The Age of Computing: A Personal Memoir
Author(s): N. Metropolis
Source: *Daedalus*, Vol. 121, No. 1, A New Era in Computation (Winter, 1992), pp. 119-130
Published by: The MIT Press on behalf of American Academy of Arts & Sciences
Stable URL: http://www.jstor.org/stable/20025423
Accessed: 30/08/2010 18:46

N. Metropolis

The Age of Computing: A Personal Memoir

IN THE HISTORY OF MODERN TECHNOLOGY, computer science must figure as an extraordinary chapter, and not only because of the remarkable speed of its development. It is unfortunate, however, that the word *science* has been widely used to designate enterprises that more properly belong to the domain of engineering. "Computer science" is a glaring misnomer, as are "information science," "communication science," and other questionable "sciences." The awe and respect which science enjoys and which engineering is denied is inexplicable, at least to one who sees the situation from the other side.

The popular image of science has changed little since it was invented by Jules Verne and H. G. Wells. *Science* represents the search for knowledge, the conquest over nature, the discovery of some very few fundamental laws that will free mankind from worry and toil; this is as true today as it was at the turn of the century.

The word *engineering,* however, carries less exciting connotations. I recall a pleasant evening at the house of the American Academy of Arts and Sciences. In the great hall conversation and gossip flowed freely in anticipation of a brilliant lecture. A distinguished lady, a pillar of Cambridge society, was expressing her admiration for Professor S. She extolled his discoveries and his brilliant insights. "And what department at MIT does he belong to?" she finally asked, by way of indicating that our brief exchange was coming to an end. "Mechanical engineering," I answered. A look of horror crossed the lady's face. "Why, I thought he was a scientist!" she blurted out

N. Metropolis is Senior Fellow Emeritus at Los Alamos National Laboratory.

before she could cover up her gaffe. I saw in her eyes the image of a man in a dirty gray frock, a pair of pliers in his greasy hands, bent over some Chaplinesque contraption of gears and pulleys.

But, contrary to the lady's prejudices about the engineering profession, the fact is that quite some time ago the tables were turned between theory and applications in the physical sciences. Since World War II the discoveries that have changed the world were not made so much in lofty halls of theoretical physics as in the less-noticed labs of engineering and experimental physics. The roles of pure and applied science have been reversed; they are no longer what they were in the golden age of physics, in the age of Einstein, Schrödinger, Fermi, and Dirac. Readers of *Scientific American,* nourished on the Wellsian image of science, will recoil from even entertaining the idea that the age of physical "principles" may be over. The laws of Newtonian mechanics, quantum mechanics, and quantum electrodynamics were the last in a long and noble line that appears to have somewhat dried up in the last fifty years. As experimental devices (especially measuring devices) are becoming infinitely more precise and reliable, the wealth and sheer mass of new and baffling raw data collected by experiment greatly exceeds the power of human reason to explain them. Physical theory has failed in recent decades to provide a theoretical underpinning for a world which increasingly appears as the work of some seemingly mischievous demiurge. The failure of reason to explain fact is also apparent in the life sciences, where "theories" (of the kind that physics has led us to expect) do not exist; many are doubtful that this kind of scientific explanation will ever be successful in explaining the secrets of life.

Historians of science have always had a soft spot for the history of theoretical physics. The great theoretical advances of this century—relativity and quantum mechanics—have been documented in fascinating historical accounts that have captivated the mind of the cultivated public. There are no comparable studies of the relations between science and engineering. Breaking with the tradition of the *Fachidiot,* theoretical physicists have bestowed their romantic autobiographies on the world, portraying themselves as the high priests of a reigning cult.

By their less than wholly objective accounts of the development of physics, historians have conspired to propagate the myth of science as being essentially theoretical physics. Though the myth no longer described scientific reality fifty years ago, historians pretended that all was well, that nothing had changed since the old heroic days of Einstein and his generation. There were a few dissenters, such as the late Stanislaw Ulam who used to make himself obnoxious by proclaiming that Enrico Fermi was "the last physicist." He and others who proclaimed such a possibility were prudently ignored. Physicists did what they could to keep the myth alive. With impeccable chutzpah, they went on promulgating new "laws of nature" and carefully imitated their masters of another age. With dismaying inevitability, many of these latter-day "laws" have been exposed as quasi-mathematical embellishments, devoid of great physical or scientific significance.

