acm

Sigcue Bulletin

Computer Uses In Education

Volume 12, Number 4 ACM SIGCUE Bulletin October, 1978

TABLE OF CONTENTS

Editor's Column 1

Chairman's Message 1

Computer Science Education for Preservice Elementary School Teachers (Moursund) 3

Some Problems of Translating LOGO to a Romance Language (Curado and Valente) 11

Memo: Elementary and Secondary Schools Committee (Moursund) 30

Bibliography 33

Calendar 40

News 32, 37, 38, 39

Sigcue Bulletin

A Publication of the ACM Special Interest Group on Computer Uses in Education

OFFICERS OF SIGCUE

Chairman
David Kniefel
Director of Academic Services
New Jersey Computer Network, Inc.
P. O. Box 390
New Brunswick, NJ 08903
201-932-8070

Vice Chairman
Stuart Milner
Center for Educational Technology
School of Education
The Catholic University of America
Washington, DC 20017
202-635-5816

Secretary-Treasurer
Joyce Statz
Computer Science Department
Bowling Green State University
Bowling Green, OH 43403
419-372-2330

ADVISORS

Charles Mosmann
Computing Facility
University of California
Irvine, CA 92664

Robert P. Taylor
Teachers College
Columbia University
New York, NY 10027

Karen Duncan
Director, Office of Computer Resources
College of Dental Medicine
Medical University of South Carolina
80 Barre Street
Charleston, SC 29401

SIGCUE Bulletin Editors

Mike Folk
Mathematics Department
Drake University
Des Moines, IA 50311

Stuart Denenberg
Department of Computer Science
State University of New York
College at Plattsburgh
Plattsburgh, NY 12901

EDITORIAL ADVISORS

Karl L. Zinn
UM-CRLT
109 East Madison
Ann Arbor, MI 48104

Wallace Feurzig
Bolt, Beranek, and Newman
50 Moulton Street
Cambridge, MA 02138

Tim O'Shea
LOGO/AI Laboratory
University of Edinburgh
Forrest Hill
Edinburgh, EH1 2QL
SCOTLAND

The SIGCUE Bulletin invites contributions of reports and technical notes, position statements, news items, comments and abstracts of recent publications. Reports and technical notes emphasize design and application of languages and systems in support of instruction and learning. Position statements may discuss any issue bearing on educational uses of computing. News items cover activities of the SIG and groups of overlapping interest. Conference details and calls for papers appear as they are made available.

Reports in the Bulletin are unrefereed working papers unless otherwise stated. Material may be reproduced for non-commercial purposes with credit to the author and the SIGCUE Bulletin.

The SIGCUE Bulletin is published quarterly in January, April, July, and October. Back issues are available from the editor and from ACM headquarters at $2 per issue for ACM members, $4 for non-ACM members, and $5 for institutions.

Membership in SIGCUE, which includes the Bulletin, may be obtained through SIGCUE. Association for Computing Machinery, P. O. Box 12105, Church Street Station, New York, NY 10249. Annual dues are $5 for ACM members, $3 for ACM student members, $12.50 for non-ACM members, and $16.50 for institutional membership. A membership application can be found on the inside back cover of the Bulletin.

Note: This issue rec'd at ACM HQ for production on 9/15/78.

EDITOR'S COLUMN

This issue contains reports of research in the areas of computer literacy, and LOGO. Moursund sets forth a curriculum designed to impart a computer literacy to future elementary education teachers so that they, in turn, can pass the light to their students. Valente and Curado describe and attempt to solve some of the semantic difficulties that arise when a programming language (LOGO) is implemented in a natural language other than English.

We also reprint a memo we received from David Moursund on the formation of an ACM sponsored elementary and secondary schools committee. We encourage readers to read the memo and respond.

Readers will note from the letter below that the second of your editors is Stuart Denenberg. We have divided up the editorial duties with Stu handling the reports and Mike working on reviews, calendar, announcements, and final preparation of the document.

Letter to SIGCUE Readers:

I look forward to acting as co-editor of SIGCUE with Mike Folk. Joyce Statz has done a remarkable job not only as editor, but as critical reviewer, researcher, and sounding board. Now that I have the stack of correspondence associated with submitted articles I can fully appreciate the fine work she has been doing -- the quality of her work is on par with the editors of Scientific American who competently extend their helping hand in the writing and rewriting of articles (but Joyce is to be more admired since she receives no pay!).

I hope to carry on this tradition with Mike and to continue to provide our readers with interesting and enjoyable articles about the uses of computers in education.

Stu Denenberg

CHAIRMAN'S MESSAGE

A meeting on August 26, 1978 between Representatives of ADCIS, AEDS, and SIGCUE, resulted in the following draft statement of purpose and goals for the National Educational Computing Consortium (NECC):

NATIONAL EDUCATIONAL COMPUTING CONSORTIUM*
STATEMENT OF PURPOSE DRAFT
AUGUST 26, 1978

The National Educational Computing Consortium is a group of non-profit professional societies that are dedicated to the improvement of education thru computing. The common issue which has led to the banding together of these groups is a concern for computer use in the learning process. With the growing impact of computers in our society, it is evident that computing

* SIGCUE's involvement in NECC is subject to approval by the ACM SIG Board and Executive Committee.

will continue to have an increasingly important role in the learning society in the next several decades. It is incumbent on us as societies whose members are professionals in educational computing at all levels to be concerned with the direction and nature of this growth. To that end, we believe that there are benefits to be gained from joint activity which are not securable through individual efforts of the constituent groups.

GROUPS

The National Educational Computing Consortium has identified four initial goals:

1. Influence the direction of Educational Computing through activities which provide information and advice for policy formation in government and industry.

2. Conduct special studies and projects which are of broad significance to the future of the use of computers in education, drawing upon the diverse expertise represented in the constituent societies.

3. Provide the constituent societies and the learning society in general with a source of comprehensive information concerning funding opportunities, meetings, and other activities related to computers in education.

4. Generate joint symposia, workshops, and conferences.

MEMBERSHIP

Membership will be non-profit multi-disciplinary professional societies who have as their major purpose the improvement of education thru computing.

The steering committee of the Consortium will be composed of a representative from each member society. The charter members of the consortium are ADCIS, AEDS, and SIGCUE. Additional members will be added to the consortium by unanimous consent of committee members.

The formation of NECC provides a combined constituent member base of approximately 3,000. As other societies join, this base will grow providing a broader base of expertise and potential for joint activities.

The above statement of purpose has been approved by the executive committee of SIGCUE. Thus it appears that the early work of Alfred Bork and Kent Morton is beginning to bear fruit and at least a foundation for cooperation has been established.

On another topic, we have within the last fiscal year through our dues, increase and extreme frugality, wiped out our $4,509 deficit. This refreshir financial news puts us in the enviable position of being able to consider expanded services and new activities for our members. We look forward to planning and initiating new functions through 1979.

I would appreciate any feedback from the membership related to NECC and its proposed purpose. Also, I hope all of you who attend ACM '78, September 4-6 in Washington D.C., will come to our business and technical meeting wher NECC will be a major topic.

David R. Kniefel
Chairman

will continue to have an increasingly important role in the learning society in the next several decades. It is incumbent on us as societies whose members are professionals in educational computing at all levels to be concerned with the direction and nature of this growth. To that end, we believe that there are benefits to be gained from joint activity which are not securable through individual efforts of the constituent groups.

GROUPS

The National Educational Computing Consortium has identified four initial goals:

1. Influence the direction of Educational Computing through activities which provide information and advice for policy formation in government and industry.

2. Conduct special studies and projects which are of broad significance to the future of the use of computers in education, drawing upon the diverse expertise represented in the constituent societies.

3. Provide the constituent societies and the learning society in general with a source of comprehensive information concerning funding opportunities, meetings, and other activities related to computers in education.

4. Generate joint symposia, workshops, and conferences.

MEMBERSHIP

Membership will be non-profit multi-disciplinary professional societies who have as their major purpose the improvement of education thru computing.

The steering committee of the Consortium will be composed of a representative from each member society. The charter members of the consortium are ADCIS, AEDS, and SIGCUE. Additional members will be added to the consortium by unanimous consent of committee members.

The formation of NECC provides a combined constituent member base of approximately 3,000. As other societies join, this base will grow providing a broader base of expertise and potential for joint activities.

The above statement of purpose has been approved by the executive committee of SIGCUE. Thus it appears that the early work of Alfred Bork and Kent Morton is beginning to bear fruit and at least a foundation for cooperation has been established.

On another topic, we have within the last fiscal year through our dues, increase and extreme frugality, wiped out our $4,509 deficit. This refreshing financial news puts us in the enviable position of being able to consider expanded services and new activities for our members. We look forward to planning and initiating new functions through 1979.

I would appreciate any feedback from the membership related to NECC and its proposed purpose. Also, I hope all of you who attend ACM '78, September 4-6 in Washington D.C., will come to our business and technical meeting where NECC will be a major topic.

