



## Oral and Personal Histories of Computing at Cornell

[Introduction](#)[Timelines](#)[Stories](#)[Pictures & Poems](#)

### Mike Newman Recollections, 1961 - 1999

I came to Cornell in 1960 and was in the Class of 1964, Engineering Physics. I was in Grad School for a while, working in the Cornell Center for Radiophysics and Space Research with Tom Gold among others. I was working on pulsar theory and then I got distracted flying airplanes until 1977 when I came back to Cornell as an employee.

The first computer dedicated to monitoring building systems was installed in 1974 or 1975, somewhere in there, and so this was a couple of years after that. The original machine was an IBM System/7, which was diskless. The initial program loading function was done by paper tape. Whenever anybody had a change they wanted to make to the database the IBM system engineer had to go up to Syracuse and punch a new paper tape and bring it down. Somewhere about 1977, I had just started, IBM came around and said they had an upgraded System/7 that featured not only disk, but had fixed and removable platters which each held 10 Megabytes, and which had an interface which allowed you to connect serial devices like modems, for the first time. There was a company in Texas called Fischbach and Moore, F & M Systems, which had developed a multiplexer interface, a device that could communicate over a serial link at 1200 baud. This device allowed us to talk to remote multiplexers which could connect to sensors and actuators and read digitized temperatures and pressures or flows depending on the sensors. So, for the first time we really had a way of getting access to fairly reliable and accurate measurements of process variables for building automation and control functions.

The first System/7 was sold to the Utilities Department. Ed Hartz, now retired, was in charge then. It was a turnkey deal. IBM came along and said they had this power management program. I guess it followed from the Arab oil embargo in 1973 or so. Cornell said OK; this is something we can do to reduce our energy consumption. The function of the system was to look at pulses from the Kite Hill substation and to shed load. Plus it could be set up for scheduled stopping and starting of devices. Statler Hall was one of the early installations.

However, the story really starts in 1964 when Cornell bought through Honeywell a manual control panel called a "Selectrographic 6" which used relays and DC voltages sent out on a 110 conductor wire. This allowed specific pieces of equipment in the newly built Clark Hall, which was the point of it, to be controlled from a central location. The location selected was Chilled Water Plant 1 because this plant was built to provide chilled water to Clark Hall. The projects went together. It was decided to have someone on site to monitor the chillers, 24 hours a day, 365 days a year, so they put this electromechanical switching and monitoring system, which had no computers, in Chilled Water Plant 1 and connected it up initially to Clark Hall. As time went on, it was connected to 15 to 20 buildings at the high water mark. It basically let you see the on-off status of fans and pumps. There were 4 sets of 10 push button switches, numbered 0 to 9, and so you could pick an address like 1274 from the 4 columns of switches which closed a bunch of relays and you would get a path to a relay which would indicate whether a particular pump or fan was running. You could hit a stop or start button to send a 48 volt signal would be sent to turn the unit off or on. It had a neat feature, which we later lost, an intercom system where you could hear what was going on in the mechanical room where the selected unit was installed. You could hear the fan starting or stopping when you hit the button and you could find out what was happening. You could even talk to any mechanics who might be working in the mechanical room. All of that stopping and starting was done manually based on a notebook the operators had up until the mid 1970s when the first IBM System/7 was installed.

The way the IBM and Honeywell systems connected was through a huge multi-pole relay, called a T-bar relay, which physically switched all of the 40 switch buttons and the other signal and control lines over to the computer. The System/7 had what we now call binary inputs and outputs which could sense voltages on a particular line and activate voltages to do the actual stopping and starting. The idea was that this file from paper tape had the addresses of all these different fans and pumps and the thing sat there and looked at the pulses from Kite Hill and decided if a certain threshold had been breached and then would go through a list of selective loads to cycle on and off, trying to keep the total demand down. Of course, we only had access to several hundred kilowatts out of a total load of about 25 megawatts, so load shedding never really amounted to much.

Another problem with the system was that you could inadvertently access a device without knowing it if a short circuit developed between any of the address lines. This occurred one day and caused the ventilation in a chicken barn to get shut down - with fatal results to some of the chickens. Progress is not without occasional victims...

