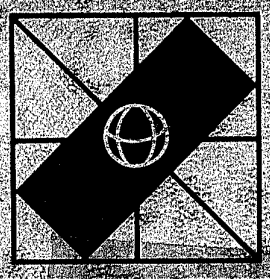
computers in education

edited by

o. lecarme and r. lewis



IFIP

north sevier

Computers in Education, O. Lecarme and R. Lewis (eds.)

A SYSTEM FOR TEACHING PROGRAMMING BY MEANS OF A BRAZILIAN MINICOMPUTER

Sueli M. dos Santos
Associate Professor

Marilia R. Millan
Assistant Professor

Departamento de Informática
Pontificia Universidade Católica
Rio de Janeiro - Brasil

The design of an automatic system for teaching a simple programming language is described. The teaching system is flexible enough to be applicable to other subjects, and is part of a larger system which includes the language processor and a dedicated time-sharing operating system. The system will be built in a minicomputer that is being developed in Brazil. Communication between users and machine will be established through terminals. The general features of the teaching system are emphasized rather than the details of its implementation.

1. INTRODUCTION

To find an efficient way of turning computer science into an understandable discipline for a beginner is as yet one of the main problems in the field of Computer Education. This is particularly true in Brazil where no systematic program of education in that field exists at any level before college. However there is an educational reform program under way now in Brazil one of whose aims is to provide the system with tools adequate enough to cope with the problems growing out of the fast economic development the country has been experiencing for some years. To make students familiar with computing methods beginning from the elementary school level may be considered part of that reform effort.

The semi-automatic system described here is a system devised to introduce students at secondary level to problem solving methods through a minicomputer. The high level programming language we use is a Portuguese version of a modified BASIC language (PBASIC). The main change that was made is the introduction of constructions of the 'DO-WHILE' type instead of the usual instructions of iteration type common to most of the BASIC versions. This modification makes it possible to construct programs using block structures. Our intention was to accomplish two objectives: first, to teach how to program in a structured manner from the beginning; second, to use a simple language in Portuguese since the system is tailored for secondary level students. Therefore the BASIC Automatic Teaching System (BATS) intends to be an introductory course on computers for students at secondary level. We hope that this course will be a step towards the implementation of a systematic program for that school level. On the basis of such assumptions the use of a Portuguese version of BASIC is totally justified.

Computer Aided Instruction (CAI) methods were chosen due to the shortage of human resources in the area of computer education. Our intention in designing the course was to make it applicable also to the training of instructors for a computer education program.

Two premises are assumed in BATS design: first, the necessity of individualizing the learning process; second, that the users have different educational backgrounds. Hence the decision to divide the information provided by the course into two levels: optional and compulsory. The optional information set tackles the problems stemming from gaps in the user's information background and from the user's need to receive further information on certain topics.

Several types of information are provided by BATS: explanatory texts, exercises, examples, quizzes, references. BATS emphasizes the number of exercises, examples and quizzes. Explanatory texts are, in general, direct and concise. A manual containing more detailed information is provided along with BATS.

During the course the user has to write programs in PBASIC and test them by means of an activation of the PBASIC processor by BATS.

The system design is flexible enough to be adaptable to other subject matters. This claim is justified on the basis of the fact that the main parts of BATS, that is, the data-base (content of the course) and the monitor (the program that manages the course) are independent. Thus the same monitor can be easily changed to handle a different data-base. To teach a completely different subject, for example History, the changes in the monitor need only be made with respect to the way we handle optional and compulsory information.

In designing our system we took into consideration several automatic teaching systems mentioned in the references. However the peculiar situation of the Brazilian School System forced us to attempt the design of a system to cope with the problems placed by it.

The lack of space does not allow us to discuss here the advantages and disadvantages of our system compared with others. It is our intention to tackle this problem in a later paper.

2. THE TEACHING SYSTEM

2.1. Structure of the data-base

The course is divided into lessons. Each lesson is a sequence of information units that may be described as a system having a nucleus with which several peripheral units are associated. The central nucleus contains the main information of the unit. Peripherical units contain the following types of information: examples, exercises , additional explanations, references, hints and so on.

A lesson, in general, is a sequence of the following sets of information units:

i) an introductory unit containing a short description of the topic to be taught.

ii) the lesson core containing the fundamental information. Its purpose is mainly teaching and training.

iii) the lesson summary to offer the student a possibility of receiving a concise version of the lesson.

iv) the lesson review containing the main information of the lesson along with different sets of examples and exercises.

v) the quiz whose results will be taken into account in the process of deciding which kind of action has to be taken with respect to the student.

vi) the evaluation questionnaire whose purpose is to gather information about the opinion of the student regarding the lesson.

