as sure as you can that you are making the right choice...the first time.

Obviously, you will need much more information than is contained here in order to make intelligent choices about educational computing. For those of you just starting, we feel that you may benefit from our experience. We are delighted to share these hints with you!

THE "HARDWARE JUNGLE"

The tendency is to "cut corners" where possible, because of the costs. As an example, some educators intend to use tape recorders rather than the more expensive disk drive for data storage. It is no secret that most of these people regret working with tape recorders for data storage and soon go to the more reliable disk drives. One drive may be fine for a student study position, but two drives for the instructor/programmer are not only desirable but often required. You may save money by buying a computer with reduced memory, however, many programs require at least 64K. In order to "cover yourself", in the event future programs require additional memory, you need the flexibility of increasing memory as you need it. There is a real temptation to purchase a so-called "clone" computer rather than the "real thing." In spite of what you may think, these computers are not always 100% compatible and there is no assurance that they will remain so as the major companies move to protect their interests. What appears to be "cheaper" today may not necessarily be so tomorrow.

Vendors may try to sell you a variety of add-on peripherals that you may not need. Generally, modems are not required in the beginning. A color TV monitor may or may not be important to your application. Some computers easily allow you to use foreign language character sets, while others may not allow this. You will not be satisfied with having to place accent marks to the side of letters requiring them.

Planning for the future of computers in language learning is, at best, risky business, particularly in a field where the half-life of hardware and ideas can be less than five years.

COMPUTER LITERACY: WHAT IS IT AND HOW DO WE ACQUIRE IT?

On many campuses, there is an unmistakable convulsion of enthusiasm for personal microcomputers and for large-scale instruction related hardware. The real worry is not whether but how we accomplish this "computer revolution."

Today, we are drowning in information but are starving for knowledge. We can make the most extremely complex hardware, but unless it's useful and accessible to the average user, we have failed to live up to the potential of the Information Age. It is not too early for language professionals to sift through this new knowledge.

The term "computer literacy" is difficult to define precisely. It is often thought of as a single, all-encompassing skill, but in reality, it is actually a complex of many skills. If you can accomplish the following, you obviously would be considered to be "computer literate:"

1. The ability, given ready-to-use software, to operate a computer and to utilize software effectively.
2. The ability to describe and/or explain the components, operation, and uses of a computer.
3. The ability to use a machine-specific programming language to program a computer to perform a specific task.
4. The ability to manipulate programming tools in order to move from one computer to another, using a variety of programming languages(Steffin, 1983). (I would consider this to be an advanced stage of computer literacy).

The acquisition of computer literacy as defined above cannot be attained without a significant effort on the part of the learner. Self-instructional software is readily available and makes the first component of computer literacy relatively easy to achieve. Clearly, the ability to explain, describe, and apply knowledge of the computer and how it works, requires computer language skills. I have seen adults "wilt" in the presence of young children who often seem to easily master the concept of processing information sequentially. Actually, there are programs designed to teach beginning levels of these skills. For example, tutorials are available for helping one learn to program in BASIC, PASCAL, etc. The authoring languages such as SuperPILOT and DASHER also are relatively easy to master.

Formal programming classes or seminars, whether they take place in the school or at computer stores, can help one advance programming or literacy skills. Also, joining a computer club or working with colleagues who have a similar interest in attaining computer literacy will help enormously.

The skills required for dealing with multiple computers, complex operating systems, and the other challenges that professional computing presents, will require a more in-depth commitment to training. Actually, the language lab director or the language teacher would not need this level of "literacy" in order to function entirely adequately with most microcomputer applications. This level of sophistication is reserved for those who wish to be "gung ho" and would require training in several college level courses.

It is important to not try to move too fast in the acquisition of these skills as it is easy to become discouraged. Because of the time required to learn this new knowledge, you should not put off the opportunity to begin right away. One might compare the learning of a computer language to the learning of a foreign language. Computer literacy will be derived from a variety of sources, including the school, the home, and the learner's own motivation to gain mastery and understanding.

ORGANIZATIONS INVOLVED IN EDUCATIONAL COMPUTING

Computer Assisted Language Learning & Instruction Consortium (CALICO)
233 SFLC, Brigham Young University, Provo, UT 84602
(801) 378-6533

Association for Educational Data Systems (AEDS)
1201 16th Street, NW, Suite 506
Washington, DC 20036
(202) 833-4100

Minnesota Educational Computing Consortium (MECC)
2520 Broadway Drive, St. Paul, MN 55113

Association for the Development of Computer Based Instructional Systems (ADCIS)
Computer Center
Western Washington University
Bellingham, WA 98225
(206) 676-2860

CONDUIT
The University of Iowa
Oakdale Campus
Iowa City, IA 52242

Society of Data Educators
983 Fair Meadow Road
Memphis, TN 38117
(901) 761-0727

HOW COMPUTERS CAN HELP YOU.

DRILLS, PRACTICE, AND REMEDIATION--The computer never gets tired of helping the learner master foreign language skills. The advantages of immediate feedback on work the student has accomplished should not be underestimated, particularly since the problem is eliminated of the student working alone without correction and reinforcement.

TUTORIAL--Properly designed programs can present the rule, examples of the use of the rule, and sample problems offering the opportunity to practice and obtain feedback. Tutorial programs also can be formatted to lead the learner along a sequential path from known territory to new knowledge, then be asked to rearrange and process this information to achieve learning objectives.

CREATIVE PROBLEM SOLVING--Since problem-solving skills require complex behavior on the part of the learner, simple tutorial skills are seldom enough to promote the development of such skills. Simulations that serve as models of real-world interactions have the greatest probability of achieving these ends. It should be noted, however, that human dialogue and interaction between the learner and the teacher are essential if the potential of simulations for teaching problem-solving skills is to be fully realized.

TESTING--The interactive test can be an intimate blend of testing and learning, unlike most traditional modes of testing that offer no immediate opportunity to improve. For other applications, the computer can save the teacher a lot of work in presenting questions and grading student's answers. For four years now, We have used a simple microcomputer to randomize exam sheets, thereby giving each student a "unique but similar" exam. This has helped discourage cheating in our self-paced courses. Also, students seem to like the idea of immediate feedback and score when taking an exam at a terminal.

DIALOG--Permits an unstructured "conversation" between the student and the computer(A type of tutorial). The educator is not telling but is leading the learner by means of carefully planned questions, each successive question depending on the learner's previous response.

Someday, adequate voice conversation dialogues may be carried on with the computer, although the state-of-the-art of voice recognition units today are quite primitive and far from being suitable for teaching correct pronunciation of foreign languages.

DATA ANALAYSIS--There are numerous "gradebook" programs on the market today that allow the teacher to keep a running and detailed account of student performance records and scores.

WORD PROCESSING--Many instructors now store exam questions and information on floppy disks, allowing them to retreive files and update them in any manner, often with a minimum of typing.

LEARNER PROFILES--Getting beyond the experimental stage, profiles will soon be on the market, allowing testing to discover how an individual learns. These tests will reveal such variants as dyslexia, right or left brain dominance, and other individual learning patterns. There is much for us to learn about learners and teaching. Without doubt, this is an exciting age!

Obviously, there are many other applications of the microcomputer that can help improve efficiency in achieving our goals. Articles in this brochure indicate other possibilities.

ADVANTAGES OF CAI

--The number of course offerings can be increased through the use of CAI. Various languages, specialized low enrollment courses, or personalized tracks of study can be offered effectively without a corresponding increase in faculty. This does not infer that the computer would replace the teacher but simply assist and aid the instructor to multiply his or her efforts, and in a sence, increase production with little or not more effort than with the traditional approach to teaching.

--CAI frees the teacher from time consuming, routine, and sometimes monotonous tasks such as drills, which can be performed easily and tirelessly by machines. This helps improve the teacher's effectiveness by allowing time for working with students with special needs on a one-to-one basis.

--The problem of the "off step" language course also can be effectively dealt with without requiring additional faculty.

--CAI tends to meet the diverse needs of students with varying backgrounds in that every student has a chance to progress at his or her own pace and ability to the same level of efficiency as the better adept language student. The student who is a fast learner or who has a better background may be able to master a single program in a short period of time while the slower student may choose to stop the program frequently and review before achieving the higher level of preformance on an objective examination.

--CAI gives instantaneous feedback and reinforcement to the learner on quizzes and is capable of providing on-the-spot special instructions on difficult points or concepts. Such flexibility is not available through conventional audio language lab programs.

As the popularity of the personal computer increases, so will the opportunity to distribute self-contained, auto-tutorial language courses to students who wish to study at home, adult learners in hospitals, rest homes, correctional centers, etc.

"IF COMPUTERS ARE SO GREAT, WHY DON'T THEY MAKE THEM SO THEY ARE EASIER FOR TEACHERS TO OPERATE?"

Quite simply, the computer companies are not dependent upon educational institutions for survival, while Textbook publishing companies are. Undoubtedly, as computer companies realize that there is an important financial potential in sales to schools, they will respond with friendlier equipment and operating packages. Daily, we see overtures being made to education by the microcomputer industry. No response on our part could make it go away, although we must be careful not to make knee-jerk reactions to external trends and pressures.

"WHAT'S THE HURRY? WHY NOT WAIT AND SEE WHAT HAPPENS BEFORE JUMPING ON THE MICRO BANDWAGON?"

Dr. John Wedman of the University of Northern Iowa (1983), answers this question very well, I feel. " Time, or the lack of it, is biasing our perspective. As a view of the future of computers in education is formed, that view must be in light of the phenomenal growth of computer technology. From the beginning of time up to 1980 there have been about one million computers. Today, several manufacturers expect to each produce an equal number in a single year! The rapid growth in computer technology, in terms of quality, quantity, and accessibility forces us to make decisions without the advantage of lengthy consideration. Education has not even reached agreement on the question of whether or not hand-held calculators should be allowed in the classroom. Now we are confronted with the computer. In education, we must learn to stop waiting so long for an idea to get ripe that the idea turns rotten."

Most of us will agree that the concept of the language laboratory is excellent. Equally, most of us probably can think of examples of lab installations that are very limited in their effectiveness. Sometimes, the manufacturers of this lab hardware dictated to language professionals the specifications and capabilities of the equipment, too often resulting in installations that were either intimidating or too complex for the average language teacher to use effectively.

We must not allow the future of computers in language teaching to be the product of the computer industry. Neither should it be the product of outside groups attempting to impose their conceptualization of computer literacy for foreign language teachers. We, as foreign language professionals, must bear the burden of properly integrating technology as it best fits our needs.

Valid educational material must involve experienced teachers. Whether teaching is done by computer or any other device, effective educational materials must be designed by effective foreign language teachers.

"HOW DO I KEEP UP WITH WHAT IS HAPPENING IN FOREIGN LANGUAGE CAI?"

For starters, I would suggest you join the Computer Aided Language Learning & Instruction Consortium (CALICO). You may contact: Dr. Frank R. Otto, CALICO, 233 SFLC, Brigham Young University, Provo, UT 84602.

Although CALICO would be a primary source of information, virtually all educational journals publish articles related to CAI, of which many applications could apply to foreign language instruction.

"I HAVE A COLLEAGUE WHO IS AFRAID THAT THE COMPUTER WILL REPLACE HIM..."

> "You can lead an old horse to water. But can you get him to use a computer?"
>
> Arielle Emmett

"The use of technology in education has grown steadily over the years, but only rarely has reached its potential as an effective means of communicating information in the teaching-learning process. While most schools have moved beyond the horse and buggy stage of using instructional materials, it may be appropriate to categorize them as being in the Model-T stage rather than being in the jet age. Each of us has heard the old arguments that motion pictures will replace teachers, that television will replace teachers, and now that computers will replace teachers. The fact remains that nothing has replaced the teacher no matter how good or bad they have been in the classroom"(Huffman, 1984)

The microcomputer, like other media before it, will not replace other learning resources, but it can enhance them. Teaching can be enhanced, programs can be enhanced, school systems can be enhanced. Most resistence comes from a lack of knowledge, the view that the computer is inappropriate in foreign language education, or the fear of being replaced by machines. Also, as students become familiar with computers, faculty may feel threatened by their own lack of espertise in an area where students might be more knowledgeable than they.

Early CAI advocates, in their zealotry, presented the computer-assisted-instruction concept to prospective faculties in a manner that confirmed the fears of teachers--that computers were going to replace, not assist, them. I would guess that there are statements within this brochure that could also be construed to reinforce this fear also. This is not an easy point to objectively discuss.

Once teachers understand the following, most such fears will be dispelled:

--The computer can solve problems for you. Once anyone grasps that connection, learning is secondary to the goal of "problem-solving."
--The computer can save you lots of time.
--The computer is a "personal" computer. It's yours; it's an extension of your mind; it's customized and tailored to fit you and your needs.
--The computer makes communications easier and people more efficient.
--The computer can be fun.
--The language teacher knowledgeable in computers will become more marketable.
--The computer extends your imagination, knowledge and skill base. This taps into one's sence of identity, again, and is a motivator for those who are classically "achievement-oriented" or "inner-directed"(Emmett, 1983).

CAI must be viewed as an adjunctive learning tool, to be used in conjunction with traditional teaching methods. There is a need for both teachers and computers, and a particular need for instructors who can use CAI.

"WHERE DOES ONE OBTAIN GOOD FL COMPUTER SOFTWARE?

Generally speaking, many software language programs have been very disappointing, however, this situation is beginning to change quite rapidly.. Already there are some programs available that can be very useful. The field is wide open for the language teacher who wishes to create original and high quality software programs. By using programs such as PILOT, DASHER, etc. you can produce your own software, although one should not be misled into thinking that this is necessarily easy and can be done quickly. A list of software sources is available elsewhere in this publication.

"THERE ARE MANY COMPUTERS ON THE MARKET. WHICH ONE SHALL I BUY?"

Today, there are more than a hundred vendors competing for the CAI market and dozens of brands of microcomputers available. During the last year, various computer companies got out of the business of producing microcomputers, at least two of which were produced by large companies. There are "clones" advertised as being cheaper and having more capability than the more expensive leading brands they are trying to displace. It is a confusing situation, to say the least.

Choose a machine that can meet your instructional objectives. This choice will be based heavily on the type of software available, programming languages it uses, compatibility with other systems being used for similar purposes, memory needed, is it flexible and easy to use?, good documentation?, does foreign language courseware exist for it?, can it be expanded?, service options?, and so on.

Be careful of the cheaper "clones." While many of the Apple and IBM look-alikes are advertised as capable of running virtually all software packages available for the "real thing," experience shows that this is not always the case. There are quite a few programs that will not work on the clones. Also, some have purchased machines that at one time were compatible with major brands, only to find that the major companies have somehow changed their operating system or legally restricted use of their operating system by the smaller manufacturers. You could be left holding a lab full of computers that are very restricted in their use.

Today,(4/20/84), the following computers are only a few from which to choose: APPLE and Clones, IBM and Clones, Molecular, DEC, Wang, Sony, Altos, Fortune, Xerox, Radio Shack, NEC, Corvus, Texas Instruments, Atari, PET, HP, etc.

Each of the above seems to have special strengths: some undoubtedly have more advantages for business/accounting applications; some excell at games; and a couple might be considered for applications in the language laboratory. What gives each machine its strength? SOFTWARE! No matter how fancy the computer or what is in "vogue," the number and and quality of courseware programs available for the computer tips the hat in its favor. At the University of Idaho, we use the Apple II Plus micros because of their ease in handling foreign language characters, the number of software programs available, and because we know that there are dozens of FL educators across the country who are writing programs for the Apple. The new computers that use icons and a mouse are tempting and cute, but they simply will not meet our particular needs, at this point in time.

For word processing and statistical work, we often use the IBM PC (The keyboard "fits me better"). In the audio section of the learning lab we use audio-active-compare tapedecks because they meet our particular needs be than do other types. For students of music, we use stereo decks. Obviously, each lab will have special requirements that will dictate the precise type of hardware needed. There are "many roads leading to Rome." Choose a vehicle that will "get you there," and do so efficiently and comfortably.

HOW RELIABLE ARE MICROCOMPUTERS?

In our experience, slide projectors and tape recorders require more maintenance than the microcomputer. Although everyone might not be interested, we have purchased a maintenance "diagnostics diskette" which not only provides for speed adjustment of disk drives but also allows one to pinpoint most problems. Usually, repairs are made by simply replacing a defective chip or switch. In nearly five years, practically all of our problems (which have been few) resulted from slight corrosion of chip or board contacts. More recent micros either have gold plated contacts or soldered chips, which eliminates many of these problems. Like most electronic equipment, there are environmental considerations to keep in mind. Heat, dust, and humidity should be kept to a minimum, if possible.

Static electricity is the enemy of all magnetic media stored on floppy disks. Fingerprints, dust, and dirt also can raise havoc with diskettes. Always keep a backup copy of important programs. Also, it is not a bad idea to have a power filtering device between your computer and the wall outlet. This prevents power surges or electrical line interferance from causing you to lose data. If you ever "go inside" of your computer, you must first "ground" yourself so as to dissipate any static electricity that might be discharged onto a computer chip, possibly causing damage. Perhaps the most serious thing you could do would be to unplug language or interface cards while the computer is powered up. This could cause a lot of problems quick! In view of this, it might be wise to keep security in mind. There are those who have been known to strip the "insides" out of a computer when no one is looking or spill a can of pop on the keyboard. Lab guidelines and training sessions for all potential users will offset most problems.

Other considerations to keep in mind have to do with the height of the desk on which the keyboard rests, the angle at which the monitor screen is mounted, and so on.

"CAN PROGRAMS EASILY BE TRANSFERRED FROM ONE BRAND COMPUTER TO ANOTHER?"

No. Some programs can be "adapted" so that they will function on a different brand computer, but this is usually a job for the professional programmer. Many times, it is easier to just "start from scratch" and reprogram from the "bottom up." Some of the more difficult programs to transfer are those that were written for the large mainframe or mini computers where memory was not a problem. BYU is in the process of transferring language courses from their TICCIT system to microcomputer compatible systems, but I wouldn't hold my breath until they are available in Spanish, French, and German. It is a long and tedious process. It will be exciting when these packages are finally available because I suspect that they will be some of the finest obtainable.

The reasons for the problematic transfers have to do with different memory structures, and incompatible operating systems and central processing units.

In some cases, it is possible that the microcomputer can act as a terminal to a larger computer, thereby providing for access to some of the more sophisticated mainframe programs.

"WHY NOT JUST HOOK INTO THE LARGE TIMESHARING MAINFRAME COMPUTERS RATHER THAN USE THE MICROCOMPUTER?"

Cost--Micros are cheaper to operate and more cost-effective. A classroom full of microcomputers is less expensive that a time-sharing system with 30 terminals.

Confidence--When a timesharing system is down, ALL WORK STOPS. How many times have you heard, "We're sorry, but the computer is down"? With a lab full of micros, one "dead" micro will not stop the class.

Convenience--Micros are more portable, allowing the instructor to take them into the classroom or home for work.

Control--With the micro you do not share the central processing unit with anyone. You are in control of your environment. Real-time (uninterruptable) applications are thus practical where they weren't before.

"DOES RESEARCH 'PROVE' THAT CAI IS EFFECTIVE?"

Conclusive findings and conclusions on instructional effectiveness of the use of microcomputers in the classroom are difficult to find, although there have been a number of excellent research studies on the effectiveness of CAI and CMI. Generally, it is shown that:

--CAI has the potential to be an effective instructional aid when measured through the results of student achivement. It appears to be more effective in tutorial and drill modes than in problem-solving or simulation modes. Tutorial and drill modes seem to be more effective for low-ability students than for middle or high-ability students (Hausmann, 1979).
--When students are permitted to proceed at their own rate, they will generally learn more rapidly through CAI than through traditional instructional methods.
--CAI, as a supplement to regular classroom instruction, is at least as effective as other means of individualized supplemental instruction.
--With few exceptions, both students and teachers are highly enthusiastic about CAI as an instructional mode.

"WHAT IS INTERACTIVE VIDEO?"

Interactive video is nothing more than programmed instruction that is a marriage of television and the computer, thereby resulting in a sophisticated learning tool. Together, TV's visual impact and the computer's responsiveness do a better job of explaining than the TV or computer can do alone. They demonstrate and explain from both visual and verbal perspectives. They test skills and recognize and evaluate performance. The ability of interactive video to simulate reality is considered by some to be its greatest strength.

Each of us has heard of studies that indicate that learners retain approximately 25% of what they hear, 45% of what the see and hear, and 70% of what they see, hear, and do. Because interactive video will not progress from learning segment to learning segment without a response from the user, the learner is involved 100% of the time.

Interactive video can be a powerful and persuasive instructional tool, capable of greater variety, flexibility, and relevancy than training materials we have used in the past. Initial testing indicates that interactive video speeds up the learning process.

Systems range from simple hand-held devices that can random-access taped material on the TV screen to sophisticated systems with computer-generated responses, graphics, and sound effects. Other systems include the capability of "multiple-branching" to optional reinforcement or video segments, allowing the student to move through a program of any length and complexity at his or her own speed. Some systems interface with videotape, others with videodisc.

The student first views a segment of videotape on a monitor. The developer, who has "authored" the lesson into the computer system, asks the video segment to pause; and questions appear on the screen in computer generated text. Depending on the capabilities of the particular computer hardware and the limitations of its authoring system, questions can require numeric answers, true/false, multiple choice, matching, fill-in-the-blank, or one or several-word answers.

The student enters the answer into the system, usually by keyboard, and the screen immediately acknowledges the response. Right or wrong, or "almost right, but..."

The answer dictates the subsequent direction the program takes. A right answer may send the student to the next video segment for more information, showing the filmed action (16mm, slides, video, etc.) until it comes to the next decision point. An incorrect answer can send the student back to view the same video segment again or the instructions in the program can tell it to go to remedial segments of the videotape that shows the consequences of each answer.

Students get as many opportunities to answer the questions as they need to get them correct...with no one looking over their shoulder or getting impatient. By reviewing content and reanswering questions, each student should be capable of 100% understanding when finished. Some programs keep a record of both correct and incorrect answers, the percentage of answers correct, the time taken to answer each question, and other pertinent information, and then furnish a printed copy of this information to both the student and the teacher.

While the use of a video tapedeck in interactive video is impressive, with a videodisk system, it can be spectacular! It boggles the mind to think that on one small plastic videodisk, one can place up to 54,000 individual pictures (frames), the equivalent of a half hour of television. Probably the major advantage of the videodisk over videotape would be the very rapid access time to any specific frame on the disk, whereas the videotape may have to fast forward or rewind to appropriate frames, often requiring more than a few seconds.

(For additional information about interactive video and foreign language teaching, please refer to the article by Dr. Larry Gale, "Montevidisco...")

THE FUTURE BEGINS NOW

It has become necessary and economically feasible to embrace computer and television technologies to help alleviate the fiscal crisis in education and to shift from a labor-intensive to a capital-intensive instructional system. The diminishing costs of computer hardware, especially the microcomputer, and the ability to combine the computer with video technology by means of computer controlled video playback equipment, will enable schools to effectively solve many of their most pressing instructional problems if appropriate stand-alone courseware becomes available. Interactive video has the potential to alter the delivery system in ways that current technology cannot.

In the future, it may be predicted that many persons will be able to earn substantial credits toward college degrees by means of remote learning at stand-alone systems or through "down-loaded" sytems from satellites or cable television. When this happens, all rural areas of the country will benefit. We have just scratched the surface of what can be accomplished with computers in education.

The time is right to give serious consideration to the development of foreign language interactive video software to meet projected needs.

THE LANGUAGE LAB OF THE FUTURE..."THE WRITING IS ON THE WALL"

Computer configurations are changing daily. Within a relatively short period of time, hand-held calculators have been upstaged by briefcase telecomputers. Everywhere you turn, you find portable computers being used for thousands of applications that were unheard of just a few years ago. Today's machines are getting friendlier (take a close look at the Macintosh by Apple) and smaller. Newer configurations are more portable and powerful than ever and can be operated in a car or plane. Many factories are now training employees via interative video. Virtually every school in the country has microcomputers at their disposal.

The language lab will change and be more efficient with this technology. I boldly predict that before fifteen years pass, the better language labs in the nation will be heavily equipped with interactive video learning carrels. These installations will be carefully integrated and easy to work with. Recordable videodiscs will be the delivery format. It is likely that most students will have access to their own personal machines. This is an information society and the old delivery systems will be enhanced by this revolutionary tool.