David R. Kniefel
Chairman

COMPUTER SCIENCE EDUCATION FOR PRESERVICE ELEMENTARY SCHOOL TEACHERS

by

David Moursund
Dept. of Computer Science
University of Oregon
Eugene, Oregon 97403

This material contains a summary of my NCC 78 panel discussion presentation, as well as some detail and references that are not easily presented in a short talk.

The Problem

The elementary school teacher is expected to be broadly knowledgable in subject matter, in pedagogy and learning theory, and in dealing with children. Thus, the college program of the preservice elementary teacher is designed to cover all of these areas. At one time a high school education was considered adequate for the elementary school teacher. Later a two year normal school program was deemed sufficient. Now a four year college degree, and often a fifth year of additional studies, is required. Even with all of this, however, many people feel that elementary teachers are not adequately prepared.

Moreover, there is considerable controversy among educators as to the proper balance of college credits and experiences among the various components of the preservice teacher's program of study. For example, teacher training institutions typically require between 4 quarter hours and 12 quarter hours of mathematics "content", and a course in mathematics "methods". Many college mathematics faculty consider this to be insufficient. They assert that most elementary school teachers are inadequately prepared in mathematics and that this is the reason why so many students are poor at mathematics.

But the preservice teacher has studied mathematics for many years. Typically the preservice teacher has successfully completed the mathematics requirements in grades 1 - 9 and quite often has taken an additional year or two of mathematics during grades 10-12. Thus the preservice teacher enters college with considerable insight into the content of elementary school mathematics.

Contrast this with teacher preparation to make effective use of calculators and computers. Computer scientists ask why elementary teachers are not incorporating calculators and computers into the elementary school curriculum. But the elementary teacher has never experienced the use of either device in an academic setting. They did not exist, or were much too expensive for students, when the teacher was in elementary or secondary school. Even today the typical teacher training institution does not offer calculator and computer training specifically suited to the needs of elementary school teachers. This is a serious problem, which computer science departments and faculty can help to solve.

Current Economic Consideration

Good quality LED calculators retail in the $5-$10 range. For $15 one can buy a LCD calculator, with 2000 hour battery life. Factory built micro-computers from Radio Shack and from Commodore cost $600 for a minimal configuration. While cost is still a major factor, it is no longer the dominant consideration. It will be even less of a consideration in the future, if forecasters of the future of micros are at all accurate.

What Teachers Need to Know

A survey [1] of what computer-related courses and topics might be offered or required of elementary school and secondary school teachers resulted in the general consensus by computer educators that elementary school teachers should be computer literate, and that they should be knowledgeable concerning both teaching with and teaching about calculators and computers. The next three sections of this paper discuss these three areas.

Computer Literacy

For some years now computer educators have been advocating universal computer literacy. The report [2] recommends instruction for computer literacy at the junior high school level. The doctoral thesis [3] gives a comprehensive survey of computer literacy literature as well as an analysis of some possible components of computer literacy.

In brief, computer literacy is a knowledge of the capabilities, limitations, applications, and possible effects of calculators and computers. It is particularly relevant to elementary teachers for two reasons. First, teachers are role models. If they are ignorant and fearful in an academic area, such as calculators and computers, they are apt to transmit this attitude to their students. Certainly they will be unable to correct the misconceptions that children acquire about computers via television, movies, and comic books.

Second, the overall content of elementary education should take into consideration the world the child will face as an adult. To cite an example, at one time many people made their livings as bookkeeping clerks. One of the necessary skills was speed and accuracy at pencil and paper arithmetic. Now, of course, such a person has access to an electronic calculator, and likely has access to a computer. What is important to this position now is good understanding, learning, and problem solving skills. Certainly speed and accuracy at pencil and paper arithmetic are of decreased importance.

Teaching Using Calculators and Computers

Teaching using computers, often called computer assisted instruction, is a vast and well established field. It is certainly too vast to adequately cover in this paper. Perhaps it suffices to say that CAI is becoming more and more common, and that economic considerations have proven to be a major barrier. The microcomputer systems discussed earlier in this paper will go a long way toward destroying the hardware cost barrier. They will also broaden the user base, and thus help spread the cost of software and courseware.

Another important idea is suggested by Seymour Papert's "turtle" project. Papert's approach to computers in elementary education is very hardware intensive. The language LOGO gives young students the ability to deal with an electronic music generator, and electromechanical or CRT turtle, and other interactive input/output devices. In this environment very exciting learning occurs without the use of additional software. Reference [4] contains 23 additional references.

The ordinary 4-function calculator can be used for many instructional purposes. Some of them are discussed in [4,5]. It is evident that relatively little material of this sort has been developed, and that it is not obvious to most teachers how to make effective use of calculators as an aid to instruction of such information [6].

Teaching About Calculators and Computers

The article [9] discusses goals for calculator education in the elementary school. Many mathematics educators now advocate introducing calculators into the elementary school, and integrating them into the curriculum. If this is to be done successfully, then teachers will need to learn how to do it, and what to teach students about calculators.

A more difficult question is what to do about computers. Papert and his followers advocate immersing children into a computer-rich enviornment. The learning is by doing -- by developing, debugging, testing, and using procedures to run an electromechanical turtle, a CRT turtle, a music box, a bell, etc. The teaching of LOGO at the primary level, or a language such as BASIC at the upper elementary level, has been widely examined. The issue is not whether children can learn to program a computer; rather, we are faced with the problems of whether we want to teach them to do so, and who will do the teaching.

There are important aspects of computers besides computer programming. Many computer literacy topics are suitable for inclusion in the elementary school curriculum. The whole concept of machine-assisted problem solving is a good example. If the teacher is computer literate, then one can expect some of this to "rub off" onto the students.

Solutions

Many people's first suggestion for a solution to the teacher training problem we have been discussing is to implement a new course, Computer Literacy for Teachers, and require teachers to take it. Such a course is discussed in [8,9] and was tried out on five campuses in Oregon with about 100 preservice and inservice elementary teachers during winter quarter, 1975. People teaching the course included computer scientists, computer science educators, mathematicians, and college of education faculty. Team teaching was used in several of the courses.

More recently a course based upon [5,10,11] was developed at the University of Oregon. A course outline is given in the next section.

Course Outline

In summer and fall terms of 1977 the Department of Computer Science at the University of Oregon offered a three quarter hour course

entitled "Introduction to Calculators and Computers for Elementary School Teachers". The course was open to any pre-service or in-service teacher and had no prerequisite.

There are four major parts to the course; they are integrated and interwoven to reinforce each other:

1. Running computer library programs and studying examples of calculator and computer activities and materials suitable for use in the elementary school.

2. Studying, listening to lectures, and discussing materials designed to raise the education-oriented computer literacy of teachers.

3. Developing low-level skills in programming in time-shared BASIC.

4. Studying problem solving, and learning the use of calculators and computers as aids to problem solving.

The course is a "content" rather than a pedagogy course. But it includes substantial study of, and hands-on experience with, materials suited to the elementary school.

Week #1 Overview of the course and of the field of calculators and computers in education. Introduction to problem solving and to the ideas of procedure, algorithm, and heuristic. Capabilities, limitations, and similarities and differences of calculators and computers. Definition and discussion of computer literacy, illustrated by a sample exam. Introduction to the vocabulary of calculators, math processing computers, and timeshared computers. Discussion of calculators versus microcomputers versus minicomputers and larger computer systems.

Week #2 Learn how to log onto a timeshared computer system, call and run a library program, and log off. One hour of hands-on experience (during a formally scheduled class) running library programs. Additional hands-on computer experience through required and optional assignments. Calculator familiarization, including experience in using at least two different types of calculators. Introduction to the calculator/computer number system and pecularities of their arithmetic (caused by truncation or rounding, underflow and overflow).

Week #3 History of calculators and computers, making use of films and slides as appropriate. Overview of computer hardware and machine

organization. Relation between hardware and software. Higher level languages, and history of BASIC. Reading and modifying BASIC programs from a computer library. (The student's first programming experience consists of making very small modifications to a library program.) Continued hands-on experience with calculators, with emphasis upon learning to use the memory features of inexpensive calculators.

Week #4 Representation of a procedure via block diagram or flowchart, natural language, and a programming language. Introduction to top-down analysis and stepwise refinement. Writing simple straight line programs that do not involve variables. Use of built-in functions in BASIC, and the importance of subprograms. Introduction to the use of calculators as an aid to instruction (for example, as an aid to teaching mathematics) and hands-on experience with a variety of calculator assisted instruction materials. Use of Texas Instrument's "Little Professor" or "DataMan" machines.

Week #5 The use of numeric and string variables in BASIC. Use of the INPUT, READ-DATA, and LET statements. Writing straight-line programs involving these features of BASIC. Continued hands-on experience with calculator activity materials. Students begin to design and test (with kids) various calculator activities. Materials designed or discovered (via reading) by students are shared with the whole class.

Week #6 The idea and importance of repetition or looping in a computer program. Use of the IF-THEN and GO TO statements in BASIC. Study of possible goals for teaching elementary school students about calculators. Brief discussion of programmable calculators and comparison with computers. Discussion of robots and other "intelligent" machines. Discussion of artificial intelligence and its implications for education.

