

COMPUTER TRAINING AND EDUCATION:

The Picture in 1962

F. J. Gruenberger

July, 1962

P-2597

COMPUTER TRAINING AND EDUCATION:

The Picture in 1962

F. J. Gruenberger*

The RAND Corporation, Santa Monica, California

I want to address myself to the problems of education and training in the world of computers. It is a big world, and getting bigger. For any measure applied to it, the second derivative is positive. For example, taking top speeds as a rough index of computing power, the total computing capability of the U. S. will at least double in the next twelve months. This doubling will be due almost entirely to mass production--a production breakthrough leading to more speed per dollar would probably increase the rate of growth.

It is dangerous to extrapolate from such figures, but it is fascinating to conjecture on the long range implications

*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 is to be presented at the Northwest Computing Association Annual Meeting, Seattle, August 9, 1962.

of a rate of growth which is so high. Either the present users of computers are going to step up their consumption by large factors in the next decade, or the number of users will greatly increase, or both. My personal inference is that the number of persons who have regular access to a million executed instructions is likely to increase by a factor of a thousand or so within ten years. All of these people must be at least trained, if not educated. The question is, what's so different about this educational problem from the problem of training and educating masses of people in any other skill or technology?

Part of the answer lies in the differences between the computing world and all others. Let me list some of the things that are different about the computing world.

1. I have already touched on the extremely high growth rate, which is probably exceeded by no other technology.

2. The computing fraternity exhibits a degree of cooperation among its users which is unique. We are all familiar with the cooperative programming groups that have sprung up around machine types. Compared to other fields, much less information is kept proprietary in computer work. In fact, just the converse is the norm; individual users race to fill central libraries with better versions of utility routines, interpreters, and even new FORTRAN compilers.

This cooperation extends to machine time, too. Every prudent user arranges for backup of his machines on a reciprocal basis.

It is difficult to think of corresponding examples of either type of cooperation in any other industry.

3. Every computer user can obtain, on delivery of his late-model machine, a detailed description of just how obsolete it is. This is due partly to the long lead time involved in installing a machine, coupled with the dynamic improvement in hardware and software. We always have dangled in front of us how much better off we will be in two years. Among both users and manufacturers, the complacent person is more than half dead. The point here is that our industry tends to be livelier than others.

Consider the case of a plant manager who has (a) just replaced his truck fleet with the latest model trucks, and (b) just replaced his old computer with the latest model. In the one case he can feel reasonably assured that a portion of his plant is now operating at optimum efficiency; in the other case, he is apt to have doubts. For one thing, he knows of better equipment already available (it is two years since the computer was ordered); for another, he has good reason to suspect that his computer people don't know how to get maximum use out of the machine.

4. Since computers are the ultimate in precision instruments, they can be described with precision; moreover,

the effectiveness of the hardware/software combination can frequently be estimated for any job or mix of jobs. This is not to say that we see any evidence that this is done; merely that it is possible to a greater degree in computing than in other industries.

5. The computer offers vast power to an individual to a degree not found in any other technology. An individual cannot operate a B-52 by himself; if he could, he is seldom in a position to make effective use of 15 seconds' worth of its capability. But John Q. Citizen can buy--anywhere--that million executed instructions. The going rate is about one dollar, and falling. Computing power seems to be a fluid commodity available to all.

If computing is different from other technologies (and the list of differences could be extended), then I think it follows that its training problems differ also, if only in size. The manufacturers and users are already aware of having an acute training and education problem on their hands; I suspect that the problem is growing a great deal faster than any possible solutions so far considered. I am told that the number of IBM 1401's on order approximates the total number of computers now in the world. Considering that there are lots of other machine types on order, the implication to me is that the real problems in training and education are still ahead of us. The movement of computers

into the secondary schools, for example, is just beginning. That movement is likely to have a steep wave-front and hence a significant shock-wave.

Is anything being done to solve the problems before they become really acute? Of course something is being done. But I contend that the efforts are woefully small, in view of the magnitude of the pending task. Those who need computing knowledge will get it (or what appears to be a fair facsimile). They won't know what they aren't getting, and they will proceed to use computers anyway. I think we are proposing to bail out the ship with a teaspoon. Perhaps this is a better metaphor: we are gearing up to change the tire while going down the freeway at 70 mph. There ought to be a better way, and since we computing people keep telling each other how smart we are (an added item to that list of things that are different about computing), we ought to look for it. We may find it by accident, but that's a rather slim hope.

Almost exactly three years ago I addressed the Northwest Computing Association on the subject of computing in the universities. At that time, I divided up the domain of the university computing center into four parts: within-campus computing service; outside (i.e., contract) work; training; and research. I suggested then that one might be hard pressed to name more than six universities that were doing anything significant in the last two categories

(training and research). The absolute number is higher today (perhaps as many as ten), but the percentage hasn't changed much. It's probably gone down.

What little is being done centers around industry. The computer manufacturers are in the forefront; they must train their own people as well as many of the users. Many industrial users, of course, are up to their ears in training new people on the job.

