Oral History of John William (Bill) Poduska Sr.

Interviewed by:
Gardner Hendrie

Recorded: October 9, 2002
Weston, Massachusetts

CHM Reference number: X2724.2004

Gardner Hendrie: I'd like to start with your very early history. Where were you born and brought up?

John William (Bill) Poduska Sr: <laughs> Well I was born and raised in Memphis, Tennessee, with my sisters and my mom and dad in a little house. My father was a radio and TV repairman, worked in a department store, and I got my interest in electronics from that. I was a ham radio operator at the time too.

Hendrie: When did you become a ham?

Poduska: Oh, I was-- I don't know. I was probably 16 at the time or 15 or something like that so it was probably 1953, something like that.

Hendrie: How many brothers and sisters did you have?

Poduska: Two sisters. I'm in the middle of the birth order. I am a middle child. My oldest sister is a physician. Her field is gerontology. She ran the department of gerontology for the University of Texas in Galveston. My younger sister is in the computer game also. She was a vice president at IBM, lives in Atlanta now. Now she's out on her own and enjoying the consulting role a little bit.

Hendrie: What were your first introductions to computers or electronics beyond your ham experience?

Poduska: Well, I came to MIT to go to school, obviously for my undergraduate degree, and I was an electrical engineer. I thought that communications was the field that I wanted to go into and got interested in continuum physics, turbulence, plasmas and fluid flows, and then solving some difficult problems in, I guess, about my senior year there. I got interested in using computers to solve some of these problems. Now, computers we called them. Of course, that was in the days of the IBM 650 and the 704, and we thought it was really hot stuff to have a card reader, you know, that could read a 100-card program and compute for 10 minutes or something. I think it was a case where I wrapped my arms around the tar baby and the tar baby wouldn't let loose of me and so I switched fields during that next few years of my studies more and more from continuum physics into computation. By the time I got my degree, my doctorate, I was teaching the computer courses at MIT, the famous 6251, 645, 647, 641, ran those courses for four years. I went on the faculty after that for four years.

Hendrie: What did you get your undergraduate?

<inaudible>

Poduska: All three, electrical engineering and computer science, bachelor's, master's and doctorate, all at MIT in seven years, which is fast, in fact too fast. I wouldn't recommend that for folks though. You don't do a lot of other things <laughs> while you're doing that.

Hendrie: When did you decide you wanted to become an engineer and go to MIT? When you were a kid, what did you think you wanted to do when you grew up?

Poduska: I always wanted to be in the electronics business. I always thought of a military career also. In fact, I was at West Point for a while before I went to MIT. I guess my earliest decisions about being in the electronics business, the communications business, were oh, I don't know, from the ninth grade on. Yes. Always fiddled around with stuff.

Hendrie: Tell me about thinking of going to West Point. What led you to first want to do that and then what went on?

Poduska: Well, <coughs> that's a sadder part of my life actually. I come from a military family so the military is an important part of my culture.

Hendrie: Was your father in the military?

Poduska: Yes, he was, in World War II. He was in the Coast Guard, as a matter of fact in the Mississippi, Caribbean, and Gulf of Mexico. My father was killed in a hunting accident in December of my senior year in high school, and we were a family of very modest means and I got an appointment to West Point and so I grabbed that very, very quickly. I did that in oh, say January--I don't know—January or February, a month or so after my father had died and, thinking things through more calmly and with less panic. I decided that MIT was a much better place for me to get an education than West Point. I went to West Point. I was there in Beast Barracks, as they call it, and I resigned and went to MIT, but it was no lack of love for West Point. That's a wonderful place, and I went and got my commission in the military and was a first lieutenant for several years and actually in that time period I thought again about making the regular army a career but then decided against it and came back and was on the faculty at MIT for four years.

Hendrie: You went to West Point and you did the first year there?

Poduska: No. Actually, I was only there for about three months.

Hendrie: Only there three months and then you transferred and obviously you'd applied to MIT and gotten in.

Poduska: No. That's another remarkable story. I was in bivouac at West Point <laughs> out on one of these marches that we had in the summer time, and I applied to MIT I believe it was the 13th of August and I matriculated on the 21st of September, and I thank God that I took what would now be called the SATs--at the time they were called the college board exams—and I was very relaxed, of course, in the process of taking them and I think I did pretty well--in fact I know I did pretty well—and I think that influenced the folks at MIT, and somehow or another I applied in August and I got in September. <laughs> A remarkable story, but I'm a religious person besides. I could put it in that context, but I think I

was certainly very fortunate to have gone through the trauma to the family, my father's death and my sisters in college too and going through that period and coming out on the other end nine months later as a freshman at MIT. It worked out well.

Hendrie: Tell me more about your experiences at MIT. Were there any professors there who were particularly influential in the sort of decision? You went to be an engineer, you weren't going to be in physics, you weren't going to be in computers. You were thinking about communication.

Poduska: Well, Yes. I was thinking about electrical engineering and focusing on communication. I mean there were a lot of people there that had a great influence on my life. Perhaps the one who had the greatest influence was Corby, Corbató, Fernando José Corbató who was arguably the inventor of time sharing with the system called CTSS at MIT on I think a 709, IBM 709, if you can believe it, you know, the early vacuum-tube machines. He introduced me to the field of computation and teaching with the course 6251 and 645 and 641. He was an author. He was the principal behind CTSS. He was assigned as I think the PI on Project MAC [the MIT Project on Mathematics and Computation], when it was first instituted out of DARPA although Bob Fano and J.C.R. Lickliter were the project directors. I think Corby— I might have this garbled a little bit, but I think he was the principal investigator on the project. He was obviously a luminary, a touring award winner, and just a great guy. He lives right here north of Boston. He probably had the most influence on me. In the earlier years, I think people like Bill Jackson and Herb Woodson. Herb Woodson—I called him Woody—went on to become dean of engineering at the University of Texas at Austin Another person who had a substantial influence on me was Paul Gray, who was, of course, later the president and then the chairman at MIT. As it turns out, Paul interrupted his education to go into the army for a couple of years after his bachelor's and then came back to get his doctorate. When he came back, I was a senior at the time and we were both working in Woodson and White's Laboratory. I was assigned to him as a technician. I was assigned to Paul Gray as a technician to do his grunt work while he was doing <laughs> his thesis and I got to know Paul pretty during that process. Later on, he lived in the town of Winchester where we lived and our children went to school together. He had a long-term sort of continuing influence on me and my career throughout that whole period.

Hendrie: You said that you did go in the army for a little bit. Where was that in the progression of things?

Poduska: After I got my doctorate. I went all the way and got my doctorate, went through ROTC at MIT, and was commissioned into the army. I was assigned at Fort Monmouth in a department known as the surveillance department, one of these innocuous names which has to do with listening to significant events at substantial distances away, and you could <laughs> put a lot of spin on that. Of course, it was many years ago and technology has well overtaken the kind of stuff that was going on, but it was very interesting to see what you could find out by simply listening, that is to say having microphones out in fields and listening to things that happen at long distances away. It is extremely interesting.

Hendrie: Was there computation involved?

Poduska: You bet.

Hendrie: You would do computer processing.

Poduska: Pick the signal out of the noise and find the direction and magnitude.

Hendrie: While you were at Fort Monmouth, was that your first contact with computer control?

Poduska: Yes, it was. Very good memory

Hendrie: Tell us the story of how that happened.

Poduska: Well, we needed to get a computer to essentially pick the signal out of the noise and these were acoustic signals that we needed to work with 3C [Computer Control Company, Inc] had just come up with a computer called the DDP-24. I'm sure you recall that machine. Its competitor was an 18-byte machine from DEC and I'm trying to remember the nomenclature of that one.

Hendrie: Probably a PDP-1.

Poduska: It seems like to me it was a PDP-4 or something beyond that--

Hendrie: Could be a PDP-4 also.

Poduska: --but my memory is vague on that, but anyway we did a little bit of a bake-off, a little bit of a competition there, and the word size turned out to be important both in terms of addressability and accuracy of computation and so we went with the DDP-24. From that association I got to know the folks at 3C who were at Framingham, of course, at the time. When I came back to MIT and was on the faculty there, I took on a consulting assignment with 3C where we built the [DDP] 124. I think the [DDP] 224 had already been built if my memory serves me correctly. So I built the [DDP] 124 [garbled] and that's, of course, where I met you in talking about the [DDP] 116.

Hendrie: I remember hearing that you had come to the attention of the people at Computer Control because you had not bought any index registers with the DDP-24, but then somehow they discovered that you had them in your machine. Can you tell that story?

Poduska: Well, yes. <laughs> I had analyzed the machine a little. You know, to people listening to this, this must sound very strange because in those days an index register was not just a small, little area of a chip. It was a substantial number of plug-in modules which [each] might have been, say, 4 by 6 inches in size and there might have been 10 of them or 12 of them involved in this. So we're talking about that there was a substantial amount of hardware, and I figured out that to get the index registers in the machine, all you had to do was plug in the modules. So we bought the machine, then we bought the modules, and we plugged the modules in and the index registers worked, and pretty sure that was what was going to happen. The fall-back, of course, was if it didn't work we could always go out and buy them,

you know, and have 3C put them in. Part of this is understanding the procurement mechanism of the government also. I guess we've all played this game a little bit before, but when you're having an open-source procurement or competitive procurement, you have to evaluate all the things that you need to do and you have rather a fixed budget for doing so and budgetary considerations matter, but once you have a computer installed, expansions to that computer, whether it's adding memory, adding disks, adding index registers, that's a sole-source procurement because obviously you can't buy index register modules from Digital to plug into a 3Cmachine. So I mean a little bit of that is gamesmanship about just exactly how do you play the system to get what you need to do the research.