SOURCES FOR INTERACTIVE VIDEO INTERFACES

Allen Communication, 140 Lakeside Plaze #2, 5225 Wiley Post Way, Salt Lake City, UT 84116

BCD Associates, Inc. 5809 N.W. Fifth St. Suite 101, Oklahoma City, OK 73128

CAVRI Interface Video 26 Trumbull Street, New Haven, CT

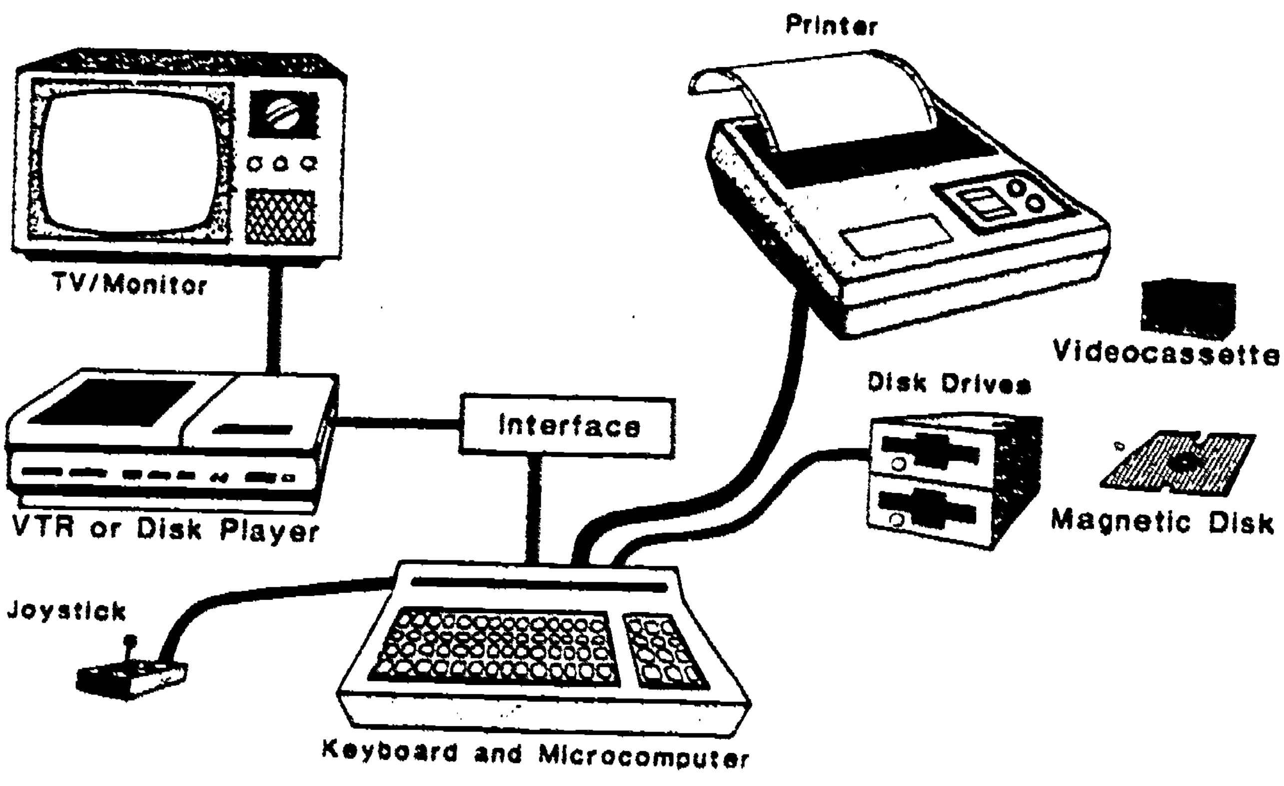
GENTECH 4101 N. St. Joseph Ave., Evansville, IN 47712

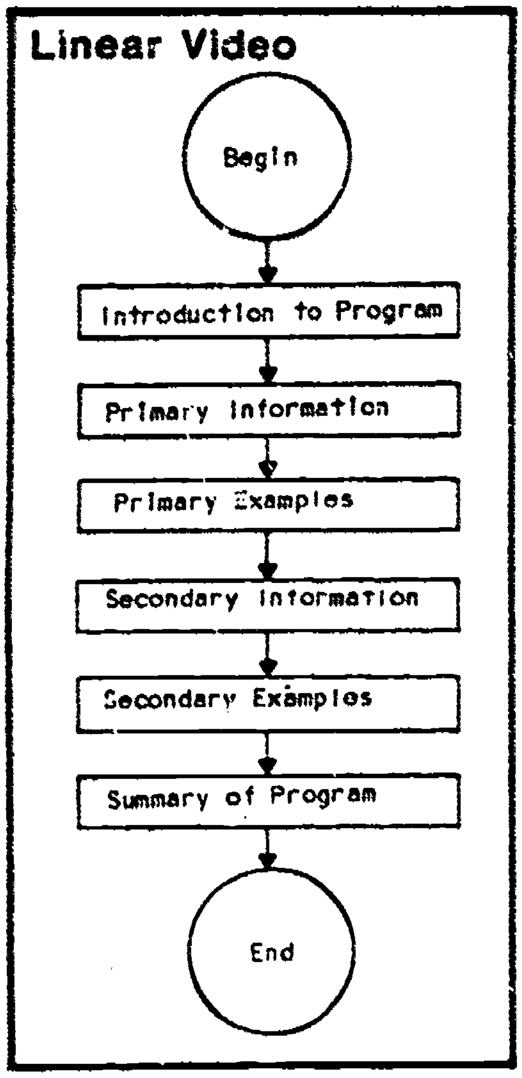
Whitney Educational Services 1777 Borel Place, Suite 416, San Mateo, CA 94402

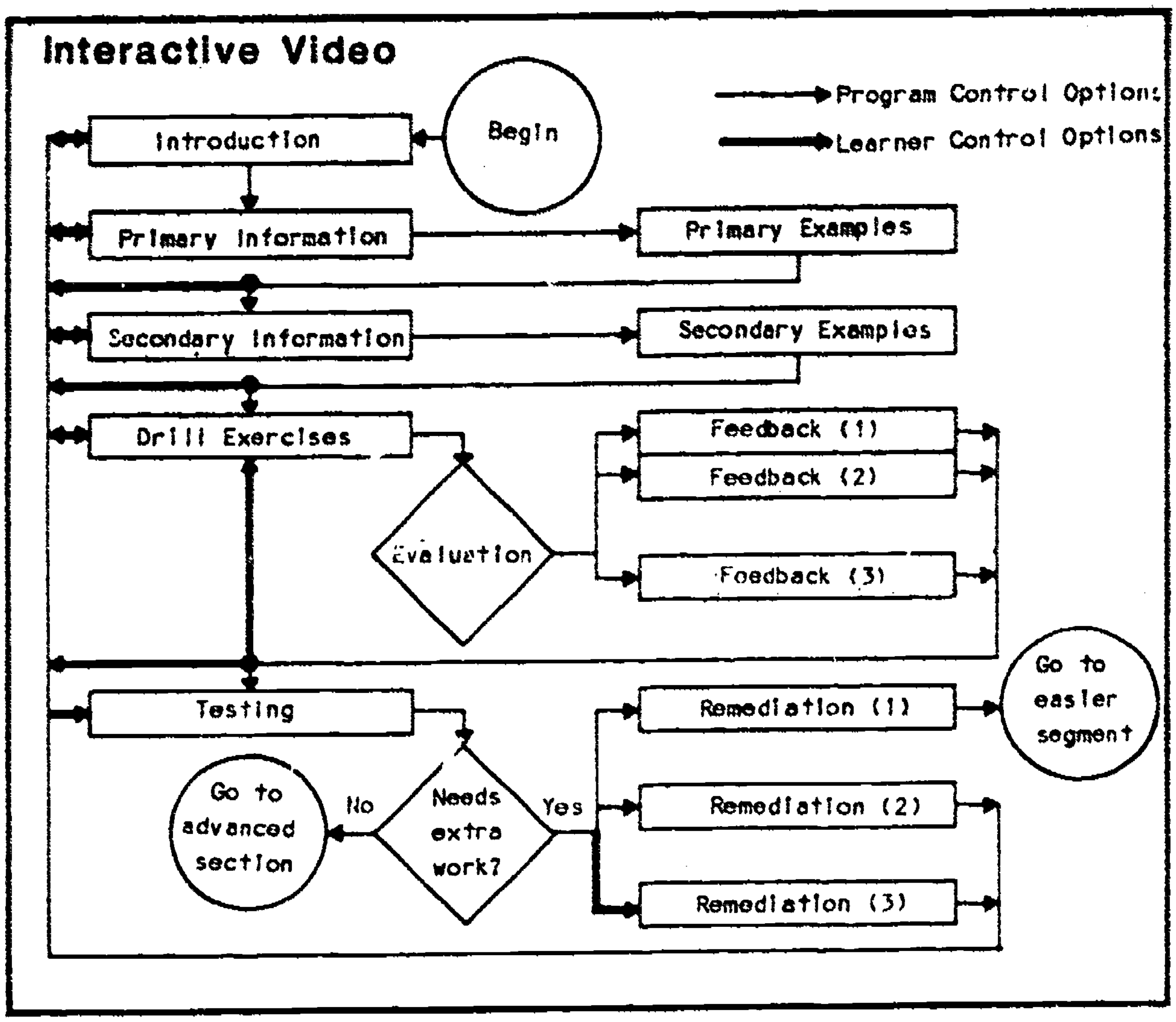
Bell and Howell 7100 North McCormick Road, Chicago, IL 60645

Obviously there are others, but this should get you started. As always, the buyer should be careful. In the early stages, many of the above systems had deficiencies. Compare carefully. Some are quite flexible. Quality of documentation varies. Software support programs vary in quality. Some cost less but offer practically the same features as more expensive systems.

Simple Interactive Video Microcomputer System



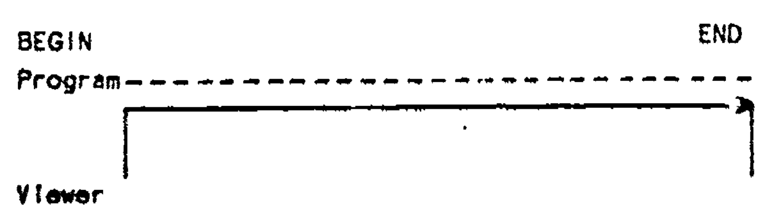




A linear video program goes through lessons from the beginning to the end. All students must view the entire contents of the program. In interactive video, the program has branching formats that allow or require the student to make choices, resulting in a dialogue between the student and the computer. In this example, the learner can practice before being tested. The computer makes sure that students have mastered all the materials before proceeding to the next segment.

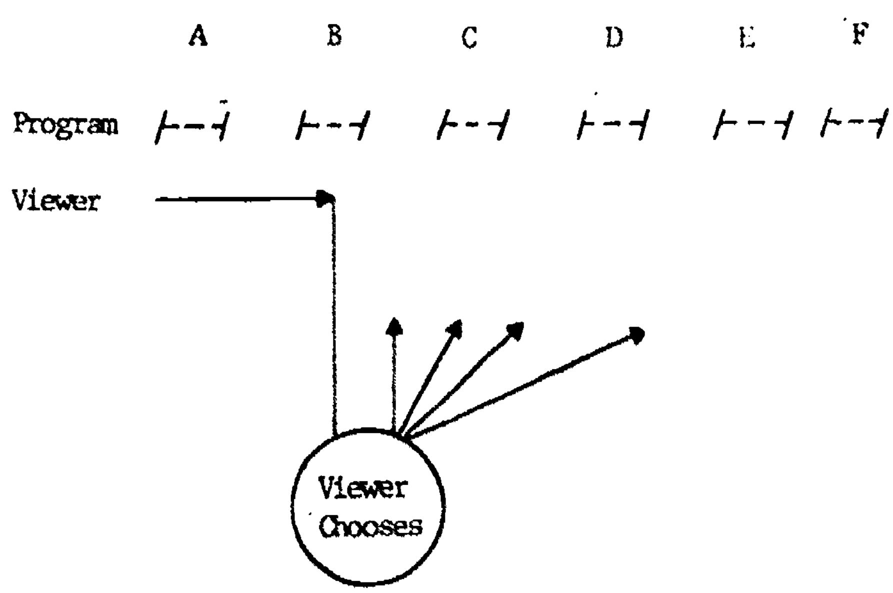
Levels of Interaction

LINEAR VIDEO----Movies or Broadcast TV
The viewer watches a program designed to be seen from beginning to end.



LINEAR VIDEO----RANDOM ACCESS
The viewer accesses particular sections of a linear program that are of interest.

INTERACTIVE VIDEO----Video tape or Video Disc
The viewer chooses what will be seen, in response to a computer graphics menu. There are variable paths through the content.



WRITING YOUR OWN CAI LESSONS

Any teacher or trainer who has the experience of carefully structuring student lesson plans, and a certain degree of "computer literacy," can build computer self-instructional packages (which are just more explicit student lesson plans). When a student works independently with materials without the direct assistance of a teacher, the materials must explain it all and explain it well. A high percentage of students should be able to successfully complete a given lesson without calling for assistance from the teacher. At the same time, the lesson must not be so explicit that it bores the talented student. The basic steps are as follow:

OUTLINE THE CURRICULM AND OBJECTIVES

Self-instructional packages become building blocks which in the end create the whole unit. It is necessary to look at the whole before building individual lessons. The whole of a unit has sub-topics and sub-sub-topics. Each sub-topic should represent a clearly stated instructional purpose and performance objective which can be achieved in less than an hour of the student's time. For younger students or some language drills, the lessons might be even shorter. Keeping the steps small is a basic principal for any self-instructional lesson(Stiehl/Streit, 1983).

SELECT A LESSON TOPIC

Choose an appropriate topic from your curriculum that you would like to prepare for self-instructional delivery. Keep in mind that some objectives might be more effectively handled by live presentations or group discussions. You should ask the following questions about the program:

1. What is the purpose of the lesson?
2. What entry level skills are required for the lesson?
3. What will the student be able to do when finished with the lesson?
4. How will the student show what has been learned?

KEY AUTHORING STEPS

--RESEARCH THE CONTENT

--DRAFT FLOWCHARTS AND SCRIPT

--WRITE THE PROGRAM

Once you begin creating a program, you'll soon discover the complexity of the task faced by instructional designers each time they plan a new program. Personal experience indicates that no matter what program you produce, there is a very good chance that it will have unforseen problems that, "in the trade", are commonly called "bugs." You may soon find that bugs take on more forms than the human mind is capable of comprehending!

--PRESENT THE PROGRAM

Once you have "debugged" the program, you now may evaluate how your learners benefit from and enjoy using the program. You will be proud when your students can actually work through that first lesson without your assistance. You may now make any final changes and replicate the disks for distribution.

--ANALYZE FEEDBACK

You will want to keep track of suggestions and ideas for improved design for future programs.

CAI AUTHORING LANGUAGES CAN SAVE YOU TIME

The easiest way to obtain programs for use in helping you teach foreign languages is to purchase completed programs. Unfortunately, at the present time, the selection is relatively poor and you probably have computer applications in mind that are not readily met by these commercial programs. You may wish to "do it your way" or have the drills correspond closely with your textbook. The alternative is to write the programs yourself. Most people begin by studying BASIC computer programming language either on their own or in formal classes (the latter is preferred). Unfortunately, this will probably be more time consuming than you ever imagined. You may find that a year, or even two, will have passed before you are able to put together programs in BASIC that utilize the potential of the computer and meet your strict requirements. There is a "shortcut." Learn to use an "authoring system!"

Authoring systems allow someone with little or no computer programming experience to author CAI lessons with up to a 90% reduction in time required to learn the programming techniques. The trade-off is one of flexibility since authoring systems rely on a predesigned structure. In the opinion of most, this is of no great consequence when you consider that authoring systems offer high speed lesson execution and, relatively speaking, little production time.

At present the two most popular authoring systems for foreign language CAI are DASHER (a very easy-to-use "answer processor" for creating drills) and SuperPILOT, which readily lends itself to high quality CAI sequences utilizing hi-resolution graphics, sound, macro characters, and other techniques that would take the average person more than a year to master for use in programming in BASIC. In addition to these authoring systems, there are perhaps a dozen or more on the market, some of which may lend themselves to FL CAI.

In the early 1970's the original PILOT language was developed at Western Washington University to allow instructors who knew little about computer programming to write computer programs that provide for individualized tutoring, drills, practice, and testing. The original language had only eight instructions and took only a short while to learn. Since that time, PILOT has been revised and improved many times and today is probably the most popular and flexible authoring language. It is advertised that most early versions of programs written in PILOT are compatible with modern versions. While many computer systems have special versions of PILOT available for them, some offer more features than others. My experience with IBM PC and ATARI versions ,at this point in time, indicate that they are quite limited when compared to the Apple version.

SuperPILOT is a complete programming language that comes with an author disk (and backup) which contains the language and editors; a disk of sample lessons, special characters, and sound effects; an excellent two-disk tutorial (SuperPILOT Co-PILOT); and two manuals. At about $200.00 it represents quite a value. SuperPILOT requires 64K and two disk drives in order to author lessons, while students need only one drive. There is no additional cost involved for student disks except for blanks. An optional statistical package is available (SuperPILOT LOG, $50.00) which allows the teacher to keep track of student progress and analyze records. Also, Superpilot is compatible with various interactive video interface systems. Presently, we are working with SuperPILOT and the BCD interactive video system and are encouraged by the results.

SuperPILOT has 26 simple and logical commands, each consisting of only one or two letters. Less than a dozen commands are used regularly, most of your time being spent in typing in the contents of your lessons.

While SuperPILOT is good at presenting information on the screen, accepting answers from the student, evaluating each answer according to criteria selected by the instructor, providing feedback, and so on, it is easy to add color graphics, special foreign language character sets, sound effects, and as mentioned, keep records of student performance, and interact with AV equipment.

The size and color of text can be changed instantly. The instructor can change the size and thickness of letters, produce printed copies of lessons, and many other nice things that are quite difficult or impossible to accomplish with other authoring languages. Stephen M. Weissmann says that, "Using a language such as BASIC, it is generally recognized that to create an hour of quality computer interaction involves at least 200 hours in development. With SuperPILOT, that figure is closer to 20 hours"(Weissmann, 1983).

While it is obvious that SuperPILOT has many strengths, it also has some weaknesses. The editing program is somewhat slow, although actual running time is generally acceptable. My real "beef" with the system lies in the tedious and confusing manner in which the reference manuals are written. Because of this, the new learner will spend from 20 to 100 hours mastering the techniques need to effectively and efficiently use SuperPILOT. This is still a far cry from the time that would be required to program similar programs in BASIC. Also,one should keep in mind that a more specialized authoring program, such as DASHER, might more easily meet your immediate needs.

USING THE MICROCOMPUTER TO REPLACE YOUR TYPEWRITER

As I sit here typing, I know deep inside that once you discover the ease of using a word processing program, you will be hesitant to return to the conventional typewriter except for the smallest typing jobs. Following, I have listed a few of the many tasks a good word processing program can accomplish:

--Correct errors in any part of your text in seconds.
--Merge a paragraph from a "paper" you gave last year to the text body of an article you are presently writing, and do so in seconds.
--Remove or insert words, sentences, and paragraphs anywhere in your text within seconds.
--Have your text fit onto sheets properly, before it is printed out
--Change margins, add **bold characters**, use superscripts or subscripts at the stroke of a key.
--Instantly center lines or justify margins, and so on.

Each computer system seems to have dozens of different word processing programs available for it. Which is best? That is difficult to say. Each seems to have advantages and disadvantages. I would strongly suggest that you try out a few programs before you purchase. I tested four different word processing programs before I found one that I like--and I'm not sure I would recommend it to the neophyte because I think that it requires too much time before one gets the "hang" of all the commands. Ease in learning and use is important.

FOREIGN LANGUAGE CHARACTER SETS AND MICROCOMPUTERS

If you will require foreign language character sets, you will find that most word processing programs will not easily allow for them. Some program character fonts can be customized, however. A check with dealers should give you the most recent developments in this area (They change daily, it seems)

It soon becomes apparant to the linguist that the American Standard Code of Information Interchange (ASCII, pronounced "Ask-key"), did not have foreign language people in mind when it was adopted. This presents problems for those of us who wish to process foreign language text or incorporate FL characters into our CAI sequences.

There are ways to get around this problem, however. Certain word processing packages do allow for foreign character sets. Some CAI authoring programs also have a built-in character set (e.g. DASHER) or a character font editor, allowing you to define your own character sets as your wish. SuperPILOT, by Apple, contains a relatively easy to use set editor although you had best not assume that the FL character fonts supplied with the program actually work. Of the 15 sets we purchased, none was completed). Also, programmers might be interested in knowing that, for the Apple Computer, Apple Toolkit has a routine that allows you to create your own foreign language character sets for use in programs written in BASIC.

If you are new to computers, you may wish to consult with someone before spending too much time figuring out how to create your own character sets using one of the editors. The first time can be very confusing...the second time will be a "snap."

MANAGEMENT APPLICATIONS FOR THE MICROCOMPUTER

GRADEBOOK MAINTENANCE

There are a number or grade-book programs available to classroom teachers that enable them to maintain attendance, test, and keep track of homework records for each student. However, such record keeping presently is in a simple and formative stage as compared with what may be available not too long from now. Consider for a moment the possibility of a teacher's being able to monitor directly via the student's interaction with the computer. Assessing diagnostic data for each student's approaches, successes, and failures in problem-solving activities gives the educator an opportunity to prescribe the most fruitful learning situation.

GENERAL CORRESPONDENCE, MEMOS, AND REPORTS

Word processing software offers a tremendously powerful way to create, alter, store and print text-oriented data. Thus, general letter and report writing can be greatly assisted. Many word processing programs allow the combination of more than a single file of data. This feature can automatically merge a file with a standard letter with a second file containing an address list. The result is a series of letters addressed to many persons--a valuable tool for communications. Once text files are created, storage is magnetic. This reduces the need for physical paper files and recall is rapid.

CATALOGING

It is here that special database software programs shine. Consider the ability to electronically file names, addresses, telephones, pay rates, use history, films, cassettes, books, equipment, etc. Again, physical storage is minor since no paper is required. Recall is swift and can be done using various search parameters. For example, you might search for all the taped programs in stock that feature French female singers during the last two years.

CAPITAL, PERSONNEL, AND EXPENSE BUDGETS

Spreadsheets and database software are of great help in developing, recording, and modifying the details of any operation, whether it be a store or a language laboratory. Initial budget information is input and can be tracked against actual dollar commitment thoroughout any given time period. This gives you a much better idea of where you stand at any moment. In addition, forecasts can be developed to project how you'll come out at the end of the year, and comparisons can be made against how you're doing in relation to prior time periods. Accounts receivable/payable and billings also can be handled with appropriate software programs.

SCHEDULING AND CONTROL

Spreadsheet programs will allow you to record the allocation of your human and physical resources and to predict those slow periods when you will need to develop fill-in work or busy times when you should arrange for increased support.

PRODUCTIVITY MEASUREMENT

From data collected above, you can determine many useful statistics that can help you to plan and budget more effectively.

PROGRAM LOGGING AND EQUIPMENT CONTROL

If your operation includes the distribution of audio or video by cable or some other form of transmission, the personal computer can be assigned to monitor and/or control the operation of the network. Tape recorders can be started, stopped and recycled. Program listings can be generated for either paper or electronic distribution.

INVENTORY

All hardware and software can be controlled in an inventory program in the personal computer. This allows you to analyze the usage data to determine what is used and in what volume.

EQUIPMENT AND MAINTENANCE INVENTORY

If you log the equipment that you use and record each maintenance review, over time, patterns will emerge. For example, you will be able to compare the repair history of different brands of the same type of equipment. If you do your own maintenance, you'll even be able to identify specific electronic or mechanical components that have short lives.

INTERNAL/EXTERNAL COMMUNICATIONS

Your personal computer can be connected to other computers within your school or across the country through hard wiring, sharing of magnetic mediums or telephone networks. This would also allow you to take advantage of the "electronic mail" features of computer software.

LABELS AND SHIPPING

The labels for program tapes or shipping can be printed on an almost automatic basis.

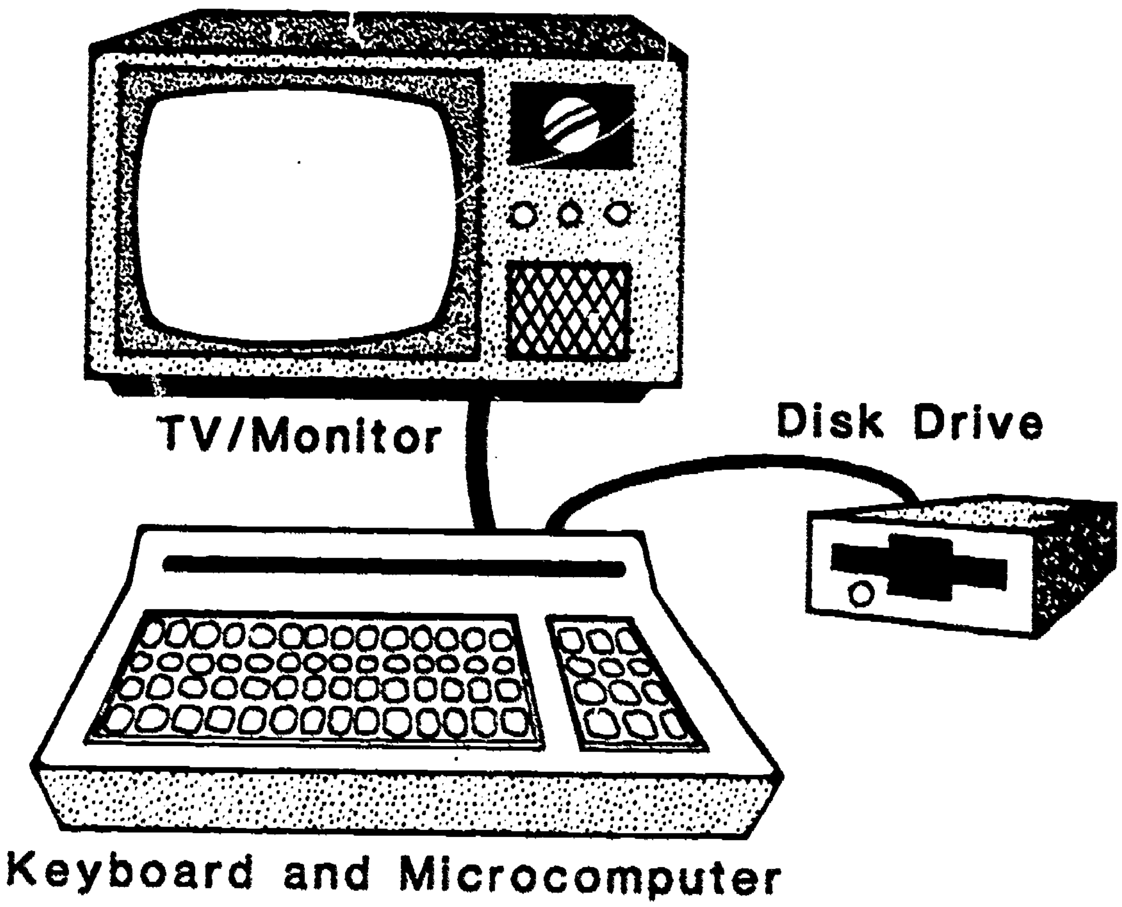
SECURITY

You may be surprised to know that it is possible to let your personal computer monitor entrances and exits at your facility with the addition of electronically controlled locks.

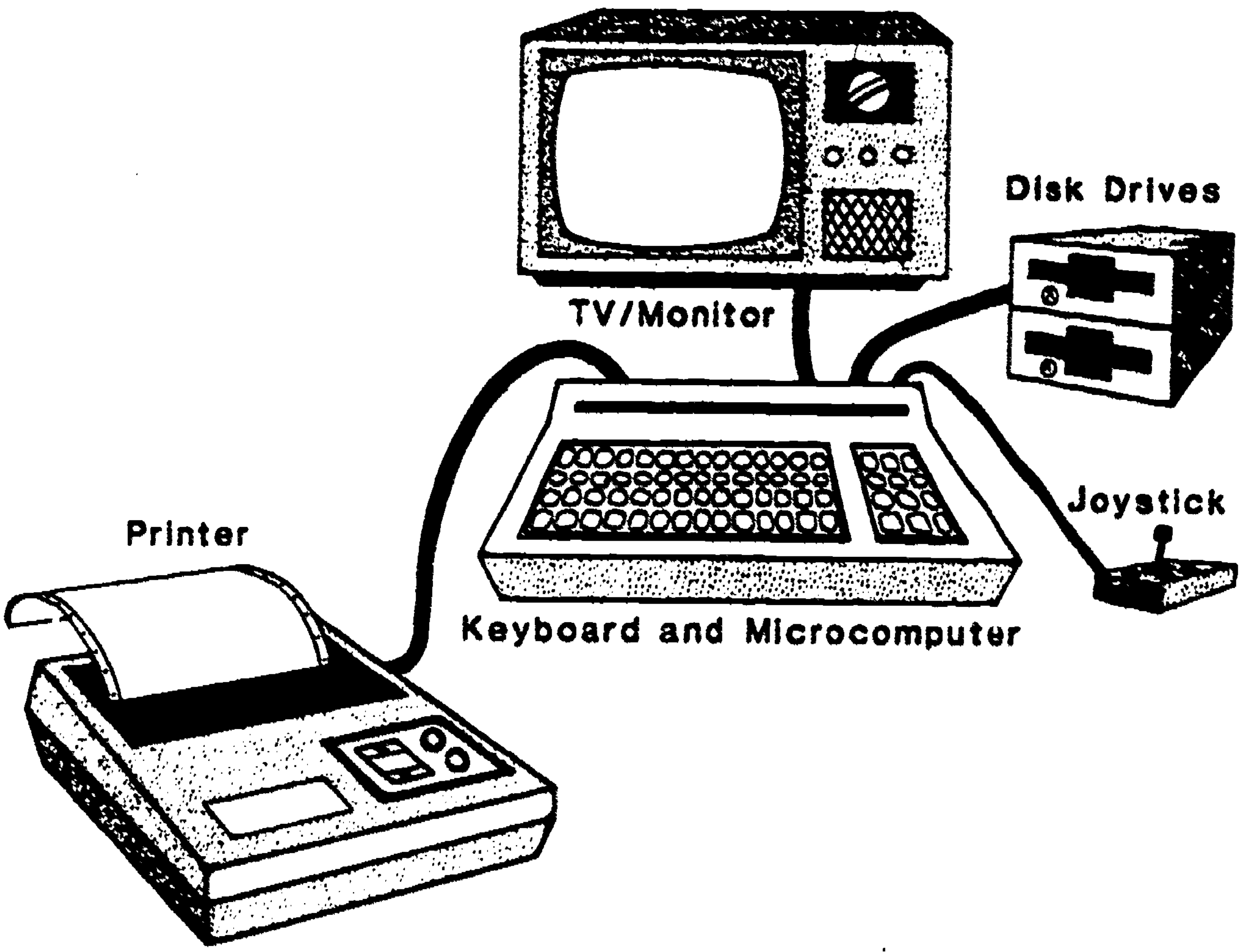
As you can see, the personal computer can be particularly useful decision-making and process device for the language lab director or media manager. It must be emphasized, however, that a personal computer and software do not provide "magic" solutions. Few software programs can be used without some modification or study. This will require an investment in time to learn and keep learning, but the rewards are there.