The file from paper tape was loaded into the system. A new paper tape from Syracuse was needed when you wanted to add a point or change the length of time that a device would be cycled off or things like that. The next generation System/7, in 1977 or so had a card reader. This new system was sold to Cornell because it had a disk and a new upgraded offering from IBM called Power Management II and Basic Monitoring II, both of which had been developed at the Palo Alto Research Center by IBM staff. It had the ability to access what are now fairly conventional serial asynchronous modems, of the sort we use today except they ran at 1200 baud instead of the 56K we run today. Of course those 1200 baud modems were a great improvement over their 110 baud and 300 baud predecessors. We acquired this System/7, spec'd by a guy in Utilities named Eric Dickman. Eric had worked with IBM on the acquisition of this upgrade before he left to go to work for Digital Equipment Corporation in Maynard, MA, just as I accepted the Cornell job.

When I came along the people running Design & Project Management, the old name for Facilities Engineering, asked me if I knew anything about computers. I said I had taken a course and so I was assigned the duty of getting the new system up and running. My recollection is that I took a course in IBM 360 Assembly Language and CUPL when I was a student. The IBM System/7 was, however, programmed in EDX, Event Driven Executive, a macro assembler language. I went to a two day IBM course in Arlington, VA, to learn EDX. IBM had already developed a new machine called the Series 1 and Series 1 EDX and System/7 EDX were comparable so I was able to learn enough that when I came back I could make modifications to the source code.

The first thing I did was to change some of the text strings. Now, you have to remember that the printer was a 5028 Teletype which "clunked out text" and which was also the input device. When we went to initially load our database with the help of Fred Tingley, an IBM systems engineer out of Syracuse, we had to first compile the statements into executable code. This required us to switch from EDX to DSS/7, Disk Support System/7. We set about compiling the database, the very first time. The System/7 had just been turned on and I knew very little about the system. Fred and I sat there and the lights flashed but nothing happened for 1 hour, 2 hours, 3 hours, 4 hours, 5 hours! Finally we went out for dinner at the State Street Diner. By the time we got back it was about 10 o'clock at night. Fred decided to go home and I stuck around for a while longer to see what was going to happen. About midnight the Teletype spun to life and started spewing out paper. It was that yellowish paper on a roll which we called the "dead sea scrolls" since you always ended up with this huge roll of paper. After printing out about 10 feet of paper down at the bottom we got a message that a period had been used somewhere instead of a comma and the compilation had failed! We figured it took about 8 hours to do the compilation. After changing the period to a comma, or the other way around, whatever it was, we compiled the program again. In the morning when I came in sure enough the compilation had finished and we were ready to move ahead. It had only taken a day to do it.

It soon became obvious (duh!) that this was not a viable process. We found out that IBM supported a DSS compiler which ran on the 360 mainframe. That was the beginning of a massive change in everything as we became much more dependent on the mainframe environment. We could compile the program in 2 to 3 seconds on the mainframe instead of the 8 hours it took us on the System/7. However there was still a catch since the output, the executables, could only be conveyed through the medium of punched cards. So I used to spend a lot of time in the basement of Day Hall, getting to know the terminal operators over there, some of them who are still at Cornell. Card decks a foot and a half to two feet long would get punched out and then you would go to the basement of the Chilled Water Plant and load cards into the 129 card reader. You would hope to hell that the cards didn't get chewed up. Every so often you would read 400 or 500 cards and the 497th card would jam in the machine, to be extracted only through the use of a device called a "card

saw." (Now that is a true relic of the old days!) Sometimes the only thing to do was go back and repunch the whole deck since there was no way to punch that one particular card. Of course, if you dropped the deck you were done. This was a strong motivation to figure out a better way to communicate between the System/7 and the mainframe.

At that time, I had the only terminal, the only computer access device in the Humphreys Service building. This was about 1977-78, It was a Datamedia 1521 on a Sytek connection. We figured out that if a terminal could talk to the mainframe, why couldn't we set up one of the lines coming off the System/7 to somehow connect to Sytek, and somehow login and execute a file transfer. This was long before FTP and the other internet protocols. Somebody had written something called "SFIBRE". This was even before Kermit and it was a very rudimentary file transfer mechanism indeed. It was designed to run on some kind of primitive PC, before the IBM PC, I think a Terak. Greg Busby, Class of '82, and I got the source code and figured out how it worked and we wrote something in EDX which emulated the functionality of this program. We hooked it up and started watching stuff going back and forth and what happened was that it would get various distances into the file transfer and just hang up and die. We didn't have protocol analyzers at first and we were pretty much flying blind. What we found out was that when the mainframe was talking to a terminal it would send a prompt character ">" followed by a flow control XON character to the terminal at which point you could start typing your input. In our file transfer program when we saw the XON (X'13') come down from the mainframe we thought we could start sending our response, request, or our acknowledgment of the previous data. What we didn't know was that the mainframe wasn't actually ready to receive data. It was timed to wait for a human to start typing, for the better part of a second, whereas the System/7 would start sending data the instant it saw the character. Basically the mainframe was not ready to receive data so we built in a timer and diddled with the delay until it was right, about 250 milliseconds, an insight we obtained while having lunch at Manos Diner one day! Once we did that, we had file transfer. The day of punched cards was over. This was about 1982. At that point we were in pretty good shape. We could do compilations on the mainframe and keep everything there and then download stuff over the Sytek link. Getting electronic file transfer going was a major breakthrough!

