

An Interview with
DAVID L. SCHLEICHER

OH 381

Conducted by Arthur L. Norberg

on

24 January 2006

Rochester, Minnesota

Charles Babbage Institute
Center for the History of Information Processing
University of Minnesota, Minneapolis
Copyright, Charles Babbage Institute

David L. Schleicher Interview

24 January 2006

Oral History 381

Abstract

David Schleicher begins with a description of his background and education at Mankato State University (Minnesota), and provides details of his first professional job. He describes his early tasks at IBM, his move to IBM Rochester, and the management structure of which he was a part. Among the technical projects he discusses are the Fort Knox project, the System/38 computer system, the Silverlake project, and the various aspects of AS/400 development, including RISC processing and the AS/400. He describes programming at IBM Rochester over many years, including support software for manufacturing, integrated data bases on the System/38 and the AS/400, microcoding, and the Rochester Programming Center. Concerning management, he discusses the management styles of Tom Furey and Glenn Henry, and in a broader context, compares evaluation of personnel by managers at Bell Laboratories and IBM. There is some discussion of the patenting process at IBM Rochester. He ends by describing his role in the coordination of programming in IBM, his move to Austin, Texas, and back to Rochester, and his work on OS/2.

TAPE 1 (Side A)

Norberg: Today is January 24th, 2006. I'm in the home of Mr. David Schleicher of IBM for an interview concerning his years at the IBM Rochester unit. Can we begin by my asking you some things about your background? Your date of birth for example? Your place of birth? Your parents and your education?

Schleicher: I was born October 5th 1942 in a hospital in Red Wing, Minnesota. I grew up in Zumbrota; my parents, my dad was born in Zumbrota, and never left it. He lived 80 years in all. He never had a desire to go any place beyond the bounds of Zumbrota, took a vacation occasionally but by and large he and my mom stayed in Zumbrota. So I grew up very close to Rochester. I can remember when Rochester announced that IBM was moving to Rochester in 1956, I believe. And it was in all the newspapers. IBM had come in and bought land on the sly (so it wouldn't cause a huge inflation in the prices) so the announcement was a big surprise to the area. So I grew up in a town of about 1800 people and the leaders of the community were the doctors, the bankers, the lawyers and the blessed few who got to work for IBM. (laughter) They may have been working on manufacturing equipment, tops on bottoms that sort of thing, but as far as residents in Zumbrota were concerned it was such an elevated tier over what was traditional in southeastern Minnesota, that working for IBM elevated you into that same strata as professional lawyers and doctors.

Norberg: Regardless of what level you were working at in the community?

Schleicher: Nobody ever knew, nobody ever took the time or the trouble to even understand, quite frankly, what they were doing there. Because they'd ask for an explanation and they wouldn't understand it. "We're soldering resistors onto cards." So, always from the time I was in my high school years, IBM was always kind of like on a platform because of its reputation in the area and its impact on the area. It dramatically changed the whole profile of ambitions for people coming out of high schools into the community. First they would try to get a job at IBM. I graduated from high school in 1960 and I went to school in Mankato at Mankato State. I graduated from there in 1964. Computer programming back in those days consisted of...you had a choice of two career fields in computers at that point in time. One was you could go over and learn about bank proof machines and card entry, taught by the business area. Or you could take Fortran and machine programming language over in the math department, and that was the equivalent of computer 101 at the university. (laughs)

Norberg: I've been through it.

Scheilcher: I was a math major. I had a degree in math and a major in economics as well. So I found I was going to graduate in 1964 and I was going to have to find a job. I wanted to get married, then Vietnam happened. So I started looking around and lo and behold I suddenly discovered that finding a job wasn't all that easy. I interviewed with Univac and I finally lined up a job with St. Paul Fire and Marine. I was going to be an actuary. And then I got a call from IBM. I had sent out an application to an insurance

company down in Omaha, Nebraska, and I got a call from IBM Federal Systems Division. Would you like to come down? They had my interview from the insurance company, they'd like me to come down and go to work. So I went down and interviewed, got the job. They had one opening. The insurance company sent my application to the marketing side of IBM: marketing was not looking for field representatives at that time. So they sent my application down into the basement where the Federal Systems Group was. And lo and behold they had a guy that was single, he was going to be drafted, and rather than be drafted in Omaha he decided he'd go east with IBM. So he went east and volunteered for the draft. Basically I was his successor. So then I got the job.

Norberg: Which is 1964?

Schleicher: 1964. And the guy that I replaced volunteered for the Navy. They said, 'No, we won't take you because you have a bad ear.' He said, 'Well, maybe I'll volunteer for the army.' They said, 'Well if the navy doesn't want you, I'm sure they don't want you either.' I finally got a job in IBM, replacing a guy who was going into the service, who never went. So, I worked in Federal Systems Division first in Omaha then in New Jersey for seven years but my parents were still back here and needed help. I came to Rochester on a requested transfer when I was 29 and was a first line manager.

Norberg: What is a first line manager?

Schleicher: A first line manager is basically IBM's representative to the employees and the employee's representative to IBM. Because you are directly responsible for managing non-managerial employees, you look after their production in work. So you're kind of a combination of shop steward – the high end meaning of this – as well as company rep.

Norberg: But not a project manager?

Schleicher: No. Generally a first line manager gets a piece of the project, some kind of small project. When I came back, IBM was building a corporate wide manufacturing information system. They had one for every plant. They were suddenly discovering they had to share parts. It used to be each plant was self sufficient, but suddenly they were discovering that it made more sense to share some machinery that would basically give them some flexibility and cost savings. They really wanted a common manufacturing information system. I then took jobs primarily in the engineering side...managing microcoders. I was working on microcode for basically a data transcriber. It took your data from the keyboard then wrote it onto an early version of a disk. And all that microcode would have to do everything in a small space because you didn't have any room.

Norberg: You didn't have any room? What do you mean by having room?

Schleicher: There was virtually no memory. Memory constraints were just so tight. Well, I'll give you an example. I went and worked on the System/38 as a first line manager. We designed code at that time for 64 kilobytes of memory. You could get 96 megabytes maximum per disk. You really had to know how to squeeze. Everything was very tight and constrained. Anyway, I went to work on the System/38 while the 34 was on the market place. The 38 was designed to come in and basically be the whole new basis for the Rochester product line. It would replace the 32 and the 34 which replaced the 3. This system was going to come in and basically be the product future for small businesses with a brand new architecture that was considered to be extendable up to whatever technology you could get for what we thought would be maybe 20, 30 years. It used advanced technologies that were intended primarily for the System/360. Layered architecture, innovative database, communications in everything, integrated into the system such that when presented to the end user you didn't concern yourself about data, you didn't worry about whether it was coming from the disk, or whether it was coming from some I/O device, or coming across the communication line. It was just data to the application program, and the end user saw it basically as data as well. That was fundamentally the architecture we were focusing on for System/38. We took the architecture of this design that was coming out of the East Coast and put it on the system. And the East Coast guys dropped it because it was too complicated for them. They already had System/360. But we took it on. The thought was at that point in time that we would deliver the system to the market place in 1977. It had the hardware, the programming capability, the storage technique, database and information systems, library. It had everything.

Norberg: Now were these being developed within IBM? Within Rochester? Outside of IBM?

Schleicher: All Rochester. I don't think we got anything from anywhere else.

Everything was dreamt up, designed and produced here. It was supposed to come out in 1977. I eventually became a second line manager, managing the operations side, basically the data storage, and all the testing, the integration. It became clear in 1976 that there was no way we would make it to market in 1977. So it got pushed out again and they were now talking about 1978...so it got delayed again. The time when we got delayed again was a major trial, because we were running out of fire power in Rochester. So the System/36 basically came into the market, saving us all. We didn't ship the System/38 until 1980. And at that time, the 36 had this huge customer base of applications. The 36 had everything except profit. Rochester's humming along, selling tons of systems but they were all so heavily discounted, their profit margin was very dismal. Meanwhile the System/38 came out by the following year with the expectation that we'd sell like the System/36, but the actual numbers were much smaller. But the ones that we sold far eclipsed in profit the System/36 record. As much as the corporation was disappointed about the 38 for being late and having, from their view point, marginal success in numbers in the marketplace, and you couldn't drive a Rochester production facility without numbers, the 38 made the business case work. It made significant profit. I should be embarrassed to tell you how much profit there was in the System/38. (laughs) It was a very, very profitable system, because we sold basically to what was emerging as

a high end market because companies wanted to do centralized data processing, but they wanted to do it with store-bought applications with a minimum amount of bother. Those large companies fueled the growth of the System/38. So that suddenly we found, in the early 1980s I think, that we had two product lines that were competing now in the traditional East Coast systems marketplace. We were competing with systems from the fifties. IBM said, 'Wow, we've set up five product lines, all overlapping in the same space.' The 38 ran right up to System/360 market place. And in fact the 38 and the 36 were not competing with each other. In the early years of IBM, between, like in the 1950s, 1960s when I first worked there, into the 1970s and even into the 1980s, the corporation seemed to have a philosophy that said internal competition is very good. It was felt that led to the very best products on the line, all the right characteristics that would let the products compete. So, here sit these five warring product lines, and the corporation now was saying this was a difficulty. Plus we're starting to hear from our customers that this was confusing. Why should I buy a 36 as opposed to a 38; why should I buy a 38 as opposed to a 4300? Why are these things not speaking to one another. They don't communicate very well; they can't share applications at all, in fact they don't even interconnect very well. So when the corporation came out with a project to remerge, to consolidate the systems into a more cohesive line... (cuts out)

Norberg: Let's just work with this machine now and not worry about the other one.

Schleicher: I'll try to speak into it.

Norberg: The microphone is right there. I'm sorry for the interruption, problems with the other machine. You were talking about Fort Knox and the necessity to have a uniform platform so that the machines could all share both programs and data.

