

# **The Early Years of Academic Computing:**

## **A Memoir by Kenneth M. King**

*This is from a collection of reflections/memoirs concerning the early years of academic computing, emphasizing the period from the 1950s to the 1990s when universities developed their own computing environments.*

©2014 Kenneth M. King  
Initial Release: June 2014

Published by The Internet-First University Press  
Copy editor and proofreader: Dianne Ferriss  
The entire incremental book is openly available at <http://hdl.handle.net/1813/36810>

Books and Articles Collection – <http://ecommons.library.cornell.edu/handle/1813/63>  
The Internet-First University Press – <http://ecommons.library.cornell.edu/handle/1813/62>



---

## **B. Memoir by Kenneth M. King**

### **Preface**

The two most serious oriental curses are reputed to be: “May you live in interesting times and may your spouse understand you”. People involved in University Computing in the early years certainly lived in interesting times. I have long believed that Universities have not gotten enough credit for the many things that they did to advance technology and its applications. I believe that Bill Arm’s memoir, providing an inside view of the development of time-sharing and distributed computing, two huge advances in the development of operating systems and improving the human interface to computers, is a great start at chronicling some of those contributions. When Bill Arms approached Bob Cooke and me about publishing his memoir in the Internet-First University Press (IFUP), I suggested that he try to create an “incremental book” by persuading other pioneers to contribute their memoirs. The incremental book idea was developed by Bob Cooke as part of our joint effort to get scholarly information on the Internet. The idea was that we would publish chapters or sections in a book on the Internet as they are finished, rather than wait until the whole book is completed. Bill suggested that I develop my own memoir and that we then jointly approach other pioneers to create an incremental book composed of memoirs by people who labored at Universities and may recall events and contributions that helped change the world. This second section in our book covers my observations about life in the trenches at Universities during the period between 1953 and 1993. The book that we hope to create is possibly an effort not unlike the story about an elephant touched by blind men. People that we hope will contribute will certainly have different slants on major events at the Universities that helped change the world.

# Contents

<b>Preface</b> .....	<b>2.1</b>
<b>1. Computing at the Watson Laboratory at Columbia University in the 1950s</b> .....	<b>2.3</b>
<b>Program Languages are Developed</b>	
<b>2. Computing at Columbia University in the 1960s</b> .....	<b>2.7</b>
<b>Computer Literacy</b>	
<b>Impact of Computing Technology Changes</b>	
<b>3. Computing at City University of New York in the 1970s</b> .	<b>2.15</b>
<b>The City of New York Goes Broke</b>	
<b>4. Computing at Cornell University from 1980 to 1987</b> ...	<b>2.21</b>
<b>Major Computing Events at Cornell</b>	
<b>Cornell Becomes a National Supercomputer Center</b>	
<b>Creating NSFnet</b>	
<b>5. EDUCOM Activities from 1987 until 1992</b> .....	<b>2.35</b>
<b>The EDUCOM/NCRIPTAL Outstanding Software Awards</b>	
<b>The NREN Becomes the Internet</b>	
<b>6. Looking Back</b> .....	<b>2.41</b>
<b>7. The Future Lies Ahead</b> .....	<b>2.45</b>

# Chapter 1

## Computing at the Watson Laboratory at Columbia University in the 1950s

My involvement with computing at Universities spanned the years from 1953 to 1993. During that period Universities were responsible for many major advances in computing technology, and University Computer Center staff and University Faculty played a big role in many of those advances. There were four major thrusts in the effort to advance computer technology: efforts to improve hardware performance; efforts to improve computer throughput (primarily directed toward improving operating systems); efforts to provide network access to computers; and efforts to improve the productivity, ease of use, and tools available for people who were programming computers. In the early years, hardware performance and throughput trumped efforts to improve the human interface, if those improvements came at the expense of throughput.

During the 1950s, there were debates on the issue of whether or not Universities should be involved in building their own computer hardware. The Institute for Advanced Study at Princeton built the IAS computer following a design by John Von Neumann from 1945 to 1951, the University of Pennsylvania built the Eniac computer in 1947, and the University of Chicago was building Maniac computers until the early 1960s. By the end of the 50s it became clear that it was impossible to compete with commercial vendors of hardware, and Universities stopped building their own computers. However, people who started a number of commercial hardware companies had deep connections to University Engineering Schools. Over the years that spanned my career, University contributions to the advance of technology were primarily in the areas of computer operating systems, application packages, languages, networking, and the computer human interface.

I became involved with computing as a graduate student in the Physics Department at Columbia University. The IBM Company maintained and funded The Watson Scientific Computing Laboratory at Columbia University, and up until the early 50s the laboratory was stocked with IBM's latest and greatest commercial hardware, primarily tabulating machines. It was at the time one of a few premier University Computer Centers in the world, and IBM provided instruction in programming to its scientific customers at the Laboratory. In 1953 I wanted to solve some interesting problems in Quantum Mechanics, and IBM was building a new computer at the Laboratory for the Navy, called the NORC (Naval Ordinance Research Calculator). From 1954 to 1958 it was the world's fastest computer. It had a whopping 2000 words of cathode ray tube storage with a memory access time of 8 microseconds, an online card reader, card punch, line printer, and some magnetic tape drives. On a good day it could execute about 20,000 arithmetic operations per second. It was programmed in decimal machine language and during its construction I and another graduate student had access to it at odd hours. For its dedication in December 1954 I computed a table of logarithms in a few minutes that circa 1600, Napier had spent many years computing. Unfortunately for Napier, his logarithm tables were necessary to support navigation across the seas, and the age of exploration could not be postponed 400 years waiting for the NORC. I also computed pi and e, the base of the natural logarithm system to a million places, in something like 20 minutes, because John Von Neumann, who was attending the dedication, was interested in testing the digits for randomness. As predicted by theory they were indeed random. In the 50s and 60s, using the most advanced computing technology available at the time, Wallace Eckert and Harry Smith spent years at Columbia University computing the orbit of the moon to enough precision to support the Apollo space program. For computers with small memories backed by tape drives, any program whose data didn't fit into the available memory

proceeded at close to tape access speed. Today this calculation could be done on a laptop in minutes with our gigabyte memories. Unfortunately for them the age of space exploration couldn't wait.

When the NORC left at the end of 1954, the equipment remaining at the Watson Laboratory supported my computing activities, along with a slide rule that every physics student carried at that time. The principle-computing engine in 1954 was an IBM Card Programmed Calculator (CPC), which consisted of an IBM 402 tabulating machine connected to an IBM 604 electronic computer and a card punch. The tabulating machine was controlled by a large removable wired program board, and the 604 was capable of executing 40 program steps on the data on each card fed into it by the 402, and punching the result onto another card. It was altogether a fun machine to work with and capable of executing about 25 arithmetic operations per second. In 1955 the CPC was augmented by a pair of IBM 650s. The 650 was a drum machine containing 2000 words of storage, and each instruction specified an operation to be performed on data at some address on the drum, followed by the address of the next instruction in the program. A drum was a rotating cylinder coated with a magnetic material. The drum rotated at 2500 revolutions per minute. The challenge was to locate the program steps and data on the drum so you didn't have to wait for the drum to rotate all the way around to find the data for the current instruction, or for the next instruction to be executed. Carefully programmed, it was capable of executing a few hundred arithmetic operations per second.

## **Program Languages are Developed**

In 1955 Stan Poley at the Watson Laboratory wrote the first program I had encountered that improved the machine-user interface. We later came to call these programs "killer apps". The program, called SOAP for "Symbolic Optimizing Assembly Program", allowed you to use symbolic names for computer operations and data, and the assembler optimized the location of instructions and data on the drum to improve throughput. At the time I thought, "It can't get any better than this"! SOAP ended machine language programming forever. A couple of years later, when I became the manager of the Laboratory, we entertained students from the Bronx High School of Science for a couple of hours twice a month. A 13-year-old student in the group undertook the challenge of writing an improved version of SOAP. He enthusiastically described his program to me. He had named his program "CLAP" for "Computer Language for Assembly Programming". It was my awkward duty to suggest to his teacher that his program possibly could benefit from a different name. This name was used at the time to describe an inharmonious medical problem and perhaps still is. Those students were wonders and proved to me that people at a very young age could become skilled programmers.

In about the middle 50s, John Backus from IBM came up with the idea of creating a computer language that allowed programmers to enter formulas into their code and have a compiler translate the formulas into machine language code. He needed \$200,000 to produce the compiler, and an IBM Vice President thought it prudent to have John Von Neumann review the idea. On a hot summer afternoon Von Neumann appeared at the Watson Laboratory to listen to Backus. An IBMer present at the meeting reported to me that Backus gave an inspired presentation with particular emphasis on the fact that the compiler would have an optimizer that allowed it to produce code comparable in efficiency to assembly language code. It would also reduce program size and save programming time. Code efficiency was a critical issue in those days and trumped improving the programmer interface. The room was hot and conference rooms and offices were not air-conditioned in those days. My informant said that during the presentation Von Neumann struggled to keep awake. After the presentation he voiced concern that introducing floating-point arithmetic capabilities into scientific calculations could cause people to lose control of error propagation, resulting in the possibility that buildings and bridges designed with the aid of computer analysis might collapse. But he concluded that it was probably worth trying. His tepid approval caused great anxiety in the Backus team, but the project was approved and, as they say, the rest is history. Fortran resulted in a huge increase in the computer market and it was clearly a "killer app".

Replacing fixed-point arithmetic with floating-point arithmetic was also a “killer app”. In writing a program with fixed word size and boundaries, the necessity to figure out where to put the decimal point in every piece of data, to get maximum precision, took enormous amounts of time. But he was right to worry about a potential problem. If in your program you subtracted two numbers of approximately equal size and either or both of these numbers contained a small error (experimental or round off), you would be left with a floating-point number that was all or almost all error. It was not uncommon for people to treat computer output as if it had been written by lightning on granite. When the IBM 704 computer was available in the late 50s at an IBM Service Bureau in Manhattan, IBM provided a small amount of free time to Columbia people. When you compiled a Fortran program on the machine, the signal that the compiler had successfully compiled your program, and that execution had begun, was the appearance of a “word overflow” error light on the console of the 704. Perhaps this was an accident or someone on the Backus team remembered that afternoon at the Watson Lab.

In those halcyon days there were several hundred users of the equipment at the Laboratory; you signed up for time a week in advance and, if you were lucky, you were allocated an hour or more each week of exclusive access to a machine. People would hang around the Laboratory hoping that someone might not show up to use their time, and if someone ran over their time, cutting into your time, the verbal exchanges could get X-rated.

On one occasion, a mathematics professor was using his hour and seemed to experience serious anxiety that the card punch on the 650 would run out of blank cards. He nervously hopped up and down, and in the interval between cards being punched he would periodically quickly raise the hopper lid and flip a few new blank cards into the punch hopper. At the end of his allotted time he pulled his output from the card punch and observed that the decaled edge of the cards seemed to be randomly distributed in every corner of the deck. He had failed to feed all the cards face down, 12s-edge-first into the punch. It was an easy mistake to make. Every tabulating machine had a different rule for inserting cards; some face up, some face down, some 12s edge first, some 9s edge first. That’s why college graduates were hired to put cards into those machines. A card had a printed face with the decaled edge on the top left. The top of the card was called the 12s edge and the bottom was called the 9s edge. I thought that the Professor was going to cry. He said, “I’ve wasted a whole week”! I told him that his output could still be rescued because the printer didn’t really care whether the cards were face up or face down and he could still print his output if he inserted the cards properly into the printer. He seemed very puzzled and held the thick deck of cards over his head turning and twisting the deck as he looked at it. He suddenly exclaimed, “That’s right, the holes go all the way through”! As he had this aha moment he flipped his hand and his output flew into the air and was distributed across the computer room floor. It was now my sad duty to inform him that he had indeed wasted a week. At the time I pondered whether his observation on the holes might serve as a starting point for a new computer-based academic discipline, but then dismissed the idea.

In 1960 IBM established a Systems Research Institute in Manhattan in a building across from the United Nations for graduate-level training of its System Engineers. The first faculty, all part-time, included Fred Brooks who managed the development of the IBM 360 family of computers and its software, Ken Iverson who was the developer of the computer language APL, and me. I taught Numerical Analysis. This experience convinced me that there was an academic niche for studying computer-related issues in higher education that went well beyond the design of hardware. The first Computer Science Department established at a U.S. University was at Purdue in 1962.

So how important were computers to Columbia University in support of Research and Education in the 50s? There were a fair number of projects in the Social Sciences and a few at the Medical School that involved statistical calculations. There were a small number of projects in Astronomy, Physics, Chemistry, Engineering, Economics, and Geology, and a handful of projects from all the other disciplines at the University. Computers and computing tools were primitive and trying to analyze systems of any complexity was almost impossible. Problems involving ordinary differential equations or systems of linear equations began to come into range. Numerically solving partial differential equations except in a few rare cases was not possible. But with the ad-

vent of electronic computers at the end of the decade, there were people who were beginning to dream about understanding things like the weather, and the structure of molecules. There was a programming course at the Watson Lab taught by the legendary Eric Hankam for IBM's scientific customers that Columbia students could enroll in, and I taught a course in Numerical Analysis for the Mathematics Department. But computing instruction involved only a small number of students. This situation changed dramatically in the next decade.

## Chapter 2

### Computing at Columbia University in the 1960s

About 1961 IBM decided to move its education program located at the Watson Lab to upstate New York and to close the Computing Laboratory at the Watson Laboratory. It negotiated with Columbia University a plan for Columbia to build its own Computing Center. I began exploring various job opportunities at a number of places. One day I received a call from an architectural firm asserting that they were designing an underground computer center building for Columbia and they had been told that they needed to consult with me if they had any design questions. I had heard nothing about this from Columbia but told them that I would be glad to help. About a month later I received a call from a Columbia Dean saying that he would like to talk to me. At the meeting he said that after consulting with a number of people, Columbia wanted to hire me as the Director of the new Columbia Center. I agreed to serve as Acting Director for a year while I continued to explore other opportunities, some of which were very interesting.

As construction of the underground building that would house the Center started, I needed to hire a staff and prepare for the delivery of an IBM 7090 and IBM 1401. The 7090 was a computer designed for scientific calculations and in it transistors had replaced vacuum tubes. The role of the 1401 was to prepare batch input tapes for the 7090 and to print output from tapes produced by the 7090. In looking for staff there were not a lot of experienced people around. I hired a few people from the Watson Lab staff, some very bright Columbia and Barnard graduates with no computing experience, and an operator with IBM 704 experience. We studied the manuals describing the Fortran Monitor system and had help from some very good IBM system engineers. Over time we developed a world-class staff. When I became manager of the Watson Lab Computing Center in 1957, I asked Wallace Eckert, the Director of the Watson Laboratory, for advice on managing people. He said: "Just hire good people and keep out of their way". His office was in a building one block from the Computing Lab and he would appear at the Computer Lab about once a year and mumble something to the staff about our doing a good job, and disappear for another year. I tried to follow his advice on management during my entire career.