(Note: Much of this section was adapted from an article written by William C. Hight, Video Manager, Vol. 7, No. 5, May 1984, pages 18-33).

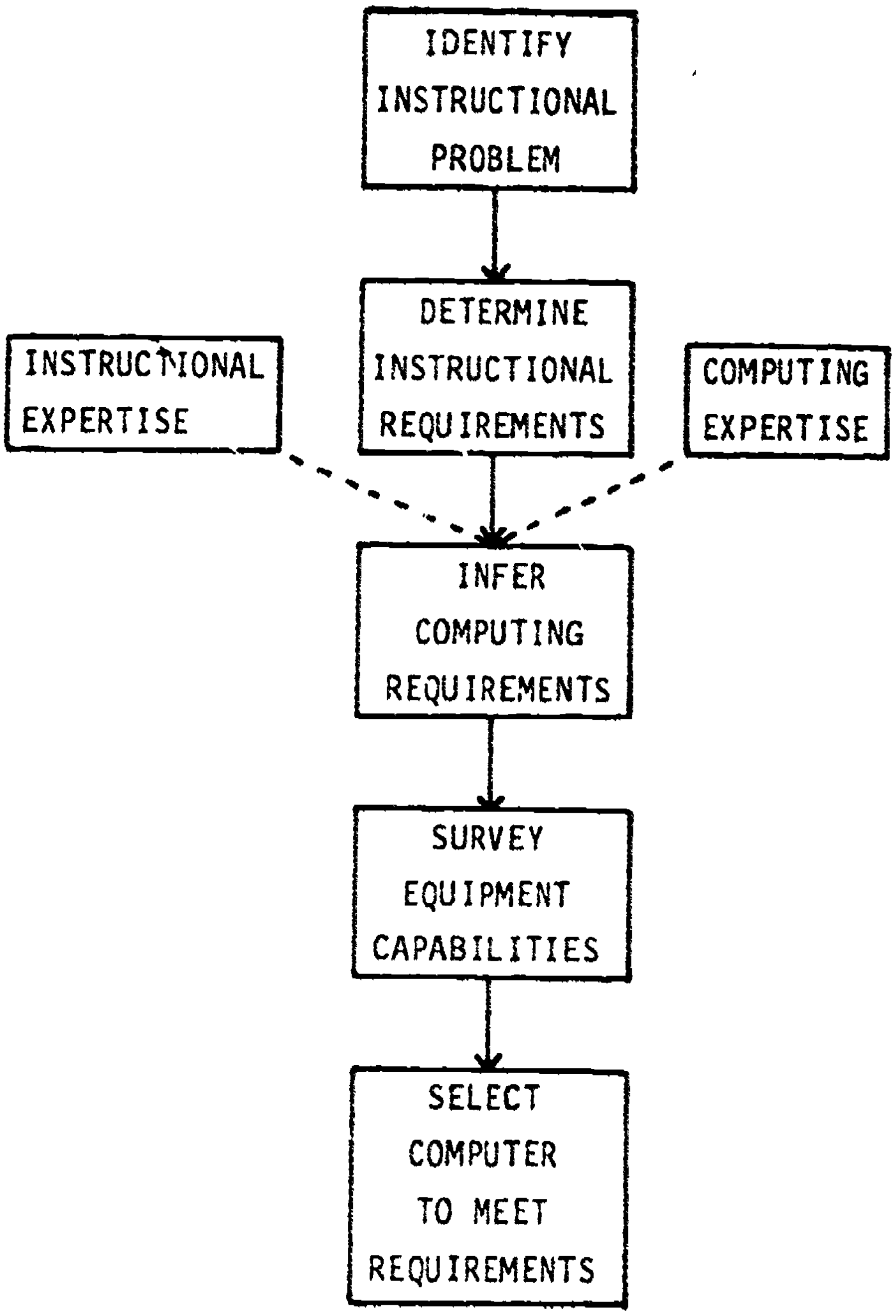
LEVEL 1
Microcomputer System



LEVEL 2
Microcomputer System



MICROCOMPUTER SELECTION MODEL



(Thomas, 1979)

Choosing an Instructional Microcomputer

Things to keep in mind

1. For what purpose?
   - Distribution of application types
   - Type of user
   - Future

2. Is it flexible and easy to use?

3. Can it be expanded?

4. Does it have good documentation?

5. What are service options?
   - Purchase contract
   - Board exchange

6. How much?
   - Initial
   - Later
   - Amortization

7. Does it need to be portable?

8. Good instructional courseware available?

BEWARE!!!

1. Many manufacturers under-financed
   a) Advertise products before production
   b) Here today gone tomorrow
   c) Software written by inexperienced programmers

2. Expansion capability

3. Documentation
   - User (operating)
   - Language
   - Systems

4. System peripherals supported in software

What Capabilities or Options?

1. Ease of use (try it out!)
2. Processor (Speed) (Compatible)
3. Memory Size
   - Minimum
   - ROM
   - Amount Available for Application
   - Maximum
4. Character set
   - Upper and lower case
   - Size
   - Special Foreign Language font
5. Number of Characters per line and number of lines per display
6. Programming language(s)
7. Storage
   - Cassette (generally not recommended)
   - Floppy disk
     - Mini
     - Dual (At least one setup should have two drives)
8. Graphics
   - Color
   - Resolution
9. Computer as terminal (communication interface)
10. Printer
    - Quality and Speed
11. Other peripherals
    - Slide projector
    - Music board
    - Speech synthesizer
    - Random access audio
    - Video disk or video tape interface
    - Tape recorder control
12. Software
    - Utilities
    - Courseware
13. Quality
    - Warranty
    - Reputation

CHOOSING A PRINTER

Somewhere along the line you will need a printer, either to print out copies of drills and exams or for word processing. Again, you will need to make a needs analysis before making your final choice. Currently, most printers you will find are of the "dot-matrix" type which print letters made up of small dots. These printers vary quite a bit in price and quality. Many find this type print tedious to read, although the better quality dot-matrix printers now have a "correspondence mode" which simulates letter quality. The cheapest printers use a special thermal paper which can be expensive. Dot matrix is nice in the sence that print speeds are relatively fast (up to 200 characters per second, although correspondence mode is half as fast). Dot matrix printers also are more flexible in that often they can be used more easily for graphics, charts, and foreign language character sets.

If you require typewriter quality printing and do not need to print lots of graphics, a letter quality printer might better fill your needs. These usually utilize a "daisy wheel" or "thimble" printing element which renders a finished copy virtually identical to that produced on a typewriter. Often, letter quality printers cost more than comparable dot matrix printers and the print speed is quite slow (from 14 to 40 or so characters per second). Some machines are multilingual or allow one to substitute print wheels for various languages. You should be careful, however, to acertain whether or not the printer you wish to buy is compatible with the software programs requiring foreign language character sets.

Another type of printer that is just now coming into its own is the ink jet printer. The goal of the ink jet is to print letter quality characters at the speed of the dot matrix printer by spraying ink onto the paper. At present, these machines tend to be more expensive and have their own special problems, although I suspect that the future will be bright for this technology.

Also, you should be aware that connecting cables and interfaces usually are not included in the price of a printer and must be ordered separately. Most require either a "serial port" or a "parallel port." They are not interchangable. Be sure that the printer you purchase is compatible with your computer.

Many printers feed paper like a common typewriter, others have paper feed mechanisms that feed sheets into the printer one at a time. Computer paper with holes along each side is designed to be used in machines equipped with a "tractor feed." This keeps the paper in alignment as it is being fed.

If you do your own programming, word processing, or use the computer to print out exams or other materials, you will be glad that you have a good printer.

DISK DRIVES

Magnetic disks offer almost unlimited data storage capability for the microcomputer. These disks are available in various sizes, either rigid or flexible. The disk drive reads data from, or writes data onto, one or more disks or "diskettes." The Winchester disk is rigid, sealed permanently in the disk drive unit for protection from contamination and stores larger amounts of data than the "floppy disk." The latter is a flexible circle of plastic, most often 5 1/4" in diameter. Recently, even smaller floppy disks are beginning to be used.

A typical application of a hard disk Winchester drive might be for recording information and grades for 1000 or more students. Floppy disks, which are more common and much cheaper, are used for more temporary storage and usually contain programs for specific tasks.

Student learning stations often are equipped with one disk drive, while some applications require two drives. Two disk drives often can accomodate larger programs and provide for a large data base.

APPROXIMATE COSTS* OF ADDING ONE STAND-ALONE MICROCOMPUTER SYSTEM AT THE

UNIVERSITY OF IDAHO

	APPLE IIe	IBM PC
MICROCOMPUTER, KEYBOARD, 64K RAM, GREENSCREEN MONITOR, TWO DISK DRIVES, CONTROLLER CARD	$1,378.00	$2,590.00

ADDITIONAL ITEMS AND APPROXIMATE COSTS

COLOR MONITOR	Approx. $300-$400
PRINTER (Price depends on quality and print format)	Approx. $400-$2,000
Apple SuperPILOT Authoring Language	Approx. $200.00
Joystick	Approx. $50.00
Interactive video interface	Approx. $600.00 and up

*NOTE! The above prices reflect bid prices as of 1/30/84.

A BUYER'S GUIDE TO FOREIGN LANGUAGE STUDY

ACORN SOFTWARE PRODUCTS INC.
1945 Gallows Rd.
Suite 705
Vienna, VA 22180
(703)556-9788
French, Italian, German, Spanish
TRS-80 Models, I, III; Atari 400,800; IBM Personal Computer

ADVANCED OPERATING SYSTEMS
450 St. John Rd.
Suite 792
Michigan City, IN 46360
(317)298-5400
Many languages
TRS-80; Apple

AURORA SYSTEMS INC.
2040 E. Washington Ave.
Madison, WI 60005
(608)249-5875
Hebrew
Apple II

AVANT-GARDE CREATIONS
P.O. Box 30160
Eugene, OR 97403
German
Apple

COMPU-TATIONS INC.
P.O. Box 502
Troy, MI 48099
(313)689-5059
German, French, Spanish
Apple II; Atari 800

COMPUTEP
C.P. 7
St. Lau, Quebec
Canada H 4WZ
French tutor
Apple II

CONDUIT
100 Lindquist Center
University of Iowa
P.O. Box 388
Iowa City, IA 52244
(319)353-5789
Spanish
ple II, II Plus, IIe

DAVKA CORPORATION
845 N. Michigan Ave.
Suite 843
Chicago, IL 60611
(312)944-4070
Hebrew
Apple; TRS-80: Atari

GEORGE EARL SOFTWARE
1302 S. General McMullen
San Antonio, TX 78237
(512)434-3618
Spanish, French
Apple II, II Plus
Commodore PET

ICXJL
845 N. Michigan Ave.
Suite 843
Chicago, IL 60611
(312)787-7856
Hebrew
Apple; TRS-80 Models I, II, III

INSTANT SOFTWARE
Route 101
Peterborough, NH 03458
(800)343-0728; (603) 924-9471
Russian
North Star; Heath; Commodore; TRS-80 Models I, III; Apple; TI 99/4A

JAGDSTAFFEL SOFTWARE
345 Brenda Lee Dr.
San Jose, CA 95123
(408)578-1643
Japanese Katakana, Russian Cyrillic
Apple II, II Plus

KRELL SOFTWARE CORP.
1320 Stonybrook Rd.
Stony Brook, NY 11790
(516) 751-5139
German, Dutch
Apple II; Atari; DEC; Commodore PET; North Star Horizon; IBM Personal Computer; TRS-80 Models I, III; MA/COM

MICRO LEARNINGWARE
P.O. Box 2134
North Mankato, MN 56001
(507)625-2205
German
Radio Shack Color Computer; TRS-80 Models I, III; Apple; Commodore

POWERSOFT INC.
P.O. Box 157
Pitman, NJ 08071
(609)589-5500
Many languages
Apple II

PROGRAM DESIGN INC.
11 Idar Court
Greenwich, CT 06830
(203)661-8799
French
Apple II; Commodore PET; TRS-80 Models I, III; Atari

SCHOOLHOUSE SOFTWARE
290 Brighton Rd.
Elk Grove, IL 60007
(312)526-5027
French, German, Spanish Russian
Apple II

SEI(SLIWA ENTERPRISES INC.)
P.O. Box 7266
2013 Cunningham Dr.
Suite 238
Hampton, VA 23666
(804)826-3777
Spanish, German, French
Apple II, II Plus, IIe; IBM Personal Computer and TRS-80 Model III in 9/83

SELECT INFORMATION SYSTEMS CO.
919 Sir Francis Drake Blvd.
Kentfield, CA 85058
(415)459-4003
Hebrew
Apple II; Cado Systems; IBM Personal Computer

SYNERGISTIC SOFTWARE
830 N. Riverside Dr. Suite 201
Renton, WA 98055
(206)226-3216
Many languages
Apple II; IBM Personal Computer

Buyer's Guide continued:

TYCOM ASSOCIATES
68 Valma Ave.
Pittsfield, MA 01201
(413)442-9771
French, German, Spanish
Commodore PET:
Soon available for
Commodore 64

XTRASOFT
P.O. Box 91063
4910 Roman Dr.
Louisville, KY 40291
(502)499-1533
Many languages
Heath

EMC Corporation
300 York Avenue
St. Paul, Minn. 55101
(800) 328-1452
French, GErman, Spanish

Opportunities for Learning,Inc.
8950 Lurline Ave.
Dept. CB830
Chatsworth, CA 91311
(213) 341-2535
French, German, Spanish

Bel Air Software, Inc.
P.O. Box 1045
Bell Air, MD 21014
(301) 879-8784
Bel Air deals in a wide variety
of foreign language and ESL
CAI programs--Also, their prices
are often discounted.

DRILL PROGRAMMING CONSIDERATIONS

(Note: This very informative article was written by Dr. James P. Pusack of the University of Iowa who is one of the leading foreign language educators working with CAI. He is the author of DASHER, a foreign language "authoring program" distributed by CONDUIT).

Foreign languages are by nature an area where computer-assisted instruction should thrive. Mastering a language's patterns and exceptions demands hours of practice and repetition--requirements which are seldom met in a traditional classroom. In addition, the instructor in the traditional classroom can seldom provide individual attention to accomodate each student's abilities and prior training. Computer-assisted instruction can help relieve these problems. For example, students can use computer-based drills to practice applying rules of the new language as long and as often as necessary, without straining instructional resources or the instructor's patience. Students can also adjust the pace and content of the instruction to meet their particular needs, skipping over lessons on topics they have already mastered to concentrate on areas where they need more practice.

The potential of CAI in foreign language learning has been readily demonstrated by development projects and hundreds of programs written since the mid-1960's. Yet widespread adoption of CAI in language instruction has been hampered by several persistent problems: lack of equipment, lack of computing skills, suspicion of technology, and lack of support for creating software. Moreover, this area seems condemned to reinvent solutions because so many of the programs are not reusable at other institutions with different computers or are not publicly documented in a way that could allow others to build on a proven foundation. The results of a 1978-1978 survey by Olsen show that these problems are widespread, but that numerous institutions are nevertheless engaged in developing foreign language CAI programs (Olsen, 1980).

Incorporating computers in foreign language instruction has been approached with various strategies. Thus, each approach must be viewed in the context of the instructor's goals, especially in terms of the importance placed on grammar, vocabulary, listening comprehension, writing, reading, speaking, and culture. No single computer program offers assistance in all of these skill areas; few are aimed at more than one or two skills. The spectrum of foreign language CAI ranges from tutorials resembling dialogues where all answers are predictable and the subject matter is broken into discrete units, to comprehensive simulations of real-life situations where the objective is to help the student understand a cultural process.

TUTORIALS

Tutorials present lesson material and evaluate the student's responses in a dialogue format. Based on the student's performance, the tutorial then adjusts the pace of the presentation to the student's progress. This type of "selective branching" is especially useful in remedial and review work. Tutorials are typically used as supplementary modules on difficult learning objectives, such as the distinction between the verbs <u>ser</u> and <u>estar</u> cr the prepositions <u>por</u> and <u>para</u> in Spanish. But with the growing popularity of the personal computer, the self-contained tutorial will doubtless become more popular.

The weak spot of the self-contained tutorial lies in the absence of the instructor and class, where language learning is usually converted to real-life situations. If the nature of language is communication, CAI will never by able to replicate all the ways the learner may need to apply his skills. Even when coupled with other audio and video media, the tutorial package will still function best when the student has recourse to a human teacher.

DRILL

Drill programs are the primary way computers are now used in langauge teaching. Mastering the rules of a foreign grammar is a complex task which is made more difficult by interference from the student's native grammar. Students need extensive practice in applying the rules of the new language. This routine process involves predictable patterns of right and wrong answers and thus is a natural application for computers. An added advantage of computer-generated drills is that students and instructors can spend more class time on less predictable forms of communication, such as speaking and listening.

Since the computer itself has no capacity for understanding and no internal analog for the relationships expressed in sentences, a CAI program must generate drill material as a series of tasks, usually questions and answers or cues and responses. Student-initiated questions and free-form answers lie outside the capacity of today's machines and programs, except in highly restricted contexts. Simulated dialogues between a foreign language speaking computer and a student will remain a chimera until we make enormous strides in basic linguistic research, artificial intelligence, and machine efficiency. Even with such advances, the simulated dialogue may well exceed the practible limits of CAI.

Since the computer cannot master the complex grammar of a natural language, it must then be used to simulate classroom drill procedures which are highly controlled and relatively predictable interactions. For example, the student might be asked to change a cue in the present tense to the past tense. CAI language programs most often present the cue in the target language and instruct the student to formulate a response in the target language. Responses from the student in the native language or in the form of multiple choice answers are, respectively, too complex and too simple for effective CAI. The target-language answer is the preferred response because it requires that the student apply rules within the target language.

Creating language drills on the computer requires two steps: creating the cue and analyzing the response. Computer-programmed approaches to drill exercises can best be understood in terms of the ways they handle these two steps.

CREATING THE CUE

Cues may be either:

1) literal items stored in a file from which the program selects and presents a cue; or

2) rule-generated items which the program generates using syntactic transformations stored as program statements.

A simple example of a rule-generated cue might be a drill program in which the student practices the numbers between zero and one million. Since every language has rules for constructing these numbers, the basic elements could be stored as literal items in a file and the rules by which the elements are combined to form larger numbers could be included as statements in the programs. Many aspects of elementary grammar can be handled this way. However, the more complex the grammar objective, the more difficult it becomes to create correct, let alone meaningful, sentences by programmed rules of syntax.

The decision to use either literal items or rule-generated items as cues depends on the grammar objectives and the programming expertise of the author. Rule-generated cues are a reasonable approach to create drills for regular verb forms. For irregular verb forms, especially if they are used in whole-sentence cues, it may be far easier to write hundreds of drill sentences than to find appropriate and programmable rules. But since CAI lessons are usually devoted to very basic grammatical structures and vocabulary, an exhaustive analysis of the grammar of the natural language is certainly not necessary in order to write effective CAI.

ANALYZING THE RESPONSES

To the student, the cue looks the same whether they are fabricated by the computer or read from a file. In answer-processing, on the other hand, the student will perceive great differences in the way his or her response is evaluated. The student's right answer can easily be verified by any CAI program. The instructor simply specifies one or more "right" answers, taking into account such common mistakes as missing punctuation, capitalization, and spacing. The key, then, to understanding different types of answer-processing is the <u>wrong</u> answer. Clearly, if students produced only right answers, we would not need CAI. Moreover, if the program could only tell the student that an answer is right or wrong, CAI would help very few students. The analysis of the wrong answer is what leads the student to the right answer.

Because answers can be wrong, rule-governed analysis is much more problematical than rule-produced cues. If the student's response consists of only one or two letters (e.g., an adjective ending), the rule is easily constructed. The program scans for easily predictable patterns, such as roots and endings, and then displays a verbal messge to help the student pinpoint the precise nature of the error. Evan at this level of complexity, however, problems arise when the student's answer contains several errors.

At the sentence level, it is not yet feasible to construct rules which account for all cases. At the same time, most instructors will balk at constructing drills where the student types unattached morphological endings. Nevertheless, the combination of rule-generated cues with rule-governed answer-processing is a very powerful means of creating effective CAI lessons.

AUTHORING LANGUAGES

The most prevalent form of answer-processing requires that the author provide a number of "correct" (i.e., anticipated) wrong answers for each question. This method is especially popular because it is supported by several general-purpose authoring languages which aid the instructor in writing lessons. These authoring languages provide standardized question and answer formats with special features for examining segments of the student's response. For example, using the "partial answer-processing" features, an author can check for the presence or absence of right or wrong character groups. Each anticipated wrong answer is tied to a specific verbal diagnosis which is relayed to the student when his response matches the anticipated answer. One or a series of such messages leads the student to the correct answer. Each drill item acts as a small tutorial on the pitfalls of a given set of rules.

Whenever all likely wrong answers can be listed with appropriate messages, author languages are very effective mechanisms for processing the student's responses. The shorter the response, the fewer the variables and combinations which must be taken into account. Problems arise, however, when the response requires longer sentences. And even when all likely wrong answers can be determined, the instructor must also decide which errors to diagnose and in what order. Thus, every drill item becomes a demanding programming task. A further difficulty arises in the leap from detecting an error to describing that error. The more complex the program's analysis of the wrong answer, the more likely that mistyped words will be diagnosed as specific anticipated wrong answers or that students' thought patterns will not conform to the instructor's expectations.

GENERAL-PURPOSE ANSWER PROCESSING

The alternative to verbal diagnosis by rule or by pattern detection is a generalized comparison between the student's answer and correct answer. DASHER uses a method which relies on a graphic, rather than a verbal, diagnosis of the student's errors. The advantage of this approach lies in the speed with which exercises can be written, since each item requires only a question and a single answer. A disadvantage may arise when graphic diagnosis fails to help the student arrive at a correct answer, while a well-executed verbal analysis might have succeeded. Also, the conversational, tutorial feeling of the interaction may be lost.

Adams (1968) describes one of the earliest and most extensive projects using this graphic diagnosis approach, the SUNY-Stony Brook German program. Decker (1976) and Allen (1972) have also used approaches which mark the correct answer in various ways as a means of diagnosis. Nelson 91976), however, calls this approach into question, favoring rule-governed analysis because if its anticipated long-term power to generate a virtually unlimited number of items.

The above description of approaches to drill programming suggests the current limits of foreign language CAI. These techniques can be combined with tutorials, testing, individualized instruction, computer-managed instruction, and various other media, but the limits still remain those imposed by the computer's modest ability to handle natural languages and to compensate for human error.

FURTHER READING

The following references are recommendations for further reading in the field. They are meant to help put DASHER into the wider context of possibilities for using technology in language teaching.

Adams, E. N., H. W. Morrison, and J. M. Reddy. "Conversation with a Computer as a Technique f Language Instruction." The Modern Language Journal, 52 (1968), 3-16.

Allen, John R. "ELSE at Dartmouth: An Experiment in Computer-Aided Instruction in French." The French Review, 44 (1971), 902-912.

Allen, John R. "Individualizing Foreign Language Instruction with Computers at Dartmouth." Foreign Language Annals, 5 (1972), 348-349.

Birdsong, David. Computer-Assisted and Programmed Instruction in Foreign Languages: A Selected, Annotated Bibliography." CAL-ERIC/CLL Series on Languages and Linguistics.

Bernhardt, Lewis. "Computer-Assisted Instruction in Russian-Language Courses." Russian Language Journal, 27 (1973), 18-26.

Bernhardt, Lewis. "Computer-Assisted Learning in Russian: Some Notes on Work in Progress." Russian Language Journal, 29 (1975), 71-84.

Curtin, Constance, et al. "Teaching the Translation of Russian by Computer." Modern Language Journal, 56 (1972), 354-360.

Decker, Henry W. "Computer-Aided Instruction in French Syntax." Modern Language Journal, 60 (1976), 263-267.

Fox, J. "Computer-Assisted Learning and Language Teching." Audio-Visual Language Journal, 10 (1972), 103-105.

Hart, Robert S. "Language Study and the PLATO System," Studies in Language Learning,

Hope, Geoffrey. "Elementary French Computer-Assisted Instruction." Foreign Language Annals 15 (1982), 347-353. 3 (1981), 1-24.

Kalbouse, George. "Computer-Assisted Instruction in the Teaching of Russian." Slavic and East European Journal, 17 (1973), 315-321.

Marty, Fernand. "Reflections on the Use of Computers in Second-Language Acquisition." System, 9 (1981), 85-98; 10 (1982), 1-11.

Nelson, G.E., et al. "Two New Strategies for Computer-Assisted Language Instruction (CALI)." Foreign Language Annals, 9 (1976), 28-37.

Olsen, Solveig. "Foreign Language Departments and Computer-Assisted Instruction: A Survey." Modern Language Journal, 64 (1980), 341-349.

Pusack, James P. "Answer-Processing and Error Correction in Foreign Language CAI." System 11, No. 1 (1983).

Scanlan, Richard T. "Computer-Assisted Instruction: Plato in Latin." Foreign Language Annals, 5 (1971), 84-89.

Scanlan, Richard T. "Computer-Assisted Instruction in Latin." Foreign Language Annals, 13 (1980), 53-55.

Taylor, Heimtraut F. "Students' Reactions to Computer Assisted Instruction in German." Foreign Language Annals, 12 (1979), 289-291.

Turner, Ronald C. "Carlos: Computer-Assisted Instruction in Spanish." Hispania, 53 (1970), 249-252.

Weible, David M. "Teaching Reading Skills through Linguistic Redundancy." Foreign Language Annals, 13 (1980), 487-493.