Week #7 Detailed study of stepwise refinement as an aid to problem solving. Experience in developing detailed natural language procedures via stepwise refinement. Study of examples in which the resolting procedure is suited to implementation on a computer or calculator. Overview of computer science, with major emphasis upon information retrieval and simulation. Hands-on experience in using programs designed to illustrate these areas.

Week #8 Study of debugging and program testing ideas. Use of the RND function in BASIC, and study of computer simulation examples. Work on writing a computer assisted instruction drill and practice program in arithmetic; study of a computerized Career Information System and hands-on experience with it. Overview of computer assisted

instruction, computer managed instruction, and computer augmented learning. Hands-on experience with computer programs designed to illustrate these areas.

Week #9 Complete computer programming assignments. (Note that the DIM statement (lists and arrays) and the FOR-NEXT statements, as well as other more advanced topics, are not covered.) The BASIC programming is intended to be a very low level introduction to programming. Introduction to school administration uses of computers, and to business data processing. Study of Suppes' and other CAI materials. Discussion of major computer eeucation projects such as Plato, Solo, Ticcit, and LOGO.

Week #10 Complete calculator activity worksheet assignment. (Each student is to develop calculator materials suitable for use at the elementary school level.) Discuss problems involved in trying to implement changes to the elementary school curriculum. Cite and discuss examples where major successes have occurred. Continue to study sample curriculum materials and to gain hands-on experience in their use. Discuss what the future holds in terms of hardware, software, and curriculum materials.

References

1. Moursund, David: "Opinions on Computer Science Courses for Precollege Level Teachers," presented at SIGCSE meeting, February 23-24, 1978, and published in the proceedings of that meeting.

2. Conference Board of the Mathematical Sciences Committee on Computer Education "Recommendations Regarding Computers in High School Education," April, 1972.

3. Neill, Michael: "An Emperical Method of Identifying Instructional Objectives for a High School Computer Literacy Curriculum," Ph.D. Thesis, University of Oregon, June, 1977.

*4. Beisse, Brougher and Moursund: "Calculators in the Elementary School," Oregon Council for Computer Education Special Report, May, 1976.

5. Moursund, David: Calculators, Computers, and Elementary Education, The Math Learning Center, 325 13th Street N.E. Unit 302, Salem, Oregon 97401 (June 1977, $7).

6. Calculator Information Center, 1200 Chambers Road, Columbus, Ohio 43212. (They have a number of free reference bulletins.)

7. Moursund, David: "Calculator Metaphors, and Goals for Calculator Education in Elementary Schools," Presented at the February 3, 1977 SIGCSE meeting and published in the SIGCSE Bulletin containing the proceedings of that meeting.

*8. Drummond, Larer, Moursund, Sue Waldman, and Wally Waldman: "Computers in the Elementary School: A course for Teachers," Oregon Council for Computer Education Special Report, January, 1975.

*9. "Computers in the Elementary School: Report of the May 17, 1975 conference held in Eugene, "OCCE Special Report, May, 1975.

10. Moursund, David: BASIC Programming for Computer Literacy, McGraw-Hill Book Company, 1978.

11. Billings and Moursund: Are You Computer Literate? The Math Learning Center, 325 13th Street NE, Unit 302, Salem, OR 97301 (March 1978, $5).

* The Oregon Council for Computer Education is a professional organization of people interested in instructional uses of computers. A year's membership is $5.

OCCE
c/o Carol Forshee, Treasurer
P.O. Box 321
LaGrande, Oregon 97850

SOME PROBLEMS OF TRANSLATING LOGO TO A ROMANCE LANGUAGE

by

F. Curado
J. A. Valente +*
Dept. Ciencia da Computacao,
UNICAMP,
Campinas - SP, Brazil

1. Introduction

Making the computer available to children and other naive users requires attention to the structure of the computer language. Such attention has been given to the design of the language LOGO which attempts to use syntactic forms which feel natural to an English speaker. This paper discusses problems which arise when we attempt to transport into another linguistic environment the design of LOGO and experience accumulated with its use. Specifically we are concerned with making a version of LOGO for use in Brazil.

The only similar enterprize we know is the translation of LOGO into French by Guy Montpetit in Montreal [2]. However our approach is very different. Montpetit made an early decision to stay close to the English usage, especially the usage of the MIT 11-LOGO. This decision was dictated partly by practical consideration: he needed a "French LOGO" immediately, i.e. without first having to do research on this aspect of the problem. But also, the decision reflected strategy of first doing the simplest thing in order to gain experience and appreciate the problems. Since the Quebec group has now provided the experience we do not have to repeat it and have engaged in a study of the more subtle issues involved.

+ Presently on leave of absence at MIT-LOGO Group
* The authors acknowledge the help of Seymour Papert, MIT LOGO Group

We note that the implications of our work are not confined to LOGO alone. Although we frame all our discussion in terms of LOGO, what we say is relevant to the general problem of designing a computer language intended for non-professional users.

In this paper we intend to discuss some of the language-dependent features of the English version of LOGO (ENG-LOGO) and to show that the translation to some other language may not allow the full use of the power embeded in the English version.

We first cover the essence of LOGO principles. After an analysis of ENG-LOGO we go on to discussing some specific translation problems which we foresee as we translate LOGO into Protuguese.

2. Language and Learnability in LOGO

To create a frame of reference, we first go over some ideas that should be quite familiar to the LOGO community. We feel that we will not be simply translating words: we will be translating the words that translate ideas. Hence we ought to talk about these ideas.

The design of LOGO is related to pedagogical principles in two directions which one could call: getting into it and getting something out of it [1].

The first is concerned with the initial steps toward programming. LOGO is designed to bring naive users into it in a smooth, easy, pedagogically sound way; and many features of its structure reflect attempts to do this. The second direction focuses on using computer work as a means to improve the learners ability to learn in other domains. This is also reflected in the structure of LOGO.

The two directions actually coincide insofar as the experience of learning LOGO should be a model for what good learning should be.

We concentrate in this paper on some aspect of the design of the language which involves these pre-occupations and which may be undermined by a literal translation into another language.

We should emphasize another aspect of LOGO philosophy: This is that learning something, including LOGO, must always be seen in

the broader context of a mini-world with several properties:

--there must be a pool of existing knowledge;
--there must be the possibility of active intellectual exploration;
--the new knowledge should emerge as an amalgam of the learners' previous knowledge of the domain, his self-knowledge and powerful problem solving techniques;

The LOGO-Turtle world is a model example of such a mini-world. Focusing on it, we view ENG-LOGO as one component of a system which interrelates anthropomorphism, body syntonics, heuristics, debugging and similar concepts to create descriptions of events, things and processes. By the proper manipulation of these descriptions, learning becomes what it always should have been: a natural process of self-fulfillment.

It is very possible that a different mini-world leads to a different view of what is important in a LOGO language design. Thus we see that the problem of translating LOGO is related broadly to the expectation of how LOGO is to be used, (see the discussions of TO - DEFINE usage in 3.3 and 4.1).

3. An Analysis of ENG-LOGO

3.1 The Grammar of Verbs in LOGO and English

A powerful feature of LOGO for beginners is the easy and "natural" way in which the beginners can define a new command. Programming appears in the metaphor of teaching the computer. It benefits from the child's existing model of teaching (illustrating the getting in) and in turn enriches his concept, for example, of definition (illustrating the getting out). A typical case in which LOGO resembles natural English is the definition of simple commands to produce turtle action. Consider the dialog:

```
? FORWARD 100                    /turtle moves forward
? CIRCLE                         /This will produce an error message
YOU HAVEN'T TOLD ME HOW TO CIRCLE  /This is the computer error message
```

The child's response is to "tell the computer how to CIRCLE". Thus:

```
TO CIRCLE
1 FORWARD 1
2 RIGHT 1
3 GO TO LINE 1
END
```

The smoothness of this process reflects aspects of English syntax. In particular the fact that the imperative used in the command

```
? CIRCLE
```

is lexically identical with the infinitive in

```
"YOU HAVEN'T TOLD ME HOW TO CIRCLE"
```

and the mixed imperative/infinite connotation of

```
? TO CIRCLE
```

In an inflected language such as Portuguese the process is less smooth. So this is one situation we shall have to worry about. Several possibilities are suggested in section 4.1.

A related exploitation of a special property of English is the use of English nouns as LOGO verbs. Thus a program to draw a house is written on the model:

```
TO HOUSE
----
----
----
END
```


It is not unnatural in English to use the word HOUSE as a command. We do not say it is grammatical. The sense in which it is natural is a research topic in psycholinguistics. We cannot fully appreciate the subtle meaning differences (if any exist), between the verbs TO HOUSE or TO FLAG and the use of procedures with these names which draw a HOUSE or a FLAG. The question is further complicated when we notice that TO POLYGON or TO PETAL do not exist as verbs.

The problem illustrates the complex, interdisciplinary nature of the task of designing a language. But in the meantime it is clear that there is a kind of "resonance" with the linguistic intuition of a native English speaker which is not evoked from a Brazilian ear by the literal translation. We see from this also how the multi-disciplinary effort needs a "poetic" component as well as scientific component.