I was The Director of Computing at Columbia from 1962 to 1971. During that period equipment was upgraded about every two years. The 7090 became a 7094; a 7040 was added and connected to the 7094; a 360/50 and a 360/75 replaced the 7094-7040 system to form a 360/75 and 360/50 coupled system; and a 360/91 was added and attached to the 360/75, displacing the 360/50. Operating systems changed apace.

The 7090 was capable of executing 100,000 floating-point operations per second and had a list cost of \$2.9 million. Universities, however, enjoyed a 60% IBM educational discount. The 360/91 was about 50 times faster and the list cost of a complete system ranged between \$15 and \$20 million. In 1965 Herb Grosch stated Grosch's law: Computer performance increases by the square of the cost. This became the basic argument for Universities to operate a large central computer. You got more bang for the buck with a faster computer. With the development of computers on a chip and microcomputers, Grosch's law was superseded about 1980 by Moore's law. Moore's law stated that the number of transistors on silicon chips doubles approximately every two years. Gordon Moore also stated this law in 1965.

During the transition from the Fortran Monitor System to IBSYS on the IBM 7094, a hacker helped us. One night a message was printed out on the console of the 7094 that said, "The phantom strikes", and all of the tape

drives were rewound past their load point and the tape ends were left unattached to their rewind reels and dangling in their columns. There was no way the hardware was supposed to let that happen.

The tape drives were attached to a channel control box, which had a wired-in program controlling the drives, and the 7094 sent instructions to the channel program directing its tape control actions. Ken Iverson had written an APL description of the channel program and apparently some student had read it. In this case a student had discovered that if a tape was at its load-point marker, and a set of cleverly designed 7094 instructions were sent to the channel-control program with precisely timed delays between those instructions, the channel program began looking for the load mark on the tape after the tape had already coasted past it. A rewind instruction then brought the tape off its rewind reel.

At the time we were running Fortran monitor batches and IBSYS batches with the operators dismounting and mounting the system's tapes between running each batch. IBSYS enabled running a number of additional applications not available under the Fortran Monitor System. With the discovery of this bug, which had now become a "feature", we were able to put both operating systems on the same tape—separated by load and end tape markers—and include IBSYS and Fortran jobs in one batch, with the system spacing over the load and end of tape marks to the system called for by a program without changing any of the instructions in either operating system. Over the years, a number of computer bugs were reclassified as features. A few years later I taught a course in operating systems for the Electrical Engineering Department and there were students in my class who were smart enough to become world-class hackers. Operating systems contained hundreds of thousands of instructions with the number increasing rapidly. Removing all vulnerabilities was impossible. There were always a few students who wanted to "make their bones", to use a Mafia term, by bringing the system down and I was teaching them what they needed to know to do it. A University Computing Center Director at a meeting once advised his fellow Directors that if they caught a good hacker, hire him or her. I found this to be very sound advice.

Computers at that time had very small main memories with access times measured in microseconds supported by secondary storage like disks, drums, and tapes with access times measured in milliseconds to seconds. In order to increase throughput, evolving operating systems tried to hold multiple programs in main memory so that when one program was retrieving data on a secondary storage device, the computer could switch to run another program in memory that was ready to run. This strategy greatly complicated operating systems.

When the IBM system 360 was introduced, its operating system OS/360 provided multiple options for multi-tasking. Unfortunately, it took a long time to get most of the bugs out of the system and University environments being very diverse, experienced many more system crashes than commercial sites. A system crash meant taking and printing a memory dump, identifying the program causing the crash in order to remove it, and restarting the batch. This process took many minutes of time. The systems group was then tasked with finding the bug and removing it. Users of the Computer Center measured service quality by turnaround time, the time between job submission and receipt of program output, and crashes were very bad for turnaround time. This was a miserable time to be managing a University Computer Center.

Users of large-scale IBM scientific computers had a user group called SHARE. In the 60s I chaired the university's SHARE group. Universities were always willing to share useful software and SHARE meetings were a place to learn about interesting applications being developed at other Universities. Technical attendees shared information on operating system bugs and the problems associated with installing updates to the operating system. These updates were frequent and sometimes affected many users, which did not make them happy. At Universities, major system upgrades were almost always scheduled for August when most faculty members were away. This had the consequence that some faculty would return in September, tanned and eager to resume their research and instruction, only to discover that, as President Nixon once said, "the information you have been given is no longer operational".

Every evening most of the SHARE attendees would convene in a large space to attend an informal event called SKIDS. At the center of the meeting space were tables full of booze and ice. Information exchanges at that event were often very useful. I quickly learned that the stress associated with the conversion to OS/360 in Industry was almost equal to that at Universities. We had less control over our users and a lot more crashes. You could approximately measure a person's stress level by how long they stayed at SKIDS. Late one evening at SKIDS, a person who managed computing at an insurance company told me that the Vice President for Marketing had a year earlier proposed to the Board that they augment the customer account number by a digit to record policy-holder characteristics that would aid marketing. When a board member asked if that would screw up data processing, the Marketing Vice President replied, "Hell, its just another hole in a card". There was also the annual Ivy League Chowder and Marching Society's meeting of Ivy League Computer Center Directors. The rate of change in membership in that Society was also a good measure of the stress. At the meetings, members of the Society would estimate how many steps ahead of the hangman they were at their institution. This generally averaged to about half a step.

In the 60s it became possible to connect printing typewriter terminals to computers, and later CRT (Cathode Ray Tube) terminals. Those terminals enabled programmers to bypass keypunches and produce, edit, and submit their programs directly to the batch and retrieve output to a terminal. These terminals could be hardwired to a computer by connecting twisted-pair wire from the terminal to the computer, or they could be connected over telephone lines using modems. The modems were acoustic couplers that you inserted your telephone receiver into and they converted an analog signal into a digital signal. It was easy enough to connect terminals at the computer center, and a terminal room was created, but challenging to hardwire terminals in remote buildings to the center. Columbia had heating tunnels bringing steam to buildings on the main campus and they contained steam pipes wrapped with an insulator that I hope was not asbestos. In any event it was reported to me that in the dead of night, armed with a flashlight, a staple gun, and a wheel of twisted-pair, unidentified people would staple twisted-pair to the back of the pipes out of sight and bring the wire to the basement of buildings where unindicted co-conspirators from a Department were waiting. This was probably a violation of some code but evidently there were people on campus who believed that if the end doesn't justify the means, what the hell does, or something like that. Several decades later it became necessary for many Universities to rewire their entire campus to accommodate computers, but the 60s were the dawn of the remote computing revolution.

Computer terminals enabled the next major advance in computer operating systems, this one called time-sharing. MIT had developed a time-sharing system in the early 1960s called CTSS that ran on a 7090 that was not capable of supporting a large number of users. Much more ambitious undertakings were started at MIT (MULTICS) and at Dartmouth (DTSS) in the middle 60s on hardware more amenable to the task. The goal was to provide a large number of users with shared terminal access to a computer with real time interaction. These systems, installed at a number of Universities, were very successful in environments where most users ran small programs. This was a characteristic of most instructional computing. But alas, in environments where research, administrative, and educational users shared a large central computer, some running very large programs, a time-shared computer could spend most of its time rolling these big programs in and out of memory from secondary storage and throughput was hurt. Throughput still trumped user ease of use almost everywhere. Columbia did not adopt time-sharing because very large research and administrative programs were a big part of the load. Computer courses used fast Fortran and PL1 compilers developed at Waterloo, Purdue, and Cornell to support instructional computing. Terminal support programs like Wilbur from Stanford enabled users that had access to terminals to enter, edit, and submit jobs and to retrieve output from a terminal, which provided some of the convenience of time-sharing without its inefficiencies.

Terminal access enabled the dawn of the age of social computing. Users were assigned a small amount of online space to store the programs that they were editing and some data. An encryption capability was provided so that sensitive information, like medical data containing patient names, could be protected. I was told that the

gay community at Columbia was using this capability to support a dating service. You “friended” someone by giving him or her the password and encryption key to the file. The encryption key was called the magic word. I was also told that the students involved in the 1969 insurrection at Columbia used this capability to coordinate their activities.

## Computer Literacy

As the 60s advanced Universities began to recognize that computers and computing were an important area of instruction and research. At Columbia, a programming course that I taught in the Electrical Engineering Department attracted hundreds of students each semester. The term “Computer Literacy” became an important issue for discussion, although a lot of people didn’t understand what it meant and some regarded the phrase as a linguistic barbarism. An attempt at a simple explanation of the term follows. If you look at the content of papers in academic journals, you will find that there is a hierarchy of languages used to describe the system being studied. For relatively simple systems there may be a lot of mathematics in the paper, because mathematics permits the description of systems without ambiguity. If you are mathematically illiterate, however, there are papers containing mathematics that you will not be able to understand. Papers describing systems too complex for mathematical description usually employed technical English and ordinary English to convey the points trying to be made, but ambiguity is embedded in both. Linguists point to as simple a sentence as “Mary had a little lamb”. This could mean: Mary owned a little lamb, Mary ate a little lamb, or Mary had sex with a little lamb. People whose native language is English, but who are illiterate, are unable to read and understand the knowledge contained in papers written in English. The new languages in the language hierarchy enabled by computers are formal or computer languages. By combining mathematics and logic, these languages make it possible to describe complex systems without ambiguity, and they enable analyzing systems too complex for mathematical description. They also enable the capturing of intelligence that can be applied to a broad range of problems. Thus, understanding how to model systems employing a computer language is a literacy issue and an invaluable capability for any well-educated person. Farther down the language hierarchy are papers employing metaphors and poetry to describe systems as complex as humans. Maybe the hierarchy should be inverted with poetry at the top and mathematics at the bottom.

In the late 60s a faculty colleague in Electrical Engineering and I wrote a letter to the Provost at Columbia telling him why we thought it was important for Columbia to consider creating a Computer Science Department. He sent us a two-sentence response: “How can you create a science around a device? If computer science, why not X-ray science?” Columbia did not get around to creating a Computer Science Department until 1979. When I was very young, I asked my grandfather at the depth of the Great Depression if he had had a tough life. He replied: “During my life I’ve learned that you lose some and you lose some. You have to learn to take the bitter with the unpleasant”.

Columbia University had started a Seminar Program some years earlier that involved inviting people outside of Columbia to meet with Columbia people to get a varied perspective on some subject. The seminars were recorded and the discussions transcribed with the notion that at some point a book could be created. There was one at that time dealing with criminal justice. Outside members included former criminals as well as members of the criminal justice system to get a varied perspective.

In the late 60s Ted Bashkow from Electrical Engineering and I created a Seminar on the Impact of Computers on Society. Outside members came from as far away as Brookhaven, Yorktown, and the Bell Labs. The Seminar members would meet at the bar in the Faculty Club about 6 PM and would shortly adjourn to a conference room to hear a speaker. The Faculty club had a bartender who asked me what I would like to drink the first time I showed up at the bar. The second time he simply asked: “The usual Dr. King?” After that he would prepare my drink and hand it to me as I reached the bar saying simply “Good evening Dr. King”. He set a standard for

service that I have never seen equaled. About 1970, we invited the Provost to speak at the Seminar. To our great surprise, he said that he was willing to participate in a discussion on the impact of computers on Society with the members of our Seminar. He turned out to be a very charming person full of amusing anecdotes. We talked about Computers and Society but not about the subject of the earlier letter. It was the first and only time I ever met him. I came away wondering if, instead of reporting at the level of buildings and grounds, I reported at a level where I had more frequent access to him, if I could have changed his views. About 25 years after I left Columbia I was invited to speak at the Seminar. I was astonished to hear that it had survived all of the intervening years and accepted the invitation. As I approached the bar at the Faculty club at 6 PM the Bartender said: "The usual, Dr. King?" This exchange erased all of the bad memories I had of some of the events that had occurred during my Columbia years.

Speaking of bad memories, I will now turn to the subject of how computers were financed at Columbia. When the center was started, Princeton and Columbia argued that access to computers was as important to researchers as access to the library and, like the library, computer costs should be put in the indirect cost pool and not charged directly to Federal grants. They secured agreement to this practice and users of the Center were not be charged for computer time while this agreement was in effect. Once a year a Federal Auditor would show up at the center to look at time records to determine if the University was correctly including time used by researchers with research grants in its indirect cost pool. We maintained a closet with monthly computer time use records by project and jobs run. In a year's time the stack of printer output paper was about 6 feet high. When the auditor arrived we would offer to transport the stack to the conference room if that would be more convenient for his analysis. He always responded, "I don't think that that will be necessary", and left after looking at the stack for a few minutes. In fact Columbia was always totally honest in allocating computer time to the indirect cost pool.

In 1967 the Government decided that it would no longer accept inclusion of computer costs in the indirect cost pool. Research grants had to be charged directly for computer time. This change led to the end of "civilization" as we knew it. Many researchers had small grants that could not be augmented to include their computer costs. For others it was easier to get one time money to buy a mini-computer than to get recurring funds for computer time. When computer time seemed free, other researchers did not consider the issue of whether or not a dedicated computer that they controlled would make them more productive than standing in line at the computer center. They now considered that question and frequently the answer was "yes", owning their own computer would make them more productive. Thus the "computer wars" began. The University decided that buying a departmental or project computer was forbidden unless a petitioner could prove to me that there was no way to solve their problem at the Center. This put me squarely between former faculty friends and the administration. Computer wars were common at a number of Universities. Their Computing Directors were often among the SKIDS attendees that closed the bar.

I recall a few other inharmonious events during my years at Columbia. In the first winter in the new underground computer center we discovered that the string of offices down a hallway were all in the same heating zone. At one end of the hallway was my office, which was over a loading dock. At the other end of the hallway was a conference room over a University heating plant. In order to keep the temperature in the conference room at 85 degrees, the temperature in my office had to be about 55 degrees. The staff with offices close to mine wore coats. The staff with an office close to the conference room wore tee shirts. I wrote a letter to the firm that had designed and built the center demanding that they fix this problem. I received a response telling me that they were no longer responsible for building problems.

I responded with a letter telling them that I was, at my own expense, commissioning the Long Island Casket Company to create a bronze plaque to be hung in the computer reception room. The plaque would simply state: To commemorate the nobility of their firm, and below that the date on their letter denying responsibility. I stated in the letter that when visitors, humble and distinguished, visited the center and asked what the firm

had done to merit this distinction, we would tell them. I ended the letter by stating that it was my hope that in the years ahead, when Columbia men gathered in the evening over cigars and brandy, and the conversation turned to great feats of engineering, their firm would be remembered. Shortly later I received a call from the firm telling me that they would, at their expense, fix the problem in about 3 months time. The caller then said that he hoped that I would still be there when the fix was completed. I told him that if freed from my University responsibilities, I would have plenty of time to explore whether or not the anecdotal value of this episode merited inclusion in the Talk of the Town section of the *New Yorker* magazine. He hung up. A short time later I received a communication from the Trustees telling me that they had sole approval authority over all commemorative plaques hung in University buildings. This was the only contact that I ever had with the Trustees of Columbia University.