Schleicher: Moving off onto a Fort Knox type of machine was really traumatic for Rochester. We were used to building our own stuff and in fact being in competition with the systems that we were supposed to be cooperating with on architecture and marketing strategies and all these things. And in fact, the thing that probably scared everybody the most was that Fort Knox was largely driven out of the East Coast. They sent a gentleman out here to head up Fort Knox named Alan Sheer, basically to make sure that Rochester didn't get off track. It was traumatic for Rochester when Fort Knox came onto the scene. We were developing products at that time, new models and new generations of System/36, new models and new generations of System/38. They were going on in parallel to this Fort Knox effort, and it became pretty clear after a couple of years that Fort Knox was just too heavy. They [corporate] had too much invested, too much in technology and in various business systems to find a way to converge them all to one common architecture. It wasn't working. So eventually they whittled the systems down to three. The 36, the 38, and the 4300. Well, let's see if we can't get these three on the same hardware and architectural base. And we were working on that while at the same time we were working on the System/36 and 38 product line. It became obvious again as we were designing the same stuff. We had basically three different groups working. Three programming and engineering groups: the 36 group, the 38 group, and the Fort Knox group. And the guys that were working on Fort Knox were basically our best and

our brightest. And they kept coming back and saying, 'You know, we're just redesigning stuff that we already have. It's not really a major step forward and improvement to us. Some of the hardware's nice, some of the stuff that we're doing with buffers and channels and processors is good stuff. And some of the I/O devices are good stuff. But as far as the architecting, the microcode, and software and all the applications platforms if you will, we're just reinventing stuff we already have.' So a group of independent thinkers, if you will, began to take a look at seeing if there wasn't a way we could reset the System/36 and the 38 applications to coexist in a single hardware and software architecture. I had people coming into my office – By that time I was the programming manager – and I had people coming into my office on a weekly basis saying, 'Dave, why are we doing this? It doesn't make any sense. We've got architecture and code in hand that is equal to or better than what we're trying to produce at Fort Knox.' But saying we were going to go it alone again was politically incorrect. I mean this was not a message that would go real well on the East Coast. And frankly we were a little afraid too, whether or not we could continue to survive as an independent unit. We were really an island unto ourselves from 1956 straight through to the 1990s. So eventually some technical people came in and said we have a way that we can merge System/36 and System/38 architectures together and we can deliver that in the delivery window at half the price that the corporation was willing to take and allow for Rochester. They wanted the product within a certain dollar range and they wanted the product within a certain delivery window. And that was basically the promise that we made to the corporation that we'd stay within our limit. So these people came in...

Norberg: Who were 'these people'?

Schleicher: Oh, Brian Clark, Dick Bains, Pete Hanson, Dale Dahl, these were all people that had significant technical responsibilities. We had people on the marketing side, Sue Aldrich, she was at Rochester working with System/36, System/38 by now. These people basically said, 'Just let us work on this other idea,' basically let's see if we can't merge the 36 architecture, merge the System/36 application platform, onto a System/38 architecture that uses Fort Knox hardware and a new set of user interfaces that would be considered modern in 1988. And I went to Pete Hanson and said, 'Pete, you lead this team of people' – there was a whole bunch of people – that would work part time on it. They had other jobs. I said, 'You get a System/36 application running unchanged on a System/38, and we'll take it to the corporation.' And these guys went out, these people went out, and recruited hours away from people that had specialization on System/36 or 38. They worked nights and weekends, and spent a few million dollars in the process. And eventually, they were able to demonstrate System/36 applications on a System/38 running some Fort Knox hardware, a system that in time evolved into Silverlake.

Norberg: Now can I ask a couple of questions here. When you say that it cost several million dollars, was this in staff time? In overtime, or whatever? Or are these salaried people and the money is being spent on something else.

Schleicher: Well, part of it is being certainly the hours. All of these people were staff people. You could consider the hours contributory. But in fact, it was diverted from some other things. They could've been applied to Fort Knox directly. So labor was a big part of it. But we eventually built hardware. We had a machine room in the basement of Building 30. And we put together a mini lab down there. With modified equipment that was based on the Fort Knox hardware, running System/38 architecture and Fort Knox hardware. So I had money to spend there. We eventually got the 36 application running. And lo and behold it ran without change and it ran light years faster than when it was running on the System/36, because we had all that brand new hardware. So we set up kind of a little theater down in this laboratory. Put up temporary walls, some drapes, put in chairs, and we started inviting decision makers in the laboratory and in the division to come and take a look at it. And they all walked away saying, 'Holy cow, they've got 36 applications and 38 applications running simultaneously on the same machine. And that may be good enough. It may be good enough.' We had converged on a hardware platform that was consistent at least between the 4300 and the 36 and the 38. That's three systems that converged hardwarewise. And we got the two of the largest applications platforms, the 36 with a huge application platform, the 38 with a huge application platform and we've got those people working together, maybe that's good enough. So we went to the corporate management committee and we said, 'We will deliver this product within the dollars, within the timeframe that we had committed to previously.' And we would deliver it in 1987. This was in 1985 and the corporation was very enthusiastic. We didn't have a schedule, we didn't have a development plan, we didn't have anything but a clear conviction that we could build it.

Norberg: Who was head of the laboratory at that time?

Schleicher: At the time that the proposal was made it was Tony Mondello. We were working on the Silverlake proposal while Tony was here. But when Fort Knox went belly up, you know, Tony took the fall. They brought in a new guy, Tom Furey, to take and produce this new product that we're talking about, this diverse product. Tom showed great courage. I mean, he took this out to the corporate management committee, basically said, 'Here's what we're going to do. I don't know whether our chances are one in a million, one in a thousand, it's clear that we're entering uncharted waters here. We just have the Rochester reputation and the Rochester commitment to do this.' So Tom came back with the order from CMC to go ahead and we were off and running. You're on your own though. Ask for help from anybody, but this is a Rochester project, get it done. Well it wasn't quite that. We had a software laboratory in Toronto that helped with the languages and utilities. But it was basically Rochester...

Norberg: What was the software you were using?

Schleicher: We had Cobol. We had Fortran. We had PL/1,...and we had a proprietary language that was unique to the 36 and the 38...

Norberg: RPG3?

Schleicher: RPG. RPG was the biggie and RPG3. We had all these languages.

Norberg: Steve Schwartz?

Schleicher: Not that, not that. This was a different Schwartz whose name escapes me. Anyway, Rochester got the job. No schedule, no build plan, just the conviction that we could do it. The first schedule I got back said that we could deliver the system in January 1988. I said, 'No, no, no. We can't do that. That's outside of the window that we committed to.' So they said, 'Let's...we're going to have to come up with something brand new. Something different.' And they thrashed around a bit. We'd always had a process at Rochester for developing programs, systems software that allowed for change. It really did. Our background was being change oriented. Every release came out and had big changes in it. So we were able to accommodate change. What we never had was the basis for a continuous integration of change. We said, 'Let's build the tools for the process that allows us to build the system piecemeal and make changes every step along the way. And this process, it was probably the heart of Rochester's invention I think. Putting the pieces of platforms together requires creativity, but the real innovation was how we built them. Because we had very short limits, a couple of years, and we had no build plan, we really didn't have a build plan, we were building this thing learning as we went. So we had to make a process of continuous integration, continuous integration of change. So we'd take building blocks put them together and turn it into a prototype. Then we'd take other parts and add it to this and keep growing these things piecemeal. And as we went along the way we'd find things that we'd have to say, 'Oh no. That

wouldn't work or this is a better thing.' Because we were designing it as we went and it was basically designed by trial and error. The user interface was particularly made that way. We were developing the user interface in piecemeal, all the way along. But that enabled us to absolutely minimize the documentation. We didn't spend a minute writing specifications. We didn't spend a whole lot of time coming out with product plans and all this stuff. We'd just build it by prototype and we'd show it to people. And suddenly we discovered that we had enough of it together, we should start to show it to people who really mattered, and that was the people who were going to buy the system. So we began a process of basically field sharing. We brought users to Rochester and we built a laboratory that they could wheel out, test any application. And they obviously saw things that they didn't like. And we'd have to go back and make changes. It didn't do any good to have them in here if we weren't going to listen to them. So they told us as we went along, Don Kastella had a group of people that brought these applications to developers and customers into these laboratories. And they were working on engineering level hardware and fundamentally pre-system test software. So they encountered the same things we saw, the same problems we saw. Those two things, continuous change and continuous integration, plus early access to customers and applications made the AS/400 successful.

Norberg: This had never been done before?

Schleicher: Not to my knowledge. Corporations had user groups and things like that they were starting up. Poughkeepsie for example had advisory councils. But we never let them

into our systems a year before product release. We had a very secretive environment. If you wanted to surprise the market place, we had code names for everything and everybody was, you know, had top secret restricted documents, all this other jazz. So opening up the doors of the laboratory and bringing in hundreds of application developers and customers, individuals, was really considered revolutionary.

Norberg: Now, how did corporate respond to this?

Schleicher: They were worried about it. You know, "Tom used his head." Tom said rather than building a wall, we'll bring in a review team of respected people across the corporation. And we'll bring them in once a quarter, and we'll show them exactly what we're doing and what we've done and where we're at and where our schedule is and everything else. They'll review it in about a week's time and then go back and report to the corporation, tell them what we were doing. And I think, that lack of secrecy kept the trust level up in the corporation. Plus the fact that people who came out and reviewed it said over and over again 'Wow!' They went back and sang our praises. I think because they were connected to laboratories and they would have loved to have that kind of freedom.