Hendrie: If my memory serves me correctly, you saved $10,000 once, what they charged for each index register.

Poduska: We did. We did.

Hendrie: How long were at the signal corps? How long did you work there or stay in the army?

Poduska: Well, there was a period of four years when I was on the faculty at MIT as an assistant professor and I was in the army. The actual active-duty time was probably 18 months out of that, but it was back and forth a little bit because I would be on assignment to Fort Monmouth and then back in the Boston area for a variety of reasons, part of it on the academic cycle reason for preparing coursework and part of it because of some of the needs of the military for the assignments that they had me on Basically out of that four years, out of that 48 months, about 18 of those months were actually active duty in the army.

Hendrie: The army was perfectly to happy to have you work between—

Poduska: Oh, delighted, delighted, because—

Hendrie: You could always just resign. <laughs>

Poduska: Well, no, no. I had a two-year stint, but what it meant was that they could put me on the assignments that I needed to go on at their timing. I'm being vague here because I don't have any idea what the classification of these issues is anymore, but the point is that the timing of the army's needs was not necessarily day by day. They needed to send me to different places to do things and they were happy to spot that at the time. We had some convenient arrangements with the commanders at Fort Monmouth. I mean it was, you know, it worked out well for everybody.

Hendrie: While you were at MIT in your teaching portion of your career, were you doing any research or did you—

Poduska: Yes. That was the early days of Project MAC, and we were using—for those of you who have really long memories—it was a combination of people at Bell Telephone Laboratories, General Electric's

computer operation mostly at Phoenix, and Project MAC at MIT. Project MAC now has become LCS or has morphed into Lab for Computer Science. The machine we were using was a [GE] 635, I think, and we at Project MAC designed special hardware mostly for segmentation purposes and virtual-memory purposes that GE built for us. That meant we had to build special compilers and assemblers for the [GE] 645 and design the operating system around it. I was in the operating system group, but I also worked on the operating system, semaphores and various collectives and such things. But perhaps my longest-lasting contribution was I wrote the assembler for the [GE] 645. I again remember assemblers really get a machine code from sort of vaguely symbolic form into the bits and bytes that go into the machine. I wrote the assembler for it with all the screwball addressing that it had. You might be interested to know that I wrote the assembler in Fortran. Why Fortran? Because I was determined to do it in a higher-level language. I just didn't have the energy to do it any other way, and Fortran was the only higher-level language available on the [GE] 635. It turns out it's an effective way of doing things. Fortran has its warts and pimples, but it also has enough higher-level language features to make it worthwhile, composition, loop control, conditional control, common areas, symbolic names. It also has numerical statement labels and gotos and other annoying things. I wrote this—

Hendrie: That weren't too helpful, yes. <laughs>

Poduska: Well, you know. It was the best tool we had, but I wrote the assembler. <laughs>

Hendrie: Tell me more about your career while you're at MIT and teaching. Any other stories or things that happened there that come to mind?

Poduska: Well, I think that was the time when basic computer-science teaching was beginning to explode. Now, what we called computer science at the time I think we would now think of mostly as programming and programming skills. The courses we taught, while they were not at the electronic level—I mean we didn't teach about transistors and such things as that—still there were basic programming-level skills. How to use the machine, how to write programs, how to do effective things that sound so simple now, loop control, conditional statements and such. We used Fortran a lot, but we also used a higher-level language called MAD, the Michigan Algorithm Decoder, <laughs> which was a very interesting language somewhere short of ALGOL and Pascal but north of Fortran in its capabilities. The point I'm getting at here is that the enrollment was burgeoning, that we would fill all of [lecture hall] 10250 at MIT for all semesters for the lecture part of the course

Hendrie: How many people is that roughly?

Poduska: You know, the lecture hall holds maybe 400 people or something like that, 400 to 600 people, and that was the lecture portion and there would be 10, 12 different sections. I guess my first introduction to organizational skills in fact, managerial skills at that level, came from just running those courses. I mean we had tons of people to deal with and the problem sets, and at that time scheduling computer time for people to run their programs which was, in that day with the limited resources, really quite a difficult task. I mean even things now like scheduling the punch card machines, you know, the old 26s for punching cards. I mean this sounds so Neanderthal, but that was the problem. We didn't have all that many card-punch machines at MIT so that you could handle hundreds, nearly thousands, of people doing problem sets and punching cards and jamming the machines and doing the things that students do.

Hendrie: What did the students run their test problems on?

Poduska: They ran them on the available machines which, over that period of time, progressed, oh, I forget exactly when, but the 709, the 7094, the 7094-2 I think it was. We began at the end of that four-year period to have people using time sharing with the terminals that were around, and that was CTSS, which stood for the compatible time-sharing system. I'm not sure what it was compatible with, but that's what we called it<laughs> So, you know, I guess the thing I remember the most was not how profound the courses were. They might have been in their time frame. They were certainly advanced in terms of what was available to the people there, but the magnitude of the impact that it was having on the student body and on the faculty and just on management of those courses and, of course, now ECS at MIT is I think 40% of the engineering school and continues to grow and I have to keep the lid on it to keep things sort of in order.

Hendrie: Do you give some of the lectures or do you have guest lectures?

Poduska: I gave all the lectures. Yes. I was the lecturer. Yes. Well, I gave all the lectures of whatever course I happened to, 625 one mostly, but also the freshman courses, which were 641 and 645 for MIT—

Hendrie: Did you prepare the syllabus?

Poduska: Oh, God, yes.

Hendrie: Did the whole thing.

Poduska: Yes. Yes. Yes.

Hendrie: Did you enjoy the teaching part?

Poduska: Oh, loved it. Loved it. Loved it.

Hendrie: When did you decide and what went through your mind when you decided that maybe you'd do something else?

Poduska: Well, there were a number of things. Perhaps the most important, and this one is geographical things almost, Project MAC was in building 645 of Tech Square. Interestingly, the building was 645 and the machine was named 645 and there were three big buildings around there, and it turns out that NASA had just built a research center in one of the Tech Square buildings, and it was just populating it and one of them was called the Electronics Research Center It was the major laboratory and it was populating the computer research laboratory as part of ERC, and they asked me to join that and run a branch of that laboratory called the manned computer systems laboratory.

Hendrie: Was this associated with MIT at all or—

Poduska: Well, yes, in the sense that there were a lot of faculty involvement with ERC and the guidance laboratory, which is now DRACO Laboratory, was very closely associated with ERC. This was in the Apollo era now. This was 1966 through 1970. Apollo 12 had not flown. Kennedy's death had occurred in 1962, I think, something like that. So after he had made his commitment to the Apollo program landing on the moon and two years later from that, when things were getting a little dicey., The programs were well behind, the ERC was set up and they asked me to join and take a look at it, and our mission for that laboratory was to look at the manned computer interactions for future manned missions including the three we were focused on was the earth orbiting—this was in 1966 now—the earth orbiting laboratory, the lunar base station, and the manned Mars mission, and of those only their laboratory and its successor projects, had been ones that have come into fruition. We had a dandy time with all sorts of graphics and tactile interfaces, voice recognition, voice output, color graphics. It was a very interesting and a very well-funded program. The Apollo program is one of those seminal points in the history of man with the history of computation, the history of folks around at MIT who were in that world, and for a guy like me whose interests really are aviation and computation, it was really pretty right in the cross hairs. So shouldn't be too surprised to hear that one of the companies that later was founded was called Apollo Computer and there is a very good nexus, a very good connection, a very close connection, between the naming of that company and the experiences at NASA's Electronics Research Center.

Hendrie: When you were at the research center, my understanding is that you did some work on the DDP-516 to turn it into a time-sharing machine. Could you tell us about that?

Poduska: <laughs> Yes. That was very interesting. This was still in the days when machines were made with plug-in modules as compared with highly integrated circuits. As a matter of fact, we didn't even use low levels of integrated circuits, I mean like the 7400 series. The things were made out of transistors and diodes and small, little packages. We bought the 516, which was the successor of the machine that you designed, the 116, got into the guts of it and added a paging mechanism to it. We used paging as the principal memory-management mechanism of the machine. It had some interesting characteristics in that the virtual address space was addressable with the 16-bit number—it was a 65k—but the physical address space was an 18-bit number of 256k. In other words, we had a physical address space larger than the virtual address space. Of course, it's just the opposite now as it should be. We were able to put the paging in there and to turn it into a time-sharing machine. We built an operating system on it called, the first single-user system, called DOSS—what else-a disk-operating system, and then later an operating system that was time shared among seven users--TS DOSS time shared DOSS—based in large measure on the Multics operating system that came out Project MAC. Now, it was very simplified Multics It did not have all the segmentation features or all of the gateways in protection backing this, but it certainly had the notion of coupling the processes and the use of virtual memory to carefully manage the resources of the machine among the users. It was Quite successful and it was TS DOSS that we used to build the first operating system that Prime put together, which was at the time called Primos, prime operating system, but it was for all intents and purposes a copy of the TS DOSS system that was built at NASA.

Hendrie: Did the TS DOSS system have concepts like privilege mode and all of those kinds of things?

Poduska: Oh, absolutely. A privilege mode for the operating system, specialized gateway calls between the user code and the operating system, and sufficient protection so that there was no way to crash the operating system from the user space in any ordinary kind of way. There are always wormhole kinds of things that you can do, but there was no way to sort of override the code, and there was protected virtual address spaces, protected procedure calls, argument transfers that were protected and in a way it certainly was not perfect, but it was, as far as memory management mechanisms can protect against invasion, it was a very secure system.

