ED 089 748 IR 000 460

AUTHOR Bellardinelli, Mario
TITLE Criteria and Educational Tools in the Training of Programmers.
PUB DATE 19 Nov 71
NOTE 11p.; Paper presented at the International Conference on Training for Information Work (Rome, Italy, November 15-19, 1971)

EDRS PRICE MF-$0.75 HC-$1.50 PLUS POSTAGE
DESCRIPTORS Computer Assisted Instruction; Computer Programs; *Computer Science Education; *Educational Programs; Electronic Data Processing; Information Processing; *Programers; *Programing; Programing Languages; Speeches; *Teaching Methods; Teaching Procedures; Teaching Techniques
IDENTIFIERS Analytical Method; Global Method

ABSTRACT

The training of computer programers could be more efficient and more effective if two major changes were implemented. First, the analytical method of teaching programing and programing languages should be replaced by the global method. The former provides introductions to information processing and programing and then teaches the programing language; it proceeds from the particular to the general and maintains artificial distinctions among related components, thereby forcing the student to deal with abstract concepts and to delay practical experience until the end of his training. On the other hand, the global method provides only brief introductions to particular aspects of programing and plunges students quickly into writing their own programs. It is more effective because it gives them actual experience, thereby paralleling the manner in which natural languages are learned. Secondly, the education of computer programers would be greatly enhanced if the students were allowed to use the computer itself more often; such hands-on experience stimulates interest, reinforces factual learning, and promotes creative development. (PB)

CRITERIA AND EDUCATIONAL TOOLS IN THE TRAINING OF PROGRAMMERS

Premise

In discussing training in informatics the accent is very often placed on the content rather than the methods; on that wich is taught rather than the way in which it is taught.
It is said that the EDP specialist is a professional of a high technical level and that he who trains these specialists must be capable of transmitting this high degree of technical competence. The didactic discussion while useful is certainly a secondary factor.
Personally I am very far from convinced of this point of view. Indeed I am sure that many of the difficulties encountered in the training of new EDP professionals lies in the lack of good teachers. There is no shortage of persons who "know", but there are very few who are capable of teaching what they know.
It is not merely a question of content but also of methods. Indeed at the level of initial training what counts is not so much knowing but knowing how to teach.

I. The analytical and the global methods

The global method which has so clearly demonstrated its excellence in the teaching of natural languages, is not so well considered in the teaching of programming languages in the informatics fields.
Indeed it can be said that the traditional analytical method still goes unchallenged.
Let us, for example, have a look at the way in which a programming course is normally organized and what are the didactic criteria behind its development. There are three fundamental and clearly distinct subjects to be taught. Hence the course is structured according to the following three modules:

1. introduction to EDP, covering general concepts of automatic data processing: information theory, elements of Boolean algebra, numeration systems, the structure and functional logic of the computer, the organization of data on supports, etc.;

2. introduction to programming, concentrating essentially on flow-charting;

3. programming language.

Two characteristic aspects are to be noted in this type of structure. The first is the fact that pupil comes into contact with programming (module 2) only after a full survey of the introductory concepts (module 1), which up to this point necessarily remain very abstract. The second is that the flowchart and the

./.

language which are the inseparable elements of programming, are taught at separate times. In other words, the pupil must have completed the flowcharting aspect prior to any instruction in the programming language, which will enable him to create an individual program.

Let us now see how the teaching of a programming language is tackled. We shall suppose that COBOL (Common Business-Oriented Language) is to be taught.

A start is made with the general rules: programming module, admitted and unadmitted symbols, the lenght of names, reserved words, programmer's words, punctuation, etc. Great emphasis is also placed on definitions: for example the concepts of division, section, paragraph, sentence or phrase or the concepts of file, records, compounded and elementary data.

The survey of the rules and general concepts is followed by second stage in which the analysis of the elements of the language, division by division and section by section is studied. Since a programming language, unlike a natural one, is a strictly codified language, it is essential that the format of every phrase and the meaning of each item is clarified extremely well.