Late 1987, we learned the earliest we could possibly ship the system was in the second quarter of 1988. And I remember the Vice Presidents came out, and we had to tell them that we weren't going to deliver in 1987. That was a big deal because we were confident that we could maintain Rochester's revenue and profit through 1987. The System/36 and System/38 profit that we had that were still flowing into the market. 1988 it'll go in the

ditch. And 1988 was a bad year to go into a ditch because we had profit problems in the corporation in 1988. And Rochester may be small potatoes relative to Poughkeepsie, but we produced a very nice profit all the while.

Norberg: But why would the revenue go into the ditch if you had a new machine still going out of the production line, were there people waiting for what became the AS/400?

Schleicher: Well, with all these application developers and all these customers coming in, well word got out. It's impossible to keep it a secret. It was a poorly kept secret anyway, you know? (laughs) So, customers could see that they didn't want to wait to upgrade until 1988 so they said, 'As long as my investment is good on the System/36, System/38, we'll continue to buy systems. But if it's going to come out in 1988, they will want the new stuff. And so they bought and they really did. We knew that was going to happen. But when we couldn't make early 1st quarter...when we said we were going to make second quarter...late first quarter, we thought maybe in March (we thought it would be more conservatively be in 2nd quarter) that we could make up sufficient volumes from March in nine months...we could make up sufficient volumes to maintain Rochester's revenue and profit for 1988. Well, we kept working on it, and it was in system test and we got all kinds of good news relative to application packages that we were going to shift. But we couldn't meet the quality part by March. It was supposed to be done but there were still too many problems in the system for us at this point to just ship it. And we said that 'we need another six months.' Which put us out into September and that caused great alarm. Because there's no way we could take and make up volume in 8 and 9 months. So I said,

‘What are we going to do?’ What we ended up doing, we couldn’t place an early ship date in until late June. We said we will ship to customers, maintain it in the field, in sufficient quantity to make our numbers for the year. That was possible for a couple reasons. First off the quality was pretty good. Second we had a lot of customers who were already running their applications on it, on the hardware. So we knew their stuff worked. So we started selling basically systems in late June, early July. Pre-ship. Rochester’s version of a beta test program. To customers who had been in full cooperation with us. And others, who understood that they were getting early ship hardware and they were willing to take the chance.

Norberg: Can you tell me what early ship really means here?

Schleicher: Well, we weren’t done testing.

Norberg: Ok, but the machine is ready to operate in all its functionality.

Schleicher: Right. Everything is functioning. Ninety percent of it worked flawlessly. They can put it into an establishment, they can run their applications, and they can run the business on these systems. Most delayed cutting over to running a business on it which I would have, you and I would do anyway. You wouldn’t just throw out your System/36 and take your chances with a new system... (laughs) But they were willing to go so far as to purchase this system. They weren’t on lease; they were generally purchased

systems. So we announced in June, we did a formal ship data in September and an early ship program starting immediately.

Norberg: Something like 400 machines were pre-shipped weren't they?

Schleicher: Some number in that vicinity. But the other good news about that is that the people that bought those systems bought big ones. Because they were the big people at IS departments that could afford to do this sort of thing. They could bring them in and start cutting over the applications and do those type things. So they bought big machines which helped our revenue total, and big machines also helped our profit total. The net result was we made our numbers for 1988. Everybody was singing our praises.

Unfortunately we shipped about two million, two and a half million lines of software which we didn't develop it as quick as we thought. And if you shipped this stuff at the rate of one error per thousand lines of code, you're shipping 2500 hundred errors. And nobody in the corporation knew how to get their quality to better than one per thousand. I think that one per thousand was considered to be pretty hot stuff on new software. So all of a sudden there's that problem. We knew we had this built-in nest of problems that were going to have to be worked out over time. And the other problem is that the System/36 group had been writing applications using System/3, the 32, the 34, the 36. (laughter) Ok, and what they had done is they found all sorts of tricky ways to do things. And little ways to get behind the RPG and build over the execution level code. And there's no way we could test all those out. And frankly, since it was below the application base of new software, there was no 36 execution level code running, it was

System/38 execution code that was running. So all of a sudden, where these applications were called up we were getting noise, in the first quarter of 1989, we got noise from application developers that said, 'Wait a minute! What is this...all these bugs, problems!' Steve Schwartz came out and basically said you people are a disappointment to me. (I was there in the room; I was back on a business trip. I wasn't even working in Rochester. I was there for that.) And you people have let us down, so as of right now we're no longer going to be working on release two, we are going to fix release one. Every problem will be treated at the very moment. So, we basically turned the laboratory into a field support group. Not entirely, there was also other work going on. We basically saturated the market place with developers and field support people. I remember one customer interviewed in one of those trade magazines: 'You know,' he says, 'I have a problem with the AS/400. I hate to report it to Rochester because as soon as I report a problem, within twelve hours, I've got three people in my lab area,' (laughs)... 'that want to work on this problem. It interferes with my ability to do work.' And our customers and our application developers stuck with us but it was a bad 1989 for systems development. And it just delayed some of the future development, but I think it cemented particularly the System/36 buyers confidence that Rochester can deliver. People were smart enough to realize that it was really System/36 running on a System/38 for that is what you've got going here, and new hardware tucked in the new system. I think 36 customers or 36 application developers were worried that Rochester was abandoning them. And when we came to the rescue in 1989, I think we convinced them otherwise.

Norberg: Now, let me drop that for just a few minutes. I want to get something else clear. First of all when System/36 and 38 converged into AS/400 that only fixed two of the systems that the corporation was concerned about. It didn't fix the other three that were sitting out there in the market. What happened to them?

Schleicher: Well the 4300 went through a convergence with System/370 products. So the 370 product and the 4300 converged. So that simplified that level even more.

Norberg: Now we've just got one sitting out there?

Schleicher: That one we did away with. We basically said you're not going to do that one anymore.

Norberg: That's system 1? or 7?

Schleicher: Series, I don't remember.

Norberg: Long ago...One of those anyway.

Schleicher: That one they basically said we'll live without it. That I think had a lot to do with the fact that people were starting to get notions of PCs. And eventually IBM came out with the personal computer. And I think that they said, 'Well we'll just build it.'

We'll do scientific processing using a different mechanism, we'll do networks of small computers.

Norberg: Did the AS/400 line feed into 370 line as well? That is, it provided some of the pre-processing?

Schleicher: We were prevalent in large corporations that had centralized processing using the 370. So you might have an accounting department for example, that would be using an AS/400 that was feeding information into a centralized database that communicated material through a 370.

TAPE 1 (Side B)

Schleicher: This stuff's evolving though. The way customers wanted to run their shops was evolving all the way through the 1970s. In the 1970s everybody was still very much focused on centralized data processing. Large IS departments, everybody running on centralized computers, most of it done in batch, very little of it done interactively. Then we went to a thing called distributed processing, and that eventually evolved into client server processing. All that was taking place basically in the 1970s, so the System/38 hit the market place as a distributed systems, so did the 36 eventually. Small businesses bought the System/36, but large corporations bought it in quantity to do distributed data processing. System/38 was purchased as a...regional node in a distributed processing

network. It would be basically doing centralized data processing at a departmental level. And then departmental level information is consolidated into a corporate level. And that eventually evolved in the 1990s, actually in the late 1980s, into client server processing. One other thing happened that made a big difference in the AS/400. We wanted to get into the client server business.

Norberg: We? The corporation or Rochester?

Schleicher: Both. We were selling big 370 versions of servers, but corporations wanted to do distributed client server processing. They wanted to have nodes distributed throughout the corporation that ran PCs. We wanted to get in on that business. But the corporation still considered the AS/400 Rochester product line as kind of a fringe business. 1989 it was still 370s and by 1990 it was 370s with PCs. And Rochester, you just keep doing your little 5 billion dollars worth of business out there (laughs) and keep bringing in a hefty profit out of that 5 billion dollars, and we won't pay a lot of attention to you. But there still was a feeling that we were still too far from average, because we continued to produce our own, we had our own processor available. We could actually produce prototype processor technology right in our Rochester laboratory – or Rochester site. We had that business going until the late 1990s. But the corporate decision makers on the technology said, 'We want you off your proprietary processor family, and onto RISC systems. You can't continue to keep the Burlington factory going because you don't generate sufficient volume to get the price level, the cost level down so that we can make a profit on this, on these processors, as processors. We want you off that

proprietary stuff.’ So we hemmed and hawed around, and eventually decided we would go with RISC engines. In 1990 we retooled the AS/400 family over onto, onto this platform. That was very expensive. A lot of engineers would probably tell you today that it was a step backwards in technology as well as performance.

Norberg: Why’s that?

Schleicher: Because RISC processing (reduced instruction set computer) in order to get to performance from a reduced instruction set you had to build high functioning instructions. And the reduced instruction set by the time we moved onto it I think had 300 instructions going. A large number of very high function instructions had eventually evolved into RISC, but it was supposed to be a small volume of small function instructions that would perform like crazy. A lot of this from parallelism. By the time we moved onto it, it had this large instruction set and it didn’t convert – it was hard to take and put that instruction set into silicon and make it work correctly.

Norberg: Has that problem been solved incidentally?

Schleicher: Oh yes, big time. RISC engines today are....well, first off RISC engines gave the capability of a lot of things on the AS/400 that we couldn’t do before either. All of a sudden, because RISC engines were built to be stacked so that we’d run a lot of parallelism. It could have multiple processes working on a single application stream that could work at the same time. Parallelism would give you very high speed, very high

throughput. It allowed the AS/400 systems to put other environments on the platform, like the PC applications, running as a parallel processing platform on the AS/400. RISC basically makes that possible. I just got a note here recently from a friend at work and there are still limitations in the design of an AS/400 that prevent it from making steps that it wants to make but RISC is a good investment. It's continued growth in parallelism and basically high performance make the AS/400 much more flexible.

Norbert: When the RISC concept was added to the AS/400, is that when the name changed from application systems to advanced systems? Or did it have another name altogether?