Note: Reprinted from the
CALICO JOURNAL
Vol. 1, No. 1
June 1983

A CALI GLOSSARY FOR BEGINNERS

Randall L. Jones

Language teachers who make the effort to become familiar with the computer and its application to language teaching often encounter an irony that they perhaps do not expect: the necessity of learning a new language, or at least a substantial number of new specialized terms in their own language. Many of these terms are acronyms and have no meaning apart from the world of computers, e.g., *ASCII* and *ROM*. Others are English words which have taken on a new meaning, e.g., *boot* and *menu*.

The following list is intended to assist the language teaching specialist to become familiar with some of the more common terms used with the microcomputer and especially with computer-assisted language instruction. The list is admittedly not complete. Furthermore, readers may discover that terms used in some of the definitions are themselves in need of further clarification. We hope, however, that the glossary will at least help you become more familiar with the jargon that is used by your colleagues who are conversant with the computer and its function.

Algorithm—A detailed set of logical instructions for accomplishing a task. Computer programs are usually based on algorithms.

Applesoft™—A version of the BASIC programming language implemented on the Apple computer (See "BASIC," "Integer BASIC"). Most Apple computers purchased during the past three years come with Applesoft.

Array—A structured set of data items which are related to each other and which have a common name. The individual members in the array can be accessed using the array name and subscript numbers. For example, the array PREP may have as members the English prepositions, *after, at, behind, by, for, from, in,* etc. PREP(1) would refer to *after*, PREP(2) to *at*, etc.

ASCII (American Standard Code for Information Interchange)—The most common method used for encoding characters (alphabetic characters, numbers, punctuation marks, etc.) in a way that can be understood by the computer. A Standard ASCII character set has 128 characters. The individual characters may, however, differ from one application to another, depending on the purpose for which the character set is being used.

BASIC (Beginner's All-Purpose Symbolic Instruction Code)—A programming language which is available on virtually every microcomputer. It uses standard English words in its instructions and is designed to be relatively easy to learn and use.

Bit (binary digit)—The memory of the computer can recognize only two numbers, viz. 0 and 1. This is simulated electronically by using a device such as a switch which is either off (0) or on (1). A single bit has only two possible states. A series of two bits, however, can have up to four states, i.e., 00, 01, 10, 11. A configuration of eight bits can have up to 256 states (2 to the power of 8), and thus is able to represent all upper and lower case letters of the English alphabet, single numbers from 0 to 9, all punctuation marks, as well as many other special characters. Microcomputers are often referred to as 8-bit or 16-bit machines, depending on whether the memory word is made up of eight or sixteen bits.

Boot—Early computers could be started up only by typing in a rather long list of special instructions. Later these instructions were programmed permanently within the computer, and it was only necessary to type in one or two instructions or merely to turn on a switch. The computer was thus said to be able to "lift itself up by its own bootstraps." From this came the expression "to boot" or "to boot up" the computer, i.e., to start it up and bring it to the point that it is useable. The word "boot" has nothing to do with kicking the computer, either figuratively or literally.

Dr. Randall Jones is the Director of the Humanities Research Center and Professor of German at Brigham Young University. He is Director of the German TICCIT Project. He received his Ph.D. from Princeton University and is the author of numerous papers and articles on CAI.

Branching—The process of interrupting the sequence of instructions in a program to go to a different point in the program. For example, in a language lesson the program might branch to point x if a student response is correct, but to point y if it is incorrect. Branching allows lessons to be more versatile and tailored to individual needs.

Bug—A logical error in a program. (See "Debug".)

Byte—A configuration of bits sufficient to represent a functional piece of information, e.g., a letter of the alphabet or a number. The memory of a microcomputer as well as external storage devices such as disks are measured in terms of kilobytes (e.g., 64K = 64,000 bytes) or megabytes (e.g., 10 meg = 10,000,000 bytes). These numbers are in reality not as exact as they may appear, but in fact have been slightly rounded off.

CAI—Computer-Assisted Instruction.

CALI—Computer-Assisted Language Instruction.

CALL—Computer-Assisted Language Learning.

Card—A special purpose circuit board which can be plugged into a microcomputer. For example, a 16K RAM card consists of a number of memory chips and is used to increase the internal memory of the computer. A disk controller card is an interface between the computer and the disk drive(s).

Chip—The invention of the microchip or semiconductor is the single event that made the microcomputer a reality. The staggering reduction in the size of the electronic circuitry also made possible a reduction in cost. Chips in a microcomputer are generally used either to store information temporarily (see "RAM") or to store pre-programmed information used by the computer see "ROM").

Communicative—Those exercises which require the student to become involved in an interpersonal and meaningful exchange of "new" information (Widdowson, 1978).

Course Authoring Language—A computer programming language which is specially designed for writing CAI lessons.

Course Authoring System—Special software designed for writing CAI lessons. Unlike a programming language or course authoring language, a course authoring system requires the lesson author simply to write in the essential text for each lesson. The formatting and packaging is then done automatically.

Courseware—A set of lessons designed to be used with a computer. The name was originally used to differentiate between the instructional material and the software that was used to write and run it.

CRT (Cathode Ray Tube)—A television-like screen used to display information from a computer. A CRT is generally referred to as a monitor today.

Cursor—A special mark which appears on the moinitor screen, used to indicate the position of the next keyboard entry.

Debug—Locating and correcting a logical error in a computer program.

Dot-matrix printer—A computer-controlled printer which forms characters and graphic images by a pattern of points or dots. Each character is defined by a two-dimensional grid or matrix. For example, a grid of 7 x 9 has 63 dots. The principal advantages of a dot-matric printer are low cost, speed and flexibility, i.e., the fact that it can print a variety of characters and graphic patterns. The disadvantages is that the quality of the characters produced by most inexpensive dot-matrix printers is not as good as those produced by a conventional printer.

Drill and practice—A CAI lesson which is an adaptation of a workbook exercise, e.g., fill in the blank, select the correct answer, etc.

Feedback—An evaluation of a student's response to an item in a CAI lesson, to the work done on an entire lesson or to the student's progress over a period of time. It may consist of a simple positive or negative statement, e.g., "That's right!" or "Sorry, try again." to a more general analysis of the student's work, e.g., "You are doing very well on this lesson. I suggest that you go on to Lesson 7C now."

File—A collection of logically related records which are usually stored on a disk, e.g., a program, a data set, a lesson, etc. A file has a file name, which is used to access it, and in some cases a file type or extension to identify what kind of file it is.

Firmware—In certain cases it is convenient to record a program (software) on a ROM chip (hardware) and plug the chip into the circuitry of the computer. For example, the entire set of instructions which runs the Applesoft™ language in an Apple II Plus microcomputer is on a ROM chip.

Floppy disk—A thin, round flexible mylar disk used for storage of microcomputer programs and data. Most commercially available software comes on floppy disks.

Graphics—The generation of shapes or pictures from a computer program which usually appear on the monitor screen, but which in some cases can be printed out. Some CAI lessons use computer-generated graphics to illustrate certain points.

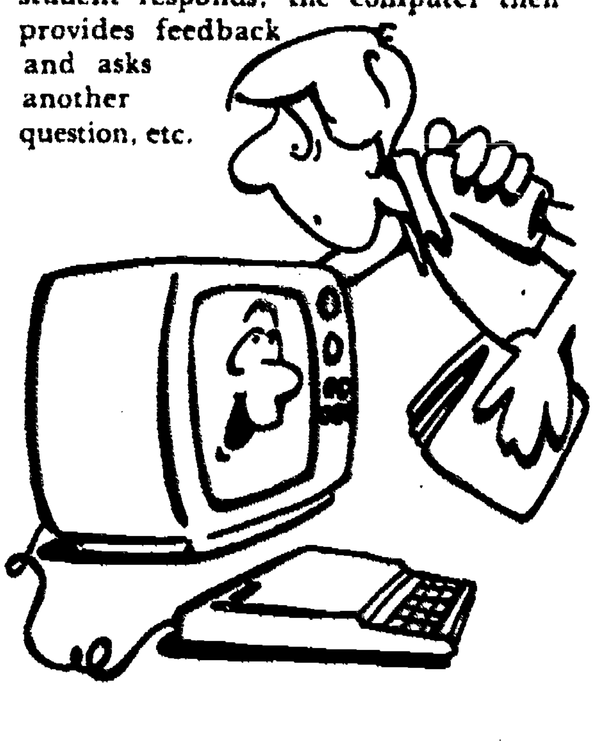
Hard disk—An external storage medium which is similar to a floppy disk, but which has a much larger capacity (usually measured in "megabytes"), transfers data at a much faster rate, and which is considerably more expensive.

Hardware—Any part of the computer which is tangible.

Input—(noun) Information or data which is "put into" the computer; (verb) Putting information or data into the computer by means of a terminal keyboard, a magnetic tape, a magnetic disk, etc.

Integer BASIC—A version of the BASIC programming language which was implemented on the Apple™ computer when it first came on the market. It is rarely used for programming today, although many programs still exist that were originally written in Integer BASIC.

Interactive—The ability for a user to be able to "converse" with a computer. A CAI lesson, for example, is interactive. The computer asks a question, the student responds, the computer then provides feedback and asks another question, etc.



Interface—An electronic device used to connect the computer to a piece of peripheral hardware such as disk drive, a printer, etc.

K—An abbreviation for *kilobyte*. (See "Byte.")

Load—Transferring a program from an external storage device to the memory of the computer.

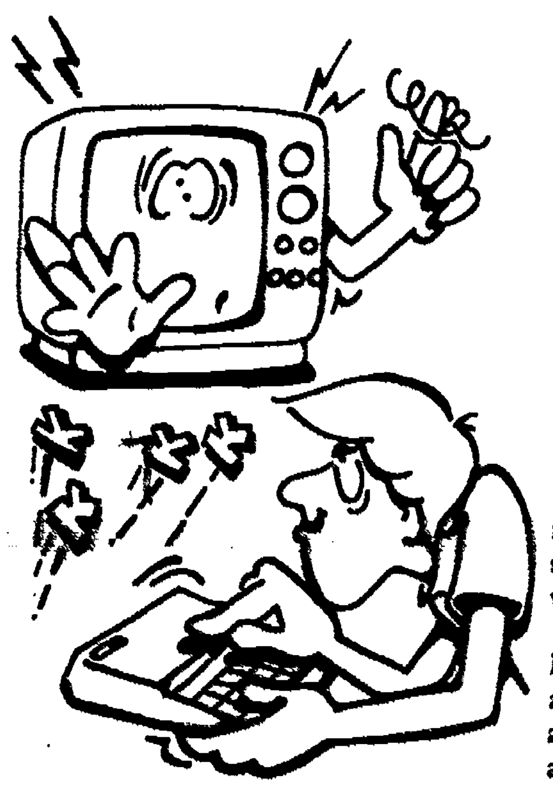
Mainframe computer—A large stationary computer which has an extensive amount of memory (usually more than two megabytes) and disk space and is able to perform several different tasks simultaneously. A mainframe computer is usually shared by a number of users. The stereotype image of a machine with banks of blinking lights and rows of spinning tapes is usually associated with mainframe computers.

Meaningful Exercise—Those exercises which require the student to know the meaning of key words in the exercise item in order to respond correctly.

Mechanical Exercise—Those exercises which focus on one or more discrete elements of language (linguistic forms) without regard for the meaning inherent in these forms of elements.

Meg—An abbreviation for *megabyte*. (See "Byte.")

Memory—The part of the computer which can store directly accessible information. A standard memory size for most microcomputers today is 64K, although some have as little as 2K and others have as much as 512K (see "RAM").



Menu—A list of choices available to the user, which is usually displayed on the monitor screen. For example, a CAI course may have several lessons. The user can usually select the desired lesson by pressing a number or letter. Many programs are said to be "menu-driven," i.e., the user simply makes decisions as new menus are provided, until the desired point in the program is reached.

Microcomputer—A small portable computer which is relatively inexpensive and which is designed for a single user. Most microcomputers can perform only one task at a time.

Minicomputer—A medium size computer (roughly the size of a refrigerator) which falls between the microcomputer and the mainframe in terms of memory size and disk space. It is generally used for a specific purpose, e.g., research, accounting, multiple user word processing, etc.

Modem (modulator/demodulator)—A device which converts computer data to a signal that can be transmitted over a telephone line. It can likewise reconvert a signal coming into a computer through a telephone line so that it can be understood by the computer.

Monitor—A television-like screen attached to the computer which is used to display information.

Pascal—A high-level structured computer language which is available for many microcomputers, minicomputers, and mainframe computers.

Peripheral—A device which can be attached electronically to a computer, usually for input and output.

PILOT (Programmed Instruction Learning or Teaching)—A programming language which is designed for writing CAI lessons. This type of computer language is often referred to as a course authoring language.

PLATO™ (Programmed Logic for Automatic Teaching Operation)—A CAI system developed at the University of Illinois and now owned by the Control Data Corporation. PLATO is available by remote access from virtually anywhere in the world.

Program—(noun) A series of computer instructions designed to accomplish a specific task or series of tasks; (verb) To write a computer program.

RAM (Random Access Memory)—Memory in the computer which can be accessed directly. Each byte in memory is addressable, i.e., the computer knows at all times what kind of information each of the bytes currently holds. The computer can read, erase or alter the data in each of the individual bytes directly. RAM in most microcomputers is "volatile" memory, i.e., it is lost when the power to the computer is shut off.

Record—One line or unit of data in a file.

ROM (Read Only Memory)—Preprogrammed memory in the computer which contains special instructions for the basic operation of the computer. The existence of ROM in a computer is transparent to the user, as it is not available for temporary storage. ROM is not volatile, i.e., it remains even after the power has been shut off.

Run—To begin, i.e., to put into motion, a program on the computer.

Software—The programs or sequence of instructions which tell the computer what to do.

TICCIT™ (Time-Shared Interactive Computer-Controlled Information Television)—A CAI system developed at Brigham Young University by the MITRE Corporation, and now owned by the Hazeltine Corporation.

Transportability—The ability of software and courseware to be developed on one computer, then used on another one.

Tutorial—Courseware which provides language instruction to the student, rather than merely giving exercises.

User-friendly—The ease with which a computer program or lesson can be used by one who is not familiar with it. For example, if the user makes an error the program will provide information about how to correct the error.

Winchester-type hard disk—A fixed hard disk with a large memory capacity (usually from 5 to 40 megabytes) which transfers data at a rate much faster than floppy disks. The name Winchester is not a brand name, but has become a generic term for virtually all hard disk systems used with microcomputers. The legend is that IBM once developed a hard disk system that would store 30 megabytes of memory on 30 tracks. It was referred to as the 30-30, thus the name Winchester.

Note: Reprinted from the CALICO JOURNAL
Vol. 1, No. 4
March 1984

COMPUTER-BASED FOREIGN LANGUAGE INSTRUCTION IN ILLINOIS SCHOOLS

Robert L. Blomeyer, Jr.

Mr. Robert Blomeyer is a Ph.D. candidate in the Educational Policy Studies Department of the College of Education at the University of Illinois, Urbana/Champaign. During the past three years he has worked as a Research Assistant for the U of I Language Learning Laboratory. During his work there he has been the Lesson Design Coordinator of a PLATO component to a USDE Title VI project producing Spanish for Agriculture Purposes teaching materials, co-participant in a series of computer literacy workshops for Illinois foreign language teachers, and most recently, author of a position paper for the Illinois State Board of Education on the use of computer-based foreign language instruction in the state of Illinois.

PREFACE

The information contained in this report is based on findings made during an ongoing piece of educational research entitled Computer-based Foreign Language Instruction in the State of Illinois. It should be understood that because the study is not completed, present conclusions are subject to revision prior to final completion of the project. Names of the schools involved and the particulars of events observed will not be revealed to protect the privacy of the individuals and institutions involved.

The research is being supported by the Language Learning Laboratory of the University of Illinois at Urbana-Champaign and the Illinois State Board of Education, where the author is working as an intern in the Program Planning and Development division under the direction of Mr. Paul Griffith. The contents of this report are the responsibility of the author and no official endorsement by either the Language Learning Laboratory or any division of the Illinois State Board of Education should be inferred.

Microcomputers have suddenly invaded classrooms nationwide. The National Council on Educational Statistics (NCES) 1982 survey indicates that there are approximately 96,000 microcomputers in public schools (NCES, Early Release, 1982). This is nearly triple the 36,000 microcomputers that were present in the schools in the fall of 1980 (NCES, Early Release, 1982). More recently, a national study of educational technology in the U.S. reported that 53 percent of all schools had at least one computer as of January 1983 (Center for the Organization of Schools, 1983). This massive influx of hardware into the nation's schools creates a wealth of technological resources which can be used in addition to more traditional methods of instructional delivery.

The humanities are no exception to this generally increasing utilization of computers. Although, a limited amount of effective courseware exists for use in some subject areas, there now exists a significant body of commercial software available to implement aspects of foreign language instruction in the microcomputer-based medium. Language teachers in some school districts are beginning to incorporate these commercial materials into their syllabus design. In some cases the teachers are becoming actively involved in the design and production of their own foreign language courseware. But in the State of Illinois, these teachers are still in a minority. The vast majority of language teachers are trying to determine the potential and limitations of the medium for foreign language, English as a second language, and bilingual instruction.

Language Instruction On Mainframe Computer Systems

Existing studies on integrating CAI with language teaching are based primarily on research with mainframe computer systems rather than with microcomputers that are now being used in public schools. The mainframe is a computer system with a large central memory to which many terminals are attached. These terminals work on a time-sharing basis off the mainframe. As the name implies, the microcomputer is a small, free-standing computer system with a more limited memory than the mainframe and it can generally be used only by one person at a time.

Lesson designers now writing in professional publications tend to give general descriptions of the lesson content and of the structural and technical aspects of the program design and hardware configuration (including details of any audio or speech synthesis devices used to supply aural stimuli to the student). They give little information on strategies for the best ways to use CAI software in second language curricula.

The strengths of mainframe CAI language lesson materials all relate to the ability of the materials to deliver self-pacing, individualized lessons consisting primarily of vocabulary and grammar drills. These require student responses to written stimuli presented on a line printer, a CRT (cathode ray tube) monitor, or a plasma panel (as in the PLATO system). In some programs, students respond to an aural stimulus produced by various audio or audio-visual and speech synthesis devices (Van Campen, 1981; Hart, 1981; Marty, 1981; Kidd and Holmes, 1982). Instant feedback on the appropriateness of a given response can be provided to the student. In addition, more advanced diagnostic routines have been developed to provide remedial vocabulary and grammar information (Marty, 1981; Levin, 1981; and Barson et al., 1981). A possible future development of this technique may be the use of intelligent programs and parsing routines that analyze written free expression (Marty, 1981; Hart, 1981). The greatest limitation of CAI material for language instruction is the inability of the technology to measure oral language production. The majority of these language CAI programs use drill and practice routines that are similar to the type usually associated with the skill learning practices of the audio-lingual method. At present, second-language CAI has limited possibilities for strategies intended to increase students' oral communicative competence. According to Van Campen, Markosian, and Seropian, the principal drawback to CAI language instruction is that the computer cannot evaluate oral input. It cannot hear (Van Campen et al., 1981). This will continue to be the case until CAI gains the capabity to measure or analyze speech.

Some simple speech recognition devices have been developed using powerful mainframe computers (Electronics, 1980), and several companies are marketing speech recognition devices which can be used with micro-computers and are capable of recognizing approximately 40 - 80 words. It is not reasonable, however, to anticipate that in the near future a hardware system will be developed that can respond to natural speech in a dialogic manner (Marty, 1981).

There now exists a significant body of commerical software available to implement aspects of foreign language instruction in the microcomputer- based medium.

Despite the limitations of computers for oral language instruction, it has been demonstrated that CAI can be effectively delivered with recording devices to document student oral performance for self-evaluation or for evaluation by the instructor (E-Shi Wu, 1981). If teachers use the computer to introduce and practice vocabulary and grammar information, they are freed from tutorial time for concentration on oral skills and other activities that cannot effectively be simulated on a computer (Barson et al., 1981; Van Campen et al., 1981). It appears that the most effective means of integrating computers into language instruction is to combine these supplementary and remedial uses of CAI with a classroom teacher (Kidd and Holmes, 1982).

Language Instruction on Microcomputers

As indicated previously, most discussions of language CAI software have been concerned with mainframe computer systems. Information on the use of microcomputers in the public schools is limited primarily to data on the numbers located in the schools and their general use within the curriculum, e.g., computer literacy, basic skills enrichment, administrative uses, etc. (NCES, Early Release, 1982). A recent survey by Marketing Data Retrieval Services indicates that microcomputers present in the surveyed schools are used for language instruction in only two percent of all cases (MDRS, 1982). According to James Pusack of the University of Iowa, widespread adoption of CAI in foreign language instruction has been hampered by several problems, including lack of equipment, lack of computing skills, suspicion of technology, and a lack of appropriate computer programs (Pusack, 1982, p. 64).

Little comprehensive information is available on the specific contents of available microcomputer programs for second language studies. An existing resource for information on computer-based foreign language courseware is found in the February 1982 Newsletter of the Northeast Conference on the Teaching of Foreign Languages. In a short, but highly informative article, John Harrison gives a listing of all the commercially produced software packages currently available, with substantive information on those he has personally reviewed. The article lists approximately 83 packages and lessons from a variety of sources. Among these are many drill and practice format exercises and a smaller, but significant, number of simulation type programs involving use of the target language in order to participate in the machine-simulation of various complex scenarios.

Since the publication of the Harrison article in the Northeast Conference Newsletter, a series of reviews that have been written by Gerald Culley and his students at the University of Delaware has been included in issues of the publication (Newsletter No. 14, August 1983). Professor Culley and his students were also responsible for publication of guidelines for software evaluation and a series of software reviews that were the product of an NEH summer institute at U of Delaware in the summer of 1982 (Culley and Mulford, 1983).

Most recently, a resource bibliography has been published by the National Center for Bilingual Research entitled: *Microcomputers in Bilingual and Foreign Language Instruction: A Guide and Bibliography* (National Center for Bilingual Research, June 1983). This publication contains a brief but complete introduction to concepts and terms useful for describing CAI in a manner relevant for language instruction. Its main content is a two hundred page inclusive bibliography of software resources for foreign languages, bilingual/multicultural education and English as a Second Language. This is an annotated bibliography of the courseware providing information on hardware requirements, prices and availability. The information does not include qualitative assessment of either the content or design of the programs. Assessment of the suitability of the materials is left to the reader. A listing of schools currently using CAI in language classes is provided for persons who wish to contact indicated resource persons for recommendations and suggestions. The publication plans to update the bibliography and other databases as additional information becomes available.

Beyond the reviews by Professor Culley and his students, evaluations of existing CAI software are limited to (1) data on student performance (This is seen occasionally in a control/experimental group design using CAI as the dependent variable. See: Van Campen, 1981 and Van Campen et al., 1981) and (2) articles by authors and designers on the accomplishments of the courses. At the present time few studies report on the human factors of CAI use or on the broader impacts of CAI on the process of classroom instruction. A recent article by McCoy and Weible (1983) provides some speculation on some of the human aspects of microcomputer use in foreign language teaching. The authors are foreign language teachers with experience in both the use of the CAI medium and the production and use of video materials on magnetic tape and videodisc. They conclude that microcomputers *might* be able to free language teachers from routine classroom work for more communicative activities. McCoy and Weible suggest that a better approach to language CAI would be to incorporate the use of video materials with microcomputer courseware. Courseware based on videotapes or videodisc recordings of natural language dialogues could provide training materials to boost oral comprehension and provide contextual cueing of a sort that has previously been possible only in actual conversational practice. Unfortunately, such materials exist only in prototype form and will probably remain too expensive for use in most public school foreign language programs (V. Stevens, 1983).

A possible future development may be the use of intelligent programs and parsing routines that analyze written free expression.

Although instruction involving oral conversation is presently limited by the high cost or the unavailability of the necessary technology, there exists another possibility for integrating microcomputers into instruction in foreign language communication skills. Readily available communications packages provide both the hardware and software for sending and receiving written communications or electronic mail. These tools are already being used by elementary school students in various parts of the United States to facilitate the exchange of computer-mediated discourse.

Preliminary evidence indicates that computer-based discourse could have linguistic characteristics more like oral communication than traditional letters (Scollon, 1982, p. 19). Although at present there is no known application of this facility for electronic mail to languages other than English, the only element present lacking is the special characters necessary for correct spelling in the given languages (accents, etc.). It is easy to speculate that with the availability of compatible communications packages in various parts of the world, student-to-student electronic communications might well encourage the development of communicative competence.

Existing foreign language microcomputer courseware primarily exploits the presentation of written text on a computer screen in instructional formats that rely heavily on drill and practice exercises. Some audio devices have become available in recent months, but these are expensive and limited to the vocabulary stored on the audio tape or in the speech synthesis device (Hertz, pg. 24). Although some other formats for foreign language courseware do presently exist (simulation type programs and language games), they are primitive, few in number and of questionable value for classroom language learning.

Since at present the greatest value of computer use in foreign language instruction seems to be as a supplement to the teacher, he or she must play a central role not only in controlling the integration of software into the language studies curriculum (Putnam, 1983) but also in the design of the courseware (Russell, 1983).

Two central points seem to emerge from the existing body of literature on computer-based language instruction. The most frequent recommendation common to the sources reviewed is that foreign language teachers should have control over the vocabulary and specific grammar items in foreign language courseware. This factor is viewed as critical for the successful integration of the computer-based activities into the total syllabus. The second point is that foreign language teachers must become knowledgeable about the alternatives available for implementing computer-based foreign language instruction. Individual teachers must attain a reasonable level of computer literacy before they can begin to explore the possibilities and actually preview materials that might be useful to them.

However, at present there appears to be no single definition of the term computer literacy appropriate to the teachers considering CAI. Indeed, definitions of literacy may be as numerous as the software programs that perform a variety of instructional and non-instructional tasks (Levin and Souviney, 1983). One of the more common conceptions of computer literacy views it as a dynamic range of possibilities rather than a single definition. This viewpoint has been used by Robert Hertz to describe four levels of computer literacy for language teachers:

1. the computer-using teacher,
2. the nonprogramming author of courseware content, 3.
3. users of authoring systems, and
4. the teacher-programmer (Hertz, pp. 14-19)

It seems then that the definition of computer literacy is dependent on the context of the individual school and the structure of its foreign language teaching program. Observation of foreign language teaching programs where the foreign language teachers are actually using microcomputer-based courseware suggests that the term computer literacy requires a functional definition to match the context in which it is being used.

Observations on CAI and Foreign Language Teaching

Two of the three schools participating in this research are very large suburban high schools in the Chicago area. In the following narrative, these schools will be referred to as school one and school two. A third high school (with an enrollment close to the state's average enrollment of about 2000-2500) is located in a downstate school district.

It is clear that the limited number of situations studied cannot be representative of all the school situations possible in a state as large and diverse as Illinois. However, it is the considered opinion of the author that many foreign language teaching situations at the secondary school level are similar enough to justify some methodological conclusions and limited recommendations.

1. CAI can be incorporated into FL classrooms at both the beginning and advanced levels of language instruction and used as a supplement to a variety of strategies and approaches.

The primary factor in the foreign language teachers' decision to use CAI to supplement classroom teaching was their conclusion that some part of the instruction was suitable for delivery by the computer. In each case observed thus far the courseware used was designed or modified by the teacher or by a cooperating teacher in the same school. As a result the contents of the computerized lessons have been very closely tailored to the vocabulary and grammar particulars of the individual classes.