Items (i), (ii), (iii) and (iv) will always be present; (v) and (vi) may be present or not.

2.2. Method

2.2.1. General Approach

The course is offered to the student as a predetermined sequence of lessons. This restriction is justified by the limitation imposed by the size of the machine. Besides that, a sequential way of presenting a course has being used extensively either in automatic or conventional teaching systems. There is no experimental evidence to show the advantage of using a system that provides the student with a total freedom of choosing his own path through a course [4]. However, several ways of escaping this rigid framework are tried. The student may require at any point the summary or review of any previous lesson. He may also require the repetition of a lesson, granted that some restrictions are respected. At the end of a group of lessons there is one to which a quiz is associated. Within the range of such a group a student may repeat any lesson, but after taking the associated quiz he is not allowed to repeat any more lessons in this group. If a student fails a particular quiz the adviser of the course may provide remedial actions to the student and allow him to take another quiz. It is obvious that there is an upper bound for the number of times a student can repeat a lesson. Within a lesson the information units are also given in a predetermined sequence. A student is only permitted to ask for the repetition of the information unit he is working on. If the student interrupts a teaching section or the normal sequence of the course before finishing an information unit he has to repeat it at the time he returns to the teaching system or to the point he was in before. There are two levels of information in the teaching system: compulsory and optional. The student has the possibility of choosing some peripheral units associated to certain nuclei. Nuclei and quiz peripheral units are at the compulsory level. Peripheral units containing examples, exercises and references are at the compulsory or optional level. The ones containing additional explanations and hints are always at the optional level.

The compulsory information always follows the predetermined sequence. The optional information is given only if the student asks for it. There is a linked list of optional information associated to some compulsory information. The access to this list is possible only on the basis of its first element, which is reached from the compulsory information associated with it. A student may at any moment interrupt the optional information list but he cannot have random access to its elements.

In general, the information will be given in the following sequence: (i) nucleus; (ii) examples ; (iii) exercises; (vi) quiz; (v) evaluation questionnaire. In this compulsory sequence (iv) and (v) may be present or not.

The relation between compulsory and optional information is the following:

Information type	Optional information available
nucleus	additional explanation; reference
additional explanation	additional explanation; reference
compulsory example	example
optional example	example
compulsory exercise	exercise; reference; hint
optional exercise	exercise; reference; hint
compulsory reference	reference
optional reference	reference
hint	hint
quiz	-
answer	-

Exercises are associated with an anticipated answer list. Such a list tries to foresee different ways of answering the same question. The next informa-

tion to be given to the student depends on whether student's answer matches the anticipated answers or not. The student has the right of trying three times to find the correct answer. If he fails on the last try the system will supply the first anticipated answer. Besides, there is a fixed span of time the system waits for the student's answer. When the time runs out the right answer is supplied. The student may at any moment request the right answer. In any of those cases a student's error is listed. Hints are available as optional information. At the end of each information unit the results obtained from the compulsory exercises are evaluated. Several types of advice may be given depending on this evaluation, for instance, "repeat the information unit", "repeat the lesson", "return to lesson n", "ask for review (summary) of lesson n" and so on. At the end of each lesson such remedial actions are compulsory. The final performance of the student in a group of lessons is estimated by using only the results of the quizzes. The student is allowed to take a quiz only when he obtains a fixed rate of right answers in the exercises contained in that group of lessons. The remedial actions are always taken in this range. The final grade of the course depends on the grades obtained in the quizzes.

During the lessons, the student is asked to give his opinion about the course. Such opinion or preferences are designed to asses the student's reaction when facing a new way of learning. This is a very unsophisticated way of estimating his reaction. Obviously, the individual observation of his advisor is essential to any judgment about the course.

2.2.2. Specific Problems

The fact that the course is intended to be an introduction to programming gives rise to several problems specific to this subject.

The main challenge for any programming course comes from the difficulty of teaching how to translate the solution of any problem into an algorithmic form.
To surmount this challenge we use ideas stemming from structured programming approach [11], [12], [13]. Thus several lessons are dedicated to problem resolution. Examples in these lessons are given showing how to solve a problem using a systematic way of thinking. It is well known that it is enough to use three types of constructions: sequence of commands, conditional and loop constructions. How to use BASIC commands to simulate the mentioned constructions is explained in a stepwise manner. First it is decided which type of construction is adequate to solve the problem. In the sequel the solution is divided into segments to which the same method is applied. After several examples, problems are proposed to be solved by the student in the same way.