One day the computer receptionist threw open my office door and shouted, "There is a nut in the computer room with a knife threatening the operators"! I immediately tried to reach security but their line was busy. As I kept trying, the lights on two other lines on my phone started flashing. My secretary had run to the computer room to witness what was happening. In those days there were lights for each incoming telephone line on your phone. The lights told you whether the line was free, busy, or if it was flashing someone was trying to reach you on that line. Compulsively I hit the button on one of the flashing lines and recognized the voice of the faculty member calling me. I said: "I can't talk to you now Bill, there is a nut in the computer room with a knife and I'm trying to reach security but their line is busy". His response was: "this will only take a minute".

On another occasion, the IBM 360/91 began crashing with a mean time to failure of about 30 minutes. The IBM Customer Engineers that were resident at the center claimed that it must be the software. The Columbia systems programmers claimed it must be the hardware, because they had not changed the software. This kind of finger pointing was not uncommon. After a few hours the Customer Engineers agreed that it must be the hardware. After working and watching crashes the rest of the day, the Customer Engineers escalated the problem to the Region, and a new set of Engineers arrived the following day. After these Engineers worked for a time they escalated the problem to a higher level and another new set of Engineers arrived. This set concluded that the problem was in a specific box but running their hardware diagnostics they couldn't find any problem with the box. On the third day a chauffeur driven limousine arrived carrying the designer of the box. He entered the computer room smoking a cigar and asked to see the log that the Customer Engineers kept of every hardware maintenance change. After looking at the log for a minute he pointed to a line in the log, uttered a couple of sentences to an engineer and started to leave. I shouted: "You can't leave until you're sure it's fixed"! He ignored me and departed. Five minutes later the computer was back up and running and a Customer Engineer handed me a one-inch wire. Removing the wire had solved the problem.

About 30 years later I got a call from a lady who was studying some recently declassified documents. In a document, Isador Rabi, a Nobel Prize winning physicist at Columbia, had warned the Government that computer hardware could be very unreliable and pointed to an event at Columbia in which a computer had been down for three days. She asked: "What was that all about?"

In 1969 there was a student insurrection on campus demonstrating against the Vietnam War. Students took over some buildings and the President called in the New York City police. The police wielded their batons on student heads and the consequence was that many uninvolved students, shocked by the sight of students staggering out of buildings with blood streaming out of their heads joined the insurrection and more buildings were taken over. At this point the President evidently decided that calling in the police was not a good idea and a stalemate resulted with students hanging out the windows of occupied buildings shouting defiance. I recall seeing a student standing on the lawn outside one of the buildings holding a sign that said: "Sarah, don't forget to take your pill". The Computer Center was largely locked down but I, standing at the entrance with an armed campus policeman, let in trusted users.

The FBI evidently had a wire on one of the members of the Students for a Democratic Society (SDS) and it was reported to me that at an SDS meeting a member proposed pouring iron filings into the core memory of our machines on the grounds that we were doing classified research. Fortunately for us, another high-level SDS member was a chemistry graduate student whom I permitted to spend the night in the center with a small number of other students who were trying to solve “very big problems”. I was told that he vehemently asserted that he lived at the Center and we were not doing classified research. Some days earlier he had asked me why a cover had been put over a printer by a user who stood at the printer to secure his output as it was printed. I explained that the man standing at the printer was a psychiatrist from the Medical School and the output he was protecting contained transcripts of patient sessions. The psychiatrist resembled Sigmund Freud. I was informed that the student persuaded the SDS to drop the iron-filing plan.

One of the pleasures of managing computing at a University was that it provided a window into a broad range of research on very interesting things. Lunch at the Faculty Club with other faculty was often particularly interesting. On one occasion there had been a headline that day in the student newspaper that said: “Barnard girls not worth losing sleep over”. That precipitated a discussion at lunch over whether or not undergraduate instruction at a men’s college needed to include training on how to deal with women. One of the faculty members present said that men were untrained in issues that went well beyond how to deal with women and proposed developing a curriculum in Life Science. We agreed to meet periodically to discuss what might go into such a curriculum. One of the members, Professor Levine from Mathematical Statistics, developed an algorithm for optimally selecting a spouse. The prescription in the algorithm was that you begin by choosing a number  $n$ , which was the number of women you were prepared to court. Then you divided that number by  $e$  (one of my favorite numbers) the base of the natural logarithm system and rounded up to the next integer. You then courted  $n/e$ . However, you didn’t marry any of them, but you remembered the best one. Now you continued to court until you found a better one and married her. I found that this was an excellent algorithm for selecting a new apartment to rent in New York City or for finding a building in which to put a Computer Center. It was the sole output of the effort to create a curriculum in life science.

## **Impact of computing technology changes**

So what impact did computing technology changes have on Columbia University in the decade between the end of the 50s and the end of the 60s? The percent of faculty involved with computing grew from perhaps 2% to I would guess about 20%. All of the science departments now had some faculty whose research depended on computing. The Biology Department, relatively new to computing use, had faculty undertaking big computer projects. The use of computers in the Business School and at the Medical College had grown substantially. The availability of statistical packages caused a dramatic growth in Social Science computing. There was an occasional project in the Humanities enabled by a new programming language called Lisp. Problems that were a dream at the end of the 50s were now being tackled. They included modeling the structure of the earth using seismographic data, trying to understand climate and weather, and programs designed to understand the forces holding molecules together. Two Columbia faculty members involved with computing at that time, Martin Karplus and William Vickrey, went on to win Nobel Prizes, and if there had been a Nobel Prize in Geology, it almost certainly would have been won by Jack Oliver, who proved that the Pacific volcanic ring of fire was caused by one tectonic plate sliding under another. Computer courses had large and rapidly growing enrollments.

These changes were enabled by substantial improvement in hardware performance made possible by the transition from vacuum tubes to transistors. Computers were now capable of executing up to 10 million instructions per second. A hardware designer at this time told me that instead of resting at the end of the 6th day, he wished that God had worked on a better velocity of light. Physicists later determined that a slight change in the velocity of light would have caused the big bang to terminate in a black hole. The velocity of light created an upper limit on the speed of a computer circuit. Main memories were still small relative to the requirements of

many problems and secondary, random-access storage devices were slow and lacked capacity. Magnetic tapes were still the primary bulk storage medium. Improvements in operating systems significantly improved job throughput. The availability of terminals connected to computers enabled the eventual elimination of punched cards as the program input medium and substantially improved the human-computer interface. Institutions that could use time-sharing systems experienced significantly improved programmer productivity and quickly were able to get a majority of faculty and students involved with computing. It had begun to be clear to me that Universities running batch processing systems would have a tough time moving the percentage of faculty using computers above 20%.

There was a proliferation of new computer languages, some like Lisp directed toward a class of applications. A burning question became “would third-generation languages like Algol or PL1 replace Fortran as the work-horse language at Universities?” That question persisted as new third generation languages like Pascal appeared and the answer was always “no”. A joke current at the time had a computer scientist climbing to the top of a mountain and shouting heavenward, “Will any computer language ever replace Fortran?” A voice came down from the heavens and said: “Yes, but not in your lifetime”. And then the scientist asked: “Will Universities ever appreciate the importance of computing to the University?” And the voice came down from the heavens and said: “Yes, but not in my lifetime”. General purpose packages such as SPSS (Statistical Package for the Social Sciences) and database management software became available that significantly improved programmer productivity in some application areas. Keeping up with new program acronyms became a real challenge. At one meeting I raised my hand and told the speaker that I did not understand his 3la’s. He asked: “What’s a 3la?” I replied, “a 3-letter acronym”.

In 1970 the Federal Government seriously reduced funding to agencies supporting research at Universities. This produced a serious problem at Columbia because the salaries of many faculty members with Research Grants were partially funded by their Grants and many of these faculty members had tenure. One proposed solution to the resulting budget deficit was to cut the budget of administrative services at the University across the board by 10%. Another proposal from some faculty members in Arts and Sciences was to eliminate the School of Engineering. The University Board of Trustees opted for the 10% solution. One consequence of the dispute was that some members of the faculty in the School of Engineering vowed to never again speak to faculty members supporting the elimination of the School. At the 25th anniversary of the founding of the Computer Center, I met a former colleague in the Electrical Engineering Department who informed me that he had remained true to his vow. The Computer Center was an administrative service and the only way to reduce the budget was to fire staff. The staff at this time was not large and many members worked many more than the 40 hours a week that they were paid for. Woe is me!

## Chapter 3

### Computing at City University of New York in the 1970s

At that time City University of New York (CUNY) had just committed to open admissions and the student population in its 19 colleges was in the process of doubling to more than 250,000 students. Any New York City student who graduated from high school was guaranteed a seat in one of its Community Colleges or its Senior Colleges. One day I received a call from a CUNY Vice Chancellor telling me that he would appreciate the opportunity to talk to me. When I arrived he announced that the computing situation at the University was desperate and they needed me to fix it. He offered me an appointment as a University Dean. Once again I was offered a job without an interview. In the discussion that followed, he told me that there were no computer wars. Faculty members were free to buy any computer that their grants would fund. Second, that the highest priority at the University was instructional support, followed by enough administrative computing to keep the University afloat. Almost all of the Colleges had small computers supporting administrative computing, but a couple of new colleges would need help. There was no immediate crisis but there were clouds on the horizon. Administrative Computing reported to me at Columbia and I knew that some of those clouds could be pretty black for a rapidly expanding University.

The most attractive feature of the job offer to me was that it rescued me from my Columbia dilemma. If I took the job I could bring with me all the Columbia people that I was faced with firing. But before accepting the job offer I said that I would develop a plan and if the plan were funded I would accept his offer. I knew the computer time used by a Columbia student in a computer course, and multiplied that by a generous estimate of how many CUNY students would be computing if CUNY had a robust computer literacy program. That load alone required two large mainframes. With that capacity we could also support research and administrative computing with no problem. I prepared a budget and the Vice Chancellor explained that the funds needed would not be a problem, because the City had increased the Universities' budget substantially to support open admissions. He walked me in to talk to the Chancellor, Robert Kibbee, who assured me that he would support my plan. I agreed to accept the job. I thought about how I would tell the Vice President that I reported to at Columbia that I was leaving Columbia. One idea that crossed my mind was that I would start by telling him that I had some good news and some very good news. The good news was that he would have no problem in achieving a 10% reduction in the Computer Center budget. The very good news was that I was resigning. But I couldn't do that. I had many great memories of my time at Columbia and I was sad to leave.

I quickly learned that there was one more hurdle at CUNY. All computer acquisitions needed to be approved by someone in the Mayor's Office of the Budget, who ran a data processing operation on a small IBM 1410 computer. Above the door to an office he shared with his deputy was a sign that read: "Abandon all hope ye who enter here" (my memory may be a little vague on this point). The Vice Chancellor arranged for him and me to meet with the Mayor's Director of the Office of the Budget. The Director was a graduate of City College and a jovial fun person to deal with. He agreed that I would have no problems getting any needed acquisition approvals from the city.

My immediate problem was to hire some key staff members and to find a site for the computer equipment that I ordered a few days after I started working at CUNY. After looking at n/e possible sites for the Computer Center, I found a new building on 57th St and 11th Avenue that had vacant upper floors that were for rent. The building had large windows with a magnificent view up and down the Hudson and of Hoboken, New Jersey. After living in a cave during my Columbia years I was determined that the new Center would have windows.

We rented space on the 16th floor and the landlord took responsibility for renovating it to our specifications with the cost being included in the rent. I could not in good conscience hire senior members of the Columbia staff. That could lead to chaos at Columbia. Among the junior staff, Ira Fuchs was a big picture person who was innovative, creative, and very technologically informed. He had been the person responsible for bringing Wilbur and Orville from Stanford to Columbia. He was the most informed person at Columbia about networking problems. Connecting computers at 19 colleges, and to a huge number of terminals distributed across the 5 Boroughs of New York City, to the new CUNY Center was going to be challenging. I was confident that if anyone could solve that problem it would be Ira. He had had no management experience but that was a problem easy to solve. I could tell him to just hire good people and keep out of their way. I hired Ira as Director of the new Center. Ira recruited a number of other junior Columbia employees who were highly talented. To reach the 10% budget cut goal, Columbia did not need to fire any Computer Center employees. They did not replace me for two years. I began to believe that I had not left a vacancy. A decade later, Ira undertook the challenge of connecting terminals and computers all over the world when he started Bitnet.

I had now embarked on a new career as a University bureaucrat. As a University Dean I was a member of the Chancellor's cabinet. My office at CUNY headquarters on 80th Street in Manhattan was several doors down from the Chancellor's. At his cabinet meetings I learned about the important things happening at the University. Most of the time the Chancellor just wanted to talk about the educational issues associated with welcoming thousands of new students to the University, many of them needing remedial instruction. At the midpoint in my career at CUNY I was promoted to Vice Chancellor, and Institutional Research and Television were added to my responsibilities. The Institutional Research Office was staffed with social scientists who evaluated educational experiments and programs across the University. The Television Office provided television services across the University and supported an experiment enabling students at four Colleges to take courses offered at another college connected by television links. Now a conglomerate, I recalled a Bob Newhart line about the confidence he felt when he flew on an airline run by the Thompkins Airline and Storm Door Manufacturing Company.

The social scientists discovered, among other things, that teaching via television worked as well as teaching with all the students in one classroom. The remote classrooms had a telephone link to the instructor, and the instructor could see the student asking the question on his or her television screen. However, the cost of television links was too high to support extending this experiment. An attempt to teach students a subject with computer terminals also produced interesting results. In a mathematics course it was learned that about 75% of the students succeeded in passing the course, but about 25% hated it and did not do well. In another analysis, it was learned that female students from deeply impoverished neighborhoods did much better than males from the same neighborhood. In particular, a single parent mother on welfare did much better than her male peers with a similar high school record. By controlling for as many variables as they could think of, they concluded that this result could only be explained by motivation.

When I arrived at CUNY I recruited a very talented group of people to modernize the Universities' administrative systems. Some of the systems they created survived for more than 30 years after I left CUNY. I spent a lot of time traveling to the Colleges to talk to administrators at all levels, faculty, and occasionally to students. This meant that I spent a lot of time in New York City traffic. As a perk, I had exclusive use of a City-owned car with City license plates. Those plates enabled me to park anywhere in the city without fear of getting towed for illegal parking. On one occasion my car broke down on the Long Island Expressway. I abandoned it and another car was delivered to me. Occasionally I would visit the Computer Center, which was an oasis away from the many problems dealt with by the people at 80th Street. I discovered that the overhead associated with being a bureaucrat was very high and not a lot of fun. In particular, trying to move hundreds of faculty that didn't report to me was like trying to herd cats.

## New York City Goes Broke

As the old saw goes: “Cheer up, things could be worse! So I cheered up and sure enough things got worse”! In the summer of 1975 it was discovered that the City was on the brink of defaulting on its bonds. The Federal Government and the State stepped in and imposed stiff conditions for guaranteeing the City loans to avoid default. An Emergency Financial Control Board was created by the State Legislature to review and approve every financial decision by the City. Employees of CUNY were not paid for several weeks until the State relieved the City of responsibility for funding the Senior Colleges and assumed control. CUNY had been tuition free since its creation. The State and City decided to end that practice.