Stress is also laid on the analysis of programming rules (for instance, the many rules governing the use of symbols in the "PICTURE" and the complex case history of the verb "MOVE").

Once all the elements of the language have been described, through a difficult work of synthesis, these are then presented in a general framework and the pupils at last reach the stage of realizing their first program.

A certain pointer which tells us the degree of application of this method, is given when the pupil is capable of writing unaided his

./.

first complete program: the longer the period that elaspes between the beginning of the course and the latter stage, the more radical is the application of analytical method.
The global method, on the contrary, is based on an entirely different approach. Its objective is to bring the pupil as soon as possible to the writing of his own program.
The course begins with a very general presentation of the structure and the functional scheme of a computer and touches very broadly on the organization of data on supports. For instance, it can be said that half a day is sufficient to impart this first introductory survey..
The pupils are then introduced immediately to the flowchart. The ideal is achieved when the pupil draws-up his own flowchart from the very first day. Naturally at this point the entire subject of flowcharting cannot be dealt with, but when the pupil has acquired a certain familiarity with very simple flowcharts, he is immediately introduced to the programming language. The more extensive study of the introductory concepts and flowcharting techniques will take place gradually together with progress in the knowledge of the language.
The approach to the language is a follows: a start is made by the writing of a simple, brief but complete program, in such a way that the pupil immediately acquires an overall understanding, even if fundamentally intuitive, of the language.
This first program will be written by the teacher together with the pupils; that is to say, in practice, the teacher will suggest and explain it to the pupils.
At this point a natural objection arises: how is it possible to write an entire program without having illustrated a single language

./.

rule?

The answer is very simple: let us recall how we learnt our native language; by repeating words and phrases that were spoken to us, intuitively associating the words with the things and certainly not by studying grammatical rules. Let us also recall the fact that this is the best way of teaching adults a second language. Why then should this not be a valid way of learning a programming language?

Hence the essential point of the global methods is this: we first of all become familiar with the whole and then the individual parts are analysed, first the language is learned and then it is perfected by studying the rules.

If this process is reversed, if a beginning is made on the individual parts to arrive afterwards at the whole, if the rules are studied before using the language, the pupils are forced to make a great mental effort to apply concepts concerning which they do not know the final meaning.

It is, on the contrary, important that the pupil should immediately achieve his own program, in such a way as really to grasp the global structure of the language. It is of no importance if this first effort of his contains some error of punctuation or does not use the programming module in an entirely appropriate way. Starting from this overall comprehension, it will subsequently be possible to study the single elements in depth without ever loosing sight of the whole.

Every time the pupil studies a new particular he will always know how to set it within the general framework.

The global and the analytical methods, therefore start from different didactic principles and develop along entirely separate and

./.

distinct lines.

Let us attempt a comparative synthesis.

The analytical method is based on the following principles:

1. practically integrated but logically distinct elements of a single technique (for example, flowcharting and the language, the single divisions of the language, the single sections within the sphere of a division), for motives of clarity and completeness are studied separately.

2. From an in-depth analysis of the individual elements the pupil is brought to an overall comprehension.

The principles of the global method, on the other hand, are:

1. The various elements of the program, even if they are logically distinct are taught as a whole.

2. Learning is realised through a spiral process composed of three fundamental phases: the intuitive understanding of the whole, the analysis of the individual parts, the rational re-comprehension of the individual parts in the whole.

./.

II. The use of the computer in the learning process

Independently of the method adopted, the opinion that it is possible to teach programming without any direct access to a computer is certanly true.
Computer logic, in fact, can neither be departed from nor does it present ambiguities. By starting from a correct flowchart and a precise knowledge of the working of every individual instruction, it is certainly possible to write a perfectly correct program or, at least, all the instruments are available for carrying out a careful verification at purely theoretical level.
When this check has been made, the program must work; if this does not take place, it is due to an accidental factor, some error in transcription, or it can mean that the checking was not sufficiently strict.
In the first case the computer itself capable of indicating the error and suggesting the correction, in the second case, on the other hand, this is possible only if the error is of a formal kind; if, on the contrary, there is a logical error more careful and strict re-checking is needed.
In either case, a good teacher, with the aid of good text books, could teach a certain programming system without ever introducing the pupil to direct contact with the computer. In the three successive stages: lesson, drill, correction, the pupil should be capable first of learning the programming rules, then experimenting with their application and lastly checking the correctness of his work and his degree of learning. When a certain program has been corrected by teacher together with all the other pupils it should no longer be thought that it could contain errors. The teacher's

./.

correction will therefore have the same value as a test in the computer.