Schleicher: No, that's when we went from applications to advanced and then eventually they changed it to stuff that I don't even...I don't understand i and e. (both laugh) It's post-Schleicher let's put it that way. Whenever I correspond with people, we still call it the AS/400 because I can't keep this other stuff straight.

Norberg: Ok. I want to go back again, because I let you go on and on because you were telling me a very nice chronological history of some of the machines and some of the projects that were instituted in order to change these and make them better. But I want to go back to the personal side again and build up a little bit of a story that could be seen parallel with the stories you've told. And that is, I want to understand something about your work activities back in the 1970s. When you came to Rochester in 1971, you said

that you had been a first line manager and that you were in charge of production? Did I understand that right?

Schleicher: I was working on manufacturing information systems.

Norbert: Ok, not the fabrication of them but the design of them, is that right?

Schleicher: When I came back in 1971, we were writing support software for the manufacturing side of the business and it was distributed...that programming effort was distributed between San Jose, Rochester, Endicott, Poughkeepsie, and Raleigh but it was basically support software.

Norberg: For which systems? Anything?

Schleicher: Pretty much the IBM system manufacturing business.

Norberg: Now how did these projects get broken down to, say if they were filtered down, to Rochester, or were they generated at Rochester?

Schleicher: They took the information systems, manufacturing information systems and they subdivided it by function. There are records that are required for engineering change control; there are records that are required for parts ordering and so forth. This was all a long time ago, we're talking 30 years ago. But basically, it was a kind of a side

line business... I'd requested a transfer back to Rochester. I'd worked on federal systems. We were working on the safeguard antiballistic missile system. That's what I was doing. I was managing people that were writing programs to drive phased array radars. You couldn't have been further from the Rochester basis of experience than I was (laughs). Before that I worked on intelligent data handling systems out of SAC headquarters in Omaha.

Norberg: For IBM?

Schleicher: Yes. Working on tracking spy satellites and converting it into a database of practical information for missiles. So I came to Rochester not to work on manufacturing and information systems. I came to Rochester because by then we thought we needed to come home. And that's why I wanted to work there.

Norberg: Why did you think you could do that?

Schleicher: I don't know (laughs).

Norberg: Did you have a choice? Were there several jobs you could have taken?

Schleicher: No, not in Rochester. Rochester was specific. Also in 1971, 1970 was the year of the wage-price freeze. We had double digit inflation. Interest rates around 10%.

Norberg: Yes.

Schleicher: Yes, and so, when I came back, we had just gone off the wage-price freeze. So people were pretty much not moving in 1971. They just weren't moving around. So I had an offer to come out, and I came out – not as a manager, I came out as a programmer. I was going to write code. And when I got there one of the managers committed an unforgivable sin and they had to remove him from management on this team. Rochester had a policy at that time because of the whole hiring freeze that before they could offer the job to a non-manager they had to offer the job to a previous manager, a person that had been manager before. Well, I had only been on the job three months. They didn't know me from beans.

Norberg: But you had managed other projects?

Schleicher: I had managed projects out in New Jersey. So they had to offer me the job. And I said, 'Sure I'll do it.' I really wanted back into management. I eventually became an advisory programmer and did technical work for Rochester in that manufacturing area, but that was a short lived interval. I had worked on manufacturing information stuff till 1974, so three years approximately. And then the project went away. You have to understand the lab environment back in those days. Back in the 1970s I think the 1960's as well, Rochester had a lot of different laboratory missions. We had guys who were working on the System/3 and gradually on the System/32, but we also had people that worked in displays. There were people in I/O devices and we had people working on

disks. We had all these various things and they all were competing for resources at the laboratory level. Each of them had a manager that reports to the lab director and had a counterpart in the manufacturing area. People that did disk manufacturing were hired and so on. It was very much a set of various disciplines that were competing for resources and dollars within the laboratory. So when I came back, I wasn't brought back into what could be considered the systems area, I was brought back into more of a I/O device area, manufacturing type of area, or support of manufacturing. So when I came off manufacturing information systems, my job offer didn't come from systems, my job offer came from I/O devices in manufacturing, in the engineering side of the business. So I managed microcoders for I/O devices for about a year.

Norberg: That would be about 1974-1975.

Schleicher: Yes, late 1974 to summer of 1975, so about nine months. I enjoyed it, it was fun.

Norberg: How many people?

Schleicher: 15.

Norberg: And who's the boss?

Schleicher: I worked for a guy named Rick Catron and we worked for a guy named Al Cataia. And Al Cataia worked for another guy. But it's interesting there are so many different personalities back in those days because you were allowed a lot of the time a large degree of flexibility. And we were so competitive. You didn't get in charge of stuff because you were there. You had to compete. Your dollars and your resources and everything depended upon how effectively you competed with your peers. So you ended up with a wide range of personalities for this job. I could tell you stories about people, you know (laughs). It was exciting, a lot of fun. But it was disruptive I thought. Because you never knew from day to day if your money was going to disappear, or be channeled off into somebody else's because you know, somebody had a notion that this is a better way to spend their money than this other work.

Norberg: You just mentioned about 4 levels of the management of the laboratory in that description you just gave me. Is that right four levels?

Schleicher: me, 2nd line, 3rd line, 4th line.

Norberg: five levels of management. There's a lot.

Schleicher: The programming center was there. It was 1st line, 2nd line, 3rd line, programming center manager.

Norberg: That seems like high overhead to me in the corporation. Is that a fair statement?

Schleicher: I don't think so to be honest about it. IBM always had had a management process or always had it. I don't know what they have now. They always had a management process that basically said, 'We want to manage people that do the work at an individual level. We want the first line...the person that manages them to be responsible to know what their work assignment is, to make sure the work assignment is up to their capabilities and their capabilities are up to the work assignment. And that we're going to appraise them and pay them. We're going to motivate them. And we are going to see to their happiness.' And that means you got to have somebody that's with them on a day-to-day basis. A large department of people, 16 people, that'll keep you hopping if you're going to manage individuals on a day-by-day basis.

Norberg: Would this be true throughout the unit, including the manufacturing assembly line and so on?

Schleicher: Yes, that's the way manufacturing works. But still that same philosophy I think is prevalent throughout. Sixteen would have been a big department, a dozen would probably be a better average. I was always fortunate to have a larger personnel department because of the nature of the stuff that I worked on. A large first line department is about 16. I don't know what they are today. And the management philosophy has changed. I can remember going into our IBM management school. As a second or third line manager...third line manager...and we had a guy stand up – this was in the eighties – and this instructor stood up and said if you have people that are unhappy

with their salaries or unhappy with their jobs, you have every right and good sense to tell them to go look elsewhere. Tell them to fill out an application and go interview elsewhere. Find out what it's like, if it's better, leave. If you suddenly discover that the grass isn't green then maybe you come back and be a happier employee. I was flabbergasted. I'd never heard of such a revolutionary idea. (laughter) Because when you signed on at IBM you had joined a family. You don't just walk off and leave the family. That's divorce! or abandonment!

Norberg: But it makes sense, when you think about it.

Schleicher: Yes, but I was 'Wow!' (laughs)

Norberg: You need to know about what the competition is. Ok. Now when you're a third line manager how many people do you have under you? Are you just dealing with the second line managers and what's the ratio there?

Schleicher: Boy that varies all over the place but in the System/38 business...I'm most familiar up in Rochester, you generally had...a third line would generally have five, seven second lines working for you, in that range. A second line was probably about that same number, five to seven first line.

Norberg: Because I've been puzzled as I've read Rochester News over the thirty years or so it was published, we've seen all these people promoted to manager of this and

manager of that...why the number of managers. Now it's beginning to be a little clearer to me. Let me ask you another question about contributions made by software developers. I'm thinking of it in the following way. In the 1970s corporations began to patent software. What sort of contributions were made either by the Rochester lab or by your group to the development of new concepts in software, whatever they were, no matter how small?

Schleicher: One of the big contributions I think...one of the things that probably had a patent was the integrated database we started in the 38. The integrated database on System/38 was considered a revolutionary thing. Database was always treated as an application. You can buy DB2, IBM's first database application, or whatever you call it, was DB2 and everybody that knew the System/38 said, 'Wait a minute! We've had one out for years and it's built right into the operating system.' So there is a lot of patents that were written on integrating database into an operating system so that basically you could access data at the operating system level out of the database as opposed to having to go back up to an application. You have a horizontal path to a database. We also layered software – I think pretty imaginatively – that was the old...that was the new operating systems architecture that was put on the System/38. It was structured. We had a horizontal microcode layer and a vertical microcode layer and an operating system layer on top of that with applications, languages and tools on top of that, finally you've reached the application. And that sounds long, but everything compiles down into vertical microcode, which deals directly with horizontal microcode and hardware. So structurally it looks long but from an application point of view, its really very short. Putting that

structure in place took five years, 1975 to 1980. But it was basically the underpinning of Rochester's continued presence as a leader in general purpose data handling systems for the last twenty-five years.

Norberg: I know that the support systems were licensed. If the integrated database concept was patented, were the support systems patented as well?

Schleicher: I'm not sure I understand what you mean.

Norberg: I guess what I'm really trying to understand is whether or not the operating systems that were developed for any of these systems were patented.

Schleicher: I think there are tens of thousands of patents right down to very short paths. We had a Data General lawsuit in the 1990s. They claimed we infringed on their patents on very short code sequences, I can't give you specific examples but various small code sequences. And one of the people that used to work for me Roger Taylor spent years working with corporate lawyers. He came back to Rochester to work with the lawyer because Data General is a direct competitor of our products here. Saying, "no, no, no that stuff came from the System/38, the System/38 had patents back in the 1970s on very short data strips." Not big sweeping things. I don't know what ever became of that lawsuit. They wanted billions.

Norberg: But now these patent things here, how does one keep track of all of this so that you know a) when you have a new concept and b) when somebody is infringing it?