In the fall of 1975 the Chancellor informed me that Abe Beame, the Mayor, had called him and that the Mayor would like to talk to me that afternoon. I traveled down to City Hall and the Mayor told me that consultants had reviewed the City’s computing infrastructure and concluded that it needed a lot of fixing. In particular the computers in the City did not talk to each other, and worst of all the Budget Office computer didn’t talk to the Controller’s computer. The Mayor controlled the city budget and the Controller paid the bills and did the accounting. The Mayor and Controller were independently elected officials. The inability of the Budget computers to talk to the Controllers’ computers was a major source of the City’s financial problems. They recommended the City create an Office of Computer Plans and Controls to fix these problems. He said that the Chancellor had agreed to loan me to the City if I were willing to head this office. I assumed that I had been selected because I would be free, and of course everyone knows that you get what you pay for. After a moment or two of contemplation I decided to accept this task because it might have high anecdotal value, and it certainly did.

While Abe Beame was Mayor I reported to him. When Ed Koch succeeded him on January 1, 1976, he created an Office of Operations, reporting to him, which consisted of 30 high-level executives on loan from major New York City corporations. The lending corporation paid their salary just as CUNY paid mine. The task of this group was to look at every operation in the City for opportunities to find cost savings through improved efficiency and reorganization. Lee Oberst from AT&T was appointed as the Director of the Office. I became attached to the Office as Deputy Director. Lee was an efficiency expert. He believed that in analyzing an operation you started with the people at the bottom of the organization who were doing the work. You watched them work and measured their output and compared it to that of commercial company employees doing the same work. You also looked for ways to increase their productivity. You then worked your way up the organization toward its top management with the goal of streamlining it. The last person you talked to was the person managing the whole operation. Needless to say, this created great anxiety among City Managers at every level.

The first assignment for two members of our group was to follow garbage trucks on their collection routes. They discovered that there were some trucks that deposited almost as much garbage on the city streets as they collected. The result of their work was the reduction of staff on each truck by one, and instead of enjoying a couple of weeks off following a heavy snow storm because the streets were impassable, garbage trucks were fitted with plows and when it snowed they were out plowing their routes. A few years earlier, after a heavy snow storm, the *New York Times* interviewed a person in Queens two weeks after the storm, who said: “The only thing moving out here are the children”. With 4000 garbage trucks equipped with plows that never happened again. However, for some inexplicable reason, the image of high-level executives in expensive black suits, white shirts, and expensive ties following garbage trucks and lurking in halls at night to watch janitors still causes me to chuckle. But amazing things were learned and lots of things changed.

I did not have to watch civil service programmers to know that they were not up to the task of rewriting the City’s administrative systems. At CUNY I had succeeded in getting all the programmers that I hired classified as academic support staff to avoid the necessity of hiring people bound by civil service rules. During my two years at the City, all system design and programming work for the City was contracted to commercial companies. Armies of programmers across the country were hired to rewrite all the key systems and a number of

new programs that emerged from the work of the Office of Operations. We worked long days. Lee convened meetings of our group at 6 AM or 7 PM so that no one could claim a competing commitment.

After my appointment as czar of computing I began working on a plan to fix computing in New York City. Consulting the best minds at CUNY, Ira Fuchs and Mel Ferentz, a computer scientist from Brooklyn College, we worked on a plan for supporting a transaction-intensive environment. Mel and Ira and I had earlier worked together to bring Unix to CUNY. I signed the first contract exporting Unix from Bell Labs to a University. At that time the Digital Equipment Corporation (DEC) was developing an operating system that could support an extend-able bunch of Vax computers on a ring network. The system was specifically designed to support transaction-intensive environments. It looked like a winner. I developed a Vaxen plan and asked Lee to check it out with the people at Bell Labs. He reported that they loved it. I asked Lee to join me in visiting DEC to see what state the system was in. At DEC headquarters their systems and sales people told us how awesome it was. Remembering the first two years of OS/360, I then asked if the company would be prepared to guarantee that it would work reliably from day one. After some discussion the sales people decided to bring Ken Olsen, DEC CEO, into the discussion to close the deal. They huddled with Ken outside the door to our conference room and explained the deal to him. After about 5 minutes Ken opened the door, looked at us, and uttered a single word: "No"! It was back to the drawing board.

I then designed a system built around large IBM mainframes. I had no problem getting the Mayor, Controller, and City Council to approve the plan. I then did my n/e thing and found a great site for a new NYC Computer Center. The site on 10th Street had formerly been a bus terminal. It had elevators designed to transport buses from floor to floor. The city rented a floor and the Controller rented the floor above to house their computer. It was now easy for the two computers to talk to each other and they were able to share communication connections to City Departments. You could drive your car into the building and into an elevator, and the elevator could drop you and your car off at the door to your office. How's that for a world-class perk? I recruited Joe Giannotti from the CUNY Computer Center to direct the new City Center, which supported the agencies reporting to the Mayor. Joe had great political as well as technical skills. Joe hired some extraordinarily good people from Columbia and CUNY to staff the new City Center.

Two years after the City was on the verge of default it was relieved of most of the restrictions imposed by the Emergency Financial Control Board, and everything again became immersed in politics. Lee went back to AT&T and I recruited a successor czar and went back to CUNY. A few years later, as a measure of how successful and important the new Center had become to the City, Joe Giannotti was promoted to the level of a Commissioner. At that level he enjoyed a car that had a flashing red light on its roof and a siren. It was always a pleasure to travel around the city with Joe. My experience over my two years as Czar was exhausting but rich in anecdotal value. I learned that some things are only accomplish-able during a period of financial crisis.

The imposition of tuition caused a dramatic drop in enrollment at the CUNY Colleges. Budgets across the University were slashed. Even tenured professors were fired. When I returned if I asked people how things were going the response ranged from: "don't ask"! to "terrible"! to "oy vey ist mir"! The budgets of the offices reporting to me were not cut. It was good to be at the front of the trough when the slop was poured in. Everyone in the Central Office was engaged in damage control. Gradually the situation improved but those were not fun times. At the Computer Center we managed to sell an IBM mainframe and from the proceeds of the sale replace it with an Amdahl (an IBM clone) that was 30% faster. This taught me that it was wise to sell and replace computer equipment while it still had value, if you could manage that. When IBM announced the 3000 series in the late 70s we managed to replace the small very aged computers at the Colleges with 3000-series computers that were 10 times faster. As the decade of the 70s ended we began to populate the Colleges with a small number of Apple II's as the wave of the future.

It is difficult for me to assess the advances in computing during the 70s because I no longer worked in the trenches. Computers got faster and operating systems improved and computer-related instruction grew dramatically. Even Columbia had established a Computer Science Department and many Universities had computer literacy requirements. On the hardware front, the number of transistors in integrated circuits on a chip had been doubling every two years, as predicted by Gordon Moore in 1965. This was known as “Moore’s law”. It was now possible to put a computer on a chip. Advances in chip density would soon dramatically change computing everywhere. Three new “killer apps”, email, spread sheets and word processing began to become important. Many Universities had local email systems starting in the 60s. They were important in eliminating telephone tag. But not many faculty members had terminals on their desks. In the 1970s, ARPANET proved that people would like to talk to people at other Universities, but this was not widely feasible at Universities until the emergence of Bitnet in 1981. The VisiCalc spreadsheet program for the Apple II was a clear winner at Business Schools. Wang word processors began to appear in departmental offices, but word processing became important to faculty members and students in the 1980s, when microcomputers attached to networks appeared everywhere on campuses. It was clear to me that the 1980s would be a lot more exciting than the 1970s at Universities.



## Chapter 4

### Computing at Cornell University from 1980 to 1987

In 1980 I received a call from Cornell University inviting me to visit the campus because they were recruiting a Director of Computing and several people had recommended me to them. CUNY had a relationship with the Mt. Sinai Medical School and there were close connections between Mt. Sinai people and Cornell Medical School people. Both Schools were in Manhattan. Through this grapevine I had earlier learned that, to use the technical term, the Cornell Medical School had been “screwed” by the Cornell Computer Center. The Cornell Medical School had been forced to move its computing off a local machine to the campus computer because the campus center was in financial trouble. It turned out that the campus computer had been unable to accommodate their load and the whole deal needed to be undone after the Medical School had sold their computer. Thus I considered Cornell as a place for which the 40-foot pole was invented, a place that you wouldn’t touch with a 10-foot pole. However, it was always pleasant to visit a campus and I agreed to talk to them.

Bob Cooke, who was the head of the search committee, picked me up at the Ithaca airport. As we headed toward campus Bob explained to me that Cornell was a very complicated place with nine Schools, each with a large amount of autonomy. I told him that I had had some experience with complicated places. He then told me that the University Computing Board had persuaded the Provost to create a Decentralized Computer Support group in Computer Services to encourage and support faculty acquiring their own computer. The computer wars were over at Cornell. As we crossed a one-lane bridge entering the campus, my view of Cornell changed 180 degrees. I told Bob that I thought that microcomputers were the wave of the future, and that minicomputers and microcomputers were the only hope of greatly increasing the involvement of faculty members in computing.

During the visit I learned that high-level Cornell executives on the financial side had back loaded the amortization of the campus mainframe in the expectation that sponsored research would significantly increase as the years passed. When this didn’t happen they had a problem. The campus mainframe was about 10 times faster than the Medical School computer. Someone decided that selling the Medical School computer and moving Medical School computing to the campus could solve the financing problems of the Ithaca mainframe. The campus computer had enough spare capacity to easily support a 10% increase in its load. The campus mainframe was at the airport, five miles from the campus, so the computer center staff had experience in supporting remote computing.

This solution looked as easy as shooting a pheasant that was walking toward you through a rain barrel with a rum raisin in its mouth. The move went forward without any benchmarking. Then, as they say, something funny happened. It turned out that administrative systems at the Medical Center had been written in APL square code. In APL it was possible to write dense logical code on a single line that would compile into many instructions. A special typewriter with its own unique keyboard and characters produced APL code. Bright programmers sought to get as much logic on a single line as possible, stretching the code to as close to 80 characters as possible. The challenge was to write a program with lots of long expressions. This was called “square code”. Evidently the Medical Center’s programs had been written by a genius. Some years earlier I had visited the IBM Yorktown Research lab and decided to drop in to chat with Ken Iverson, the author of APL, and a colleague of mine when we were both on the faculty of the IBM System Research Institute. While we chatted, someone approached Ken and asked if he could interrupt our conversation for a minute. He had a long length line of APL code that had a logic error and he needed a minute of Ken’s time to figure out where. After about

10 minutes of staring at the line, Ken figured out where the problem was. The Medical Center had acquired a computer that had microcode that increased the speed of APL programs by a factor of 10. The campus mainframe was not a model that supported this microcode. The campus mainframe was brought to its knees trying to run Medical School APL programs. Apparently this episode did not make the Cornell Trustees happy and the campus was in search of new computing leadership.

I had a very pleasant day on the campus. Everyone had good words to say about the technical competence of the computer staff and several faculty members said that Douglas Van Houweling had been particularly helpful to them. Everyone thought that computing needed more financial support from the University. At the end of the day I was taken to meet Keith Kennedy, the Provost, and he indicated that he thought that the quality of computing at Cornell was very important. He walked me across the hall to meet Frank Rhodes, the President. About a week later I received a call from Keith telling me that having completed the search process, the search committee had recommended that Cornell hire me. I told him that I would think about it.

The overhead associated with being a bureaucrat had become higher at CUNY when the State took over funding the Senior Colleges. I now had to travel to Albany to clear computer acquisitions and funding that related to them. As I weighed the delight of again being able to lunch with faculty members trying to change the world and the great view of a lake from the Cornell campus against the prospect of endless drives to Albany, I decided to accept the Cornell offer if they agreed to accept a plan that I would develop. I called the Provost and told him that I needed to spend a couple of weeks on campus to develop a plan and that I would come to Cornell if my plan was accepted. I warned him that I would be unwilling to come to Cornell unless they aspired to having a first-rate computing environment and that would not be cheap. He told me that he thought that Cornell was ready to spend more money on computing and looked forward to getting my plan. He arranged to put me up in a house on campus once occupied by Cornell Presidents. It seems that the President's house had moved off campus too. During my time on campus I talked to all of the Deans and found that some were more enthusiastic about computing than others, but none were resistive. Following the wisdom of Lee Oberst's great teachings, I had hoped to talk to them last but scheduling difficulties prevented that.

In talking to the faculty I learned that Cornell had a great Computer Science Department established in 1965. Better still, the person teaching the Introductory Computer Science course was Tim Teitelbaum. Tim had been one of the people living at night in the back of the Columbia computer room. At Columbia Tim was working on a Hough Powell device that was channel connected to the center computer. The device scanned film of particle collision events in Cloud and Bubble Chambers and digitized particle tracks. I subsequently discovered that Tim deserved a great teacher award. Tim was the dynamo at Cornell behind the computer literacy efforts in the Computer Science Department. He had written a PL1 program synthesizer for a Terak microcomputer and was in the process of converting instructional computing in the department to microcomputers.

As I talked to faculty doing research I discovered that faculty running their own computers were happy with the support provided by the Decentralized Support group led by Doug Gale. A small group of faculty in the sciences led by Ken Wilson, who soon thereafter won a Nobel Prize in Physics, had cooperatively purchased a Floating Point Systems (FPS) 190L array-processing computer that was attached to a channel of the campus mainframe. The mainframe provided input and output services for this computer that was designed to support computationally intense programs. Its computational engine was as fast as the mainframe's and could sit in the Computer Center running 24/7 under the exclusive control of a few scientists. Better still, the charges to their grants for mainframe time were small because the mainframe did not need to provide a lot of cycles to keep the array processor humming along. When asked about their future computer requirements, they told me that they could use a number of Cray Supercomputers. Doug Gale remembers that the number was five. The faculty members using the campus mainframe were not happy with turnaround time, and there were communications problems exacerbated by the fact that the computer and most of the Computer Center staff were five miles from campus. Faculty members I talked with vividly remembered the attempt to run Medical

School computing at Cornell. By this time in my career I had learned that faculty were far happier and more productive if they controlled the computer resources they required. Putting them in an environment where their research was constantly interrupted by hardware, and system changes, and other inharmonious events made them less productive and unhappy. Often, a brief bit of experience in an environment they shared would teach them to avoid computing in their research and instruction if they could. There was some support for faculty not enjoying grants containing funds for computing time that was arranged in some mysterious way by Doug Van Houweling, who managed Academic Computing Support in Computer Services.

The Administrative Computing staff lived in a basement and the staff was small. There was a room with a bunch of keypunches in it at the entrance to their space. That told me almost all that I needed to know about the state of administrative computing at Cornell. Staff morale was low. A few years earlier, Cornell had experienced a budget crisis and a committee was appointed to search for savings. According to administrative computing staff members, they recommended the elimination of administrative computing. When they discovered that this would make it impossible for Cornell to pay its faculty and staff, they recommended exempting the payroll staff from their recommendation. I suspected that this story might have been a bit exaggerated but the result of the committee report was downsizing the staff by 40%. The staff that survived could tell me the names of everyone on that committee, and some of the people who joined the staff after this event had learned their names too. Thus their legacy was preserved.