Hendrie: Was this written from scratch?

Poduska: Yes.

Hendrie: Didn't use any of the code that came with the machine.

Poduska: Well, we used the code that came with the machine for the assemblers and compilers and such and the other ancillary things that it had, tape drivers, assigned square-root routines and things like that, but the basic operating system and in fact the things beyond that like text editors and graphics system were built at NASA.

Hendrie: What was your function in building that operating system? Were you leading the group or—

Poduska: I was leading the group.

<inaudible>

Poduska: We had a lot of those smart young Turks, yes, many of whom have gone on to do, you know, other interesting, rewarding kinds of stuff and a lot of students there from MIT and Northeastern and Harvard, so those were good times.

Hendrie: How long did you stay there?

Poduska: Four years, 1966 through 1970. At the end of 1970-- Well, how do I say it? President Nixon came into office and the electronics research center in Cambridge, Massachusetts, was shut down, <laughs

Hendrie: Its political support disappeared.

Poduska: Well, its political support had evaporated but it was shut down. It then became, through the wonderful work of Jim Elms, and I don't know he managed it but he got the Department of Transportation to take it over as a research center and it remained in existence as a DOT center. They called it TSC,

Technology Systems Center I think it was, in Cambridge and it's still there to this day. I don't know what status the laboratory is at, but it carried over a lot of things. A lot of the graphical computer work was transferred from the space domain into the aeronautics domain, working with air traffic control things, especially en-route traffic control systems, noise abatement systems and such, but that was well after I had left.

Hendrie: Did you leave when NASA—

Poduska: I left when it was closed down because at that same time in Tech Square Honeywell was setting up a research center. It set up a research laboratory, a computer research laboratory in Tech Square at 575 Tech Square. I was a member of the research team for a while, and then when the director left I became director of the laboratory there in Cambridge.

Hendrie: What was that laboratory set up to do or what did it do?

Poduska: It was set up to basic fundamental research in computer science for the benefit of Honeywell. Its computer operations, but also for the rest of its scientific operations. Honeywell had a scientific research laboratory in Minneapolis and a semi conductor research laboratory which was in Minnesota somewhere. I'm not quite sure where. Honeywell had a long-term vision that computation was going to be important in its controls operation, and so it wanted to set up some fundamental research in computer science looking for future applications in control systems. Now, this was before the Honeywell-GE merger which happened in 1972, I believe somewhere in that time frame. So this was before, and it was interesting about how these paths cross again, but the group in GE, who had been doing the 645 work then in 1972 became part of the Honeywell operation. Part of this whole merger and meltdown of what we used to call IBM and deBUNCH, you know, Digital, Burroughs, Univac, NCR, CDC, and Honeywell, deBUNCH, Yes.

<Break>

Hendrie: You were mentioning Larry Roberts and his involvement in Project MAC and that he was a classmate of yours.

Poduska: Right. He was a classmate. He was very much involved in graphics things with I think sketchpad, he and the Sutherland brothers, Ivan and Bill Sutherland. He was at ARPA at the time when Project MAC first got kicked off the major funding for it, and he was very much involved in that from the get-go. I haven't tracked him in the last few years so I don't know what he's been up to, but—

Hendrie: I think he's involved in a start-up out on the west coast, but I don't remember exactly what the—

<inaudible>

Hendrie: I can't remember exactly what he's doing. You've joined the Honeywell Research Center and what year are we at?

Poduska: That was about 1970, and then in 1972 there was the merger between GE and Honeywell and lot of interesting things going on.

Hendrie: What sorts of things did you do at the research center? What were you working on?

Poduska: We continued the work on graphics and graphics processing as well as other fundamentals of operating-system work, you know, the issue of memory management, the issue of multithreading processes and such. Ugo Galiardi, another name that you may remember, was the director of the laboratory just before me. He had come from Harvard. He was a faculty member at Harvard so we had a lot of Harvard students there including people like Harold Schenk, Jeff Buzen and Ron Goldberg, who then went on to form BGS. In that period there was a lot of, oh, just an awful lot of activity integrating the Honeywell and the GE stuff together, which Ugo went off to do, and so I ran the laboratory and tried to coordinate those activities, but still it was a very young laboratory. It was only 18 months old or so when we were doing that.

Hendrie: About how many people were there at that time just roughly?

Poduska: Maybe 30, something like that.

Hendrie: This was not presumably so much theory. Did you try to reduce things to practice?

Poduska: No. You know, laboratories have all sorts of ranges of activities. Some are very applied to building the next product to the most theoretical of what the science is doing. This laboratory was intended to be somewhere in the middle, to have applicability of ideas and thoughts but not necessarily reduction to a product, which was left to the development organizations inside of Honeywell. At least that was the theory. The laboratory was only 18 months, maybe 20 months, old before I left, and I don't think it had the opportunity to really imbed itself in the culture of Honeywell, especially with the merger of GE and Honeywell coming on board and the research laboratories of GE now getting melted into the activities of Honeywell. It was also at about that time that a mutual friend of ours, Bob Baron gave me a call one Friday afternoon and I was not associated with framing what was FCO, the successor organization to 3C at the time, but he called me up and asked me if I'd meet him at his house for a drink on a Friday afternoon. He said they were forming a new company and wanted to know if I was interested in running the engineering organization of that company. They were under a very tight time constraint and they were meeting the next day, which was Saturday, and he needed a decision from me on Saturday. <laughs> I went home and talked with my wife and I met with them on Saturday, and I threw my hat in the ring and that's how I became badge 006 at Prime Computer, and it was literally a 24-hour kind of turnaround. <laughs>

Hendrie: Had he approached venture capitalists at that time? <inaudible> Where was he in this process?

Poduska: He had, he had not approached them**.** He was in the process of approaching them and--

Hendrie: He had a business plan?

***Poduska:** He had a business plan and the business plan was, interestingly, to build a machine which, to put this in context, you may recall that there was a lot of activity in those days to make machines which were hardware compatible with other manufacturers like Amdahl versus IBM and Magnuson. The Honeywell 1200, I think it was, was compatible or at least semi compatible with the IBM 1400 series, at least code translatable. You could make a code move over. The idea behind Prime Computer was to make a machine out of modern semiconductor technology. <[quotes with fingers> This meant the 7400, the schkotty level of integration, that was compatible with the series 16 machines out of Honeywell. I mean it seems so archaic now to think back on it, but the big invention of the day was called the SBC, the single-board computer, a computer that would fit on only one printed circuit board. Of course, you know, <laughs> the size of the die now that's used in machines is tiny. That was the basic business concept and the idea was this. If we were hardware compatible with the Honeywell series, we had all of the installed base to go after at higher performance and lower cost, and furthermore we had all the software that was developed at NASA that ran on the 516 to put on this machine, and further if we put the paging mechanism, the memory-management mechanism, into the Prime machines, then we had all of the advance software, that is to say the time-sharing system, as well as the other graphics systems from the NASA engagement to put on the machine, and all of this stuff was in the public domain. So it was a very opportunistic and I think well-crafted business plan and it worked really quite well.

Hendrie: When Bob was doing his plan before he had talked to you, did he have this concept of using the NASA stuff in the public domain or did you sort of bring that to the party?

Poduska: Well, I think both of those statements are true. I think Bob understood that there were many customers of Honeywell who had code that could be converted because He ran engineering at Honeywell for FCO, and he knew full well that there were many customers who had substantial amounts of code that could be converted very quickly to salable products by a new company and that among these were the stuff at NASA but there was also some real-time stuff that had been done at I think Foxborough if my memory is serving me correctly. So he had this overall vision. I think when I came on board, it validated the prospect of getting the NASA stuff and certainly greased it so that we could get that software converted over very simply. As we all know, its one thing to have the idea and it's another thing to have the capability to do it. I had the background and the skills to do it. I'm not unique. There were plenty of other people who did too, but at least I had that background and skill to bring that across and that turned out to be an important part of the Prime success story.

Hendrie: You joined Prime.

Poduska: Ran engineering.

Hendrie: Tell me about the startup days in Prime. Tell me some of the war stories.

Poduska: Oh, there's so many of those kinds of stories and so many wonderful stories, you know, of going between a product planning meeting where we would be thinking about everything from what is the instructions set to what is the software complement to the color of the machine to painting the walls and installing phones and different things and we started up a <inaudible> over on Strathmore Road, all these tiny, little buildings, polluted buildings, had been used for making wire memories, you know, the <inaudible> the plated-wire memories, and there was a lot of chemicals lying around, wonderful stuff, and, you know, all of those things of getting venture capital. There was one hilarious story about getting venture capital. Dave Dunn and Ross Roblyn had just split out of Whitney and were forming a new venture fund with some of the Bass family of Fort Worth's money. They called the company Idanta Partners and we had heard of them, I think, through just a magazine article or something. So another of our friends, Sid Halligan doesn't know the phone number for them, calls directory assistance in New York to get the phone number of IDATA He gets a phone number, he calls them up, and Sid goes through his explanation of what's going on and finally somebody on the other end says sir, I think you have the wrong number, this company is called IDANT, without the TA, and we are a sperm bank <laughs> and he said Yes, I probably got the wrong <laughs> number then, probably not what we had in mind, <laughs> but we did get in touch with Idanta. They became the first funders of the company—

Hendrie: They were the first ones to sign up.

Poduska: Yes. They were the principal ones then along with Smith Barney, a venture arm of Smith Barney with Jack Delaney, and they were the guiding elements in the investor part of the community.

