THE PROGRAMMER IN A CHANGING WORLD

Willis H. Ware

September 1963

P-2793

THE PROGRAMMER IN A CHANGING WORLD

Willis H. Ware*

The RAND Corporation, Santa Monica, California

As modern computers have evolved to their present form, the computer programmer in one sense or another always stands between the person who has a problem requiring solution and the tool (the computer) which can solve that problem. The computer programmer is analogous to the translator of natural languages. His job is to accept ideas expressed in the language of the problem and to "translate" those ideas into the markedly different language in which a machine converses with itself. As part of his job, the programmer usually assists in formulating the problem. Because of the language disparity between the problem and the machine, the programming field has grown extensively over the past ten years, and at present there is considerable discussion about whether we will be able to educate and train enough programmers to keep up.

---

*Any views expressed in this Paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

This Paper was prepared to be submitted for publication in Electronic News.

Fifteen years ago the user of a computer was willing to train himself in the language of the machine. More recently, in an effort to expedite the task of programming, artificial languages have been conceived and are well-known today as the so-called higher languages--FORTRAN, ALGOL, COBOL, and others. The net effect of the development of such languages has been to conceal from the user the primitive nature of machine language and machine operations. One consequence of higher language development has been to broaden the field of application of digital computers, making them available to a wider class of user. Even so, many scientists and engineers of today avoid the use of computers in their research because, first, they refuse to learn one of the synthetic languages, and, second, they find it extremely difficult to communicate the problem to a third party (the programmer) in enough detail to have him understand it sufficiently well to formulate a solution. Therefore, much work which really ought to be done by computer is not. As the interface between user and computer more and more closely approaches the natural language and mode of behavior of the human, the requirement for the professional programmer as an intermediary diminishes. It may be that the professional programmer as we know him today

will have to find a significantly different role to survive. Is it conceivable that the programmer will disappear from the problem-solving process?

Research in progress today supports such a thesis, and there is a continued and active development of new and more sophisticated synthetic languages. Another, perhaps more promising, research goal is to present the machine to the user in a way natural to him. One such activity is the so-called on-line time-shared mode of computer use. "On-line" implies that the user is directly coupled to his machine and that the two exchange information in a conversational manner. "Time-shared" implies that many users are coupled to the machine and that the machine is time-sequentially servicing each user, much in the same way that telephone subscribers share central office equipment. To the user, the machine appears to be immediately available, although because of its speed, it may have performed many other tasks since it last serviced a given user. Some on-line time-shared systems present the machine to the user through one of the synthetic languages; other time-shared on-line systems present the machine to him in a language and in a context for which he is already trained--for example, the language and symbology of algebra.

The end goal of on-line time-shared research is to make the machine appear as a tool which understands the same language and the same problem context as its user. Because of the intimate and prompt interaction between them, the machine is available on demand and will converse with its user in his language. The machine will help him formulate the problem; it is tolerant of his rate of thinking because of the time-shared mode; and, moreover, it does not charge him for services rendered except when actually working for him.

Where does the present-day professional programmer fit in such a future computing world? The programmer will continue to be needed for the language development implied by our desire to change the face which a machine presents; he will be needed for unusually difficult problems which tax machine capability; he will be needed in various phases of computer research. The continued growth of the computer field may be sufficient to absorb our present population of programmers in such roles. On the other hand, it may be that our ability to present a machine to its user on his terms will improve so rapidly that programming as a specialty will become a relatively small professional field.

In the light of current research, it seems certain that the role of the programmer will be a changing one. The computer may obsolete the very professional specialty which it fostered fifteen years ago. We may or may not have the large future demand presently forecast for programmers. In the long run, the demand may be much less than we now predict. Eventually it may turn out that the word "programmer" will be used only to describe a secondary skill that people from a wide variety of professional disciplines will have--the skill to use and exploit the digital computer as a problem-solving tool as one works in his field of primary training.