DOCUMENT RESUME

ED 102 940 IR 001 613

AUTHOR	Nievergelt, Jurg
TITLE	Interactive Systems for Education: The New Look of CAI.
INSTITUTION	Illinois Univ., Urbana. Dept. of Computer Science.
PUB DATE	Sep 75
NOTE	8p.; Invited paper to be presented at the IFIP second world conference on Computer Education (Marseilles, France, September 1975)
EDRS PRICE	MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS	Autoinstructional Programs; *Computer Assisted Instruction; *Computer Programs; Computers; Computer Science; *Computer Science Education; *Evolution; Man Machine Systems; Problem Solving; Programing; Speeches; Teaching Machines; *Time Sharing
IDENTIFIERS	*PLATO; TICCIT

ABSTRACT

Computer-assisted instruction (CAI) during the decade of the 60's was characterized by a number of limiting factors: insufficiently powerful computers and terminals, restriction to a few rigid "teaching strategies," and the splitting of resources among too many projects below critical size. During the present decade, CAI has undergone a remarkable change, due to a fair extent to two large-scale projects, PLATO and TICCIT, which differ in significant respects from earlier approaches to CAI. Some key aspects of this large-scale experiment involving PLATO and TICCIT are discussed, and some tentative conclusions, based on limited experience with actual instruction, are drawn. As an example of the use of such large systems, an automated system developed on PLATO for teaching computer science is discussed. (Author)

INTERACTIVE SYSTEMS FOR EDUCATION -- THE NEW LOOK OF CAI

Jurg Nievergelt
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801, USA

Computer-assisted instruction during the decade of the 60's was characterized by a number of limiting factors: insufficiently powerful computers and terminals, restriction to a few rigid "teaching strategies", and the splitting of resources among too many projects below critical size.

During the present decade, CAI has undergone a remarkable change, due to a fair extent to two large-scale projects, PLATO and TICCIT, which differ in significant respects from earlier approaches to CAI. Some key aspects of this large-scale experiment involving PLATO and TICCIT are discussed, and some tentative conclusions, based on limited experience with actual instruction, are drawn.

As an example of the use of such large systems, an automated instructional system developed on PLATO for teaching computer science is discussed.

1. TWO VIEWS OF COMPUTER-ASSISTED INSTRUCTION

Word has come down from the Dean
That, by use of the teaching machine,
    Young Oedipus Rex
    Could have learnt about sex
Without ever bothering the Queen.

(Published in the Alumni Review of Hamilton College in honor of alumnus B. F. Skinner, Class of 1926, who has become the most famous proponent of teaching machines.)

One can distinguish three stages in the process by which a new discovery affects society.

"In the first stage, we use a new discovery to do something we already do, but better. We think of an overhead projector in place of a blackboard, a time-sharing system in place of a teacher in a tutorial or Socratic mode, a video tape in place of a lecturer, a telephone in place of direct voice, and a computer in place of a skilled classroom administrator.

All important discoveries go through two stages beyond this first one. The second stage is when we employ the discovery to do something new in a new way; the third is when we modify our life pattern to take the discovery into account. I will not pretend to be able to foresee the third stage, but I think I can predict a piece of the second.

A computer can simulate a phenomenon, then present to the student the result of the simulation and allow him to study the phenomenon by changing its characteristics.

... perhaps the most moving and impressive show I have seen is a simulation of the universe, and it is still in my eyes. A random population of uniform bodies obeying Newton's law was injected in a simulated space. Then in a few seconds, right in front of me, the moving bodies in apparently random motion acquired shape. It was thrilling to see spiral, nebular, globular galaxies appear in completely unexpected fashion. I did not truly realize that the shape of the universe was defined in its gross morphology by Newton's law alone."

(Eugene G. Fubini in "Computers, communications, and the public interest", edited by M. Greenberger, The Johns Hopkins Press, Baltimore, 1971, pp. 131-132.)

The quotations above characterize two conflicting views of CAI (computer-assisted instruction) commonly held today.

The limerick is a spoof on the exaggerated expectations being voiced periodically since the early sixties, when the first generation of CAI systems gave the dormant field of "teaching machines" a new direction. The second quotation expresses an optimistic expectation that this new form of educational technology bears great possibilities, but we are only beginning to guess what these are. The passage also suggests an explanation for the fact that computers have not yet had a great impact on instruction: it is because bonds of tradition and lack of imagination have kept us from using them in proper ways.