The process of problem solving is highly time consuming for the beginner. Consequently, when a problem is proposed, it is suggested that the students prepare the algorithmic form of solution off-line. The coding must be submitted to the system on-line. The program is run by the PBASIC processor. This is accomplished by means of a communication between this processor and BATS.

Other types of exercises are provided by BATS: (i) predicted PBASIC response - the student has to analyze a set of commands and predict the outcome; (ii) constructed PBASIC commands; (iii) several types of multiple choice exercises. Some lessons are intended for introducing the student to PBASIC commands. The main preoccupation here is to enable the student to apply the commands correctly being always aware of the necessity of presenting them in clear and transparent contexts.

2.2.3. Reports on student performance

The system affords several methods of gathering data about the student's performance. Out of this data the system produces reports of two kinds, one for the course adviser, describing the performance of a single student or a group of students, and the other for the course designers about the exercises themselves without taking into account who answered them. This second type of report will be necessary for estimating the difficulty of the several kinds of exercises. The results obtained from an analysis of these reports will be taken into account to implement new versions of the course.

3. TEACHING SYSTEM MONITOR

3.1. General Description

The teaching system monitor is an integrated set of routines designed to manage the teaching process, to handle the data-base, to create and update course files and to supply reports. Each one of these objectives defines a set of routines described as follows:

(i) LEARNING - responsible for the management of the learning process using the data-base and files that store information about the student's performance.

(ii) AUTHOR - responsible for creating, modificating, deleting and reordering the data-base. At any time the above operations on the data-base are called for the results of reports will be taken into account.

(iii) REPORT - responsible for supplying reports about groups of students, single students, besides general information about the course using all the course files. Such files store information about the student's performance and the course.

(iv) FILES - allows one to create, update, transfer and retrieve disk files related to the teaching system. These routines can be considered as a set of auxiliary routines inasmuch as they are used by the other sets.

There are three modes of communication between the user and the teaching system monitor:

(i) student-mode, according to which only the LEARNING set will be accessible;

(ii) teacher-mode, according to which the four sets of routines will be accessible. This mode is only available to persons authorized to modify the course;

(iii) report-mode, according to which only the sets LEARNING and REPORT are accessible.This mode envisages primarily the course advisers.

3.2. Learning Set

Its main function is information units retrieval being the main piece in the student-system interaction. In addition it is responsible for gathering information about the course and about the student's performance.

3.2.1. Communication

There are two levels of communication:

(i) the hardware level, by means of printing terminals;

(ii) the software level, by means of a command language, which is defined below.

3.2.2. Command language

The command language is designed with the purpose of implementing the part of the method, described in 2.2., referring to the tools available to the student allowing him to modify the predetermined sequence of the course.

It includes commands appropriated for obtaining:

- next compulsory information;
- additional explanation;
- optional example;
- optional exercise;
- correct answer;
- hint;
- optional reference;
- repetition of the n-th lesson;
- end of repetition;
- repetition of the present information unit;
- review of the n-th lesson;
- end of review;
- summary of n-th lesson;
- end of session.

A command recognizer is also a part of the LEARNING set the function of which is to activate the routines required by the execution of each command.

3.2.3. Routines

The main routines of the LEARNING set are:

(i) beginning of a session;

(ii) answering analysis;

(iii) retrieving of nucleus and peripheral information.

(iv) supplying the text associated to a particular nucleus or peripheral information on the terminal;

(v) information updating of student's performance during a session;

(vi) communicating with the PBASIC processor;

(vii) evaluating the student's performance in a lesson.

3.3. Author Set

The main purpose of this set consists in the manipulation of the data base. It can be considered as a sequence of routines that permits mainly the entry of new inputs and the deletion of old information units.

The communication is made by means of routine calls.

3.3.1. Routines

The main routines of the AUTHOR set are:

(i) input of information units;

(ii) updating information about available disk storage.

3.3. Report Set

The routines in this set are designed to manipulate the information gathered by the LEARNING set. It will allow for getting reports derived from further analysis of the data about the course and about the student.

3.3.1. Communication

Communication is made by means of a command language designed to enable the user to ask for tables reporting:

- the student progress during the course;

- the progress of a group of students;

- the frequency of requirements for optional

information in a group of lessons;

- the frequency of trials, mistakes and correct answers to a group of exercises;

- statistics about how often and in which ways the course has been on the run.

A command recognizer is provided, which in particular, activates routines for executing each command.