3.2 Commands vs. Operands in ENG-LOGO

We look next at an example which illustrates yet another feature of English which is exploited by ENG-LOGO but which also shows some areas which ENG-LOGO has ignored. In making a Brazilian translation we should not see our goal merely as achieving what has been done by ENG-LOGO. A comparative linguistic study might open the door to solving problems left unsolved by the designers of ENG-LOGO.

In natural language, verbs are used to represent actions and the imperative of verbs are used to command. In the general idea of creating an "everyday life" environment, one of the features introduced in ENG-LOGO was to associate commands with the imperative forms of verbs.

This association has been done in two ways; first by choosing for primitives English terms which resemble, as much as possible, imperative forms. Appendix B presents the etymology of the commands of Appendix A. As it can be seen, the correspondence imperative-form/command can not always be followed.

The second association has been implemented by choosing the word TO to be the switch that turns on editing mode for the definition of procedures. To define a procedure is analogous to creating a new verb for the computer language, and this is used to teach the computer to do something. Since imperative forms of verbs in English are formed by deleting the word TO from the infinitive form, to use this new

command is analogous to commanding (imperative) the computer to do what the word is meant to mean.

Operations are not commands [3] and therefore the English terms chosen to name operation primitives did not have to be imperative forms of English verbs as the list of Appendix C indicates.

This analysis shows that built into ENG-LOGO is the concept that commands are verbs and that operations may not be verbs, and a new command is created by defining a new verb.

However, in some instances we may need to teach the computer a word that is not a command. Example [3]:

```
TO REPLACE :LET :L1 :L2
10 IF :LET = :L1 OUTPUT :L2 ELSE OUTPUT :LET
END

TO LISP :W
10 IF :W = " OUTPUT "
20 OUTPUT WORD REPLACE FIRST :W "S "TH LISP BUTFIRST :W
END

TO MULTILISP :S
10 IF :S = [ ] OUTPUT [ ]
20 OUTPUT SENTENCE LISP FIRST :S MULTILISP BUTFIRST :S
END

?PRINT MULTILISP[ THIS IS A RECURSIVE PROCESS ]
THITH ITH A RECURTHIVE PROCETHTH
```

In the above example, a sequence of three recursive procedures are used to scan a sentence and to replace, in each word of the sentence, the letter S by TH. The procedures are defined as operations. They are named as verbs: REPLACE, LISP, MULTLISP. Anyway we look at it, some confusion may arise. If we stick to the imperative-infinitive schema, then we are not being consistent with the fact that operations are not commands. If we decide that operations are not verbs, an inverse argument can be built.

The example also reinforces the power of ENG-LOGO to allow quite a freedom in the choice of names one can give to procedures. MULTLISP

is not even a word in English, but as it has been constructed as a generalization of the word lisp, it is natural to understand it in ENG-LOGO as a verb of the lisp family. It is perfectly valid for a space buff to call a procedure SPW37 or ART00, regardless of its suitable classification as a command, operation, switch or anything else.

One could argue that the distinction between these types of LOGO-words is not natural or necessary. As we see, it may be useful to keep the distinction as a pedagogical tool to make the point that words and concepts have different uses. On the other hand, if there were only commands, the distinction found in the present implementation would be non-existent.

3.3 Mnemonics

Mnemonics is an issue that focus many of the problems faced in the design and analysis of the LOGO language. For various reasons computer languages primitives compress more information in them than the English word used does. Questions of how much information is embedded, whether or not is possible to follow a general pattern of compressing information naturally arise.

ENG-LOGO uses mnemonics in three contexts. First to represent a larger idea: FORWARD means to move forward. Secondly, to indicate unambiguously the action or operation being performed: HIDETURTLE, BUTFIRST. Finally, to allow the use of shorthand notation (abbreviations) for some frequently used primitives: FD, BK, HT, BF. One key feature of the use of abbreviations is that they are allowed to be typed on the keyboard as inputs, but are echoed in the screen by their full name.

Not much can be done in the first two cases. It is more convenient to express LOGO words as short expressions and not always can one word express the whole idea embedded in a primitive. In one case or another the only real improvement is to find a better term to describe a certain primitive. For instance, the Ediburgh LOGO Group has substituted DEFINE for TO, because they prefer to avoid the ambiguities that TO brings to LOGO.

With respect to abbreviations the situation is a little more complex. If we want to keep the "everyday life" environment then to use abbreviations is a departure from this principle. Moreover, why bother to devise terms like HIDETURTLE, which are full of meaning, if they can be as effectively substituted by cipher. Undoubtedly they are convenient for the expert programmer, but one should wonder what impact they have among beginners. The experience of the Edinburgh Group resulted in abandoning them.

A translation project must also consider all these aspects. A decision will have to be made whether to follow the present structure of ENG-LOGO or to pursue a more independent mode of action.

In summary, what this analysis hopes to have shown is that the natural language English is an integral part of ENG-LOGO. Consequently, ENG-LOGO has to reflect an English speaking culture and one should wonder how much of ENG-LOGO should be taken to another language or culture.

Finally, the analysis also indicates the difficulties in transfering concepts from a highly idiosynchratic means of communication (spoken English) into a highly structured one (ENG-LOGO), when a goal is to keep both languages quite close.

4. Work Strategy for the Translation Project

The translation of a LOGO computer language to other non English contexts must be done is such a way that the translation of the primitives preserves the original LOGO principles.

Hypothetically it would be possible to develop a "new" computer language for the foreign setting by ignoring the work accumulated around the existing LOGO versions. It would have the advantage of avoiding problems of cultural transfers but it would not capitalize on the experience already available.

If we start from ENG-LOGO and try to translate it into a foreign language we can use available technology but we must analyse the points where ENG-LOGO is dependent upon cultural aspects that do not simply "translate".

In the case of Portuguese we will develop the Brazilian LOGO (BRA-LOGO), trying to capitalize what has been done at BBN, MIT,

Edinburgh and in the French version used in Canada (QUE-LOGO). We also mention that we do not discuss the German version because it uses English words for its primitives.

4.1 Teach-Order Metaphor

The teach-order metaphor is used when one defines a new procedure. There are three things involved in this definition: the switch that turns on editing mode, the infinitive form which names the procedure and the imperative form which is used to refer to the procedure. As it happens in ENG-LOGO, TO not only serves as a flag to turn the switch to editing mode but also is used to reflect the teach-order metaphor through the use of an infinitive-imperative structure.

For a Romance language (of which Portuguese is an instance) the first difficult point to circumvent is to maintain procedure generation by the infinitive-imperative analogy. The imperative of a verb in Portuguese is constructed keeping the radical, deleting the infinitive termination (AR, ER, IR, OR) and adding the corresponding imperative termination (E,A,A,ONHA). If the verb is irregular (and there are many such verbs), the imperative termination may vary. For example:

English		Portuguese	
Infinitive	Imperative	Infinitive	Imperative
TO END	END	TERMIN.AR	TERMIN.E
TO LIVE	LIVE	VIV.ER	VIV.A
TO LEAVE	LEAVE	PART.IR	PART.A
TO COMPOSE	COMPOSE	COMP.OR	COMP.ONHA

Obviously, the computer implementation of the infinitive-imperative in this setting has difficult parsing problems, but the question of the infinitive-imperative structure does not stop here. We are not sure that the everyday use of imperative forms in Portuguese has the same meaning as in English. It seems that in popular conversation we

form command orders using various formulations with subtle stress differences. Besides the regular form, we have found instances of using the infinitive and composite forms (VA APANHAR, literally GO TO PICK), to order. On the other hand, if from a noun we want to form a "verb" in English, we simply add TO to the noun (even if this does not mean anything). The equivalent way in Portuguese would be to change the noun by adding a suitable termination and this is quite awkward. As a result, we feel that a LOGO translation to Portuguese will have to be done with the help of a professional applied linguist to help us to understand how imperative forms are commonly used.

There are many ways of approaching the translation. Our first instinct is to keep the teach-order metaphor by simply translating the switch TO to a suitable word, maintaining the infinitive-imperative structure. In Portuguese this will not be simple to implement. In the search for alternative procedures, we found six possible ways of implementing procedure definitions in Portuguese. To help the understanding of this process we use a notation similar to the Backus Normal Form. In ENG-LOGO we have:

TO <word> =====> <word> where <word> is a name of variable

Semantically this means that the right hand side refers to the definition of a procedure which is invoked by using the left hand side.

I. PARA <word> =====> <word> where <word> is a name of variable.

PARA is a literal translation of TO. Using <word> as a name of any variable we leave the infinitive-imperative approach; although in the case where <word> is an infinitive form of a verb, this scheme is equivalent to the use of PARA <infinitive> as a way of teaching and of <infinitive> as an order. Sometimes it gives the proper ENG-LOGO emphasis.

This implementation would be equivalent to the Quebec version.

II. PARA <infinitive> =====> <imperative>

First, not all Portuguese words that finish with R are verbs. Hence the interpreter would first verify that infinitive is really a valid infinitive form, and then it would have to do one of many possible checks to see whether the corresponding imperative form is

being correctly used (a linguistic verification). The parsing problems previously mentioned would be present here.