A common element in the teaching style of the teachers observed to date has been their use of a wide variety of learning activities and formats in their classes. These include both oral and written practice on vocabulary, grammar and cultural aspects of an individual lesson's content. The computer-based lessons that have been observed in use with these more standard classroom routines are primarily drill and practice on those same areas. Computer-assisted testing was used in an Advanced Placement French class, as a simulation of the grammar test that the students would be encountering on the placement test.

The principle drawback to CAI language instruction is that the computer cannot evaluate oral input. It cannot hear.

In beginning and intermediate Spanish and German classes, drill and practice on vocabulary and grammar provided a supplementary exercise on materials encountered in other classroom contexts. These teachers felt that vocabulary review was accomplished neatly and efficiently by using vocabulary translation exercises on the microcomputers. Overall, the teachers observed seemed to be using the microcomputer-based lessons as just another of a variety of strategies within a comprehensive syllabus including passive and active skills in both oral and written contexts.

(FN: It should be noted that most of the foreign language teachers in the State of Illinois will probably need to rely more on commercially available software. Use of commercial software can imply an entirely different set of conditions regarding the control of lesson content by the teacher.)

2. CAI appears to work best as a FL classroom resource if the students have previously been introduced to educational computing in other contexts.

Two different situations have been observed: in school one, limited microcomputer resources are available in a single centralized location primarily used for math, business applications and programming, whereas in school number two, an ample number of microcomputers are available in several sites where they can be easily used for educational computing in all subject areas. School two has a course requirement for all incoming freshmen in the use of microcomputers for instructional purposes, giving them some basic concepts of programming, and advises them of the vocational choices in the computer technology area and the particular special training that could be necessary to pursue these options.

In school one, some students could complete their schooling without using a microcomputer for any purpose. Language classes in this school, which lacked any firm policy on student computer literacy, needed much more in-class instruction on the basics of using the microcomputers themselves. The students in school one were observed in conversations with the teacher that sometimes demonstrated unrealistic expectations of the microcomputer's contribution to learning, sometimes an outright aversion to their use. In these cases the foreign language teacher had to work individually with the student to develop more productive attitudes toward the use of the microcomputers or provide individualized assignments that could be completed using paper and pencil exercises or reading.

In school two, it was not necessary to take class time to instruct the students on the use or potential of the microcomputer resources. The students were all able to use the computer-based learning programs effectively. When present with classes in the computer labs, teachers supervised the use of the materials and answered questions about lesson content. This allowed the teachers to spend more time observing the work of all the students and evaluating student progress. Student evaluations also provided valuable insights about the design and content of the courseware to aid in the revision and updating of lessons already in use.

3. Although integration of computer-based materials into a syllabus is easier and more efficient when they have been produced by the teachers who use them, the design and implementation of computer-based foreign language materials is technically demanding and time consuming.

As previously noted, the cases studied showed a high degree of integration between the syllabus materials and classroom routines, but the teachers involved had paid a high price in terms of their time. In shcool one, the teacher had voluntarily taken a reduction to half-time status, in order to have the time to study programming techniques and to produce materials. In school two, the district had made a sizeable investment in an ambitious inservice training program. Initially, inservice training was provided to all teachers in the school. These first experiences were followed by voluntary summer workshops in which the participating teachers worked with student programmers to develop and implement materials to be used in the following school year. The participating teachers were paid a stipend to subsidize their attendence. After the initial workshops they used summers, weekends and evenings to write and revise instructional programs. The teachers who continued to work on materials development and teach a full schedule of classes simultaneously found that the time necessary to maintain existing lessons and work on new projects cut deeply into their personal time. They gave this time willingly because of their professional pride but it was evident that this commitment was taking its toll. All of them remarked in interviews and conversations that they did not consider it to be reasonable for administrators to assume that teacher produced materials would be a primary source of instructional software in most schools.

In school two, some of this demand for time and specialized skills was filled by providing computer aides and programming consultants to assist with the development of new materials in high priority areas. However the maintenance and revision of existing materials was still the job of the individual teachers. These aide and programmer positions seemed to be filled by persons without professional education background. Their salaries were therefore not competitive with industry and the positions seemed to have a high turnover rate. Additionally, these individuals lacked the specific skills in instructional design which makes possible the production of more sophisticated and useful courseware.

Computer-based discourse could have linguistic characteristics more like oral communication than traditional letters.

4. Most existing computer-based foreign language teaching materials have design, content, and technical shortcomings that may make them unusable by a large number of foreign language teachers in the public schools.

Although it might seem logical to assume that the use of commercial software is less time-consuming than local software development, this is not always the case. As foreign language courseware continues to come on the commercial market, the teachers participating in this study find that the task of pre-screening and selecting materials for examination is both time-consuming and complex, and is moreover made more difficult by the fact that individual distributors often are reluctant to send preview copies on approval for fear of software piracy.

When computerized materials are made available for preview, a thorough review for content and technical implementation requires large amounts of time and an understanding of the pedagogical implications involved in the design of computer-based instructional materials. In previewing software, teachers found errors in the grammar and spelling of the material and found much of it to be of questionable relevance. In one school the teachers felt they had to edit the commercial programs to remove the errors. However, this solution requires a high degree of technical sophistication in programming and in many cases the programs are protected to make modification of the computer code nearly impossible.

The problem of software compatibility is also a complicating factor. One school district owning Tandy microcomputers found that much of the available foreign language software could not be used on their particular hardware system. At the present time, there appear to be more foreign language software programs available for Apple computers than for any other variety. Although translations of these programs for other hardware are beginning to appear, schools will have to consider this problem in software selection until the microcomputer industry agrees on uniform standards for hardware and software design.

5. Two principal design strategies should be given priority in the development of computer-based language instruction: a) flexible Drill and Practice utility programs which gave the teacher control over program content, and b) comprehensive materials available as a supplement to major foreign language series textbooks.

A) Many of the teachers cooperating in this research were beginning to implement flexible open-ended utility programs that allow the specific vocabulary used in a drill or tutorial program to be varied easily without changing the structure of the program itself. In both schools the teachers were trying to write programs of this nature but lacked specific technical knowledge in design or programming. Although some commercial programs of this sort are available, their cost is generally higher than that of other available foreign language courseware. Until prices drop or schools make much more money available for the purchase of foreign language courseware, they are not likely to be widely purchased.

B) Teachers also discussed the need for the publication of foreign language textbooks accompanied by fully developed computerized materials keyed to the individual chapters and lessons. They agreed that teachers like themselves do not have the time or the resources to begin development of materials of this scope. Unless such comprehensive materials do become available, the large number of language teachers who rely upon published instructional materials will never attempt to use even the most flexible computer-based lessons.

At present I am aware of only one company that has produced a series of books with fully implemented computer-based supplementary materials. This was a French series. The foreign languages department chairperson from one of the cooperating schools had seen this series but was not, as far as I know, planning to purchase it. The largest textbook publishing houses evidently have not as yet made the decision to make the investment necessary to begin production of computer-based foreign language materials of this type.

6. Foreign language teachers working in school districts where funding for software acquisition is scarce might be prevented from obtaining financial assistance for software acquisition from federal sources (Chapter II Block Grant funding for Educational Improvement) by district-level determination that these funds are reserved for the support of computer literacy, i.e. math, science and computer technology.

An example of this restriction of federal funding was encountered in the third cooperating school in a downstate school district. In this case a foreign language teacher submitted a Chapter II minigrant proposal to the regional superintendent's office. The proposal requested funding to purchase instructional software to use in the teacher's secondary level language classes. The request for proposal had specified that the purpose of the grants was to support school activities related to computer literacy, so the teacher contacted the regional superintendent's office prior to writing the application to ask if the proposed use for foreign language instruction was acceptable, and was told that it was.

Some weeks later however, the teacher was notified by the regional superintendent's office that the grant proposal had been turned down because the proposed use of computer-assisted instruction for foreign language teaching had nothing to do with computer literacy. The regional office had decided on three areas of the curriculum as being of high priority, i.e. math, science, and computer technology. Accordingly, the proposal was given a low competitive ranking because regional specification of priorities had, in effect, provided a restrictive definition of computer literacy that excluded the use of microcomputers for foreign language instruction. This case may be an isolated one but foreign language teachers should be aware of the possibility of such restructions and cooperate in seeking to remove them.

Foreign language teachers should have control over the vocabulary and specific grammar items in foreign language courseware.

7. Whether foreign language teaching courseware is locally produced or commercially obtained, the results of its use could be influenced more by local school practices regarding the management of available microcomputer resources than the materials themselves.

A clear example of the influence exerted by management of computer resources in the local shcools is seen in the comparison between the interpretations of the phrase "instructional use of computers" in school one and school two. In the case of school one, a more limited number of microcomputers made it necessary to establish priorities on the use of equipment. Because math, programming and business applications were given a high priority for use of the available resources, the language teacher was only able to schedule the use of the site if these teachers were willing to give up their regularly scheduled times. In school two, all subject areas in the curriculum had access to the microcomputers as a medium for the delivery of instruction.

It is interesting to note that in neither school studied was there a formal statement of policy regarding the instructional usage of microcomputers. The substantial differences seemed to be in the requirement for the basic computer literacy course for incoming freshmen and in the amount of financial support available to purchase microcomputers for open use in the school curriculum and support for the local development and purchase of instructional software.

Other logistical decisions regarding the purchase and set-up of hardware facilities can strongly influence the practices the teachers must follow when using them. Not only do different hardware systems (Tandy vs. Apple, etc.) vary in features and capacity, but their arrangement either as free-standing individual workstations or as members of an instructional network will influence their efficacy. In school one, the microcomputers all had individual disk drives and were loaded with the necessary programs from individual 5¼ inch diskettes. In school two, the student terminals did not have disk drives and were loaded from a master terminal in the front of the computer lab. The duties of the teachers in the two settings were very different for these reasons.

In school one the teacher spent a great deal of time passing out disks to students and later recovering them. In cases where insufficient disks were available for all the students present, the students were often required to stop their work and borrow a disk to reload part of a program before they could continue. The teacher in school one had substantially less time to observe the work of the students and actively circulate around the computer lab to monitor their activities. Teachers using the networked microcomputers in school two generally had more opportunity to manage the instructional setting. Additionally, the supervision of program loading by the teacher allowed control of sequence and pacing of materials by the teacher and prevented the use of unauthorized materials.

Continued on page 42

An unanticipated spinoff of the requirement for a minimum understanding of microcomputer use for all the students (as observed in school two) was the recreational entry of classroom students into the program code for the purpose of altering the instructional program. This phenomenon was more marked in the case of lower level students who were less serious about their grades and general participation in the class activities. If the students were actively involved in this recreational activity, it kept them from doing their assigned work. On one or two occasions it became quite disruptive in the computer lab setting.

During my second week of observation in school two, the programs in use were revised to prevent the students from altering the programs. It was still possible for them to switch off or abort the program, but this was obvious to the teacher as their progress was monitored and class participation could be graded accordingly. This problem suggests that in cases where there is a local school requirment for student computer literacy, the degree of technical sophistication of the instructional programs should be high enough to prevent them from being altered by the student users.

Summary of the Observations

Foreign language teachers in a limited number of Illinois secondary schools are beginning to use computer-assisted instruction as part of their instructional plan. At present it appears that only in large Chicago-area schools, with the financial resources of high income suburban areas, is computer-based foreign language instruction being systematically used. Downstate school districts are known where the foreign language teachers have access to microcomputer laboratories for their classes, and where they are receiving training in the instructional use of computers. In at least one downstate school district the present budget includes the purchase of computer software. However, it appears that actual use of microcomputers-based instructiona by foreign language teachers is limited to a very few of the largest and wealthiest school districts in the Chicago suburbs.

Even where foreign language teachers are using CAI, only a small portion of the teachers in the school's foreign language department were making use of the available resources. Teachers who were already using CAI were generally very knowledgeable about instructional uses of microcomputers and most were trying to program their own materials.

These teacher/programmers were largely self-educated with some assistance from organized inservice training. Because of the special problems of designing computer-based instruction as distinct from other applications (management, record-keeping, answer judging, etc.) and the additional technical necessity of foreign language characters (accents, ets.), much of what they learned in their initial training experiences had to be adapted or discarded as new techniques were devised by trial and error.

Use of commercial software can imply an entirely different set of conditions regarding the control of lesson content by the teacher.

In all cases studied, the teachers were helping their colleagues to learn about the possibilities of CAI in foreign language teaching. These activities seemed to present a teacher self-service alternative to the other available training. These people are active as presenters at seminars and inservice training workshops that provide introductory experiences for teachers from other area school districts. They also participate in presentations such as those given at the annual Illinois Foreign Language Teachers Association Convention, which are attended by teachers from all over the state.

In the two schools where CAI was being used on a regular basis, the teachers had varying amounts of support from their departments and school administrations. One aspect of computer use that was a common influence on both situations was the scheduling of a central computer laboratory. The local priorities for use and availability of these facilities constituted one of the strongest factors influencing computer use by language classes.

- The unavailability of usable software seems to be the greatest hindrance to the implementation of computer-based foreign language instruction at the present time. Both the local production of software and the review of commercial materials for their eventual acquisition are time-consuming. In the schools observed, school support is available for software purchase but the allocation of these funds within departments was determined largely by the department chairperson in consultation with the teaching staff. Present information indicates that school support for software purchases may not be available in many school districts, particularly in the downstate school districts.

If we assume that foreign language teachers in Illinois will gain access to the use of microcomputers as they are purchased by additional school districts, two things are needed to encourage the thoughtful and appropriate use of these new resources: 1. teacher training on the basics of educational computing, with special attention to the special needs of foreign language teaching; and 2. adequate foreign language courseware availabilty.

Editor's Note: The following recommendations to the Illinois State Board of Education are reproduced for the benefit of consortium members who may have needs or interests in this aspect of CAI.

Recommendations

Recommendation 1:

The Illinois State Board of Education should make training programs available so that foreign language teachers can become familiar a) with educational computing and b) with the specific use of microcomputers for foreign language instruction.

a) The first need can be addressed by the presently existing training opportunities available through the Educational Computer Consortia funded under the statewide Computer Technology in Education Program.

b) The second can be met by making follow-up training experiences available to language teachers who complete the basic computer literacy curriculum. The scope of the existing consortia should be be broadened to include training specific to sub-areas of instruction (i.e. foreign languages) and additional specialized training experiences can be made available through alternative agencies like IFLTA or other professional organizations, institutions of higher education having demonstrated special capacity, or on a consulting basis.

Recommendation 2:

The Illinois State Board of Education should consider the long-range feasibility of amending the certification requirements for foreign language teachers to include basic familiarity with the instructional uses of computers. It is too early to determine the extent to which foreign language teachers in the State of Illinois will make use of computers; however, as computer-assisted foreign language instruction is tested through use, and as additional infromation becomes available on effective strategies for teacher training, teacher expertise in this area will become increasingly important in insuring that the use of computers does not follow the same course as did the use of language labs in the 1960's.

Recommendation 3:

The Illinois State Board of Education should consider the feasibility of a mandate for a required secondary school course in basic computer literacy, which would include introductory experiences in the use of microcomputers for educational purposes. Such an introduction should include applications in the social sciences, arts and humanities (specifically foreign language instruction) as well as the more prevalent uses in math, science and computer technology.

The present review of statewide curriculum requirements contains a recommendation allowing the substitutions of one-half unti of computer technology for an equal amount of required mathematics. However, this alternative does not take into consideration the trend toward increasing use of user-friendly software which makes the use of computers virtually unrelated to mathematics for a variety of purposes (financial business applications, word processing, and nonformal self-instruction). As a broader definition of the term computer literacy becomes more prevalent, the current state recommendation relating computer technology to math instruction, risks the implication of a definition restricting access to computer education resources in other subject areas.

Regional specification of priorities had in effect provided a restrictive definition of computer literacy that excluded the use of microcomputers for foreign language instruction.

Recommendation 4:

The Illinois State Board of Education should take appropriate action to encourage school districts with microcomputer facilities to clarify their local definitions of computer literacy and instructional applications of computers. This clarification should include a stated determination that microcomputer facilities will be available for classroom use by instructional personnel and students in all subject areas. As a policy is established allowing the social sciences, arts and humanities (including foreign language instruction) equal access to the available hardware resources, the school districts should be encouraged to make funding for the acquisition of instructional software available to the individual departments. An effective way to make this funding available (where district support is feasible) would be the creation of a new budgetary line item to be used by the departments as designated discretionary funding. In this way the departments would be encouraged to discuss the role of computer-based instruction and set internal priorities for acquisition of software based on a departmental consensus.

Recommendation 5:

The Illinois State Board of Education should consider providing support to foreign language teachers in Illinois public schools to aquire foreign language courseware. This initiative can be undertaken through two existing programs: a) the statewide Educational Computer Consortia, and b) the Chapter II Block Grant funding for the purchase of foreign languages instructional software. In addition, the state board should consider the provision of direct support for materials development projects undertaken to produce foreign languages courseware.

1) A comprehensive demonstration library of all commercially available microcomputer-based foreign language teaching materials should be acquired by the Illinois State Board of Education and made available on request to specific service regions and member schools in the statewide consortium network for training and demonstration purposes *only*. Access to such materials would greatly facilitate the preview and aquisition of materials and would provide a resource for the training of teachers in software selection. It could also provide the basis for establishing a software evaluation database where foreign language teachers could contribute their assesments of the quality and usability of the various materials, as they are reviewed in consideration for local purchasing.

2) The Illinois State Board of Education should consider the feasibility of making available specific funding for the support of software development projects that would produce quality foreign language courseware which could be distributed to foreign language teachers throughout the state at nominal cost. Although it is unlikely that sufficient resources could be assembled to produce series books with comprehensive supplementary computer-based exercises, the development of easy-to-use utility programs or drill drivers, allowing easy teacher control over specific lexical and grammatical content, might be possible. Commercial sources are not likely to produce these at a price that will make them affordable by many school districts. Subsequent distribution of suc-

cessful materials outside the state might well offset the start-up costs. Such materials-development projects should, however, be undertaken only where there is demonstrated expertise either on the part of participating teachers or in state institutions of higher education (i.e. universities and teacher training schools).

Recommendation 6

The Illinois State Board of Education should consider conducting a statewide inventory of human and institutional resources that are available to advise foreign language teachers on computer-based foreign language instruction. These resource inventories could also contribute to the development and implementation of new foreign language courseware as support becomes available. Such an inventory of resources could be maintained as an online database by the offices of the Educational Computer Technology Consortium network with access provided to the member school districts. One function of such an inventory would be to create a network of classroom teachers, programmers and other related specialists who could benefit from contact with others engaged in foreign language CAI. The teachers participating in this research were all in contact with one another through involvement in local and statewide professional organizations, but a single focused network would facilitate the exchange of ideas throughout the state and help to alleviate the inevitable frustrations that come with being at the forefront of any new educational technique or innovation. As the culture of high technology and the culture of traditional classroom teacher meet in foreign language instruction, mutual support may become one of the real needs of the teachers who become involved.

Conclusion

The recommendations expressed here are based on first-hand observation of foreign language teachers using computer-assisted instruction as one of many teaching strategies. Recent research funded by the National Institute of Education (NIE) has indicated that the major conclusions of this study are representative of research with a broader focus, including aspects of public education outside foreign language teaching and outside of secondary schools (see: Sheingold, Kane, and Endrewelt, 1983). However, this broader research is viewed by its authors as lacking any specific information on how microcomputers will affect educational practice (pg. 431).

The subject specific research upon which these recommendations are based has vividly illustrated one of the final conclusions of Sheingold, et. al.:

> The results suggest that the effects of microcomputers on education, will depend, to a large extent, on the social and educational contexts within which they are imbedded. (ibid.)

The unavailability of usable software seems to be the greatest hindrance to the implementation of computer-based foreign language instruction.

Individuals at all levels of federal, state, and local educational agencies are co-participants with the teachers and students in the events that will ultimately determine the effects of microcomputer use on foreign language learning as well as learning in all other areas of curriculum.

The teachers themselves are the ultimate agents of educational improvement and change. It is the responsibility of the Illinois State Board of Education to provide them with information and other resources necessary to realize the optimal use of computer-assisted instruction.

BIBLIOGRAPHY

Barson, J .Smith, R., Devine, D , Scholl, M., and Scholl, P., University-level CAI in French, in Suppes, P (ed.), *University-Level Computer-Assisted Instruction at Stanford 1963-1980*. Stanford: Stanford University, pp. 685-706 (1981).

Center for the Organization of Schools (CSOS), School Uses of Micro-computers: Reports from a National Survey (Issues 1 and 2), The John Hopkins University: April and June 1983.

Culley, G., and Mulford, G.(eds.), *Foreign Language Teaching Programs for Microcomputers: A Volume of Reviews*, University of Delaware (1983).

Electronics 1980. April 24

E-Shi Wu, P., Construction and Evaluation of a Computer-assisted Curriculum in Spoken Mandarin, 1981, in Suppes, P (ed)., *University-Level Computer-Assisted Instruction at Stanford 1968-1980*. Stanford. Stanford University, pp. 707-716 (1981)

Harrison, J S . Foreign Language Computer Software: What? Where? How Good? *Northeast Conference on the Teaching of Foreign Language: Newsletter*, No 13, February 1983, pp. 26-30.

Hart, R . Languages Study and the PLATO System, in Hart, R. (ed.), *Studies in Language Learning The PLATO System and Language Study* Vol 3. No 1. Champaign-Urbana: University of Illinois (Spring 1981), pp 1-24

Hertz, R., *Microcomputers in Bilingual and Foreign Language Instruction A Guide and Bibliography*, National Center for Bilingual Research, Los Alamitos, CA, June 1983

Instructional Use of Computers in Public Schools, 1982, *NCES. Early Release*. Washington D C September 7

Kidd, M E., and Holmes, G . The Computer and Language Remediation, *Programmed Learning and Educational Technology*. Vol 19, No 3. 1982, pp 234-239.

Levin, D R., Computer-based Analytic Grading for German Grammar Instruction, in Suppes, P (ed), *University-level Computer-assisted Instruction at Stanford 1968-1980* Stanford Stanford University, pp 675-684

Market Data Retrieval Systems, 1982, Westport, Conn

Marty, F . Reflections on the use of computers in second-language acquisition, in Hart, R (ed.), Studies in Language Learning The PLATO System and Language Study, Vol 3, No. 1, (Spring, 1981), pp 25-53.

McCoy, I H., and Weible, D M., Foreign Languages and the New Media. The Videodisc and the Microcomputer, in James, C. J., (ed), *Practical Applications of Research in Foreign Language Teaching* Skokie, Ill : National Textbook Co., 1983, pp. 105-152.

Northeast Conference on the Teaching of Foreign Languages Newsletter, No 14, August 1983

Pusack, J. P., Dasher: A Natural Language Answer Processor, *Conduit*. Vol. 7, No. 1, Spring 1982, pp. 64-66.

Putnam, C. E., Foreign Language Instructional Technology The State of the Art. *CALICO Journal*, Vol. 1, No 1, June 1983, pp. 35-41.

Russell, J. R., On Getting Started, *CALICO Journal*, Vol. 1, No. 1, June 1983, pp 51-53.

Scollon, R. K. Gutenberg, Babbage and Woz: Will Pac-Man Gobble Up the Humanities?, Lecture to the Washington Linguistics Club, December, 1982

Sheingold, K , Kane, J., Endreweit, M., Microcomputer Use in Schools Developing a Research Agenda, *Harvard Educational Review*, Vol 53, No 4, November 1983, pp 412-432.

Stevens, V., A Report on a Project Illustrating the Feasibility of VideoComputer Interface, *CALICO Journal*, Vol. 1, No. 1, June 1983, pp. 27-30 and 50.

Van Campen, J., A Computer-assisted course in Russian, in Suppes, P. (ed.), *University-Level Computer-Assisted Instruction at Stanford 1968-1980*, Stanford: Stanford University, 1981, pp. 603-646.

Van Campen J., Markosian, L., and Seropian, H., A Computer-Assisted Language Instruction System with Initial Application to Armenian, in Suppes, P., (ed.), *University-Level Computer-Assisted Instruction at Stanford 1968-1980*, Stanford: Stanford University, 1981, pp. 717-734.

Note: Reprinted from the
CALICO JOURNAL
Vol. 1, No. 1
June 1983

MONTEVIDISCO: AN ANECDOTAL HISTORY OF AN INTERACTIVE VIDEODISC

Larrie E. Gale

For years researchers and theoreticians have been arguing that real world experience is the only practical way to learn a language. But because many of us do not have the opportunity to live in another culture and be constantly, or even regularly, exposed to another language, apparently the only practical way that language instruction can take place for groups is in the traditional setting of a classroom with the extensive use of printed materials supplemented by audio tapes. The promise of new technologies changes this limited approach to language instruction. With the advent of videodiscs mated to microcomputers we suddenly have available very sophisticated ways of delivering instruction while simulating real world experiences.

The following reports an attempt to teach Spanish to students at Brigham Young University using interactive videodisc by simulating a visit to a Mexican village. In the process of describing this program called "Montevidisco," the problems encountered will be listed as well as their solutions. The design, production and computer program development will also be described along with the results of the project and the interest spawned by "Montevidisco."

Why Interactive Videodisc?

Recent research into cognitive development and language learning tends to support the point of view that language learning and meaning in language is based on schema or constructs that we develop as we interact with reality, as we experience a wide variety of communication exchanges. Such research also suggests that the interaction between the right and left hemispheres of the brain is such that the meaning of printed communication is almost wholly dependent upon the schemata that we have developed while interacting with the real world around us. If these conclusions are true, then our attempts to teach language primarily through the printed page (or through linearly processed aural communication) have basic, inherent weaknesses that frustrate rapid or comprehensive language learning.[1]

Research results which have significant implications for the teaching-learning process have been comprehensively studied and digested in *Instructional Message Design* by Fleming and Levie[2]. From this and similar sources we learn from that:

1. The way instructional content is organized is crucial and at least as important as the technology delivering it.
2. The amount of information communicated in X time is critical.
3. Tying new information or abstractions back to past experience is essential in any instruction.
4. Student control of the information flow increases acquisition and remembering.
5. The more opportunity to practice new ideas, concepts and skills while receiving prompt feedback, the better.
6. The contiguity of rules-examples-practice-feedback is most important.
7. Many others could be listed.

The significance of this is that the creative use and integration of interactive video into the teaching-learning process can help satisfy the conditions of learning identified by these research results.

Specifically, an interactive videodisc can be placed under student control and can 1) be used as a high density storage medium of still visuals, 2) display slow motion and freeze frame for the communication of critical skills, 3) provide random access capability of up to 54,000 separate still frames, or full motion/full color video segments in two different languages or stereo with access to still frames or motion both possible on the same side of the same disc. As mentioned above, two audio tracks are available instantaneously for two different languages, two different versions of the content, or for instantaneous feedback. Any frame on the disc can be located and displayed for the student within three seconds. In addition, the laser-read disc is durable and the video and audio quality are excellent.

Dr. Gale received degrees in linguistics from San Diego State University and a degree in instructional design and technology from the University of Utah. He presently serves as assistant director of the David O. McKay Institute.

