5521 Research Park Drive, Suite 200
Catonsville, MD 21228 USA
The Institute for Operations Research and the Management Sciences

Dick Conway Recollections, 1951 - 2000

Prepared by John Rudan, November 20, 2002

Originally published on *Oral and Personal Histories of Computing at Cornell* as a part of the repository of interviews conducted for Rudan's book *The History of Computing at Cornell*

My involvement with computing started when I was a graduate student. Andy Schultz sent me over to take a course in numerical analysis from Barkley Rosser in the Math Department. I was taking a full load of advanced math courses, for which I wasn't very well prepared, and I was working very hard. I went back to Andy in the middle of the fall semester and said that Rosser and numerical analysis were killing me. I was spending more time on that than all the courses put together and I wasn't sure that inverting matrices was going to be my goal in life. Andy was totally unsympathetic. He said, "Drop whatever else you need but you're going to stick with that one!" In retrospect he was right. Whether he foresaw the speed and intensity at which that would overtake things I can only guess, but I continued to suffer through numerical analysis. Later I wound up using many of the problems that Barkley assigned when I was teaching my own course in computing. The difference was that Barkley would assign it as an overnight homework for the next day's class and I found that this was practically a term project that I would have a couple of people work on for a couple of weeks at a time. So I attribute the whole direction my career has taken, to some extent the influence I had at Cornell, to the role of Rosser and Walker, who had just come back from a year out at the Institute of Numerical Analysis at I think UCLA, and Schultz, who had this uncanny ability to predict the way the world was going to go.

Those three gentlemen were instrumental in getting Cornell to acquire its first machine and so I was involved with these math courses using that machine. So when they decided to escalate to an IBM 650 and IBM had this requirement of teaching a course, these folks had maneuvered me into a position where I was probably the only plausible candidate to take this on. Andy simply said this was what I was going to do and I don't remember having a whole lot of choice in the matter. IBM provided materials because obviously they saw people all over the country being in the same position that Cornell was, and I was, suddenly having to teach a class where their background was modest and recent in a field that was moving very quickly. So what was then IE 3281 which became IE 9381 was just a very exciting proposition.

When the 650 arrived in 1956, I was in my second year of graduate study at the time, and the course was certainly the most interesting thing I had ever done. It was made particularly interesting by the fact that it attracted a very special group of student pioneers, brave people who were willing to try a new course with a new graduate student in a subject that wasn't very well defined. Some of the people in that course, three of them that I still have close contact with,

turned out later to be the three top executives in Emerson Electric - Chuck Knight and Bob Staley and Al Suter. We still get together and trade reminiscences of that experience. They accuse me of trying to ruin their life as badly as Andy had tried to ruin mine. Two years ago I went to St. Louis to a retirement dinner for these three guys and they were telling 650 stories and Conway stories before this whole room of people. No one else appreciated it but we did very much.

Let me tell you another 650 story. Personally I was involved with one of the first telephone-based time sharing systems with the 650. I was at the time finishing my Ph.D. thesis, which was a computer program, that simulated scheduling of a machine shop and at the same time I was having trouble with my back. Right at the data collection phase of that thesis I was suddenly confined to bed. So, my wife Edythe would go over to the Computing Center in Phillips Hall in the middle of the night, sign up for time, and she would run my programs. She didn't know anything at all about what she was doing so we had an open telephone line from the console of the 650 to my bedside table. She ran these programs and would read me some results and I would tell her how to modify some cards and she would rerun them again. She was about 7 or 8 months pregnant at the time - a period she remembers as one of considerable stress.

With the successor to the 650, the Burroughs 220, in 1958, I got involved in programming languages, an activity that would last for more than 20 years. We developed a simplified student version of Algol which we called CORC (for Cornell Compiler) and that was the beginning of 3 major and a number of minor adventures into developing processors/compilers for different languages. One of the significant roles of the first one, CORC, arose when we faced the problem of the University converting from the 220 to the Control Data 1604 and we wanted to make this transparent to the students. We didn't want them to worry about which machine was processing their work. So we produced two different versions of the CORC compiler, one to run on a decimal machine and one to run on a binary machine which produced absolutely indistinguishable results. That was a pretty big deal at the time and we carried it off reasonably well.