Historians of science have seen fit to ignore the history of the great discoveries in applied physics, engineering, and computer science, where real scientific progress is nowadays to be found. Computer science in particular has changed and continues to change the face of the world more thoroughly and more drastically than did any of the great discoveries in theoretical physics. The prejudices of the academic world have stood in the way of the historian. One wonders whether a historian of contemporary engineering could get a teaching job at a respectable university. For some reason, histories of long obsolete discoveries, such as the steam engine, are acceptable in academia: dozens of such histories have been written and, undoubtedly, dozens more will be written now that the field has become an established one. However, a history of the transistor is still beyond bounds (no such history has even been attempted, to the best of my knowledge). Thanks to the joint public relations efforts of historians and physicists, the white mane of Albert Einstein remains the unquestioned symbol of genius. It is scandalous, however, that virtually no cultivated person has ever heard of John Bardeen, whose discoveries may have revolutionized the world at least as much as Einstein's. Bardeen's midwestern background and his having taught in Urbana, Illinois, were fatal flaws that prevented his ever being recognized.

It would be tempting to conclude, after an inspection of empty library shelves, that the absence of engineering histories, recounting major discoveries, is due in part to the difficulty of gaining access to essential facts. Practical discoveries are not as easily traceable to research papers as are theoretical discoveries. Such a conclusion, however, would not be warranted. The development of any discovery of even the slightest practical value is generally thoroughly documented in reports, replete with names, careful attribution given to who did what, when, with the funding sources and dollar amounts given. Unfortunately, access to such documents, at present, is severely restricted by bureaucratic barriers deliberately placed in the way of those who have no "need to know." Only the top managers of major business corporations, certain officials of the federal government, and, in times past, selected members of the KGB in the late Soviet Union were privileged to peruse such documents.

In our rapidly changing political and international climate it is possible that such restrictions will soon be lifted. When that happens, it will be inexcusable for a historian of science to neglect the history of the great technological discoveries of our time, including, obviously, the history of computer science.

In offering some random remarks on the possibilities of such a history in the future, I would like to suggest that the history of computer science—if and when it comes to be written—will establish a new and different paradigm for history writing. It may indeed rid us of certain stereotypes common to the history of science, with its overemphasis on the history of theoretical physics.

In contrast to physics, the fundamental ideas that underlie the development and implementation of large-scale computers are almost commonplace. The principles of computer science are now so well known that they are thought to be few and simple. They are unlikely to fire the imagination of a reading public spoiled by science fiction; nor are they revolutionary ideas on which movie scripts can (or will) be written. In fact, they sound pedestrian, predictable, and instrumental, reminding us of the old adage about mathematics, that the ugliest theorems find the best applications, and vice versa. In computer science, simple ideas requiring little or no intellectual or scientific background have often worked out better than the more complex, subtle, and scientifically inspired proposals.

In universities today, students of computer science are the least historically minded group in a student population not known for its historical concerns. They seem to believe that the current concepts in the field have existed from time immemorial, like a patrimony that all have the right to access. Priorities in discovery have been unjustly attributed; individuals who had no part whatever in the development of the field, such as Alan Turing, are now given the status of heroes, while the names of those who did the hard work, like John von Neumann, are scarcely remembered.

The phenomenon of obsolescence is particularly acute in computer science; it works against the historian's task. In the age of the microchip, the history of the vacuum tube has only limited appeal. The discovery of a new computer model surrounds memories of all preceding models with a thick web of irrelevance. In examining a computer of ten or twenty years ago, our first reaction is not one of curiosity mixed with wonder and admiration, as it should be, but of embarrassment, revulsion, almost irritation. The inspection of the creations of our masters elicit smiles, or, more often, giggles. The work of our predecessors has little to teach us, not even in the lessons derived from what we perceive to be their clumsiness. In computer science, obsolescence means a total break with the past, which uniquely distinguishes this field from all others.

The relationship between computer science and mathematics scarcely resembles that which exists between physics and mathematics. The latter may best be described as an unsuccessful marriage, with no possibility of divorce. Physicists internalize whatever mathematics they require, and eventually claim priority for whatever mathematical theory they become acquainted with. Mathematicians see to it that every physical theory, sooner or later, is freed from all shackles of reality and liberated to fly in the thin air of pure reason.