Computer Services probably had the best Virtual Machine (VM) systems' programming staff in the country. There were two major IBM operating systems at that time and the one with the most interactive features was VM. Bob Cowles, who led this group, was a product of the Cornell Computer Science Department as were a number of other people in Computer Services. He had written a VM scheduler program that was widely used and in my view clearly superior to the IBM-provided VM scheduler. Dick Cogger, who was innovative and creative in supporting the campus connections to a computer five miles away, led the networking group. Overall, the staff was small but contained some very talented people. Its morale was low because many people on campus blamed them for the Medical School fiasco. However, with proper support they were clearly up to the task of rapidly improving the quality of computing at Cornell.

In assessing the computer problems at Cornell, I concluded that Cornell was a "poster child" for the shortsightedness of Federal Funding Agencies in not allowing the inclusion of computing charges to sponsored research projects in the indirect cost pool. As a consequence of that decision, many Universities had organized Computer Centers as enterprises, which were expected to earn enough revenue to cover their expenses, except for as small a contribution as possible from general revenues. All computer time was billed. At Cornell, the vicissitudes of depending on the sale of time to sponsored research caused the Computer Center to scramble to sell time to outside groups. When this failed and a budget deficit resulted, often heads rolled. Across the country uncountable amounts of computer time were wasted because sponsored research projects, unable to get sufficient funding in their grants to pay for computer time, were denied access to the campus computer. In addition, in determining the hourly rate for time, all Computer Center costs were bundled into a cost pool to be divided by the time available. This included every employee of the Center. Service Bureaus were selling time for a lower rate because the only staff they needed to support their enterprise consisted of a few operators and a few sales and technical support people. In my view, not using the Library Model to finance Computer Centers was a national tragedy.

I prepared a plan for improving computing at Cornell. First of all, computer equipment would have to be amortized at a rate that reflected its resale value. Ideally the University would get first in line as a new generation of computers was introduced and the equipment it replaced would be sold while it still had value. The goal was to insure that the University maintained state-of-the-art computers. Secondly, the University would have to create a number of computing support rooms containing networked terminals and microcomputers distributed across the campus and into the dormitories. The number of these computer-access rooms would have to be

increased as an aggressive effort to infuse computing throughout the University curriculum increased support requirements. The third goal of the plan was to insure that every Cornell graduate had received training that enabled them to understand the role of computing in their discipline. Thus the notion of “computer literacy” went well beyond teaching them how to program and supporting the requirements of the Computer Science Department.

To infuse computing broadly into the curriculum, the University had to get as many faculty members as possible computer literate. This could not happen if the primary computer resource on campus was a batch processing system. The University would have to find a way to support faculty access to computing in every discipline, including disciplines that did not enjoy grants enabling them to buy computer time. An important goal was to find ways in which computing could be used to improve faculty research productivity. Fourth, the University needed to rewrite most of its administrative computing systems to provide online access to data to every office requiring that access. This would require significant expansion of the administrative computing staff. Finally, I had two nonnegotiable demands. The first was that I report to the Provost and the second was that the Computer Center would be moved back to the campus from the airport as soon as suitable space could be found. Ideally this space would be close to the center of the campus.

I presented my plan to the Provost and he said that the University would review my plan and that I would hear from him shortly. A few days later I received a call from the Provost at 8 AM telling me that the University would accept my plan and I agreed to accept their job offer. I felt a lot of loyalty to CUNY and I was sad to leave. When I resigned, I recommended that Ira Fuchs replace me as Vice Chancellor. The Chancellor agreed and it was a wise decision. A year later Ira started Bitnet, one of the transformative events in the evolution of technology. Bitnet proved that networking was important to Universities and laid the groundwork for the creation of NSFnet, which laid the groundwork for creating the Internet.

I was the Vice Provost for Computer Services at Cornell from 1980 until 1987. When I left Cornell, the Computer Services budget had grown by a factor of three during that period. Cornell had over 600 microcomputers in rooms across the campus and in the dormitories. Many students were buying their own microcomputers at a store run by Computer Services. An important goal was to make available microcomputers costing no more than \$1000. We fell slightly short of reaching that goal due to a high New York State sales tax. Cornell had become one of five National Supercomputer Centers. All the buildings on campus including the dormitories had been rewired. The wiring plant included fiber, coaxial cables, and twisted pair. The cable supported access to television in student rooms via a connection to the local cable company. The computer center had moved from the airport to a renovated building near the center of the campus fulfilling a promise made to me by the Provost when I accepted his job offer. A building had been constructed adjoining the Computer Center to house a University-owned telephone system. With the University, rather than the telephone company, owning the wire that reached every faculty and staff member’s desk, a large number of faculty and staff had a microcomputer on their desk connected to the University network.

The University mainframe had been replaced with the latest large IBM mainframe. When it was acquired we spoke of it as the last University mainframe. It was augmented with several medium-size computers that served the computing needs of a subset of computer users. One of them was shared with The Cornell Institute for Social and Economic Research (CISER). This significantly improved the productivity of the CISER members. The administrative systems of the University had largely been rewritten to support online access to a central database and the University had an online Library catalog system.

As an historical note, in a speech at Cornell a number of years later, Bill Gates said that a Microsoft employee, who was a Cornell graduate, returned from a campus visit in 1986 and announced “Cornell is wired”. This caused the company to start developing a networking capability in its operating system. As another historical note, the last campus mainframe was retired in the summer of 2013. When I left Cornell, I judged its comput-

ing environment to be first-rate. This was enabled by unwavering support from Provost Keith Kennedy and his successor Bob Barker; support at critical times from a faculty advisory committee initially chaired by the Dean of the School of Engineering Tom Everhart, followed by Bob Cooke; and an extraordinarily talented and creative staff.

## Major Computing Events at Cornell

John Rudan has written a *History of Computing at Cornell*, which is in the Internet-First University Press (IFUP) section of the Cornell eCommons. You can find his history at: <http://hdl.handle.net/1813/82>. As a side note, Bob Cooke and I created the IFUP as part of a project that we started in 2003. A section of this Rudan history covers the details of staff organization and hardware acquisitions, and catalogs major events during the very busy period that I was at Cornell. In this memoir I will only discuss the background of major events that moved computing at Cornell, sprinkled with some anecdotes designed to convey the issues faced by people involved in University computing at that time.

In 1981 Ira Fuchs at CUNY persuaded Grey Freeman at Yale to share the cost of connecting their computers via a 9600 baud leased line to create a store-and-forward email network that Ira hoped would one day connect every scholar in the world to every other scholar. The network protocol they used was developed at IBM and an IBM group had created a small email network using this protocol operating under the radar at IBM. Ira had a friend at IBM from the time that they both were employed at the Columbia Computer Center who informed him of this development. Ira called this network Bitnet and as the network came up Ira began calling me regularly to find out when Cornell would connect. Because of a peculiarity in the telephone tariff rate for leased lines, it was cheaper to lease an interstate line than an intrastate line. When Penn State University connected to Bitnet, Cornell connected to Penn State becoming the 6th University to connect to Bitnet. Shortly thereafter, the University of Toronto connected to Cornell becoming the first international member of Bitnet. About this time Ira reinterpreted the name to mean: “Because It’s Time” network. Soon California institutions connected to each other and to CUNY. The network began to grow rapidly. University College Dublin was the first European institution to connect. The Director of Computing at the College, Dennis Jennings, later became responsible in the NSF for networking Universities to the National Supercomputer Centers to create NSFnet, which evolved into the Internet. Bitnet spread across the US, South America, and Europe. There was even a connection in the Soviet Union. When the Tokyo Institute of Technology leased a line to CUNY, it spread to Asia. A networking history document in Japan displays the first message from a Professor at the University to the US. It was a message to me from a Professor that I helped when he computed at the Watson Laboratory in the 50s. The capability of Bitnet was soon augmented to support email lists and instant messaging.

In 1987 people on Bitnet received a message from a student at Clausthal University of Technology in Germany asking them to execute an embedded program to print out a Christmas tree and receive a Christmas greeting. The program was able to retrieve your Bitnet address file and forward this message to everyone on your list. The resulting traffic created by this “worm” brought down Bitnet and all the commercial networks connected to Bitnet. In 1988 a Cornell computer science student distributed the first Internet worm. The student had received his undergraduate degree at Harvard. People at Cornell observed that students who got their training in ethics at Harvard and their training in Computer Science at Cornell could be very dangerous.

At Cornell, aside from the Cornell Computer staff, who benefited from connections with their technical counterparts at other Universities in order to share solutions to common problems, the early users of Bitnet were students who quickly discovered Bitnet and were used to hanging out in network-connected terminal rooms that were beginning to spread across campus. Faculty members soon discovered that they could undertake collaborative research with faculty members at other Universities without having to play telephone tag or use snail mail to exchange papers. Long distance calls were expensive in those days. Bitnet was a free service. It cost the University \$5K per year for the leased line to Penn State. Faculty members began to include their Bitnet address

on their business cards. When the campus was rewired to bring a network connection to every desk, a major incentive for getting a microcomputer on their desk was that it enabled email connections to their colleagues at Cornell and around the world using Bitnet. A study conducted in the late 80s showed a large increase in the number of papers authored in Journals by people from more than one University. This increase was attributed to Bitnet.

An early effort involved finding ways for faculty members without access to computing funds to compute. We created a program called “Windfall”, which enabled them to use weekend computing time on the mainframe that would otherwise be unused and wasted. Since all computing time had to be billed, we created a very low rate for this time and the Provost provided funds that were turned into “funny money” that was allocated to faculty to pay for this time. It was called “funny money” because it could only be used to buy computer time on weekends. After we started this program, a faculty member’s wife complained to me that she rarely saw her husband on weekends because he had become a wind-faller.

Cornell had a very creative Controller named Jack Ostram. In 1962 Jack was at Princeton and the person Columbia financial people worked with to get computer costs into the indirect cost pool for a few years. Jack reduced the cost of computing time by stripping out of the cost base, used to determine the hourly rate, everything except the direct cost of providing cycles. All other costs were put into something he called “the glob”, and these costs went into the University’s indirect cost pool.

In late 1983 Dan’l Lewin showed up on the Cornell campus with a small metal box handcuffed to his wrist. In the box was a Macintosh computer. He demonstrated the little jewel to me and I immediately knew that the computing world had changed forever. It had a graphical user interface with icons on a bit-mapped screen and a mouse as a pointing device. It came with only two applications, MacWrite and MacPaint, but Lewin explained that many more were under development. It introduced a new word into the computing vocabulary, WYSIWYG, “what you see is what you get”. Lewin, who was a Cornell graduate, told me that Apple was organizing a group called the “Apple University Consortium”, which would consist of 24 of the country’s most prestigious Colleges and Universities. The group would be advisory to the company and enjoy a special discount on Macs. He invited Cornell to join and without hesitation I accepted. Later that evening I called Ira Fuchs and told him that he should try to get CUNY into the Consortium. The next day he called me to report that he had talked his way in.

Shortly thereafter, a complimentary Macintosh was delivered to Cornell. I decided to demonstrate it to faculty and computer staff people present at our 5 PM Friday Wassail. People played with it and were able to navigate without any instruction. There was high excitement and glee among everyone who watched. At some point I decided to show them the 3.5-inch floppy that contained its operating system. An attempt to eject the disk from the Mac failed. I picked up my phone and dialed Steve Jobs’s telephone number. This number had been provided to Consortium members. To my surprise and delight Steve answered the phone. I explained the problem. He told me that there was a small hole under the disk and that I should insert a paper clip wire into that hole and the disk should come out. He said that he would stay on the phone until I reported success. I inserted the wire and the disk popped out. The people there cheered. I got back on the phone and told Steve that it worked just like magic and thanked him. At that moment I recalled the IBM hardware designer who left the Columbia Computer Center without waiting to see that his advice had solved a problem that had kept the 360/91 in the center down for 3 days.

I started the Friday wassail tradition while I was at Columbia in an effort to gather data that might be useful in a stress management component in a Life Science course. The wassail was presided over by a non-management employee called the “Chief Degenerate”. On Friday at 5 PM I set a modest amount of booze out on a table in my office and when this was done it was the Chief Degenerate’s responsibility to stand at my office door and blow a loud whistle called the wassail whistle. Anyone hearing the whistle was welcome to come and have a drink. On

one occasion at CUNY, a person from Brooklyn College arrived a little late. He explained that he was on a subway platform in Brooklyn waiting for a train to Coney Island when he heard the whistle and decided to cross the platform and catch a train to Manhattan instead. The Chief Degenerate had life tenure unless promoted to management and had sole authority to throw someone out of wassail or even ban someone from wassail. It was his responsibility to enforce my only rule: everyone had to make his or her own drink. No one was permitted to make a drink for someone else. I only remember one incident in which the Chief Degenerate exercised his awesome power. Someone on the Center staff started to bad mouth a staff member who wasn't there and the Chief Degenerate told the person that he would be ejected unless he changed the subject. I suspect that that person would not have evoked this reaction if he had been badmouthing a member of the administration instead of a member of the staff. I learned a lot from the wide range of people who attended Wassail. It was an event at which every level in the organization could meet. On one occasion at CUNY, I had both the Chancellor and the building janitor at Wassail. The Chancellor asked the janitor how many keys he had on his belt. On another occasion a fellow Vice Chancellor arrived with a distinguished poet. The poet asked me who my favorite poet was? I replied: "Robert Service". He replied: "that's not poetry, its doggerel", and set down his drink and left. I wondered what his response would have been if I had replied: "Robert Frost, Dylan Thomas, or Omar Khayyam, among my other favorite poets. But it's hard to beat an opening line like: 'A bunch of the boys were whooping it up in the Malamute saloon'".

Over time we populated a lot of rooms across the campus with Macintoshes. One of the sites was in the Undergraduate Library. We persuaded the Undergraduate Librarian to make a small room in the Library available for Macs. We installed about 15 of them one evening and the next morning I got a call from a Librarian. She said: "You better get over here in a hurry. There's a serious problem in the Mac room". When I got there I discovered 15 students working on Macs and another group sitting on every free space on the floor waiting for access to a Mac. The Librarian thought that they were occupying the floor space as part of a protest movement. After that, every time I saw her we started to laugh.

During the years that I was at Cornell we had a lot of small renovation projects as we acquired space for computers. One very big effort involved renovating an old building on campus and moving the computers and staff from the airport to campus. John Rudan managed these projects in addition to managing the Systems and Operations staff. One day a fellow Vice Provost told me that she had been waiting in a queue for 6 months to get her office painted and wondered how I managed to repeatedly jump everyone in the queue. She suspected skullduggery. On an occasion somewhat later I decided to accompany John on a visit to an office to reinforce the urgency of their providing a critical service for us promptly. John entered the office and greeted the 3 people in the office by name. He not only knew their names but the names of their wives and children. He asked every one of them for a family update. He then got around to discussing the reason we were there. They promised to start the next day. I never had to open my mouth.