Schleicher: IBM I don't think was ever so concerned about infringement. I don't think we spent a whole lot of time going out digging through Data General's stuff seeing if we could find an area of infringement but when Data General said that we were infringing, we wanted a solid basis to prove that that's not what happened.

Norberg: But now somebody has to know what's going on in the field in order to be able to say, 'No, this concept was developed here in Rochester, we patented it.' Is there a large group of people who are doing this inside Rochester?

Schleicher: We had patent attorneys. The way the process mostly worked was we'd come out with an idea and the first line manager would go to his department. He'd say, 'OK Guys, let's get patents now. The company wants patents so let's put together patent applications. You guys know what you got out. What do you think is new and revolutionary because you should know your specific area.' So it was more a bottom-up patent process. There was motivation to write patents. It looked good on your resume. There was money involved if you could come up with suggestions that were patent ideas. It smacked of professionalism. So people wanted to write patents.

Norberg: Ok, but patent applications cost money so was the corporation keeping a cap on how many could be done?

Schleicher: I never recall anybody ever saying, 'Don't put in a patent application.'

(laughter) Not all of them were accepted. You know they might get rejected by the patent site attorneys responsible for researching it. They were the first approval level at the plant site level. Then eventually if they were big enough, if they were considered an important patent, researchers would get involved...they would add to it.

Norberg: But I understand there were priorities within the corporation. Ok so not everything that caught the attention of a patent attorney would necessarily be sent to patent office.

Schleicher: But it was very much protection as opposed to looking for monetary gain or anything like that.

Norberg: Were there other significant patents that come to mind beside the integrated database?

Schleicher: Horizontal microcode. It was tied to the processors, early innovation that had been patented. The high level machine interface as we referred to it, it was an interface at the vertical microcode level that was heavily patented. The ways that we get basic operating system functions, storage management, exception management, process management, all these sub-functions of operating systems and how they worked in a virtual address environment were heavily patented. The System/38 had the idea of a

single store space, probably another major innovation that came out of the Poughkeepsie work. We can't claim credit for the idea. But we implemented it. Everybody else kind of divided memory up: Real memory that's addressable, memory that's on the disk, and then there is memory that is virtual that an application deals with. Applications deal with virtual memory which translates into main memory shares space with a hard drive via paging. The System/38 single level store basically that's all one address space. Well that's a huge problem because we've got data flowing all the time so how do I make sure that I maintain integrity within the virtual address space. That stuff is all down in the vertical microcode – storage management, exception management, process management, all these different management techniques that are required to manage a single level store. That single level store concept is still alive and well after twenty-five years of reimplementation and reimplementation and reimplementation.

Norberg: It seems to me that Frank Soltis in one of his two books, I can't remember whether it's the AS Book or in the i Series, said something to the effect that virtual memory had never really been as successful as the concept that was developed by the men in England until single level storage came along. Do you know whether that's true or not?

Schleicher: Let me offer you a word of caution, ok? (laughter) Maybe you've already heard it. Frank Soltis is a very good guy; he worked on the System/38 primarily as an engineer. In time he came to work at the design level and eventually he was able to marry engineering level expertise with system programming expertise. But people that refer to

him as the Father of the AS/400 will raise up a thousand eyebrows, ok, (laughter) amongst the developers, because Frank had a rare talent. He was able to take engineering and software and make it understandable to laymen. And as a result, Frank was forever out talking to Common or consulting meetings or so on. And he has a very convincing style and people in marketing just love him, consultants love him, management loves him. He could go into a customer environment and speak in layman's terms about very difficult topics. He didn't raise the title of Father of the AS/400, users did, the outside world said, 'That's Frank.' And the marketing guy says hot damn that's good because every time I can dress up and trot out Frank I'm bringing out the Father of the AS/400. Frank's no more the Father of the AS/400...(laughter)...than I am.

Norberg: I had heard that before. There's something I want to ask you, too, which doesn't have to go beyond us if you don't want it to. I have read both of his books, I have read the Silverlake book, and I've read a couple of things in Common that were interviews. And I get the feeling that there's a great deal of exaggeration in his work, not about the technical details, about what the machine could do, but about who was responsible and how Rochester was somehow working against the corporation for a while trying to develop these new products and were sort of doing it under the table more or less. Now I find that hard to believe of a corporation like IBM. What's your attitude?

Schleicher: I've got a couple of books. I got the one by Roy Bauer and those other guys. The books that I've seen are all written by people from the outside looking in. There is an article out, it actually has my name on it, written by Roger Taylor. Roger wrote the entire

article, he came into my office and said, 'David, I want to put your name on the cover.' He said, 'I want you to be the co-author of this article.' I said, 'Rog, I didn't do anything?' And he said, 'I just think that of all the stuff that's being written about the subject, to not have one with your name on it is a gross injustice.'

Norberg: Hmm, generous.

Schleicher: Yes, to a fault. I said, 'I can't do that.' 'Well, it's my article, I'm going to do it.' He said, 'All you have to do to make a contribution to it is write your resume for the foreword, what you want to have in the foreword.' So if you Google on my name there's a reference article out there that's got my name on it, but it's written by Roger Taylor. Roger Taylor was probably the smartest guy in Rochester. If there was somebody that you could say was the father of the...was the grandfather of the AS/400 its Roger Taylor. He was very, very instrumental in the software element. As a manager, he has a very quirky personality. People either loved Roger or hated his guts. He eventually became the second line manager of the operating system level of System/38 during the development. When we went in a ditch in 1978, they shot Roger Taylor and from that day forward Roger was never able to hold a key main line management position. They used Roger extensively on complex technical projects. Particularly if they were being driven from out in New York. If somebody says we want...we have a project we want Rochester's participation in it. If we thought it was dangerous, we'd sent Roger. (laughter) Roger was our technical pitbull and we used him in a wide variety of things. As a result he just has

an enormous technical background on the AS/400 and the System/38 and the System/36 and the System/3.

Norberg: What was the title do you remember?

Schleicher: You'd have to Google on my name. Roger lives in Pine Island. I tried to find my copy of the article. It's in a box some place. It's not enshrined someplace because I never felt it was...you know.

Norberg: What is the article about?

Schleicher: It's basically the detailed explanation of the development of the AS/400 from a technologist point of view. But you're going to get...but Roger is probably the most objective individual I've ever know. He also has a photographic memory and he also has worked on so many different aspects of the system, of the Rochester laboratory because like I say he works...he was the dark angel. (laughs) And he worked for me for a number of years, he went down to Texas with me after I left Rochester in 1994. He wrote this article and it is probably the most objective explanation of what really transpired.

Norberg: I wonder why we haven't come across it.

Schleicher: If I can find my copy I'll send it to you.

Norberg: Well, there must be one in the IBM library. I'll check. If I can't find it though I would like to get it from you.

Schleicher: Because...there were books that were sanctioned, Roy Bauer's book for example was kind of a sanctioned book. It was basis for a Baldrige award. It was a politically correct book that was very heavily sanctioned. Roger's article was something Roger did and he didn't make a big fanfare about it. I don't think he got any recognition whatsoever.

Norberg: That's a disappointment I'll bet.

Schleicher: Yes. But Roger did it... I'll give you an example of Roger Taylor. Roger Taylor lives in Pine Island. He's forty-five minutes away from the University of Minnesota. Roger Taylor for thirty years every Tuesday night attended courses at the University of Minnesota. He took courses that were interesting to him. He has enough credit at the University of Minnesota for like seven Masters degrees or some enormous number. He would go up and he'd take a course or two, he'd probably take two or three at a time. And he would take it and if it was interesting to him he'd complete it and get an A. If he didn't...if it wasn't that interesting, he'd drop it. So he says he's got this huge resume of classes. (laughter)

Norberg: And hundreds of withdrawals!

Schleicher: Exactly. (laughter) He has a library in his basement. In 1994 he told me he had something like thousands of volumes.

Norberg: Wow, that's a big library.

Schleicher: He has it all indexed. At public libraries when you could get people who come in – high school students, college students – for research and they don't have a book they often referred them to Roger. More often than not Roger could provide them a book. Got a photographic memory. But there really is this well done reference article by Roger that is probably the most objective description of what was accomplished in all those years.

Norberg: What gave rise to the formation of the Rochester Programming Center?

Schleicher: Well, we had decided that we were going to do Silverlake. We said, 'We need to put all our programs in one bucket. We'll put all the engineers in one bucket and gave them to ...Jim Coraza came and took all the engineering and I took all the programming. They were going to have a systems manager and we had a systems manager on the site Jeff Robertson – the System/36 family systems manager. Jeff and Tom never saw eye-to-eye.

Norberg: Tom Furey?

Schleicher: Yes. So when upper management said they were going to put a systems manager in charge, they said Jim Coraza would have that additional responsibility. What that meant was that he had the planners working for him as well as the engineers. He had the planners and the engineers. So all the programmers were working for me. There was a guy out east who's name was Earl Wheeler. Earl was the boss programmer at IBM, senior executive. Earl blest all programming throughout the corporation.

Norberg: What does blest mean?

Schleicher: If you wanted to take and put a programming center in some laboratory, you couldn't do it unless Earl said, 'uh huh.' Earl was very jealous of the autonomy of the programmers and he didn't like this business that had his programmers working for system managers who had engineers. He felt that that was too parochial of an environment. You should take the programming aside as opposed to being commingling out there with undesirables like engineers. (laughter) Undisciplined people. (laughter) So Rochester said we're going to put all the programmers together. Earl had been to Tom and said, 'Tom, Tom , Tom, when you get out to Rochester I want you to put a programming center in place.' He said, 'This business about having all these programmers strewn all over the place with system managers as their leader...no, no.' He said, 'I want a programming center out there.' So Tom came out with a charge to put a programming center in place. Now Earl had been on Rochester's case about this. From his point of view, the structure of not having programmers all in one nest was wrong.