WHAT IS THE PROBLEM?

What is so difficult about training and/or educating masses of people in computer technology? It's hard to say, but I can ask a great many more questions than I can answer.

The first question anyone has to face is this: what is it we want to teach? Do we want to teach how to get answers? Or do we want to teach how to solve problems? Or do we want to teach computing? Or computers? Or numerical analysis? Or what? These are all different subjects. Do we want to teach them all, in a one-semester course?

Some of the answers I hear, when I raise these questions, strike me as illogical. Take, for example, the person who wants to cover many of the fields listed above, and who proposes to do it by starting with a magic language (FORTRAN, NELIAC, MAD, etc.) "so the students will have something to work with right away." It sounds so plausible, especially if you say it fast.

But the person who says this didn't learn by that route himself. Moreover, we can see plenty of examples of where that approach failed--the graduate seems to be somewhat lacking in the basics. Now, I am the first to attack the line of reasoning that says "we did it this way, and it worked, and therefore it must be the correct way (or a correct way)." On the other hand, I'm inclined to defend the reasoning that says "we did it this way and it didn't work, and therefore it's probably not a correct way."

Let me raise some more questions. Will the students have access to a computer while they learn? Will they be exposed to a vast number of facts and concepts rapidly in the hope that a fair percentage will rub off? Will they "learn while doing?" (This is sometimes a euphemism for "we can't wait to do a good job, since we're desperate to get today's jobs done.") Will they have a crash course, perhaps followed by apprentice training at the feet of some expert?

I'm merely suggesting that there is room for discussion as to how the job should be done. All too often we seem to reach what seem to be at the time quite plausible answers by rationalization. For example, in the early days of the game there were many courses in computing given at places which had no computer. A bit later, courses built around a hypothetical machine became popular. And for a time it was argued vehemently that access to a wired calculator

would suffice. I submit that in all three of these situations, the real learning atmosphere began with the subsequent exposure to a real computer, and we had simply fooled ourselves. To be sure, we were forced into this rationalization by the lack of available equipment at the time.

Even today, the argument is advanced seriously that it is vital, in teaching computing, to advance to symbolic assembly language as soon as possible. This is fine--on a binary machine. The argument weakens somewhat when one postulates a computer for which the (decimal) machine language is acceptable. Notice that I am not speaking against symbolic assemblers (or even high level narrative compilers)--I am simply questioning the reasoning by which we conclude what ought to be done.

Incidentally, we do know one fine way to teach computing, but, unfortunately, the course takes 20 years. The course begins with a 602A and 405, and works up from there. I may be the last man to take this course, and I'm only in the 14th year of it now.

If the problem is large and complex (and I think it is) and little has been done to attack it systematically (and I believe that's true), then I think we should be on guard against an influx of charlatans. (Logically, we must entertain the thought that you are now listening to one. One thing in my favor, though, is that I don't claim to have

the answers; charlatans always have answers for everything.)

For example, I recently examined a "model" curriculum for a junior college course in data processing. In the 15-weeks' course, the students were to be studying punched card sorters in the third week and monitor systems in the fifth week.

In another case, a course outline stated that "... reference to number systems other than digital" would not be included. That may have been a typo; I don't know. The point is that I hardly believe we are yet ready to freeze a curriculum, particularly at the secondary school level. But in the absence of any massive attack on the problems involved in mass training, the quacks will take over, or try to. You might reflect on what they've already done with "teaching machines."

IS THERE HOPE?

Sure there is. The best thing we have going for us is the computer itself. It is what makes the problem unique; it also helps solve it. A computer is a superb teacher, all by itself. If I had to start from scratch, I'd rather have a computer and no teacher than the best teacher and no computer. In this tradeoff, the computer is constant, so given any computer and even a poor teacher, we're not in too bad shape.

The 5% law is also on our side. This is the principle that says that after you've learned one machine, the next

one requires only 5% more effort. A corollary to this is that while learning that first machine you were really spending most of your time learning fundamentals and very little time actually learning the machine.

Better equipment keeps appearing on the market. One has only to compare today's machines with the best of a few years back to realize the tremendous increase in the computer-per-buck ratio. This ratio is bound to increase even more. To put it another way, computing power is certain to be extended to more and more people--which gives us more chance to explore ways to teach the subject. Computing-on-the-end-of-a-wire is almost upon us. The new equipment is not only better in the sense of more for the dollar, but is easier to use.

There is a flood of books coming on the market, and an impressive catalog of films available. I would call particular attention to the series of six half-hour films to be shown in November on the educational TV network.

I would like to close with a brief mention of a complex and touchy problem--one that is closely tied to educational problems. Except for their faculties, the only things that schools seem to get at discount are computers and typewriters. The "educational discount" has had a long history, amounting almost to a tradition. There have been instances in which it has been allegedly used as a device to suppress competition. Perhaps it is now time to consider abandoning it;

that is, for the industry as a whole to renounce it.

I am suggesting that the educational discount on computers has served its purpose for all concerned and that at some time in the near future it be abandoned.