A critic can readily dismiss Fubini's view as optimism unsupported by facts. Experience with CAI is sufficiently recent and inconclusive that the history of this field can be interpreted equally well as a promising start on the problem of finding the right way to exploit a powerful new medium, or, at the other extreme, as a sequence of failures, each failure followed by a change of position and redefinition of the problem by those unwilling to draw the inevitable conclusions.

I will not attempt to settle the dispute between the optimistic and the pessimistic views

Invited paper to be presented at the IFIP 2nd World Conference on Computer Education, September 1975, Marseilles, France.

expressed above. In my opinion CAI can neither be justified nor rejected on the basis of achievements to date. The best one can do is to look at some representative activities in this field, past and present; to try to understand the reasons for the achievements and the failures; and then to form an opinion, based on partial evidence, as to whether or not computers will play a significant role in education. And if so, what role they will play, to what extent, and how soon this might happen.

I share the optimistic point of view about CAI, and in this paper will present some reasons for it. I wish to emphasize that the term "CAI" will be interpreted in a broad sense: in principle, any way in which a computer can assist instruction will be considered a part of this field; in practice, I will limit my discussion to the interactive use of computers, primarily because I see this as the most interesting, and somewhat controversial, mode of use. Perhaps the term used in the title, "interactive systems for education", is a better descriptor for the topic discussed in this paper. If so, it might come to replace such abbreviations as CAI, CAL, CMI, which are often interpreted in a narrow sense (several visitors to whom I have shown our instructional system ACSES have told me "but you are not doing CAI!").

2. STAGES IN THE DEVELOPMENT OF CAI

> Those who cannot remember the past
> are condemned to repeat it.
> Santayana

History is a matter of fact as well as interpretation. It is the latter, the particular way we look at history, that determines what we learn from past experience, and hence how the past influences our approach to the task of shaping the future. Giving a selective (and possibly subjective) account of some of the main stages in the development of CAI is the best way to explain my views about current and future interactive systems for education.

2.1 Ideas that shaped early CAI

The intellectual environment which gave rise to the first generation of CAI systems in the early 60's was strongly influenced by the programmed instruction movement, e.g. Skinner (1954). CAI was seen by many as a direct continuation of the mechanical teaching devices, Pressey (1926), with the processing and decision-making ability of computers finally providing the flexibility whose lack had denied success to the mechanical realizations.

The dominant mood of optimism among many workers in CAI was rationalized by arguments along the following lines: 1) education is a labor-intensive activity, 2) technology applied to other labor-intensive areas in the past has increased cost-effectiveness drastically, 3) with programmed instruction as a teaching strategy and computers as a delivery device, a technology of instruction has finally arrived, and hence, 4) CAI will drastically improve education in the near future. "Improve" had three aspects: 1) more effective instruction (learn better and faster), 2) cheaper instruction, and 3) overcome the shortage of teachers.

The argument was sufficiently alluring that it drew all kinds of scientific, technological, and commercial interests into the field of CAI. As prominent examples from this early stage of CAI let me mention the Stanford project, particularly the arithmetic drill-and-practice program, Suppes (1968), and the IBM 1500 CAI system. More information about CAI activities during this period can be found in Atkinson (1969).

2.2 Reassessing the situation

Reality did not live up to expectations. By 1970 a number of facts and conclusions that dampened the early optimism and indicated that a reorientation was necessary were gaining acceptance:

1) CAI had not caught on as a means of routine instruction; on the contrary, a number of CAI research projects of subcritical size were discontinued.

2) Programmed instruction and drill were not a universal technology of instruction, but had rather limited domains of applicability.

3) Restriction to a few fixed teaching strategies appeared to be unreasonable. Programmed instruction and drill in particular, with their rigid control of the dialog by the program, should yield to (or at least not exclude) modes where the user controls the dialog, such as inquiry and simulation.

4) CAI was still significantly more expensive than conventional instruction.

5) The goal of writing portable courseware in order to amortize the cost of lesson-preparation among more users was not within sight.

6) The computer resources (terminal, processor, software) required to implement an effective instructional dialog had generally been underestimated. In particular, line-oriented terminals appeared to be insufficient for many subject matters; a graphics display emerged as a necessity.

7) Resources had been diluted into too many projects of insufficient size; CAI research and development should be carried out by sizable groups of system designers and authors.