Hendrie: Do you remember how much you raised initially?

Poduska: Yes. I think, the first round of financing was $900,000 or something, and it lasted us for really quite some time.

Hendrie: Salaries were a lot less then, weren't they?

<inaudible>

Hendrie: Tell me. Do you remember how much of the company that you had to give up to get your $900,000?

Poduska: I think we gave up a little over half, Yes, the 55%

Hendrie: The classic amount.

Poduska: Yes. It was not a terrible amount, but it was certainly far more than say companies were giving up in the mid 90s in the dot com boom and you could get funding at 30% of the company so—

Hendrie: You started development. What were the initial products of the company?

Poduska: Well, the initial products were the Prime 200 and the Prime 300 machines. They were Honeywell compatible, relatively small, that is to say they were about the size of a large bread box, a very large bread box, but they would fit on top of a table, you know, this coffee table. There was an interesting point, though, that I think Dave Nelson used to talk about computer pricing and being more closely correlated to pounds than to capabilities, and pounds of sand was his sort of metric because one of the earlier machines we sold, the Prime 100 I think, you could buy for about $4,000. Now, $4,000 is not wildly out of line with the kind of prices that people are paying today for about the same size box. I mean a server box that fits in a rack of this size you can get for about $4,000. The capability level is vastly different, but still the price per pound is pretty constant.

Hendrie: You had a line of machines.

Poduska: We had a line of machines from the low-end OEM machines up into the highest end of the time-sharing machines that we used.

Hendrie: They all have the same instruction set?

Poduska: They had the same instruction set with some restrictions. Again this is like the index-register story, you know. I mean the multiply-and-divide instructions were options that we had on the machine and we got money for installing. Now, it turns out they had a factory cost of zero because it was micro program machine and you could either include it in the micro code set or not but, of course, like all things, we got in the pricing to market very quickly because other companies would put them in for free and we had to follow suit. The virtual-memory aspect was the part that really distinguished the machine at the high end, the Prime 300. Now this, of course was back in 1972, 1973, I think--yes, about that time frame—when we put out a time-sharing machine A commercially available time-sharing machine that used paging as a memory-management mechanism and separate virtual-address spaces. And a healthy seven-user system could be bought for maybe 50k. That was pretty powerful and what was your term you used to use, a self-selling humdinger? That's what they became and they sold very handsomely into the technical world, but then they began to sell very much into the commercial world and that's where Ken Fisher came in to run the company and Ken, who understood the commercial world, caused us to understand, sometimes kicking and screaming, you know, <laughs> that we could sell an awful lot more machines into the commercial world. By commercial world, I mean trust management, banks, commerce houses and such. They would pay higher prices. They needed other things like data base management, RPG/COBOL, which we provided, and that was the success story of Prime running it up for all issues.

Hendrie: Maybe you could talk about when you rolled out the first machines. I assume the Prime 300 took a little bit longer to get out in the marketplace.

Poduska: Actually, you know, they didn't. We turned those machines around in under a year. I think it was only nine months or so; in fact I know it was.

Hendrie: When did you start? February?

Poduska: February of 1972 I think it was, and we rolled out the first machines in October or November, and we were in full production, by the end of the year. Those were the guidelines and that was part of what the basic business plan was that we were going to design and build to get product out on the street versus to additional functionality or any other parameter which might have been say designed to fit a particular customer's need or things like that. In other words, time to market was the essential ingredient of what we were doing and that's what we did. It made some technical decisions for us. For example, micro programming in that time was a faster mechanism for getting machines to the marketplace because you had the luxury of people doing the micro programming on one side while the other people were doing the hardware design and they could converge later on. On the other hand, it's arguably slightly more expensive and slightly slower even in that time frame.

Hendrie: What did you use for a micro code store on that machine?

Poduska: We used fusible linked prompts, you know. They had just come out and we were prepared to use most anything at that time. I mean just reflecting on it sounds so Neanderthal now. IBM for its micro code store was using a form of punched card that was actually electrically edged, but they were actually using cards for pieces of paper, rectangular pieces of paper, to store the micro code for some of their machines like the 360 Model 30.

Hendrie: There was also wire rope technology—

Poduska: There was wire rope technology. They used everything they could, of course, but—

Hendrie: You ended up just using—

Poduska: We used the prompts because they just miraculously popped up for us and we had people hand programming the prompts, that is to say actually in there with a prompt burner pushing the <coughs> buttons almost like on a calculator for programming them location by location. Yes. That's what you do when you have to.

Hendrie: Who were the first customers for the machines?

Poduska: The first customers for the machines were in the Cambridge area, Adams Associates I think. Interestingly, our first customers were commercial customers and they were using the machines for-- I believe the first machines were used for constructing and maintaining financial advice to some of the customers. Not so much buying and selling stocks but financial evaluations of companies and things of that nature. To tell you the truth, it's a little vague in my memory but it was on that order. This is Charles W. Adams.

Hendrie: Yes. They published a very famous little guide to all the computers that—

Poduska: Right. Exactly. Exactly. This was associated with that in some way.

Hendrie: Part of their consulting business—

Poduska: Yes, and the exact details escape me.

Hendrie: Did they use a 300? Who was the first user of the 300?

Poduska: I think they were the first users of the thing, but very quickly after that there was a company called SCI in Philadelphia who was using the machines for bank trust management and for entering data into the trust-management software.

Hendrie: So they wrote their own software.

Poduska: Right. Right. They wrote their own trust-management software which they then deployed on the 300s, and they liked the idea of multiple terminals and the speed that they could get out of it. I think SCI still exists in some form or another as a matter of fact.

Hendrie: Just as an aside, we actually talked to the founder of SCI at Stratus as a potential investor.

Poduska: Was this Katz, Steve Katz, or Al West?

Hendrie: Katz, I think.

Poduska: Katz is a piece of work. He's an interesting guy. Yes. <laughs>

Hendrie: Paths do cross.

Poduska: Yes, they do, especially in this world.

Hendrie: So how did you – at some point the company had some sort of difficulties. I know that Bob left.

Poduska: Right.

Hendrie: And it wasn't, I don't think he was retiring per se.

Poduska: Right.

Hendrie: You know it wasn't a totally voluntary sort of thing. Tell me about what happened to the company and sort of where it got to.

Poduska: Well, I think, you know, I think many of the people who chat with you and many people who see this will be part of the venture world and from all sides, from participants in venture companies to investors to, perhaps, external board members. And, what we all observed in those is that very often, especially with successful companies, the needs of the company for capabilities of its managers and founders outstrips their ability to grow and I think that is the short explanation for Bob Baron but I think it's also the long explanation.

Hendrie: Yes.

Poduska: I mean there are clearly details and such but we became public very quickly and with a very successful public offering came into the limelight. The company was doing really quite well. It had a couple of bumps in the road like all companies do but most of these were really span of control kinds of issues with the top management team. I think at the time Bob was certainly disappointed but maybe more than that, maybe bitter about it a little bit. But I think in retrospect with a couple of decades of retrospection and introspection it was nothing more complicated than the fact that the company's needs for its managers because the company now is a 50, 80, $100 million sales range, you know, in the mid '70s when that meant something.

Hendrie: Yes, exactly.

Poduska: And Bob's career had been as a guy who had run an engineering department at Honeywell and the needs of the company outpaced the skill set.

Hendrie: Yes

.

Poduska: And, you know, that doesn't happen overnight. That's something that happens gradually and things begin to fall apart. A confidence level between management layers tends to decrease and between the CEO and the board tend to decrease and then there's usually some cathartic event. I don't even know what it was in this case. I recall there was something. It was either a missed quarter or a botched analyst meeting or something of that nature and made the decision to move on.

Hendrie: Yes.

Poduska: But it's really –

Hendrie: It's one of those things that just happens in a high gross company.

Poduska: They happen. I think they are the rule rather than the exception.

Hendrie: Yes.

Poduska: And perhaps those things shouldn't come as a surprise to us but they always seem to.

Hendrie: Yes.

Poduska: On both sides. I think many times we who are investors let things last too long without really preparing ourselves for it and without doing the things that we ought to do. We should send our managers to charm school so that they learn some of the management skills. We should send them to A&P programs. We should support them from beneath with lieutenants who can help them.

Hendrie: Yes.

Poduska: And we should prepare them for migration and transition.

Hendrie: Yes, okay.

Poduska: And this is not just being nice guys. This is benefiting the company.

Hendrie: Yes.

Poduska: Because it's very cathartic for a company to undergo leadership change.

Hendrie: Yes, you're right and it happens and it has to happen.

Poduska: Right, and of course that was not the last leadership change for Prime.

Hendrie: Yes, exactly.

Poduska: Ken came in and then Jim Henson came in and then Tony Craig came in after him and then after the computer vision merger. Jack Shields was there and then Russ Planitzer was there and then it was purchased by PTC. So, I mean, that was not the last event.

Hendrie: Exactly in its leadership.

Poduska: Right and, you know, none of this should be strange or a mystery to us. This is the way the world is. This is the way leadership is required. We don't blink twice when these things happen in professional athletics. We expect it. We should expect it in business also.

Hendrie: Yes, you're right. I think certainly and it does happen. What, you know, at Prime what were the, you know, what would you say were some of the major accomplishments on the technical side or innovations at Prime? Obviously it had the first virtual memory mini computer.

Poduska: Right.

Hendrie: And I'm not sure whether there were any competitors.

Poduska: There were none at the time.

Hendrie: I don't think [Data General had come up with anything.

Poduska: : No. No The closest at the time was, oh, the Sigma series of machines from –

Hendrie: Ah, Sigma 5 maybe.