One of the odd curiosities of that was that students at the time, for reasons that are no longer clear or evident, wrote their programs on paper using special coding forms, and they were then punched by professional full-time keypunchers. While it seems incredible today, we were trying to protect these women who were doing the keypunching from students' vulgar and obscene choice of identifier names. So there was this censoring process that if the student chose from a list of prohibited words, their program just disappeared! In the second version of these compilers, called CUPL, this was implemented within the system itself. A graduate student name Dave Freeman was involved with this and he built-in a table so that the system would automatically reject any program that used those names. Where this came back to haunt and surprise us was when an Engineering student whose last name was Fink found all of his programs being rejected because Freeman, in a burst of Puritanism, decided that Fink was an obscene term. He was not only checking variable names but also student names and rejected the program. We discovered it was very difficult to delete that word from the prohibited list because Dave had gone to the trouble of encrypting it, so that even with the source program we had trouble finding and detecting where that offending word was in the overall system. Fink has to use a pseudonym until another student, Howard Morgan changed the list.

With CUPL we began trying to make the programs error-correcting instead of just error-detecting. This doesn't make so much sense today but when you think back in terms of the mode

of computing where 24 hour turnaround was considered pretty good in most computing centers, it was a little frustrating to have your program come back and tell you that you misspelled a key word "for"! Obviously if the program knew you had misspelled the word why in the world couldn't it correct it? So we started rather gently making corrections of that sort. With characteristic academic enthusiasm we decided the objective was to make sure that any program submitted would reach execution, no matter how badly it was formed or spelled or punctuated. Eventually, our conceit was that you could submit any sort of random nonsense at all and you would be deluged with corrective remarks but one way or another an executable program would be produced. In the extreme case of course it had nothing to do with what the user intended but it was demonstrable that the corrections were often useful enough to reduce the number of successive passes, thereby the amount of time. We had data that suggested at a bare minimum we had cut the duration in half to get a corrected program. For entertainment we often would submit programs in Fortran or COBOL. There were some special tricks provided so that CUPL and later the more extreme form that came with PL/C, where we could actually run Fortran programs with some hope of useful results.

My research was still involved with simulation. We were busy writing software that would permit people to build simulation models. On the instructional side I was involved with these compilers. By the mid 1960s when it was clear that Cornell was running out of computing capacity with the 1604, I was involved somehow in the planning group that was trying to decide where we go from here. We brought a consultant to campus, named Bruce Arden, then at the University of Michigan, later at Princeton, who studied the Cornell scene at some length and wrote a report which apparently is lost from the archives. But, the essence of it was that there was a strong economic argument for consolidation and centralization. What was known as Grosch's Law at the time, basically said that the power you got out of the machine varied as the square of its size or cost or resources. So, if we could pool the money being spent on several machines at Cornell we could get some extra computing power. The idea was that the extra capacity would be worth the personnel, administrative and political hassle in trying to get different communities to cooperate on the same machine. It was a convincing argument at the time. Arden was shrewd enough, or lucky enough, to leave the report and decline to follow it up when he was offered the job of implementing his recommendations. By a process I don't mercifully remember, I got drafted to do that.

Let me back up a moment to get this in chronological order. In 1964, Bob Walker and I with strong support from Andy Schultz began trying to figure out what should be done academically and concluded that computing was going to be important enough that it shouldn't be buried in Industrial Engineering, which is where I was at the time, or Mathematics, where Walker was, or Electrical Engineering. This was going to be an important enough discipline that it deserved identification and separation. In retrospect I think that was one of the best decisions we made and gave Cornell a very substantial head start over other institutions that buried it in within Math or Electrical Engineering or both. But in any event, we spent 1964 writing proposals to raise money (and we got a $1 Million grant from the Sloan Foundation), writing proposals to various College and Graduate Faculties at Cornell to accept the idea of a new graduate field and a new organizational unit. The decision was made to make this an inter-college department that would be part of both Engineering and Arts to diffuse the rivalry or the jealousy that would result in separating it from Math or Electrical or Industrial Engineering. That too turned out to be a rather good decision in retrospect.