When mated to a microcomputer, a videodisc can provide ample opportunity for student learning, practice, or testing while at the same time managing and monitoring student activities and progress. The capabilities of interactive videodisc are limited primarily by our imagination . . . by our creativity. This is why Brigham Young University solicited and received a grant from the Fund for the Improvement of Post Secondary Education (FIPSE) and co-produced with them the videodisc program "Montevidisco," a simulated visit to a Mexican town.

Montevidisco

"Montevidisco" is a computer assisted instruction/interactive videodisc program that takes the student on a simulated visit to a northern Mexican town and in the process exposes students to real-life situations with natives speaking to them in Spanish at native speed. The program is also like an adventure game in that the student is never sure what will happen as a consequence of his remarks to the native speaker. In fact, the student may find himself in the local hospital or in jail, depending upon the decisions he makes at critical points in the program.

From the student's eyes, the program basically functions in the following way: (1) The student sits down in front of a computer terminal that has on top of it a quality, color television monitor and, after entering his name and being introduced to the system by the computer itself, he finds himself wandering onto the plaza in the center of town, (2) As he does so a native confronts him and speaking directly to him says, in Spanish, "You're an American tourist, aren't you?", 3) At this point the video disc freeze frames, the native waiting for a response, and options appear on the computer screen which give the student an opportunity to respond in at least four different ways, including the option to have the person repeat what he just said. 4) Assuming that the student elects to hear the phrase repeated, he would press the appropriate number on the keyboard at which point the computer would expect him to record his choice in Spanish, speaking it into a microphone also provided at the computer terminal. 5) As the student speaks, only his production is recorded on a cassette tape for later review and evaluation by a proctor or the instructor. If the student does not take this critical step, the computer will keep prompting the student to create that production before it will permit him to continue, 6) Once the student has spoken the phrase into the computer, he then is given the option of hearing a surrogate produce or model that phrase for him by simply pressing "s" on the keyboard. If he elects to do so, the surrogate is immediately summoned from the disc and that phrase is stated and can be repeated as many times as the student choosees, 7) The student can then re-record his production, or continue. Remembering that he asked the native to repeat himself, continuing in this case means that the disc branches back to the beginning of that scene and repeats it. Had the student selected other options the native would have tried to promote himself as a tour guide, would have explained how to get to the beach, or in other ways would have provided different alternatives. At the end of each, brief scene the process is then again repeated where the student is expected to produce, in Spanish, his choice and then continue his adventure of exploring "Montevidisco."

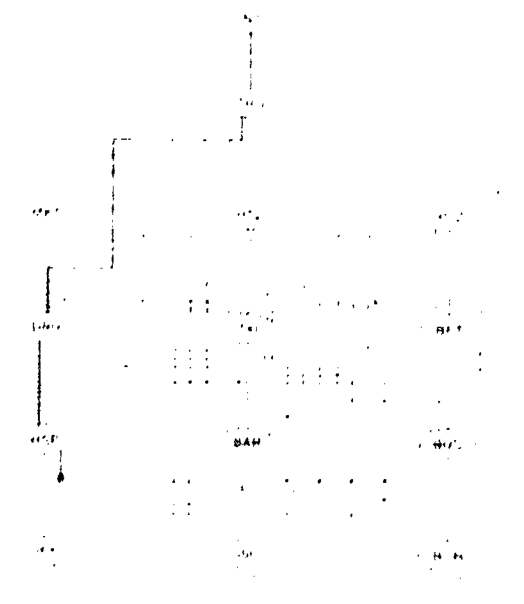
Because of the gender intensive nature of Spanish, there are two versions of Montevidisco, one for male students and another for female students. The program functions very similarly for both versions, but the options and scenes are different and, of course, the language changes to reflect the appropriate gender both in addressing the student and in the surrogate responses available.

Montevidisco consists of 28 major sequences, each of which has several scenes and each scene has at least four options. The result is over 1,100 possible op-

Programming "Montevidisco."

FIGURE 1:

A Map of the Sequences and Branches in the Male Version of "Montevidisco"



CONTENT OF THE TWO DISCS COMPARED

Male Scenes	(Code)	Female Scenes	(Code)
Introduction	INT	Introduction	INT
Tour Guide	TRG	Meals	MLS
Market	MKT	Telephone	TPH
Hotel	HTL	Hotel Room	HTR
Bull Fight	BFT	Travel	TRV
Bus	BUS	Bus	BUS
Beach	BCH	Walking	WLK
Taxi	TXI	Taxi	TXI
Hospital	HSP	Library	LIB
Drug Store	DRG	Drug Store	DRG
Police Station	PST	Macho Masher	MMR
Bar	BAR	Restaurant	RST
Disco	DSC	Shopping	MFG
Plaza	PLZ	Plaza	PLZ

tions or branches in the program, making it possible for a student to revisit "Montevidisco" several times and each time being able to visit new locations and interact with the natives in different ways.

Problems Encountered and Solved

It is not easy to script interactive video. In a normal linear production the script is written much as the final product will appear, in linear fashion. Thereafter the production team breaks that script down into a scene breakdown for production purposes, and all scenes of the same location are shot at the same time, regardless of their chronology in the script. During editing, these segments are appropriately positioned in the final linear product. However, with interactive video, there is not a linear end product until the student actually, by his own choices in using the program, creates it. Therefore, the script starts out segmented, the production aids, (such as the scene breakdown) remain segmented, as does the final footage which is prepared for pressing onto the disc. This makes it very difficult to keep track of loose ends and to retain needed continuity between scenes and throughout the sto·;

Some who ha·· ·.oduced interactive videodiscs have attempted to solve this problem by creating extensive flow charts, some of which have required literally yards of butcher paper. Our approach was to use a simple combination of a normal video script format, combined with a storyboard format which included in it the essential features of a flow chart so that branches were also represented on the script.

Another problem was assuring that all the branches made sense and that none of the options were left dangling. The pro·lem is created by the fact that if you are to present four options in the first scene, the student being able to branch to any of the four, and each of those four have four options, and each of those sixteen have four options, you can see how the production can become impossibly large very quickly. In order to prevent this, Montevidisco was designed and scripted so that several options folded back in on themselves; that is, there are several ways to get to a sequence like the police station or the drug store. This kept the production and its possibilities manageable. However, during scripting, it's not easy to keep track of all of the options and where they go. For this reason we wrote a program that would map the entire production as well as branches from each of the scenes to each of the other scenes. This program identified those branches that were left hanging and made it easier for us to identify branches that didn't make sense.

Another series of problems was encountered as we began to produce this non-linear video program. As the producer began to plan for the production, he tried to force the branching program into more of a linear mold since that was his experience. He encountered great difficulty in organizing his production aids. We feel that this is due, in large measure, to the newness of the medium and the difficulty of the producer and director to conceptualize what was required.

One of the tools that we're presently developing to help overcome the production problems is a system that we refer to as computer-assisted production. Its main purpose is to overcome the difficulty in producing production aids while at the same time keeping track of continuity and expediting the entire production process. We believe that this is possible by: 1) entering the script into a microcomputer system as it's created, 2) using the computer to help create the production aids, 3) taking that computer on location to help keep track of the production's progress, and 4) by using that information in important ways during post production. We estimate that production costs may be reduced by as much as one-third through a computer assisted production system.

With the advent of videodiscs mated to microcomputers we suddenly have available very sophisticated ways of delivering instruction while simulating real world experience.

The Production of "Montevidisco"

Hermosillo, Mexico was scouted and accepted as the shooting location during a trip a few months before production began. Because of the scouting trip, modifications were made in the script to accommodate local color. Once the town was decided upon, permission to film in Mexico was solicited through the appropriate authorities, which took some time, and required that a Mexican government monitor be on location during the shooting. Shooting permission included transporting our electronic equipment across the border. As production began, the normal frustrations and pressures in video production were encountered and were compounded by the fact that much of the talent had to be local and was untrained. Since many of the lines in "Montevidisco" are comparatively short and there is very little interaction between people in a scene (the interaction being between the student and one other individual on the screen), it was not difficult to coach untrained actors in their lines and to get them to repeat those lines with basically the same voice inflection and movement. Some of the more critical scenes and longer lines were handled by local professional actors.

Computer Program Development

As the computer program was conceptualized and developed, an evolution occurred which resulted in a program that is basically modular and which can be adapted to other functions comparatively easily. The code was written in Pascal Z (a CPM version of Pascal) which handled some functions very well but which was very limiting. (We are now developing our programs in C because of its power, flexibility, and because of the condensed nature of the code.) The computer program was developed as a template rather than a program written for the specific scenes and circumstances of "Montevidisco." This makes it possible for us to produce other videodiscs requiring a simulation approach without having to rewrite the computer program. All that is necessary is that we enter new data files.

The computer program has the videodisc player play a segment and then freeze frame, it then waits for a keyed and spoken response from the student, provides the student with the opportunity to hear the surrogate and then advances to the next video segment based on what the student had selected. It also permits the use of still frames so that the student can explore the vegetables and fruits at the marketplace, for example, and ask about each.

The conceptualization and development of the computer program was, next to the production itself, the most expensive part of the entire program. Because those costs will not have to be repeated, we anticipate realizing substantial cost savings in our next production, "Flight 505," an EFL videodisc program that we are presently co-producing with the BBC.

The Hardware Required

As "Montevidisco" was conceptualized

there were no videodisc interfaces commercially available, much less interactive videodisc systems. This required that we design and develop two interfaces: one for interfacing two videodisc players to the computer and switching between them, and another for interfacing the audiocassette recorder with the computer. The research system developed for "Montevidisco" consisted of a computer and its terminal, a color TV monitor located directly above the computer terminal and a microiphone also located near the student. It also included two videodisc players, one for the situations disc and another for the surrogate disc, a computer controllable audiocassette player, and the necessary interfaces mentioned above.

The costs for this system and its development were high and could only be justified for research. Fortunately, Sony announced an interactive videodisc system in November of 1982 which is capable of virtually all of the same functions as our research system and for a fraction of the cost. Other firms have produced interfaces which enable an Apple microcomputer to interface with industrial videodisc players. It is probable that "Montevidisco" will be re-edited to function on a single videodisc player so that it can be published for use on the Sony and Apple systems.

Preliminary Results

Two classes at BYU have used Montevidisco which makes it possible for us to suggest only very tenuous results. We had assumed that we were producing a program for more advanced students of Spanish and that they would spend considerable time interacting with the program. During this interaction we expected that they would notice, by listening to the surrogate, several of the errors that are frequently committed by such students such as the misuse of "ser" and "estar." Our experience has been, however, that the students have an inflated view of their ability to speak the language and that they seldom call upon the surrogate. This means that they simply repeat their errors. In our next research cycle we will not make the surrogate optional and hope to see the students beginning to notice their mistakes.

We also were surprised by the fact that lower level students are able to work with the system by the frequent use of the repeat option. In fact, we are beginning to suspect that this program will be of more benefit to the second year student in Spanish than we had earlier assumed.

At the present time, each student spends approximately two hours interacting with the program. We are hopeful that more students will use the system in repeated sessions when the use of "Montevidisco" is required by a professor for a grade.

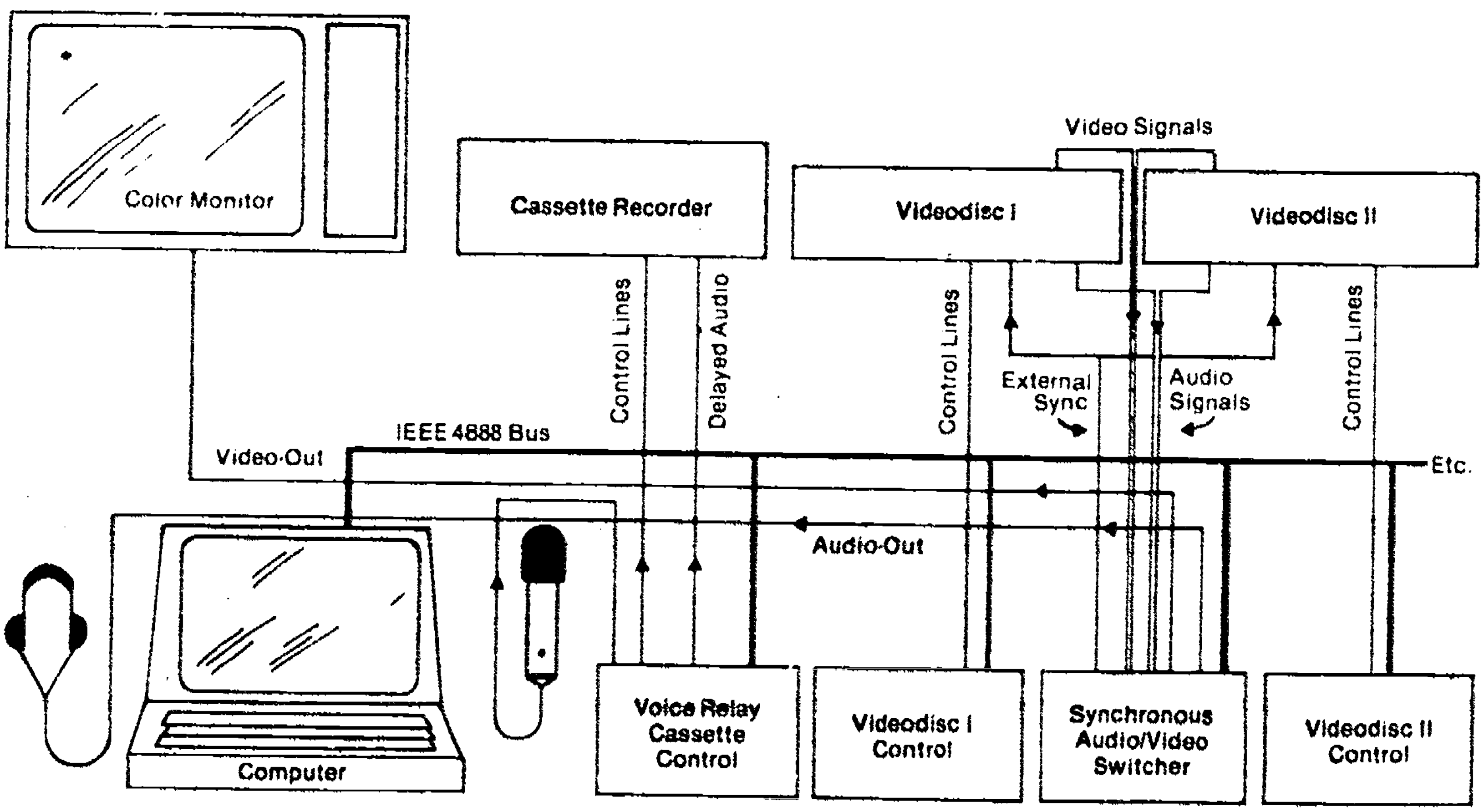
The Interest in and Future of "Montevidisco"

Our interactive videodisc work and especially "Montevidisco" has attracted international attention. Over 72 prominent visitors have been hosted at Brigham Young University, including ministers of education, members of parliament, military officers, presidents of universities and executives of corporations from the Phillippines, New Zealand, France, Puerto Rico, the United Kingdom, Canada, Mexico, Germany, the People's Republic of China, and the United States. In addition, in excess of 300 other visitors have reviewed our work here. Since January of 1983 we presented it at 15 conferences scattered throughout the United States.

Because of our videodisc work and earlier CAI work on TICCIT and PLATO, we have begun working with the BBC and agencies in the Department of

FIGURE 2:

EQUIPMENT FOR THE CONVERSATIONAL SIMULATOR



BEST COPY AVAILABLE

Defense, and have been solicited for involvement by major corporations, universities, and other organizations, such as museums, visitors' centers, and tourist attractions.

The Microcomputer Products Division of Sony has very graciously granted an interactive videodisc system to Brigham Young University which has made it possible for us to transfer much of our work to their system. We have been very pleased with its reliability and are excited about its potential, especially for language learning. The nature of the Sony microcomputer makes it comparatively easy to generate different character fonts and to generate sophisticated graphics which can be displayed

The finished product.

on the screen by themselves or superimposed on a video image. As other companies come on the market with similar systems, the competition can only benefit the field of interactive video in general.

BYU has received grants that will enable it to function as the headquarters for CALICO to publish a quarterly journal and to establish a database for related projects, personnel and literature.

Also, four major publishers have initiated contact with us, and we are hopeful that some of our work will be published and available for use in the near future.

Summary

The flexibility and powerful capabilities of interactive video hold out substantial promise for the teaching/learning process where the content is carefully designed and where the equipment systems and computer programs managing them are flexible and available at reasonable cost.

Our experience with Montevidisco, and other interactive videodisc programs, has taught us valuable lessons regarding the production, programming and delivery of interactive video instruction and, because of those lessons learned, interest in our work has been substantial.

REFERENCES

1. Arlin, P. K. "Cognitive Development in Adulthood: A Fifth Stage?" *Developmental Psychology II* (1975): 602-606.

———. "Operational Levels in Comprehension and Production of Metaphors," *Journal of Educational Psychology* (1983) in press.

Billow, R. M. "A Cognitive Developmental Study of Metaphor Comprehension," *Developmental Psychology II* (1975): 415-423.

Case, R. "Implications of Neo Piagetian Theory for Improving the Design of Instruction." Chapter in *Cognitive Development and Instruction*, Eds. Biggs and Collis. New York: Academic Press, 1980.

Eisner, Elliott. "Mind as Cultural Achievement," *Educational Leadership*, 38 (March 1981): 466-477.

Epstein, H. T. "Correlated Brain and Intelligence Development in Humans." Chapter in *Development and Evolution of Brain Size: Behavior Implications*. Ed. M. Hahn. New York: Academic Press, 1979.

———. "Some Biological Bases of Cognitive Development," *Bulletin of the Orton Society* (1980): 30.

Galin, David. "EEG Studies of Lateralization of Verbal Processes." *The Neurological Basis of Language Disorders in Children. Methods and Directions for Research* (NANCDS Monograph No. 22). U. S. Department of Health Education and Welfare, 1979.

Gazzaniga, Michael. "Brain—Corpus Collosum," *Journal of Learning Disabilities* 8 (November, 1975): 35-36.

Glassner, Benjamin. "Preliminary Report: Hemispheric Relationship in Composing," *Journal of Education* 162 (Spring 1980): 74-95.

Hunt, J. M. *Intelligence and Experience*. New York: Ronald Press, 1961.

Neimark, E. D. "Longitudinal Study of Formal Operations Thought," *Genetic Psychology Monographs*, 91 (1975): 171-225.

Piaget, J. *Success and Understanding*. Cambridge, Mass.: Harvard University Press, 1979.

Searleman, Ellen. "A Review of Right Hemisphere Linguistic Capabilities," *Psychological Bulletin*, 34 (1977): 503-528.

Shuy, Roger. "A Holistic View of Language," *Research in the Teaching of English* 15 (1981) 101-111.

Yakolev, P. A. Licours. "The Myelogenetic Cycles of Regional Development of the Brain." *Regional Development of the Brain in Early Life*. A. Minkowski, editor, Oxford, 1967.

2. Fleming, Malcolm and W. Howard Levie. *Instructional Message Design. Principles From the Behavioral Sciences*. Englewood Cliffs, New Jersey. Educational Technology Publications, 1978.

RESOURCE NOTES

Other Databases

The Northwest Regional Education Laboratory provides two major networks of computer information. June 1980 marked the inception of MicroSIFT—a network clearinghouse for microcomputer software information for teachers. In September 1982, NWREL introduced RICE (Resources in Computer Education), a computer database that that provides information on approximately 2,000 microcomputer courseware items and over 150 producers or developers of courseware.

Write Northwest Regional Education Laboratory (NWREL), 300 Southwest Sixth Avenue, Portland, OR 97204.

Catalogs, Indexes, Etc.

The Centre National Universitaire Sud de Calcul has been established at Montpellier, the second such facility projected by the Plan Informatique du Ministere de l'Education Nationale. That plan covers the joint information needs of research, instruction, management of facilities, and documentation, and the total range of computer hardware available. Details are available from Ruddy Lelouche, LISH-CNRS, 54 Blvd. Raspail. 75270 Paris Cedex 06, France.

Note: Reprinted from the
CALICO JOURNAL
Vol. 1, No. 2
September 1983

STRINGING US ALONG: PROGRAMMING FOR FOREIGN LANGUAGE CAI

Sue K. Otto and James P. Pusack

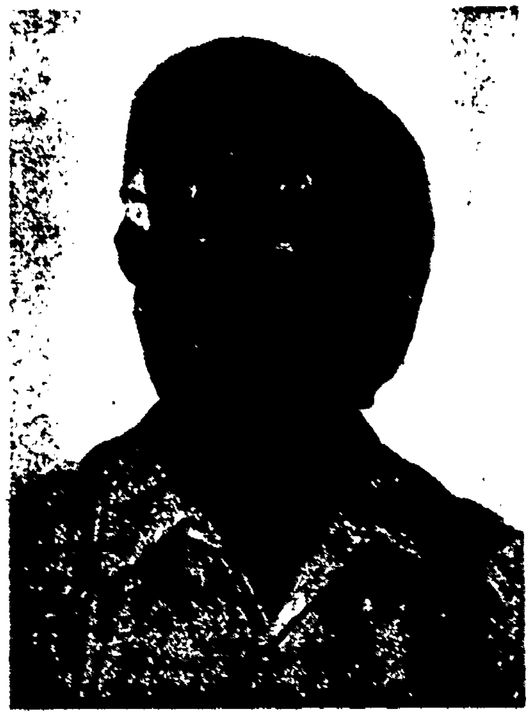
I. The Issue of Programming

The art of computer programming for computer-assisted instruction (CAI) in foreign languages is not well understood. In order to stimulate exchange of experiences and techniques in this area, we present a discussion of programming in two major sections, plus an introductory section (Section I). Section II describes an introductory programming course designed specifically for foreign language teachers. We hope it can serve as a model for similar courses at many institutions. Indirectly, it also offers the autodidact some hints on ways to focus self-study. Section III tackles more advanced questions related to the manipulation of text in foreign language CAI.

Programming consists of constructing a valid sequence of commands to the computer, so that it will perform the task the human programmer intended. To program, one must learn a programming language, like BASIC or LOGO. Asking how long it takes to do this is like asking how long it takes to learn French, or at least Esperanto. A vital question for language teachers is whether there is any point to investing money and time in computers at the level of computer programming. The answer is not an emphatic yes, but a qualified one.

The qualifications arise from the fact that there is a great distance betweeen computer literacy, that is, feeling comfortable with computers, and what can be called computer fluency. Many of the tasks teachers may be able to envision doing with the help of a computer require a high degree of expertise. Most language teachers do not have the free time, resources, and skilled assistance to acquire that expertise. A dedicated, experienced teacher-programmer, for example, could probably design and write a useful vocabulary dictation program with helpful diagnostic messages in a weekend of hard work. That amount of work would be paid off by the countless hours that students spend using the program, eliminating the need for the teacher to grade assignments or hold boring drill sessions. Reaching that level of expertise, however, could easily take months or years of programming practice. Teachers who find programming inherently interesting and challenging will find it worthwhile to move from simple projects to more complex ones, enjoying the process. Others will find any con-

Dr. Sue K. Otto received her B.S. in Foreign Languages from Iowa State University in 1969 and her M.A. and Ph.D. in Spanish from the University of Iowa in 1977. She has taught Spanish for many years at the high school and university levels. In 1976 she took a half-time position at the Language Media Center at the University of Iowa to promote and develop the Center's program of foreign language computer-assisted instruction. In 1982 she was named Language Media Center Associate Director for Project Development and Computer Assisted Instruction on a full-time basis. She is currently working on a videodisc still-image retrieval system which can be used in language learning.



Dr. James P. Pusack received his B.A. and M.A. in German from the Johns Hopkins University in 1967 and his Ph.D. in German from Indiana University in 1977. He came to the University of Iowa in 1974 and in 1975 received a faculty grant to explore the applications of computer technology to language teaching. He is now a recognized authority in this field. He is presently the foreign language series editor for CONDUIT, a non-profit clearinghouse for educational computer software, and is the chairman of the Department of German at the University of Iowa.

tact with computers exceedingly frustrating; for them, programming will be anathema.

Programming for foreign language teaching, it should be stressed, has very little to do with mathematics. Language programs which require understanding of mathematics beyond addition and subtraction are unusual. The nature of programming involves logical combinations of operations and the mastery of programming language syntax. Experience shows that language teachers find these things to be familiar concepts transposed to a new environment. In foreign language programs it is often necessary to assemble groups, or strings, of characters, to break them into smaller segments, or to scan a phrase for a specific group of characters. Much of the power of foreign language CAI derives from the ability of a computer program to store the morphological and syntactic rules of a language in the form of operations which combine or analyze groups of characters. Building an efficient verb drill, for example, involves tacking or concatenating endings onto stems. To the computer a verb stem may bear the name S$ and an ending by the name of E$. One instructs the computer to build a verb let's call it V$–by typing a statement like

```
LET V$ = S$ + E$
```

To a limited degree, the computer can exploit the very same logical patterns of language we are trying to teach. We can examine a student's answer (A$) to find the position (P) of a verb (V$) by writing:

```
LET P = POS(A$, V$)
```

at the keyboard.

(See footnote)

(Find the position (POS) within A$ (student's answer) of V$ (the verb) and store that in P)

and thus we can analyze wrong answers intelligently. The following section describes a systematic approach to the teaching of language-related programming.

II. Teaching Programming

As those of us who are self-taught can attest, getting started in foreign language computing is not easy. Indeed, the resources which are typically

A vital question for language teachers is whether there is any point to investing money and time in computer programming. The answer is not an emphatic yes, but a qualified one.

available, that is, programming courses (e.g., BASIC for business students, Pascal for computer science majors) and books and manuals designed for self-instruction, rarely lead teachers in the proper direction for dealing with foreign language CAI in a meaningful fashion. The question of how all this applies to what foreign language professionals want to do remains unanswered. It is inevitable, then, that computer courses will evolve within foreign language curricula to address the specific needs of our discipline.

The course BASIC Programming for Foreign Language CAI at the University of Iowa was developed to meet these needs. Iowa has had an active, growing program of foreign language CAI for the past seven years–an effort sustained by a nucleus of interested faculty members and by the Language Media Center. At the beginning, instructional authoring systems (IBM Coursewriter III and HP Instructional Dialogue Facility) were promoted and taught to the non-computing liberal arts community. While these languages were helpful in involving some instructors in computing, such work came to be viewed as too restrictive and as unsuited to most forms of language practice. Moreover, materials generated with an instructional language can be used by teachers elsewhere only if they have exactly the same system on their computer. The computerized verb wheel has been re-invented numerous times by language teachers using programming tools not shared by other institutions. This experience reinforces our choice of conventional programming for generating CAI materials, despite the obvious disadvantage of having to learn a programming language, which is a difficult and time-consuming task. If a vast amount of time is to be invested in production of computer lessons anyway, then it should be done with a programming language shared in some close form by many systems, so that the product is relatively transportable.