Norberg: Do you know what the advantage in that was from his point of view?

Schleicher: His point was that it's the only way we're ever going to get any uniformity for software in the corporation. Because if I have to deal with a hundred different system managers, they all basically can give me lip service and then go back and do whatever they darn well please. But if I can get them into programming centers then basically I think I can get these programming centers to work together in tools, in languages, in applications, in architectures, all this jazz. He said, 'I can't deal with a hundred different system managers.' So he sent Tom out – he had been working on Tony before Tom. Time to get that stuff straightened out in Rochester because we were one of the last. So when the AS/400 came up, Tom said, 'We're going to keep all the programmers together.' And he called up Earl and said, 'Earl, I think I am succeeding. I've got a programming center. Will you bless it?' Earl said, 'I sure will. Good job, Tom.' (laughter) 'What level are you going to manage it with?' Tom says, 'Well, I'm going to put a senior programmer in charge – I think this was my title at the time because I'd been a systems manager, I was a systems 38 manager for a couple of years...and that's a level 62 category. And Tom goes, 'We'll have a level 62 manager in charge.' Earl says, 'Like hell.' He says, 'That's an executive level position everywhere else. It'll be an executive level position in Rochester.' Tom had to say, 'Well, we've never done it that way in our division. There are no executives outside of lab directors and so forth...in this division, never happened.' Earl said, 'Wrong. If you're going to have a programming center, it's going to be an executive level position.' So next year I made exec. We put the place at the 62 level with

the promise that it would be an executive position. And so next year I became the Divisional Director.

Norberg: And you maintained that level until you left the corporation?

Schleicher: When I left the corporation I was an IBM General Manager. A Divisional Director is an E-1 level executive....let me think about this...vice president is an E-2 level exec...general manager is an E-3 level executive.

Norberg: How many levels are they?

Schleicher: The next level up is...general managers generally manage divisions.

Norberg: Divisions or units?

Schleicher: Divisions.

Norberg: Is Rochester a division rather than a unit?

Schleicher: When I was there no. Steve Schwartz was an IBM General Manager. When I went down to Boca... to Austin to work on a RISC version of OS/2, which we were trying to produce, they made me a general manager because I was going to have the Boca lab reporting to me, a big chunk of the Austin lab reporting to me, a chunk of the

Rochester lab reporting to me. They decided to make me a general manager even though I worked for a general manager. My boss was a general manager who managed the division. So there's general managers that aren't division managing general managers who reported to senior vice presidents who are group executives. They were IBM senior vice presidents who used to be on the Corporate Management Committee and they worked for the chief op...chief....what is it chairman of the board and chief operating...chief executive

Norberg: CEO

Schleicher: CEO, yes. He worked for the CEO. So theoretically as a general manager the CEO was my third line manager.

Norberg: Wasn't the head of Rochester usually referred to as a general manager?

Schleicher: Yes, the site general manager.

Norberg: Oh, so there's a modifier there.

Schleicher: There was a lot of screwing around there, and I think IBM tossed that whole structure, or modified it a lot.

Norberg: There were a lot of big changes in corporations in the last decade so that's not a surprise. Let me turn to something else here. This is beginning to get to the end so why don't we stop. (end of tape)

TAPE 2 (Side A)

[Comparison of operations in Bell Labs and IBM]

Schleicher: You couldn't remain an associate member of the staff. It was kind of like...it was kind of like boot camp for guys that were joining the scientific community.

Generally if you were an associate member for two years maybe three, but if you didn't make the job in 2-3 years probably knew that you weren't going to be successful in Bell Labs and probably you'd be well served to leave because promotions weren't going to happen. So you wanted to get a promotion from associate member to a member of the technical staff. And in order to become a member of the technical staff, basically a number of people would have to take and say, 'Arthur's a good guy. He does good work.'

Well, they had a fairly loose management structure, but you know there a first line manager might have 20-30 people working for him. So how do you get known by a lot of supervisors? You publish papers first off. And you attend every meeting known to mankind (laughs). If you could get in a meeting, you went to it. So, because you had to be known, you'd come to meetings and then you'd try to look good. Then you'd publish papers that people would read. 'So that's pretty interesting. I saw his name on a paper

here just recently, and I think I was in a meeting with him recently.’ It was the most published, meeting oriented environment I ever saw. (laughs)

Norberg: It’s like the University. Most major universities are that way.

Schleicher: And then, when you were a member of the technical staff – salaries were done on an area basis, three, four hundred people. And salaries are doled out by your ranking within that group of people. So they ranked them all. Let’s say there were 300 in the area. They’d be ranked 1-300. And the number one could get 15% increase, and by 300 could probably stop looking for it. (laughter) And they’d publish it, it would be on bulletin boards. So how did you get to be ranked #1? They got all the management team together for the area, and they did the ranking. Well, it’s got Arthur’s over here, Dave’s over here. How could a guy from five departments over decide? Because I sat in a meeting with him. (Laughs). I don’t know.

Norberg: When you read the Bell Labs histories, there are about five volumes out, it is basically a history of innovation in Bell Labs. And in there, the names that appear are those people who either won a Nobel prize, had major publications, or major patents. So I think it’s the same thing, if you have some sort of good press relation you’re in good shape.

Schleicher: Well, I was working at a level that you know, was well down from I’m sure what you’d consider a Bell Lab scientist level. These were the guys who were trying to

grind out the details, grind out implementations, and so forth, but they weren't sitting behind closed doors with test tubes. But it was a real learning experience for me. Whereas IBM had this very paternalistic management structure. Like I say, you know, first line managers are sitting right there. They're, you know, surrogate fathers for these 16 people. Bell Labs had...it was like shark infested waters. (Laughs). If you could get in the boat, you climbed in. (laughter)

Norberg: This is tape number 2, with Mr. David Schleicher. David let's go back to the Rochester programming center. Once it was established, however it occurred between Wheeler and Furey, what changes did you notice in the operations of programming development as a result of this centralization?

Schleicher: It got a lot more efficient. First, we got rid of all this competition baloney between the various programming groups and we became one. Silverlake was the catalyst that made that possible because everybody was fundamentally going to be working on Silverlake eventually. We still had a group that was working on releases of the 36, still had a group working on releases of the 38, but they all understood that Silverlake was the eventual resting spot for all of them. So instead of having a lot of competition between groups, we suddenly got a lot of cooperation. Secondly, we all converged on a single set of tools. So instead of having a set of tools for developing software over on the System/36 and another set of tools for developing software over on the System/38, we said, 'We ain't got time to screw around with this. The 38 set of tools is the best for what we need to do with continuous integration,' so we converged

everything onto one tool base and one language base. So all of a sudden now, everybody's working on the same stuff. Everybody's using the same tools. Everybody's under the same style of management. Everybody understands the end game. So we only have one, we don't have two or more. I thought it was much more efficient, and it was a lot more fun, too. Before there was always this infighting, struggling for resources, trying to get the...when you did recruitment for example, you recruited away from one another. This new environment recruited them into the organization. You knew they were going to float around to departments within the organization. They all come and they all go. It was just I thought a far more efficient organization. I don't know whether they still maintained it, but it was like a breath of fresh air I thought.

Norberg: Did you still have to work on other problems with the corporation though?

Schleicher: Occasionally, we were so heads down though on Silverlake when the programming center was formed that we...

Norberg: Heads down did you say?

Schleicher: Right. You're head down and your nose is to the grindstone – that type of heads down mentality. We were so heads down on getting Silverlake done that we basically didn't look up. In fact, after about, about a year I was forbidden to travel; I couldn't even go anywhere.

Norberg: Why?

Schleicher: They wanted me heads down in Rochester. So yeah, the last year, I believe. I had been working on these corporate projects like Systems Application Architecture. I'd been deeply involved in that and the last year they said, 'Stop and focus on Silverlake.' If you read this article, I say in this article that most of the directive implementation of AS/400 was done by a level beneath me, the guys who reported to me. That's true up to about the last year. The last year I didn't travel.

Norberg: This is 1987-88?

Schleicher: 1987. 1987 and 1988 I didn't travel.

Norberg: Where would you have been traveling before?

Schleicher: Out east. Purchase, New York, sometimes to Poughkeepsie,

Norberg: On projects?

Schleicher: Generally because.... the thing that was pulling me east a lot. Well there were two things that pulled me east. Going out talking to division executives, that pulled me East. The other one was this stuff that Wheeler was working on. He tried to pull the programming centers together. And we're trying to arrive at a single architecture, or

system so that we would stop building incompatible systems. Everybody would have the same communications, connectivity. Everybody would have the same programming language capability. Everybody would have the same database technology. That was the role that Earl had in mind for the single systems application architecture. That was going on during Silverlake. Because we were coming out with a brand new system, Earl was very, very interested in making sure that we had the right architectural componentry. Turns out we were fortunate because we did. There wasn't anything in SAA that was really diametrically opposed to what we were doing. Some of that came later. So I was kind of like, the darling boy of Earl Wheeler's kingdom of the programming centers, because I was doing everything he wanted. Because I had this whole brand new system, and it made sense to do the things that he wanted to do. He wanted to have a common user interface. Screen icons and how the things looked on a display tube. When you talk to a user. Well, what he was wanting, we were producing. So he was saying, 'Well, look at this, the guy is producing exactly what we want.' So every time we had a meeting he wanted me there, because I was kind of like good ole' Shep (laughs). He's got his sheep in the right pen over there, you know?

Norberg: Did these ideas about the icons and the way the software's being developed that was going on in Rochester? Did these ideas then get pushed into other units of IBM?

Schleicher: Yes, there was eventually an architecture published. I went back and managed the technical aspects of that architecture for two years in 1989 and 1990.