By the spring of '65 we had all these pieces in place so we were able to go out and recruit and bring some very remarkable scholars to Cornell. It was a very small field at the time; there weren't many people to recruit. We were extremely fortunate by getting three first rate people who agreed to come only if they all came. It was an all or nothing proposition, which kept Walker and Conway up late at night. We won that one. Juris Hartmanis was one of the scholars. The other two faculty were Gerry Salton who was recruited from Harvard. He was a specialist in Information Retrieval and became basically the leading academic in the whole world in that subject. Many of his ideas became even more important with Web browsers and the Internet. The World Wide Web depends heavily on the things that Salton was pioneering in the 60s and 70s. The second man was Pat Fischer, who was another theoretician. Pat was an assistant professor, no Pat came as an associate professor with tenure, I think, but I'm not sure but it may have been a year or two before he had tenure. Salton and Hartmanis both came as tenured full professors. Those three were the new outside professors, Walker and Conway were offered to the department, they didn't have to take us in, but chose to do so. Chris Pottle from Electrical Engineering was also part time in the department. Those 6 were the original faculty in the department. So, at that time I moved my academic appointment from Industrial Engineering to Computer Science and I became one of the original faculty in the department. But then a year later I took leave from the department to see if I could help, at least change, the university's computing facility.

Hartmanis became chairman and he acknowledged the fact that he stepped into a beautiful situation where the main political battles had been fought before he had to show up and the financial battles were at least postponed by having a million dollars of Sloan money in addition to what he could get from the colleges. Pottle basically retreated back into Electrical Engineering, Walker retired in the early 70s and I stayed in the department until 1983. I spent 20 years in Computer Science.

I took a two-year leave from the academic side in 1966 and was the first director of what became the Office of Computer Services (OCS) with responsibility for changing our equipment, upgrading and getting more capacity, and trying to take advantage of this square phenomena by consolidating several different computing organizations so we could buy a bigger machine. There were some really painful decisions that had to be made at the time. While I thought we did the right thing at the time, in retrospect I agonize over each one of these decisions. They had consequences that were far reaching and caused considerable trouble in the long run, but I'm still not sure whether they may still have been the best thing to do at the time, even with the value of hindsight. Take the simpler of the decisions - location. We knew we were getting a big machine, we didn't know what kind it was going to be but it was clear the direction that computing was going to was to allow remote access to the machine, so this made it possible to consider alternatives that would not have been feasible a couple of years earlier. The sudden availability of the General Electric building at the Airport Park, Langmuir Laboratory, swayed the balance and we decided to move off campus. The people who have second guessed this decision, are the OCS staff that has been moving back and forth locating from one place to another, drifting around campus. If we had really fought for a central campus location, maybe it would have been better in the long run. But we didn't have the money to do it, and the space, and we would have diverted a lot of activity to building or obtaining space. I really don't know if history would have treated it more kindly or not.

The more important question, the organization one, was inexplicably tied to the equipment section. Old timers will recall that in 1964 IBM stunned the world by offering a brand new

concept called scalability. They offered a computer line where instead of having distinct, incompatible machines for different size operations they had this one concept, System 360 that had little ones all the way to very big ones. They were promising complete compatibility so if you bought a small machine when you outgrew it, you could move to the next size machine. So with that announcement and our religious belief in Grosch's Law we said that what we want to do is get a 360 and get the biggest one we could possibly wangle.

With that background we were faced with the IBM promotion of their first time sharing machine called the model 67, a derivative of their standard model 360/65. It was apparently a rush program to counter the fact that General Electric was making inroads into what IBM regarded as their private domain, the universities. General Electric was using the Michigan Time-Sharing System, MTS to provide interactive remote access to a machine. IBM put together an incredible road show. I think the man who led it was John Iverson. (I'm sure of the Iverson but not sure of the John, and this was not Ken Iverson who was very famous at the time for APL, but he had the same last name.) Iverson with several others traveled the country and seduced computing center directors at universities, even ones who were more knowledgeable and experienced than Cornell's, into ordering this machine. They absolutely stopped GE and MTS in their tracks. Most of the large university computing centers ordered the 67.

Cornell ordered it, joining the bandwagon. It seemed like the perfect machine to support the move off campus, it was supposed to be able to do all kinds of work. As a footnote it seemed like Grosch's law would be very nicely complemented by the very different types of computing that were involved in research and administration. These machines now were going to offer multiprogramming that would thrive on some applications which were compute intensive, while other applications were i/o intensive, some would use a lot of storage and some would use little, and this rich mix of different kinds of applications would fit together well on this machine. None of that worked out nearly as well as it was supposed to. In general the development of the model 67 couldn't keep up with the promotion and it was clear by 1966 that the machine was going to do a very very small fraction of what was promised.

