DOCUMENT RESUME

ED 027 751 24 EM 007 155

By-Easley, J.A., Jr.

A Project to Develop and Evaluate a Computerized System for Instructional Response Analysis; Project SIRA. Final Report.

Illinois Univ., Urbana. Computer-Based Education Lab.

Spons Agency-Office of Education (DHEW), Washington, D.C. Bureau of Research.

Bureau No-BR-5-1179

Pub Date Sep 68

Contract-OEC-6-10-184

Note-49p.

EDRS Price MF-$0.25 HC-$2.55

Descriptors-Academic Performance, *Computer Assisted Instruction, Curriculum Development, Data Analysis, *Evaluation Techniques, Population Trends, Response Mode, *Systems Approach

Identifiers-PLATO System, *Project SIRA, System for Instructional Response Analysis

Project SIRA (System for Instructional Response Analysis) used a systems approach to develop a complete range of programs and techniques both for evaluation of student performance and for evaluation and revision of computer-assisted instruction (CAI) lesson material. By use of the PLATO computer-based instructional hardware system at the University of Illinois, SIRA developed techniques depending on pre-processed or pre-selected data and two advanced techniques making use of completely unprocessed student response data as input for general pattern detection methods. While rapid evolution of CAI software and changes in author and user population have altered the utility of early SIRA routines, the most useful SIRA-instigated functions have survived by being incorporated into present software. Appendices: glossary of SIRA programs, student reactions to extended CAI sessions, curriculum development based on student responses, and revision of course based on student response analysis. (TI)

Final Report

Contract No. OE-6-10-184

A PROJECT TO DEVELOP AND EVALUATE A COMPUTERIZED SYSTEM FOR INSTRUCTIONAL RESPONSE ANALYSIS

J.A.EASLEY, JR.
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS 61801

SEPTEMBER, 1968

U. S. DEPARTMENT OF HEALTH, EDUCATION, AND WELFARE

Office of Education Bureau of Research

Final Report

Contract No. OE-6-10-184

A PROJECT TO DEVELOP AND EVALUATE A COMPUTERIZED SYSTEM FOR INSTRUCTIONAL RESPONSE ANALYSIS

Project SIRA

J.A. Easley, Jr.

Computer-based Education Research Laboratory
University of Illinois
Urbana, Illinois

September, 1968

The research reported herein was performed pursuant to a contract with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

CONTENTS

PREFACE

I wish to acknowledge the special debt owed to Mr. R. A. Avner, Research Associate in the Computer-based Education and Research Laboratory, who has directed the efforts of programmers and other workers on Project SIRA for the past year, and, under my general supervision, compiled this final report. During my six months sabbatical leave, spent mainly at the University of Geneva in Switzerland, Professor Max Beberman has provided overall supervision. Professor Donald L. Bitzer, Director of the Computer-based Education Research Laboratory, has throughout been a ready source of inspiration and encouragement. The many programmers who worked on Project SIRA deserve a special vote of thanks for their willingness to devote long hours with inconvenient schedules, tolerate last minute changes to accomodate users of the PLATO system and accomodate to systems modifications. To all these people we are indebted for a sustained effort to help PLATO users improve their work through evaluation. We hope that our efforts will save others costly investments.

J. A. Easley, Jr.
Principal Investigator

SUMMARY

During the years 1964-1967 at the University of Illinois Computer-Based Education Research Laboratory, project SIRA used a systems approach to develop a complete range of programs and techniques both for evaluation of student performance and for evaluation and revision of CAI lesson material. Two of the most advanced SIRA techniques developed made use of completely unprocessed student response data as input for general pattern detection methods. More generally useful techniques depend on pre-processed or preselected data. While rapid evolution of CAI software and changes in the user population have altered the utility of earlier SIRA routines, the most useful SIRA-instigated functions have survived by being incorporated into present software.

INTRODUCTION

Project SIRA began in July, 1964, as an effort to facilitate formative evaluation of printed texts. This objective was to be attained by use of the PLATO computer-based instructional system at the University of Illinois to gather and analyze student responses to texts. Starting with this limited goal, SIRA has since produced a variety of general techniques intended to shorten the extensive and expensive lead time needed for the preparation of all types of instructional programs on the PLATO system. These techniques fall into two classifications: (1) methods for processing and analyzing data, and (2) methods for analysis and correction of programs. Methods of the first type speed programming by sharply reducing the necessity for the design of special response analysis routines for each new program. Once evidence of suboptimal performance by a given program is found through analysis of student responses, methods of the second type speed location of lesson design errors and introduction of corrections or modifications of the program.

The shift in the goal of SIRA came about when it was discovered that the methods which would allow attainment of the original, restricted goal were also needed by the entire range of computer-assisted instructional approaches. Thus, with little added effort, a collection of general techniques having widespread benefits could be produced. Separate funding of SIRA by the Research Division of the U. S. Office of Education in October, 1965, allowed SIRA to progress to this point of discovery and take the fullest advantage of the opportunity presented. Project SIRA has continued to study methods of helping authors of instructional programs by taking a systems approach to the problem and developing computer programs to facilitate such work.

Original Approach

Computer-assisted instruction (CAI) has typically entailed cooperation between individuals with quite different backgrounds, most often between a subject-matter specialist and a systems specialist. The profiles of relative expertise in several relevant knowledge areas shown in Table 1 give an impression of some of the types of personnel presently engaged in CAI and their respective contributions.

KNOWLEDGE AREA

SPECIALIST	Specific Subject	General Learning Principles	Evaluation and Data Systems Analysis	Programming and Systems Analysis
Subject-matter (Author)	H	M	M	L
Systems	L	L	M	H
General Education	M	H	M	L
Educational Evaluation	L	H	H	M

Table 1

Relative expertise (High, Medium or Low) of CAI Specialists in various relevant areas.

The person writing the program content (the subject-matter specialist or "author") is expected to be highly competent in the area of the material being taught, moderately competent in general learning principles and evaluation techniques, but only marginally competent in computer techniques. These skills of the author are complemented to a certain extent by those of the systems specialist who is able to provide the special computer requirements for specific teaching programs. With time, the systems specialist is able to produce general programs ("teaching languages" or "teaching logics") which allow future authors to present their materials with a minimum knowledge of computer operations or need for the aid of the systems specialist. Alternatively, such general programs have been produced _a priori_ by interaction between systems specialists and experts in general educational procedures. In either case, the general program is more likely to allow for a flexible approach to CAI than will the logic of a single program designed by one author to meet the requirements of teaching a given type of subject matter. However, in neither case is the problem of evaluation explicitly covered.

Evaluation is an operation which has both formal and informal aspects. Informal evaluation is constantly being carried out during the construction of almost any instructional program. Material which seems to be unclear to the author is altered or rewritten. Points which "experience" indicate as being difficult for students are given more elaboration, and so forth. Formal evaluation, the collection and analysis of empirical evidence, too often appears only after the development is complete. Ideally, evaluation is neither an end in itself nor a luxury that can be lightly dispensed with. The recognized importance of evaluation is reflected in Table 1 which shows this to be the only area in which specialists in subject-matter, systems, and general education all have at least moderate facility. This moderate level of ability is meant to imply that all of these individuals are able to interpret and make intelligent use of reasonably complex evaluative measures. It does not imply that these individuals will necessarily be aware of every pitfall and nuance of specific evaluation techniques relevant to CAI. To meet this need, Table 1 shows a fourth expert, the educational evaluation specialist with a thorough background in measurement, experimental design, and educational and experimental psychology, coupled with moderate experience in computer techniques. The need for such a specialist has evolved, in part, from those attributes of CAI which cause it to be attractive to users of programmed instruction. Programmed instruction has used the techniques and results of experimental psychology to a much greater extent than have other educational methods. Indeed, programmed instruction has evolved largely as the result of an attempt to apply some basic psychological principles to education. A thorough background in these principles is thus a form of insurance against having to retrace dead-end paths and having to re-extricate studies from old pitfalls.

At an early stage in the SIRA project the educational evaluation specialist was seen as a contributor to an enriched form of "author language". This author language would include provision for data collection and evaluation as a part of normal lesson presentation. It was expected that the mere availability of relatively painless evaluation techniques would lead to increased reliance on formal evaluation and better quality lessons. Having made his contribution to a better user language, it was hoped that the educational evaluation specialist, like the systems specialist, could retire from the scene. Recent evidence, to be described in a later section of this report, has shown this hope to be unrealistic. The enriched author language which remains from the efforts of the SIRA project has instead become a major tool of the evaluation specialist who, it seems, must remain very much on the scene.

The Role of SIRA in CAI

Figure 1 shows the information channels between the elements of a teaching system using PLATO hardware and SIRA evaluation techniques. A review of the information available on these channels provides a perspective for viewing some of the formal evaluation techniques in CAI. In this figure the PLATO SYSTEM block represents all of the hardware (e.g., computer, student stations) and that software which is available automatically to every user of the PLATO system. The SIRA TECHNIQUES block represents data programs and routines used on the PLATO computer for all aspects of evaluation of teaching programs. Some routines especially developed for SIRA use have been incorporated into the PLATO software, hence these two blocks overlap to some extent. the AUTHOR block, of course, represents the author or his representative while the STUDENT block stands for the source of any inputs to student stations. All five channels allow a two-way flow of information.