Second half of 1988 and through the first half of 1990. And we produced an architecture that was hugely contentious.

Norberg: Within Rochester or within the corporation?

Schleicher: All over. It was in Rochester, it was in the various sites, largely because there were two things pulling at systems development at that time. One was still the distributed/central data processing and the other the new client server. The requirements were good solid centralist programming...no, centralist system. Central or distributed systems guaranteed rock hard, iron clad data integrity. Followed only by, you know, iron clad, rock hard, system reliability. PC world, nothing could be further from the truth. What is system reliability on a PC? Well the fact that if it goes bad it'll hang up, and you'll push the restart button again and it'll come back up. You'll lose all your data in the meantime, well yeah, but that's the way it goes. So, their whole object was application friendly, user friendly, connectivity to a server. Anything that can be done, that they can make it the PC itself user friendly, to both users and application developers, was of primary importance to them. And connectivity to a server was how fast and cheaply can I shoot the data over there and if something gets lost along the way, well, what the hell. So you got two totally different philosophies, and here's Rochester trying to straddle the two. Wants to be a client server. Wants to be a server. At that time we also wanted to be a client. Eventually that died on the vine. We viewed our low end systems as client systems and our high end systems as server systems. So we were going to be both the client and the server. We wanted in but we had a history of being a

distributed processor. We had all these centralist notions about data integrity and system reliability, two driving forces that didn't mesh well with the client server environment. And that's no different than the guys in Endicott. So all of a sudden we're trying to come up with common architecture that spans all three of those environments. The central environment, distributed environment, and client server environment. And that's really four as it turns out because they eventually split servers from clients. We ended up with clients, and servers, distributors, central systems. And we want one architecture extended across the whole of them. And should we use TDSP as the mechanism for distributing communications data or SAA? Well, SAA weighs 1000 pounds, when PC wants something that weighs in ounces. The debates were awful. I went east to manage in that environment. (laughs)

Norberg: This is in 1990?

Schleicher: Yes. The way we did it, the way Earl had set it up...practically all the software IBM fellows were assigned to Systems Application Architecture. And if they weren't IBM fellows they were the guys that held the bulk of the software patents for a particular location like San Jose or someplace else. So you had all these people with these enormous investments in technologies from their environment, and egos as big as all outdoors. I mean I had people you would not believe. I wasn't in the meeting but I heard about it. Where this one gentleman, who's an IBM fellow, got into an argument with another gentleman in a large room where they're separated by tables, like in a cafeteria. And he got in an argument, he got up on that table, he walked from table to

table, so he could go over and have a chat with this individual. (Both laugh). I could tell you stories. So, while this was all going on, Rochester's stuck right in the middle of this massive problem of trying to consolidate architecture across IBM. They never did implement SAA. We published it, eventually it was published as a guide to users...this is what IBM thinks is good in this particular case. If you're going to install a central system, here's some great architecture for you. If you're going to do distributed work, here's some great architecture for you. These are the ones IBM's going to focus on in each of these environments and our goal is to make sure that you can flow from the top to the bottom, seamlessly, without having to worry about leaving IBM's product line.

Norberg: Couldn't flow from the bottom to the top?

Schleicher: You're never going to run Word on a 370, or if you do you'll be sorry you did. So all that, I guess that's my long explanation of...you asked what kinds of problems there were.

Norberg: Let me ask yet another question about that same thing. I've been looking at the technical reports, the technical report series that was published by Rochester starting in 1967 or 8, I don't remember the exact starting point, somewhere around there, and there are thousands of these. Do you remember any of those of the software?

Schleicher: I don't remember any of those...

Norberg: Ok. Well that suggests there weren't any, or at least they weren't very significant. My sense is that quite a number of these, and I've looked through...I must have looked through 10 years, just turning pages not reading text very carefully. And they were largely manufacturing things, that is, the changes in a process that would allow you to do soldering for 10 percent less and that sort of thing. But nothing at a high level, you don't see anything at a high level such as you would find in say the IBM Journal of Research and Development where concepts tend to be larger and they don't tend to be associated with manufacturing or detailing.

Schleicher: The only technical report that I remember, and I'm sure there were others that I don't recall after all these years. We came out with a very fancy System/38 technical report. And it was far less aimed at technology than it was at trying to explain a brand new concept in systems, to marketeers, customers, business, decision makers.

Norberg: So sort of a large format?

Schleicher: Yes.

Norberg: I read that. Blue cover?

Schleicher: Yes, that's it.

Norberg: Yes, I've read that.

Schleicher: That's the only one I recall. Did you think it was...do you agree with my description of it?

Norberg: Intelligible...oh yes, I learned a lot from it about System/38. One of the names you haven't mentioned with System/38 is Glenn Henry.

Schleicher: My next door neighbor. There's a case. (laughter) I never enjoyed working for anyone as much I enjoyed working for Glenn Henry. Glenn Henry could be...he was, he is absolutely brilliant. Just a brilliant guy. He's lived right next door. So I got to see him off the job as well as on the job. I reported to him directly after a year or a year and a half working on the 38. I became a second line manager. And he had all of the software development working. Glenn was brilliant. He kept a system alive that was promised to ship in 1977, it shipped in 1980. Back in IBM at those days...

Norberg: Was that the 32?

Schleicher: That's the 38.

Norberg: Oh, ok. He did the 32 first.

Schleicher: I don't even know, I don't even remember what he came here to do. He was capable of keeping that thing alive. Because first off everybody recognized he was trying

his hardest. He was totally dedicated to getting the job done. The man never slept, I don't know how he could get by on 2-4 hours of sleep. And I used to get calls at home from him at 10:00 at night, 'Hey what about this, what about that...' He'd want to talk. But he kept that system going where a lesser man I think would've given up. It would've been sidetracked if they were to divert the resources onto the old 34 and then onto the 36. He kept it alive. Secondly, he kept it on track. It could've...changed the architecture to make things simpler. Try to use more old stuff off the shelf. He never let that happen. And he also, he was a smart enough man. He truly was the father of the System/38. He understood all the nuances of the hardware, all the nuances of the software, and he had it all in his head. And there wasn't anybody going to come into his office and try to, you know, slip something by him. As a result he had technical people kind of flock to him. He had some extremely brilliant people. Roger Taylor for example was with him. And Glenn was the only guy I knew that could talk to Roger and put Roger on the technical defensive.

Norberg: Do you know what the relationship was between him and Frank Soltis?

Schleicher: Between Glenn and Frank? I think Glenn thought of Frank as being a real good engineer. Who you could trust with keeping a group in the fold as opposed to running off in another direction. So I think he considered Frank a trusted technical designer, within the engineering profession.

Norberg: There seems to be some, I suppose I could call it disagreement as to who was more important in the development of System/38. It seems Frank would like to take a good portion of the credit. One of my staff members did an interview with Glenn about four years ago, before this project all together. We were working on software development at that point. And I thought that it was a fair treatment that parceled out the glories so far as you can in a large project at that time. Whereas Frank does none of that in his history of the development, or in the appendices, that we heard about earlier. There he makes it seem as if he had developed this concept here in Rochester before Henry arrived.

Schleicher: Came out of Poughkeepsie.

Norberg: The concept did?

Schleicher: Yes. It was all...what'd they call it? FS, Future Systems. Future systems was developed as a replacement for the 360 product line family. It was research driven. Started about 1973. Glenn became an advocate, maybe Frank became an advocate too, but there would be no System/38 without Glenn Henry. Guaranteed. And without System/38 there would be no AS/400.

Norberg: That's what Bains says too.

Schleicher: Absolutely yes, without him it wouldn't have happened. I wish I could...there are more Glenn Henry stories...

Norberg: Well but we can't tell those stories in this publication. Too many people whose feelings might be hurt or some other problem of that kind. (laughs).

Schleicher: I gotta just tell you one.

Norberg: Mona, are you trying to say something to us or do you just want to listen?

Schleicher: He would be working outside in his yard, and he'd always have his pants would be down low in back...(laughs). But Glenn was the nicest man. If you worked for Glenn, you were guaranteed that Glenn was your protector. All those years, 1977, 1978, 1979, 1980, when the corporation, the division was just extremely angry at us because we were delaying it, every year we delayed it a year. And it didn't ever look like we were ever going to produce anything, about in 1978, but I never felt threatened. I was a second line manager. If Glenn gets shot they're going to shoot me. I never felt threatened. Glenn never passed that level of anxiety on past him. Maybe he was so cocksure he didn't feel it either. I don't know. But I've worked in organizations where you're kind of forever looking over your shoulder wondering what sort of thing is going to come down the pike that's going to land right on you.

Norberg: Yes, that's how it was at Westinghouse.

Schleicher: Working for Glenn nobody ever felt that. I don't think...people felt bad when they couldn't put it in when we said it was going to, and we got criticism from our peers across the site, and the neighbor next door to me said, 'What the hell is the matter with you people? You're ruining everybody's reputation out here,' and on and on. But the neighbor on the other side is Glenn Henry and I never ever heard that from Glenn Henry. So there's no question in my mind as to who brought this very...this impossible architecture into fruition. It was Glenn Henry.

Norberg: Now, you were head of the programming center. And when you were asked to go east for the two years to work on the general architecture, when you came back what did you do? Did you return to be the head of the center? Or as something else?

Schleicher: I came back as vice president of development, the divisional vice-president had all...responsible for all the development. Which was the equivalent of lab director.

Norberg: It's the equivalent of? Why not *be* the laboratory director?

Schleicher: Well, I was the laboratory director. I'm saying that it is the position of the lab director, I was the lab director but I was this development vice president. When I went east, Tom Furey followed about maybe six months later. There'd always been this great desire on the part of the site general manager to have the laboratory working directly for

him. He didn't want a lab director in Rochester, you know, he wanted all the lab functions to report directly to the site general manager.