As time went on the System/7 really ran out of gas. It really wasn't designed for multiple serial ports and increasingly we were adding new kinds of equipment to the field side of things, to the automation control side, and each of them had its own peculiarities. We confronted the choice of what to do about this situation. One possibility was the Series 1, which IBM had told us about within 2 or 3 months of the System/7's arrival. The System/7 was a sensor-based machine used by General Motors, among others, in their manufacturing processes. Its claim to fame was that you could connect sensors and actuators which had "DI-DO", digital input-digital output, modules that could be used to turn things off or on. The Series 1 was the successor to the System/7 and IBM announced that product 2 to 3 months after they had delivered the System/7 and this did not endear them to us. When we complained that we had some problems with the System/7 their response was to ask us why we hadn't bought a Series 1. Our reply was that you didn't tell us about it.

We were looking at the possibility of moving to the Series 1 because the Event Driven Executive language could be converted to the Series 1 version in a pretty straightforward way. At the same time the MicroVAX had just been announced by DEC and although it was going to mean a complete rewrite of our code we decided to go with a mainstream system from DEC where we could choose our language environment. We could write Fortran or Pascal or C which had just come along in the late 1980s, 1988. We could also get more robust peripherals. That was one of the main things the new world brought us. The System/7 was not designed as multi-terminal machine and IBM's connections to terminals didn't make use of standard EIA-232 technology. They had special adapters, special IBM terminals. In fact, our terminal multiplexer was a third party box literally built in a garage in Oregon by a guy named Bill Jackson who was, during the day, an engineer with Tektronix. It was called the "Jackson Avionics" box because Bill also built multiplexers for airplanes. With the VAX, on the other hand, you could just plug in your garden-variety serial asynchronous terminals. So, we ended up going with the VAXes. We've upgraded the machines a couple of times since then.

Increasingly now, things are going to distributed intelligence. In the early days the only way to turn things on and off was to send a signal from the computer to the multiplexer which was connected to the contactor which turned the thing on or off. About 1982, a thing called "direct digital control" became practical in the commercial building environment. Every one of the major manufacturers - Honeywell, Johnson Controls, and MCC Powers (which is now part of Siemens) - decided to use microprocessors to do control functions directly, without the need for a central computer. You would now have microprocessor-based controllers which could run autonomously but also have communication capability. Today, most such systems are even compatible with the TCP/IP infrastructure. As a result we're no longer doing much centralized control but more

monitoring and data analysis and, increasingly, everything is on the Internet using Web technology so you can check things out from any terminal using a browser. Our current system is still built on VMS and was written by Joel Bender of the Utilities Department. The data for a selected unit is gathered from the remote digital controller and displayed on the schematic for that particular building system. One sees the "H" diagram for the building air handling systems or other schematics using standard graphic depictions that show all the measured values and even setpoints.

The "final frontier" in recent years, since 1987, has been the effort to develop a standard communication protocol to replace all of the proprietary ones. (We have had 8 proprietary protocols communicating from the VAX concurrently!) This work has led to the adoption by the building control industry of "BACnet," the "building automation and control networking protocol." The protocol was developed within ASHRAE (the American Society of Heating, Refrigerating and Air-Conditioning Engineers) by a committee that I have chaired since its inception. There are now "BACnet Interest Groups" in the U.S., Europe, and Australia - all as a result of the Cornell-sponsored initiative!

---

Prepared on September 17, 1999 by John W. Rudan

Return to [History of Computing at Cornell](#) main page

Comments about this site: [jwr1@cornell.edu](mailto:jwr1@cornell.edu)

©2005 by John Rudan and [Cornell University](#)