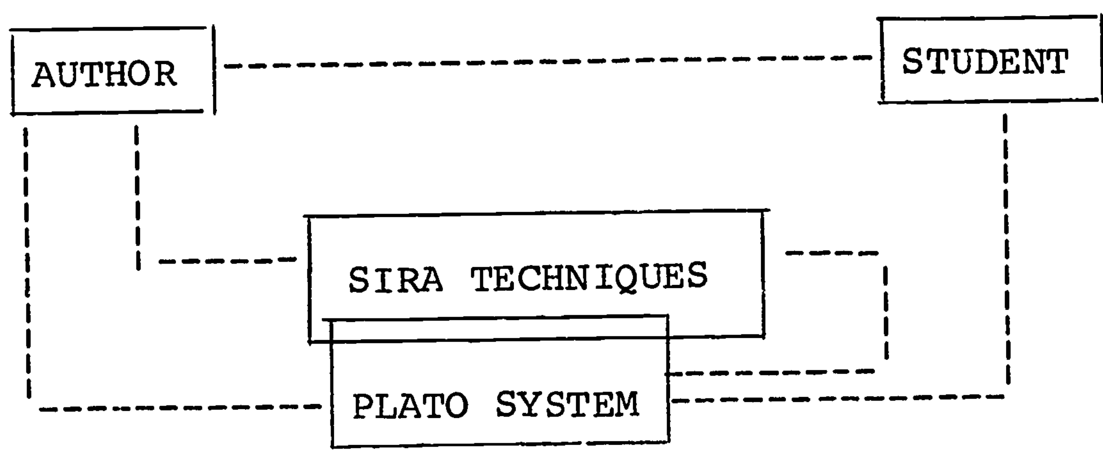

Figure 1. Information channels in a CAI system.

The AUTHOR-STUDENT channel is basic to most formal human learning situations. This channel provides the traditional forms of information such as instructions, questions, comments, and pre- or posttest data gained incidently to the presence of the CAI system. In addition, direct observations such as those done by Yarom (see Appendix II) show some interesting social and psychological responses to learning in this rather novel environment.

The STUDENT-PLATO channel provides the major link for real-time information flow during the learning operation. The PLATO system allows recording of all of this information along with the time of occurrence of both student and system responses. One early program modified for SIRA purposes (SPECTRE) used the stored student-to-PLATO information for later "replay" of student sessions for gross analysis in real time at the same stations used by the students.

The AUTHOR-PLATO channel allows communication between these units at all times. The system also permits information displayed on any student screen to be observed at a remote monitor while the student is responding to the lesson material. The remote observer may, if he needs to, type responses which appear on the student's screen. The author is also able to revise material while a lesson is in progress by use of a SIRA originated technique (AUTHOR MODE), thus allowing mistakes to be corrected or revisions to be made as soon as problems arise. Since students proceed at their own rate it is unlikely that all will be at the same point in a lesson at the same time. Hence it is possible that data for several revisions of the same specific item might be collected from different members of a single class. Such techniques enormously speed up the early development of instructional material. The AUTHOR-PLATO channel can also be used for automatic notification of the author by PLATO of specific types of behavior by students. Thus the author might be informed that student A had not made any response in the past 10 minutes or that student B had requested the aid of the teacher.

The SIRA-PLATO channel together with the AUTHOR-SIRA channel deal with the same general types of information as the AUTHOR-PLATO channel. The SIRA block between the AUTHOR and PLATO serves mainly as a means of making more effective use of information stored by PLATO and more efficient use of directions originated by the AUTHOR. The overlap between the SIRA and PLATO blocks is mainly due to the addition of proven SIRA techniques to the PLATO software package. The modified SPECTRE program mentioned above was an example of such an addition. As a result of the modifications, this system program was easily controllable from any PLATO terminal. The AUTHOR in most instances could view such a replay within seconds after the completion of a lesson, quickly moving to the part of the lesson that interested him and then viewing that part at the most appropriate speed.

Another example of a SIRA program which has been made a part of PLATO's software package is MONSTER, a program which allows several authors simultaneously to edit different computer programs using the PLATO student stations as display and input terminals. In addition to eliminating the physical handling of paper tape or punched cards, MONSTER allows the author to attempt immediately to use his altered program. If the program fails to work, further revisions can be quickly made. Like AUTHOR MODE described above, MONSTER permits a shortcutting of much of the time-consuming drudgery of program revision that is an inevitable part of lesson preparation on advanced CAI systems.

Appendix I describes the more important programs developed by the SIRA project.

Data Evaluation on the PLATO System

The quantity and variety of data that can be produced by even a few students in a CAI setting can be overwhelming. It is perhaps unfortunately possible to think of reasons why every scrap of information might be of some use to someone. For example, there _might_ be significant relationships between learning behavior and hesitations between typing individual letters of words. Also there is the tendency of some workers, who perhaps equate the cost of obtaining data with its value, to place more emphasis on the gathering and storage of data than on its analysis and use. Generally, only a few of the possible variables are of value in any given situation. To the extent that previous work has uncovered these useful variables or classes of variables, the data problem can be simplified by types of processing which retain only the variables of interest.

The ideal in CAI is the complete utilization of the capabilities of the computer to process and analyze all relevant data as they are collected. Yet, a little reflection will show that such an ideal may in fact be incompatible with the early stages of a research program. In order to process data as they are collected one must know which variables are to receive what treatment. A priori selection of variables by means other than empirical evidence is equivalent to the arbitrary rejection of all hypotheses dealing with the non-selected variables. Treatment of all variables is impossible. Even if the physical capability is present, most variables consist of transformations of combinations of raw data and must be explicitly defined.

There is a need to move rapidly yet systematically from the situation in which we are almost drowning in the overflow of (mostly uselsss) data to the point at which a few significant variables are used for control and prediction. But unless the reduction of variables is done in the proper way, there is little assurance that the reduced set of variables will be of any more use than an equal number of variables chosen at random from the original set. Premature variable selection is clearly to be avoided. A given variable chosen on the basis of past classroom experience may have simply been the best of those which were easy to measure. Limitation of response variables to things like "total-time-spent-on-lesson" or "total errors," without evidence that there are no better measures available, would seem to be a mistake.

An examination of the major programs available on the PLATO system in 1967 showed that the problem of selection of useful response measures is neither trivial nor simple.

Status of Response Measures in Early PLATO Programs

During the seven years that PLATO was in operation prior to 1968, over 300 programs were written for the system (Lyman, A Descriptive List of PLATO Programs, 1960-1968, CERL Report X-2, May 1968, CERL, University of Illinois, Urbana). Of these 300, there were 27 major programs which were (1) compatible with the third-generation PLATO III system (thus utilizing the capabilities of a modern CAI system such as PLATO) and which had (2) gone through or were in the process of going through formal evaluation procedures. Excluded from this group of 27 were service programs, demonstration programs, and engineering test programs.

The 27 formally evaluated programs typically reflected not only the effects of the work that went into each program but also the accumulated experience of up to seven years of development of similar programs.

Of the 1,693 students who used PLATO in the process of development of the programs, 753 were involved in 8 "Research" programs and 940 were involved in 19 "Teaching" programs. "Research" programs use PLATO more as a means of close experimental control than as an instructional medium. Most "Research" programs are used in psychological investigations in areas in which the relevant variables have already been reasonably well identified. "Teaching" programs have the instruction of students as their main objective. The relevant response variables in such an instructional situation have not been so extensively studied and thus the development of "Teaching" programs may involve as much research as the programs designated as "Research" programs. The difference in the researcher's confidence in the identification of relevant response variables between these two forms of programs can be seen by examining the types of data processing that were used.

Four basic types of response-data processing can be distinguished: (1) external - in which a separate program is used to sort and analyze data collected by the PLATO program, (2) internal - in which data are sorted and analyzed by the same PLATO program which collects them, (3) combined - in which a separate program is used to complete processing begun by the PLATO program, e.g., the PLATO program might collect and sort data into specific categories after which statistical analyses would be applied to the reduced data by use of separate analysis programs, and (4) none - in which the author decides not to collect student response data. Programs in which PLATO acts as a simulated laboratory, as for testing designs of electronic circuits or highway bridges, are typical of those for which data are seldom gathered.

		Program Type		
		Research	Teaching	Total
	external	2	7	9
Data Processing Type	internal	5	7	12
	combined	0	2	2
	none	1	3	4
	Total	8	19	27

Table 2.

Categories of response data processing in 27 PLATO programs.