Computer science, in a very different mode, turns to mathematics in much the same way that engineering always has. It freely borrows from already-existing mathematics, developed for altogether different purposes or, more likely, for no purpose at all. Computer scientists raid the coffers of mathematical logic, probability, statistics, the theory of algorithms, and even geometry. Far from resenting the raid, each of these disciplines is buoyed by the incursion. Statistics will never be the same given what the processing of large samples by supercomputers has made possible. The Monte Carlo method, without which computer simulations of neutron diffusion would have been impossible, was developed by Ulam and myself without any knowledge of statistics; to this day the theoretical statistician is unable to give a proper foundation to the method. In a similar way, the theory of algorithms would amount to very little without the needs of computer software. The rebirth of Euclidian geometry in the most classical vein can be traced to the requirements of computer graphics. Like any other engineer, the computer scientist does not stop to work on whatever mathematics he or she may need. Rather, a segment of the mathematical population, relabeling itself "theoretical computer scientists," meets the mathematical needs of the other computer scientists. This shift, if nothing else, has been financially beneficial.

Two branches of mathematics have been wholly revamped, indeed given a new lease on life by being required to meet the needs of computer science. Mathematical logic is one. The other is the once-obscure chapter of probability theory, now called *reliability theory*. The beginning of this transfiguration may be traced to a master's thesis written by Claude Shannon at MIT in 1939. A brief summary of his principal idea will illustrate my point.

Computers are made up of circuits consisting of large numbers of replicas of identically behaving units. Once upon a time the units were vacuum tubes; later, they were transistors; today, they are chips. Every chip processes electric signals which enter at one point and exit at another. Signals going through various chips can be connected in essentially two ways: in series or in parallel. Two chips A and B are said to be connected *in series* when the exit point of A is soldered to the entrance point of B, so that a signal entering through the entrance point of A will automatically be routed through B, and finally exit through the exit point of B. On the other hand, chips A and B are said to be connected *in parallel,* when the entrance points of A and B are soldered together, as well as the exit points of A and B. In this way, a signal entering at the joint entering point of two chips connected in parallel has a choice of whether to go through A or through B before exiting at the common exit point.

Shannon's fundamental insight was that series and parallel connection of chips are analogous to the connectives *and* and *or* of mathematical logic. Indeed, when A and B are connected in series, the resulting circuit will send a signal through if, and only if both A *and* B are processing the signal. When A and B are connected in parallel, the resulting circuit will send a signal through if and only if either A *or* B is processing the signal, not necessarily both.

By this analogy, any logical expression involving *and* and *or* (as well as the third essential logical connective, *not,* covered by a rather ingenious trick) can be replicated by circuits. Simple as Shannon's observation was, it ushered in the age of computing. The design of expert systems in our day further exploits the basic idea that circuits can be made to perform logical operation, for example, by developing circuit-theoretic devices that render the Fregean quantifiers *for all* and *there exists.*

Shannon's idea of relating series and parallel connection with the two basic connectives of logic was to bear fruit in a direction that has proved central to computer engineering. In the logical interpretation of electric circuits, truth and falsehood correspond to whether or not a chip processes a signal. A more realistic assumption, however, is that the chip will work or not with a certain probability, depending on several factors, including the age of the chip. A realistic model for this situation is to assign to each chip in a circuit an exponentially distributed random variable. Random variables corresponding to distinct chips can be assumed to be independent. Thus motivated, probabilists were led to develop a remarkable calculus, which is now known as *reliability theory.*

The principles of reliability theory are simple. If chip A has probability p of failure and chip B has probability q of failure (we disregard the possibility of these probabilities varying with time), then the probability that the series connection of A and B will fail is $1 - (1 - p)(1 - q)$, and the probability that their parallel connection will fail is pq. When p and q are restricted to the extreme values 0 or 1 one finds, as a limiting case, Shannon's interpretation of the logical connectives. Any series-parallel circuit has a certain probability of working, which can be computed by iterating the above two rules. Such a probability is called the reliability of the circuit.