Cornell had discovered that most students arriving at the University had poor writing skills, so every freshman was required to take a writing course. As students started to use word processors, someone raised the issue of whether or not a word processor improved or diminished a student's writing quality, so an experiment was initiated. A randomly selected group of students were allowed to use a word processor while another randomly selected group was required to use a conventional typewriter. It was discovered that papers from the conventional typewriter group were clearly superior to those of students using word processors. In looking into the issue, it was discovered that students using a typewriter spent a lot more time on their papers than the word processor group. Because their papers required many retyping's, they typically started work on their papers on Monday if the paper was due on Friday. The word-processing group had adopted a "just in time" model and often started work on the paper late on Thursday. The typewriter group thought about their paper as they walked across campus. After lots of discussion, it was decided that it was futile to ban the use of word processors. Instead the course was modified to require revision of papers that were less than perfect. A paper was marked up

and returned to the student for correction. Students were required to submit their papers electronically rather than on paper to facilitate the instructor's revision process.

One of my first tasks at Cornell was to rebuild the Administrative Data Processing Group and to undertake an effort to put all of its important shared data online and accessible from terminals. To manage this effort I hired an extraordinarily good person named Russ Vaught who at the time was Director of the State University of New York at Binghamton's Computer Center. Building information systems at a University is very hard. While working in the Office of Operations in New York City, we started at the bottom of an organization and worked our way up with the goal of saving money by automating procedures and streamlining the management structure. At Universities, collegiality required starting with the Vice President responsible for an administrative service and working your way down. It seemed almost never possible to acquire a commercially available system, even if good ones were available, because they made the paper flow in the office different. From the perspective of office managers, the goal was to add new services that enhanced the value of their office to the University. These new services almost always required adding staff in the office. A new system was always expensive to build and the end product usually improved service but cost more to operate than the system it replaced. That was true at most major Universities.

A major long-term controversy at Universities was the question of whether or not both academic and administrative computing should report to a University Chief Information Officer (CIO). In the 70s, as computing became more important to Universities, Computer Center Directors began to be appointed above the level of buildings and grounds, and there was a strong trend to merge academic and administrative computing under a CIO. My own view was that if they shared a large campus mainframe there were advantages to a unified management structure. However, like research faculty, administrative offices always seemed happier if they controlled their own hardware.

I learned a few things from Russ. He was constantly working on staff development. He explained to me that it was important to send his staff to opportunities for training and education. He would say: "We've got to invest in our seed corn." He quickly developed a very talented staff. In the early 80s, mainframe computers had far less CPU power, memory, and storage than a desktop PC today and the whole university shared that computer. Thus it was easy to undertake projects beyond the capability of the hardware. There were examples of large administrative systems started at a University by someone who could be characterized as possessing infinite optimism bounded only by unlimited self-confidence. These projects always ended badly. Russ carefully analyzed the requirements of every system that he was asked to implement before he committed to develop it. Russ tested new technologies in something called his sandbox. One of the technologies in his sandbox was object-oriented computing. This technology later became very important to supporting the software maintenance of microcomputers on campus. Academic computing was always a lot more fun than administrative computing and when Russ left Cornell it was to become Director of Academic Computing at Penn State.

An exception to the usual decision to build our own was the acquisition of a commercially available online Library Catalog access system. A committee was created, co-chaired by Jan Olsen, Director of the Mann Library at Cornell, and me to select a system. We visited a number of Universities that had installed a system or were in the process of installing a system. At one University that we visited, academic computing reported to the Director of the Libraries. I asked the Director when she thought the library system they were implementing would be up and running. She replied: "When it's ready. It's taken 200 years for this library to get where it is, and if it takes another 200 years to implement this system, that's how long it will take".

As we started our work, I was asked to explain the goals of the effort to a faculty Library Advisory Committee. I decided to start my presentation with a bit of humor. I quoted a Harvard Librarian who said: "The goal of a library automation system is to keep the faculty out of the Library". No one laughed. I got that "what planet is this guy from" look. As this memoir is being written, Cornell has recently closed libraries in the Physics and

Chemistry Departments and the School of Engineering because of lack of use. Paul Ginsparg, who created the arXiv system making preprints of papers available in mathematics and the sciences on the Internet, was recently quoted as saying: “today, its Google or forget it”. ArXiv has catalyzed advances in Physics by making preprints of papers available months before they appear in Journals. Access to information on the Internet from the comfort of their offices and home is apparently now keeping the faculty out of the library.

In 1984 the IBM President visited a number of Universities and was shocked to discover that their student computer rooms were predominantly stocked with Macintoshes rather than PCs. So that same year IBM started an Advanced Education Project (AEP) with major grants to 19 Universities including Cornell. With this grant Cornell acquired more than 500 PCs over the next three years. Under the terms of the grant, most of these computers were given to faculty members in return for a commitment to develop instructional material for one of their courses that used a PC. We named this project “Project Ezra” after Ezra Cornell, the founder of the University, and Cecilia Cowles managed it. We always included the Cornell Medical School in programs directed toward faculty, and they were recipients of some of these PCs. We quickly succeeded in establishing excellent relations with the Medical School. Faculty members developing software were mostly quite senior with tenure. It was generally discovered that it was a big mistake for junior faculty members to interrupt their research careers by spending time writing instructional software. The Ezra staff members at Cornell were fierce competitors and Cornell always had more projects selected for presentation at AEP meetings than other AEP Universities. I was fond of quoting the first law of the Yukon to staff members: “The scenery changes only for the lead dog; if you’re second or third the view can be fairly monotonous”.

As Cornell faculty members began developing software for microcomputers a controversy developed about who owned the software. In the mainframe days, commercially marketable software produced by University employees was treated as “work-for-hire” and copyrighted in the name of the University. Historically, software that had the biggest impact was “open source” software that was freely distributed and collaboratively developed by its users. Berkeley Unix was a prime example. As faculty members at Cornell began to develop software for microcomputers, the Cornell Trustees insisted that that software be copyrighted in the name of the University. Faculty members argued that software for microcomputers should follow the paradigm of faculty produced books, which were copyrighted in the name of the author. As the controversy raged, faculty-members took their efforts off campus and a number of privately incorporated companies were created to develop and market their software efforts. After I left Cornell the Trustees realized that the effort to own microcomputer software was hopeless and gave up.

One of the super techs on the Cornell staff when I arrived was Steve Worona. I offered him a management position but he said that he did not want to manage people so he reported to me. Steve had written the campus email system some years earlier and he maintained it and extended it. While I was at Cornell, Steve created two new systems called CUINFO and “Dear Uncle Ezra,” both of which still survive. CUINFO contained general information about Cornell of interest to faculty and students. “Dear Uncle Ezra” was a section in CUINFO that allowed students to submit questions to an anonymous Assistant Dean of students. Stripped of names, some of these questions and their answers were published in “Dear Uncle Ezra”. Sometimes the questions indicated that the student submitting the question was deeply depressed and possibly suicidal. In that case the Assistant Dean took immediate action to help the student. In other cases the answer to the question required some research and consultation with experts before an answer was published. One such question was: Where’s the best place in Ithaca to get “soul food”.

In the middle 80s we created a project called “Broad Jump”. The notion was that every “knowledge worker” needed a computer on his or her desk. The goal of the project was to put a computing device on the desk of every high-level administrative officer at the University. Steve Worona and Cecilia Cowles were assigned the task of interviewing University Officers to determine their computing requirements. They started with the President, Frank Rhodes. They began their presentation to Frank by saying: “Every knowledge worker needs a computing

device on their desk”. Frank stopped them right there. He said: “I’m not a knowledge worker, I’m a wisdom worker”. Undeterred, they next interviewed the Provost. Shortly after their visit to Keith Kennedy, the Provost called me somewhat upset. He reported that after talking to him Steve turned to Cecelia and said: “I think that he just needs a dumb terminal, don’t you?” They had better luck lower in the administrative hierarchy.

## **Cornell Becomes a National Supercomputer Center**

In 1982 Ken Wilson won the Nobel Prize in Physics. He had long been part of an effort to convince the government to support supercomputing. With the Nobel Prize he became a leader of that effort. In 1985 the NSF announced a competition to create 5 National Supercomputing Centers at Universities. In the middle of the competition period I was asked to talk at an IBM-sponsored conference in New Haven about the AEP-funded “Project Ezra” at Cornell. As the conference ended I was approached by Carl Ledbetter who asked if I had time to discuss something that he had been musing about. My flight back to Ithaca had been delayed by weather and I had plenty of time. I had met Carl earlier when Columbia University acquired a 360/91. Carl had been involved in IBM’s entry into the supercomputer market with the 360/90 series. After selling only 18 machines, IBM had abandoned the supercomputer market.

Carl wondered if Cornell and IBM could partner in creating an application to be one of the National Supercomputer Centers. The notion was to channel-connect a number of FPS (Floating Point Systems) array processors to the largest IBM mainframe to create a computer in the supercomputer range. A chemist at IBM had already shown that this was possible. In the 50s at the Watson Lab, I had written a paper that showed that if you numerically solved partial differential equations using finite difference elements, and if you selected the right coordinate system, almost all of the partial differential equations of classical physics were amenable to massively parallel computation. In fact the only hope for solving problems like understanding weather was effectively using a massively parallel computer. The problems that Ken Wilson was interested in solving were also amenable to massively parallel computation. Wilson was an ardent fan of parallel computing. At the time Cornell had two FPS array processors attached to its mainframe, so we knew how to do that. Carl and I agreed on an equitable distribution of labor. He would try to convince IBM to donate all of the IBM equipment required to create this supercomputer, and I would try to convince Cornell to submit a proposal to the NSF employing this architecture. Parallel computing was the wave of the future and this proposal would be the only parallel computer in the competition. Carl’s parting words were: “When I bring this proposal to IBM management they will probably say: “Carl, how long have you worked for IBM, not counting the rest of today?” The next morning I brought this proposal to Ken Wilson and the Provost, Bob Barker, and they were both enthusiastic. I reported to Carl that half our work was done.

We quickly developed an NSF proposal that included the donation of IBM’s latest and greatest and sent it to IBM. As the deadline for proposal submission approached all we could learn was that IBM had not yet made a decision. Three days from the deadline we informed IBM that we needed to know one way or the other immediately. Shortly thereafter, we received notification that if we were prepared to meet with Ralph Gomory, IBM Director of Research, at their Yorktown Laboratory that afternoon, we would get a decision. Barker chartered a flight to Yorktown and he, Wilson and I set off. On the flight I prepared some flip charts using a magic marker on overhead projector foil. Fortunately the flight was smooth so the result was legible. The foils extolled the virtues of parallel computing as the wave of the future and pointed out that our collaboration would be the only parallel computer in the competition. I finished with a statement that the benefit to IBM was that they would have a seat at the table where supercomputer applications and the technology necessary to solve “big important problems” were discussed.

I made my presentation to a group of about 12 people in a conference room. It quickly became apparent that aside from Gomory, none of the other IBMers present had heard of this proposal. They were all high-level managers at the laboratory. Carl Ledbetter stood in the back of the room not looking too confident. After I finished

my presentation, Ralph asked the people sitting around the table what they thought. One person stated that IBM had gotten out of the supercomputer business and he couldn't understand why IBM was interested. Other people around the table thought it might be an idea worth trying, with a few having no opinion. One person observed that participation could result in serious embarrassment for the company. Perhaps he was referring to an incident a short time earlier. In response to a comment from some IBM executive that IBM was not interested in supercomputers because the market was too thin, Wilson had told a joke that got national coverage. The story in the joke was that serious traffic congestion had developed on the only bridge crossing the Hudson River at Poughkeepsie. This problem precipitated serious discussion on the question of whether or not a bigger bridge was needed and consultants from IBM were hired to produce a recommendation. They reported that a bigger bridge was not needed because after careful watching, they had not detected a single person swimming across the Hudson at the bridge site. At the time of the joke's appearance in the press it was reported to me that some people at IBM were not amused. After everyone sitting around the table had voiced his opinion, Ralph looked up and said: "Alright, we'll do it; you will have a letter of commitment from IBM tomorrow morning". In anticipation of running a supercomputer at Cornell, Ken Wilson had created a Theory Center. The next afternoon a member of the Theory Center staff flew to Washington to hand deliver our proposal to the NSF.

Our next problem was to prepare for the NSF's site visit committee. We had a dry run. It was immediately apparent to me that no one, including Wilson, had prepared. After a presentation by an astronomer on the faculty, I asked: "Would you rather have a Cray?" He immediately replied: "Yes". I thought that it was all up the smokestack. On the day the site visit committee arrived every presenter was eloquent, particularly Wilson. Wilson had invited a parallel-computing expert to testify. He informed the committee that failure to advance parallel computing could set back the evolution of computers and result in a tragically wasted opportunity. The committee asked very few questions. Their smiling faces told me that we had won. The other winners were from Princeton, Illinois, Pittsburgh, and the University of California at San Diego.

The last-minute decision by IBM to support the joint proposal had made it impossible for them to create a contract with Cornell specifying what Cornell would do for them in return for their investment. They now wondered what I meant in Yorktown when I said that they would have a seat at the table. They proposed a contract specifying a number of deliverables. Most of the deliverables involved access to faculty members, but it was not possible for Cornell to impose any requirements on the faculty using the supercomputer. When stripped away there was very little left. On the signing of an agreement by Cornell and IBM's top corporate attorney, he said to Wilson, Barker, and I: "Gentlemen, you have given us the sleeves to your vest".

The multiprocessor system initially provided by IBM to the Cornell Center was quickly upgraded with a "vector" processing capability that significantly enhanced performance. The leader of the group at IBM that developed this capability was Carl Ledbetter. I now began to understand why Carl was musing in New Haven. Over time, IBM contributed equipment with a list value of many millions of dollars to the Cornell Center. Some years later an IBM president told me that the addition of vectors had enabled them to sell a lot of computers that they otherwise would have been unable to sell, and that their investment in supercomputing had been profitable to the company.

## **NSFnet is Created**

The first issue faced by the 5 winners of the National Supercomputer Center competition was how to connect supercomputer users who were distributed at Universities all over the country to the Centers. The NSF hired Dennis Jennings, the Director of Computing at University College Dublin, and the first European College to connect to Bitnet, to manage that effort. Fortunately, Dennis had been born in England and had an English passport, which made him eligible to serve. Dennis was thoroughly Irish with a big smile and had he been born in Ireland he would not have been eligible.

A major issue in creating this network was the communication protocol to be used by the network. There were 3 possibilities. There was an International standard called OSI (Open Systems Interconnection) that was working its way through a committee dominated by hardware vendors at the pace of a pig through a python with, in my view, the same final output. There was DECnet, a proprietary protocol developed by the Digital Equipment Corporation and used by a large number of physicists connected to ESnet (Energy Sciences Network). ESnet was funded by the Department of Energy and connected Universities to National Laboratories, and other research organizations. Physicists were likely to be big early users of the Supercomputer Centers. Finally there was TCP/IP (Transport Control Protocol/Internet Protocol), a protocol developed for ARPANET. TCP/IP broke messages into packets and the messages were reassembled at the destination computer. Packets in the same message could take a different path from origin to destination. This design reflected the fact that computers and routers were unreliable and could become unavailable in the network at random times. In this case each message packet could find an alternative route if one was available. By 1985 ARPANET was no longer operational but the protocol had survived. As a graduate student at the University of California Berkeley, Bill Joy had incorporated TCP/IP into the BSD (Berkeley Standard Distribution) version of Unix. This version of Unix, called Berkeley Unix, was widely used by Computer Science Departments. In 1982 Bill Joy was a co-founder of the Sun Computer Systems Company. TCP/IP was an “open standard” whose future evolution could be controlled by its users. Ever mindful of Lenin’s advice: “influence is good, control is better”, the University community strongly favored TCP/IP.