Internal data processing makes the most efficient use of the computer capability of a CAI system. However, internal processing assumes that all relevant variables are known and appropriate methods of analysis have been selected. External processing, on the other hand, permits postponement of deciding which variables are relevant or what methods of analysis might be appropriate. Combined processing lies between these two extremes. It seems plausible that processing procedures might go through an evolutionary development as more became known about each particular type of program. External processing would be necessary in early exploratory studies, but as soon as relevant variables had been identified with some degree of confidence a shift to more efficient combined processing methods could be made. Finally, when all testing had been completed and the program was ready for extensive application a shift to internal processing could be made for greatest efficiency.

Table 2 supports belief in the possibility of such an evolutionary trend. Far more "Research" than "Teaching" programs use internal processing (62% vs. 37%), as might be expected when one considers that the

"Research" programs are generally highly controlled tests of rather specific hypotheses generated from extensive prior studies in a given area. The low representation of combined programs seems to be explained by the fact that programmers prefer to retain the flexibility of external processing until preliminary studies are completed and all of the processing can be done internally. It should be pointed out that it is rare to find a single program which demonstrates an evolutionary development in response-data processing such as that described above. The more usual situation is that an individual programmer introduces such changes as he gains experience in writing a series of programs. This is true especially for programmers working in a given subject area for a given student population.

A less generous interpretation of Table 2 might be that evaluation was an afterthought for many of the authors of "Teaching" programs.

Random Access Storage

During the Fall of 1967, a disk storage capability was added to the PLATO system. Rental for this equipment was funded through a grant to SIRA under this contract. Previous reports have detailed the advantages that random access memory of this size would give to the PLATO system. Among these advantages are increases in the size and number of programs that can simultaneously be made available to students on the system and a larger storage capability for individual student use. The latter enables the student to retain much more of the information which he gathers during inquiry and laboratory operations on PLATO in a form which will allow rapid utilization during other portions of the teaching program.

The effect on PLATO of the additional memory provided by the disk system was multiplied by the initiation of TUTOR, a new author language developed at CERL. This language is simple enough to allow authors with no prior experience with CAI to begin writing CAI lessons within a matter of hours after being introduced to the language. With the addition of the disk system, time-shared authoring was made possible. Thus up to 20 authors could simultaneously write lessons in the TUTOR language directly onto disk storage. These lessons were then immediately available for use by students. The impact of time-shared authoring (eliminating most of the intermediate steps of getting a lesson from rough-draft to operating stage) plus a language which could be learned in hours rather than months was overwhelming. An impression of the size of that effect can be gathered from the fact that over 100 lessons were produced within 6 months of the initiation of TUTOR and disk storage. This can be compared with about 300 lessons produced during the prior 7-year life of the PLATO system.

An additional effect was a change in the type of authors who were producing the new lessons. Prior to TUTOR, most lesson authors were members of special curriculum groups and/or were individuals with backgrounds in computer programming. It is now clear that these early authors had special quantitative interests and backgrounds in educational technology which are not shared by many of the newer authors.

Changes in the Author Population

Development of SIRA techniques was advanced along two lines; a theoretical approach (discussed below) based on predictions of the ultimate applications of CAI, and an applied approach based on the needs of current users of CAI.

In an attempt to gather information from the widest possible range of current users, SIRA often has supported development of lesson material in areas for which CAI seemed to meet a need. Appendixes III and IV describe developments of this type which were carried out by SIRA personnel. The lessons resulting from this support have been solidly developed and evaluated. These lessons continue to have use in college and high school level courses. Unfortunately, the contribution toward an understanding of the needs of lesson authors was not as beneficial. The SIRA personnel who developed these lessons were simply not typical of the general author population which is now using PLATO. Even the non-SIRA authors of the years before TUTOR were a specially selected group. The difficulties of working with an experimental system and languages which were only slightly (if at all) removed from computer languages led to a self-selection of authors. As was indicated above, these authors tended to have rather extensive qualifications in educational technology, mathematics, and the experimental sciences.

The evaluational needs and interests of this select group were generally much more complex than those of the present population of authors. Similarly, the results of SIRA's theoretical approaches appear to be far beyond the general interests of present authors.

Programs such as CLASIFY and MLR (see Appendix I) were developed by Tatsuoka and Kraatz to utilize the capabilities of the computer to recognize relationships among complex patterns of response which might otherwise escape the attention of an author. These programs were intended to be a major step toward fulfilling the predictions of some of the more visionary workers in CAI to the effect that the computer would eventually use the entire structure of the student's CAI behavior to evaluate and alter the presentation of lessons. While these two programs are ready and waiting, it appears that it will be several more years before authors will be at a stage where such programs can be realistically utilized. Only rarely do current authors use or see a need for use of all of the more obvious variables. In fact there is little present evidence to show a real need for such advanced approaches as made available by CLASIFY or MLR. Nevertheless, should such a need arise in future years, SIRA has provided the initial techniques.

In summary, early development of SIRA followed two paths. One path began at the expected limitations of CAI evaluation while the other path began at what was thought to be the current practical limitations. Within this developmental framework progress was gratifying and all major goals were attained. However, subsequent experience has shown that the beginning point for practical work was biased toward an author population which was more sophisticated in evaluation techniques than was the general author population. The resulting techniques were thus of more use to the individual with direct interests in evaluation than to the general author. The last few months of the SIRA project were spent in bridging this gap.

Influences of SIRA on Present PLATO Techniques

The effects of the SIRA project on day-to-day operations of the PLATO system are most visible in elements of the TUTOR language and in a disk data sorting routine (SORTER, see Appendix I) developed by Kraatz. While all have been only recently implemented it should be recognized that they rely heavily on the hard-earned experience of earlier SIRA developments. In addition, although the visible effects of SIRA are elements of a specific language and system, the status of the PLATO system as a model for the development of other CAI systems makes these effects rather far reaching.

The TUTOR language allows the author to prescribe mathematical operations contingent on specified student behavior. This provision permits the author to do "within subjects" data analysis during a lesson or as a final operation of a lesson. In addition to student responses, data may consist of such items as numerical codes set under specified contingencies or processed values of relative time (with a precision of 1/60th second). The results of these analyses may be stored on magnetic tape by means of a request in the TUTOR language. SORTER is then used to combine stored within-subjects data for a total analysis if desired. These options exist in addition to those used for lesson evaluation as outlined in Appendix IV.

The present TUTOR language is both extremely flexible and easy to learn. Both of these factors have contributed to its overwhelming success. It is extremely unlikely that authors would be willing to give up the flexibility of the language in return for standard student evaluation measures which were not under his control. Hence, each author must build such measures into his own program when desired. Lesson evaluation is relatively simpler. One technique of lesson evaluation which is based on an early version of TUTOR is described in Appendix IV. That technique is made even more effective by the improved information provided in present versions of TUTOR.

Conclusions

Reactions of Visitors to the SIRA Project

The work of the PLATO project attracts many visitors who are highly qualified in the fields of educational technology and evaluation. It was hoped that the SIRA project could benefit from the reactions and suggestions of these individuals. The relative novelty of PLATO and SIRA however defeated this hope. The typical visitor reacted with enthusiastic interest (cf. Davis, R. B. The Changing Curriculum: Mathematics, NEA, Washington, 1967) and extensive requests for information rather than suggestions for further improvements. It became clear that until an individual had time to become familiar with the features already available in SIRA there was little hope for concrete contributions. This same conclusion was reached during a SIRA Working Conference held in November 1966. Members of this conference included Dr. Veryl Schult and Mr. Joseph A. Murnin of OSOE. Along with detailed suggestions for improvements and strategies, it was generally suggested that more extensive working ties with prospective users of SIRA be made.

Lesson Production by SIRA

A series of SIRA seminars held from 1965 through the summer of 1967 gave a somewhat closer view of the problems of the user. Members of CIRCE (Center for Instructional Research and Curriculum Evaluation) such as Thomas Hastings, Robert Stake, T. McGuire, and Jason Millman met with SIRA personnel in a joint effort to produce evaluation techniques simultaneously with new curriculum materials.

Unfortunately the difficulty of writing programs prior to TUTOR and time-shared authoring severely restricted the sample size of users who could be observed by SIRA and CIRCE personnel. This factor led to lesson production by members of SIRA and CIRCE. The QED program designed by Robert Stake along with many of the lessons mentioned in Appendixes II through IV were products of SIRA, CIRCE or SIRA-supported workers. Unfortunately production of lesson material by evaluation personnel led to the unrealistic assumptions about users mentioned in an earlier section of this report. These assumptions have been cleared away now that disk operations have allowed a "real" population of authors to begin operating.

A second factor has been the rapid evolution of new levels of software (see Appendix I). Evolution is frequently so rapid in the experimental environment of CERL that new evaluation techniques have been left without a use as a result of sudden advances in the system. Thus a full range of techniques designed for use on old student data from PLATO was made superfluous by a shift to new data records produced by the TUTOR language. If these developments had been parallel, the work of SIRA might be considered as wasted, however, viewing the changes in user languages from the systems side of the picture it is clear that SIRA has stimulated a great deal of the change in data procedures. Data processing functions which were formerly considered ancillary to lesson presentation are now a part of the TUTOR language. Functions which were originally conceived of as necessary for data analysis (e.g., extremely detailed student information) have been dropped for lack of use. Thus, "survival of the fittest" of SIRA inspired functions is probably the only way the SIRA project can be evaluated as a whole.