Poduska: Yes, Sigma 5 and Sigma 7 from –

Hendrie: From SDS.

Poduska: SDS. I wanted to say SDC but I knew that wasn't right.

Hendrie: Yes, SDS, right.

Poduska: Max Palevsky at the company.

Hendrie: Yes.

Poduska: And they were good but we were there first.

Hendrie: Yes.

Poduska: And we were there with the lowest cost form of memory management also, that is to say an embedded paging system which could get along quite nicely with almost any kind of mass storage. I mean now again, to refresh memories, there was the era where people worried about whether drum storage or disk storage or of how many heads to a disk, whether it should be those big washer machine style disks. Lots of people worrying about issues about thrashing of memory back and forth because of paging requirements of machines. But we had a simple mechanism and a cheap mechanism as it turned out and we were able to ride the silicon technology curve on all aspects. And, I guess I should mention probably one of the most important things that nearly escaped me. We were the first company to deploy commercially, semiconductor memory as the only memory system of the machine. Prime has never had core memory, never had core memory in any of its machines. That was an early decision point and we went with it. We had the original, Dave House will love this because he was such a big part of it.

Hendrie: Yes.

Poduska: We had the original, I think it was the 1103 memory parts.

Hendrie: Yes.

Poduska: The 1,024 bit parts.

Hendrie: Yes.

Poduska: And then we thought it was – we thought that heaven had arrived here on earth when we went to the four kilobit memory parts, you know, and so we were able to ride the silicon technology curve for the processor elements and the speed improvements that came there.

Hendrie: Yes.

Poduska: We were able to ride the semiconductor memory elements to get more physical memory into the machine for the requirements; therefore, able to go from seven users to 15 users to 31. The Prime system had the quirk of having a number of users that was two to the nth minus one, so it was either seven or 15 or 31 or 63.

Hendrie: Okay.

Poduska: As it went up. It was a quirk.

Hendrie: Yes.

Poduska: We were able to do that and go up to a larger number of users very quickly with more performance and in the same size box. A second thing that we did –

Hendrie: So even the Prime 100 had semiconductors.

Poduska: Always had semiconductor memory, right.

Hendrie: Okay.

Poduska: Right.

Hendrie: Yes.

Poduska: And even the OEM machine. That was a problem with the OEM machines because the question was how did you deal with power failures? And we had a battery backup in the system for dealing with power failure so the memory could keep getting refreshed.

Hendrie: Okay.

Poduska: Another part of that system was being single board computer kinds of system, these large, you know; say 20-inch square boards more or less. We were able to use the same back plane and the same peripherals when we were upgrading the machines, which is very difficult to do when you have machines made out of card components like the PDP-1 or the DDP-24 or such.

Hendrie: Yes.

Poduska: So, we got the commonality of usage going up, also upgradability in the field. When people went from the Prime 300 to the Prime 400, we could unplug the 300, plug in the 400. As a matter of fact, we didn't do that very often because we usually would upgrade the back plane too but we could at least preserve the use of the peripherals and that, so we got a lot of upgrade business and upgrade business is very lucrative. I mean margins on upgrade business are 80 percent or something like that.

Hendrie: Okay.

Poduska: So, those kind of things work. Now, on the software side we were surely the first to implement virtual memory, the first to deploy time-sharing systems at that level. You may recall there were some basic, I mean the language systems that you could buy from TRW online but the first full functionality run any language on this system anyway you wanted to was the Prime machine and it had a great appeal to the computer-aided design community as well as the commercial community.

Hendrie: Okay.

Poduska: So that worked out really well.

Hendrie: Good. What sort of languages did you offer on the machine?

Poduska: Well, we started out offering assembly language and Fortran because those were the ones that were available through the Honeywell sources and through the other sources. When Fisher came onboard, Ken, we offered RPG and Cobol but then we quickly extended to all the other algorithmic languages. Now, you may recall in that era there was quite a battle going on between the algorithmic languages of Pascal or C.

Hendrie: Yes.

Poduska: Bell was into the game out there with software tools and the Unix systems and its allies, as well as all the other kinds of languages that were popping up venues. I mean you recall there were a bunch of Lisp machines also.

Hendrie: Yes.

Poduska: That scheme had come into it. So, we actually provided all those languages, PL1. We had a PL1. Bob Freiburghouse did the PL1 for us at Prime.

Hendrie: Okay.

Poduska: So, we had all of those languages there. The ones that really stuck were Fortran and Cobol.

Hendrie: Okay.

Poduska: And, you know –

Hendrie: Those were the big users during the big era.

Poduska: Those were the big users. Fortran on the scientific side and Cobol on the commercial side.

Hendrie: Now, was your operating system written in a higher local language?

Poduska: Yes, Fortran.

Hendrie: Okay, now was that unique or was that very common at that period?

Poduska: I think it was. I think it was unique in that fundamental sense that we were the only ones who were doing it.

Hendrie: Yes.

Poduska: And, in a way it was a tour de force. I mean nobody would claim that Fortran is an ideal language for writing an operating system. It was also in the era when computer resources were scarce enough that people were really worried about efficiency and there's no question that Fortran written operating system code is less efficient than assembler] but the trick was to use it to accelerate code development and use it for the features that Fortran could provide. I mean you really could continue to compile. You really didn't have to trace down the dressing errors. There really was a symbolic location of variables and loop control, even certain levels of recursion and certainly levels of composition of function calls. With all the difficulties that Fortran has, go-to statements, numerical statement labels, peculiar if statements and frequency statements. it is certainly far better than using machine language code. So, it took a little bit of pressure from the top down to get the operating system guys to use it. Once they finally saw that they could do things so much quicker in Fortran and once I'd cut the deal with them that said do it first in Fortran, meter it, and if you can find something with a two-to-one improvement go re-implement it in machine language. Once I made that deal with them, that compromise with them they were willing to do it. Now, you might ask how many times was Fortran code redone.

Hendrie: Of course.

Poduska: Essentially zero. I mean, you know, disk drivers and interrupt controllers, Yes.

Hendrie: Yes.

Poduska: But –

Hendrie: The kind of thing, Yes –

Poduska: Schedulers and –

Hendrie: All the higher level constraints never got rewritten.

Poduska: Right, never.

Hendrie: Just the really grubby.

Poduska: Only the grubby.

Hendrie: Physical stuff way down at a lower level.

Poduska: Right.

Hendrie: That Fortran is least appropriate for.

Poduska: Exactly. I mean people compare it with sharpening knives. I mean if you sharpen two knives you've done them all. I mean there's no sense in doing that then doing another. People get converted but it took a little bit of Tennessee diplomacy to make that stuff work.

Hendrie: Okay, good. When had you become convinced that that was the right thing to do?

Poduska: Oh in NASA days. I mean that's one of the things that, well even before that, MIT days, that's one of the things that was just so clear to me that, and it was almost a religious tenet too, I mean that you can do things in higher level languages and save all the time, then go back and redo.

Hendrie: Yes.

Poduska: Get the concepts down first and then re-implement them.

Hendrie: Okay. So, even if it's just quick prototyping –

Poduska: Right.

Hendrie: -- it still is the right thing to do.

Poduska: But, of course, these things are – I was going to say self fulfilling. Actually, I meant regenerative in the sense that you get skilled at figuring out the things that you could do with Fortran that work efficiently and the things that don't work so efficiently.

Hendrie: Yes.

Poduska: I mean procedure call and loop control are pretty efficient. Complex arithmetic statements maybe not but you don't do a lot of scientific or arithmetic statements in systems language.

Hendrie: Yes.

Poduska: Programming.

Hendrie: Right.

Poduska: So, you avoid those kinds of things and it works out fairly well.

Hendrie: Now, when you started doing your compilers, did you also continue to implement – did you implement Cobol in Fortran or any of the other things besides the outlying system and the assembler?

Poduska: We did. We implemented, we had a basic system. We implemented that in Fortran. We had an RPG system that we implemented in Fortran. The Cobol system I think we did not because I think we bought our Cobol system from a commercial provider.

Hendrie: Okay.

Poduska: That's my memory but my memory is hazy to tell you the truth on that.

Hendrie: Yes. Okay. All right, so the in-house things you continued to use then.

Poduska: Yes, right.

Hendrie: Tell me a little bit about the people at Prime. Did you bring any of the people with you that you had worked with at either Honeywell or the NASA?

Poduska: Yes, as a matter of fact we did. All seven founders of Prime came from Honeywell or NASA. I was the only one that came from NASA out of that. The next layer a lot of the engineering people and the technical marketing people came out of NASA and the MIT connection.

Hendrie: Okay.

Poduska: Dave Uden and Don Clemens and that whole set of folks. Some of them had mutual connections, connection between the two, Mike Sporer is an example. They have both an MIT connection and a Honeywell connection.

Hendrie: Had Greda worked at Honeywell also?

Poduska: Greda had worked at Honeywell but he was also an MIT alum and I had known him through the MIT connection.

Hendrie: Oh, okay.

Poduska: Closely. He's I think two years behind me or something like that.

Hendrie: I see, okay, very good. So, you had a –

Poduska: It was a – but you know none of that should be surprising. The Boston area is a Mecca of the computer industry. It has been for years. It was probably a more singular Mecca at the time and so it's not surprising that people who live in this area would have run across each other in multiple context just like you and I have.

Hendrie: Yes.

Poduska: And all the other folks who have been in that field have too. We keep running across each other, you know. It's like we don't have any other friends. What's wrong with us here?