BASIC Programming for Foreign Language CAI, offered for two semester hours of credit, is not intended to be a computer literacy or computer appreciation class. It is an introductory programming course designed to teach the essential elements of the BASIC language, to lay the groundwork for sound program design, and to present programming strategies useful in foreign language CAI. The course must be taught by an instructional programmer with extensive language teaching experience. It draws primarily faculty and graduate students from the university's foreign language departments. In the summer, however, it has attracted foreign language teachers from the local school district. Important bonds of shared interests and new channels of communication between the university and the local community have resulted from this contact.

Choosing BASIC as the target language was relatively easy because of its popularity, especially in micro-computing, and because of its built-in string handling capabilities. Selecting a textbook was another matter, since the best ones are oriented toward business mathematics. The text used in the course *Programming for Poets: A Gentle Introduction Using BASIC* (by Richard Conway and James Archer, Winthrop Publishers, Inc., Cambridge, MA, 1979) represents a compromise: while it is not the best textbook for BASIC, its

Note: The computer uses and stores all information in the form of combinations of 0's and 1's. Alphabetic characters are translated into such combinations. Therefore, in order to indicate to the computer that we want it to consider a certain thing as an alphabetic expression rather than an algebraic or numerical quantity, we must label it in a special way. In BASIC the label for alphabetic expressions is the dollar sign ($). Any expression labeled with $ is an alphabetic character or string of characters (a space is also considered a character – a null character, but nonetheless a character.)

exercise material and treatment of concepts are best suited to a liberal arts audience. The emphasis is specifically non-mathematical; none of the exercises begin with instructions like Figure the interest on... or Update this personnel record.... Over half the text and exercises are dedicated to handling string data. The subset of BASIC treated by the authors is, of necessity, a relatively small one common to all dialects. The book's modest intent is to teach an understanding of rudimentary programming statements and concepts. Like the course itself, it does not pretend to transform the reader into a consummate programmer. Its flaws consist of occasionally sloppy programming (editing) of some examples and a generally non-structured approach to program design. Nevertheless, the textbook serves its purpose: to provide additional explanations and exercises at the beginning and to ease the transition into use of a specific BASIC reference manual. BASIC manuals are indispensable tools for the programmer, but are not readily usable by the novice.

Although they have decided to take a computer class, our students often arrive with many real or imagined barriers to computing firmly in place: math anxiety, machine mystification, suspicion of the proliferation of computers in our society, the view that computers and humanism are mutually exclusive. And they have few, if any, accurate perceptions about computers and programming. Therefore, during the first week of the course, the students get a concentrated dose of computer literacy, formulated to clarify the nature of the machine and its capabilities and to dispel the myth of the computer as an intelligent entity. This introduction to computing includes a film about computer hardware and software, an illustrated lesson on computer jargon (bits, bytes, chips, machine code, memory, CPU's, I/O, disk drives, etc.), and an encounter with a terminal. This is followed by a tour of the university computer center with a close-up inspection of a PRIME 750 super-mini computer and an Apple microcomputer, either of which can be used in completing class assignments. At the same time, the class reads an *Introduction to Computer-Based Education*, a general treatise on CAI published by the Digital Equipment Corporation. This briefly surveys the history of CAI and brings into focus the basic types of computer applications relevant to educators.

At this point the actual study of programming begins. A helpful parallel can be drawn between learning a programming language and learning a foreign language. One acquires a new vocabulary and some rules of syntax and then applies them to create cohesive, meaningful structures. It takes time to assimilate the basics and an even longer time to be able to use the language in a sophisticated manner. These are consoling thoughts for students to keep in mind while working through the initial acquisition phase, when use of the language produces only trivial programs. Such reflections are also appropriate at the end of the course when programming skills are still relatively unsophisticated.

[illegible] ... foreign language curricula to address the specific needs of our [illegible].

The first lesson covers the terms algorithm, program, data, and variable and introduces the READ, DATA, and PRINT statements. Although these concepts are relatively simple ones to grasp, another factor complicates the issue for students: the unfamiliarity of the medium. Frustrations arise from insecurities and confusion about how to sign on to the computer, how to invoke BASIC, how to name and save programs, how to retrieve them, how to make changes and corrections in them, how to save alterations once they have been made, and how to stop working. Guided practice sessions offer the most effective remedy for the intimidation felt by most beginners. The first of these sesssions takes place during class; the instructor talks the students through the procedures while they work at the terminals. This is followed by a second programming session guided by a written tutorial. With this preparation students then attempt the programming exercises for the first lesson. Throughout the course, students enter and de-bug their programs using a CRT; then they prepare and hand in a paper printout with a listing and run of each program.

In five sessions the class covers loops, relational operators, conditional statements, logical operators, standard output formatting, GOTO, LET, and arithmetic operators and functions. The exercises for these topics provide practice in manipulation of both string and numeric data. The textbook materials play a prominent role in discussions and assignments. By this time the students have realized that advanced math ability is not a prerequisite for programming. Of notable interest, however, is the phenomenon of algebra interference in understanding the assignment of values to variables, especially when the LET is suppressed. The statement often looks like an equation, yet it cannot be interpreted as one. This has hindered more than one student's comprehension of the value-assignment function in programming.

When the course moves beyond the lessons enumerated above, no further considerations of numeric functions are made beyond INT (INTEGER) and RND (RANDOM). Instead, the focus is strictly on handling and manipulation of string data and files in an interactive setting. Since BASIC dialects vary so widely in the areas of string and file handling, the text does not treat them in detail. Therefore, it must be virtually abandoned in favor of programming reference manuals and handouts with explanations and sample programs, all of which use foreign language examples. New topics for study include additional print formatting techniques, concatenation of strings, the LEN (LENGTH) function, substringing with MID, LEFT, and RIGHT, the INDEX (or POS) function, the string functions for case conversion, editing blanks and conversion of strings to numbers, GOSUB, and the INPUT statement contrasted with INPUT LINE. The principles stressed in this phase are the following:

1. String data can be combined or dissected in many ways which can be used to good advantage, particularly in dealing with the formal regularities of a language.
2. Screen format should be uncluttered, easy to read, and free from computer-generated prompts (such as the ! and ? input prompts) which do not have a logical meaning within the text.
3. Serious evaluation of input assumes careful checking and editing prior to any matching step.
4. All input, even numeric, is best handled via the INPUT LINE statement to avoid loss of data because of punctuation and to allow a check for inappropriate input, which might generate error messages or bomb the program.

Program reading is an important and profitable activity now that the class is above the base level. Through examples and more complex programming exercises with strings, students begin to glimpse the computer's real potential for language manipulation.

It is at this point that the concept of top-down or structured program design is introduced. Each student must design and program a final project following this method. Creating a structured program design is the most difficult exercise of all because it requires the most critical of programming skills: the ability to break a complex task down into a set of discrete components that, when expressed in programming code, work together in an efficient way to accomplish the task. Initial design of the project is undertaken while students continue to learn about programming.

The class now delves into file handling: creating, opening, closing, reading from, writing to, and detecting the end of files. For students and faculty auditors, one of the major fringe benefits of the course is the introduction to text editing (word processing) which forms part of the unit on files. Files, of course, hold the key to expanding lessons once an algorithm has been developed to handle some specific task using text stored in a fixed format. Strangely, many novice programmers are extremely reluctant to make the switch from storing data in program statements to storing it in files; text in a file seems somehow not visible or real enough.

German verb program on an Apple

The final BASIC language concepts presented to the class are arrays, the random arrangement and selection of array elements, and a few additional common programming constructs such as ON X GOTO GOSUB and DO loops. The last several sessions are spent in examining instructional languages, analyzing different types of answer processing, discussing specific display problems of foreign languages, and comparing the virtues of microcomputers and mainframes.

Two major programming exams are given during the term; on these exams students correct syntax and logic error in programs, describe the anticipated output of programs, answer brief essay questions, and write short program segments. The final project takes the place of a final exam. For this project the student selects some computer-based task, then designs and writes a program to handle it. Most projects are drill and practice programs characterized by limited input (usually one word) and limited feedback, in the form of simple right/wrong evaluation or of somewhat more complex error anticipation based on a partial matching routine. Representative projects have included German adjective endings, French verb transformations, French indirect and direct object pronouns, and Spanish number exercises. Typically, the programs present items in a random fashion, keep track of items missed for later re-presentation, filter input to improve matching efficiency, and insulate users against cryptic computer error messages.

The students who take this course cannot immediately claim to be expert programmers, though some do go on to achieve greater proficiency. The class does provide, however, a solid introduction to many important fundamentals of BASIC programming and language-oriented CAI. Many students discover that they do *not* want to become programmers. Nevertheless, they have gained new skills and perspectives which enable them to assess software intelligently, to alter other people's software to suit local needs, and to communicate effectively with a professional programmer in collaborative efforts to produce CAI materials. These results may, in the long run, prove as important to the cause of foreign language CAI as turning out accomplished programmers.

III. Programming Techniques

Programming for applications in foreign languages often goes beyond the techniques and examples found in introductory manuals and courses. The core of many advanced programming considerations in the field is the handling of text. In this section we introduce a number of issues in the area of text handling and present some sample programming. This section, consequently, is not addressed to the curious beginning programmer, but primarily to the colleague who may be faced with difficult programming tasks. We hope to initiate a dialogue at a level which is technically interesting, without becoming arcane. The examples are rendered in Applesoft BASIC, for the Apple II/IIe microcomputer, but the problems and solutions offered will, we hope, illuminate serious instructional programming on any microcomputer.

One common misconception among teachers exploring computing for the first time is the idea that vast storehouses of power are lost when we choose, or are forced to choose, microcomputers over large computers (mainframes or minicomputers). Not only is the capacity of many microcomputers—in terms of program size—greater than on some large machines; it may also be possible to obtain faster execution time on a microcomputer. While time-sharing computers are likely to be inherently faster than personal computers, the fact that they have to share their capacity with thirty-two or sixty-four or more users often cancels out this advantage during popular daytime hours, when language students may prefer to drop in for a fast drill session. More important than this comparison of speed and capacity is the fact that microcomputers can provide text-handling capabilities not typically available to users of time-shared BASIC, even on very powerful machines. This is because, first, microcomputer BASIC languages allow more intricate programming techniques, sometimes with direct access to memory locations and machine code routines; and, second, because time-sharing systems usually employ a mishmash of terminals across campuses, most of which offer no method of handling foreign-language characters. Techniques described below provide a means of exploiting these potential advantages.

Text at the lowest level consists of strings of individual characters. Most microcomputers which support graphics also permit the user to design and use, without additional hardware, specialized character sets. On the Apple, for example, there are at least two commercially available character generating packages. These programs employ a kind of etch-a-sketch system which allows the user to turn the dots of each character on and off. Turn on two dots over the letter 'a' and you have an umlaut. The so-called high-resolution character generator also provides the lower-case characters missing on the Apple II. In a matter of hours, unused characters like the square or curly brackets can be replaced with all the diacritical marks needed for Western European languages. No hardware is required. When the revised character set is loaded and invoked, all text is drawn on the screen, but most other features of the program remain the same.

> ***Files hold the key to expanding lessons once an algorithm has been developed to handle some specific task.***

Used in tutorials and simulations where students do not type responses in the target language, this display capability quickly upgrades the visual qualities of any program. Colleagues who balk at upper-case-only text with crudely simulated accent marks will find the computer display much more civilized; one more barrier to acceptance will have been removed.

Unfortunately, substituting character sets is only the first step in converting a program to an interactive foreign language text-handling system. While foreign language characters can now be displayed, they are still not fully available to the student. Often, the tilde is tucked away over the 9, the grave accent replaces the dollar sign, or the circumflex cannot be typed from the keyboard. Even if all the characters can somehow be typed, there is a pedagogical objection: they are not handled in any mnemonic way. What is needed is a memorable input system. For German, for example, all four of the alien characters can be handled the same way, by typing them as control characters. This requires a modification of the input routine, so that characters typed by the student can be filtered and converted to the desired foreign-language character, stored, and then displayed. Few programs currently available even attempt this.

Input filtering is only one aspect of the larger question of input control. In instructional computing, the standard forms of input are simply inadequate for reasons almost too numerous to mention. BASIC input routines choke on commas, may not support lower case, never handle accents mnemonically, never provide immediate reaction to special help commands, handle editing badly, never lock out irrelevant characters from the keyboard, and never perform intelligent management of word boundaries at the right margin. Programs which simply INPUT I$ will, therefore, never meet the standards of high-quality programming needed for text-handling in language teaching. These standards meet or exceed those expected for the best text-editing programs and share many objectives with word-processing.

An instructionally sound input routine for textual responses should perform the following kinds of tasks:

1. Get characters one at a time from the keyboard.
2. Allow mnemonic typing of special characters, such as accents.
3. Filter out or ignore digits and other keystrokes which have no value in any correct answer.
4. Allow backspacing to edit input, but not allow extra backspacing past the location where input commenced.
5. Check whether a maximum permissible input length has been reached.
6. Check whether RETURN has been pressed, indicating termination of input.
7. Check whether a special character indicating a special command

(e.g. help, menu, gloss) has been typed.

8. Handle wraparound at the right side of the screen, so that words are not broken.
9. Handle upshifting
10. Optionally check for other input trouble spots, such as leading and multiple spaces.

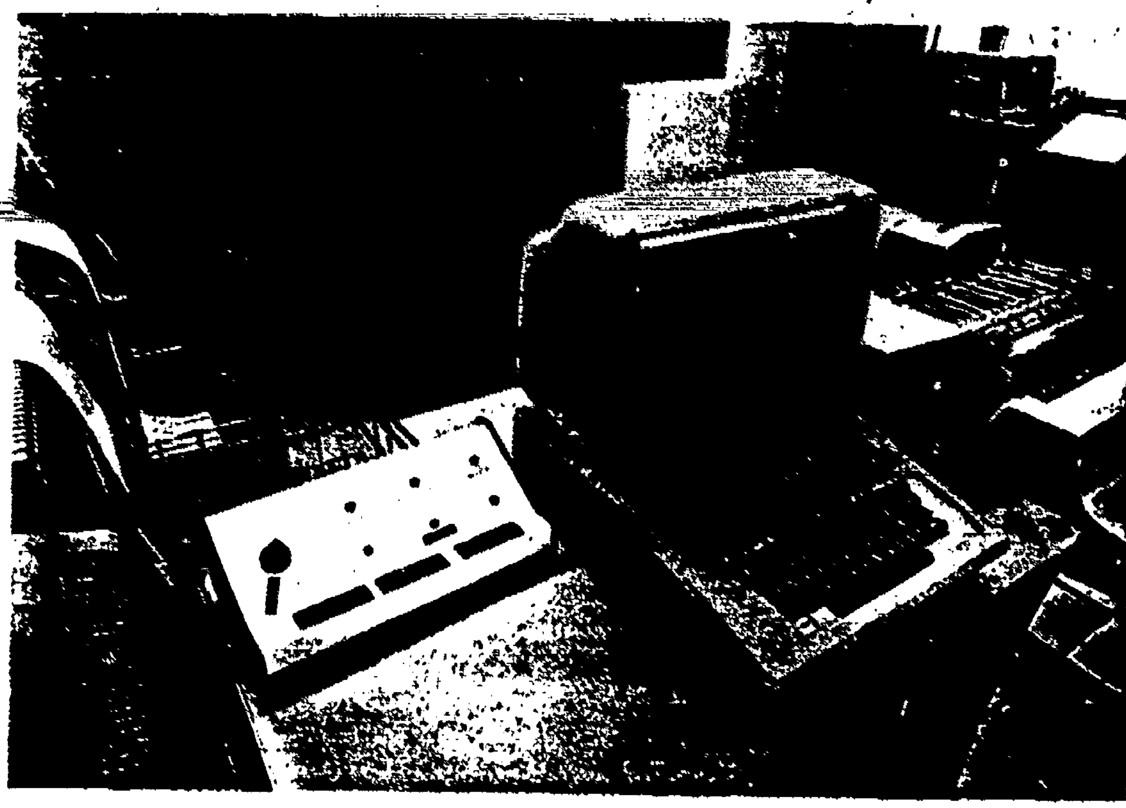
A text input routine which performs all these tasks is an elaborate affair. It uses the GET I$ statement, which polls the students for a single character but does not display it. Once the character has been obtained, it is examined, transliterated, sometimes deleted, or used as the basis for a whole series of operations. All of this must take place between the individual keystrokes of the student's answer. Doing this efficiently in BASIC is no trivial task. The ideal is to undertake as many input-controlling functions as possible without making input too sluggish.

Controlling input, without resorting to machine code, requires the use of a programming construct not often explained in the manuals. For want of a better name, the construct can be called the logical for-next loop, which is a devious exploitation of the familiar for-next loop. For fast execution of BASIC code, it is necessary to compress the code into as few lines as possible and to avoid, at all costs, the use of any time-consuming GOTO's inside the input loop. Compressing code usually entails avoiding IF...THEN statements, as well, since tests for a logical condition with IF frequently lead to extra lines of code.

A standard for-next loop looks like this.

```
5 S = 0
10 FOR J = 1 TO LEN(A$)
20 IF MID$(A$,J,1) = " "
   THEN S = J: GOTO 40
30 NEXT J
```

In line 20 the program uses the MID$ statement to examine each character of A$; the variable J will change from 1 to 2 to 3, etc., indicating the position of the character. This piece of a program would find the location of the first space in the string A$, a typical operation necessary to isolate individual words in an answer. It performs its task perfectly, but slowly, because it uses three lines, rather than one. As part of an input routine, lines like these create enormous difficulties and involve a heavy overhead in cumbersome GOTO's.



Apple program development station

The conceptual structure of a logical for-next loop can be viewed as follows:

```
10 FALSE = 0
  : TRUE = NOT FALSE
  : FOR TEST = FALSE TO
    TRUE STEP TRUE
  : TEST = [perform logical test]
  : NEXT TEST
```

We call this a logical for-next loop because termination of the loop is based on a logical test, rather than counting the number of iterations. Logical operations in BASIC yield a decimal value of 0 when false. If you type PRINT 1 = 0 or PRINT CAT = DOG on your microcomputer, it will respond by typing a 0, meaning the logical operation yielded a result of false. If a condition is not false, either 1 or -1 will be returned from a test, depending on the machine. The Apple returns 1 if you type *PRINT 3 = 3 or PRINT NOT 0* or PRINT DOG = DOG. Doubtless to the dismay of mathematicians, these logical values can be used in arithmetic calculations, including the numeric portions of string operations. A concrete example of the above construct may help to make this clearer. The following routine keeps polling the student for an answer until the student types either YES' or NO:

```
10 FOR J = 0 TO 1
  : INPUT "ARE YOU DONE?";
    X$
  : J = (X$ = "YES" OR X$ =
    "NO")
  : NEXT
```

Understanding this single line of code is essential for the subsequent discussion. The looping variable J begins as 0, or false and strives to reach 1, or true. It is set to either 0 or 1 *inside the loop* during every iteration. The logical test of X$ determines the value of J. If the condition in parentheses is true, J becomes 1 and the loop terminates. Otherwise, J is reset to 0 and the input line is executed again. As an aside, we note that the parentheses are not needed, although they help humans read the logical condition correctly. If this is still not crystal clear, typing in and running the previous two pieces of code may prove helpful.

Now consider the first example we gave, a for-next loop intended to find the first space in a string. Using a logical for-next loop, we can speed up this operation radically and reduce it to a single line of code (see Figure 1):

The kindest reaction to this line of code is to say that it is not straightforward. It illustrates a logical test for a given condition—whether a character is a space. The looping variable J begins at 0, representing a false condition, and strives to reach 1, the true condition. The value of S is set by a rather unfamiliar-looking, but legal statement which multiplies the current position (P) in the string times the result of a logical operation. As a result, when P indexes a non-space, S will be P times 0, or 0. When P indexes a space, S is set to P times 1, or P. The last statement in the loop, J = (S OR P > L), resets the looping variable J by testing whether S is non-zero or the current position has exceeded the length of A$. If a space is found, J will be 1; if the whole string has been tested, J will be 1; otherwise, J will be 0. Whenever J is reset, i.e., false, the loop continues, examining the next position of the string.

Devious as it may seem, the logical for-next loop offers programmers a powerful mechanism for compacting code and reducing execution time. It avoids GOTO's by carrying out a few additional arithmetic and logical operations which microcomputers handle in an extremely efficient way. The entire for-next loop, including the condition test, can be contained on a single line, eliminating the need for skipping over lines. Since it makes code hard to read, this structure should only be used when necessary. Unfortunately, the manipulation of text often requires its use.

Returning now to the question of text input, we can illustrate a pared-down version of an input subroutine which uses a logical for-next structure to process individual characters as they are typed. It employs a numeric array, TT% (the % sign is the sign used in BASIC to mark numeric arrays), to convert the ASCII values of the input to the desired values in the character set and to upshift whenever the escape key has just been pressed. For each character's ASCII value, the table TT% supplies a new, translated value. Values greater than 0 are normal characters; values less than 0 are special cases, such as the RETURN key; characters to be ignored are given a value of 0 in the table and discarded. Variables, rather than constants, are used throughout to speed up execution. The comments at the right explain some further details (see Figure 2):

Notice that there are multiple NEXT statements for the single FOR IL statement and that there are several ways to terminate input, with several associated RETURN statements. Additional statements can be added to handle the automatic wrapping of lines at the right margin so that words are not broken and to check for other special input conditions.

This construct can be called the logical for-next loop, which is a devious exploitation of the familiar for-next loop.

By combining a smart input routine with a redesigned character set, foreign language programmers can obtain high-quality interaction at the level of display and student polling. What happens *after* the RETURN key is pressed is a separate phase of instructional computing which separates primitive flashcard programs from serious tutorials, problem-solving, drills, or simulations. Any of these forms of CAI which involve text input will require much more elaborate text-processing routines. Serious programmers must develop a library of reliable general-purpose routines which perform the most common text-related operations at maximum efficiency.

Here we suggest the most common routines, many of which should make frequent use of logical for-next loops to speed execution:

1. A *trim* routine which removes leading, trailing, and multiple blanks from a student's answer.
2. An *edit* routine which converts all occurrences of one pattern to another pattern in a given string; it should also delete patterns, i.e., convert a pattern to the null string.
3. A *transliteration* routine which converts single characters to associated single characters; the most common usage is for upshifting input.
4. An *index* routine which finds the first occurrence of a pattern in a given string, looking from left to right; a more powerful version could also look from right to left on demand.
5. A *count* routine which counts the occurrences of a pattern or character in a given string.
6. A *fields* routine which picks out the nth field in a string divided up by delimiters, e.g., Good./Great./Fine./
7. A *match* routine which permits segments of an answer to be ignored during a comparison; for example, "have&eaten" could be used to specify a case where any string beginning with "have" and ending with "eaten" would make a match, no matter which characters come in between.
8. A *best match* routine which finds the longest substring common to two strings.
9. A *file reading* routine which ignores commas and other delimiters in lines of text input.

Each of these routines must be tested thoroughly to verify that they handle borderline conditions, such as null parameters, correctly or that they return an error message for invalid input. In practice, even the best BASIC versions of these routines may perform too slowly when very long strings are evaluated. The solution is to obtain machine-code versions. The index routine, which is the most critical, is available on many machines already; since Applesoft BASIC does not have it, a number of clubs and magazines have published machine-code versions.

Once these routines are available, a programmer in the field of language teaching can begin to link them together to form fairly powerful algorithms to analyze student input. Extraneous characters and unwanted punctuation can be removed (*editing*), variation due to capitalization can be offset (*transliteration*), individual words can quickly be isolated (*trim* and *index*) and evaluated (*match*). If student answers do not

Figure 1

```
10 P = 1
   : L = LEN(A$)
   : FOR J = 0 TO 1
   : S = P * (MID$(A$,P,1) =
     " ")
   : P = P   1
   : J = (S OR P > L)
   : NEXT
```

```
set start position to 1
set up length of string
start logical loop
test for space
move pointer forward
test to terminate
```

Figure 2

```
50 LN = C0
   : I$ = NL$
   : U$ = C0
   : FOR IL = C0 TO C1 STEP C0
   : GET J$
   : IF NOT LEN (J$) THEN
     NEXT
52 AS = ASC(J$)
   : AS = TT%(AS,US)
   : US = C0
   : IF AS > C0 THEN
     J$ = CHR$(J$)
   : PRINT J$
   : I$ = I$ + J$
   : LN = LN + C1
   : IL = LN = ML
   : NEXT
   : PRINT BE$
   : RETURN
54 AS = ABS(AS)
   : IF AS = BS THEN IF LN
     THEN
     LN = LN - C1
   : I$ = MID$(I$,C1,LN)
   : PRINT BS$KE$;
60 IF AS = ES THEN US = C1
   : NEXT
63 IF AS = SL THEN GOSUB 30
   : IL = C1
   : RETURN
65 IL = AS = RT
   : NEXT
   : PRINT
   : RETURN
```

```
set length of input line to 0
set input string to null string
turn off upshift switch
begin endless input loop
get a single character
discard null character
get ASCII code of character
filter and shift using table
reset the upshift switch
if new value is positive
convert back to a character
display it on the screen
add it to input string
update length of input
check length of line
if ok, get next character
otherwise, ring bell and
return
convert value to positive
check for backspace
if possible, shorten string
by discarding last character
print backspace and clear line
if escape was typed, turn on
upshift and get next character
if slash was typed, do special
command, like help, and
return
check if return was typed
get the next character or
terminate input with carriage return
and return
```

match precisely, further analysis will be necessary to guess the student's intentions (*best match*) and to produce helpful guidance. The challenging area of answer processing is too diverse to cover here, but any approach will certainly require the library of routines we have proposed.

A final area of text-handling which presents problems in BASIC is the use of the concatenation operator (+) to assemble text based on syntactic or other criteria. In contrast to many of the techniques discussed above, which deal with evaluating student input, concatenation is most critical when programs attempt to create rule-generated samples of language. A pseudo-transformational grammar can be imbedded in the logic of a program so that practice material need not be prefabricated. Instead, phrases and sentences are assembled from a compact lexicon and a set of rules. This always involves the joining together of strings into larger strings based on complex systems of decisions. Whenever a condition is tested to decide whether a string should be included or omitted, additional lines or GOTO's may result. Here is a very simple example:

```
100 M$ = "You have"
110 IF S < 90 THEN M$ = M$ +
      "not"
120 M$ = M$ + "met the mastery
      criterion of 90%."
```

It is possible to avoid this dilemma by abusing the fact that the MID$ string operator permits a length of 0 for its argument. This results in a single line of code:

```
100 M$ = "You have" +
      MID$("not",1,4*(S<90)) +
      "met the mastery criterion of
      90%."
```

If the score (S) is less than 90, the length of the second string will be 4 times 1 (since S<90 will be true). If S is 90 or greater, the length will be 4 times 0, since the logical condition will be false; the "not" will not be used.

Condition tests like these can also compare string values. This technique allows the concatenation of substrings like

```
MID$("German",1,6*(I$="G"))
  + MID$("French",1,6*(I$="F"))
```

which picks the first element if the student inputs G and the second if the student inputs F during, for example, a menu step for selecting which foreign language is desired.