Not everybody could be expected to agree with all of these points, but I believe these points express fairly well the collected wisdom that workers in the field think they have learned from the early CAI experiments. In section 3 I will discuss some current CAI projects, and we will see that each one of them has chosen to respond to some of the points mentioned above. There is a great diversity among leading current CAI projects, but they all share a concern to get away from the failings of early CAI, each in its own way. Before doing so, however, I wish to touch upon an area which started out as a direct antagonist to CAI, but is currently in a state of coexistence with it. The two are destined to merge into the general field of interactive systems for education.

2.3 Computers, problem solving, and general education

During the mid-sixties, concurrently with the early CAI wave and undoubtedly spurned by a missionary drive to enlighten the CAI enthusiasts, another movement to bring computers into education gained visibility. Its main premise was that a computer is such a great tool and toy that its greatest educational impact will materialize only if students are given full control over it, that is, are programming it to solve problems of their own choice.

The most widely known representative of this movement is the LOGO project at the Massachusetts Institute of Technology, Papert (1970). Perhaps the most eloquent statement of the position that the main role of computers in instruction is as a subject to be taught rather than as a medium for presentation of instructional material was made by Luehrman (1972) in a paper with the provocative title "Should the computer teach the student, or vice-versa?".

This rhetorical question implies a decision which, fortunately, need not be made. The attitude that one need not make a sharp distinction between the use of computers as an instructional device and as a tool for problem solving is gaining ground. The SOLO project at the University of Pittsburgh, Dwyer (1972), has long combined a "dual mode" (the student interacts with a teaching program designed to guide him by the hand) with a "solo mode" (he uses the computer on his own, that is, interacts only with software available to any computer user). When "teaching machine" and "problem solving tool" are viewed as dual mode and solo mode, respectively, it becomes clear that the distinction is one of degree, not of principle. The antagonism of the "problem-solving" exponents towards CAI can only be understood historically, as a reaction against the trivial use of computers as "electronic page turners". If one accepts the statement that the student should interact with the computer in whatever way is most pleasant, interesting, and conducive to learning, then it is evident that anything from drill to unsupervised programming can be a reasonable use of computers in education.

3. THE DIVERSITY OF CURRENT CAI PROJECTS

> If you don't know where you are going,
> any road will take you there.

While there was widespread agreement in the early 70's that CAI had to undergo some major changes (see section 2.2) in order to succeed, there was considerable diversity of opinion about what changes should be done at all, or were most urgent. At the risk of oversimplification, I present the following summary of opinions held by workers of different backgrounds.

Administrators: consolidate CAI research in a few large projects, develop portable CAI systems.
Educational experimenters: drop traditional CAI, teach the use of computers as problem solving tools.
Educational theoreticians: drop teaching strategies which use strong program control, emphasize learner control.
Engineers: develop better hardware, in particular terminals.
Programmers: drop traditional CAI languages, use or develop author languages that are close to general-purpose high-level programming languages.

All of these views are represented in the wide spectrum of today's CAI projects. In 1971 the U.S. National Science Foundation committed itself to support two CAI projects on a large scale: PLATO at the University of Illinois, and TICCIT at the MITRE Corporation. Both of these projects utilize hardware which is significantly better than what was available in the 60's, particularly the terminals. The fact that opinions differ on exactly what the most suitable hardware configuration for CAI is, is much less significant than the common recognition that insufficient hardware was one of the drawbacks of early CAI. The two projects differ completely, however, in their attitude towards the preparation of instructional material.

While TICCIT is proud of the uniform style of its courseware, based on a theory of instruction and generated according to a systematic process, Bunderson (1972), PLATO is equally proud of the "Darwinian approach" most of its authors take towards lesson writing: try everything you can think of, and if you keep your eyes open and are prepared to throw away unsuccessful material, the good stuff will survive. I expect that TICCIT will generate courseware of uniform quality, and I know that PLATO has some excellent lessons along with a fair amount of poor material. Given that we are at a very early state of development of the art of lesson writing, I prefer the latter situation.

NSF also supports on a smaller scale the continued development of PLANIT, Feingold (1967), with the goal of establishing a hardware-independent CAI system, and a user community to exchange instructional material. The major drawback of this approach is that, when lessons are written without knowing what type of terminal is available, one can assume only that it accepts one line at a time, of unspecified length. This rules out any use of pictures, and the mechanics of interaction are often awkward. Another drawback, possibly temporary, is that most time-sharing systems available today do not accommodate efficiently nor effectively the job-mix characteristic of CAI: many short jobs requiring almost instant response. But the first drawback is the decisive one, and because of it I do not expect that hardware-independent CAI systems will become successful in the foreseeable future.