Perhaps the major contribution of the SIRA project has been the fact that the data collection system which has evolved was begun from a point almost diametrically opposed to that of many CAI projects. The keeping of the fine-grain data required for programs such as SPECTRE has been generally rejected for practical reasons by designers of large CAI systems. PLATO retains the option to collect data of this precision though in practice today such data are rarely collected. The important point is that, through SIRA, programs which make a serious effort to completely <u>use</u> such data have been produced. Throughout the life of SIRA, there was no time at which the project was seriously limited by the PLATO system. Thus techniques which would be matters of the future for other CAI systems have already been developed at CERL. It is now clear that the complete utilization of student response data in CAI is limited by neither computer hardware nor software, but rather by the ingenuity of the individual instructor or lesson author.

SIRA Personnel

J. A. Easley, Jr.	Principal Investigator	1964-1968
R. A. Avner	Project Director	1967-1968
Charles Bridges	Research Assistant	1965-1966
J. Richard Dennis	Research Associate	1967-1968
Anabeth Dollins	Research Assistant	1967-1968
Mariellen Gilpin	Research Assistant	Summer 1967
William Golden	Research Associate	1967-1968
Bruce Hicks	Consultant	1967-1968
James Hicks	Research Assistant	Summer 1966
Elizabeth Kendzior	Research Assistant	Summer 1967
Scott Krueger	Computer Programmer II	Fall 1965
James Kraatz	Research Assistant	1966-1968
Jane Millar	Research Assistant	1966-1967
John Norton	Research Assistant	1966-1967
James Rubovits	Research Assistant	1966-1967
Kikumi Tatsuoka	Research Assistant	1966-1967
Brian Voth	Research Engineer	1966-1968
Beveraly Wilson	Research Assistant	1964-1966
Nitza Yarom	Research Assistant	1966-1968

Appendix I

A Glossary of SIRA Programs

The general term "program" actually includes at least three distinct levels of software. These levels are:

1. Systems level - e.g. programs such as CATO
2. Language and Logic level - e.g. programs such as TEXTTESTOR, TUTORIAL and TUTOR written in systems languages.
3. Lesson level - e.g. lesson material written for presentation via a language, logic, or (rarely) a systems routine.

The three levels are the result of an evolutionary process. Earliest lesson material and analysis routines were all written directly by use of systems level software. Thus production of nonstandard material in the early days of PLATO required highly specialized knowledge. Producers of software at each level must be able to use programs of at least the next higher level. Introduction of the language and logic level first made it possible for authors to produce lessons (e.g., via AUTHOR MODE of TEXTTESTOR, TUTORIAL or TUTOR) without the need to work at the systems level. SIRA followed the evolutionary trend by providing new programs to aid the user in fullest utilization of the lower level software. Survival of the functions of a SIRA technique during periods of rapid evaluation of programs at a new level is often the only form of evaluation possible.

Each of the program descriptions below indicates the level of possible users. Certain programs are also useful at higher levels. The list does not include options such as AUTHOR MODE which are tailored to each specific program in which they are used.

BXDOPE, XDECODE: These routines allow CATO student response data to be processed by use of FORTRAN. Student response data is recorded in 54 word binary records on magnetic tape. Several bytes of information are packed into each word. Processing of these tapes is somewhat difficult and cannot be accomplished with ordinary FORTRAN programming. BXDOPE is a computer subroutine which, when called, returns all the information associated with a single student input. Tape reading and buffering problems are handled by the subroutine. XDECODE unpacks the information returned by BXDOPE. (Kraatz, Level 2)

CLASIFY: Searches student response data for specific sequence types (not necessarily specific sequences). Relevant student responses are first separated by the investigator into various classes. Sequences of response classes are then specified. CLASIFY detects and makes available for further processing all such sequences occurring in the student response data. CLASIFY utilizes a nonparametric decision rule [see Tatsuoka, K. "A Multiple Separation Function for Pattern Classification", CSL Report R-321 (1966)] in detecting sequences, as opposed to the regression technique of MLR or a simple bit-by-bit comparison method. (Tatsuoka, Level 2)

CUMCURV: Prints a graph of total student responses per unit time for individual students. Up to 62 different event types may be specified. Time intervals may range from 1 to 9999 60th of a second. (Dollins, Level 3)

DODAD: Provides an easy method for insertion of diagnostic output statements in a PLATO teaching program. The inserted statements provide the programmer or author with a printed copy of events occurring within a program at any given time. Information thus provided is useful in locating logical errors in a program and in determining more efficient means of processing student response data. (Kraatz, Level 2)

MLR: A general sequence classification program based on the multiple linear regression model. MLR accepts as input student response data recorded in the standard format of the PLATO system (dope). MLR identifies and provides a sample listing of response sequences. A sequence is distinguished by the content of its initial element. The list of relevant initiating elements or classes of elements is a parameter supplied by the investigator. Sequences in the sample listing are classified by the investigator according to any rule of his choosing. Information available in the sample classifications is utilized to compute regression coefficients, from which a classification rule is formulated. Further student response data supplied to MLR will be organized into sequences and classified according to the rules thus formulated. The sequences so classified are made available for further processing or for use in modification of the classification rule. (Kraatz, Level 2)

MONSTER: a magnetic tape editing program which allows up to three editors to insert, delete or rearrange records or parts of records on a tape. All manipulations are done from PLATO terminals and effects are displayed on the screen. A standard character set may be output for PLATO programs. This program is used mainly by systems programmers rather than lesson authors. A complete description is contained in "Project SIRA Second Mid-Year Report September 1966, USOE Contract OE-6-10-184." (Bloome and Krueger, Levels 1 and 2)

NEWSORT: with each student input, the PLATO system automatically records a word containing fixed information relative to the input. A second word containing up to 10 bytes of variable information may be recorded relative to the state of the teaching program. Information contained in the two words constitutes a single response record relative to a given input. Using NEWSORT, values, or ranges of values of variables occurring in a response record may be specified from a PLATO terminal. Records conforming to the specifications are recovered from the student data and stored. The subsets so obtained may be counted, listed, (by elements or in their entirety) resorted, and refined according to new specifications. An early version of this program was called DOPESELECT. (Norton, Level 2)

NURD: A processor for student data arising from the program IMNURSE, a series of lessons in maternity nursing. Hard copy of all student responses, along with limited summary information is provided. (Kraatz Level 3)

SORTER: A general data processing routine for binary or BCD records stored on magnetic tape. Special provisions allow maximum ease in sorting data originating from TUTOR teaching language and GENERAL teaching logic. Full use is made of the disk system as an intermediate storage medium. Records may be sorted on up to six levels at a time. Ordering of the output of sorts may be on an alphabetic, numerical, or temporal basis. Output may be printed or displayed on the screen of a PLATO terminal. SORTER may be called from TUTOR author mode (and TUTOR may be called from SORTER) however computer storage limitations do no permit simultaneous use of SORTER and TUTOR. (Kraatz, Levels 2 and 3)

Appendix II

STUDENT REACTIONS TO EXTENDED SESSIONS OF COMPUTER-ASSISTED INSTRUCTION

The most usual reaction among students upon first exposure to CAI is an attitude of eager interest. However, we cannot assume that this same attitude will be retained for extended periods of exposure. Nitza Yarom investigated reactions of students from two distinct populations to such extended exposure. The first section of this appendix will deal with her observations made on freshman high school students with a predominantly upper middle-class background. The second section will deal with her observations made on junior and senior level high school students from culturally limited backgrounds. In both cases any conclusions are to be considered tentative and are intended only to serve as a basis for further investigation of the points noted. The third section will describe some general conslusions.

General

Computer-assisted instruction seems to enable school children to learn more nearly according to their individual capacity than is possible in regular classroom instruction. While watching children during computer-assisted instruction, it is commonplace to see different children in different stages of the problem at the same time. Some children are quicker in catching onto the principles of the problems given; their progress is rapid compared to others who need more exposure to the initial steps of the problem and more trials to get the right answer. Each child is individually responsible for coping with the problem, and any difficulty he may encounter can be detected and recorded by the instructional system. When observing classes in the process of regular instruction, it is much more difficult to detect those students who have specific problems. Usually those students who are more intelligent and quicker participate the most in class. Two or three distinguished students are the ones who give the answers and the teacher may have only a vague notion of how much understanding the others have.

Classes that received more preparation, even though they might be less capable than other classes who received less preparation, functioned better in solving problems given by the computer. They functioned better in the sense that the mechnaical problems were overcome in the first few minutes, and the rest of the time could be devoted fully to the instructional problem. The percentage of right answers or answers that showed evidence of learning were significantly higher in classes with better preparation. The best specific type of preparation for CAI has not yet been determined, but it is already clear that, no matter what the method is, it should be preventive and cannot be completely dependent upon intuition or "trial and error" attempts of students. It is yet to be found, for example, whether a theoretical or a practical preparation is more effective or whether using prepared forms (as instructions while working on the computer) helps significantly or not.