4. CONCLUSION

The authors believe that the system described is a reasonable attempt to solve the problem of designing a single course, able to be adaptable to users of different educational backgrounds. This problem is a real one because to design different courses, on the same subject, for different homogeneous groups of students is economically unsatisfactory. In addition, to circumvent the economic problem the optional set of information of the system provides a certain level of flexibility without burdening the course with complex implementation problems. The size of the machine for which the course is designed limits any pretention of a highly sophisticated automatic teaching system. This machine is being developed at the University of São Paulo, under the responsibility of the 'Laboratorio de Sistemas Digitais (LSD)'. Hopefully, a prototype of it will be ready by March 1975. Beginning then tests of the system presented in this paper will be performed by means of two user terminals. The testing sample will be composed of secondary school children. Before then, however, tests will be conducted using a simulator of the machine, developed as part of the constract CONSUPULC /GUARANYS and which is run on a IBM 370 /165, located at the Rio Datacentro (RDC), Ponti ficia Universidade Católica.

The schedule of this current project establishes September 1975 as the deadline for its completion. The possibility of large scale utilization of the system herein described will depend on the economic marketing potentiality of the machine being developed. Nevertheless, due to the attractiveness of the system which is completely inde pendent of the particular machine, it is highly probable that it will be compatible with other minicomputers which will be utilized all over Brazil in a short span of time.

5. REFERENCES

(1) J. Schultz, J. Lenahan, C. Clatur - "*NOET-I System Manual*", Synnoetics Laboratory, Wiscon sin - University (1969).

(2) J. Friend, R.C. Atkinsons, "*Computer Assisted Instruction in Programming: AID*", Technical Report nº 164, Psychology & Education Series, IMSSS, Standord University (1971).

(3) J. Friend, J.D. Fletcher, R.C. Atkinsons, "*Student Performance in Computer-Assisted Instruction*", Technical Report No. 184, Psychology & Education Series, IMSSS, Stanford University (1972).

(4) M.H. Beard, P.V. Lorton, B.W. Searle, R.C. Atkinson, "*Comparison of Student Performance under Three Lesson-Selection Strategies in Computer-Assisted Instruction*", Technical Report No. 222, Psychology and Education Series, IMSSS, Stanford University.(1973)

(5) R.R. Fenichel, G. Weizenbaum, J.C.Yochelson, "*A Program to Teach Programming*", CACM, Vol. 13, Number 3 (1970).

(6) A.M. Noble, "*Presenter: An Author Language for Computer-Assisted Instruction*", Bionics Research Reports No. 10, Bionics Research Laboratory School of Artificial Intelligence University of Edinburgh (1973).

(7) J. Carbonell., "*AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction*", IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No.4 (1970)

(8) R.J. Seidel, "*Is CAI Cost-effective? The Right Question at the Wrong Time*", Educational Technology (1969).

(9) C.J. Lucena, "*Computer Education: A New tool for Social and Economic Development*", Invited Paper, Jerusalem Conference on Information Technology (1974).

(10) C.J. Lucena, L. Martins: "*Guidelines for Organizing the Computing Activities of a University in a Developing Country*". Proceedings of the "Primeras Jornadas Latino-Americanas de Computacion", Buenos Aires (1971).

(11) H.D. Mills, "*Mathematical Foundations for Structured Programming*", Federal System Division, IBM (1972).

(12) N. Wirth, "*Systematic Programming: An introduction*", Prentice-Hall, INC, Englewood Cliffs, New Jersey (1973).

(13) O.J. Dahl, E.W. Dykstra, C.A.R. Hoare, "*Structured Programming*", Academic Press, London and New York (1972).

NOTE: This work is supported by the Brazilian Federal Government through a Special Team of Work (Grupo Especial de Trabalho, GTE) created by the decree number 68.267 in February 18, 1971. It is part of a project for development of basic and application software for a Brazilian minicomputer (Projeto CONSULPUC/GUARANYS). The authors are part of a team involved in the design

and implementation of a conversational system for teaching programming and processing programs using a Portuguese version of a modified BASIC language. The other professors who are members of this particular team are: Albrecht von Plehwe, Dr.rer.mat., University of Freiburg, Germany; Emmanuel Passos, M.Sc.; PUC-RJ, Renaud Leenhardt, M.Sc. PUC-RJ.

ACKNOWLEDGMENT

We want to thank Dr. Wilfried Brauer of University of Hamburg for his reading and criticizing a first version of this paper. His help improved very much the clarity of the presentation.