Norberg: Who was this who wanted that?

Schleicher: Larry Osterwise. So when Tom left, and I left, Steve Schwartz says, 'Larry you're a good guy, you got her.' So all of a sudden the lab ceased to exist as an entity. So there was no lab in 1989. Larry had it for two years. Larry wanted to manage the lab so he didn't have to put up with a lab director. But he didn't want to manage the problems of the laboratory. He thought he could treat it like manufacturing, where you have a manufacturing manager and that manufacturing manager basically ran manufacturing. Larry kind of provided oversight, budgets and all this other stuff. He thought he could do that in the lab. He had no clue of the amount of dependence there was on somebody to keep all these disparate portions of development in harmony with one another. First thing an engineer wants to do is have all the say. If you ask an engineer what the biggest problem of the system is, is it's going to run all that damn software. And if you ask a programmer what the biggest problem of the system is, 'the hardware never works until about three weeks before we're going to ship it.' So you've got all of these problems. So Larry consolidated all the 3rd line managers that previously reported to the lab director now reported to the site manager. As well as the manufacturing guys as well as the personnel guys, Larry had it all. Larry's number one goal in life was to win the Malcolm Baldrige Award. That was...his personal dream was to win this award for the corporation, for Rochester. Meanwhile, there had been a

study done on better ways to organize developers and manufacturing organizations based on the concept of red-blue-green management. And they tried to implement that across the site, but really they were focused on the laboratory because Larry, being a manufacturing guy thought the laboratory was running amuck and spent millions of dollars wastefully with too much structure, too much overhead and poor streamlining of profit. So here's Larry, he wants to win the Malcolm Baldrige Award, has everybody working for him, he doesn't want to manage beyond the 3rd line level, and he only wants to get involved in crises that require his personal seal of approval. But he also wants to win the Malcolm...he wants to put this whole management process in place because somebody told him it was a good idea. A guy told him it was a good idea, a guy named Jim Flynn. Based upon Japanese management techniques. It turned out to be an enormous, huge problem. People screaming and hollering. Estimates coming out of the development laboratory, out of sight. Larry turned into a pipe pass through. A good lab manager's job is to keep the division guys out of the laboratory, otherwise they were always down, they're taking notes here, this projects been added, that project's been added, and that project's been added.

Norberg: Like the US Congress.

Schleicher: Yes and all of a sudden, the place was just a mad house. So Mona and I are out east. Mona is working for Comdisco, so she's got a career started out there, and I'm working for Earl, and had no plans to come back to Rochester. All of a sudden, I get word from another guy, corporate executive, says, 'Dave, we want you to go back to

Rochester.’ I said, ‘Well, I’m not too sure I want to go back to Rochester.’ He says, ‘Dave, you got to go back to Rochester.’ And so I went to talk to people who I knew at the corporate level, and I said, ‘I don’t want to go back to Rochester, why the hell should I go back to Rochester?’ They basically said, ‘Dave, you’d better have a very, very good reason not to go back to Rochester’ (laughs) ‘You don’t have to go, but you’d better have a very good reason’. So Mona and I packed up and came back. And we reformed the laboratory.

Norberg: And was Osterwise gone by that time, or was he still there?

Schleicher: Nope, Larry was there for just a few months, probably about three months. I came back in June and the Malcolm Baldrige celebration was in May. We didn’t go to the Malcolm Baldrige celebration, did we?

Mona Schleicher: []

Schleicher: That was in May, well I was back by then. The Malcolm Baldrige award had just been awarded and there was a big celebration. Larry, that was his piece de résistance and he...they pulled him east to run corporate quality. And they replaced him with Bob Unterberger. So Bob and I were peers. He was on one side of the hall and I was on the other side.

Norberg: But by that time, the laboratory manager was a vice president, you said.

Schleicher: Yes, I was vice president of development, and Bob Unterberger was vice president of manufacturing.

Norberg: Now, was this the first time that was the case?

Schleicher: I think Larry might have been a vice president...I don't know...

Norberg: You see that's when Rochester News went out of publication, so we can't follow the stream of events after that. It stopped publishing in well, 1991. But they had stopped doing all of their management changes two years before that. And nobody keeps organizational charts that mean anything. They may have one in Armouk, but it's certainly not of use.

Schleicher: Yes, it was chaos. Projects were budgeted in terms of person years, how many person years does it take, a person year is worth about 100,000 dollars at that time. So I came back, I still had to get the lab back on track and give us a plan that's affordable and that you can deliver on. I came back and spent about two months going through the plans that had been committed out of the laboratory, and found out that we were like 300 person years over committed. That's 30 million dollars, that's serious money. And one of my first jobs was to go back east and say, 'This is not affordable, what you guys committed out of the laboratory is not affordable'. And meanwhile we're moving into a new house, and my boss had through his marketing manager heard that I was out there

taking the laboratory and running away with it (laughter). That I was basically abandoning the divisional direction and running it right down my own track. I got a call from Bob Labant, he says 'Dave' – his secretary says, 'Bob wants to see you on Tuesday.' I said, 'I don't think I can get there on Tuesday, I have this other commitment, I gotta...' 'David you don't understand, you are to be in Bob's at 11:00 on Tuesday.' All right, I fly out on Monday. Tuesday I go over to the headquarters building, and am supposed to be at a meeting with Bob at 11:00. So I cool my heels in the outer office until, Bob finally shows up around 12 or 12:30. He goes in his office. Still cooling my heels out here. I call Mona I said, 'Mona,' - we were going to have somebody come in, carpeting or we were going to do something, maybe the basement – 'put that on hold (laughs) because our Rochester stay may be at an end.' Which I wouldn't have cared, I'd just go back east. I had friends in high places, I wouldn't be fired. But Bob had apparently talked to some other people had found out that what he'd heard was untrue. So he calls me in, and everything was all sweetness and like, 'Dave! How long have you been sitting out there?' (laughs) So the good thing about that was that it cleared the air between me and Bob and from that point on he trusted me 100%. And I really enjoyed working for Bob. But unfortunately I worked for five different general managers in four years. Or divisional managers for four years. And so, I never got a chance to work very long with any particular one.

Norberg: Was one of these Don Stevenson?

Schleicher: No. I worked for Bob Labant, when I returned.

Norberg: Oh you said, division managers, my mistake. He was here in Rochester wasn't he? How often did you see him?

Schleicher: That was long, long ago before I reached the age of corporate wisdom.

Norberg: (laughs) It can't be that long ago! Ok, all right, that's a key element that you don't know. All right, have I missed anything that you think is important, questions that we've glossed over. Seems to me to be rather thorough, but you may be aware that I've missed something.

Schleicher: I think if you're going to bring it up to the current time, I think you should really understand what happened, basically when I left. 1994 I was ready for something new; they didn't force me out, but they didn't stand in my way to leave.

Norberg: To go to Austin?

Schleicher: To go to Austin. Because I think that, Steve Ladwig, who was my replacement, I think he was told at the time when he got the job that the handwriting was on the wall that they are going to do away with the autonomy of Rochester. Rochester up to that point in time...the division could decide they could do whatever they wanted, but the lab director or the site general manager basically said yes or no. We'd do it nicely, we would try to do it cooperatively. Very often, they would want to do something that

would be contrary to what we thought would be good for the Rochester site and we'd say, 'Oh, couldn't do it,' and we were sufficiently respected that it meant something. So I left and Steve came in place. Steve didn't have that reputation, he didn't have 30 years of experience, he had 15 maybe. And so first thing you know...and Bob Unterberger left. And they decided it would be a good idea...they didn't need a site general manager anymore. The site general manager position was eliminated.

Norberg: But doesn't that person have more responsibility now? It's called something or other...Minnesota...

Schleicher: There always was an IBM executive for the state. And that was always the site general manager, which is more of a lobbying role than anything. It wasn't something that anybody paid any attention to. But when the site general manager left and they didn't replace him with a site general manager, somebody became the IBM executive for the state, and I think that was Steve Ladwig. Steve Ladwig lasted about maybe another year or two after Unterberger left, but then Ladwig left and they didn't replace him.

Norberg: Who was running the shop?

Schleicher: They ran it all out of headquarters. So as soon as direct management of the site, both on product development and on a manufacturing standpoint were gone, it was run out of the division in New York. They decided then that wasn't working very well

either and so they sub divided the site so that now Poughkeepsie has a piece of Rochester and I think Austin, Texas, has a piece of Rochester. I don't know who all. I've heard that practically every division in IBM has a piece of Rochester reporting to them directly.

Norberg: That's a problem only for the people who are left here.

Schleicher: Well, what has happened since 1994 is that Rochester's previous autonomy vanished. Rochester now is more of a job site for people outside of Rochester.

Norberg: Do you think that might be consistent with the changes in the corporation to move more towards software and services and away from manufacturing, and therefore a unit like Rochester is not quite as important anymore.

Schleicher: A unit like Rochester probably can't exist in a corporation that wants to be driven by services. Because you got to put a hundred different faces on the place depending upon who you're talking to. And Rochester always had one face. It probably was inevitable. But it is a far different place than it was in 1996 even.

Norberg: So the fiftieth anniversary may be the anniversary of note.

Schleicher: I don't think Rochester will ever disappear as an IBM site. I think that's secure. It no longer exists as IBM-Rochester, that's disappeared. It's now, you know, this hunk belongs to this, this hunk belongs to that. But that's viewing it from the outside

because I'm clearly an outsider. I've been gone for 10 years, been gone from Rochester 12 years. But from everything that I can follow, people no longer consider themselves like they did in the 1980s, 1970s, you were a member of IBM-Rochester. (laughs)

Norberg: Now you're a member of some other unit that's doing a particular kind of work for the corporation.

Schleicher: Yes.

Norberg: That's very informative. Well thank you very much, it was very helpful.

Schleicher: Yes, it was fun.