It cannot be denied that all this is true and that the method generally adopted in the various programming schools is very close to the one described here. Contact with the computer, in fact, both for reasons of availability, and didactic difficulties, is very often reduced to the minimum.

However such a method creates notable difficulties in the learning process. The lack of a continuous interaction with the computer gives the pupil the impression that instead of learning how to use a machine he is working on mere abstractions. He is first frustrated in his natural desire for a practical and effective check of his work, he is less certain of what he learns and the stimulus to learn new ideas is weakened. The use of a computer during the learning process has, above all, a very strong psychological effect.

I consider that three fundamental didactic objectives can be achieved through direct contact with the computer: to encourage an interest in learning, to strengthen facts already and provide the opportunity of carrying out creative work.

2.1. The computer as a promotor of interest

In the teaching of programming the introductory part is always very difficult and is justifiably given very special attention. I have experimented with the traditional method based on the two distinct modules described above: introduction to EDP and flowcharting. I found that this posed considerable didactic difficulties because the first part ~~was so~~ abstract and unstimulating.

./.

Turning to the global approach, I decided to start the course with a practical demonstration of work carried out by a computer. A very simple and easily understandable procedure was chosen (invoicing with the updating of warehouse stocks), realized on a small card computer system. Each pupil was able to see the performance of all the individual operations and was also able to take the part in the carrying out of the programme by introducing data or asking for certain results.

At the end of the demonstration a very great interesting in knowing the structure and the working of the computer has always been shown. All the introductory concepts could be clarified in a most effective and very quick way because it was always possible to refer to things already known and not mere abstractions. But after this first experience the pupils immediately begin to want to have a different contact with computer; no longer in the sense of seeing and understanding what it can do, but in the sense of making it do something. That is to say, the pupil wants to see if he is capable of commanding the machine. For this he needs to learn how to program as soon as possible. The analytical method tends to defer too long this second and most significant contact with the computer. The global method, on the contrary, enables this objectives to be reached quickly and easily.

2.2. The computer as a verification instrument

The pupil that has been capable of writing his first program, the instrument for the domination of the machine, wants to test its effective validity.

./.

As long has he has been unable to effect a test of this kind, he cannot proceed to other programs with adequate certainty. It is extremely important the first test does not become a frustrating factor, due to banal difficulties of an organizational character or for operational errors. At the outset it is wise to employ very simple exercises, with pre-arranged check data. Where possible, also as regards the program, it is very useful to employ pre-punched cards. The compilers' diagnosis unfortunately is neither always very clear nor a help in locating an error. This is one of the main difficulties in the way of an easy dialogue with the computer. When the pupil has got through the first test and has acquired the fundamental elements of the programming language, he is ready for more independent learning.

2.3. The computer as a means of creative activity

At this point the computer no longer serves to check the accuracy of certain ideas acquired theoretically, but for performing a creative activity. This is the moment when the pupil passes from the lesson-drill-testing didactic phase to a much more autonomous phase. It is therefore a question of assigning to each pupil, or rather groups of pupils an exercise of a certain entity, which involves not only the writing of the program but the carrying through of a complete procedure, albeit it simple.

In this way the pupil is able to grasp not only the working of an individual program but the concatenation of several programs and must necessarily arrive at a wide use of the system's software. It is of no importance if all the instru-

./.

ments that he must use have been previously explained at theoretical level. The purpose of the exercise instruments, which he discovers for himself and learns to use. The teacher at this point does not disappear but assumes the role of suggester and consultant.