Secondly, a lot of the freedom to name procedures is lost, creating a "cook-book" approach to parts of LOGO. It seems clear that this alternative should not even be tried.

III. <char.> <word/infinitive> =====> <word/infinitive>

A variation of the cases I and II, where the switch TO is translated into a simple ASCII character that does the switch.

The teach-order metaphor has to be brought "outside" LOGO. The instructor has to say that <char.> is the symbol which allows the computer to be taught.

If it is used with the construction of I, it may be acceptable.

IV. <infinitive verb - root+R-termination> =====> <root+imperative termination>

This is the infinitive-imperative ENG-LOGO analogy, implemented in Portuguese. The switch to invoke editing mode becomes the R-termination of the verb. Case II discusses the feasibility of this approach.

V. PARAQUESE <word> =====> <word> where <word> is a name of variable

In this case, TO is translated by a sequence of words (PARA QUE SE) which lead, in Portuguese, to the use of an imperative form as the name of <word> . The idea of commands being imperative is maintained but, as we are not checking the linguistic validity of <word> for its being imperative, we are back to the free interpretation of the analogy, as in case I. It has also the disadvantage of using three words to form a composite LOGO word.

VI. <reserved word> <word> =====> <word> where <reserved word> :

form command orders using various formulations with subtle stress differences. Besides the regular form, we have found instances of using the infinitive and composite forms (VA APANHAR, literally GO TO PICK), to order. On the other hand, if from a noun we want to form a "verb" in English, we simply add TO to the noun (even if this does not mean anything). The equivalent way in Portuguese would be to change the noun by adding a suitable termination and this is quite awkward. As a result, we feel that a LOGO translation to Portuguese will have to be done with the help of a professional applied linguist to help us to understand how imperative forms are commonly used.

There are many ways of approaching the translation. Our first instinct is to keep the teach-order metaphor by simply translating the switch TO to a suitable word, maintaining the infinitive-imperative structure. In Portuguese this will not be simple to implement. In the search for alternative procedures, we found six possible ways of implementing procedure definitions in Portuguese. To help the understanding of this process we use a notation similar to the Backus Normal Form. In ENG-LOGO we have:

TO <word> =====> <word> where <word> is a name of variable

Semantically this means that the right hand side refers to the definition of a procedure which is invoked by using the left hand side.

I. PARA <word> =====> <word> where <word> is a name of variable.

PARA is a literal translation of TO. Using <word> as a name of any variable we leave the infinitive-imperative approach; although in the case where <word> is an infinitive form of a verb, this scheme is equivalent to the use of PARA <infinitive> as a way of teaching and of <infinitive> as an order. Sometimes it gives the proper ENG-LOGO emphasis.

This implementation would be equivalent to the Quebec version.

II. PARA <infinitive> =====> <imperative>

First, not all Portuguese words that finish with R are verbs. Hence the interpreter would first verify that infinitive is really a valid infinitive form, and then it would have to do one of many possible checks to see whether the corresponding imperative form is

being correctly used (a linguistic verification). The parsing problems previously mentioned would be present here.

Secondly, a lot of the freedom to name procedures is lost, creating a "cook-book" approach to parts of LOGO. It seems clear that this alternative should not even be tried.

III. <char.> <word/infinitive> =====> <word/infinitive>

A variation of the cases I and II, where the switch TO is translated into a simple ASCII character that does the switch.

The teach-order metaphor has to be brought "outside" LOGO. The instructor has to say that <char.> is the symbol which allows the computer to be taught.

If it is used with the construction of I, it may be acceptable.

IV. <infinitive verb - root+R-termination> =====> <root+imperative termination>

This is the infinitive-imperative ENG-LOGO analogy, implemented in Portuguese. The switch to invoke editing mode becomes the R-termination of the verb. Case II discusses the feasibility of this approach.

V. PARAQUESE <word> =====> <word> where <word> is a name of variable

In this case, TO is translated by a sequence of words (PARA QUE SE) which lead, in Portuguese, to the use of an imperative form as the name of <word> . The idea of commands being imperative is maintained but, as we are not checking the linguistic validity of <word> for its being imperative, we are back to the free interpretation of the analogy, as in case I. It has also the disadvantage of using three words to form a composite LOGO word.

VI. <reserved word> <word> =====> <word> where <reserved word> :

APRENDA (LEARN) , or ENSINE(TEACH), or DEFINA (DEFINE),etc. and <word> is a name of variable.

We try to keep the teach-order metaphor using a different name for the editing switch. The choice of the proper word needs more thinking and some experiments with possible candidates. In the case of LEARN or TEACH there are subtle emphasis-meaning problems. DEFINE seems to be more impersonal and more computer oriented. It has the advantage of avoiding awkward situations that are present even in ENG-LOGO (TO POLYGON). This approach also eliminates the command-operation duality of certain user written procedures which only return values. Unless some serious reservations about this approach are raised, it seems that BRA-LOGO should go in this direction, specially because the teach-order metaphor is explicitly introduced if LEARN or TEACH is used.

4.2 Translation of the LOGO Primitives

Under the point of view of translation, primitives present special features, which range from technological to metaphorical. The actual translation of words is generally a simple task, provided that the main general questions of language design and use have been understood.

The technological problem arises when the new natural language uses a set of characters different from the American one. In Portuguese this set is a little different and this difference already imposes heavy burdens. If we go to languages like Greek, Russian or Japenese the situation is much worse.

Portuguese has one different letter Ç (the squiggle was added afterwards by hand!), and four characters used to modify the sound of letters (´ , ` , ^ , ~). Character sets in Brazil come from foreign manufacturers who do not use these characters, and character sets appropriate for Portuguese are not very common.

Sometimes the use of these characters is fundamental and the case of translating MAKE shows this quite clearly. MAKE translates naturally into FAÇA. If Ç is not available we must use C instead, and the word becomes FACA. If FACA were not a Portuguese word, that might be adequate, but FACA means KNIFE. The consequences are obvious.

Adult users normally accept these discrepancies, although even for them the lack of special characters may introduce ambiguities that are hard to solve (as in the case of E - AND and E - IS). For the younger user these problems should be totally avoided and one must develop the proper technology to solve them: modification of keyboards, character codes, etc..

Software and metaphoric problems are present when we look at the etymology of LOGO words, at the use of their abbreviations and at the usage of noise words. Many actions that should be performed by LOGO commands or operations can be well expressed by only one word. Although FORWARD really means move forward, one easily gets by without move. The same does not happen with HIDETURTLE, BUTFIRST, etc.. In ENG-LOGO composite words are formed either using full words or abbreviating one of them (BTOUCH, XCOR, EMPTYP, etc.).

This seems to indicate that the heuristics used for LOGO words are something like:

a) find a natural language word which represents the meaning of the action
b) if not possible, find two or more words to do it: if the resulting composed word is to long, abbreviate one of them.
c) test it for usability.

   c1) if acceptable, end
   c2) return to a)

Two problems arise. First the choice of the proper word(s) to convey the idea. Example: Does BUTFIRST really express the idea of all the elements of a list except the first one? Second when the composite word is too long we must decide which way to abbreviate it. Example: BACKTOUCH is too long and BTOUCH can be made to mean the same thing, but maybe BACKT would be acceptable, as in EMPTYP.

It is then clear that the dependence of the LOGO language on a natural language is important in order to retain the "everyday life" objectives of the environment in which LOGO is expected to operate.

4.3 Abbreviations and Noise Words

As we have discussed previously, to use abbreviations in a LOGO environment may take the learner out of the general idea of "life-like" communication with the machine. In one sense, abbreviations are third level translations.

If they are used (and we are not sure if they should be), their formation should follow precise rules to avoid things like:

CLEARSCREEN	CS
PRINT	PR
PRINTOUT	PO
PRINTSCREEN	PRS

The use of abbreviations should also follow ENG-LOGO implementation which accepts abbreviations but prints out full words. Following the Edinburgh experience, it seems advisable to maintain LOGO short words so that we do not need to introduce beginners to abbreviations.

With noise words the opposite happens. They are used to make communication more "life-like". Example: PRINT FIRST OF MUMPS, where MUMPS is a procedure which outputs a list. OF was originally used in BBN-LOGO, but the MIT 11LOGO version dropped it, as well as other noise words (keeping only THEN).

The decision to use or not to use noise words has to refer to the level of adoption of anthropomorphic ideas, although the introduction of some of them in Portuguese may also create nasty difficulties. For instance, OF THE in Portuguese can be DA or DO.

4.4 Error Messages

Error messages are important feedback information that show how well or poorly the language is understood by the users. One is then faced, initially, with the task of anticipating the kinds of errors that beginners will make, and this will depend on how this language is implemented.

When a computer language is specified, the designer must have models of the user and of the way this language is going to be used. These models will condition him to establish the kinds of most common errors and the form of the corresponding error messages.

In LOGO, the use of anthropomorphic ideas will also have to be brought into consideration because they will induce the learner to certain kinds of behavior which will have to be caught by an appropriate use of error messages.