It seems beneficial to introduce the students gradually to the machine, letting them cope first with the mechanical problems. Only when these mechanical problems cease to interfere with the process of understanding and solving the instructional problems can true instructional research begin.

I. Reactions of Middle Class Students to PLATO

After overcoming the mechanical problems and the initial excitement, the students, no matter what their age was, became very much involved in the problems presented. Sometimes they did not notice for a relatively long time (3 minutes) that somebody stood behind them while they worked. Sometimes their behavior involved talking to themselves, especially among students who worked alone.

In comparison with the classroom situation, it seemed that the intensity of attention was greater - perhaps because there was personal responsibility for solving the problem, or because there was frequent direct and personal contributions to its solution.

It would be interesting to find a theoretical explanation for this phenomenon. Using common sense, it is suggested that it may be due to the fact that the student is the initiator and the main contributor. This allows him a great degree of independence which means less dependence on others for help and hence more concentration.

In the beginning of observations the impression was gained that boys were more involved and interested in the process of learning than girls and tended less toward social interaction which was not associated with the problem-solving task itself.

Now the impression is that this difference is more manifest in younger children. Girls from 11 to 13 years old tend to be more overtly alert to social events and to stop their work whenever another girl makes a remark or several girls gather to talk. But girls from 14 to 15 years old tend to behave like this much less. In fact, there is no observable difference in the degree of interest and involvement between older girls and boys.

This difference in behavior was also less likely to occur when the work was done in pairs. Perhaps the more sociable and attention-demanding girls were those who preferred working in pairs (creating in this setting a group that was sufficient for their social needs) while the more quiet and interested girls preferred to work alone.

It seems that working in pairs had, somehow, an advantage over working alone. The first advantage was for the process of problem-solving itself. Two students provided each other with ideas. Thus, there were more ideas to be tried for problem solution. In some cases, one of the pair was more expert, having better understanding and more knowledge than the other. This student could show the other how to deal with the problem, with the result that both gained a fuller understanding. There was also an economic aspect of working in pairs. One student usually took charge of the clerical work, i.e., writing down numbers, reading the data, and keeping the papers in order. The other student punched the buttons, tried to figure out what to do next, etc. While working together there was a great deal of verbalization directed toward the clarification of ideas. Socially, it seemed that most students enjoyed this situation very much.

As for students who worked alone or preferred to work alone - their acheivement did not appear in any way to be slower or worse than those who worked in pairs. It is true that they moved from one operation to the other more slowly than the pairs, but, as there was some amount of time wasted by the pairs on games, laughing, etc., there was some compensation. Moreover, it did not seem that people who work alone experience loneliness. There were immediate appeals to neighbors, people looked around for advice, shared experiences, etc. It seems that they always had their friends around and it rarely reduced to a situation in which they felt the presence of only themselves and the computer.

II. Reaction of Culturally Limited Students to PLATO Inquiry Instructional Programs

Some 60 economically disadvantaged students from the University of Illinois' 1967 Upward Bound summer program were invited to try several inquiry-type PLATO programs as a supplement to their mathematics program. Most of the students were high school juniors with a few seniors included.

These students represented an extreme in background which, with one exception, had not been studied on the PLATO system up to now. Prior subjects were generally highly selected University High School students or college students. However, despite their generally poorer academic and socioeconomic background, the Upward Bound students (UBS) appear to adapt more readily to the mechanical aspects of working with the student station equipment. This observation should be tempered with the fact that the UBS were in smaller classes (8-10 rather than 20), were generally more focussed on theoretical problems to be solved, and received more individual attention. It was clear that the degree of individual attention given the UBS reduces the effectiveness of PLATO as a means of reducing teacher load. More efficient introductory material should eliminate this problem. They clearly provided a critical test of the PLATO programs used.

Students were divided into two groups based on their math background. Those with the better background appeared slightly more capable and more interested in computer instruction as evidenced by their problem-solving behavior on computer programs and their better attendance record at the voluntary sessions.

Introductory lectures demonstrating the use of student station equipment was found to be helpful on some of the mechanical problems but insufficient to enable them to meet the demands of the programs. Direct exposure to the equipment with programmed introductory material is recommended as the best approach for new students.

Reactions to Specific Programs

The two-parent genetic simulation program (GENO) - Both data analysis and direct observation of UBS showed a rather mechanical and poorly structured approach to the problems with little evidence of actual learning. The same observations were made on a slightly more complicated population genetics program (GENO-POP).

The geometry figure-constructing program-here also the UBS showed poor performance despite evidence of interest in certain features of the program. A general observation (which holds for other programs as well) was that instructions for the program (1) assumed too much prior knowledge and (2) were not developed in simple stages, thus appearing to be so overwhelming to the students that they were simply avoided. Students tended to rely on directions from instructors.

Arithmetic problem - Students in the one group exposed to this drill-type program reacted very favorably to its rapid reinforcement and simple structure. This is the one PLATO program that was developed with under-achievers in mind.

NINSECT - Only two students were given this rather complex program. They reacted favorably but required a great deal of personal attention from instructors.

PROOF - Merely as an exploration, this program was tried with three students who found it excessively difficult - though possible - to master. The point of constructing proofs of single theorems in algebra seemed to escape them.

Possibly simpler programs should have been given first. This would have increased morale a bit, but probably would not have reduced the tendency to respond in the mechanical fashion on the more complex programs. These students may be reinforced by things (like having slides change or figures plotted) which are not the things the program is intended to reinforce (solving a problem).

III. Interaction of Motivational and Intellectual Factors in CAI

Extensive observation of student behavior during PLATO classes has led to the conclusion that major consideration should be given to distinguishing between intellectual and motivational factors in CAI programs. Students may, for example, be equivalent in intellectual ability as evidenced by their progress in early segments of lessons. Once the novelty of CAI wears off, however, motivational differences begin to appear.

These motivational differences are evident in groups which differ in age, sex, and socioeconomic background. Two variables affecting motivation may be distinguished; (1) quantity and (2) type of reinforcement. Reinforcement may be "social," giving direct recognition to the student as an individual (e.g., "That's very good, John"), or "factual," giving information in terms of task content, (e.g., "8 + 6 = 14"). The quantity of such reinforcement can range from zero in a reading assignment which is not discussed to 100% in a programmed lesson in which every student response is reinforced by appropriate comments. The most important observation is that too much reinforcement (or reinforcement of the wrong type) appears to be as discouraging to some students as the complete absence of reinforcement.

Within the three group variables of age, sex, and socioeconomic background, the relationships between quantity and type of reinforcement leading to optimal motivation showed the following trends:

(a) Age Quantity - decreases with increased age
Type - shifts from social to factual with increased age

(b) Sex Quantity - young females require more than males
Type - females oriented more toward social

(c) S-E Background
Quantity - lower classes require more than middle
Type - lower classes respond more to social

Appendix III

CURRICULUM DEVELOPMENT BASED ON STUDENT RESPONSES

by

Elizabeth Kendzior

Introduction

The style of curriculum development under discussion can best be described as a cooperative venture involving students, teacher and computer programmer. The general procedure is quite simple - each brings his own interests and talents to the task of developing methods and materials useful to the teaching-learning process. Its success depends upon the availability and quality of communication among the participants. What is unique about this style of curriculum development is the immediacy of feedback and the active participation of students in developing their own curriculum.

Procedure

The more specific procedure is a cyclical one which, for purposes of discussion, can be divided into four phases. No sharp lines of distinction can be found in practice.

Phase I. Conventional-classroom Exploration and Identification of Problems

The term "conventional-classroom" is used here to differentiate between the typical classroom activities and those directly involving the PLATO computer. It does not signify any particular teaching style or method, but it is an integral part of this style of curriculum research.

Rather than having teachers and curriculum specialists design a bit of curriculum which subsequently is tested with students, here the approach is first to identify problem areas as they develop in an on-going class and then to actively solicit the help of the students in resolving these problems.

Phase II. Conventional-classroom Planning

Once some problems or problem areas have been identified, the students and teacher work together at clarifying and/or resolving these problems. When the proposed solutions are not practical in terms of conventional classroom capabilities, the possibilities of computer assistance are considered and specialists in computer programming are consulted.

At this point, the teacher assumes the role of coordinator of communications for she is in the best position to oversee the entire operation. In addition to her training and experience in teaching and the subject matter, only minimal training in the uses of the computer is necessary for effective articulation of all interests.

Phase III. Computer Programming

Primary responsibility for implementation of the pooled ideas is now placed in the hands of a specialist in the field of computer programming. However, the teacher and, through her, the students continue to serve as actively participating consultants. A high rate of exchange of knowledge and ideas is characteristic of this phase, too.

Phase IV. PLATO-classroom Implementation

Once the computer program is in workable form, it is presented to the students. These students may be the same students whose ideas went into the planning of the program and/or they may be students who did not participate in its development.