IV. CONCLUSION

As our discussion makes clear, the range of programming skills and tasks facing language teachers is very broad. Here we have not touched on graphics and list-handling structures which are also often needed by language teaching programs. It is apparent that not every language teacher will become a dedica-

Continued on page 47, column 1

Otto & Pusack

(continued from page 33)

ted programmer. The purpose of programming courses is to help teachers decide just how deeply involved they will become. All language teachers will use off-the-shelf software, authoring systems, and home-brewed programs to create the spectrum of materials best suited to their courses and curricula. The most important objective to keep in mind is our obligation to encourage the production of the highest quality materials using the computer to the fullest, but in appropriate ways. In order to accomplish this, language teachers must become familiar with computers at least to the degree of becoming computer literate and familiar with available language software.

RESOURCE NOTE

Cambridge University Press has launched a new series of books under the general title of *New Directions in Language Teaching*. The series is co-edited by Howard B. Altman and Peter Strevens.

The series hopes to serve the interests of language teachers and others who wish to be aware of major issues facing the profession today, who seek to understand the theoretical underpinnings of current debates, and who wish to relate theory to classroom practice. These books are designed to provide stimulating discussions of important new developments in language teaching theory and methodology.

Those who would like to contribute a volume to the series or to make suggestions for volumes should write to either of the series editors: Prof. Howard B. Altman, Department of Classical and Modern Languages, University of Louisville, Louisville, KY 40292, USA; Prof. Peter Strevens, the Bell Educational Trust, 1 Red Cross Lane, Cambridge CB2 2RU, ENGLAND.

Book Review by Netta Gillespie.

Handbook of Interactive Video, edited by Steve and Beth Floyd. White Plains, NY.: Knowledge Industry Publications, Inc., 1982. $34.95.

There is a genuine need for books that provide, on the one hand, intellectually rigorous explorations of the philosophical and psychological aspects of education via technology on the one hand, and for no-nonsense how-to-do-it works on the other. Too many of the non-scholarly books and periodicals one sees in the field of educational technology seem to fall into some middle ground, tending to repeat platitudes and to provide nothing really new in the way of technical information.

Unfortunately, *Handbook of Interactive Video* is one of those works. What it provides is a collection of essays by industry-based instructional technology professionals who tend to emphasize training rather than education, and who seem to envision resources in terms of time, money, and personnel that are beyond the wildest dreams of most of us in the field of education. For example, one essayist remarks that a team of 10 to 25 people involved in the production of an interactive program is not unusual. Another estimates a budget for such a lesson to be $98,500. There also seems to be a great deal of confusion on the rather basic issue of whether one is talking about discs or tape; the essays wander back and forth (sometimes the wandering occurs intra-essay) between the two.

For an absolute beginner in this area--if you can imagine someone about to embark on the development of interactive video-dcomputer materials who is innocent of all knowledge of interactive video/computer materials who is innocent of all knowledge an interactive program might consist of, or what instructional problems it might best address,--the essays may prove helpful. Frankly, I have trouble imagining such a reader.

Much more interesting than the body of the work are the brief descriptions of some exemplary interactive projects, including the CPR training project of the American Heart Association, the Jeppeson Sanderson Flight Training Program, and the MIT Surrogate Travel Program; these accounts are especially valuable in having been written by individuals active in the development of the projects. Lists of suppliers of equipment, including interface devices, may also prove helpful, though in such a rapidly developing field, the lists are doubtlessly out of date.

Basically though, the practical aspects of interactive production and the serious discussion of the relative merits of various approaches and technologies as they relate to the educational task at hand that one would like to see in such a work are lacking. Most of the material in this book is too elementary and general to be of much interest to professionals in the field of education.

Netta Gillespie is Video Coordinator for the University of Illinois (Urbana-Champaign) Language Learning Laboratory and a candidate for the Ph.D. in Educational Psychology with an emphasis on the applications of technology to learning and instruction.

Index
Major Microcomputer Journals

If you are always wondering where you can read or find an article on topic A with regard to microcomputers, you might check out *COMPendium*. It indexes just about all the major microcomputer journals and cross indexes by subject. It offers synopses of over 800 articles each issue. It is published by EPICUROUS Publishing, P.O. Box 129, Lincolndale, NY 10540. It costs $22.00 per year and is also available on disk for $250.00 per year. The disk service is called INdisk and offers a BASIC program on the disk which allows the user to access any of the information in *COMPendium* by key-word.

Note: Reprinted from the
CALICO JOURNAL
Vol. 1, No. 4
March 1984

USING MICROCOMPUTER WORD PROCESSORS FOR FOREIGN LANGUAGES

Kim L. Smith

ABSTRACT

This paper presents a description of the programs and modifications needed to do word processing using foreign language characters.

The contents of this paper were presented at the 1983 International Conference on Computing in the Humanities held in June, 1983 at North Carolina State University at Raleigh, North Carolina. It was not published at that time.

During the past five years, many programs have been written which allow the use of a microcomputer as a word processor. Most of these programs are adequate for any English word processing applications. But they are frustrating for the linguist who desires to prepare manuscripts in any non-English alphabet language. Any language which uses accent marks, special characters or different alphabets have to be entered using English equivalents. It can be very irritating to have to read Russian which has been typed with English letters. True, the text could be printed out on a printer which has a Cyrillic type element, but the manipulation of the text would be much easier if it could be typed on the screen using Cyrillic characters.

There are several word processing programs which do allow word processing using foreign language characters.

The Word Processor Program

The word processing program which I selected for use is ScreenWriter][, written by David Kidwell, and marketed by Sierra On-Line Systems of Coarsegold, California. This program was originally released in 1981 as SuperScript and was renamed SuperScribe][in May of that year and was finally given its current name in May of 1982. These name changes were made as a result of conflicting product names. I will refer to the program as ScreenWriter throughout the remainder of this paper.

ScreenWriter first appeared on the market in February of 1981. A professor of Linguistics at BYU pointed out an ad for this program. The company claimed that the program could use soft character sets (character sets which could be designed by the program user).

I purchased a copy of the program and proceeded to experiment with it. There were some problems in using the soft character sets which had to be resolved. But with some telephone consultation with Sierra On-Line, I was able to resolve the problems. I will give further information on how the program must be modified later in the paper.

The word processing features of this program are very sophisticated considering the relatively low price. I have found that this program has many features which are not found in most word processors. It has all of the normal editing features found in most word processors, but also has the ability to create four separate indices for any document. One could be used for a table of contents, one for a glossary, one for an alphabetical index and perhaps one for a topical index. The user determines how they will be used. ScreenWriter also has the ability to process data files to create the ubiquitous form letters which abound in these days of computers. The form letter or mail merge option usually costs additional for other word processing pro-

Kim L. Smith, is a Research Associate at the Humanities Research Center. Mr. Smith served in the United States Army for fourteen years as a translator, intelligence and traffic analyst and instructor of Chinese and traffic analysis. He is one of the few military linguists to be certified as a Professional Linguist in the Chinese Language by the National Security Agency.

His professional experience includes 11 years as a translator and translator supervisor, 4 years as an instructor and instructor supervisor in Chinese. He has lived for eight years in Taiwan, Hong Kong, Thailand and Okinawa. He is fluent in Chinese Mandarin and Chinese Cantonese. He is a graduate of Brigham Young University in Chinese and Asian Studies. He currently serves as chair of the Character Fonts and Graphics Special Interest Group in CALICO.

grams. So in addition to the foreign language capability there are many desirable features which make this an excellent all-purpose word processor.

Use of ScreenWriter at Brigham Young University

In my experiences there has usually been some resistance on the part of Humanities faculty members to accepting the use of computers. However, when I showed this program to a professor of Russian, he was immediately converted. He now does most of his word processing in English, Russian, and English-Russian mixed text on the APPLE computer. He also does initial input of Russian text for files which are transferred to the Humanities Research Center's IBM 370138 main frame computer for processing.

This program is widely used in the College of Humanities because of its foreign language capabilities. Some faculty have entered the world of computing through use of the microcomputer as a word processor. They later wonder what else the computer can do for them and they increase their knowledge and abilities to do more research, aided by the computer.

A class in Linguistics which introduces students to the use of computers in languages has used this program on the Apple for two years for foreign language processing. Members of the faculty are now using this program to do word processing in Russian, Spanish, German, French, Navajo, and English.

Development of the Character Sets

The character sets used in this program may be developed with several different programs. The one which I use is the ANIMATRIX program found on the DOS 3.3 TOOL KIT disk sold by APPLE Computer. The ANIMATRIX program is compatible with the program which is used to print foreign characters on a dot-matrix printer.

This program allows you to create any character set which will fit into a seven by eight matrix. The character set must be designed to match the type element which you will use if the printing will be done on a printer which uses a ball or daisy wheel element. The characters are built on a screen using cursor controllers (sounds much better than game paddles). Figure 1 shows the screen used to build the character sets, while Figure 2 shows the listing of a character set.

Program Modifications Necessary To Use The Foreign Character Sets

It is necessary to make some changes to the program called APP2 which is on the ScreenWriter Master disk. The program should be modified according to the directions found in Appendix A. These modifications will work with version 2.0 of ScreenWriter. A new version which will work with the new APPLE IIe computer will be released soon. I am currently working on an advance copy of the program in order to set up the method of using foreign character sets. It has been modified to allow input of the character sets but there are a few changes which must be made before it is released.

This word processing power combined with its foreign language capability would allow any person to work with almost any alphabetic language.

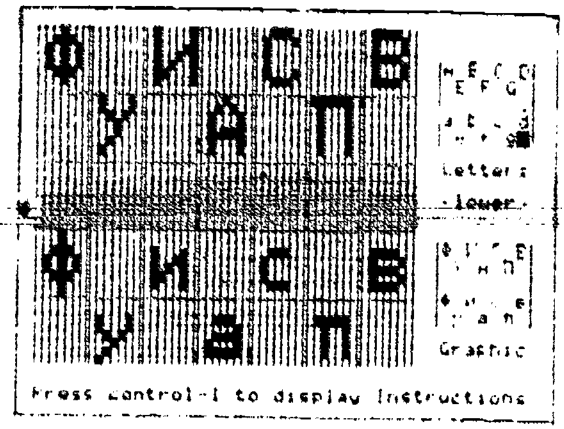


Figure 1 -

Use of The Character Sets in The Wordprocessing Program

When the editor is selected on the main menu of the program, it will pause and ask if an alternate character set is desired. If so, you will be directed to remove the master disk and to insert the disk which has your character sets on it. I keep a separate disk which holds all of the special character sets which I develop for this program. After the alternate character set is loaded, the master disk is re-inserted into the disk drive and the program completes loading. When editing begins, all characters on the screen will appear in the alternate character set. This may be a

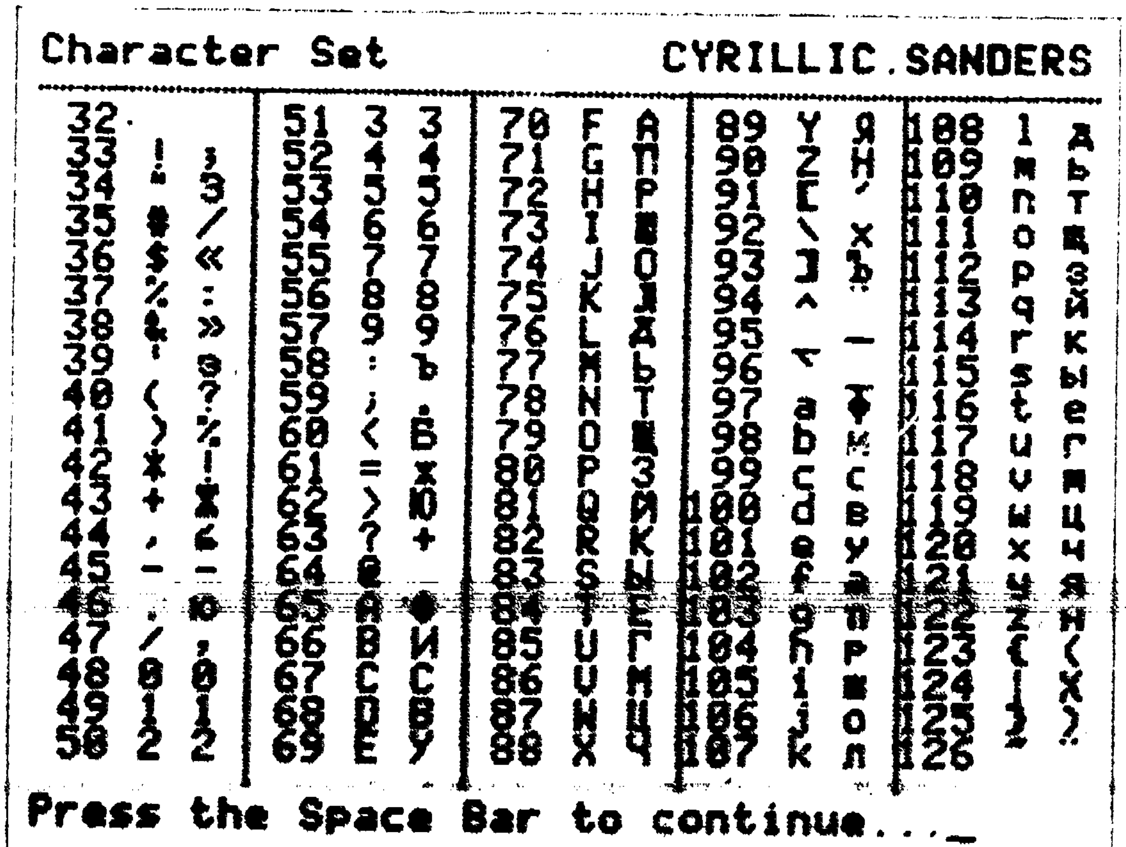


Figure 2 -

bit disconcerting to see the command line appear in Cyrillic or Greek letters! (See figure 3) At the present time there is no way to correct this problem, but it is a minor one. After the character set is loaded, typing may be done as normal.

I recommend that you restart the program when ready to print. If you go directly into RUNOFF (the printing portion of the program) from the EDITOR, the menus will appear in the foreign language alphabet. This again would pose a problem if a non-Roman alphabet language has been used.

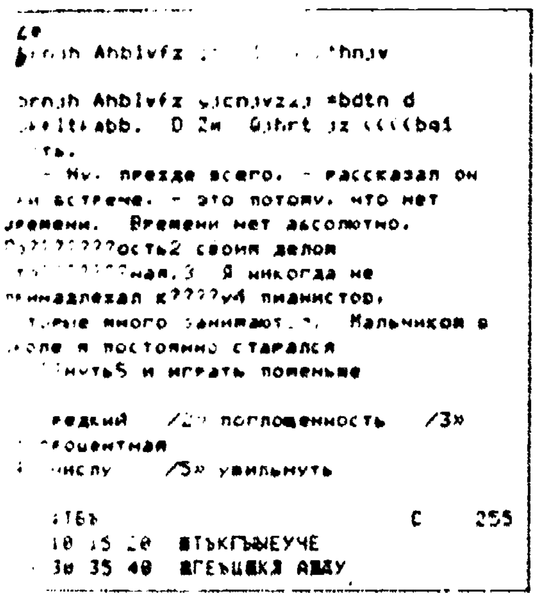


Figure 3 -

Printing of Foreign Character Sets on Various Types of Printers

If you have a printer and interface card which are supported by the program *Complete Graphics Writer* by Computer Station or other software which allows you to print high resolution graphics characters, you may print your files with a dot matrix printer, otherwise a daisy wheel or ball element printer will need to be used.

Conclusion

I highly recommend this program for anyone who is working with a limited budget. But even though it is inexpensive, ScreenWriter is a program which can perform most of the word processing that anyone could desire. This word processing power combined with its foreign language capability would allow any person to work with almost any alphabetic language.

APPENDIX A

Alternate Character Sets

ScreenWriter][supports the use of alternate character sets. The character table resides at memory location hex $D00. The alternate character set can be created using the Keyboard Filter, Typesetter or Animatrix formats. The table must, however, be inverted for use with ScreenWriter. The program UPSIDE DOWN on the master disk will perform this conversion for you. You may design the character set in any keyboard layout that you desire. You may only alter the large character (40 column) set at this time. This gives you the ability to type in one character set in 40 column mode and then to switch over to the normal ASCII set in 70 column mode. This, for example, could allow the use of Russian in 40 column mode and English in the 70 column mode for bilingual applications.

You must make sure that you have a printer which will allow printing of your soft character sets! Any daisy wheel or ball element printer can be used or any dot-matrix printer which can be driven with the Combined Graphics Writer program produced by Computer Station of St. Louis, Missouri. This program will not only work with Animatrix character sets, but will allow printing of any soft character set.

To load in an alternate character set with ScreenWriter, version 2.0, you must add the following lines to APP2:

```
7810 GOSUB 10000
8810 GOSUB 10000
10000 CALL -936
10010 PRINT "ALTERNATE
      CHARACTER SET? (Y/N)"
      ;:GET Q$
10020 IF Q$ () Y THEN PRINT D$:
      RETURN
10030 PRINT "PLACE LIBRARY
      DISK IN DRIVE 1"
10040 INPUT "ENTER SET NAME:
      ";SN$
10050 IF SN$ = "C" THEN PRINT
      D$"CATALOG": PRINT
      "(SPACE) TO CONTIN-
      UE";:GET Q$: PRINT D$:
      HOME : GOTO 10040
10060 IF SN$ = " " THEN 10080
10070 PRINT D$;"BLOAD."
      ;SN$;,"A$D00"
10080 PRINT "PLACE
      SCREENWRITER DISK IN
      DRIVE 1": PRINT "(SPACE)
      TO CONTINUE" ;:GET
      Q$:PRINT D$
10099 RETURN
```

After saving APP2 then reboot the disk. When the menu appears select option number 7 and recustomize your disk. When your file appears on the screen it and the command lines in both the editor and runoff modes will appear in the new character set. These screens are written with the 40 column character set. This will take a bit of adjustment for you if you are using Russian or other non-Roman alphabet languages!

Note: *As later versions of ScreenWriter are released, please contact Kim Smith for any changes which concern use of special character fonts.*

APPENDIX B

Sources of Programs

The programs mentioned in this paper should be available at any store which sells Apple computer software. If they are not available they can be ordered from the software companies listed below.

ScreenWriter][—Retail Cost $129.95
Sierra On-Line
Sierra On-Line Building
Coarsegold, CA 93614
(209) 683-6858

Combined Graphics Writer—Retail Cost $54.95
Computer Station
11610 Page Service Drive
St. Louis, MO 63141
(800) 325-4019 (Orders only)
(314) 432-7120

DOS 3.3 Tool Kit—Retail Cost $75.00
(The program ANIMATRIX is on this disk)
Apple Computer Inc.
10260 Bandley Drive
Cupertino, CA 95014
(408) 996-1010

Computer
Assisted
Language
Instruction
COnsortium

WELCOME TO CALICO

The response to the formation of CALICO (Computer Assisted Language Instruction Consortium) has been excellent; in fact, the interest expressed to date has been much more than anticipated. While the participants who attended the recent ET and High Technology symposium are looking forward to the next annual conference, there is a definite need to support the goals of our consortium now so that we may establish an exemplary organization and journal.

Specific tasks of our consortium include:

1. **Enlisting the support of leaders** in the language teaching community. Please share your ideas with us so that we may grow.

2. **Establishing a clearinghouse** for dialog among professionals interested in applying high technology to the solution of problems identified in the teaching and learning of languages.

3. **Collecting and disseminating information** that will enable interested vendors, researchers, teachers, and administrators to share their ideas on the most effective applications of technology toward the improvement of language teaching and learning.

4. **Cooperating** with vendors, universities, courseware cooperatives, publishers, libraries, professional organizations, and individual members of CALICO and subscribers to the journal.

5. **Providing descriptions and evaluations** of available hardware, software, courseware, and peripherals applicable to CAI in language teaching and learning.

6. **Promoting the awareness and use of technology** in diverse areas of research and instruction within a variety of language learning environments.

7. **Reporting on research and advancements** in the areas of speech synthesis, machine translation, foreign-language character display, artificial intelligence, information retrieval language processing, and other areas of concern as expressed by members of the consortium and readers of the journal.

8. **Discussing** [illegible]ations and state-of-the-art advances in technology in ways that are meaningful t[illegible]hers and administrators seeking to improve the effectiveness of language instructional programs.

P.O. Box 7310, University Station, Provo, Utah 84602 (801)378-6533

9. **Contacting universities and donors** of scholarship funds for students enrolling in courses emphasizing applications for technology to language teaching and learning.

10. **Relating to concerns expressed** by special interest groups concerned with the applications of language and technology to specific career needs.

11. **Drafting a constitution and by-laws.**

Since we are publishing a quarterly journal in September, December, March and June:

1. **We are soliciting articles for publication.** A few areas of interest include:
   - Selection/Design and Use of CAI Hardware and Courseware
   - Teaching and Learning/Language with CAI
   - Guidelines for Design and Development of CAI Courseware
   - Interactive Language Learning Strategies
   - The Benefits and Limitations of CAI language instruction in different skill areas based on different delivery media
   - Effectiveness of CAI in language learning
   - Maximizing the Effectiveness of Scoring and Recordkeeping
   - Others: Send us your ideas; better yet, send us a good article.

2. **We are accepting advertising** from vendors of equipment, courseware publishers, courseware development programs, university study programs in instructional science, computer science and related areas, teacher education programs, etc.

3. **We are compiling mailing lists.** Please complete the **CALICO Questionnaire** so that your name is added to our mailing list.

4. **We are preparing the first edition of our journal for release by June 1, 1983. There will be no charge for this edition. (Subscriptions will begin with the September, 1983 issue.)**

REFERENCES

Allen, Brockenbrough S. "The Video-Computer Nexus: Toward an Agenda for Instructional Development," *Journal of Educational Technology Systems*, Vol. 10, No. 2, 81-99.

Bernhardt, Lewis. "Computer-Assisted Instruction in Russian-Language Courses." *Russian Language Journal*, 27 (1973), 18-26.

Birdsong, David. *Computer-Assisted and Programmed Instruction in Foreign Languages: A Selected, Annotated Bibliography*. CAL-ERIC/CLL Series on Languages and Linguistics.

Bork, Alfred and Franklin, Stephen. "The Role of Personal Computer Systems in Education," *Association for Educational Data Systems Journal*, Vol. 13, No. 1, Fall, 1979, 23-24.

Case, R., "Implications of Neo-Piagetian Theory for Improving the Design of Instruction," Chapter in *Cognitive Development and Instruction*, Academic Press, New York, 1980.

Cassell, Douglas A. "Putting the Micro Into Perspective," *Datamation*, August, 1978.

Clausing, Gerhard and Wood, Cecil. "The Computer-Tutor in Media Aided Language Programs," *National Association of Learning Laboratory Directors Journal*, Vol. VIII, No. 3, Spring 1974, 11-20.

Curtin, Constance, et al. "Teaching the Translation of Russian by Computer," *Modern Language Journal*, 56 (1972), 354-360.

Decker, Henry W. "Computer-Aided Instruction in French Syntax," *Modern Language Journal*, 60 (1976), 263-267.

Emmett, Arielle. "Overcoming Computer Resistence," *Personal Computing*, Dec. 1983, 80.

Fox, J. "Computer-Assisted Learning and Language Teaching," *Audio-Visual Language Journal*, 10 (1972), 103-105.

Hausmann, Kevin. "An Overview of the Effect of Computers in Post Secondary Instruction," *AEDS Monitor*, Vol. 18, No. 4, Nov. 1979, 32-37.

Hight, William C. "Computers in Management," *Video Manager*, Vol. 7, No. 5, May 1984, 18-33.

Hope, Geoffrey. "Elementary French Computer-Assisted Instruction," *Foreign Language Annals* 15 (1982), 347-353. 3 (1981), 1-24.

Huffman, Stanley A. Jr. "An Investment in the Future of Education," *Media Management Journal*, Vol. 3, No. 3, Spring 1984, 1.

Kalbouse, George. "Computer-Assisted Instruction in the Teaching of Russian," *Slavic and East European Journal*, 17 (1973), 315-321.

Kearsley, G., Hunter, B., & Seidel, R. J. "Two Decades of Computer Based Instruction Projects: What Have We Learned?", *Technological Horizons in Education Journal*, Vol. X, No. 3, January 1983, 90-94.

Marty, Fernand. "Reflections on the Use of Computers in Second-Language Acquisition," *System*, 9 (1981), 85-98; 10 (1982), 1-11.

Molnar, Andrew R. "The Use of Computers in Education," *Technological Horizons in Education Journal*, Vol. 3, No. 2, February 1977, 18-22.

Nelson, G. E., et al. "Two New Strategies for Computer-Assisted Language Instruction (CALI)," *Foreign Language Annals*, 9 (1976), 28-37.

Olsen, Solveig. "Foreign Language Departments and Computer-Assisted Instruction: A Survey," *Modern Language Journal*, 64 (1980), 341-349.

Pusack, James P. "Answer-Processing and Error Correction in Foreign Language CAI," *System* 11, No. 1 (1983).

Patton, Peter. "Planning for Microcomputers in Higher Education," *EDUCOM Bulletin*, Fall, 1981, 2-8.

President's Science Advisory Committee. *Computers in Higher Education*, (U.S. Government Printing Office, 1967).

Ricketts, Dick and Seay, Jean. "Assessing Inexpensive Microcomputers for Classroom Use," *Association of Educational Data Systems Journal*, Vol. 13, No. 1, Fall 1979, 90.

Roecks, Alan. "The MICA Project," *AEDS Journal*, Vol. 12, No. 3, Spring 1979, 146-154.

Scanlan, Richard T. "Computer-Assisted Instruction in Latin," *Foreign Language Annals*, 13 (1980), 53-55.

Soper, James H. "Computer-Assisted Instruction in Foreign Languages: Past, Present, and Future," See *Official Program, ACTFL/AATF/AATG Joint Annual Meeting*, 25-27 November, 1982, 74.

Steffin, Sherwin. "The Quest for Computer Literacy," *Softline*, Vol. 2, No. 3, January 1983, 21.

Stevens, Vance. "A Report of a Project Illustrating the Feasibility of Video/computer interface for Use in ESL," *Computer Aided Language Learning & Instruction Consortium Journal*, Vol. 1, No. 1, June 1983, 27-30.

Taylor, Heimtraut F. "Students' Reactions to Computer Assisted Instruction in German," *Foreign Language Annals*, 12 (1979), 289-291.

Thomas, David B. "The Effectiveness of Computer-Assisted Instruction in Secondary Schools," *AEDS Journal*, Vol. 12, No. 3, Spring 1979, 103-116.

Thomas, David B. and McClain, Donald H. "Selecting Microcomputers for the Classroom," *AEDS Journal*, Vol. 13, No. 1, Fall 1979, 55-79.

Turner, Ronald C. "Carlos: Computer-Assisted Instruction in Spanish," *Hispania*, 53 (1970), 249-252.

Twarog, Leon and Walters, Garrison. "Mastery-Based, Self-Paced Instruction in Foreign Languages at Ohio State University," *The Modern Language Journal*, Vol. 65, No. 1, Spring, 1981, 1-23.

Vinsonhaler, J. and Bass, R. A. "A Summary of Ten Major Studies of CAI Drill and Practice," *Educational Technology*, July 1972, 29-32.

Weible, David M. "Teaching Reading Skills Through Linguistic Redundancy," *Foreign Language Annals*, 13 (1980), 487-493.

Weissmann, Stephen M. "SuperPILOT," *InCider*, Vol. 1, No. 9, September 1983, 54-64.