At the other end of the spectrum with respect to the quality of the terminal, is the Smalltalk system developed at the Xerox Palo Alto Research Center, California. It is an experimental system designed to teach young children about computers -- and does so by providing a dedicated computer with a fast graphics terminal for each user. While such a system is not economically feasible today for use in schools, it may well become so within a decade.

CAI projects in other countries exhibit a similar diversity of characteristics as do the ones in the US. In Germany for example, where CAI is currently very active, a large CAI system is in routine use in Heidelberg where all courseware is written in APL. At the University of Karlsruhe, Schmitt has developed a hardware-independent CAI system, LEKTOR, which features a modern high-level programming language; and Bode has implemented a learner-controlled information system, LEGIS. Several other systems use more traditional CAI systems and languages.

4. WHAT DOES THIS ALL HAVE TO DO WITH EDUCATION?

The first problem one has to face when attempting to understand a rapidly changing, heterogeneous movement such as CAI, and to predict its future development, is to clarify the issue as to what will be seen as a continuation of the current movement, and what will be classified as belonging to some other subject. Depending on whether a certain change is seen as evolution or revolution, a movement may blossom or die merely as a matter of definition. If one's conception of CAI is too restrictive, one will probably come to the conclusion that the field is dying or already dead, or, at best, that it is a specialized technique whose importance has been grossly exaggerated, but is really applicable only to a narrow domain of instructional activities.

Let us look at some of the major phases of this movement, whose history we have traced briefly. It began to assume an identity when the first mechanical teaching machines were designed; it spanned the programmed instruction area, with its emphasis on "active response", "immediate feedback", "knowledge of results"; it gave rise to some of the first time-sharing systems in operation; and it is now beginning to include communication networks of sizable proportion. The conception presented here is that the main characteristic common to all of these phases is that they are dealing with devices for interactive use in education. Nothing more, nothing less.

All other aspects appear to have been less persistent. There is a period when learning theory is emphasized, and the field is an appropriate testing ground for psychological theories and research. There are the moments when educational issues come to the fore, such as educational objectives, or evaluation. There are the trends to use computers as tools for problem solving, and as graphic communication media.

All of these aspects are present, none should be allowed to dominate. Whenever one of these aspects was pursued to the exclusion of all others, the movement appears to have spawned off a branch that whithered away.

If one accepts the point of view that we are dealing with all aspects of interactive systems for education, rather than with a particular subset thereof that fits preconceived constraints, then I find it not too difficult to guess what the near future will hold in store, say the next 10 years. Two points stand out:

- a rapid spread of sophisticated interactive systems in schools of all kinds, businesses, and later homes, to be used for education, information services, and increasingly, entertainment
- lesson material that is prepared without any particular ideaology, theory, or self-imposed constraints, according to the motto "anything goes that you can dream up and program".

If these two guesses are accepted as working hypothesis, then one can deduce almost with certainty that this field of interactive systems for education will have the following characteristics over the next decade:

- computers with fancy terminals will be seen more and more as communications media which offer powerful possibilities not available in conventional media, such as newspapers, books, radio, television, films. Hence this field will develop its own profession or craft, of authors, producers or whatever they will be called.
- it will be a very fertile ground for application of advanced computer science techniques,

such as those developed in artificial intelligence research, computer graphics, information systems, and data base management systems. It is most unlikely that over the next decade, this field will develop in scholarly, theoretical directions, as it has for a brief period in the past. Its potential commercial importance will push for rapid development; the attraction of experimenting with new modes of presentation previously unavailable will lead to emphasis on clever gadgetry and ad hoc techniques.

Ultimately, the field of interactive systems will probably assume an identity of its own, and applications to different areas such as education, entertainment, management, or anything else, may be considered to be only variations on the same theme.

The question in the title of this section was not intended to be rhetorical, but rather to raise a fundamental issue. While the question may have no objective answer, everybody involved in the field of CAI should answer it to his own satisfaction, because his attitude towards this issue will profoundly influence his view of and approach to CAI.

My own answer is: "No more and no less than books, blackboards, films, and other media have to do with education. What is really at stake is that we have to develop a craft of writing interactive programs for communication of all kinds".