Whichever the case, the PLATO-classroom experiences are recorded in a variety of ways. In addition to recording the teacher's impressions, a psychologist and other "outside" observers are on hand to register their impressions. Records of student-computer communications are also available by means of the computer service programs NEWSORT and SPECTRE. Most important for this style of research, direct evaluation by the students follows each PLATO-classroom session.

These diverse sources and types of evaluation typically result in a host of suggestions for improving the existing program and initiating new ones, but the procedure does not end here. While the computer programmer is making the suggested improvements in the first program, the teacher and students not only remain in contact, ready to give their second impressions, but also continue exploration and identification of new problems in the course of their normal activities. Thus another cycle begins.

A roughly chronological account of the development of a portion of a subcourse in genetics and evolution will serve to illustrate this procedure. Two sections of the Science I course at the University High School of the University of Illinois participated in this project. Both covered essentially the same material in the long run, but the sequencing and time allotment of individual sections of the unit varied because of scheduling problems and the desire to space the work so that a follow-up group was available. For the most part, Group I took the lead and Group II served in a follow-up capacity. Particular attention to these differences in treatment will be made only when they seem to have a significant bearing on the results of this research. The class time involved in this project was approximately five hours per week for one quarter of the school year. Of this time, the students spent from 5-10 hours actually working with the computer.

Group I began this unit by working with some curriculum materials developed by the School Science Curriculum Project. Their NINSECT GAME was modified into NINSECT — A MODEL OF INFORMATION TRANSMISSION at the discretion of the teacher. The basic concepts and procedures remained the same, but the approach toward use of the materials was geared to the age level and science background of the current students.

In brief, NINSECT introduces the students to such fundamental concepts of genetics as the gene, chromosome, sex-linkage, dominance-recessiveness, segregation and independent assortment [although none are labeled as such]. This is done through the construction of hypothetical insects, called ninsects, from cardboard pieces representing body parts. Choice of body parts is decreed by bits of information coded as punches on IBM cards which have been prepared to randomly present all possible combinations of traits. Once the student has mastered the game, analogies are made to the real world of genetics.

The ages of the present students and their previous experience with model-building in science seemed to dictate the modification of NINSECT from a game to a mechanical model of information transmission. Furthermore, this modification seemed to add better long-range utility to NINSECT as a legitimate aid to understanding the processes of science and of genetics in particular.

Although NINSECT served as an excellent introduction to genetics, some problems did arise. For one, the physical limitations of the hardware involved restricted the number of matings and offspring that could be dealt with at any one time. This situation led to the more serious problem of adequately representing the random nature of gamete formation and combination, for in the early versions of NINSECT only the four possible offspring phenotypes were constructed. Many decisions made for the sake of simplicity (e.g., representation of the ten NINSECT traits on one set of chromosomes) would have to be followed up if misrepresentation of the complexities of genetics were to be avoided. But since such complexities were beyond the intended scope of NINSECT, the need for additional methods and materials was obvious.

Two separate approaches for resolving these difficulties resulted. One effort aimed toward preparing NINSECT for manipulation with the aid of the computer. However, this clearly would require a considerable amount of time. The other approach involved the computerized simulation of two-parent matings and seemed to offer more immediate promise.

Computer Manipulations of NINSECT

Conventional-classroom use of NINSECT recommended it as an excellent means for arousing the students' interest in genetics and at the same time providing them with much needed concrete representations of abstract concepts. But it also had its limitations, so an ideal solution seemed to include making provision for both physical manipulation (available through NINSECT proper) and greater flexibility (available through use of the computer). The students themselves commented that they liked the idea of NINSECT but after a while found it too rigid and devoid of challenge.

In particular, they requested some means of simultaneously preserving both the parents and offspring. Moreover, they wanted to be able to go on and "construct" additional generations. With NINSECT proper, these feats were possible but not feasible because of the hardware and space required. The computer could provide a more practical solution.

The students also very quickly caught on to the fact that only four different offspring phenotypes resulted, but for many "four" had now become an almost magical number. These might be overheard saying, "but there have to be four," when in fact perhaps only two different phenotypes were possible for that set of parents. For those students who came to understand the chance element in the formation of any one offspring, disenchantment with NINSECT set in. These tried to overcome the problem by shuffling the cards, but this too was awkward. Random production of offspring became a goal of the computer program.

A similar problem developed when the students were asked to predict the parent genotypes from the offspring phenotypes. A few industrious students set out to construct all possibilities but this project became unwieldly and they gave up. This indicated a need for easy mobility from the genotype information to the phenotypes and vice versa in the computer program.

In addition, the teacher's task of judging the students' work was time-consuming and many students were held back while waiting for the teacher's assistance. A tutorial type program in which the students could pace themselves and receive individual attention was suggested.

Finally, after several discussion periods during which specific procedures to overcome these difficulties and to implement these ideas were outlined, the computer programmer set out to work on the assignment which developed. A preliminary computer version of NINSECT has been completed but to date has not been presented to students.

GENO — Two-parent Simulation

In addition to previously mentioned problems, conventional-classroom experiences with NINSECT also reveal the students' growing interest in more general applications of the genetics principles introduced. This reaction, coupled with the desire to deal with large numbers of randomly generated offspring, led to the specific recommendations employed in the GENO program.

For one, it was suggested that the entire alphabet be available for use in representing genes so that the students could make their own designations, especially since arguments typically arose over the choice of symbols. It was hoped that this simple provision would appeal to the students as well as assist them in understanding that such symbols are just that — symbols — and that they are arbitrarily chosen.

It was also suggested that confusion over the role of chance exhibited in the work with NINSECT could be overcome by illustrating the relationship between individual events and large numbers of random events by providing options for selection of individual offspring as well as bulk numbers of offspring.

Further recommendations called for the addition of an arithmetic section so that the more meaningful relative frequencies of genotypes and phenotypes could be readily obtained from the tallies provided. This provision was a direct result of annoying delays created by the necessary but simple and time-consuming calculations required in this type of work.

Again, after more specific discussions relating to the actual programming of these ideas, another assignment was given to the programming specialist. Within a relatively short period of time, GENO was ready for presentation to the students.

GENO — PLATO-classroom Implementation

After a brief introduction to the mechanics of the computer (keyboard layout, etc.) during the concentional-class period, Group I was taken to the computer-classroom. To give direction to their activity, a NINSECT-related problem in game format was presented to the students. This first encounter with the computer was disappointing from the students' point of view, but produced a wealth of valuable information for our research.

Direct student evaluation is well illustrated by some of the comments the students made upon completion of this particular PLATO-classroom session:

> "Bit of a waste...all computer did was calculate possible offspring."
>
> "All were a bit rushed...at least I was rushed...should learn about the machine one day and then do problems another day."
>
> "Difficulties were mainly with the computer and not the problems." [All but one student cited this as the major problem.]
>
> "Wanted to calculate offspring but nothing happened." [Failed to press correct key in most cases.]
>
> "Machine screen fades out after a few seconds and makes reading difficult. One machine didn't work at all."
>
> "Seemed to be errors in the computer...tried a problem and got an impossible answer. Once called for 30 offspring and got 20 and 20."
>
> "There was no way to get back to the beginning from arithmetic." [There is a way but it obviously was not clearly explained.]

"No erase button. Didn't know we had to wait for 30 seconds before pressing another button."
[When first choice did not appear immediately, students pressed the same key again. This offspring was recorded by the computer but not considered by the student.]

The teacher's impressions concurred with the students' statements. The mechanics of the situation overshadowed everything else. The record sheet for the assigned problem turned out to be poorly designed. The time delay between the prep session and the actual session made the former almost totally ineffective. The excitement always present when students are taken from their regular school building to another building was another distracting element. And, of course, the shortcomings in the computer program itself only added to the confusion. But in spite of everything, no one was particularly discouraged. The aura of excitement that usually surrounds young people when mention of computers is made had been reduced a little, but the students in general reacted as members of a research organization and seemed to take their disappointment in stride. General impressions of students' reactions to computer-assisted education made by other observers are presented in Appendix II.

A more precise analysis of the session, which is most valuable to the programming specialist, is available through the computer service programs NEWSORT and SPECTRE.

These programs help pinpoint trouble spots of which even the students may not be aware. For example, in addition to coorborating most of the criticisms offered by the students, dope analysis of this session revealed that many students were failing to follow a typed-number request for offspring with the necessary "period" and thus were forced to begin their problem over and over again; some never did find their way out of this bindup in the course of the session. Others were unsuccessful in adding more offspring because they repeatedly failed to press the "plus" key at the appropriate time. And on one occasion, a "hang-up" (an error causing the computer to automatically cease operation) was traced to the absence of a program check for numbers too large for the computer to accommodate.

When all these diverse forms of analysis were put together, another assignment for the programming specialist resulted. In the meantime, Group I returned to the conventional classroom — but within a couple of weeks many of the suggested improvements in the program had been made and Group II was readying itself for its first visit to the computer classroom.

Group II began this unit with a more direct approach to the topic of genetics. They, too, dealt with NINSECT (non-computer version) but then proceeded to a more thorough discussion of patterns of family inheritance and the nature of the hereditary material before approaching the GENO program.