<tape change>

Hendrie: Let's talk a little bit more about where there are stories involved, because those are always interesting to people. Did you have any particular potholes on the way to getting those first products out at Prime that you remember, things you didn't think were going to happen, didn't plan for, and of course you overcame them because you got them out.

Poduska: The big hardware engineering issue at the time was how do we get all those parts onto a single board? The compromises that existed around that to get all those parts onto a single board. They were challenging engineering questions at the time. They would be child's play today.

Hendrie: Yes.

Poduska: So that, for example, just the question of whether we were going to have a two layer or four layer or six layer printed circuit board to do all the interconnects with components on the board was a significant question, significant not only for the cost of the board but more importantly for the routing time because all these things were hand routed. I mean we didn't have adequate software for doing the routing of the boards. So, it was a significant engineering question for time and time to market which was our primary thing as well as costly and that pushed us in a couple of directions. It made us jump on the flash memory, fusible link memory, not flash memory, the fusible link memory.

Hendrie: Yes.

Poduska: For the micro code very, very quickly and it made us go to more micro code control of things because it takes less silicon space to do it and, you know, it pushed us hard. We, on the other hand, found a solution to all those problems and we were, I guess, a little lucky to do it but we found it. That was the principal engineering side. The software side went like that. We had no significant problems with the software, mainly from the original strategy that we had taken, that is to say we were going to use the Honeywell base software and the operating systems that were around it. We were going to make the hardware compatible enough. You know no hardware is absolutely compatible but we were going to solve the problems of incompatibility by fixing the hardware.

Hendrie: Yes and you had micro codes so you had a consistent strategy.

Poduska: We had the micro codes.

Hendrie: That you could go [inaudible].

Poduska: Right, , exactly and we had this mechanism for doing it. The biggest problem we had along that line you probably more than anybody else should be able to figure this out. This had to do with shift instructions. You may recall that the shift instruction in the Series 16 when you gave a shift count of zero didn't shift anything, did a shift zero but in the Prime machines because of the way the micro code was implemented a shift to zero would actually shift 31 places or 32 places or something.

Hendrie: It took the other approach.

Poduska: Approach and that caused some software incompatibilities but not many and that's what we had to get around.

Hendrie: Okay.

Poduska: So.

Hendrie: Oh, wow. All right. <laughs>

Poduska: I mean it was at that level.

Hendrie: Yes.

Poduska: This minutia level of stuff that we did.

Hendrie: Yes, okay, good.

Poduska: But it worked out well.

Hendrie: Good. On the board what did you end up with? Did you end up with a six layer board or did you end up in the upper range?

Poduska: We ended up with a four layer board.

Hendrie: Okay.

Poduska: And it was hand laid out and ultimately we went to a higher number of layers of the board as we went to higher level processes as much for signal control as anything else because the four layer board was all signal. We ended up with the six layer board with the power and ground in the middle.

Hendrie: Ah.

Poduska: And then higher numbers of levels and such. We got via control going and then –

Hendrie: That sounds very classical.

Poduska: Oh Yes, it is, and of course the printed circuit board technology was advancing rapidly too.

Hendrie: Yes.

Poduska: Propelled by the fact that we were providing computers to people in printed circuit board world. At the time I think the company was Cytech or somebody like that who was doing the printed circuit boards for us. We provided them the computers and got a little bit faster and software got a little better.

Hendrie: Well, they originally had their own special purpose computer if I remember correctly.

Poduska: Yes, they did.

Hendrie: Designed to –

Poduska: They did.

Hendrie: -- give them a competitive advantage.

Poduska: Right.

Hendrie: Of course they weren't able to sell it to anybody else.

Poduska: Right.

Hendrie: But, oh well. All right, I'd forgotten about that.

Poduska: That's a mistake. That's a mistake that's been made by a lot of folks.

Hendrie: Right.

Poduska: You know in a lot of different contexts.

Hendrie: Exactly.

Poduska: Everything from [inaudible] equation machines to list machines to –

Hendrie: Yes, and the list machines.

Poduska: And the list machines, right.

Hendrie: Yes. Okay. Let's see. Do you have any other stories that you might like to get on tape about the time during Prime before we get to the beginnings of Apollo?

Poduska: Well, I think the things I remember the best were the transition stories and the growth. I mean we took over a good part of the buildings in the Newton Industrial Park right there at 120, right across from Walnut Street where Honeywell had been.

Hendrie: Yes.

Poduska: We ended up taking over the carousel site, what was the carousel site [inaudible].???

Hendrie: Right.

Poduska: With about 300,000 square feet of building for engineering.

Hendrie: Yes.

Poduska: So, I mean this growth really went very quickly. We ended up setting up a research team at Prime which mirrored the research organization that I had run for Honeywell in the sense that it was an advanced technology and positioned below pure science but well above applied engineering kinds of things. Dave Nelson ran that.

Hendrie: Yes.

Poduska: And the result of that was that a lot of the ideas that were developed there came to be part of what Apollo was.

Hendrie: Okay.

Poduska: In large measure because Prime could not or would not support that operation.

Hendrie: Okay.

Poduska: An interesting story I think and perhaps an instructive story has to do with the transition from Prime to Apollo. We talked earlier about Bob Baron.

Hendrie: Yes.

Poduska: And the question of events and the growth of the company overtaking Bob. Now, there's other stories like that relates to me leaving Prime and going to Apollo is one that's a little different but similar. The facts are that I was running engineering at the time and Fisher was running the company, Ken Fisher. And, Ken's idea about deployment of the assets of the company was much more oriented towards growing the marketing organization and growing the sales organization and me running engineering I wanted to spend a lot more money and a lot more time and effort in developing product and putting it out in the marketplace. These are two conflicting points of view. They have validity in different organizations one way or the other. It's not so much in retrospect even. It's not so much a question of who is right or wrong. It's a question of which was the company going to go? Ken's the CEO. He gets to choose and for that reason he invited me to leave Prime computer. You know, again, without saying who is right or wrong about things like that because I mean there is so much luck involved in that anyway, in retrospect I view that as the right decision both ways. I stood my ground for what I thought was the right thing to do but Ken stood his ground also for saying that was the right thing for him to do in running the company. And, I should say that Ken Fisher and I are very good friends now. He's the one that sponsored me into the Westin Golf Club as a matter of fact.

Hendrie: Is that right?

Poduska: And we have the same birthday, December 30th.

Hendrie: Oh, my goodness. Now, is he still in Florida?

Poduska: No, no, he lives right down the road here.

Hendrie: And so he's back here? He was in Florida for a while wasn't he?

Poduska: He lives 600 yards this way.

Hendrie: Yes.

Poduska: Well, he goes down there every winter now.

Hendrie:: Ah, okay.

Poduska: But he and Barbara are wonderful people. But, you know, it's one of these transition points in life that there is a difference between what one's opinion about what should be done is and what one's personal view about another person is.

Hendrie: Yes.

Poduska: And for those who might view this tape and have transition issues from company to company, here's an example that I point to that Ken Fisher and I are good, solid friends, always were, might have had our arm wrestling contests but he wanted to do things one way. I wanted to do another. He's the boss. He gets to choose. Apollo sprang out of that. Apollo had its own set of successes. He doesn't begrudge me the success at Apollo. I can't say he was wrong.

Hendrie: For doing what he did.

Poduska: For doing what he did, yes.

Hendrie: He had to choose.

Poduska: He had to choose.

Hendrie: Okay, well let's talk a little bit about what sort of work went on at Prime in the area of networking, particularly local area networking. Prime, I think did some relatively early work in that relative to the other mini computers or at least some of the mini computer –

Poduska: It certainly did. That was, in fact, thank you for reminding me of that because I had forgotten about it frankly. Because it was in the latter days of Prime, well not the latter days of Prime but it was certainly after the initial flurry of activity. The virtual machines, virtual memory, and the virtual machine concept, the operating systems and that. That was the first three, four years. After that, when we were in fact propelled a lot by the fact that we were in the commercial marketplace where we had the network machines.

Hendrie: Yes. When did Ken come to Prime?

Poduska: I would say four or five years into it 1975, 1976, 1977, somewhere in that time frame.

Hendrie: Okay, great.

Poduska: It was, I would say, 1978,1979 that we got into the full networking issue. And, we did it with I guess what would be called a point-to-point interconnection today. It was certainly not Ethernet or token ring kind of stuff.

Hendrie: Yes, I don't think.

Poduska: Although there was a token passing back and forth between the machines to do things and it operated at relatively high speed.

Hendrie: Yes.

Poduska: That is to say ten megabits a second for machines that were local. Local meant building to building. Now, we did that with this enormous co-axe, twin-axe.

Hendrie: Okay.

Poduska: Which is hard to do but we did it. We also had Internet connectivity over phone lines, both asynchronous and synchronous phone lines, so you could do everything you could do over the high performance links you could do over the phone lines, either synchronous, bi-sync at the time or the asynchronous lines essentially a modem.

Hendrie: Yes.

Poduska: Twelve hundred baud modem was thought to be wonderful.

Hendrie: Yes.

Poduska: And so we pointed that out to people, I remember using the example that if you wanted to you could compile a program with the source coming from Cleveland and the object going to Daytona with the machines sitting in Framingham because the operating system was that robust and the networking system was that robust. We were able to share files, able to essentially mount directories from one machine to another, one disk to another. That gave us enormous capabilities for backup and sharing of information from machine to machine. It was partly done in response to the needs of the commercial world but it also certainly enhanced our position in the commercial world because those guys, even then, had tons and tons of archival data.

Hendrie: Yes.

Poduska: You know that they needed to get at quickly, and in different branches and a different machine.