5. A CASE STUDY: AN AUTOMATED INSTRUCTIONAL SYSTEM FOR COMPUTER SCIENCE

Four years ago I started planning the design of a sizable automated instructional system, as a result of a number of serendipitous coincidences. One of these was that the PLATO-IV system, then in the design and construction stage, promised to become a very interesting interactive system (it has lived up to this promise). Another main reason was the rapid growth in the enrollment in our introductory programming courses (it is now at about 2000 students per semester), which forced us to teach these courses in large lecture sections -- an environment that is not conducive to learn a skill which requires active participation. The hope to improve the quality of these courses by providing a unique system where the student can learn on his own, and actively practice anything he has learned anytime he wishes, gave the final impetus for the start of this project.

A ground rule for the design was to look at <u>every</u> aspect of an introductory programming course, instructional or administrative, and, without any preconceived ideas about what should be assigned to a computer and what should not, try to automate everything in any way that looked feasible; with the expectation that actual use would later show which tasks had been successfully automated, and the unsuccessful ones would simply whither away.

After three years of implementation, with an effort in excess of twenty man-years which produced approximately a million words of code, our system ACSES (Automated Computer Science Education System) is now experiencing its first large-scale use. In the spring semester of 1975, about 650 students are spending 2 hours per week at a PLATO terminal as part of a 3-hours-per-week programming course. Experiments with smaller numbers of students over the past year and a half had given us no conclusive data for comparing the performance of students off and on PLATO -- and I don't expect any conclusive findings in the near future. Plain observation, however, makes it clear that a majority of the students like the experience of working on PLATO -- this was true even of our first experimental groups who had helped us debug our system (they were not volunteers). The high degree of student satisfaction gives us the confidence that our system is doing <u>something</u> right, but it is too early to pass judgment on it.

ACSES is not a course in the conventional sense, does not involve any curriculum development, and does not have any educational philosophy of how one should or should not teach. It is the equivalent of a library and a laboratory, designed to turn a PLATO terminal into a rich environment where you have at your fingertips many useful things for learning about computers and programming.

The structure of ACSES can best be explained by grouping its many parts into five major components.

1) The <u>library of lessons</u> contains over 100 relatively self-contained instructional modules on such topics as conventional programming languages (PL/1, Fortran, Basic, Cobol, APL), programming concepts (such as loops, blockstructure, recursion, etc.) explained without reference to any particular programming language, and computer applications to various areas (business data processing, numerical and statistical computation, simulation). Different lessons use entirely different styles of presentation and teaching strategies. Authors frequently try to use animated graphics to display the behavior over time of a model which the student can manipulate. Fundamental concepts of programming are often introduced via a mini language which has been carefully designed to highlight that one idea without extraneous topics; in this case the lesson will contain an interpreter so that the student can write and run short programs. Such lessons coexist with games, drills, tutorials. We encourage authors to be creative and unconventional in designing a lesson, and the only thing we attempt to standardize is the function of certain keys that allow the user to move around lessons in a uniform way.

2) The programming laboratory, designed by T.R. Wilcox (1975), is an interactive system for program entry, editing, execution and debugging. It is a table-driven system adaptable to any conventional high-level language, and has been implemented so far for subsets of PL/1, Fortran, and Cobol suitable for a first programming course. Particular attention was given to error diagnostics and debugging features, Davis et. al. (1975). During program entry and editing a syntax check is made for every character typed in, with increasingly specific error messages available at the press of a key. For the analysis of runtime errors the student may engage in a dialog in which the system traces the flow of control backwards, asking the student questions about the validity of certain values which might have caused the error.

3) The guide is an advising system which converses in simple English sentences about the following topics:
   - instructional material available in the library (what it is about, how lessons are related to each other, etc.)
   - what the student is supposed to do on the system (if he is registered in one of our courses)
   - what the student has done so far
   - how different computing concepts are related to each other, and where they belong in a classification of the subject matter.

   The guide has been implemented by J. Pradels (1974) and D. Eland.

4) The exam system, currently still under development, is intended to automatically generate problems according to an instructor's specification, to grade the student's solution, and administer the exam (data collection and security aspects). A part of it which uses the programming laboratory to interactively grade programs written by students is complete, Barta (1975).

5) The communications system allows students, instructors, and the management of ACSES to converse to each other on-line from terminal to terminal, or to leave messages. This facility uses the fact that PLATO terminals can communicate with each other.

A few special projects deserve separate mention. Among the lessons there are a few which might be called automatic tutors. These are artificial-intelligence-type programs which have a specific domain of knowledge and attempt to follow the student's thought process as he writes a program, giving hints and corrections. One, written by Mateti, is concerned with programs for sorting; another with PL/1 programs for symbolic differentiation, Danielson (1975).