Reliability theory is concerned with the design of circuits of high reliability at a minimum cost. No computer circuit can be designed without allowing for the possibility that one or more components may fail (what von Neumann was the first to call the "synthesis of reliable circuits from unreliable components"). Soldering two or more chips in parallel will increase the reliability, since a signal will still go through even if one the other fails. If chips cost nothing, we could achieve perfect reliability by soldering together chips in multiple parallel connections. In practice, however, the costs of such a design would be prohibitive. Soldering chips in series decreases the cost of the circuit, but it also decreases the reliability. In computer design, the engineer is forced to fall back on his or her own wits (or on those of mathematicians) to design (or "synthesize") circuits of high reliability at a minimum cost.

The design of complex systems of high reliability—whether airplane wings, telephone networks, or computers—is a daunting task. It is unquestionably the central issue of today's computer science. Some of the most ingenious mathematics of our day is being developed in response to the needs of reliability theory.

Although the basic rules for the computation of reliability were long known, it took several years during and immediately after World War II for the importance of the concept of reliability to be explicitly recognized and dealt with. Only then did reliability computation became an essential feature in computer design.

The late Richard Feynman was one of the first to realize the centrality of reliability considerations in all applied scientific work. In the early days of the Manhattan Project in Los Alamos (in 1943 and early 1944), he tested the reliability of his first program in a dramatic fashion, setting up a day-long contest between human operators working with hand-operated calculators and the first electromechanical IBM machines. At first, human operators showed an advantage over the electromechanical computers; as time wore on, however, the women who worked with the calculators became visibly tired and began to make small errors. Feynman's program on the electromechanical machine kept working. The electromechanical computers won out by virtue of their reliability.

Feynman soon came to realize that reliable machines in perfect working order were far more useful than much of what passed for theoretical work in physics, and he loudly stated that conviction. His supervisor, Hans Bethe—the head of T-Division (*T* for *theory*) at the time and a physicist steeped in theory—at first paid no attention to him. At the beginning of the Manhattan Project, only about a dozen or so hand-operated machines were available in Los Alamos; they regularly broke down, thereby slowing scientific work. In order to convince Bethe of the importance of reliable computation, Feynman recruited me to help him improve the performance of the hand-operated desk calculators, avoiding the week-long delays in shipping them to San Diego for repairs. We spent hours fixing the small wheels until they were in perfect order. Bethe, visably concerned when he learned that we had taken time off from our physics research to do these repairs, finally saw that having the desk calculators in good working order was as essential to the Manhattan Project as the fundamental physics.

Throughout his career, Feynman kept returning to the problem of the synthesis of reliable computers. Toward the end of his life, he gave a remarkable address at the fortieth anniversary of the Los Alamos Laboratory where he sketched a reliability theory based on thermodynamical analogies. In contrast to Bethe, John von Neumann very quickly realized the importance of reliability in the design of computers. It is no exaggeration to say that von Neumann had some familiarity (in the 1950s) with all the major ideas that have since proved crucial in the development of supercomputers. Von Neumann realized very early the advantage of parallel computation over series computation. He knew that the day would come when series computations would reach their physical limit, namely, the velocity of light, and that only a computer based on the principles of parallel computation could exceed that limit. Curiously, however, his choice of series computation in preference to parallel computation (now referred to as the "von Neumann computer") was the result of his negative experiences with the first experiments he devised to test the effectiveness of parallel computation. Repeatedly frustrated by his inability to achieve the required synchronicity in a simple parallel computation experiment that he set up (an impossible task in his time), the failure kept him at a distance from all ideas of parallelism for the rest of his life.

The first large-scale electronic computer to be built, the one that may be said to inaugurate the computer age, was the ENIAC. It was built at the Moore School of the University of Pennsylvania by an engineer and a physicist—Presper Eckert and John Mauchly. Their idea, trivial by the standards of our day, was a revolutionary development when completed in 1945. At the time, all electromechanical calculators were built exclusively to perform ordinary arithmetic operations. Any computational scheme involving several operations in series or in parallel had to be planned separately by the user. Mauchly realized that if a computer could count, then it could do finite difference schemes for the approximate solution of differential equations. It occurred to him that such schemes might be implemented directly on an electronic computer, an unheard of idea at the time. They managed to sell their idea to the US Army, which authorized funding of the project, on the condition that the machine be used at the Aberdeen Proving Grounds for ballistic computations. A Captain H. Goldstine was chosen by the Army to supervise the project and was to benefit greatly from the interaction with Eckert and Mauchly.