One valid alternative is to try to separate error messages into syntactic and semantic ones. Another is to create error messages which will indicate to the user that the statement is ambigious. A third alternative is to follow a hierarchy of most common errors.

To separate, a priori, what is a semantic or syntactic error is quite difficult. Even in the case where one decides to impose more stringent grammar rules in order to facilitate error analysis, languages of the class of LOGO are powerful enough to give to the user room to "find" a case where the error system does not work. The case of FD100 versus FD 100 in ENG-LOGO serves to illustrate the points above mentioned. FD100 may be a procedure name (semantics) or a typing error (syntactics). If one decides to restrict procedure names to be formed only with letters (a more constrained grammar) then it is also possible to accept easily FD100 or FD 100 as valid LOGO statements.

To create error messages that give alternative ways of looking at the error requires ingenuity and constant updating of the score of messages. It is technically simple to implement, but one has to carefully maintain the balance between too little or too much information to the user. In the case of FD100, a message of this kind would be: "YOUR STATEMENT WAS NOT UNDERSTOOD. YOU EITHER FORGOT A BLANK OR HAVE YET TO EXPLAIN WHAT FD100 MEANS".

The third alternative can be implemented simplisticly or in a more eleaborate manner. The simple approach would concentrate on the most common errors (something difficult to foresee in a completely new language). The elaborate manner would analyze the error in a tree fashion and sequentially present the user with alternatives.

We do not know yet how to proceed. It will depend upon the need to rewrite or simply adapt the BBN-TOPS 10-LOGO that is now available in Campinas. It will also reflect the experience that the UNICAMP-LOGO Community will have as LOGO is used more and more with children and adults.

5. Conclusion

We hope to have made clear our contention that LOGO languages may be quite dependent on particular features of the corresponding natural language. The consequences are twofold. First, one should take a close look into ENG-LOGO to analyze these dependencies in order to decide which ones should be capitalized on and which ones should be avoided.

Secondly, one should approach a translation to a foreign language very carefully, trying to profit from what was seen in ENG-LOGO and from what can be used from the foreign natural language structures.

The analysis of ENG-LOGO also shows some important features of computer languages in general, which seem to have not been well studied. Issues of COMMAND vs OPERAND, and information compressing in primitives must be dealt with if one wants to understand how people employ (or should employ) computers to solve problems. The translation of LOGO to Portuguese may help to clear some of these points.

Finally, we hope that the ideas brought forth in this paper will be useful to the American LOGO Group interested in creating a Spanish version of LOGO for use in the U.S..

Bibliography

[]] - Papert, S. - "Notes for a LOGO Book"; unpublished notes; August, 1977.

[2] - Montpetit, G. - "Un Glossaire des Primitifs LOGO"; unpublished paper; Groupe LOGO, College Edouward, Longueil, Canada; 1973.

[3] - Abelson, Goodman and Rudolph - "LOGO Manual"; Cambridge, Mass: LOGO Memo 7; 1974.

[4] - Goldenberg, E.P. - "A Glossary of PDP 11 LOGO Primitives"; Cambridge, Mass: LOGO Memo 16; 1975.

APPENDIX A

Basic ENG-LOGO primitives used by beginners [4].

General purpose

EDIT	END	ERASE
GOODBYE	PRINT	READ
REQUEST	TO	WRITE
MAKE	TEST	

Turtle purpose

BACK	FORWARD	LEFT
PENDOWN	PENUP	RIGHT
CLEARSCREEN	HEADING	HERE
HIDETURTLE	HOME	NODISPLAY
SETX, SETY	XCOR, YCOR	WRAP
SHOWTURTLE	BTOUCH	FTOUCH
LAMPOFF	LAMPON	LTOUCH
RTOUCH	TOOT	

List purpose

BUTFIRST	BUTLAST	COUNT
EMPTYP	FIRST	LAST
SENTENCE	WORD	

Numeric purpose

DIFFERENCE	EQUAL	GREATER
LESS	PRODUCT	QUOTIENT
RANDOM	REMAINDER	SUM

Program control purpose

GO	IF	IFFALSE
IFTRUE	STOP	

APPENDIX B

The etymology of the ENG-LOGO commands from Appendix A.

Verb	Non Verb	Unclear
EDIT	GOODBYE	PENUP
ERASE	FORWARD	PENDOWN
PRINT	LEFT	
PRINTOUT(*)	RIGHT	
READ		
WRITE		
BACK		
HIDETURTLE(*)		
HOME		
SETX, SETY(*)		
SHOWTURTLE(*)		
TOOT		
MAKE		
TEST		

*jointed two words

APPENDIX C

The etymology of the ENG-LOGO operations from Appendix A.

Non Verb	Unclear	Verb
XCOR, YCOR(*)	REQUEST	COUNT
HEADING	SUM	
HERE	PRODUCT	
BTOUCH(*)	QUOTIENT	
FTOUCH(*)	EQUAL	
LTOUCH(*)	DIFFERENCE	
RTOUCH(*)		
BUTFIRST(*)		
BUTLAST(*)		
EMPTYP(*)		
FIRST		
LAST		
SENTENCE		
WORD		
REMAINDER		
RANDOM		
GREATER		
LESS		

* jointed two words

MEMORANDUM

To: People interested in computer usage at the precollege level

From: David Moursund, Chairman
ACM Elementary and Secondary Schools Subcommittee

Date: June 16, 1978

The ACM Secondary School Subcommittee and the ACM Teacher Certification Subcommittee have been merged into a new committee, the ACM Elementary and Secondary Schools Subcommittee (ES^3). Final approval for the formation of this new committee was given by the Curriculum Committee on Computer Education at its June 6, 1978 meeting held in Anaheim, California. David Moursund, who was chairman of the Teacher Certification Subcommittee, was appointed chairman of the new committee. David Johnson, who was chairman of the Secondary School Subcommittee, has resigned and has accepted a Mathematics Education Professorship in London, England. The CHARTER of the new committee is as follows:

ELEMENTARY AND SECONDARY SCHOOLS SUBCOMMITTEE (ES^3)

The Elementary and Secondary Schools Subcommittee is the ACM representative in the elementary and secondary schools area. It is an outgrowth and consolidation of the Secondary Schools Committee (formed in 1967) and the Teacher Certification Subcommittee (formed in 1975). The committee is assigned the task of helping to foster and guide the study of and instructional use of computers at the elementary and secondary school levels. The purposes of the committee are:

1. To identify goals and objectives for instruction on computer science (including computer literacy) at the elementary and secondary school levels, and to assist in their implementation.

2. To identify goals and objectives for instruction on computer applications and implications at the elementary and secondary school levels, and to assist in their implementation.

3. To develop guidelines for the preparation of teachers of computer science (including computer literacy) at the elementary and secondary school levels.

4. To develop guidelines for preparation of teachers in disciplines other than computer science, at the elementary and secondary school levels.

The committee will meet regularly at national conferences.

There are many possible topics that the ES^3 could work on. Some suggestions that were discussed at the Anaheim meeting and/or at the Computer Science Conference this past winter are as follows:

1. To develop specific and detailed recommendations for computer-related content to be taught at the precollege level. This might include specific computer science and computer literacy courses. It could also include units of study or specific topics to be incorporated into courses that currently do not discuss computer-related topics.

2. To develop specific and detailed recommendations on teacher training. These might include a program of study for teachers wishing to be prepared and/or certified as "computer science teachers." It could also include recommendations for elementary teachers and for secondary teachers wanting or needing a less intense exposure to computers.

3. To address the topic of deciding what microcomputers facilities can help meet the needs of a school. Many teachers and school administrators who have very little background in computer science are being faced with a computer acquisition problem. A tutorial-type paper could be written to familiarize them with some of the general considerations and options. It could help guide beginners through the computer acquisition maze.

The ES^3 has a modest budget from the Association for Computing Machinery. It appears likely that this budget may be sufficient to help defray the expenses of about a dozen people to attend a two-day planning and working session of the committee. This meeting is tentatively being planned for December 2 and 3, 1978 in Washington, DC. This is the two days prior to ACM '78, to be held December 4-6 in the same location.

There are three ways to participate in the work of this committee.

1. Passively. Send your name, address, and a brief description of your involvement in the computer education field to David Moursund. You will be put on a mailing list to receive copies of the documents produced by the Committee.

2. Actively, via correspondence. Write a description of one or more of the major problems you would like the committee to work on. Suggest what the committee might do, and how you are willing to help. Write your thoughts on the three problems listed earlier in this memo. Write about how these problems are being solved in your school, school district, or state.

3. Actively, via correspondence and attending meetings. ES^3 will meet during the ACM conference December 4-6 in Washington, DC. It will meet during the Computer Science Conference February 20-23, 1979 in Dayton, Ohio. It will meet during the National Computer Conference next spring in New York. Perhaps you live close enough to one or more of these locations to attend a meeting. If so, plan to come.