Having learned from Group I's experiences, careful consideration and much time was devoted to introducing the students to the computer.

One conventional-class period was spent in a discussion of the general purposes and capabilities of the computer; special attention was given to its usefulness in regard to the current problem in genetics and to allaying the students' fears of damaging the computer. Once again a specific type of problem was assigned to direct their activities, but the first PLATO-classroom session was devoted to "getting acquainted" with the computer.

This procedure paid dividends because both the first and second sessions in the PLATO classroom proceeded relatively smoothly, despite a temporary equipment failure. Student reactions following the sessions paralleled those of Group I but the criticisms were much milder in comparison. They found their expectations of the computer's capabilities were exaggerated, in spite of the briefing session, and they reported those difficulties in the program which Group I identified but which had not yet been corrected. The only other noteworthy difference between the groups was some hesitancy on the part of Group II to accept the results of the computer. Several followup conventional-classroom sessions involving the tossing of coins corroborated the computer results and seemed to set their minds at ease. This reaction may be related to a tendency for both groups to regard the computer more as a novelty than as a real aid to education during their early experiences with it.

Subsequent considerations of the experiences with these groups led to a separation of the arithmetic program from the GENO program proper so that ARITH could be used in combination with a variety of other programs being developed. They also produced the recommendation that a program enabling students to graph their results on the screen as they proceed be developed. This proposal stemmed from the observed advantage of letting the student get an immediate visual image of the data generated. GRAPH, as this program is called, was the result of this suggestion.

Upon returning to the conventional classroom, Group I turned its attention to the topics of mitosis and meiosis. After struggling to make sense of microscopic slides and diagrams illustrating these processes, a film on mitosis was presented. Student response to this film gave rise to another opportunity for computer assistance.

During the showing of this film, many students came close to outright slumber, but their awakening was quite audible when time-lapse photography of mitosis appeared on the screen near the end of the film. Response was so favorable that this section of the film was shown twice, despite the fact that the film itself included several repeats of the mitotic sequence at various speeds. Had time not run out, the students would have welcomed even more reruns because the teacher could not respond to everyone's questions in time with the film. In this situation a picture, that is, a moving picture, literally was worth more than a thousand words.

As a result of this experience, a computer program called MOVIE is under development. At present, only preparation of the slides themselves is needed to complete the program. A basic format which permits each student to control the direction and speed of whichever slides are inserted has already been tested and proved workable. What remains now is to see whether MOVIE can capture and capitalize upon the rare type of excitement generated by the mitosis film.

Somewhat later, student inquiries during work with the A-B-O blood group provided a natural transition from family genetics to population genetics and launched the most active period of student involvement in curriculum development experienced in this project. The specific qurstion the students raised at this time was, "How can the frequency of type O blood in the population be so large (45%) while that of type B (10%) is so small, considering that type O is produced by a double recessive?" With this question and the lively discussion it evoked, the students identified their own problem and included in their diagnosis the claim that the model developed for family data did not "seem to fit" population data.

The task the students set for themselves was to build a model of population genetics which could explain a change in genotype and phenotype proportions over time. At the suggestion of the teacher, the first attempt at building a mechanical model of population inheritance employed the flipping and shuffling of coins to represent the behavior of gametes in a given population, but this model was criticized by the students. For one, there was both confusion and skepticism over the restriction that each generation begin with the same actual number of individuals; they argued that the size of the population was significant. As a result, use of relative frequencies of genotypes and phenotypes was recommended. But more importantly, with their repeated use of the phrase "in reality" they indicated a growing credibility gap between a coin-tossing model and the real world. As a partial solution, the teacher pictorially introduced a situation actually occurring in nature, industrial melanism in moths, but did so in an open-ended fashion which left the challenge of explanation to the students.

Turning to a real life problem seemed to make a big difference to the students because they eagerly accepted this challenge. Although only provided with photographs of the moths in their natural surroundings and the fact that both tree color and moth frequencies had changed over time, they readily identified and classified the many factors which might have played an important role in the change of phenotype frequencies; these ranged over what the scientist would call natural selection, differential migration and mutation. They also readily, but reluctantly, came to realize that considering all of these factors at once would make theirs an extremely difficult task, so for the time being, attention was focused on building a model which incorporated one major environmental factor, predation rate, and the model of family inheritance previously developed.

Even within these boundaries, many opposing proposals emerged. Although controversy arose regarding best estimates of the predation rate, the most lively debates centered on the gene combinations responsible for the phenotypes under consideration. One student argued that the observed changes in phenotype frequency for the moths could only have occurred over the given time period if the gene for peppered coloration was dominant over the gene for melanic coloration; another argued that melanic would have to be dominant over peppered, and still another put in a proposal for codominance. The obvious solution was to consider all three hypotheses and let simplicity and predictive power determine which would be more tenable.

In designing a mechanical model to simulate natural conditions, many smaller controversies developed. It was agreed that relative frequencies rather than actual numbers would be used to represent the distribution of phenotypes and that one roll of two dice (judged on an odd-even basis) by each student/moth would serve as a means for generating parental gametes. However, the problem of simulating random matings was never resolved to everyone's satisfaction. All suggestions for randomizing this procedure (pulling names of students out of a hat, etc.) seemed too tedious, so eventually this was accomplished by having each student/moth "randomly" pick another student/moth partner.

The number of offspring generated per couple became another controversial aspect of the developing mechanical model. At first it was suggested that each student/moth pair generate as many as possible within a given time period. This idea was later discarded because some students effectively argued that what would be represented would be the students' own mechanical skills rather than the natural situation. There were no birth rate estimates for the natural situation and no legitimate reason existed to suspect differential birth rates with respect to the genotypes. An average number of offspring per family was accepted as perhaps a better alternative; this number was arbitrarily set at six. Some students strongly criticized such a small number but reluctantly conceded on the grounds that simulating larger numbers would probably require weeks of class time.

To simulate the last step, determining predation survivors, the device of throwing dice again was used because it better represented the probabilistic nature of any individual organism's struggle for existence than did the earlier notion of eliminating the exact number of offspring corresponding to the best available estimate of survival rates for the given phenotypes.

The only remaining complication which set in before the mechanical model could really be put to use was converting actual counts of survivors to relative frequencies prior to their serving as the parents of the next generation. Because no way could be found to accommodate fractions of a percent in such a student-dependent mechanism, they were forced to round off the number of surviving offspring to the nearest five percent. This practice came under heavy criticism and later was repeatedly cited as a major flaw in the model.

These difficulties notwithstanding, the mechanics were worked out in such a way as to make them readily adaptable to the various hypotheses proposed. After a few practice funs, the students easily grasped the mechanics of the model and the entire simulation proceeded relatively smoothly. It took approximately one class period to generate seven generations, and in all, about four class periods were devoted to producing data for a melanic-dominant model and a peppered-dominant model. [The codominance model was tabled for the time being because the results of the simulation efforts to date had already provided more to do and to consider than could be handled during class time.]

Fortunately, from one point of view at least, it was just at this time that school closed for a two-week vacation period. During this break, the programming specialist went to work at what amounted to programming the model developed in the conventional classroom. Having been in communication all along, little additional planning was needed. The students' criticisms regarding accurate simulation of random processes, the desirability of working with larger numbers, and the importance of preserving all data had already been taken into consideration and could be resolved with relative ease with the aid of the computer.

During the interim, the program GENO-POP was prepared. It was geared to deal with two genes (alleles) and predation rate and permits work with population sizes up to 100.

GENO-POP — PLATO Classroom Implementation

Upon the students' return from vacation, only the announcement that their ideas had been programmed for the computer was needed to renew interest in the genetics problem at hand. However, what seems important here is not the ultimate fate of the respective hypotheses but rather the students' reactions to the computerized version of their model.

Probably because it very closely resembled their mechanical model, the students had little trouble in manipulating the program. They seemed to understand what they were doing and, for the first time, expressed an appreciation for the assistance of the computer. With the computer, they were able to cut the generation time for offspring by at least half; some of the more adept students were able to obtain in one PLATO-classroom session an amount of data which required 3-4 conventional-classroom sessions. Moreover, each student was free to work on his own hypothesis and to manipulate the predation rates accordingly. Two PLATO-classroom sessions were devoted to exploration of the industrial melanism problem; this was followed by several conventional-classroom sessions devoted to the discussion and interpretation of the computer-generated data.

An important by-product of these discussions was a vigorously renewed attack of their population model, and with it the inadequacies of the computer program. What highlighted this reaction, however, was the students' apparent change in attitude toward these shortcomings. They were no less happy about the uncertainties that remained after this investigation as opposed to others conducted during the year, but rather than going off on an emotional tangent so characteristic of their earlier reactions, they expressed their concern with constructive criticisms. They seemed to have gained a real understanding of the genetics principles involved and, in addition, the complexities of dealing with these principles both in the natural setting and in the teaching-learning setting.