Hendrie: So, I could be running the program on one machine and I could access files that were on another machine.

Poduska: Yes, right.

Hendrie: Without doing handstands.

Poduska: Right in an autonomous way, that is to say that the mechanism you use for accessing a file on your own machine was the same mechanism you use for accessing the file on another machine.

Hendrie: Yes.

Poduska: The same open, close read, write, overwrite, and reposition commands.

Hendrie: Just had to have the right file name.

Poduska: Right.

Hendrie: And the system had to know whether it was in the right path.

Poduska: In the right path to get through.

Hendrie: Okay, so it was a path.

Poduska: Right.

Hendrie: Okay.

Poduska: And there was also a procedure-to-procedure mechanism where you could have procedures on different machines sending –

Hendrie: Calling data.

Poduska: -- data through pipes back and forth to each other.

Hendrie: Okay.

Poduska: We didn't call them pipes but that was the effect of it.

Hendrie: All right.

Poduska: But I don't remember what we called them but I know you can do it.

Hendrie: Okay, very good, and this was all implemented in Prime in some version of Prime.

Poduska: Yes the king of that hill was a fellow named Bill Farr.

Hendrie: Okay.

Poduska: Who shows up every now and again too. He did both the hardware and the software design of that stuff.

Hendrie: Really?

Poduska: Yes.

Hendrie: Very good. Well, he originally designed the console on the DDP 516.

Poduska: Yes, same guy.

Hendrie: Yes. All right, talk to me about when the idea came into your head and where it came from for doing something that, you know, eventually became called a workstation?

Poduska: I told you the story about how Prime happened at least to me, this overnight kind of thing and the story for Apollo is almost parallel to it in a serendipitous nature because I left Prime at the end of the year in 1979. December 31st was my last day and what I fully intended to do was to take three months off and go do some skiing and, catch up with the kids and do all those things.

Hendrie: Do all those things, Yes, that you'd sort of given up.

Poduska: Right.

Hendrie: That you'd sort of not done as much as you wanted to, Yes.

Poduska: During the last few months at Prime, I had taken the opportunity because I was not operating in the executive capacity for the last two or three months or whenever it was. So I took the time to catch up on all the things that were going on, especially all the stuff that was going on in Dave Nelson's shop. The research shop and what was going on in the outside world. I had read one report. I still have a copy of it upstairs somewhere. It's a little thin report, can't be more than 20 pages or so, on just standard university sort of printout. It was called the SPICE Program from Carnegie Mellon and SPICE meant Stored Program Interactive Computer Environment or something like that.

Hendrie: Yes.

Poduska: I don't even know what it was but it described a workstation. It described a machine that was dedicated to the essentials, dedicated to one person.

Hendrie: Yes.

Poduska: With scant attention paid to efficiency of computation, that is efficiency of computer usage. We don't care if the machine sits idle in other words. We want it to be fast but it can have idle time, have a graphics interface, keyboard, mouse, use for a person, typically a technical professional doing interactive computation in the furtherance of their work. That was a seminal document for me. It introduced me to the notion of workstation at a very, you know, at a profound level. So, I was talking and I read that in, oh, I don't know December of 1979 or somewhere in that time frame but still I hadn't thought about putting a company together. I began to write down some ideas. I talked with Dave Nelson about some of the other technologies that were going on, in particular things like token ring networking, graphical interfaces, and the emergence of not the single board computer, but the single chip computer, namely the Motorola 6800 at the time.

Hendrie: Okay.

Poduska: And from that set of ideas we wrote down a little bit of a plan in longhand and then about a week into January, right after New Year's, I get a call from Bill Draper who was a principal of Sutter Hill and he said what are you doing? He had been an investor in Prime by the way.

Hendrie: Okay.

Poduska: He said what are you doing? I said, well I'm beginning to think about things and had some ideas but we haven't thought about them too much yet and he says well why don't you come on out and talk to me about them. So, I got on a plane and on the plane wrote up a business plan in longhand.

Hendrie: Okay.

Poduska: And about the middle of January, the 15th of January, went out and spent an afternoon with him. He asked me what I wanted to do and I said, well, I guess we want to put together the company. He committed to an investment in the company at that time and that's how Apollo was born.

Hendrie: Wow.

Poduska: In the next few weeks we got Helman, Gall and then Venrock involved but Sutter Hill was still the lead venture capitalist.

Hendrie: Now, had you known Joe Gall?

Poduska: I had known Joe Gall vaguely.

Hendrie: Okay.

Poduska: Before that but the people I knew the best were Sutter Hill.

Hendrie: Okay.

Poduska: I mean Dave Anderson, Bill Draper, and Paul Wythes and who was never a principal guy but a senior guy there. So, they were the people I knew. We wanted to get Graylock into it but there was a lot of resistance from the Prime board for Graylock to be involved because they were still so heavily involved in Prime.

Hendrie: Oh, Graylock was involved in Prime, okay.

Poduska: Graylock was involved in Prime.

Hendrie: Yes, obviously you raised more rounds before you went public.

Poduska: And they were later rounds and, in fact, Graylock came into Apollo later on also.

Hendrie: Okay.

Poduska: But the three big guys at the beginning were Sutter Hill, Venrock with Ted McCourtney.

Hendrie: Yes.

Poduska: And Helman and Joe Gall was the principal there.

Hendrie: Did you know Venrock?

Poduska: I did not know Venrock.

Hendrie: So, you got leads. I mean Bill Draper said hey you got to get some East Coast people.

Poduska: Well, I had not been in the venture world at all because I had been running engineering at Prime.

Hendrie: Yes, exactly.

Poduska: So, I didn't know those people but as it turns out Ted McCourtney comes from Memphis, Tennessee, which is my hometown. Then we didn't know each other there but astoundingly his father and my mother knew each other. <laughs>

Hendrie: Oh, my goodness.

Poduska: One of these small world kinds of things.

Hendrie: Exactly.

Poduska: But that's we founded Apollo then in February of that year and we had our first products out the door in another remarkably short period of time, that is in October of 1980.

Hendrie: My goodness.

Poduska: We did a set of things that was similar to what we'd done at Prime, that is to say we used a lot of technology that was available to us. Bell Telephone Laboratories had come out with not only Unix and C, but also a set of things called software tools.

Hendrie: Yes.

Poduska: Which were very similar to the package of C and of Unix, the Unix of that time. We built an operating system around it. We used Pascal as the implementing language.

Hendrie: Oh.

Poduska: Because Pascal is arguably a superior language to C, certainly in that time frame. Of course the clear winner today is C and C++ and C Sharp now beyond that. But at the time Pascal served us really very, very well. We wrote the operating system. There we did everything from scratch. We wrote the operating system, the compilers, Pascal, Fortran, the assembler, and such, using the Motorola 68000 then, which had just come out the door.

Hendrie: Okay.

Poduska: Just come out the door. Now, an interesting thing about that from an engineering point of view and you'll appreciate this is that we were trying to decide how to price and market the machine and we knew that it was going to be performance related. The question is what kind of performance were we going to be able to get out of the 68000.

Hendrie: Yes.

Poduska: They were providing the four megahertz parts, now its four gigahertz

Hendrie: Right.

Poduska: They had promised the six megahertz part and the eight megahertz part was yet to come. The question was what were we going to be able to use in the machine? Now, in today's notion of Intel parts they're escalating an AMD price where they're escalating clock frequencies. It seems obvious that you want to build something that will accept them all. It wasn't nearly so obvious at the time. We decided we would build the machine as if it would handle the eight megahertz part even if we could only get the four. We decided to print multiple sets of collateral information for the marketing people and multiple price lists to take advantage of whatever speed we could get. And, as it turned out, we didn't get entirely lucky but we were able to deliver the six megahertz part in the machine and it really did mean a 50 percent improvement in performance on both the graphics and the performance side.

Hendrie: Good.

Poduska: So, the question of matching the ski slope of technology with the advent of products into the marketplace is one that we took careful note of and I think it contributed a lot to the success of Prime. We had all the organizations combined in that notion too, that is sales, marketing, engineering.

Hendrie: Yes.

Poduska: Manufacturing too.

Hendrie: Yes.

Poduska: The manufacturing. Yes, manufacturing didn't have any actual construction kind of problems but they certainly had procurement issues. I mean it was no small matter to have 1,000 ancient parts sitting on your shelf.

Hendrie: Right, exactly. You had to watch out.

Poduska: Right, so you had to do something about making sure you didn't have dead inventory.

Hendrie: Okay, so you designed the machines to work with any of them.

Poduska: Any speed right.

Hendrie: Make sure. That requires a little work in the clock circuitry and timing and things like that.

Poduska: Yes. On the engineering side it takes that but the most interesting thing is the marketing side. How do you decide how you're going to price things and what marketplace you're going to approach depending on the speed of the machine?

Hendrie: Yes.

Poduska: Because from four to eight really was a two-to-one performance level change.

Hendrie: Yes.

Poduska: And it made a lot of difference to the computer design community, even at that time.

Hendrie: Okay, interesting. Now, who did you take, sort of recruit to be the founders and sort of lead the technical team to go do this?

Poduska: Badge number two was Dave Nelson. He ran the research organization at Prime Computer. I think it's fair to say that a lot of the ideas for the Apollo computer came through Dave. I'm not saying he invented all of them.

Hendrie: Yes.

Poduska: He certainly didn't invent the microprocessor or token ring networking or discover software tools but he did evaluate them. He understood what they were and how they fit in the world and how they could be used to make an effective workstation..

Hendrie: Yes.