As indicated previously, the ES^3 has a very limited budget. It plans to use this budget to help defray expenses of a few people who wish to participate actively in the committee's work. This willingness must be demonstrated via correspondence to David Moursund. Write out your ideas in detail and mail them to:

David Moursund
Dept. of Computer Science
University of Oregon
Eugene, Oregon 97403

▽ △ ▽ △ ▽ △ ▽

NSF COMPUTER LITERACY STUDY

A two-year study of COMPUTER LITERACY is being conducted by the Minnesota Educational Computing Consortium (MEDD) under a grant from the education division of the National Science Foundation. It is intended that the study results will provide valuable information regarding the ways elementary and secondary teachers are using computers, or teaching about computers in their classrooms. The effects of these various ways upon student knowledge of, attitudes toward, and ability to use computers are of particular interest. As a result of the study, computer literacy instructional materials and measurement instruments will be developed.

To aid in the development of computer literacy materials and tests, it would be helpful if persons who are doing work in the computer literacy area would send sample course/unit descriptions, objectives, curriculum materials and measurement instruments to the Computer Literacy Project, MECC, 2520 Broadway Drive, St. Paul, MN 55113 (1-612-376-1145). For further information contact Dan Klassen or Tom Hansen at the following address: Minnesota Educational Computing Consortium, 2520 Broadway Drive, Saint Paul, Minnesota 55113, Telephone (612) 376-1122.

Books

Bailey, Daniel E. (Ed.) Computer Science in Social and Behavioral Science Education. Englewood Cliffs, NJ, Educational Technology Publication, 1978. 520 pages, $19.95. A collection of papers treating computer science as an instructional tool in the social and behavioral sciences. The seven sections (covering thirty-five chapters) are: (1) the place of computer science in social and behavioral science, (2) statistics and methodology, (3) social impact, (4) computer based enhancement in social and behavioral science education, (5) simulation and modeling, (6) two bibliographies (science fiction, and computing in social and behavioral science education), and (7) the bottom line (cautioning advice and counsel).

Dwyer, Thomas A. and Margot Critchfield, BASIC and the Personal Computer. Addison Wesley, 1978, 438 p., $12.95. A very comprehensive book both in its coverage of BASIC and of the capabilities of turnkey microcomputer systems. Loaded with examples, exercises, and projects involving a much wider than usual range of things to do with a computer. Educators who use microcomputers or BASIC, or both, should have a book at this book. Critchfields illustrations are a delightful bonus.

Hunter, Beverly, Academic Computing at Denison University, a Case Study, Hum RRO (300 N. Washington St., Alexandria, VA. 22314), 1978. First in a series of 21 planned monographs emerging from the NSF-supported study "Exemplary Institutions in Academic Computing."

Le Cuyer, E.J. Introduction to College Mathematics with A Programming Language. Springer-Veilag New York, 1978, 420 pp., $14.80. One of several math texts we have seen recently which integrates APL into its presentation. This one covers most traditional precalculus topics, including set theory, logic, matricis and determinants, and elementary function. Differential and integral calculus are also introduced, in two chapters. Probability, statistics and trigonometry round out the coverage.

APL was originally designed as a language for communicating mathmatical ideas. This text represents a serious attempt by the author to use it for just that. Conventional notation is presented in parallel with the APL notation.

Although an admitted APL enthusiast, we have never used APL in the manner indicated by this book because we've never had the ready access to APL that is required. We would welcome assessments by any readers who feel they can comment on this sort of use of APL in a reasonably objective fashion.

Periodicals

Casmey, Howard B. "Computers and Education! A Commissioners View" MAEDS Journal, Vol. 2, No. 1, Winter 1978. Discusses the role of CAI and CMI in fostering mastery learning in schools on a statewide basis.

Charp, Sylvia, and Robert N. Sebastian. "Cost Effectiveness Considerations in the Use of a Combined CAI/CMI System," The Journal Vol. 5, No. 2, March/April 1978, p. 42 from their abstract: The difference between CAI measures of cost-effectiveness and those of a combined CAI/CMI system are discussed. Proceeding on a conservative assumption that the effectiveness of a CAI/CMI system is equal to that of labor-intensive systems of instructional management, a detailed analysis of the cost of Philadelphia CAI/CMI system as the "instructional hour," or time-on-task within the management system, the current range of cost of CAI/CMI is computed between eight and forth cents per instructional hour, compared with a median labor cost of over fifty cents per instructional hour. The annual cost per student of $35.09 makes the use of a combined CAI/CMI system highly competitive with alternative strategies and media of instruction.

Costello, Scott H. "Readability and the Computer," MAEDS Journal, Vol. 2, No. 1, Winter 1978. Computers can be used to determine readability of text much more efficiently than humans, but not without problems.

Dennis, J. Richard, "Training Preservice Teachers to Teach With Computers," in AEDS Journal, Vol. 11, No. 2, Winter 1978. A position paper in which the author outlines an instructional model for training preservice secondary teachers.

Folk, Mike, "A Simple Microcomputer-controlled Programmable Robot." Calculators/Computers, Vol. 2, Issue 5, May 1978, p. 11. A low cost "tutle" project is described. The turtle, computer and software are discussed.

Hallworth, H. J. and Ann Brebner. "Computer Assisted Learning for the Mentally Retarded," Calculators/Computers, Vol. 2, Issues 4 and 5, April and May 1978. A two-part article describing principles guiding a project on computer assisted learning at the Vocational and Rehabilitation Research Institute in Calgary, Canada.

Howze, Glen, "An interactive Software System for Computer - Assisted Testing," AEDS Journal, Vol. 11, No. 2, Winter 1978. TICAT (Tushegee Institute Computer-Assisted Tester) can be used for (1) development of exam item files; (2) creation and/or administration of exams, and (3) scoring, record keeping and report generation. Development and use of the system are described.

Lund, Charles and D. Thomas King, "QUICKSCORE: A computerized Test and Survey Analysis Package," THE Journal, Vol. 5, No. 3, May/June 1978, p. 35. A program which scores multiple choice tests and provides analysis of results. Thorough description of how the package works, and even a section on "How to Hustle the Equipment and Obtain the Package."

Molnar, Andrew R. "The Next Great Crisis in American Education: Computer Literacy," THE Journal, Vol. 5, No. 4, July/August, 1978 pp. 35-38. Our society is changing from one based on production of goods and services to one emphasizing science and knowledge, resulting in new and different demands on American education. One such demand is the achievement of an acceptable level of computer literacy among the population. At the same time, American education is drawing away from the sort of commitments needed to meet new and different demands. The result, Molnar argues, will be an educational crisis. He suggests some ways in which schools might meet these demands, as well as the social costs if they are not met.

Statz, Joyce, "Compu-Kids: A Project for 'Children' of All Ages," The Arithmetic Teacher, Vol. 25, No. 8, May, 1978. Describes a project begun several years ago at Bowling Green State University in which a faculty member and several students teach BASIC to about forty children per quarter. Goals and outcomes are described, with ideas on how school teachers could get their own project started.

"11th Annual Guide to Audio/Visual Instruction for Data Processing." Computing Newsletter for Schools of Business, Vol. XI, No. 9, May 1978. Primarily a listing of the courses available from the major vendors in the field.

SOURCES

Sources reviewed regularly for inclusion of articles in the Bibliography include those in the following list. If there are other important education journals which should be consulted, please communicate with the editor.

ACM publications, including the Communications of the ACM, Computing Reviews, and a number of publications of the Special Interest Groups.

AEDS Journal, a quarterly publication is available from the Association for Educational Data Systems, 1201 Sixteenth Street, NW, Washington, DC 20036.

Audio-Visual Instruction, published 10 times a year by the Association for Educational Communication and Technology, 1126 16th St. NW, Washington, DC 20036

CATC Digest, published quarterly by ETS, available free from Lynne Geller, Editor, Educational Testing Service, Princeton, NJ 08540.

Computer Education, a publication of the Computer Education Group of the British Computer Society, available through the editor, Peter Lambert, National Computing Centre, Quay House, Quay Street, Manchester, M33HU, England.

Computers and Education, published quarterly by Pergamon Press, Inc., Maxwell House, Fairview Park, Elmsford, New York 10523.

Computing Newsletter for Schools of Business, published nine times a year by the College of Business Administration at the University of Colorado, Colorado Springs, CO 80960.

Creative Computing, published bi-monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

EDU, the Education Products Group Magazine of Digital Equipment Corporation, available from the Education Products Group, DEC, Maynard, MA 01754.

Educational Technology, published monthly by Educational Technology Publications, 140 Sylvan Avenue, Englewood Cliffs, NJ 07632.

H-P Newsletter, the Hewlett Packard Educational Users Group Newsletter available from Educational Users Group, 5303 Stevens Creek Blvd., Santa Clara, CA 95050.

Journal of Computer Based Instruction, published by ADCIS, available from JCBI, 215 South Main, Clarion, IA 50525.

Journal of Educational Data Processing, published quarterly by IDEOS, P.O. Box 867, Soquel, CA 95073.

Journal of Educational Technology Systems, published irregularly by the Society of Applied Learning Technology and available from Baywood Publishing Company, 43 Central Drive, Farmingdale, NY 11735.

MAEDS Journal of Educational Computing, published by the Minnesota Association for Educational Data Systems (at least once per year), MAEDS, 1925 West County Road B2, St. Paul, MN 55113.