The best concrete evidence of this reaction came from the students themselves. In addition to reconsidering their own conceptual model of population genetics, they enthusiastically responded to an invitation to make recommendations for improving the computerized simulation as well. Specifically, they were asked if — taking into account the limitations of the computer — they could suggest ways to make use of the existing program in dealing with the additional factors they identified as important in population studies and/or could make realistic requests of the programmer to achieve this end. They took this invitation seriously, even though no formal school requirements were attached, and working individually or in small groups produced several different plans for approaching genetics problems with the assistance of the computer. Indeed, the teacher had very little to do with these and the students were completely on their own for the PLATO-classroom tryouts.

One approach several students took was to employ a factor other than predation rate. Differential migration was a popular choice, but the procedures for dealing with this problem were quite varied. Some students simply substituted emigration rate for predation rate. It is interesting to note there that at first most of these did not appear to recognize that their conceptual change in the problem was not "sensed" as such by the computer; this insight came later and proved helpful in understanding that several factors could produce the same effect in natural populations, too.

Others studied the effect of removing a constant number of individuals from each generation rather than a proportional number from each genotype or phenotype; this procedure required that some of the calculations be done "by hand". A few students combined predation or emigration influences with variable population sizes thus exploring the phenomenon of genetic drift. Still others worked on the effect of emigration before versus after reproduction.

Two especially enterprising boys developed a fairly intricate procedure for studying the effects of immigration upon a given population. They first assumed that a constant number of individuals from Population B would be permitted into Population A; then while one student generated Population A data at his station, the other generated Population B data at another station to determine the genetic makeup of the immigrants. Finally, they got together and "by hand" adjusted the population data accordingly before going on to the next generation.

Notable designs and even more tedious "by hand" manipulation were employed by three boys who tried to include mutation rates in their population studies and by one girl who wanted to investigate non-random mating. In all of these cases, the computer was used to produce the base population but then the bulk of the work had to be done without the computer. In one of the mutation problems the student assumed that the mutation rate of B $\rightarrow$ A was .01. For each genotype he first had to calculate the change in type and number, then adjust the totals accordingly, and finally convert these to percentages in keeping with the 100 limit set by the computer program. Having done this, he would feed the information into the computer, produce another generation, and run through these calculations again. For the study of non-random mating, a similar procedure emerged. Starting with the assumption that homozygous-recessives had a measurable tendency to mate only with themselves, the student generated a base population. She then separately computed "by hand" the mating products of both segments of the population, pooled the information, converted to percentages and once again fed parental data into the computer to produce yet another generation.

With a reporting and discussion of these studies when back in the conventional classroom, the students' active participation in this research project temporarily came to an end. At this time, one student very well summarized events with the suggestion that we try to simultaneously incorporate all of the separate factors investigated into one population study. All agreed, however, that despire the groundwork laid this would still be a tremendous task and as such might better be handled as an individual rather than class project. The teacher, too, felt it was time to turn to other problems in science, but before doing so she questioned them about their over-all reaction to participation in this kind of a research venture. The result — a majority of the class said they would be interested in and available for future work on this project, even though they will have completed the course and will have to give of their free time.

Guided by the students' reactions and comments, work on this project is continuing. Their proposals are presently being incorporated into a computer program to be called ECO-SIM. With this program, one will be able to simulate numerous ecological parameters simulatneously. The students' interests and needs in talking to each other in the course of their work on the computer has also led to a modification of a program, called TALK, which permits visual intercommunication via the computer terminals.

Appendix IV

REVISION OF COURSE CONTENT BASED ON STUDENT RESPONSE ANALYSIS

by

J. Richard Dennis

A geometry teaching project carried out by the author from November, 1967, to April, 1968, provides a specific example of the use of computer-collected data to aid in the task of revising subject matter. The purpose of this project was to develop a set of lessons for junior high school students which would allow them to use the symmetry properties of triangles and quadrilaterals to explore the standard Euclidean properties of various figures.

This project was one of the first at PLATO to use the disk system during student sessions. The disk system was used to hold student restart information from one session to the next. This feature, together with the multiple lesson capacity of the teaching logic used for this project, made it possible for each student to be actively engaged with the subject matter during each class session. Even with only the 15 lessons used in this study, by the time the fastest student in each group finished, the slowest student still had at least five lessons to cover. Yet it was never necessary for a student who finished a lesson early to stop and wait for the slower students to catch up. It was also never necessary to schedule the entire system for make-up work for a student who was absent. When he returned, the lesson he was working on previously was put in memory and he continued from there. These features greatly reduced the amount of the instructor's time needed for routine matters of running the class.

The approach to topics in geometry used in this study is new to the curriculum of the United States. In developing new presentations of subject matter trial materials which are intended to convey certain information are prepared and presented to students. As the presentation progresses, the reactions of the students are observed. The students are also asked questions which can be answered only if the material has done its job. Based on the students' responses to these questions the curriculum developer gets certain hunches. These fall generally into two categories: 1. things are going well, and 2. things are not going well. If the latter is the case, the author looks specifically at the questions the students were unable to answer and the incorrect responses which were given. This information suggests revisions which can be made to overcome the difficulties that the students encountered. These changes are made and the process is repeated until the students exhibit the desired type of interactions with the chosen topics.

The procedure used in this study was to present the subject matter to five successive groups of students. Each group was small enough

(5-7 students) for the author to observe the students as they executed the lessons. Questions were asked mainly by the computer, but if a student encountered trouble and asked for help, other questions were asked by the instructor in an effort to ascertain the sources of the difficulty. The computer kept a record of what each student did in executing the program. If such record keeping is carefully designed by the experimenter he can, in many cases, guess the thoughts of the student with a fair degree of accuracy. Questions asked by the instructor help further to sharpen these guesses. To keep records on the questions and responses made in person, the subject matter program had a unit (frame) accessible to both instructor and student in which comments and notes could be recorded. As the instructor asked questions of the student he could thus type notes that were recorded by the computer. Later, when these records were sorted according to lesson and student, these additional questions and comments appeared together with the string of incorrect responses given to the computer's questions.

Another advantage of computer control in projects of this type is the frequency of possible revisions. For example, as students execute lessons, if they are allowed to go at their own rate it is not very long until there is at least one lesson between the fastest and the slowest. This means that within any one trial, one can make and evaluate the results of revisions several times. Particular attention is paid to the problems of the more rapid students. Revisions in the lesson are made before the next class session in which that lesson will be needed by slower students, and particular attention is then paid to the slower students as they execute that same section of the program. This procedure allows several attempts at unusually difficult problems within a given group of students, as well as between one group and the next.

It is the process of identifying necessary revisions that we are interested in here. First, however, some of the author's beliefs about student responses should be clarified. No attempt was made to eliminate or minimize individual incorrect responses. There are several reasons for this, the most important of which is the large amount of time required to achieve such a goal. Also, it is the author's personal belief that in order to write error-free programmed instruction material, the difficulty of the tasks required of the student must be reduced to a level that prevents him from making judgments about the concepts involved. Also, for this specific application the author intended to make rather extensive use of planned errors. For example, the author, in questioning students about properties of, say, a parallelogram, might ask:

"Could a parallelogram have congruent diagonals?"

Inherently, this is a self-correcting item since there are exactly two responses, "Yes" or "No". For those students who answer incorrectly, an opportunity to present them with more information is missed if the reply to the incorrect response is just a statement that it is incorrect. Instead, the author presented the students with a counterexample and a statement such as:

"Sorry, here is a parallelogram with congruent diagonals."

Instead of attempting to eliminate all student errors from the subject matter program, the author's major concern was the elimination of all <u>sequences</u> of errors of more than 2 or 3 in length. When a student tries several incorrect answers to a question, any continuity between previous exercises and the present or future exercises is interrupted. So, for this author's purposes it was important to have a sequential record of the performance of each student.

It is interesting to note that among the first group of students to execute these geometry lessons, sequences of errors occurred an average of more than twice per student per lesson. By the third group of students such interruptions were occurring less than once in two lessons for each student. It was also noted that, although no particular effort was made to minimize individual errors, as the number of sequences of errors was reduced, the number of individual errors was significantly reduced also. (There may or may not be a relationship here. The simultaneous events are merely noted.)

The teaching logic used in this study provides the student with a dictionary which is stored in memory and is always accessible to the student. To get the information contained in such a dictionary the student must request it. This requires that the author first make a guess as to the manner in which the student will make his requests. For single words there are few problems but for concepts denoted by several words or a sentence; it is difficult to predict how the student will phrase his requests. The computer collected data was useful in refining the dictionary since a record was kept of all attempts to seek information of this type and the words used by the student in his attempt. The data for the first trial of these lessons also made it obvious that the student must be provided with a means of first finding correct spellings of the terms included in the dictionary. Approximately 1/3 of the requests for dictionary information during the first trial of the lessons were unsuccessful due to incorrect spelling. Many of these were not pursued further by the student because he had no way of finding a correct spelling and apparently felt he should not request it of the instructor.

Another measure of the degree of refinement of a set of lessons for CAI is the number of correct answers given by students which are not accepted by the computer. During the first trial of the geometry lessons there were 48 such answers. By the third trial there were only two.