Alone among the large computers of the time, the ENIAC was designed with paramount concern for reliability. It consisted of eighteen thousand vacuum tubes wired together, with full allowance made for redundancies that would increase reliability. Most of the maintenance work involved the replacement of vacuum tubes that went out of order. To many observers unfamiliar with reliability computations, it seemed a miracle that the ENIAC worked at all. Enrico Fermi, who later was to become one of the first physicists to perform large computer experiments, made only one incorrect prediction so far as I know: he mistakenly computed the reliability of the ENIAC on the basis of the mean free time between vacuum tube failures; he announced that the machine could never work, scarcely realizing that the ENIAC was far more reliable than the counting apparatus in his lab.

In spite of all predictions to the contrary, the computer worked for periods of several hours without error. The designers of the computer resorted to all manner of precautions to keep the vacuum tubes from failing, including keeping "heaters" on at all times. I remember distinctly the time when the ENIAC was dismantled and packed for transportation to the Aberdeen Proving Grounds. Each of the wires was carefully marked and then clipped; I never believed that Mauchly and Eckert would be able to put it back together again. They did, and the ENIAC proved to be a great success.

At the time the ENIAC was installed, von Neumann was a consultant at the Aberdeen Proving Grounds. Realizing that the ENIAC was being underused, he proposed that it be put to work on a computation that would simulate a one-dimensional thermonuclear explosion, following on the notions of Edward Teller's group at Los Alamos. The computation was finally made, and the ENIAC came through with flying colors. The experiment came to be known as the "shakedown cruise" of the ENIAC.

At the end of the war, von Neumann and I began to plan the building of a more powerful computer in Los Alamos, which would benefit from the experience of the ENIAC and the reliability lessons that it had taught us. I spent a year at the Institute of Advanced Study in Princeton to discuss detailed plans with von Neumann. Edward Teller, who was then beginning to do his calculations on thermonuclear reactions, enthusiastically encouraged us to go ahead with the project.

The MANIAC took several years to build. It was finally operational in 1952, and a more realistic computation of a thermonuclear reaction was finally tried on it, with great success. Of all the oddly named computers, the MANIAC's name turned out to be most unfortunate: George Gamow was instrumental in rendering this and other computer names ridiculous when he dubbed the MANIAC "Metropolis And von Neumann Install Awful Computer." Fermi and Teller were the first hackers. They would spend hours at the console of the MANIAC. Teller would spend his weekends at the laboratory playing with the machine. Fermi insisted on doing all the menial work himself, down to the least details, to the awed amazement of the professional programmers. He instinctively knew the right physical problems that the MANIAC could successfully handle.

His greatest success was the discovery of the strange behavior of nonlinear systems arising from coupled nonlinear oscillators. The MANIAC was a large enough machine to allow the programming of potentials with cubic and even quartic terms. Together with John Pasta and Stanislaw Ulam, he programmed the evolution of a mechanical system consisting of a large number of such coupled oscillators. His idea was to investigate the time required for the system to reach a steady state of equidistribution of energy. By accident one day they let the program run long after the steady state had been reached. When they realized their oversight and came back to the computer room, they noticed that the system, after remaining in the steady state for a while, had then departed from it, and reverted to the initial distribution of energy (to within two percent).

The results were published in what was to be the last paper Fermi published before he died. Fermi believed this computer-simulated discovery to be his greatest contribution to science. It is certainly the first major scientific discovery made by computer, and it is not fully understood to this day (though it has spawned some beautiful ideas).

In the same year that the MANIAC was inaugurated, 1952, the first public demonstration of computer reliability was instrumental in convincing the public of the importance of computers. Howard K. Smith employed the UNIVAC on television to predict the outcome of the presidential election. Shortly after the polls closed (within half an hour, actually), the UNIVAC predicted an Eisenhower landslide. The programmers' disbelief that immediately followed this prediction and their subsequent retraction made the computer's prediction all the more astounding. The rise of computer science can be traced to that day.

The history of computer science since 1952 is far more complex. The underlying mathematical and engineering ideas were already known at that time and have since varied only in detail. The gap between these ideas and their implementation, however, was to grow wider as the demand for speed and reliability increased. In fact, the discontinuous leaps forward in computer design went hand in hand with advances in chemistry and material science. The discovery of the transistor, and later the introduction of the miraculous chip, are the two main stages that mark turning points in computer science. It is my hope that a historian of computing will some day tell the fascinating stories of these inventions.