On-Line, published 5 times per year by Project EXTEND and CRLT at the University of Michigan, available from Karl Zinn, U-M CRLT, 109 East Madison Street, Ann Arbor, MI 41804.

Oregon Computing Teacher, published by the Oregon Council for Computer Education, 4015 SW Canyon Road, Portland, OR 97221.

People's Computers, published six times per year by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025.

Personal Computing, published bimonthly by Benwill Publishing Corporation, 401 Louisiana SE, Albuquerque, NM 87108.

THE Journal (Technological Horizons in Education), available from the Circulation Department, THE Journal, P.O. Box 992, Acton, MA 01720.

****

ACM HOLDS WORKSHOP ON COMPUTERS AND SOCIETY

The Association for Computing Machinery (ACM), with support from the National Science Foundation (NSF), held a workshop on courses and educational materials on computers and society and computer literacy on July 16-19 in Williamsburg, Virginia.

In addition to the curriculum workshop the project is also developing a comprehensive bibliography of materials on computers and society and computer impact.

The opening day of the workshop was devoted to the presentation of forward looking position statements regarding computer impact. These were amplifications of working papers distributed prior to the conference and will be collected into a conference record to be published. The presentations included discussions of privacy issues, computing, manpower requirements in computing, legal issues, computer crime, implications of computers on medicine, computer uses in government, and Electronic Funds Transfer System.

Following these general discussions, small working groups discussed the nature and scope of computer impact on society and ways in which the topics might be approached in education programs. Five areas were pre-college education for the computer specialist, post college education for professionals in other than the computer field that are involved in aspects of computing, and programs of public information. These recommendations will be discussed at the SIGCSE Technical Symposium, COMPSAC '78, and will become part of the workshop proceedings.

In addition to the presentation at the conferences mentioned above, the bibliography and proceedings of the workshop are to be published in the late fall and the final project report completed by early spring, 1979.

ANNOUNCING

Third Annual National Scholastic Programming Contest

Under the sponsorship of the ACM Student Membership and Chapters Committee and Upsilon Pi Epsilon (Computer Science Honorary), the Third Annual National Scholastic Programming Contest will be held in Dayton, Ohio on February 21, 1979 in conjunction with ACM's Computer Science Conference.

The contest will be similar to the regional programming contests and the first two national contests. Two teams of four student programmers each will be chosen to represent each ACM region, hopefully through a regional contest. The contest is conducted over an eight-hour period in which the teams devote their efforts to producing four ANS FORTRAN programs which produce the correct results when run with the "hidden" official data. Team membership is limited to currently enrolled students taking a half-time or larger course load who had not been employed as full-time programmers for a period longer than a summer or one normal co-op period.

Each ACM Region (except European) is expected to hold a Scholastic Programming Contest before Mid-January, 1979. The ACM Regional Representatives will be aware of contest activities within their respective regions and will certify the teams for competition in the National Contest, which will partially support the travel costs for the teams going to Dayton.

For further information on Regional or National Contests, please contact:

Dr. John R. Metzner
Computer Science Department
University of Missouri - Rolla
Rolla, MO 65401
(314) 341-4496

EMPLOYMENT REGISTER
ACM Computer Science Conference
Stouffer's Dayton Plaza Hotel/Dayton Convention Center, Dayton, Ohio
February 20 - 22, 1979

The Seventh Annual Computer Science Employment Register will be conducted at the Dayton Computer Science Conference. This Register, the only one of its kind, aids in matching computer scientists and data processing specialists with employer opportunities.

The purpose of the Register is to provide a mechanism for establishing contact between applicant and employer in a professional manner. The Register operates as follows: the applicant completes a form giving identifying information, education, publications, experience, interests, references, position and salary desired. Provision is made for submission of an anonymous form if desired. The employer completes a similar form. This information enables the applicant and employer to determine the desirability for follow up.

As in past years, the policies and procedures, listed below, will be in effect:

1. Both applicants and employers must file their registration on official forms. Three different forms will be used: (1) applicant, (2) academic, and (3) business, industry, and government. These forms may be obtained from and completed forms should be returned to:

   Orrin E. Taulbee
   ACM Computer Science Employment Register
   Department of Computer Science
   University of Pittsburgh
   Pittsburgh, Pennsylvania 15260

2. Closing date for acceptance of forms is January 30, 1979. The inclusion of a late form cannot be guaranteed.

3. The completed forms will be compiled to form four books of listings: student applicant, non-student applicant, academic employer, and business, industry and government employer. Multiple copies of these books will be available at the Conference for review.

4. Charges: Applicant
   (a) Student - No charge (must be certified as student at time of filing by signature of Department Chairman.)
   (b) Non-student - $5.00
   (c) Anonymous - $10.00 for student or non-student
   Employer - $30.00 per form submitted
   Personal copies of books - $25.00 each book

   A check drawn on U.S. bank for the appropriate amount (payable to Computer Science Employment Register) must be sent with the completed form.

5. The Employment Register Staff will operate a message desk and maintain employer sign-up sheets at the conference to facilitate making contacts. Actual arrangements for interviews will be the responsibility of the employer and the applicant.

6. Information on availability and cost of books (including late listings) after the Conference may be obtained by writing to the above address.

EMPLOYMENT REGISTER STAFF CANNOT ASSUME RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS TIMELINESS, OR GOOD FAITH SHOWN IN THE APPLICANT OR EMPLOYER LISTINGS.

CALENDAR

November 14-16, 1978 Washington, DC

EDUCOM Fall Conference and Annual Meeting. Mayflower Hotel. Theme: The Reality of National Computer Networking for Higher Education - EDUNET. An assessment of current experience, current limitations, and the future role of computer networking for colleges, universities, and non-profit corporations. Chairman: William Miller, Provost, Stanford University. For information, contact Carolyn P. Landis, EDUCOM, P. O. Box 364, Princeton, NJ 08540.

November 30 - December 1, 1978 Tucson, AZ

Conference on retreival and use of educational resources. Contact Lotus M. Knief, OIIS, College of Education University of Arizona, Tucson, Az 85721.

December 4-6, 1978 Washington, DC

ACM '78. Sheraton Park Hotel. See program note on page.

February 20-22, 1979 Dayton, OH

ACM Computer Science Conference. Stouffer's Dayton Plaza Hotel and Dayton Convention Center. Will include state of the art papers, and short submitted reports on current research by students, faculty and researchers in the computer and information sciences. Abstracts only are required; the deadline for submission is December 1, 1978. The conference also features an employment register, book displays, and an instructional equipment display. For information contact Lawrence A. Jehn, Computer Science, University of Dayton, Dayton, Ohio 45569. (513 339-3831.

February 22-23, 1979 Dayton, OH

SIGCSE Symposium. Stouffer's Dayton Plaza Hotel and Dayton Convention Center. For information, contact Douglas Kerr, Computer and Information Science, Ohio State University, 2036 Neil Avenue, Columbus, OH 43210.

February 26 - March 1, 1979 San Diego, CA

ADCIS Winter Meeting. Contact John J. Hirschbuhl, 206 Carroll Hall, The University of Akron, Akron, OH 44325, 375-7781.

APPLICATION FOR SIG MEMBERSHIP*

ASSOCIATION FOR COMPUTING MACHINERY
1133 AVENUE OF THE AMERICAS, NEW YORK, N. Y. 10036

Please enroll me as a member of the SPECIAL INTEREST GROUP ON COMPUTER USES IN EDUCATION (SIGCUE)

Membership includes Newsletter subscription. Please make checks payable to:
ACM, Inc., P.O. Box 12105, Church Street Station, New York, N. Y. 10249

Name (Please print or type) _____

Mailing Address _____

_____

City _____ State _____ Zip _____

Signature _____

☐ New address. Please change my ACM record.

*NOTE: For ACM members renewing within the next three months, do not use this application. Simply add this SIG to your renewal invoice when you receive it and return with the appropriate additional payment.

Annual Membership Dues are $5.00 for ACM Members, $3.00 for ACM Student Members and $12.50 for Non-ACM Members.

☐ **ACM MEMBER**
ACM Member No. _____
Send no money now.
Dues are payable when ACM membership is renewed.

☐ **ACM STUDENT MEMBER**
ACM Student Member No. _____
Send no money now.
Dues are payable when ACM membership is renewed.

☐ **NON-ACM MEMBER**
Enclosed is annual dues of $12.50
Bylaw 7 Section 3 of the ACM Constitution states that non-ACM members may be members of a SIG provided that their major professional allegience is in a field other than information processing. Please state your major professional field _____

☐ **SUBSCRIPTION TO SIGCUE NEWSLETTER ONLY**
Enclosed is payment for annual subscription of $16.50
Note: Subscription is included with membership dues.

☐ Please send information on ACM Membership.

Association for Computing Machinery
1133 Avenue of the Americas, New York, New York 10036

NON-PROFIT ORG.
U.S. POSTAGE
PAID
NEW YORK, N.Y.
PERMIT NO. 2397

1691120 00 7903
LIZA J LOOP
PO BOX 945
COTATI
SEP25
CA 94928