Ken Wilson’s wife was manager of the UNIX Support Group in Computer Services and an ardent supporter of TCP/IP, so I had no doubt where Wilson stood on this issue. I wondered if Wilson would be willing to go to the wall with the physics community if they organized in support of DECnet. I quickly got my answer. Dennis Jennings reported that at an NSF Science Advisory Committee meeting, Wilson had taken off his shoe and banging it on the table shouted: “TCP/IP, TCP/IP, TCP/IP”! Dennis reported that, aside from him, no one else at the meeting had the foggiest idea of what he was talking about. This story got broad currency and we never heard a peep from the physics community about DECnet. Some years later, at an anniversary of the start of NSFnet, Wilson, Jennings, and I were on a panel of pioneers to talk about the early days. Jennings brought up the shoe-banging episode and Wilson confessed that at the time he didn’t have the foggiest idea either.

In the fall of 1985 Jennings convened a meeting of the Directors of the supercomputer centers at the National Center for Atmospheric research (NCAR) in Boulder, Colorado, to discuss the creation of a network. Sitting around an outdoors table we architected a 3-tier network. At the first level was a multi-connected network connecting the Supercomputer Centers to each other. This was called the “backbone network”. At the next level were regional networks connecting Universities in a region to one of the supercomputer centers on the backbone. At the highest level was a local campus network connecting users at a University to a regional network. When we finished the design of this network, Dennis asked: “and the protocol?” Wilson and I shouted: “TCP/IP”, and with a broad Irish grin Dennis said: “Done!” Dennis then told us that he was allocating \$250K to the funding of the backbone. He stated that he had not gotten permission to allocate that amount but that at the NSF it was often easier to get forgiveness than permission. The next day Cornell ordered 56 kilobit-per-second lines to form the backbone, and the University of Illinois took responsibility for acquiring the routers for the network. Thus NSFnet was born. It began to be referred to as the NRN, the National Research Network.

The next day I invited the Directors of Computing Centers at Research Universities in New York to a meeting at Cornell scheduled for two weeks later to discuss the formation of a New York State Regional Network. At this meeting Richard Mandelbaum from the University of Rochester and I agreed to seek Federal and State funds to support the creation of New York State Education and Research Network (NYSERnet). We quickly secured State and Federal support to launch it. The next problem was to secure continuing funds. At the same time, Regional Networks were being formed across the country with many of the major movers and shakers coming from the Bitnet community. Doug Gale who had been Director of the Decentralized Computer Support Group at Cornell had moved to the University of Nebraska as Director of Computing. He organized the creation of

MIDnet serving Arkansas, Iowa, Kansas, Missouri, Nebraska, Oklahoma, and South Dakota. Building NSFnet was going to cost a lot of money, and we needed the Federal Government to step up to that challenge.

EDUCOM was an academic computing organization that had been involved in pushing networking since its founding in 1964. In 1984, Ira Fuchs got a grant from IBM to create a Bitnet Information Center. Among other activities this Center managed the creation of routing tables, which needed to be continuously updated as the network grew. He decided to locate this center at EDUCOM in Princeton. As a result of its previous history and the association with Bitnet, EDUCOM meetings were the principal forum at which networking issues were discussed.

At an EDUCOM meeting in 1986, Doug Van Houweling approached me with an idea. Doug had been Director of Academic Computing when I came to Cornell in 1980. In 1981 Doug was appointed CIO at Carnegie Mellon. Doug was a unique talent and his departure was a great loss to Cornell. Doug, who had a Ph.D. in Political Science, told me that we needed to organize an effort to convince Congress to fund the expansion of NSFnet to every College and University in the country. To accomplish this we needed to have an active program of educating Congressmen on the importance of this effort. Not-for-Profit organizations like EDUCOM could not “lobby,” but they could “educate”. He suggested that he and I jointly convene a group of CIOs with the hope that we could create a group of Universities that would fund this effort. We were jointly able to put together a meeting of about 12 CIOs. At this meeting we were able to convince 5 Universities to pony up \$25,000 each to hire someone to lead this effort. There were a few more Universities whose CIOs said they would probably join this effort but they needed to check with their University before committing. By 1988 this group had 40 members and the annual dues were set at \$5000. We decided to name the group The Networking and Telecommunications Task Force (NTTF). It was chaired by Douglas Van Houweling. With the money we had raised at that meeting we could start recruiting someone to lead this effort. Ed Shaw from Stanford told us that the perfect person to lead this effort was Mike Roberts, who was Stanford’s telecommunications guru. We authorized Ed to try to convince Mike to take the job. He shortly reported that he had accomplished this mission. We got agreement from EDUCOM to embed Mike in their organization, and Mike reported to EDUCOM’s Princeton office. As he arrived the President of EDUCOM resigned and Mike was appointed Acting President, setting back our effort until EDUCOM could recruit a new President.

In the middle 80s, developments at MIT and Carnegie Mellon had the promise of changing the world. “Project Andrew” at Carnegie Mellon and the “Athena Project” at MIT were developing Distributed Computing Systems that were designed to support an integrated computing environment across the campus. People began talking about 3M workstations on every desk. A “3M workstation” could execute a million instructions per second, had a million-byte memory and a million-pixel display. The expectation was that in the near future most people would personally control all of the computing support they required.

Over the period that I was at Cornell the most important thing that changed was the attitude of people on campus toward computing. After the middle 80s when I had lunch in the faculty club, it was not unusual to hear fragments of conversations at tables around me about computing. Students were buying large numbers of computers and it became common to see computers on faculty members’ desks. The Computer Services staff felt needed and appreciated. This was in stark contrast to the mood on campus when I arrived in 1980.

Bill Arms chaired a search committee to select a new EDUCOM President. In the summer of 1987 he told me that the EDUCOM Board had decided to offer the job to me. I was happy at Cornell, high above Cayuga’s Waters. However, the challenge of advancing networking to connect every scholar in the world to every other scholar was irresistible. A few years earlier I had seen a lady on the beach at Cape Cod. She wore a tee shirt and on the back of the tee shirt it said: “If we can put a man on the moon, why don’t we put all of them there?” Unlike that grand technological challenge, connecting all the scholars in the world might just be attainable. I decided to accept the offer.



## Chapter 5

### EDUCOM Activities from 1987 to 1992

In 1987 EDUCOM was a hand-to-mouth organization. It had been living rent-free in space at the Educational Testing Service (ETS) provided by Henry Chauncy, one of the founders of EDUCOM. Henry, also the founder of ETS, had retired and under new leadership ETS needed the space that EDUCOM occupied. As I arrived Mike Roberts, the Acting President, was in the process of moving EDUCOM to rented space in Princeton. EDUCOM had virtually no surplus and had a small line of credit at a local bank. However, membership was on the rise and it was enjoying some overhead from the IBM grant supporting the Bitnet Information Center. At the next Board meeting it was decided to relocate EDUCOM to Washington, DC, when our one-year lease expired.

We found space in a building at 16th and L in DC and were now positioned to let Mike Roberts and the NTTF educate Congressmen. The NTTF became highly influential as the only organized networking group on the ground in DC representing Higher Education. On L Street we were one block from K Street where all of the lobbyists hung out. Far enough away to avoid the stigma of being confused with people who were lobbyists rather than educators, but perhaps close enough to learn a few tricks. Actually the only thing I learned from them is that the Barbecue Industry was responsible for moving daylight saving time up a month.

EDUCOM membership and EDUCOM conference attendance began to grow rapidly. In 1988 EDUCOM created an annual Net Conference in Washington, DC. If you were interested in networking, or the development of instructional software for microcomputers, the EDUCOM conferences, were where the action was. Higher education had become a big market for microcomputers and software. Vendors jostled each other for prime space in the vendor product display area at the EDUCOM conferences, and IBM and Apple tried to outdo each other in hosting EDUCOM attendees. For a period it became possible to attend an EDUCOM meeting without paying for a meal if you were adept at getting invited to vendor presentations. By the end of 1988 EDUCOM was no longer a hand-to-mouth organization.

### The EDUCOM/NCRIPTAL Outstanding Software Awards

EDUCOM began pursuing two major thrusts in support of academic computing. One thrust, led by Steve Gilbert, mobilized the talents of Directors of Academic Computing at Universities in order to identify and promote quality software developed at their institution that improved education. In 1987 EDUCOM collaborated with the National Center for Research to Improve Post Secondary Education (NCRIPTAL) at the University of Michigan to create the EDUCOM/NCRIPTAL outstanding software awards program. A national review panel judged software submissions on their importance to the undergraduate curriculum, factual accuracy, and comprehensiveness. This effort was supported by the Annenberg CPB project, NeXT, Apple, and IBM. In 1987 there were 139 submissions and 27 awards for outstanding software. The number of submissions grew as the annual

EDUCOM/NCRIPTAL 1989 Trophy for Best Engineering Software



awards program continued. One winner from Cornell was Bob Cooke and his team for a finite elements program. This program allowed faculty to assign students a homework problem that would have generated a PhD a few years earlier. In 1991 Joe Wyatt, the Chancellor of Vanderbilt University, challenged EDUCOM members to produce and identify 100 outstanding software success stories. Judith Boettcher from Penn State chaired this national effort. In 1993 she edited a book titled: *101 Success Stories*.

In 1991 IBM started an annual awards program recognizing an individual for outstanding contributions to advancing computer-related education. This program was managed by EDUCOM and called the “Lou Robinson Award”, named to honor a long-time IBM Director of Education who had been a tireless and inspirational speaker at Colleges and Universities on the subject of computer education. Along with a cash award came responsibility to speak at the EDUCOM conference in the year of the award. The first award winner was John Kemeny, the President at Dartmouth, who with Tom Kurtz developed the Basic computer language and started one of the first time-sharing efforts at a College. Kemeny was determined to see that wherever computing education was going, Dartmouth would get there first.

Kemeny was a chain smoker and airline regulations that prohibited smoking made it impossible for him to take a long flight to the next EDUCOM Conference to speak, so I went to Dartmouth and we taped a video interview. An edited version of several hours of interview was shown at the EDUCOM conference and was a real winner. Kemeny explained to me that cigarette companies had provided free cigarettes to soldiers during World War II, as their contribution to the war effort, and he returned from the war with a severe addiction problem. We talked about a lot of things unrelated to computing in education. He had been chairman of a commission investigating the Three Mile Island nuclear power plant accident in 1979 and we talked about that. You can watch this video at: <http://www.youtube.com/watch?v=HHi3VFOL-AI>

The next year’s winner was Seymour Papert. Seymour, from the MIT Media Laboratory, had developed the programming language Logo that enabled very young children to start programming by controlling a robot called a turtle.

In 1990, the Association of Research Libraries (ARL), EDUCOM, and CAUSE joined together to form The Coalition for Networked Information (CNI) in order to create a collaborative project focused on networking that would integrate the interests of academic and research libraries (ARL) and computing in higher education. The first Executive Director was Paul Peters. The three sponsoring organizations sought to broaden their respective community’s thinking to encompass digital content and advanced applications to create, share, disseminate, and analyze such content in the service of research and education. CNI holds an annual conference in Washington and is still going strong under the leadership of Clifford Lynch who succeeded Paul Evans as Executive Director.

EDUCOM had a consulting service that provided consulting on computing issues to Colleges and Universities by members from similar institutions that enjoyed high-quality computing. I joined a number of these groups. I learned that leadership at the top level of the University was a key element in creating a quality computing environment. At one College we ended our visit by interviewing the President. He said that he thought computing was a distraction to learning and was planning to create computer free zones on campus.

## **The NREN Becomes the Internet**

The second major effort at EDUCOM was building a worldwide scholarly network. Mike Roberts and the NTTF led this effort. Over a period of time the NTTF developed 4 goals:

1. Connect every scholar in the world to every other scholar, surmounting the barriers of space and time to scholarly collaboration. (This goal came out of Bitnet.)
2. Enable access from the network to all scholarly information worth sharing.
3. Build a knowledge-management system on the network that enables scholars to surf through this sea of knowledge in a simple and intuitive way.
4. Enable building knowledge on a new, dynamic, multimedia foundation in addition to building it on static information in print format.

There was a fifth goal that never got any traction: To enable enable students to find instruction any time, any place, on any subject. Some people found this goal threatening to the future of Universities that still employ batch-processing systems to educate students.

When EDUCOM moved to Washington we discovered two staunch allies in Congress, Al Gore in the Senate and George Brown Jr. in the house. While Al Gore's father was in the Senate, he had sponsored the bill creating the National Highway System, and Al was eager to create a National Information Superhighway based on the NREN. As early as 1986 Al Gore began to extol the importance of high-speed networking to the future of the country. I believe that he has gotten nowhere near the credit he deserves for creating the Internet. NSFnet was now referred to as "The National Research and Education Network" (NREN). The NTTF had easily persuaded Stephen Wolff, who succeeded Dennis Jennings as the NSF person responsible for networking, to allow education as an acceptable use on NSFnet and the NRN became the NREN.

The goal of the NTTF was to create a network connecting scholars. Al Gore's goal was to create a network connecting everyone. That worried me until I remembered what Mae West had once said: "too much of a good thing can be simply wonderful!" and suddenly connecting everyone seemed like a good thing. But the NREN was our sandbox and letting everyone play in that box was frightening. The NTTF threw in with Al, and Al created a bill called "The High Performance Computing Act of 1991". There were three things that we could do to help Al: we could provide technical help with the bill's contents; we could provide experts to testify before congressional committees; and we could mobilize University Presidents to call their Senators and Congressmen to tell them how important the NREN was to their University. Our primary interface in Gore's office was Mike Nelson. Mike was a geologist by background, and we were fond of saying that Mike had come to Washington to learn the difference between a rock and a hard place.

The bill was enacted on December 9, 1991. In signing the bill, President George H. W. Bush predicted that the Act would help "...unlock the secrets of DNA, open up foreign markets to free trade, and [establish] a promise



**The Gore bill passes: 1991**

of cooperation between government, academia, and industry”. All those things certainly happened, probably on a scale far greater than he could have imagined. On the afternoon of the passage of the bill, I was invited to a small celebration in Gore’s office and I have a picture of Al at that party with an unindicted co-conspirator.

In 1993 Al Gore became the Vice President, and in April the NSF announced that the NREN would become privatized in April 1995 to become the “Internet”. On hearing that news, it struck me like a bolt out of the blue: This could lead to the end of “civilization” as we knew it!