Poduska: And he was a good collector of that. Badge three was Mike Grida who ended up running the engineering organization along with Dave.

Hendrie: Okay.

Poduska: We hired people then from Digital, Charlie Specter who became chief operating officer, people from Data General like Bob Antonucco who ran a manufacturing organization, Ed Zander who went later on to be CEO, I think he was CEO of Sun or he was at least second to Scott.

Hendrie: Yes, I think he was second to Scott.

Poduska: I forget what his title was Barry Fidelman who now is running I think Atlas ventures was VP of sales. Bob Bishop ran our international sales organization. He's now CEO of Silicon Graphics, I think.

Hendrie: Yes.

Poduska: And we had a good team.

Hendrie: Had some good people. Had some very good people.

Poduska: Had a fun time.

Hendrie: Good.

Poduska: And the company really ran very rapidly. We went on to have our initial public offering in March or April, I think it was, of 1983. At the time that was quite a short time.

Hendrie: Yes.

Poduska: And, we were at that time, I'm told, the highest valued technical public offering . There were financial organizations that had higher but the highest technical one. I don't think we hold that title to this day.

Hendrie: Right.

Poduska: But we were at the time and you know it would be nice to say that was all skill and preplanned. The facts are that we happened to hit the IPO market right on the nose. We could not have found a better time than the springtime of 1983 to hit that marketplace.

Hendrie: Yes.

Poduska: And with, I don't know, $20 million of revenue we had a $340 million pre-money cap file so we had a secondary the following October where we were up to $800 million.

Hendrie: Just unbelievable.

Poduska: Just very heady stuff and this is 1983. This is not 1993 and 1994. This is 1983.

Hendrie: Well, there was a little peak in1983.

Poduska: You bet there was and we hit it right on the nose.

Hendrie: We just barely made it out into the market at Stratus in September of 1983 and we were – it was nip and tuck. If we had been another month later –

Poduska: Right.

Hendrie: Probably not.

Poduska: Right.

Hendrie: Or two months later.

Poduska: Right.

Hendrie: But anyway, yes, it was a great time.

Poduska: It was a great time.

Hendrie: So, tell me a little bit about, you know, who were the first customers at Apollo?

Poduska: The first – we had a lot of small, one at a time customers because it's the kind of product that lends itself to one at a time customers.

Hendrie: Yes.

Poduska: I mean there was the self contained thing. It could be used by a single person, so we had a lot of academic customers but our first large OEM customers were Auto-trol, the CAD people.

Hendrie: The CAD people, okay.

Poduska: And in large measure the mechanical CAD people, like Auto-trol and GE Calma at the time.

Hendrie: Okay.

Poduska: But then we got quickly into the e-CAD business with Mentor Graphics, for example.

Hendrie: Yes.

Poduska: Then, into the text processing business with, what company was that in Cambridge? I'll think of the name in a moment. But we ended up with a lot of volume sales. We call them OEM sales but they're not OEM in the old sense, in the more traditional hardware sense where you buy a component and it's installed and some other component. But rather, they were workstations which were sold to software people for their incorporation of software and then later delivery to their customer and sometimes they changed the name plate and sometimes they didn't.

HENDRIE: Yes.

Poduska: But it certainly made the later on sales and sales extension of the product very easy because there's only one place to go to get another Apollo computer to increase your productivity and buy another license from Mentor or from Calma.

Hendrie: Right, yes.

Poduska: So that worked out really quite well and, in fact, that part of the business grew much faster than the individual workstation part of the business as you would expect because there's just so much more volume there.

Hendrie: Yes.

Poduska: We had one competitor at the time who was called Three Rivers and they had a system called the Perq system, P-E-R-Q I think.

Hendrie: Yes.

Poduska: And it was an acronym meaning something or other who had sprung out at least in part from Carnegie Mellon. I don't know the full pedigree of that company.

Hendrie: Yes.

Poduska: But they did.

Hendrie: But it was located in Pittsburgh?

Poduska: I believe it was, yes.

Hendrie: I believe it was too.

Poduska: And there is at least a nexus of some sort. I think Ed Fredkin was involved in that company.

Hendrie: Hum, okay.

Poduska: As an investor or principal of some sort or other. I'm not absolutely sure of that. But, in any case, it was a bit of a slugfest but we managed to win and we won I think for some business skills and broad backing from our investor community which helped us an awful lot. I mean Sutter Hill and Venrock, in particular, and Warren Helman of Helman Gall are names that bring attention in the community and we had West Coast, East Coast, and New York.

Hendrie: Right.

Poduska: So, we were in pretty good shape there. But there was also the technical thing which was a lesson that I relearned and that is that the single part processors once, like the 68000, once they grab a toehold have a much more accelerated technology path than building processors out of even – out of discreet components, not discreet, but smaller ICs and even though they might even have very high performance by themselves.

Hendrie: Yes.

Poduska: So part of the machine was built out of the AMD 2901s and 2903.

Hendrie: Ah, okay.

Poduska: Which are great parts.

Hendrie: Yes.

Poduska: And ASIC [Application-specific integrated circuit] had followed but they could not keep track with the performance gains that were coming out of the single part processors like the 68000. There was a 6801 or 68001 or whatever that thing was called.

Hendrie: Yes.

Poduska: Units up to 16 or 32 megahertz in 18 months. Well, 32 megahertz over four, that's about eight-to-one.

Hendrie: Yes.

Poduska: And, you know, that's really hard to do with discreet components and it went on from there.

Hendrie: Yes.

Poduska: And that lesson was one that helped us at Apollo and, in fact, it's the converse of that which hurt us so badly at Stellar later on.

Hendrie: Okay.

Poduska: Isn't it wonderful how hindsight –<laughs>

Hendrie: Yes, some things, they were there and it's really hard to observe them.

Poduska: And, of course, it could have gone the other way.

Hendrie: It could have gone the other way.

Poduska: Because if you look at the graphics market, graphics processors are all custom components.

Hendrie: Right.

Poduska: And their performance has moved along at a very good clip.

Hendrie: Yes.

Poduska: So, I mean, you know, who knows? That's why they have horse races.

Hendrie: Tell me a little bit about some of the potholes in terms of getting Apollo sort of getting the technical side of getting Apollo machine up and running or did it all just go very smoothly?

Poduska: The initial part of it went remarkably smoothly. We kicked off in February. We had the product out the door in October. And while we had some quality issues, it worked. I should really call them incomplete engineering issues. I mean we kicked the product out the door very, very quickly.

Hendrie: Yes.

Poduska: And so we had all sorts of software bugs and hardware bugs and stuff like that so we got through those in a relatively big hurry.

Hendrie: Yes.

Poduska: I think partly because OEM became so important to us or, at least, volume sales, which meant that we could make the same thing, you know, over and over and over again. Prime Computer, of course, had very much more customized equipment. I mean somebody wanted a tape driver here or a card reader there or this much memory or that much memory and we did a lot of reconfiguration of Prime stuff, not with Apollo's. Apollo's you could –

Hendrie: Pretty much cookie cutter.

Poduska: You could put them in a box and ship the boxes and that was what it was going to be.

Hendrie: Okay.

Poduska: So, we got through those things very quickly. The problem we had at Apollo in expanding the marketplace was to try to figure out which direction to go.

Hendrie: Okay.

Poduska: The first workstation came out with a price range of about 40K or 50K.

Hendrie: Yes.

Poduska: And the question is should we go upstream to a more powerful machine in the 80K kind of level, again with the droop of pricing as technology marches on, or should we go to the lower end and introduce a machine 15K to 18K? Well, we ended up doing both and the question was the timing. We decided to go upstream first because it seemed like people were crying for more processing power, especially in the CAD marketplace.

Hendrie: Yes.

Poduska: And then we went downstream with a single box computer, I guess in concept a great deal like the current Macs or e-machines, that is to say the processor and the monitor and all that stuff were in one box and the keyboard was separate, called the DN 200 I think it was called.

Hendrie: Now, what was the upscale machine?

Poduska: Upscale, I forget what it was called. I think it was called the 400D and 400, something like that.

Hendrie: Okay, Yes.

Poduska: And we had a lot of nomenclature. You know we had a lot of nomenclature for the machines that would vary from the volume produced machines to the end user machines, so I mean I'm not sure I could tell you even with the scorecard in front of me exactly which one is which.

Hendrie: Okay, I understand.

Poduska: But at least we did that. The low end machine turned out to be quite a thundering success. I mean you really could put those in a box and ship them out and cut them out with a cookie cutter and we found out there was a great university environment for them.

Hendrie: Oh, okay.

Poduska: So we sold a lot of machines. We and Apple went together to sell to the University of Michigan and Apple sold them 3,000 machines and we sold them 300. It was a ten-to-one ratio that they put in a gigantic network. Of course then the networking capabilities of the Apollo came in because they became then sort of cluster point – not cluster points, node points for connecting the Apple machines.

Hendrie: Ah, okay, very good.

Poduska: Yes, we had some interesting things like that.

Hendrie: Excellent.

Poduska: But it was remarkable smooth sailing.

Hendrie: Okay. Yes, no giant headaches.

Poduska: No, none like that. I mean the biggest headaches in growth then came over from the sheer numbers.

Hendrie: Yes.

Poduska: I mean its one thing to put out 300 machines in a quarter. It's another thing to put out 3,000 machines in a quarter.

Hendrie: Oh, yes.

Poduska: And we built plants up in Exeter, New Hampshire, down in Puerto Rico. We got up to when I left 4,700 people or something like that.

Hendrie: Oh.

Poduska: It was a big operation.

END OF INTERVIEW