Cornell, which had made plans and commitments based on those promises was in deep trouble. So the alternatives we faced at the time were three:

- We could try and postpone any acquisition and live with the 1604 which was badly overloaded already.
- We could switch to a different type of machine and the obvious candidates at the time were faster machines from Control Data. There was a 3600 and there was a 6400. We probably could have afforded a 3600 but we all hungered for a 6400 but even this was going to be a terrible stretch and it wasn't clear you could put the administrative work on that machine. So we couldn't use Grosch's Law to wangle a bigger machine. Anyway, that was the second alternative to get a conventional scientific machine, next size bigger, which would be a continuation of what had happened from the very beginning - CPC to 650 to 220 to 1604, so this would have been a conventional decision that would have made the research community very happy if somehow we could figure out how to finance it.
- The third alternative was to make the move to a multiprogramming IBM machine but as an interim arrangement, go with a 65 which was the base on which the 67 was built and take advantage of the fact that you could later upgrade to a 75 or even to a model 91

which was a supercomputer of the day. The compatibility would mean that the 65 would not have a dead end. Or conceivably when the 67 got sorted out, which parenthetically it never did, one could switch to the 67.

There was a lot of agonizing. IBM was pressuring, they were very interested in preserving the account as they were losing accounts all over the country. It turned out that in spite of later requests from Cornell to give extraordinary help, I think all they did was give us extra systems engineers and field engineers that were here at the time beyond what was ordinarily allocated. But, we decided to go ahead with the original 67 timetable with the 65 machine. I wish I knew today which of those three courses of action would have been better in the long run for Cornell. Because it certainly complicated a whole lot of things.

I wound up getting involved in attempts to patch and repair and extend some of the software that was available for the 65 because we had moved off campus and we somehow had to deliver services to campus. Some of those initial deliveries were made in station wagons, but we also had to do it over a communications conduit. I guess if I were doing it again, I wouldn't have tried to change manufacturers, change mode of machine, move off campus and integrate two rival organizations all at once. I'm a little more realistic today about the personnel difficulties of trying to put two organizations together, who didn't want to be put together, driven only by economics and an imperative from on top. We wound up fighting battles internally with some of our staff, externally between the research and administrative communities and personally with trying to help the systems programming staff, notably Bill Worley, put together an extended version of the remote batch operating system.

If I summarize much of that era, I would say that Grosch's Law was repealed at about that time. It didn't hold nearly as well to the end of the 60s and 70s as it was supposed to, because none of us really appreciated the burden that would be imposed by overhead of these systems. When you were running the 220 and the 1604 the machine spent all of its cycles computing for the user. By the time one was running a multiprogramming, even batch system let alone 67 timesharing, there were horrendous estimates that you were losing 50 or 70% of the overall machine capacity doing internal bookkeeping. Grosch's Law just evaporated in the face of that. It turns out that the visionaries, as was usual in computer science, were anywhere from one to two decades optimistic about how things would happen. The vision of a computer utility and everybody operating remotely, being promulgated in the early 60s, didn't really happen until the early 80s and then it happened only because there were rather dramatic changes in hardware. Consolidation was no longer an issue.

So I inflicted a lot of future difficulty on the University by these choices we made with regard to equipment, location and organization. I really don't know whether the University would have been better off with variations of any of those or whether this was advantageous by forcing us to address some of those issues earlier. Perhaps getting us earlier into expectations of remote computing than we otherwise would have done, by moving off campus we forced the creation of user support groups in various colleges which probably wouldn't have happened as well. So, I don't know to what extent I'm rationalizing and trying to excuse the things we did but it just isn't clear. We clearly tried to do something before it was do-able.

Let me end on this note. When I retired to return, with great relief, to the academic world, both my wife Edythe and I regard those two years as basically a lost time. Edy was home alone raising three small children and I was spending all night in the computing center. It convinced me I didn't want to spend any more time in administration, which probably saved me from certain

career mistakes I could have made subsequently. But it was not the time that I look back with any feeling of great pleasure. Painful personnel decisions that were involved in it, and it was the first time that I ever had to do things like that. It was the first time I was involved with decisions where I was not really sure of the way to go. It was painfully educational!