I talked to Glenn Ricart about creating a small group that would contain some CIOs and networking superstars to try to create a virtual academic network on top of the Internet. We created a group of 25 people plus a representative each from EDUCOM, Internet II, CNI, and Cause. Glenn suggested calling this group the “Stone Soup Group” after an old folk story. It subsequently evolved into the Common Solutions Group (CSG) and is still going strong. Its email list provides me with my current window into networking and academic computing issues at Universities.

In the late 80s the international standard OSI protocol began to emerge and some European institutions, particularly in Germany, began to employ it rather than TCP/IP. This required a protocol conversion gateway between TCP/IP and OSI networks. TCP/IP was a lighter, more nimble protocol and had by this time developed huge momentum. In the United States there was only one TCP/IP holdout and that was the Library of Congress. The person running computing at the Library was Henriette Avram. Henriette had developed the MARC card catalog format, the international standard for bibliographic and holdings information that was the format used at most libraries. She was a legend in the library community and a staunch believer in International Standards since she had developed one. In order to put the Library of Congress catalog on the Internet, the University of Maryland created a PC-based gateway between the NREN and the library’s internal network.

I talked to Henriette a number of times, trying to persuade her to convert to TCP/IP but she seemed unmovable. It was always fun to talk to Henriette, particularly about the halcyon days when she was a 701 programmer. One day someone from Berkeley told me that students there had developed a highly successful persuasion technique. It was called: “talking a person to death”. You would mobilize students to massively attempt to persuade the person by any means at hand. I started sending people to talk to Henriette but not many could spare the time. I informed Henriette of my strategy and she was amused. She said that she always enjoyed talking to computer people. In a conversation one day with the NSF Director of CISE, the Directorate for Computing, Information Science and Engineering, he mentioned that he was visiting the Library of Congress the following day. I suggested that he drop in for a conversation with Henriette if he had time. The next day, Henriette called me in a highly agitated state. She asked: “Who was that man?” I asked: “What man?” She said that a man had barged into her office and shouted: “Who the hell do you think you are?” He then ordered her to convert to TCP/IP and stalked out. I told Henriette that I would “call off the dogs”. In 1992 an IBM Vice President was the keynote speaker at the Net 92 conference in Washington. He announced that IBM was abandoning support of the OSI protocol in favor of TCP/IP. At the end of his talk Henriette walked up to me and said: “I give up. He’s bigger than me.” I replied: “Henriette, can I buy you a drink?”

In 1992 Bob Kahn, Juergen Harms, and I signed papers of incorporation creating the Internet Society as a non-profit organization, and we appointed a founding Board of Trustees with international representation. The Internet Society was created to support the growth and technical evolution of the Internet as a worldwide educational and research infrastructure. The Society internationalized the Internet evolution effort. On the following page is a picture of the founding board holding a copy of the Society’s charter.

You can learn about the contributions of these Internet pioneers by going to Google. The Society is still going strong.



### **The Internet Society was started in 1992**

The Founding Board: In the front row from left to right are Ira Fuchs, Ken King, Bob Kahn, Juergen Harms, and Anthony Rutkowski. In the second row are Lawrence Landweber, Hideo Aiso, Lyman Chapin, Kees Neggers, Tomaz Kalin, Vinton Cerf, Froda Greisen, Geoff Huston, and Michael Roberts

The University and Research communities created new Internet applications in the early 90s. Among the most significant were web browsers at CERN and the University of Illinois, arXiv at Los Alamos, the Internet Gopher at the University of Minnesota, CU-SeeMe at Cornell, and the handle system for digital library objects at CNRI. In particular, web browsers caused an explosive growth in Internet users.

I believe that Universities have never been given proper credit for their role in creating the Internet. By the time NSFnet was started in 1985, networking was already a global University phenomenon. It was the zeal of the University computing community fired by academic goals that fueled the explosive growth of networking. Universities were the laboratory in which Internet technology and applications were built and scaled up to work with millions of interconnected people. University people provided most of the creative energy and did almost all of the work in creating the NREN that evolved into the Internet. Universities trained the people that brought the Internet into the commercial world. The role of ARPANET has been well documented but the role of Universities has not. It is my hope that the Memoirs in this incremental book will redress that problem<sup>1</sup>.

---

<sup>1</sup> An oral presentation of portions of this Memoir were presented in a lecture on February 17, 2011, for the Cornell Association of Professors Emeriti: “The Origin and History of the Internet, a lecture by Ken King” at <http://ecommons.library.cornell.edu/handle/1813/22368>



# Chapter 6

## Looking back

A lot of progress has been made on The NTTF goals:

1. Connect every scholar in the world to every other scholar, surmounting the barriers of space and time to scholarly collaboration. (This goal came out of Bitnet.)
2. Enable access from the network to all scholarly information worth sharing.
3. Build a knowledge-management system on the network that enables scholars to surf through this sea of knowledge in a simple and intuitive way.
4. Enable building knowledge on a new, dynamic, multimedia foundation in addition to building it on static information in print format.

I think that we can check goal 1 off as having been accomplished. In the 80s the challenge was to connect every scholar in the world to every other scholar. Today the challenge is to connect every person in the world to every other person.

Progress has been made on goal 2, but a lot of work remains. Some breakthroughs have been achieved toward open publication: major milestones include the preprint archive ArXiv maintained by the Cornell Library and PLOS (The Public Library of Science) that distributes open-access peer-reviewed articles on the Internet. Clearly information “wants to be free”. It is the only non-depletable commodity. The more it is used, the more valuable it becomes.

In the early 2000s Bob Cooke and I began talking about problems associated with expanding the amount of information available on the Internet. Bob was concerned because libraries were beginning to drop journal subscriptions because of their rapidly increasing cost. The journal publication system involved the Government and Universities paying for research, but the scholars were publishing in journals that required the scholar to assign the copyright to the journal. The journals then sold their content back to libraries at ever-rising (greater than CPI) cost. University Presses were unable to publish scholarly work in some disciplines because the market for books in the discipline was too thin. Electronic publication of books was now feasible, but Universities were continuing to buy miles of books every year. Finding a mechanism to distribute and pay for electronic versions of books consistent with copyright laws was and still is a major hurdle to their replacing physical books. Electronic books are content-separate from a physical object that restricts usage to one person at a time. One copy of an eBook on the Internet or in the cloud could serve every University in the world. An eBook can contain multimedia material and is closer to software, videos, and music than a physical book. A site licensing mechanism could solve this problem. Just keeping physical books on a shelf in a library costs something like \$6 per year. Something like 90% of the books acquired by a research library are never circulated. The paradigm for book acquisition is “just in case”, rather than “just in time”. Just freeing the money spent on shelving and processing miles of physical books every year could pay for a lot of pay-per-view access to electronic content. There seemed to us to be something very wrong with the current system. I was additionally concerned because I hoped to check off NTTF Goal 2 in my lifetime.

Bob was Dean of the University Faculty at Cornell at the time. We concluded that the solution to a lot of these problems was promoting Internet publishing of all scholarly work as a first choice if it were economically feasible. Bob believed that if every University assumed responsibility for the publication of the scholarly work of its own faculty, and published it in open-access mode on the Internet, they could save money.

In addition, Internet publication would vastly increase exposure to scholarly ideas produced at their institution.

We succeeded in getting a 3-year grant from the Atlantic Philanthropies Foundation to pursue the economics of Internet publication in the hope of finding a solution that would result in the Internet providing access to all scholarly information worth sharing. With funds from the grant we joined the DSpace Consortium that was supporting the development of software at MIT enabling Universities to archive information on a server whose content could be distributed on the Internet. We also acquired a server managed by the Cornell Library that DSpace required to hold its archive. At the same time we created the Internet-First University Press (IFUP) as the vehicle enabling us to experiment with Internet publication. With excellent support from the Cornell Library we discovered that lots of valuable scholarly information could be published at very low cost. We published a number of out-of-print books. One by Jack Oliver, entitled *The Incomplete Guide to the Art of Discovery* has had more than 100,000 downloads. We've published books that couldn't be published by a commercial press because the market was too thin. We've video taped Symposia and lectures by distinguished faculty members and published them on the Internet. We've put online all issues of the Cornell Alumni magazine, faculty memorial statements, and the minutes of the Faculty Senate. We have put Departmental histories online and are in the process of trying to videotape an interview with every emeritus faculty member covering their career. We have succeeded in getting most of Cornell theses and dissertations online. The DSpace archive has become the Cornell eCommons, and DSpace has merged with a Cornell-produced archive system called Fedora to create DuraSpace. You can find material published by the IFUP<sup>2</sup> at: <http://ecommons.library.cornell.edu/handle/1813/62>

From this project we have learned that it is very inexpensive to scan and OCR (optical character recognition) printed documents and to videotape lectures, interviews, and other events. The University and faculty own the copyright to lots of material of scholarly value and this material can be put on the Internet with a modest effort. Publishers are often willing to return the copyright of an out-of-print book to a faculty member. This was the case with Jack Oliver's books. If every University undertook responsibility for publishing the scholarly material in its archives that it or its faculty owns, the result would be of enormous value to the world.

Search engines like Google have enabled progress on goal 3. Artificial intelligence based systems like the IBM Watson program promise significant progress in the future. A lot of progress has been made on goal 4, enabled by inexpensive video-cams and editing and other image and sound processing software. Words like "visualization", animation, and "virtual reality" are used to describe progress. When I taught an introductory Computer Science course, I would tell my students: if you can see it and it's there, it's real; if you can see it and it's not there, it's virtual; if you can't see it and it's there, it's transparent; and if you can't see it and it's not there, it's gone. So much progress on electronic display of information has been made that the future of hard cover books and print documents like newspapers are now threatened.

Attaining "computer literacy" and infusing computing into the curriculum were huge challenges from the 50s to the 90s. Today many students graduate from high school with the skills that we struggled to impart and support during that period. Enrollments in introductory computer science courses have dropped as a conse-

---

2 See also: "Compressed version of the Final Project Report to Atlantic Philanthropies" at <http://ecommons.library.cornell.edu/handle/1813/3459> and "The First Ten Years of the Internet-First University Press and CAPE's Histories and Biographies Project" at <http://ecommons.library.cornell.edu/handle/1813/36253>

quence. Nearly every student shows up at college with a computer, frequently more than one, and they carry it around with them. The computer in their pocket is as fast as the fastest supercomputer in the mid-80s and it has a larger memory and more random access storage. It connects them to friends around the world via instant messaging and email. With Skype they can even see and talk to people around the world. It connects them to vastly more information on the Internet than they could have found in all the libraries in the world in the 90s, and almost every document is accessible with a full text index.

Research in almost all disciplines has become computer-dependent even in the Humanities. Faculty members in most disciplines control all the computing power they require. In the sciences, social sciences, and engineering the curriculum has been transformed by access to “application packages”. Progress in the structure and methodology of instruction has been slower to change. In the 90s people were fond of pointing out that if a physician slept for a hundred years, and if when he woke up he visited a hospital, he would feel like he had been dropped into a planet in some other galaxy. If a professor slept for 100 years and walked into a classroom he would still find a blackboard and chalk and feel right at home. Today a screen often replaces the chalkboard and a PowerPoint presentation replaces chalk on blackboard but not a lot more has changed in many disciplines. Video-taping a lecture to a class with a large number of students and putting it online probably has more value to students than requiring them to sit in a classroom with little or no opportunity to ask questions. Long ago, educators discovered that interactive education is more effective than passive education, but most of education is still based on a sage on a stage.

Today’s supercomputers are massively parallel with more than 3 million computing cores. They are more than 10-million-times faster than a supercomputer in the 80s. In signing the Gore bill in 1991 President George H. W. Bush opined that computers would one day be able to decode DNA, which was at the time one of the grand challenges of science. Today DNA decoding has opened up many new fields of research. Recently I read that falcons were genetically similar to parrots and woodpeckers, and the apple is a member of the rose family. Development of software has become a cottage industry with new apps proliferating at a rapid rate. Universities are no longer the source of most educational software.

Computer service organizations have gotten smaller and are again largely enterprise units. Their heads are now in the “Cloud”. If a University can find a computer service like email in the Cloud, it is cheaper to contract for it than to support a locally developed system. CIOs at Universities no longer have the financial wiggle room to invest in new ideas. The only large-scale computing project designed to change the world that Universities are currently involved in is Internet II. This attempt to create the next generation Internet is a collaborative effort involving industry and many Universities.

In the 90s Universities were measured by the number of books in their library and big Universities added miles of books every year. Today JStor provides network access to back issues of more than 2000 journals with full text search. Miles of library bookshelves are no longer needed. Electronic books are cheaper than hard cover books and come with full-text search capability, and sometimes with multimedia elements, but using them presents copyright problems that have not yet been solved. Since Gutenberg invented the movable type printing press in 1450, great Universities were built around great libraries. Today the Internet provides access to information found via full-text searches that vastly exceeds the locally stored information in any University library; but University libraries still largely measure themselves by the number of books on their shelves.



# Chapter 7

## The Future Lies Ahead

Super computers 1000 times faster than today's fastest computer are on the drawing board using current chip technology. Far faster quantum computers are a gleam in computer designers' eyes. The consequences of this increase in capability on everything boggles the imagination. Technology allowing people to talk to their computer with artificial intelligence assistance in locating information is just around the corner. Artificial intelligence assistance in many areas, even in grading and editing papers, is being developed. Networks connecting homes will be supported by fiber optics cables capable of delivering data at rates more than 1000 times faster than current home data rates.

The impact of emerging technology on Universities is likely to be profound. Current education delivery systems are labor intensive and consequently increase in cost faster than the inflation rate. Tuition increases are often at twice the rate of inflation. Nothing can continue to grow exponentially without a day of reckoning. At many Colleges, adjunct faculty members that earn small salaries without a benefits package are now teaching courses. A recent development is the emergence of Massive Open Online Courses (MOOCs). MOOCs frequently enjoy enrollments in the thousands. A Faculty member teaching one of these courses gets more feedback from students in one semester than he or she would get in a lifetime of teaching in a classroom. This feedback enables the course and its delivery system to improve over time. The MOOC courses are available anytime, any place, to any student. This is the goal that I failed to get included in the NTTF list in the 80s. If great Universities can cooperate to create great courses, why shouldn't they all use those courses? Colleges are beginning to give credit to students taking these courses. I expect this trend to grow rapidly.

Current educational systems are largely batch-processing systems. When they become distributed systems, students will have courses available to them wherever they are that far outnumber the courses available in a College today, as well as access to the Internet library. Virtual Laboratories will replace real ones. If you have a question in the middle of a lecture, you will pause the lecture and ask Professor Siri for an answer. You won't even have to raise your hand. It doesn't get any better than that. However, this can lead to reducing the social interactions between students and the mentoring relationship of students with faculty members that is important in education. Someone has pointed out that the invention of the printing press freed the monks from their transcriptional responsibilities and they went off and created the wine industry. Freeing many faculty members from their teaching responsibilities will have unpredictable consequences for Universities and the world. When the NREN was privatized to form the Internet, I worried that this could lead to the end of "civilization" as we knew it. It has certainly changed almost everything.