As one might expect from a group of computer scientists, we are also experimenting with a high-level author language, KAIL, designed and implemented by D. Embley and W. Hansen. A few lessons have been written in KAIL, but all the rest of ACSES is written in Tutor, the standard language available on PLATO. Tutor was designed by P. Tenczar, and is described in Sherwood (1974).

6. CONCLUSION

The design and implementation of ACSES has been a valuable experience in interactive systems, particularly with respect to the problem of designing effective dialogs between man and machine. In no application other than education would this problem have been of such central importance: because the student cannot be assumed to be a trained operator, the mechanics of conducting the dialog must be easy; in order to retain the student's attention for long periods, the dialog must be interesting and pleasant; to assure that he learns, the dialog must be somewhat demanding. How do you satisfy all of these requirements?

When I prepared myself for this project by reading much of the literature on CAI, and some on educational technology in general, I found very little that was of direct use for designing an automated instructional system. I came to the conclusion that there is no systematic body of knowledge which is of relevance to such a task. I am afraid that this paper does not change this situation at all. The advice I might give to someone intent on building a computer-based instructional system could be summed up in a few phrases: get the best terminals you can pay for, good programmers, try everything out in actual instruction as soon as possible, and follow your nose.

7. REFERENCES

Atkinson, R. C. (1969), "Computer-assisted instruction: a book of readings", Academic Press.

Barta, B. Z. and Nievergelt, J. (1975), "An interactive system for automatic examination of programming skills", in these proceedings.

Bunderson, C. V. (1972), "Mainline CAI, necessary but not oppressive", AFIPS Conference Proc., Vol. 40, 399-405.

Danielson, R. and Nievergelt, J. (1975), "An automatic tutor for introductory programming students", Proc. Fifth Symp. on Computer Science Education, SIGCSE Bulletin, Vol. 7, No. 1.

Davis, A., Tindall, M. H. and Wilcox, T. R. (1975), "Interactive error diagnostics for an instructional programming system", Proc. Fifth Symp. on Computer Science Education, SIGCSE Bulletin, Vol. 7, No. 1.

Dwyer, T. A. (1972), "Teacher/student authored CAI using the NEWBASIC system," Comm. ACM, Vol. 15, No. 1, 21-28.

Feingold, S. L. (1967), "PLANIT - A flexible language designed for computer-human interaction", AFIPS Conference Proc., Vol. 31, 545-552.

Luehrman, A. W. (1972), "Should the computer teach the student, or vice-versa?", AFIPS Conference Proc., Vol. 40, 407-410.

Papert, S. (1970), "Teaching children thinking", Proc. IFIP World Conf. on Computer Education, 73-78.

Pradels, J. L. (1974), "The guide: an information system", Report 74-626, Dept. Computer Science, Univ. of Illinois, Urbana.

Pressy, S. L. (1926), "A simple apparatus which gives tests and scores and teaches", School and Society, Vol. 23, 373-376.

Sherwood, B. A. (1974), "The TUTOR language", Computer-based Education Research Lab., Univ of Illinois, Urbana.

Skinner, B. F. (1954), "The science of learning and the art of teaching", Harvard Educational Review, Vol. 24, 86-97.

Suppes, P. and Morningstar, M. (1972), "Computer-assisted instruction at Stanford, 1966-68", Academic Press.

Wilcox, T. R. (1975), "An interactive programming system for education", forthcoming.

ACKNOWLEDGMENT

I am most grateful to the Computer-based Education Research Laboratory at the University of Illinois for their support during the entire development phase of the ACSES system on PLATO, and for providing a stimulating intellectual environment which has shaped many of my views. I would like to acknowledge particulary the help of D. L. Bitzer, P. J. Tenczar, B. A. Sherwood, W. M. Golden, R. W. Blomme, D. Alpert, F. M. Propst.

The development of ACSES would have been impossible without the contributions of many people at the Department of Computer Science, in particular T. R. Wilcox, H. G. Friedman, R. G. Montanelli, B. Z. Barta, R. L. Danielson, A. M. Davis, D. R. Eland, D. W. Embley, W. J. Hansen, F. Izquierdo, P. Mateti, J. M. Milner, J. L. Pradels, E. M. Reingold, M. H. Tindall, D. S. Watanabe, L. R. Whitlock. The encouragement and support provided by E. D. McWilliams and J. N. Snyder throughout this project have been essential.

This work was supported in part by the National Science Foundation under grant EC-41511.